Pose-Free Structure from Motion Using Depth From Motion Constraints

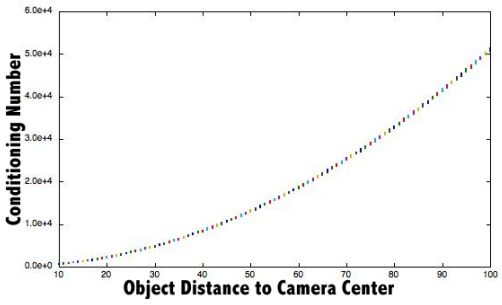
Ji Zhang†, Mireille Boutin††, *Member, IEEE,* and Daniel G. Aliaga*

***Abstract*—Structure from motion (SFM) is the problem of recovering the geometry of a scene from a stream of images taken from unknown viewpoints. One popular approach to estimate the geometry of a scene is to track scene features on several images and reconstruct their position in 3D. During this process, the unknown camera pose must also be recovered. Unfortunately recovering the pose can be an ill-conditioned problem which, in turn, can make the SFM problem difficult to solve accurately. We propose an alternative formulation of the SFM problem with fixed internal camera parameters known a priori. In this formulation, obtained by algebraic variable elimination, the external camera pose parameters do not appear. As a result, the problem is better conditioned in addition to involving much fewer variables. Variable elimination is done in three steps. First, we take the standard SFM equations in projective coordinates and eliminate the camera orientations from the equations. We then further eliminate the camera center positions. Finally, we also eliminate all 3D point positions coordinates, except for their depths with respect to the camera center, thus obtaining a set of simple polynomial equations of degree two and three. We show that, when there are merely a few points and pictures, these "depth-only equations" can be solved in a global fashion using homotopy methods. We also show that, in general, these same equations can be used to formulate a pose-free cost function to refine SFM solutions in a way that is more accurate than by minimizing the total reprojection error, as done when using the bundle adjustment method. The generalization of our approach to the case of varying internal camera parameters is briefly discussed.**

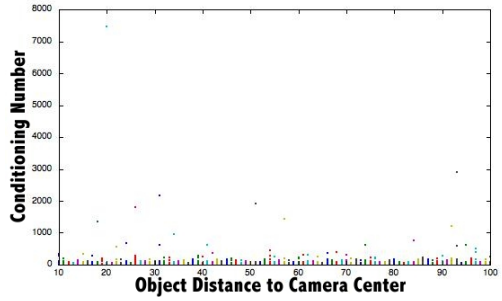
I. Introduction

A core challenge of today's computer technology is to be able to accurately simulate large 3D environments that contain complex structures. On the one hand, it is very costly to set up an experiment that will provide enough precise data to be able to reconstruct the 3D structures accurately. On the other hand, manually creating a 3D model is time consuming. So there is a great need for a simple and low-cost automatic system that would be able to acquire the photogrammetric information of the scene (surface texture, color, reflectance, etc.) and to virtually recreate its effects. Recreating the photogrammetric effects relies, in parts, on precisely knowing the shape and position of the surfaces contained in the scene. In other words, one needs to know the geometry of the scene to be able to model it in a realistic fashion.

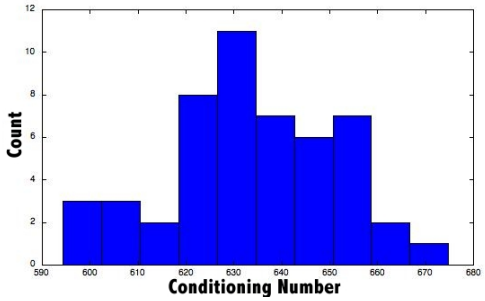
 This research was funded in parts by NSF grant 0434398. †jeffrey_zhangji@yahoo.com, Bloomberg LP, ††mboutin@purdue.edu, School of Electrical and Computer Engineering, Purdue University, *aliaga@cs.purdue.edu, Department of Computer Science, Purdue University.



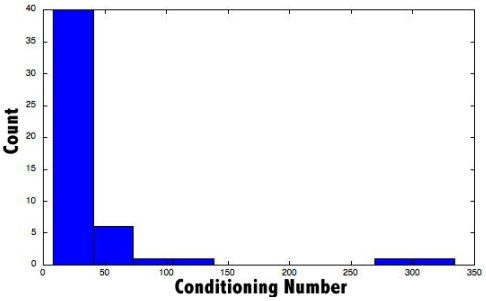
(a) Conditioning of BA



(b) Conditioning of depth-only SFM



(c) Conditioning for BA at $d = 10$



(d) Conditioning for depth-only SFM at $d = 10$

Fig. 1: Conditioning of the SFM problem.

SFM formulation	No. of variables	Comp. Time
Standard (Eq. 1)	36	$>$ 2 weeks
Angle-free (Eq. 8)	27	39m25s
Pose-free (Eq. 11)	24	3h35m16s
Depth-and-pose-only (Eq. 22)	21	13m18s
Depth-and-translation-only (Eq. 23)	12	12m44s
Depth-only (Eq. VIII-A)	9	1m49s

TABLE I: **Computation time comparison of homotopy-based solution for different SFM formulations.**

A common way of obtaining the geometry of the scene is to acquire images using a camera and to track features on these images. In theory, given enough observations of each tracked features and assuming these are generic, their 3D position (along with the camera poses) can be recovered. Indeed, one can write down a system of equations relating the tracked feature positions to their images. In a generic situation and with enough pictures/points, this system has a unique solution (up to an unknown global rigid motion and rescaling of the scene in the case of a camera with fixed internal parameters.) We say *in theory* because, in practice, accurately computing the position of the 3D features, the so-called problem of structure from motion (SFM), is very difficult. While many numerical schemes have been proposed for SFM, they often display numerical instabilities. One important source of these instabilities is the presence of the unknown camera pose parameters inside the equations to be solved. In particular, the set of possible 3D feature positions can change drastically when a small change in the camera orientation is made. So the camera orientation must be very precisely determined in order to get an accurate estimate of the 3D position. But it is often impossible to accurately compute the camera orientation from the pictures. For example, experimentation has shown that a translation along the x axis of the camera plane can easily be confused with a rotation along the y axis of the camera plane. Error analysis of this phenomena have been carried out by several authors (e.g., [1] and [25]). In particular, it was shown in [8] that the two most popular classes of 3D motion estimation algorithms (those optimizing the epipolar constraint from point correspondence and those minimizing the negative depth based on the normal flow) cannot, in general, distinguish between rotation and translation. This is an inherent problem related to the shape of the cost function that is being minimized, and thus is not related to the specific numerical algorithm used for optimization.

One way to try to improve the accuracy of a computation is to overconstrain the solution. In the case of SFM, one can attempt to obtain a better estimate of the 3D feature positions by considering a large number of pictures simultaneously. The method of *bundle adjustment* (BA) is a refinement step for SFM which allows one to do so in a global fashion. (We shall describe this method more in details in the next section.) The term *bundle* refers to the ray of light linking the 3D tracked features to the camera center. Adjusting the bundle is accomplished by minimizing a cost function that quantifies the total reprojection error of the tracked features. This is done by iteratively improving the guesses for the 3D feature positions and the camera pose parameters. Typically, all the pictures available and all the features tracked are used for this step. In practice, when the initial guess is close to the solution, BA indeed often improves the SFM solution. Unfortunately, in general the computations may diverge or fail to converge in a reasonable number of iterations. Even when the algorithm converges, the accuracy of the solution obtained may not be satisfactory. Experiments have shown that the total amount of motion (rotation and translation) between the pictures is the most important factor in being able to recover structure accurately [13]. For example, if all the pictures are taken from the same side of the object and with a similar camera angle, it is difficult to accurately recover the depth of each object point with respect to the camera center. This is because, in the case of SFM solutions that estimate all external motion parameters, the projections of the translational and rotation errors on the image are perpendicular to each other, and the rotation around the Z axis has the least amount of ambiguity [8]. Thus, in order to resolve the ambiguity in the object point positions through bundle adjustment, the Z axes of the pictures taken into account should not be too similar to each other. This will be investigated in more details in Section IV.

One obvious solution to improving the conditioning of SFM would be to remove the camera pose parameters from the problem. For example, one could think of simply measuring the camera pose. Unfortunately, accurately measuring the pose, especially the camera orientation, is difficult and requires a nontrivial experimental setup. An alternative solution, which is the one we are pursuing in this paper, is to algebraically remove the pose estimation problem from SFM, that is to say, to manipulate the set of equations that need to be solved for SFM until all camera pose parameters have been eliminated.

Algebraically removing variables from a set of equations amounts to projecting the constraints they define onto lower-dimensional subspaces. For the case of polynomial equations, systematic elimination techniques based on the concept of Gröbner bases have been developed and implemented in symbolic computation software such as MacCaulay [9], Singular [10], and Magma [4]. But the computational complexity of these algorithms remains a major obstacle, especially in the case where the coefficients of the polynomials are unspecified.

The theory behind these algebraic tools was used in [24] to recast several SFM problems in a simpler setting. For example, by counting dimensions, they obtained constraints on the number of points tracked and the number of pictures that guarantee that a zero-dimensional solution to the SFM problem exists. They also used Gröbner bases to show that, with seven tracked feature points and two views, each of the 3D features coordinates can be obtained by solving a third order polynomial in one variable (yielding at most three possible solutions, in accordance with [7]). But to the best of our knowledge, these polynomials are not known in their general form. For a 2D world, however, Tomasi and Shi [22] [21] proposed a pose-free formulation of SFM which exhibits good noise immunity. This formulation is in terms of tangent of angles, thus not polynomial, and the extension to 3D is, as stated by the authors "technically less than straightforward".

As we shall show in the following, it turns out that pose-free formulations of SFM with fixed internal camera parameters can be obtained using some basic algebraic manipulations along with some simple facts from invariant theory, instead of

Gröbner bases. The idea of using invariant theory for parameter elimination was initially suggested by Bazin and Boutin in [3]. In this paper, we carry this approach further in order to obtain a fully pose-free formulation of SFM (theirs only removed angles) while maintaining a low-degree polynomial formulation (theirs was in terms of rational functions). We go even further and eliminate several object points coordinates as well, conserving only the depths of the object points with respect to the each camera center. This has the effect of significantly improving the conditioning of the SFM problem. Indeed, as we show in the following, our equations in the depth parameters can be used to formulate a cost function that can be used to refine solutions obtained with other SFM methods. Our experiments indicate that the refined solution is significantly more accurate than if it had been refined by minimizing the total reprojection error. However, like total reprojection error minimization, our method can be sensitive to point mislabeling, so for practical application, an implementation following a RANSAC framework would probably be best.

Another effect of our proposed elimination is to significantly reduce the number of variables involved in the problem, and thus its complexity. In particular, this allows us to solve small size SFM problems directly (i.e., without any initial guess for the solution, and obtaining all solutions at once) using homotopy methods. While fully numerical approaches are perhaps faster than approaches involving symbolic computations for solving SFM problems in practice, this this opens up the possibility to analyze small SFM problems (e.g., degeneracy, number of solutions, etc.) including non-generic ones.

Other authors have successfully used symbolic-numerical technique to solve SFM-related problems with fewer variables. For example, a three-view triangulation was obtained in [6] by computing a Groebner basis so to rephrase the problem as a joint-eigenvalue problem. A similar approach is used to solve for the pose parameters between two views in [20]. The same problem can also be solved by using a Groebner basis to eliminate all but one variables from a given SFM problem (i.e., with numerical coefficients- as opposed to variable in our case), thus obtaining a degree 13 polynomial in a single variable [17]. Other work is focused on the estimation of other camera parameters (e.g., [15]). However, our work appears to be the first where symbolic-numerical techniques are shown to be effective for recovering the object in a pose-free fashion.

This paper is organized as follows. We begin by stating the standard mathematical formulation of the SFM problem in Sect. II. The Bundle Adjustment (BA) method, in relation to our proposed new SFM formulation, is summarized in Sect. III. In Sect. refsection:conditioning, we demonstrate the extent to which conditioning issues are common using numerical experiments and show that the pose-free method we are about to propose addresses this problem. In Sect. V, we eliminate the camera rotations from the SFM equations and obtain a camera-angle free formulation of SFM. In Sect. VI, we further eliminate the camera centers to obtain a fully pose-free formulation of SFM. The coordinates of the object points are eliminated in Sect. VII, yielding our new proposed SFM formulation in terms of the depths only. A global solution method based on homotopy for our SFM formulation in terms of depths is discussed in Sect. refsection:homotopy, including a comparison of the complexity associated with other SFM formulations within the context of homotopy methods. In Sect. IX, we use our SFM formulation in terms of depth to propose a (external camera) pose-free cost function for refining initial guesses for the SFM problem. A statistical interpretation of our approach is given in Sect. X and the generalization to the case where the internal parameters of the camera vary is discussed in Sect. XI. We conclude in Sect. XII.

II. Structure from Motion (SFM)

Denote by N the number of features tracked on a sequence of images numbered from 1 to J. Let $P_1, \ldots, P_N \in \mathbb{R}^3$ represent the 3D coordinates of the feature points, and let $(x_{1j}, y_{1j}), \ldots, (x_{Nj}, y_{N,j})$ represent the 2D coordinates of their projection on image j, for $j = 1, \ldots, J$. The relationship between the 3D features and their projection on the images can be written as

$$\begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} = c_{ij} M_j \begin{pmatrix} P_i \\ 1 \end{pmatrix}, \text{ for all } i, j, \tag{1}$$

where M_j is a 3-by-4 matrix (called the projection matrix) containing the camera parameters for the j-th image, and c_{ij} is a positive real number representing the depth of feature i from image j. One can write M_j as the product of two matrices

$$M_j = K_j B_j,$$

where K_j is a 3-by-3 matrix containing the internal camera parameters, and $B_j = (R_j, -R_j C_j)$ is a 3-by-4 matrix containing the external camera parameters: R_j, a 3D rotation matrix, and C_j, the camera center position for the the camera center of the j-th image. When the camera is internally calibrated, one can assume that K_j is the identity. We shall make this assumption from now on, until generalizing to the projective camera case in Section XII.

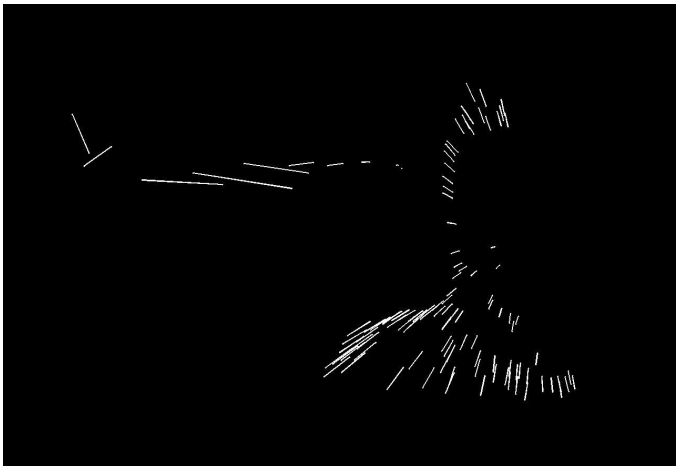
In SFM, one attempts to compute the feature coordinates P_i's given the coordinates of the projections p_{ij}'s. In Equations 1, the unknowns are the 3D points coordinates P_i's, the projection matrices M_j's and the depth constants c_{ij}'s. Observe that all $3NJ$ equations contained in this set are invariant under a simultaneous rigid transformation and rescaling of the 3D features. Indeed, if $P_1, \ldots, P_N, c_{11}, \ldots, c_{NJ}$ is a solution of Equation 1, then $\lambda(RP_1 + T), \lambda(RP_2 + T), \ldots, \lambda(RP_N + T), \frac{c11}{\lambda}, \ldots, \frac{c_{NJ}}{\lambda}$ is also a solution, for any 3D rotation R, any 3D translation T, and any positive number λ. Therefore, one can only reconstruct the geometry of the scene up to an unknown rotation, translation and rescaling. This means that we could arbitrarily fix seven of the unknown parameters and solve for the remaining $3N + 6J + NJ - 7$ unknowns.

In a generic situation, Equations 1 form an overdetermined system of constraints for these unknowns when N and J are big enough. Because of measurement errors and floating point arithmetic, it is impossible to satisfy all equations simultaneously. So one seeks an approximation of a solution:

$$\begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} - c_{ij} M_j \begin{pmatrix} P_i \\ 1 \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \text{ for all } i, j. \tag{2}$$

III. The Bundle Adjustment Method (BA)

First proposed in the context of photogrammetry by Brown [5], the bundle adjustment method (BA) was popularized by



(a) Initial solution guess using the Eight-Point Algorithm.

(b) Solution refined using proposed depth-only equations.

Fig. 2: **Giraffe reconstruction using depth-only equations**.

Hartley [11] and Triggs et al. [23] in the computer vision community. BA consists in solving for all the 3D tracked feature positions and all the camera pose parameters simultaneously by minimizing a cost function. The cost function typically used is the sum of the squares of the distances between the reprojections of the 3D reconstructed feature points and the observed projections, a quantity called the *total reprojection error*, i.e.

$$C\left(M_1,\ldots,M_J,P_1,\ldots,P_N,c_{11},\ldots,c_{NJ}\right) = \sum_{i,j}\left\|\begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} - c_{ij}M_j\begin{pmatrix} P_i \\ 1 \end{pmatrix}\right\|^2, \quad (3)$$

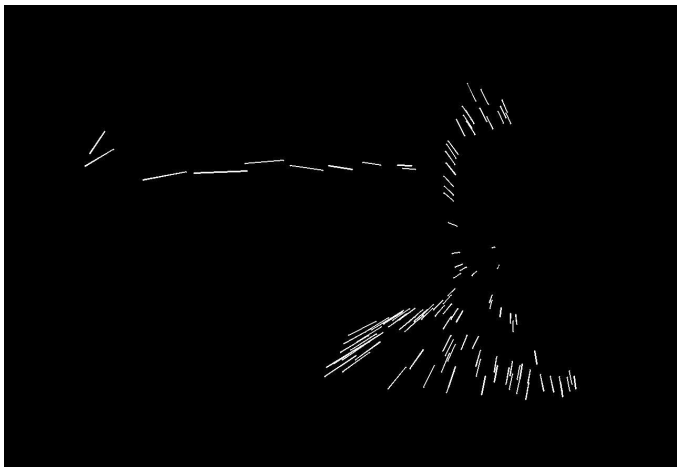
where $\|\cdot\|$ represents the L_2 norm. Note that we view the constant c_{ij} as variables on which the cost function depends explicitly, even though they are directly dependent on the other unknowns P_i and M_j.

As one can see, this cost function is the sum of the squared norms of the left-hand-sides of Equations 2. So one could attempt to solve this problem using a solution method for polynomial equations.But the number of variables involved in this problem makes this approach too slow to be effective, as will be demonstrated in Section VIII. Instead, the minimization is typically performed numerically using the Levenberg-Marquardt minimization algorithm, as proposed by Hartley in [11]. Even though the computational cost of this approach is fairly high given the number of variables to optimize, it is manageable. Moreover, it can be reduced significantly by exploiting the sparse structure of the problem, as in [16].

Given a good initial guess, BA can be quite accurate, much more so than any other SFM method currently available. So, in practice, BA is almost always applied to the results obtained with other methods, as a last refinement step. But as we mentioned previously, the numerical problems created by the need to estimate the pose are observed in this approach as well, since the camera pose parameters are an intrinsic part of the equations to be minimized. In particular, when all the views are taken from the same side of the object and with a similar camera angle, BA has difficulty recovering the depth of the object points accurately. This can be better understood by studying conditioning of the BA problem, which in do in the next section.

IV. Numerical Conditioning of SFM

As stated in the introduction, the numerical conditioning of the traditional formulation of the SFM problem can be poor. Our claim is that removing the camera pose parameters from



(a) Solution of Figure 2 a) refined using total reprojection error minimization.

(b) Top view of a).

Fig. 3: **Giraffe reconstruction by reprojection error minimization.**

the problem generally improve its conditioning. In the following, a *depth-only* formulation of SFM problem which does not include any (external) camera parameters, will be derived. In order to justify the need for this new SFM formulation, we first show that situations for which the condition number of BA is undesirably high are quite common.

To show this, we placed the camera center at the origin and generated objects consisting of 5 points drawn uniformly at random within a unit cube centered at $(0, 0, d)$, for $d = 10, 11, \ldots, 1000$. The variable d is viewed as the "distance" from the object to the first camera center. Fifty objects were generated for each distance and two pictures were taken for each object. The first picture was taken with the object in its initial position, and the second picture was taken after translating the object by a vector $(0.5, 0.5, 0.5)$ and rotating it by $\pi/4$ radians in the x-y plane. We evaluated the condition number of the Hessian matrix of the total reproduction error (Equation III) for this pair of pictures at the true object coordinates and camera parameters. The results are plotted in Figure 1 a). As one can observe, the condition number increases in a more or less quadratic fashion as the distance of the object to the first camera center increases. The histogram of the 50 condition numbers obtained for the smallest distance ($d = 10$) is plotted in Figure 1 c): as one can see, the condition number varies between 590 and 670. Similar results were obtained when the camera orientation between the two views was drawn uniformly at random between zero and $\pi/4$.

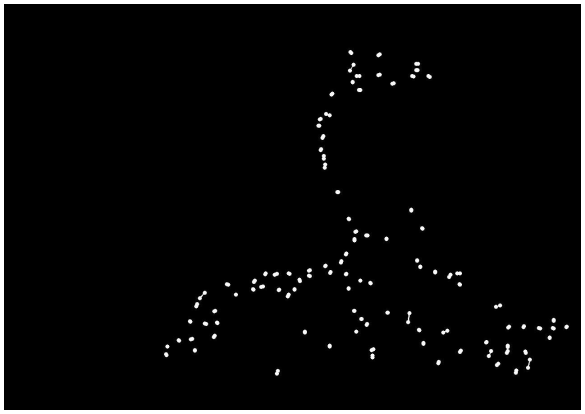
For comparison, we obtained the condition number of the Hessian matrix for the least squares version of one of the depth-only formulations we shall propose later (the first ten equations of (VIII-A)). As one can see in Figure 1 b), the condition number appears to be independent of the distance to the object, and is significantly less than for the standard SFM formulation; the average condition number for our proposed formulation is about 31, while that of the standard form of the SFM problem is more than 1.9×10^4. The histogram of the 50 condition numbers obtained for the smallest distance ($d = 10$) is plotted in Figure 1 d): as one can see, the vast majority of the condition numbers obtained were below 50, although for a few exceptional objects, they were in the 300 range. Such a variation in the condition number is to be expected, as the conditioning is also influenced by the closeness of the points of the object, and our random object selection procedure did not put any constraint on the distance between the object points. But overall, the improvement on the conditioning provided by our method is quite significant. Note that similar results were obtained when the camera orientation between the two views was drawn uniformly at random between zero and $\pi/4$.

V. Orientation-free SFM

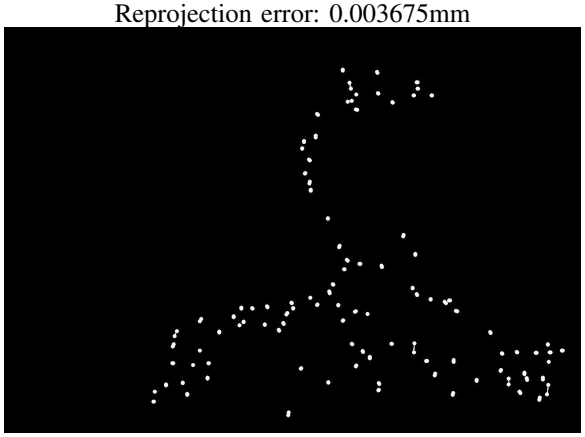
In this section, we eliminate the rotations R_j from Equations 1 under the assumption that $M_j = (R_j, -R_j C_j)$ (i.e., when the internal camera parameters are fixed). This will provide us with an equivalent formulation of the problem of SFM which allows us to solve for the object points and the camera centers without solving for the camera angles. If needed, the camera angles can be recovered a posteriori.

Let us divide Equation 1 by c_{ij}, and let $\gamma_{ij} = \frac{1}{c_{ij}}$. We have

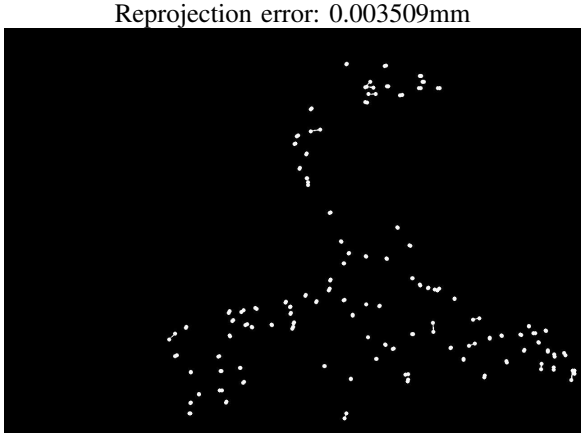
$$\gamma_{ij} p_{ij} = M_j P_i = R_j P_i - R_j C_j, \text{ for all } i, j, \qquad (4)$$



Reprojection error: 0.003675mm



Reprojection error: 0.003509mm



Reprojection error: 0.004219mm

Fig. 4: **Illustration of the small reprojection error of Fig. 3**

where $p_{ij} = (x_{ij}, y_{ij}, 1)^T$. In other words, the vector $P_i - C_j$ and the vector $\gamma_{ij} p_{ij}$ are related by a rotation R_j. One can observe that this is an action of $SO(3)$, the group of rotations in 3D, and use the invariants of this group action to eliminate the R_j's, in a similar fashion as in [3]. Alternatively, one can also eliminate the $R_j's$ by simple algebraic manipulation. For simplicity, this is the approach that we shall take. There are many ways to do this. For example, for any i and $\bar{i}$ among $1, \ldots, N$, observe that we have

$$\gamma_{ij} p_{ij} = M_j \begin{pmatrix} P_i \\ 1 \end{pmatrix}, \qquad \gamma_{\bar{i}j} p_{\bar{i}j} = M_j \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix}.$$

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

$$\gamma_{ij}\gamma_{\bar{i}j} p_{ij} \cdot p_{\bar{i}j} = \left(P_i^T, 1\right) M_j^T M_j \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix},$$

$$= \left(P_i^T, 1\right) \begin{pmatrix} R_j^T \\ -C_j^T R_j^T \end{pmatrix} (R_j, -R_j C_j) \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix},$$
$$= \left(P_i^T, 1\right) \begin{pmatrix} \mathbb{I}_{3\times 3} \\ -C_j^T \end{pmatrix} (\mathbb{I}_{3\times 3}, -C_j) \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix},$$
$$= (P_i - C_j) \cdot (P_{\bar{i}} - C_j).$$

We thus obtain the camera-orientation-free equations:

$$\gamma_{ij}\gamma_{\bar{i}j} p_{ij} \cdot p_{\bar{i}j} = (P_i - C_j) \cdot (P_{\bar{i}} - C_j), \text{ for all } i, j. \quad (5)$$

Observe that this equation can also be obtained using a geometric argument, as the dot product between the two vectors is unchanged under a simultaneous rotation of both vectors.

Another way to remove the matrices R_j's from the equations is to take three pictures, say i, $\bar{i}$ and $\tilde{i}$, and to observe that the volume spanned by the three corresponding picture points satisfies, by Equation 4,

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

Since the volume spanned by three vectors $v_1, v_2, v_3 \in \mathbb{R}^3$ is the same as the volume spanned by Rv_1, Rv_2, Rv_3, for any rotation $R \in SO(3)$, the right-hand-side of the above equation can be replaced by

$$R_j^T M_j \begin{pmatrix} P_i \\ 1 \end{pmatrix} \cdot R_j^T M_j \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix} \times R_j^T M_j \begin{pmatrix} P_{\tilde{i}} \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} \mathbb{I}_{3\times 3}, -C_j \end{pmatrix} \begin{pmatrix} P_i \\ 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbb{I}_{3\times 3}, -C_j \end{pmatrix} \begin{pmatrix} P_{\bar{i}} \\ 1 \end{pmatrix} \times \begin{pmatrix} \mathbb{I}_{3\times 3}, -C_j \end{pmatrix} \begin{pmatrix} P_{\tilde{i}} \\ 1 \end{pmatrix}$$
$$= (P_i - C_j) \cdot (P_{\bar{i}} - C_j) \times (P_{\tilde{i}} - C_j). \quad (6)$$

We thus obtain another set of camera-orientation-free equations:

$$\gamma_{ij}\gamma_{\bar{i}j}\gamma_{\tilde{i}j} p_{ij} \cdot p_{\bar{i}j} \times p_{\tilde{i}j}$$
$$= (P_i - C_j) \cdot (P_{\bar{i}} - C_j) \times (P_{\tilde{i}} - C_j), \text{ for all } i, j. \quad (7)$$

Putting together Equations 5 and V, we obtain the following system of equations, where no rotation matrix appears:

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

for all $i, \bar{i}, \tilde{i} = 1, \ldots, N$ and all $j = 1, \ldots, J$. However, this system contains some obviously redundant equations. 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 a smaller system of equations, such as

$$\begin{aligned} \gamma_{ij}\gamma_{1j} p_{ij} \cdot p_{1j} &= (P_i - C_j) \cdot (P_1 - C_j), \\ \gamma_{ij}\gamma_{2j} p_{ij} \cdot p_{2j} &= (P_i - C_j) \cdot (P_2 - C_j), \quad (8) \\ \gamma_{ij}\gamma_{1j}\gamma_{2j} p_{ij} \cdot p_{1j} \times p_{2j} &= (P_i - C_j) \cdot (P_1 - C_j) \\ &\quad \times (P_2 - C_j), \end{aligned}$$

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

We claim that, for $N \geq 4$ and wherever there exists i_0, j_0 such that $P_{i_0} - C_{j_0}, P_1 - C_{j_0}, P_2 - C_{j_0}$ are not coplanar, Equations 8 forms a *complete set* of camera-orientation free equations, in the sense that solving this system 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. The proof, which already appeared in [26], is reproduced in Appendix I for completeness.

Note that other authors have exploited the idea of using camera-orientation-free SFM equations, but never a complete set. For example, the cosine of the angles used in the equations of the *pyramid method* [27] can be obtained by taking the ratio of some of Equations contained in our system. However, the latter does not form a complete set since the degree-three equations contained in our system cannot be recovered from the pyramid method equations. In other words, the initial system describing SFM prescribes constraints that are not encoded in the pyramid method equations.

Note also that while Equation 8 forms a complete set, it does not contain any redundant equation (i.e. it is a minimal set) in the case where $J = 2$. But while minimality is interesting from a theoretical perspective, in practice the asymmetric role played by the different picture points may cause some numerical problems. In certain circumstances, it may thus be preferable to symmetrize this system with respect to the point indices before solving it. More precisely, more equations can be obtained by replacing the first and second point indices by other point indices in order to insure that $i = 1$ and $i = 2$ do not play a more significant role than the other i's.

VI. Pose-free SFM

Having eliminated the camera angles from the SFM equations, we will now eliminate all the camera center coordinates C_j's. One can do this using invariant theory by looking at the SFM equations for each picture index j,

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

as describing an action of the special Euclidean group parameterized by R_j and C_j. The fact that there exists R_j and T_j mapping each P_{ij} to its corresponding $\gamma_{ij} p_{ij}$ implies that each j^{th} point configuration $(\gamma_{1j} p_{1j}, \gamma_{2j} p_{2j}, \ldots, \gamma_{Nj} p_{Nj})$ is in the same orbit as the point configuration $P_1, P_2, \ldots, P_N$. The fundamental invariants of the diagonal action of the group of rotations and translations in $\mathbb{R}^3$ (which are are well known [18]) thus lead to pose-free SFM equations.

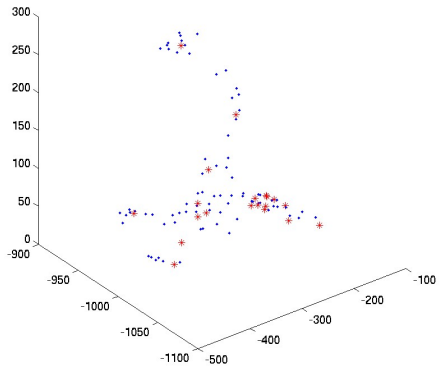
Alternatively, one can also obtain the same result through some basic algebraic manipulations. For example, consider the following equations from Equation 8:

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

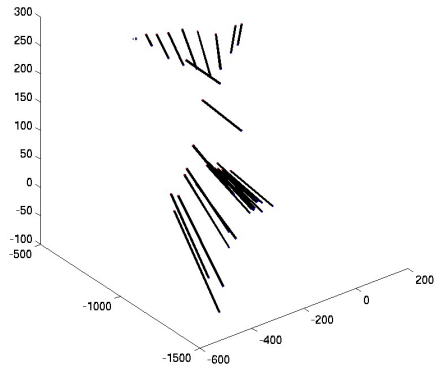
Observe that

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

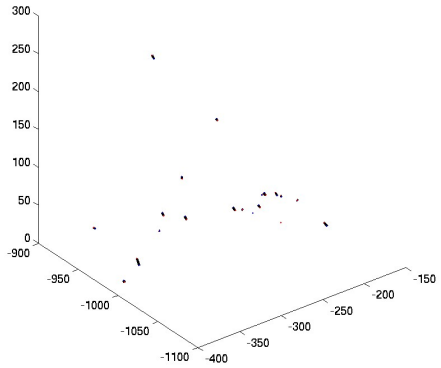
In the last expression, we have the terms $(P_i - C_j) \cdot (P_1 - C_j)$ and $(P_1 - C_j) \cdot (P_1 - C_j)$, which are respectively the right-hand-side of The fourth and the first equation of (9). The value



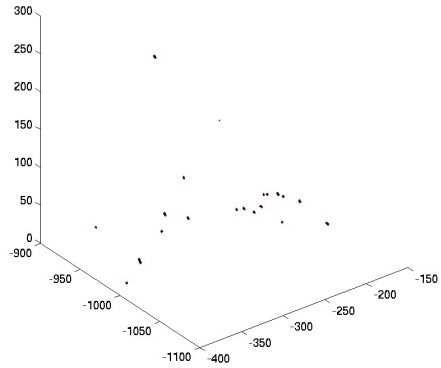
a) Selected Giraffe Points.



Eight-Point Algorithm reconstruction



b) Reconstruction using Eq. 17.



c) Reconstruction using Eq. 16.

Fig. 5: **Comparison of pose-free reconstruction using a subset versus all depth-only equations.**

of the other term, $(P_i - C_j)\cdot(P_i - C_j)$, can be deduced from the right-hand-side of the 6^{th} equation along with the right-hand-side of the first and second equation of (9). Thus $\|P_i - P_1\|^2$ can be written in terms of the right-hand-side of the first, 4^{th}, and 6^{th} equation of (9). Replacing these right-hand-sides with their respective left-hand-sides, we obtain the camera-pose-free equation:

$$\|\gamma_{ij}p_{ij} - \gamma_{1j}p_{1j}\|^2 = \|P_i - P_1\|^2, \text{ for } i = 2, 3, ...N.$$

Using similar arguments, we can generate the following set of camera-pose-free equations:

$$\begin{aligned}
\|\gamma_{ij}p_{ij} - \gamma_{1j}p_{1j}\|^2 &= \|P_i - P_1\|^2, \text{ for } i = 2, 3, ...N. \\
\|\gamma_{ij}p_{ij} - \gamma_{2j}p_{2j}\|^2 &= \|P_i - P_2\|^2, \text{ for } i = 3, 4, ...N, \\
\|\gamma_{ij}p_{ij} - \gamma_{3j}p_{3j}\|^2 &= \|P_i - P_3\|^2, \text{ for } i = 4, ...N, \quad (10) \\
\|\gamma_{ij}p_{ij} - \gamma_{4j}p_{4j}\|^2 &= \|P_i - P_4\|^2, \text{ for } i = 5, 6, ...N, \\
(\gamma_{4j}p_{4j} - \gamma_{3j}p_{3j}) \cdot (\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}) &\times (\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j}) \\
&= (P_4 - P_3)\cdot(P_1 - P_3)\times(P_2 - P_3), \\
&\text{for } j = 1, 2, ...J.
\end{aligned}$$

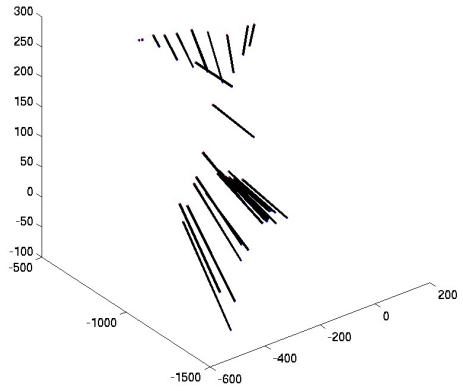
These equations could also have been obtained by observing that the Euclidean norm and the signed triangular area are unchanged under any orientation preserving rigid motion.

Assuming $P_4 - P_3, P_1 - P_3, P_2 - P_3$ are not coplanar, this is a complete equation set (see Appendix II). For $N \geq 5$ and $J = 2$, it is also a minimal equation set, in the sense that removing any equation from the set would introduce new (invalid) solutions. For example, removing the first equation for $i = 5$ would remove the knowledge of the value of the quantity $\|P_5 - P_1\|^2$ from the equations, as this quantity cannot be recovered from the other right-hand-sides (since there are two different values of P_5 with the same $\|P_5 - P_2\|^2, \|P_5 - P_3\|^2, \|P_5 - P_4\|^2$). Thus that equation cannot be inferred from the other equations. Similarly, we cannot remove the second, third and forth equations. Finally, if the last equation was removed, for some j_0, the system would have two distinct solutions for $(\gamma_{1j_0}, \gamma_{2j_0}, \gamma_{3j_0}, \gamma_{3j_0})$, one being the reflection of the true solution. It thus cannot be removed.

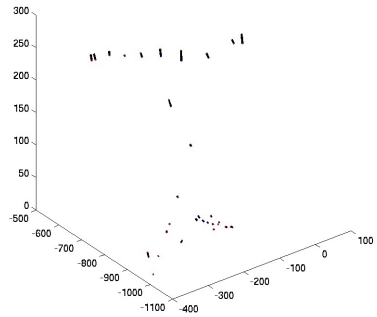
The asymmetry of the system with respect to the points could create numerical problems. To avoid this, one can first solve the symmetric system

$\|\gamma_{ij}p_{ij} - \gamma_{\bar{i}j}p_{\bar{i}j}\|^2 = \|P_i - P_{\bar{i}}\|^2$, for $i, \bar{i} = 1, ...N, j = 1, ...J$, and remove the extra solutions using the remaining equations:

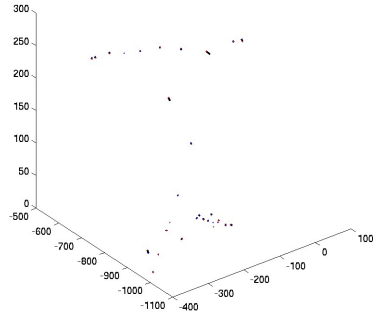
$$\begin{aligned}
&(\gamma_{4j}p_{4j} - \gamma_{3j}p_{3j}) \cdot (\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}) \times (\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j}) \\
&= (P_4 - P_3)\cdot(P_1 - P_3)\times(P_2 - P_3), \text{ for } j = 1, \ldots, J. \quad (11)
\end{aligned}$$



a) BA refinement of Fig. 5b) (no improvement).



b) BA refinement of Fig. 5c) (worse).



c) BA refinement of Fig. 5d) using BA (slightly worse).
Fig. 6: **BA refinement of different initial guesses**.

VII. Depth from Motion (Depth-only SFM)

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}. We will now eliminate all the remaining variables except the γ_{ij}'s. To do this, we observe that the right-hand-sides of all the equations contained in Equations 10 are all independent of j. Thus, the left-hand-sides for different j's must be equal. We thus obtain the following system of *depth-only* equations:

$$\begin{aligned}
&\|\gamma_{ij}p_{ij} - \gamma_{1j}p_{1j}\|^2 = \|\gamma_{i\bar{j}}p_{i\bar{j}} - \gamma_{1\bar{j}}p_{1\bar{j}}\|^2, \text{ for } i = 2,...N,\\
&\|\gamma_{ij}p_{ij} - \gamma_{2j}p_{2j}\|^2 = \|\gamma_{i\bar{j}}p_{i\bar{j}} - \gamma_{2\bar{j}}p_{2\bar{j}}\|^2, \text{ for } i = 3,...N,\\
&\|\gamma_{ij}p_{ij} - \gamma_{3j}p_{3j}\|^2 = \|\gamma_{i\bar{j}}p_{i\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}\|^2, \text{ for } i = 4,...N, \quad (12)\\
&\|\gamma_{ij}p_{ij} - \gamma_{4j}p_{4j}\|^2 = \|\gamma_{i\bar{j}}p_{i\bar{j}} - \gamma_{4\bar{j}}p_{4\bar{j}}\|^2, \text{ for } i = 5,...N,\\
&(\gamma_{4j}p_{4j} - \gamma_{3j}p_{3j}) \cdot (\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}) \times (\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j})\\
&\qquad = (\gamma_{4\bar{j}}p_{4\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \cdot (\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}})\\
&\qquad \times(\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}), \text{ for distinct } j,\bar{j} = 1,...J.
\end{aligned}$$

With a similar proof as the one presented in Appendix II, one can show that the above is a complete set of equations. By a similar argument as for the pose-free equations discussed in Section VI, one can also show that it is a minimal set for the case of $J = 2$ views. One can also work with a symmetrized equation set such as:

$$\|\gamma_{ij}p_{ij} - \gamma_{\bar{i}j}p_{\bar{i}j})\|^2 = \|\gamma_{i\bar{j}}p_{i\bar{j}} - \gamma_{\bar{i}\bar{j}}p_{\bar{i}\bar{j}})\|^2,$$
$$\text{for distinct } i,\bar{i} = 1,2,...n, \text{ and distinct } j,\bar{j} = 1,2,...J, \quad (13)$$

and later remove the extra solutions by enforcing the remaining equations:

$$\begin{aligned}
&(\gamma_{4j}p_{4j} - \gamma_{3j}p_{3j}) \cdot (\gamma_{1j}p_{1j} - \gamma_{3j}p_{3j}) \times (\gamma_{2j}p_{2j} - \gamma_{3j}p_{3j}) =\\
&(\gamma_{4\bar{j}}p_{4\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \cdot (\gamma_{1\bar{j}}p_{1\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}) \times (\gamma_{2\bar{j}}p_{2\bar{j}} - \gamma_{3\bar{j}}p_{3\bar{j}}),\\
&\qquad \text{for distinct } j,\bar{j} = 1,2,...J. \quad (14)
\end{aligned}$$

Both Equations 12 and Equations 13-14 are homogeneous systems of equations, so their solutions are only defined up to a global scale factor. Since all γ_{ij}'s are strictly positive, one can set one of them, say γ_{11}, equal to one and solve for the remaining ones. Note that once the depth γ_{ij}'s are known, then one can recover the 3D points P_i's linearly by solving the overdetermined linear system of equations

$$P_i = \gamma_{ij}p_{ij},$$

(e.g., by computing the Moore-Penrose pseudoinverse). Having set the depth scale by setting $\gamma_{11} = 1$, the 3D object points are then uniquely determined up to a rotation and a translation.

VIII. Global Pose-free SFM by Homotopy

Algebraic variable elimination in a polynomial system reduces the number of unknowns that need to be taken into account while solving the system, which can reduce the complexity of the computation. Furthermore, it can open the door to the use of algebraic-based technique for either analyzing or solving the system. This is particularly interesting in the case where numerical techniques do not work particularly well, such as degenerate or ill-conditioned cases, or when trying to understand a large parametric class of cases.

For SFM, variable elimination down to only depth parameters drastically reduces the size of the problem. For example, given five generic 3D points projected on a (generic) pair of images, the standard SFM formulation involves a total of 36 variables: 15 variables for the 3D points P_i, 10 depth parameters γ_{ij}, one of which can be set to fix the scale ambiguity, and 12 camera parameters (9 for position and 3 for orientation) for the second camera (we set the coordinate system in the first camera to our default coordinate system). Removing the camera angle parameters brings the number of variables down to 27. Further removing the camera center variables yields 24 variables. Removing the 3D points coordinates finally brings the number down to only 9. See Table I.

With today's computers, many symbolic-numeric solution techniques are very efficient for solving problems with not

too many unknowns. The following numerical experiments demonstrate that the depth-only SFM problem formulation we propose has few enough parameters to be handled by numeric-symbolic techniques. Thus, it may be possible to use such techniques to better characterize and/or analyze some degenerate or ill-conditioned cases, as well as other cases that cannot be easily understood numerically.

A. Algebraic-Numerical Experiments

There are many numerical methods for solving polynomial equations near an initial guess. Without an initial guess, or if one desires to find all solutions of the system, then one can attempt to solve the system algebraically. However, algebraic methods cannot generally handle a large number of variables or equations. A class of methods called *homotopy continuation methods*, or simply *homotopy*, uses a mixture of numerical and analytical techniques to address this problem. The idea is to modify the system into a simpler system for which all the solutions are known. Then the modified system is slowly evolved back onto the initial system by continuously varying its coefficients. The path between the two polynomial systems is divided into small steps, and at each step the solutions of the corresponding system are obtained numerically by using the previous system's solutions as initial guesses.

For a small number of points N and pictures J, our depth-only equations (either Equations 12 or Equations 13 and 14) can be solved by homotopy in Maple using the package PHCmaple [14]. For example, consider the case of J=2 pictures (say $j = 1, 2$) and N=5 points (say $i = 1, 2, 3, 4, 5$) on each picture. Equations 13 and 14 then form the set

$$\begin{aligned}
\|\gamma_{11}p_{11} - \gamma_{21}p_{21}\|^2 &= \|\gamma_{12}p_{12} - \gamma_{22}p_{22}\|^2,\\
\|\gamma_{11}p_{11} - \gamma_{31}p_{31}\|^2 &= \|\gamma_{12}p_{12} - \gamma_{32}p_{32}\|^2,\\
\|\gamma_{21}p_{21} - \gamma_{31}p_{31}\|^2 &= \|\gamma_{22}p_{22} - \gamma_{32}p_{32}\|^2,\\
\|\gamma_{11}p_{11} - \gamma_{41}p_{41}\|^2 &= \|\gamma_{12}p_{12} - \gamma_{42}p_{42}\|^2,\\
\|\gamma_{21}p_{21} - \gamma_{41}p_{41}\|^2 &= \|\gamma_{22}p_{22} - \gamma_{42}p_{42}\|^2,\\
\|\gamma_{31}p_{31} - \gamma_{41}p_{41}\|^2 &= \|\gamma_{32}p_{32} - \gamma_{42}p_{42}\|^2,\\
\|\gamma_{11}p_{11} - \gamma_{51}p_{51}\|^2 &= \|\gamma_{12}p_{12} - \gamma_{52}p_{52}\|^2,\\
\|\gamma_{21}p_{21} - \gamma_{51}p_{51}\|^2 &= \|\gamma_{22}p_{22} - \gamma_{52}p_{52}\|^2,\\
\|\gamma_{31}p_{31} - \gamma_{51}p_{51}\|^2 &= \|\gamma_{32}p_{32} - \gamma_{52}p_{52}\|^2,\\
\|\gamma_{41}p_{41} - \gamma_{51}p_{51}\|^2 &= \|\gamma_{42}p_{42} - \gamma_{52}p_{52}\|^2,\\
(\gamma_{41}p_{41} - \gamma_{31}p_{31}) \cdot (\gamma_{11}p_{11} - \gamma_{31}p_{31}) &\times (\gamma_{21}p_{21} - \gamma_{31}p_{31})\\
= (\gamma_{42}p_{42} - \gamma_{32}p_{32}) \cdot (\gamma_{12}p_{12} - \gamma_{32}p_{32}) &\times (\gamma_{22}p_{22} - \gamma_{32}p_{32}).
\end{aligned} \tag{15}$$

where γ_{11} can be set to one to fix the scale ambiguity.

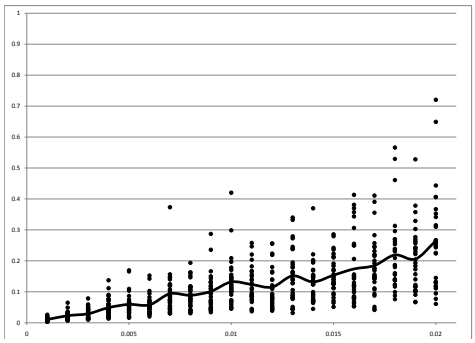
PHCmaple requires the number of equations to equal the number of unknowns. So one can pick the first 9 of the 11 equations contained in the system and obtain their solutions; this introduces extraneous solutions that can be removed by plugging into the remaining two equations. We picked a generic configuration of five points reconstructed the 3D point positions from two images using this method. After solving the first 9 equations, we obtained 234 solutions, 44 of which were real. Since we were looking for positive and real solutions (as the depths γ_{ij}'s must be positive and real), we were able to get rid of most extraneous solutions immediately. The remaining ones (a total of 8) were plugged back into the last 2 equations in order to get the true solution. The computation took less than 2 minutes on a PC with a 2.66GHz Intel(R) Core(TM) 2 Duo Processor with 3GB of RAM (Table I).

In contrast the standard SFM formulation (Equations 1) cannot effectively be solved by homotopy because it contains too many variables. Indeed, we took the same 5 points and two pictures as above, set the coordinate system in the first camera to our default coordinate system and the first depth parameter γ_{11} to one, and entered the sine and cosine appearing in the equations as free parameters in the camera rotation matrices R_j. The constraint $R_j^T R_j = \mathcal{I}$ and det(R_j)$= 1$ were then entered as additional polynomial equations into the system. PHCmaple ran for more than two weeks without returning an answer. To illustrate our claim that this is due to the number of variables, we attempted to solve four other formulations of SFM with PHCmaple, each with a different number of variables. All computations were done with the same 5 points viewed on the same two pictures, and all were run on the same PC.The results are summarized in Table I.

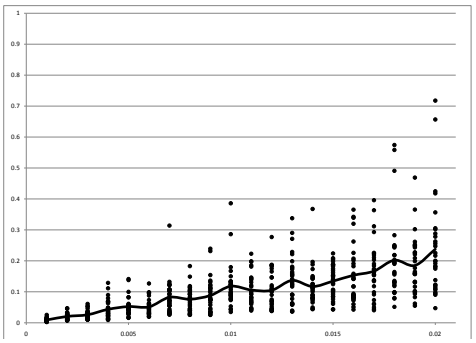
We began with the angle-free formulation (Equation 8), which for 5 points and 2 pictures consists of 27 variables and 27 equations. Solving it using PHCmaple took 39m25s. We found a total of 171 solutions, 7 of which were real and 164 of which were complex. Only one of them, the true solution, was real and positive. For the pose-free formulation (Equation 11 and 11), we had a system of 24 variables and 27 equations. Solving 24 equations among these 27 using PHCmaple took 3h35m16s. We found a total of 1183 solutions, 907 of which were real. We selected the positive ones and plugged them back into the remaining equations to find the true solution.

Of course, there are other ways to eliminate variables. While this work concentrates on removing the pose parameters in order to improve the conditioning of the SFM problem, it is worthwhile to discuss some other formulations for the sake of further analyzing the computational advantage of variable elimination. For example, one can keep the pose parameters and eliminate the 3D point coordinates, as shown in Appendix III. This also has the effect of reducing the complexity of the problem within the context of homotopy methods, allowing PHC Maple to solve the case of 5 points on two pictures in about 13 minutes. Further eliminating the camera angle (see Appendix III), slightly decreases the computation time.

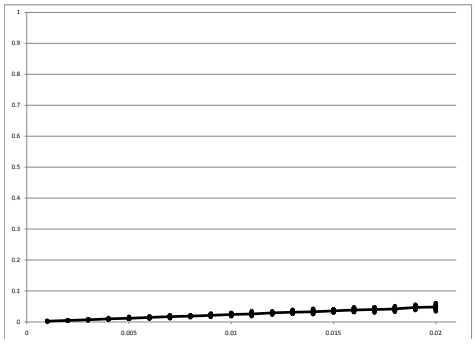
Note that all solutions of the corresponding SFM problem, including complex and negative ones, can be found by homotopy. Homotopy methods can also be used to solve under-determined problems. Thus, the fact that our pose-free formulation of SFM has few enough variables to be solved by homotopy is good news. Indeed, it is not inconceivable that one can solve and/or analyze some non-generic SFM problems using this approach. The solution method we proposed in this section is not necessarily the fastest. Indeed, PHCmaple is one homotopy package among others and, certainly, a custom-built implementation would yield faster result than this all-purpose package. However, for generic cases, it is unlikely that any technique involving symbolic computations will perform faster than state-of-the-art purely numerical techniques. On the other hand, the analysis of non-generic cases is typically done in an off-line setting, in which speed is not necessarily critical.



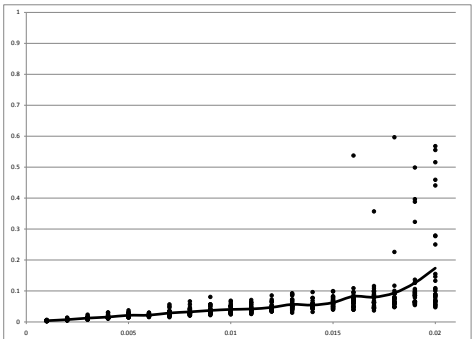
(a) Accuracy of eight point algorithm followed by triangulation.



(b) Accuracy of refinement of a) obtained by minimizing total reprojection error.



(c) Accuracy of refinement of a) obtained by minimizing depth-only cost function.



(d) Accuracy of refinement of a) obtained by minimizing reduced depth-only cost function.

Fig. 7: **Comparison of reconstruction accuracy for depth-only cost functions and total reprojection error.**

IX. Local Pose-free SFM by Numerical Optimization

Beyond a small number of points and a small number of pictures, or when the results of a generic SFM problem must be computed within a fraction of a second, the global solution method discussed in Section VIII is not the most effective. In that case, a two-step approach that consists in obtaining an initial guess for the solution and in subsequently refining this solution is preferable. There exist many methods for obtaining an initial guess. It is beyond the scope of this work to argue what method is best. But the proposed depth-only equations (either (12) or (13)-(14)) can be used to formulate a pose-free cost function to refine any given initial solution guess for the object points. For example, one can use (13)-(14) to formulate the following depth-only cost function:

$$\sum_{j\neq\bar{j}}\sum_{i\neq\bar{i}}\left[\|\gamma_{ij}p_{ij}-\gamma_{\bar{i}j}p_{\bar{i}j}\|^2-\|\gamma_{i\bar{j}}p_{i\bar{j}}-\gamma_{\bar{i}\bar{j}}p_{\bar{i}\bar{j}}\|^2\right]^2$$
$$+\sum_{j\neq\bar{j}}\left[(\gamma_{4j}p_{4j}-\gamma_{3j}p_{3j})\cdot(\gamma_{1j}p_{1j}-\gamma_{3j}p_{3j})\right.$$
$$\times(\gamma_{2j}p_{2j}-\gamma_{3j}p_{3j})-(\gamma_{4\bar{j}}p_{4\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}})\cdot(\gamma_{1\bar{j}}p_{1\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}})$$
$$\left.\times(\gamma_{2\bar{j}}p_{2\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}})\right]^2. \quad (16)$$

One can also obtain a depth only cost function using Equation 11:

$$\sum_{j\neq\bar{j}}[\sum_{i=2}^{N}(\|\gamma_{ij}p_{ij}-\gamma_{1j}p_{1j}\|^2-\|\gamma_{i\bar{j}}p_{i\bar{j}}-\gamma_{1\bar{j}}p_{1\bar{j}}\|^2)^2$$
$$+\sum_{i=3}^{N}(\|\gamma_{ij}p_{ij}-\gamma_{2j}p_{2j}\|^2-\|\gamma_{i\bar{j}}p_{i\bar{j}}-\gamma_{2\bar{j}}p_{2\bar{j}}\|^2)^2$$
$$+\sum_{i=4}^{N}(\|\gamma_{ij}p_{ij}-\gamma_{3j}p_{3j}\|^2-\|\gamma_{i\bar{j}}p_{i\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}}\|^2)^2$$
$$+\sum_{i=5}^{N}(\|\gamma_{ij}p_{ij}-\gamma_{4j}p_{4j}\|^2-\|\gamma_{i\bar{j}}p_{i\bar{j}}-\gamma_{4\bar{j}}p_{4\bar{j}}\|^2)^2$$
$$+((\gamma_{4j}p_{4j}-\gamma_{3j}p_{3j})\cdot(\gamma_{1j}p_{1j}-\gamma_{3j}p_{3j})$$
$$\times(\gamma_{2j}p_{2j}-\gamma_{3j}p_{3j})-(\gamma_{4\bar{j}}p_{4\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}})\cdot(\gamma_{1\bar{j}}p_{1\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}})$$
$$\times(\gamma_{2\bar{j}}p_{2\bar{j}}-\gamma_{3\bar{j}}p_{3\bar{j}}))^2]. \quad (17)$$

A. Numerical Experiments

We now perform numerical experiments to substantiate our claim that pose-free (depth-only) cost functions lead a more accurate refinement that when using the total reprojection error, as in the standard bundle adjustment. Note that, in all our experiments, the internal camera parameters are fixed and assumed to be known. All minimizations were performed using the function *lsqnonlin* in MATLAB. The MATLAB functions we wrote to solve our depth-only equations are available at www.ece.purdue.edu/~mboutin/code. All computations in this section were performed on a PC with a 2.66GHz Intel(R) Core(TM) 2 Duo Processor with 3GB of RAM.

Our first experiment consisted in reconstructing a giraffe model captured in our lab. The size of this model is about

$300mm \times 200mm \times 300mm$. The camera we used was mounted on a Microscribe Arm G2LX mechanical arm manufactured by Immersion Corporation, which allowed us to precisely measure the camera position (within one millimeter) and orientation (within a fraction of a degree) for each picture. We took a total of 10 pictures of the giraffe and used the Kanade-Lucas-Tomasi feature tracking software package [19] to track the features. As the camera positions were all on the same side of the model, we were able to select 100 points that appeared on all ten pictures. While the camera positions and orientations obtained from the mechanical arm are not used in our reconstruction process, they were necessary to obtain a ground truth solution for the 3D giraffe points in order to quantify the accuracy of our reconstruction. More precisely, the ground truth solution was obtained by triangulation for each pair of views, thus obtaining a overdetermined linear system of equations, which was solved by singular value decomposition to obtain the 3D point coordinates.

In order to demosntrate the improvement obtained by using a pose-free (depth-only) cost function, we first needed an inaccurate initial guess. We obtained this initial guess using the the Eight-Point Algorithm [12] to recover the essential matrix and thus the camera pose. The intrinsic camera parameters were assumed to be known throughout the computation. We subsequently recovered an initial guess for the Giraffe points. The results, which are illustrated in Figure 2 a), were obtained in Matlab in about 0.22 seconds of CPU time. As the pictures were all taken from the same side of the object, the resulting reconstruction was not very accurate.

Figure 2 shows the reconstruction obtained after refining this initial guess with the pose-free (depth-only) cost function of Equation 17. The total CPU time used for that second step was about 530 seconds. As one can see from the figure, the results are very accurate despite the fact that the views were only acquired from one side of the object.

For comparison, we also refined the the initial guess of Figure 3 a) by minimizing the total reprojection error through varying the external camera parameters (camera position and orientation) and the object point 3D coordinates. Note that the internal camera parameters were kept fixed during this optimization. This step took about 63 seconds but did not yield any significant improvement. The inaccuracy of the reconstruction is especially obvious from a top view (Figure 3 b), where the high ambiguity of the giraffe point positions along one direction is observed.

The main reason that the cost function defined in Equation 17 leads to a more accurate reconstruction than minimizing total reprojection error is that it does not contain any external pose parameters. When camera pose estimation is not well conditioned, large variations in the pose estimation can lead to very small variations in the reprojections. As a result, one can be very close to the optimal total reprojection error while being very far from the true camera pose. Thus, if the initial guess for the camera pose is inaccurate, minimizing the total reprojection error may not yield any significant improvement. This phenomenon can be observed in Figure 4, where we show four different reprojections of the solution obtained by total reprojection error minimization (Figure 3). The reprojection displayed corresponds to the first, forth, seventh, and tenth view. As one can see, each of these reprojections is highly accurate even though the 3D reconstruction is not, thus illustrating the bad conditioning of the problem we are attempting to solve. Note that the reconstruction error, measured as the average Euclidean distance between all points and their corresponding reconstruction, is more than 19.745mm. In contrast, our solution (Figure 2) has a reconstruction error of only 5.164mm.

The difference between bundle adjustment and refining one of our proposed pose-free cost function is more drastic in our second set of experiments, in which we randomly selected a set of 20 Giraffe points and considered the same ten pictures acquired with our mechanical arm. As before, these 20 points were reconstructed using the Eight-Point Algorithm followed by triangulation. The reconstruction is illustrated in Figure 5a). One notices the particularly bad accuracy of this reconstruction. Indeed, the reconstruction error, measured as the average Euclidean distance between all points and their corresponding reconstruction, is 294.65mm. Nonetheless, refining this solution using the cost function defined by Equation 17 produces a very accurate result, with a reconstruction error of 3.03mm. By using the cost function of Equation 16, which uses a larger set of equations, we can refine the initial guess even better, with a reconstruction error of only 2.49mm. (See Figure 5d).) In constrast, refining the Eight-Point Algorithm solution by minimizing the total reprojection error does not lead any noticeable improvement, as the error of the resulting reconstruction is still almost the same, at 294.62. It is also interesting to note what happens when our pose-free solutions are refined by minimizing the total reprojection error. In this particular case, the reconstruction error actually increases. Indeed when the pose-free solution of Figure 5c) was refined using BA (using the true camera parameters as initial guess for the camera parameters), the reconstruction error increased from 3.03mm to 5.81mm. Similarly, when the pose-free solution of Figure 5d) was refined using BA (again using the true camera parameters as initial guess for the camera parameters), the reconstruction error increased from 2.49mm to 3.17mm.

Our next experiment aims to demonstrate that, in general, refining with a pose-free (depth-only) cost functions leads to a statistically better accuracy than minimizing the total reprojection error. We generated 30 random 3D points and projected them onto two images separated by a translation of $(2, 0, 0)$ and no rotation. More precisely, the camera centers used to acquire the images were $(-1, 0.5, 1)$ and $(1, 0.5, 1)$, respectively. We added Gaussian noise to the projection before using the Eight Point Algorithm to recover the external camera parameters (the internal camera parameters were fixed) and subsequently recover an initial guess for the 3D reconstruction (by triangulation).

We repeated this experiment 30 times for 20 different values of the standard deviation of the noise. For each reconstruction, we computed the reconstruction error as the average L^2 norm of the point-wise difference. As one can see from the graph in Figure 7 a) the output error increases as the input noise increases. These initial guess were, of course, not particularly

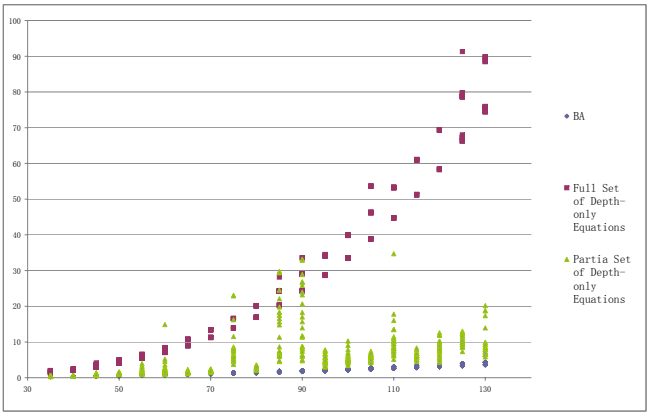


Fig. 8: **Comparison of running time for minimizing different cost functions**

accurate: our aim was to use them to showcase the difference in accuracy of a refinement with our proposed pose-free (depth-only) cost functions and that of a bundle adjustment refinement.

Having obtained an initial solution guess, we subsequently, refined the reconstruction by minimizing the total reprojection error (while keeping the internal camera parameters constant). The results were obtained in 0.203 second in average. The error of the corresponding reconstruction is plotted in Figure 7 b). For comparison, we also refined the initial solution guess by minimizing the following depth-only cost function:

$$E = \sum_{i=1}^{N-1} \sum_{\bar{i}=i+1}^{N} (\|\gamma_{i1}p_{i1} - \gamma_{\bar{i}1}p_{\bar{i}1}\|^2 - \|\gamma_{i2}p_{i2} - \gamma_{\bar{i}2}p_{\bar{i}2}\|^2)^2 + ((\gamma_{41}p_{41} - \gamma_{31}p_{31}) \cdot (\gamma_{11}p_{11} - \gamma_{31}p_{31}) \times (\gamma_{21}p_{21} - \gamma_{31}p_{31}) - (\gamma_{42}p_{42} - \gamma_{32}p_{32}) \cdot (\gamma_{12}p_{12} - \gamma_{32}p_{32}) \times (\gamma_{22}p_{22} - \gamma_{32}p_{32}))^2, \quad (18)$$

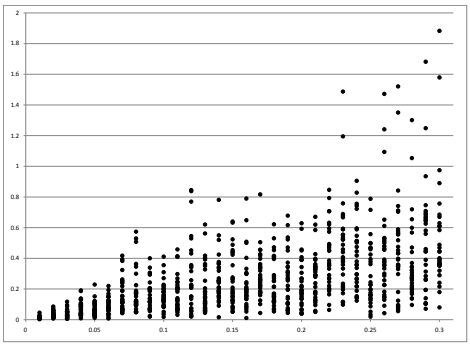
for $N = 30$ points. This function uses the symmetrized equation set (Equation 13) along with Equation 14. The average Euclidean distance between the 3D points we obtained and their true value is plotted in Figure 7 c). A significant improvement over minimizing total reprojection error can be observed. The CPU time used for this step was about 1.027 seconds in average, which is about five times the CPU time used to minimizing total reprojection error. The time difference increases with the number of points considered. This is because our cost function uses one equation for every pair of points while the total reprojection error uses only one equation per point. So our cost function contains more terms than the total reprojection error. Actually the number of cost function terms of our method is $(N(N-1)/2+1)(J-1)$ versus NJ for the total reprojection error, for a data set with N points and J images. The relationship between the number of reconstructed 3D points and the CPU time is shown in Figure 8. As one can see, the relationship between the running time and the number of points in minimizing the total reprojection error is almost linear while the one in our depth-only method is quadratic. However, we can reduce the size of our cost function by only considering a fraction of our proposed depth-only equations. More precisely, instead of using the full set depth-only cost function (Equation 18), one can use a partial set of depth-only equations to obtain a reduced cost function, such as this one, which uses the complete and minimal equation set (Equation 12):

$$E_{reduced} = \sum_{i=2}^{N} (\|\gamma_{i1}p_{i1} - \gamma_{11}p_{11}\|^2 - \|\gamma_{i2}p_{i2} - \gamma_{12}p_{12}\|^2)^2 + \sum_{i=3}^{N} (\|\gamma_{i1}p_{i1} - \gamma_{21}p_{21}\|^2 - \|\gamma_{i2}p_{i2} - \gamma_{22}p_{22}\|^2)^2 + \sum_{i=4}^{N} (\|\gamma_{i1}p_{i1} - \gamma_{31}p_{31}\|^2 - \|\gamma_{i2}p_{i2} - \gamma_{32}p_{32}\|^2)^2 + \sum_{i=5}^{N} (\|\gamma_{i1}p_{i1} - \gamma_{41}p_{41}\|^2 - \|\gamma_{i2}p_{i2} - \gamma_{42}p_{42}\|^2)^2 + ((\gamma_{41}p_{41} - \gamma_{31}p_{31}) \cdot (\gamma_{11}p_{11} - \gamma_{31}p_{31}) \times (\gamma_{21}p_{21} - \gamma_{31}p_{31}) - (\gamma_{42}p_{42} - \gamma_{32}p_{32}) \cdot (\gamma_{12}p_{12} - \gamma_{32}p_{32}) \times (\gamma_{22}p_{22} - \gamma_{32}p_{32}))^2. \quad (19)$$

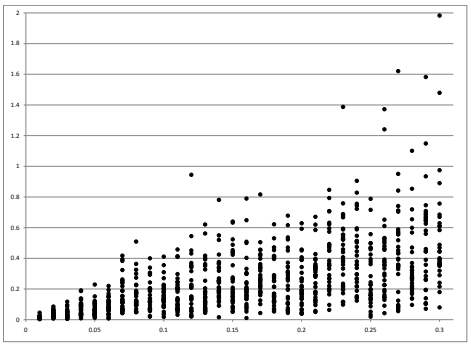
This cost function contains $(4(N-4)+7)(J-1)$ terms, which makes the CPU time of the reconstruction comparable to that of total reprojection error minimization, as shown in Figure 8. As we can see from Figure 7 d), the accuracy of the reconstruction is not significantly affected.

As stated earlier, the reason that the cost functions Equation 18 and 19 are more accurate than the total reprojection error is that they do not contain any external camera pose parameters. The most troublesome parameter is the camera angle, as a small change in camera angle can make a large difference in the 3D point positions. Even when one knows the camera position precisely, it is still highly difficult to obtain the reconstruction accurately by minimizing the total reprojection error, as long as the camera angle estimate is inaccurate. To substantiate this statement, we ran another experiment. Again we used 30 randomly generated 3D points and the same 2 cameras. We added Gaussian noise with incrementally large standard deviation to the camera orientation, and then used the noisy camera pose along with the projection data to reconstruct an estimate of 3D points. The error between the estimated 3D points and the ground true is shown in Figure 9 a). We then minimized the total reprojection error in order to refine this initial solution. Figure 9 b) shows that this did not lower the reconstruction error significantly. In sharp contrast, our depth-only method, both *full* and *partial*, give highly accurate results, as shown in Figure 9 c) and d) respectively.

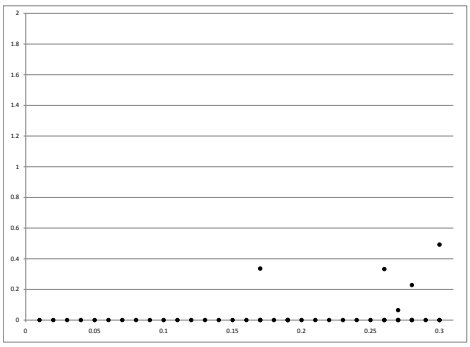
While the reconstruction error obtained with our approach is smaller than when using BA, even in the presence of outliers created by the Gaussian noise added to the points, it is still affected by noise: the more noise the less accurate the reconstruction. As with any other reconstruction method, the presence of mislabeled points is problematic, as the coordinates of mislabeled points can vary drastically from their true coordinates, which can lead to a poorly converging optimization. To illustrate this, and in order to better separate the effect of noise from mislabeling, we repeated the 20 point giraffe experiment while artificially adding further and further outliers



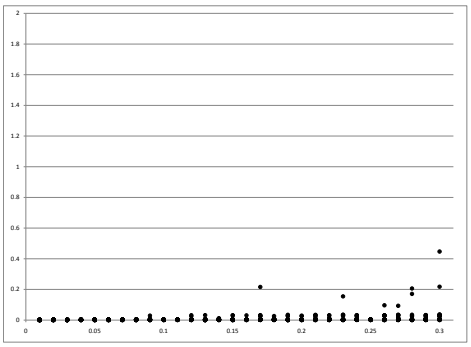
(a) Accuracy of initial 3D reconstruction after perturbing camera angle.



(b) Accuracy of refinement of a) obtained by minimizing total reprojection error.



(c) Accuracy of refinement of a) obtained by minimizing depth-only cost function.



(d) Accuracy of refinement of a) obtained by minimizing reduced depth-only cost function.

Fig. 9: **Effect of error in the camera angle estimate with perfect knowledge of the camera position.**

to the data. This was done by moving the image of some specific points towards the top left corner position of the image while recording the effect on the reconstruction accuracy. More specifically, we first modified the image of the 10^{th} point on the 10^{th} image by moving it on a straight line towards the upper left corner of the image: the position of this outlier on the line is parameterized uniformly using the parameter t. This corresponds to adding 0.5% outlier to the data. We repeated the same experiment by simultaneously moving the 10^{th} image of the 10^{th} point and the 9^{th} image of the 9^{th} point both towards the upper left corner of the respective image. This time, we used the same parameter t to parameterize the movement of both points. This corresponds to adding 1% outlier to the data. Finally, we repeated the same experiment by simultaneously moving the 10^{th} image of the 10^{th} point, the 9^{th} image of the 9^{th} point, and the 8^{th} image of the 8^{th} point both towards the upper left corner of the respective image. Again, we used the same parameter t to parameterize the movement of all three points. This corresponds to adding 1.5% outlier to the data. As one can see from the graph, both for the cost function of Equations 17 and that of Equation 16, moving the outliers further and further away produces somewhat erratic results passed a certain threshold. Thus when there is a potential for mislabeling, the use of RANSAC in combination with our approach should be considered.

X. Statistical Interpretation

The total reprojection error is generally accepted to be the best measure of reconstruction accuracy. Part of this belief is based on the fact that minimizing the total reprojection error yields the maximum likelihood estimate for the camera pose and object reconstruction when the image error is zero-mean Gaussian [5] [11] [23]. In general, a maximum likelihood estimator is viewed as a good estimator because it is asymptotically unbiased and efficient. In other words, as the number of observations tends to infinity, its bias tends to zero and the expected value of its mean squared error tends to the Cramér-Rao lower bound. Thus, in the asymptotic limit, no unbiased estimator for the camera pose and object reconstruction can be more accurate than an estimator that minimizes the total reprojection error when the image error is zero-mean Gaussian. In the experiments presented in Figure 7, we show that the mean squared error of our estimator is much smaller than that of the maximum likelihood estimator. This is not a contradiction because our estimator is biased. Recall that our equations are obtained by projecting the standard SFM equations onto a subspace. This is a standard regularization technique used to increase the conditioning of an estimator. While this adds a bias to the estimator, the average accuracy is often improved. See for example Chapter 4 of [2] for a simple example in linear algebra.

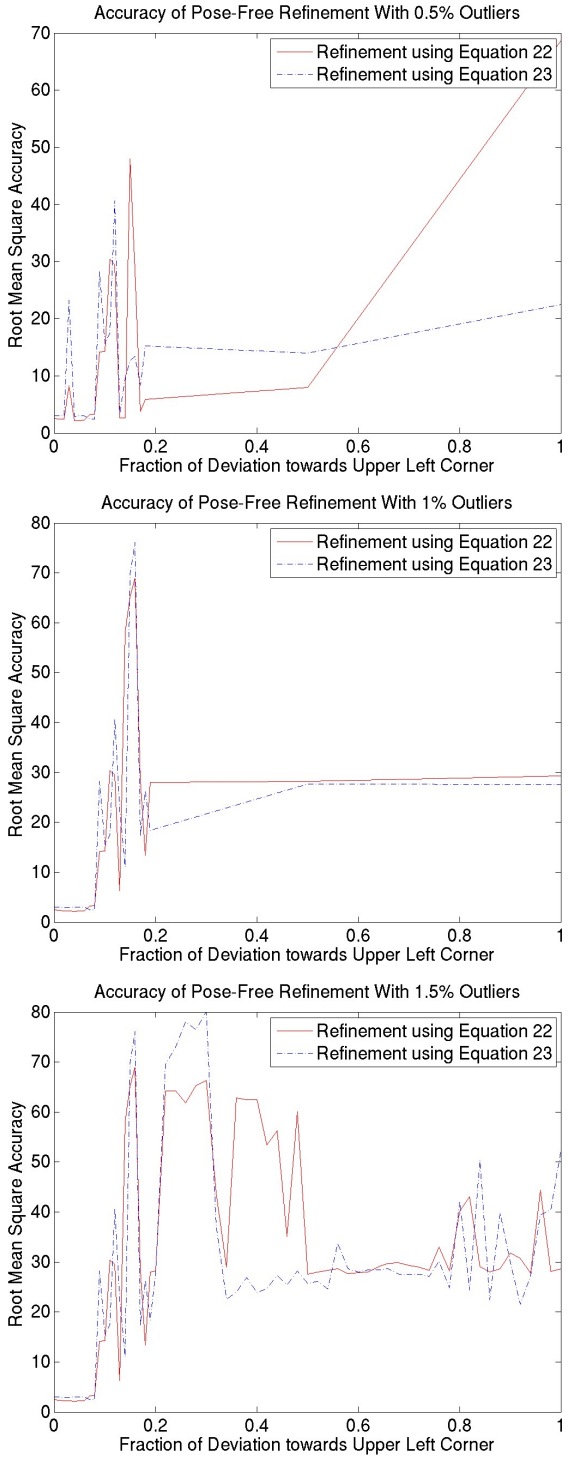


Fig. 10: **Effect of Outliers on Accuracy.**

XI. Future Extensions

The results we obtained so far are only valid when the internal camera parameters are fixed. However, they can easily be extended to include variable internal camera parameters, as we show below. To illustrate how this can be done without any knowledge of invariant theory, we begin by considering the case where only the focal lengths of the cameras are variable and unknown before generalizing to the case where all internal camera parameters are variable and unknown.

Let f_j denote the focal length of the camera at picture j. Then Equation 4 becomes

$$\gamma_{ij}\begin{pmatrix} f_j^{-1}x_{ij} \\ f_j^{-1}y_{ij} \\ 1 \end{pmatrix} = R_jP_i - R_jC_j, \text{ for all } i,j. \qquad (20)$$

Following our previous reasoning, we can eliminate the external camera parameters R_j and C_j, thus obtaining the following modified version of our depth-from-motion Equations 13

$$\left\| \gamma_{ij}\begin{pmatrix} f_j^{-1}x_{ij} \\ f_j^{-1}y_{ij} \\ 1 \end{pmatrix} - \gamma_{\bar{i}j}\begin{pmatrix} f_j^{-1}x_{\bar{i}j} \\ f_j^{-1}y_{\bar{i}j} \\ 1 \end{pmatrix} \right\|^2$$

$$= \left\| \gamma_{i\bar{j}}\begin{pmatrix} f_{\bar{j}}^{-1}x_{i\bar{j}} \\ f_{\bar{j}}^{-1}y_{i\bar{j}} \\ 1 \end{pmatrix} - \gamma_{\bar{i}\bar{j}}\begin{pmatrix} f_{\bar{j}}^{-1}x_{\bar{i}\bar{j}} \\ f_{\bar{j}}^{-1}y_{\bar{i}\bar{j}} \\ 1 \end{pmatrix} \right\|^2, \qquad (21)$$

for distinct $i,\bar{i} = 1,2,...n$, and distinct $j,\bar{j} = 1,2,...J$. Solving these equations would yield the values of the unknowns f_j, γ_{ij} up to the projective ambiguity of the reconstruction.

In order to obtain a fully "pose-free" formulation, we would need to further eliminate the internal camera parameters f_j. More generally, if all internal camera parameters were unknown and allowed to vary from picture to picture, then Equation 4 would be replaced by

$$\gamma_{ij}\begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} = K_j\left(R_jP_i - R_jC_j\right), \text{ for all } i,j,$$

where K_j is a diagonal matrix containing all the internal camera parameters, and we would need to eliminate all the K_j, R_j, C_j, and P_i. Observe that the right-hand-side of that equation corresponds to an affine transformation of the object point P_i. We can assume that the determinant of K_j is equal to one, in which case the transformation is restricted to an equi-affine transformation. The (joint) invariants of the group of equi-affine transformations on $\mathbb{R}^3$ are well known: they are generated by the volumes $v_{i_0i_1i_2i_3}$ spanned by sets of 4 points $P_{i_0}, P_{i_2}, P_{i_3}, P_{i_4} \in \mathbb{R}^3$, where

$$v_{i_0i_1i_2i_3} = \det\left(P_{i_1} - P_{i_0}, P_{i_2} - P_{i_0}, P_{i_3} - P_{i_0}\right).$$

Therefore, we have the following SFM equations,

$$\left|\gamma_{i_1j}p_{i_1j} - \gamma_{i_0j}p_{i_0j}, \gamma_{i_2j}p_{i_2j} - \gamma_{i_0j}p_{i_0j}, \gamma_{i_3j}p_{i_3j} - \gamma_{i_0j}p_{i_0j}\right| =$$

$$\left|\gamma_{i_1\bar{j}}p_{i_1\bar{j}} - \gamma_{i_0\bar{j}}p_{i_0\bar{j}}, \gamma_{i_2\bar{j}}p_{i_2\bar{j}} - \gamma_{i_0\bar{j}}p_{i_0\bar{j}}, \gamma_{i_3\bar{j}}p_{i_3\bar{j}} - \gamma_{i_0\bar{j}}p_{i_0\bar{j}}\right|,$$

for all distinct $i,\bar{i}$ and all distinct $j,\bar{j}$, which are free of both internal and external camera parameters. Note the geometric interpretation of these equations (volumes between the object points are preserved under equi-affine transforms), which extends the geometric interpretation for the fixed internal camera parameter case previously discussed (dot products, Euclidean distances and signed areas are preserved under orientation preserving rigid motions).

XII. Summary and Conclusion

We proposed a formulation of the problem of structure from motion (SFM) with fixed internal camera parameters in terms of polynomial equations of degree two and three in the depth of the object points with respect to the camera centers. This formulation, obtained by algebraic variable elimination, is equivalent to the standard SFM formulation but it does not involve any of the (external) parameters of the camera. By *equivalent*, we mean that it encodes exactly the same constraints on the depth parameters as the standard SFM

formulation: no constraint is added, and no constraint is removed. Thus its solution set corresponds to the projection of the solution set for the standard SFM formulation onto the space spanned by the depth variables. Moreover, the remaining unknowns all depend linearly on the depth parameters.

In small well-determined cases, our formulation can be solved in a global fashion using homotopy methods. In particular, the case of five points on two pictures can be solved under two minutes on a 2.66GHz Intel(R) Core(TM) 2 Duo Processor using a homotopy package for Maple [14]. This solution approach yields all solutions of the problem, including the complex ones, and the true solution can be selected as the only real positive one. In contrast, the same program was unable to solve the standard SFM formulation after running for more than 2 weeks. While the current running time may be too large for many applications, this result noteworthy from a theoretical point of view. In particular, it implies that algebraic-based methods could, potentially, be used to analyze and/or solve SFM cases that are difficult to solve numerically, such as the degenerate cases or cases where the object or picture points are defined by a parametric expression.

In the over-determined case, one can use our equations to formulate simple external-camera-pose-free cost functions to be minimized. Our experiments indicate that, given a noisy input, minimizing a pose-free cost function leads to a statistically more accurate solution than minimizing the total reprojection error as in the Bundle Adjustment method (BA). When all the possible equations are used to formulate the cost function, the minimization is more computationally expensive than minimizing the total reprojection error. One can select a smaller set of equations before formulating the cost function. If the set is small enough, the run time becomes comparable to that of minimizing total reprojection error, but the resulting accuracy is still significantly better. While the reconstruction error obtained with our approach is smaller than when using BA, even in the presence of outliers created by the Gaussian noise added to the points, it is still affected by noise: the more noise the less accurate the reconstruction. As with any other reconstruction method, the presence of mislabeled point is problematic, as the coordinates of mislabeled points can vary drastically from their true coordinates. Thus in the presence of outliers and/or mislabeling within a large number of points or pictures, our method should be combined with a RANSAC approach so to only used the best picture points available.

The total reprojection error is generally accepted to be the best measure of reconstruction accuracy. This belief is based on the fact that minimizing this error yields the maximum likelihood estimate when the image error is zero-mean Gaussian. Our results emphasize the importance of also considering numerical conditioning when deciding on a best solution strategy, as a biased estimator such as the one we propose can beat the accuracy of the maximum likelihood estimate.

We generalized of our approach to the case of a projective camera (i.e, when the internal camera parameters vary from one picture to the next). This extension was obtained using the invariants of the group of equi-affine transformations in $\mathbb{R}^3$. In future work, it would be interesting to study the advantages provide by this fully pose-free formulation of the general formulation of the SFM problem for a pinhole camera.

APPENDIX I

To show that Equations 8 form a complete set of camera-orientation free equations, we show that solving Equations 8 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 show that Equations 1 can be deduced from Equations 8. We begin by using a well known fact from invariant theory [18] which states that if some vectors $v_1, \ldots, v_N$ and $w_1, \ldots, w_N$ satisfy $v_i \cdot v_k = w_i \cdot w_k$, 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 $\gamma_{ij}\gamma_{1j}\gamma_{2j} p_{ij} \cdot p_{1j} \times p_{2j} = -(P_i - C_j) \cdot (P_1 - C_j) \times (P_2 - C_j)$, 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 obtain Equations 1.

APPENDIX II

To show that Equations 10 form a complete set of camera-pose-free equations, we need to show that solving Equation 10 for all P_i'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 C_j's and R_j's. Again we do this by showing that Equations 1 can be deduced from Equations 10. We use a fact from invariant theory which states that if vectors $v_1, \ldots, v_N$ and $w_1, \ldots, w_N$ satisfy $\|v_i - v_k\| = \|w_i - w_k\|$, for all $i, k = 1, \ldots, N$, then there exists an orthogonal matrix A and a translation vector T such that $v_i = Aw_i + T$, for all $i = 1, \ldots, N$. Thus, for every index j, there exists an orthogonal matrix A_j and a translation vector T_j such that

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

But the determinant of A_j cannot be negative, otherwise

$$\gamma_{4j}\gamma_{1j}\gamma_{2j}(p_{4j} - p_{3j}) \cdot (p_{1j} - p_{3j}) \times (p_{2j} - p_{3j}) = -(P_4 - P_3) \cdot (P_1 - P_3) \times (P_2 - P_3),$$

which contradicts the fifth equation (unless $P_4 - P_3$,$P_1 - P_3$ and $P_2 - P_3$ are co-planar.) Hence, each A_j is a rotation matrix.

APPENDIX III

To obtain a SFM formulation that does not contain 3D point parameters, we begin with the standard SFM equations (Equations 1) for five 3D points on a pair of pictures:

$$p_{ij} = c_{ij}(R_j P_i + T_h), \text{ for } i = 1, \ldots, 5 \text{ and } j = 1, 2.$$

By rigid motion invariance, we can set the first camera position at the origin of the coordinate system: $R_1 = \mathbb{I}$ and $T_1 = \mathbf{0}$. Then the first 5 equations become:

$$p_{i1} = c_{i1} P_i, \text{ for } i = 1, \ldots, 5.$$

Isolating the P_i in each equation and replacing into the remaining five equation sets and setting $\gamma ij = 1/c_{ij}$, we get

$$\gamma_{i2} p_{i2} = \gamma_{i1} R_2 p_{i1} + T_2, \text{ for } i = 1, \ldots, 5. \quad (22)$$

This gives us a system of equations where the 3D point parameters have been eliminated, i.e. a *depth-and-pose-only* SFM formulation. Using the same technique, one can eliminate

the 3D points parameters from *angle-free* SFM formulation to obtain a *depth-and-translation-only* SFM formulation.

$$\begin{aligned} \gamma_{i2}\gamma_{12}p_{i2}\cdot p_{12} &= (\gamma_{i1}p_{i1}-C_2)\cdot(\gamma_{11}p_{11}-C_2), \\ \gamma_{i2}\gamma_{22}p_{i2}\cdot p_{22} &= (\gamma_{i1}p_{i1}-C_2)\cdot(\gamma_{21}p_{21}-C_2), \\ \gamma_{i2}\gamma_{12}\gamma_{22}p_{i2}\cdot p_{12}\times p_{22} &= (\gamma_{i1}p_{i1}-C_2)\cdot(\gamma_{11}p_{11}-C_2) \\ &\quad \times(\gamma_{21}p_{21}-C_2), \end{aligned} \tag{23}$$

for all $i = 1, \ldots, 5$.

REFERENCES

[1] G. Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, 1989.

[2] Richard Aster, Brian Borchers, and Clifford Thurber. *Parameter Estimation and Inverse Problems (International Geophysics)*. Academic Press, har/cdr edition, January 2005.

[3] Pierre-Louis Bazin and Mireille Boutin. Structure from motion: a new look from the point of view of invariant theory. *SIAM J. Appl. Math.*, 64(4):1156–1174, 2004.

[4] Wieb Bosma, John J. Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *J. Symbolic Comput.*, 24:235–265, 1997.

[5] D.C. Brown. A solution to the general problem of multiple station analytical stereotriangulation. Technical Report 43, Patrick Airforce Base, Florida, 1958.

[6] Martin Byröd, Klas Josephson, and Kalle Åström. Fast and stable polynomial equation solving and its application to computer vision. *Int. J. Comput. Vision*, 84(3):237–256, 2009.

[7] O. D. Faugeras and Steve Maybank. Motion from point matches: multiple of solutions. *Int. J. Comput. Vision*, 4(3):225–246, 1990.

[8] Cornelia Fermüller and Yiannis Aloimonos. Observability of 3D motion. *Int. J. Comput. Vision*, 37(1):43–63, 2000.

[9] Daniel R. Grayson and Michael E. Stillman. Macaulay 2, a software system for research in algebraic geometry. available at `http://www.math.uiuc.edu/Macaulay2`, 1996.

[10] Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. *Symbolic computation and automated reasoning, The Calculemus-2000 Symposium*, chapter SINGULAR 3.0 — A computer algebra system for polynomial computations, pages 227–233. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[11] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proc. of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 237–256, London, UK, 1994. Springer-Verlag.

[12] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997.

[13] C. Jerian and R. Jain. Polynomial methods for structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1150–1166, 1990.

[14] Anton Leykin and Jan Verschelde. PHCmaple: A maple interface to the numerical homotopy algorithms in PHCpack. In *Proceedings of the Tenth International Conference on Applications of Computer Algebra (ACA'2004)*, pages 139–147, 2004. Software available at http://www.ima.umn.edu/∼leykin/PHCmaple/index.htm.

[15] H.D. Li. A simple solution to the six-point two-view focal-length problem. In *European Conference on Computer Vision*, pages IV: 200–213, 2006.

[16] M.I.A. Lourakis and A.A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug. 2004.

[17] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

[18] Peter J. Olver. *Classical invariant theory*, volume 44 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 1999.

[19] Jianbo Shi and Carlo Tomasi. Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, 1994.

[20] H. Stewenius, C. Engels, and D. Nister. Recent developments on direct relative orientation. *ISPRS J. Photogrammetry and Remote Sensing*, 60(4):284–294, June 2006.

[21] C. Tomasi. Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 913–918, June 1994.

[22] C. Tomasi and J. Shi. Direction of heading from image deformations. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 422–427, June 1993.

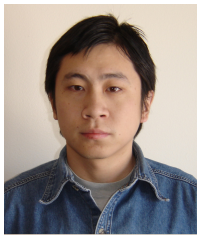
[23] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proc. of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag.

[24] Michael Werman and Amnon Shashua. The study of 3D-from-2D using elimination. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 473–479, 1995.

[25] Gem-Sun Jason Young and Rama Chellappa. Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy flow field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(10):995–1013, 1992.

[26] J. Zhang, M. Boutin, and D.G. Aliaga. Variable elimination for 3D from 2D. In *Visual Communication and Image Processing conference (VCIP), IS&T/SPIE joint symposium*, San Jose, CA, Jan-Feb 2007.

[27] Wang 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.



Ji Zhang Ji Zhang completed his doctoral work in mathematics under the mentorship of Mireille Boutin at Purdue University in May 2009. His research areas include Computational Algebraic Geometry, Computer Graphics and Vision. He is now working in Bloomberg LP as a Financial Software Developer.

Mireille Boutin Mireille Boutin is an Assistant Professor in the School of Electrical and Computer Engineering at Purdue University. Dr. Boutin obtained her Ph.D. degree in mathematics from the University of Minnesota under the direction of Peter J. Olver. Her current research interest include light-weight image processing for portable device applications and computational mathematics.

Daniel G. Aliaga Daniel G. Aliaga is an Associate Professor of Computer Science at Purdue University. He is a researcher in computer graphics and computer vision, and in particular in acquiring, modeling, and rendering 3D objects and scenes. Dr. Aliaga obtained his Ph.D. degree from the University of North Carolina (UNC). He has served on numerous program committees, on several NSF panels, as journal editor, as conference and paper chair, and authored over 60 papers.