Angle Independent Bundle Adjustment Refinement

Jeffrey Zhang
Dept. of Mathematics
zhang54@cs.purdue.edu

Daniel G. Aliaga
Dept. of Computer Science
aliaga@cs.purdue.edu

Mireille Boutin
Dept. of Electrical and Computer Engineering
mboutin@purdue.edu

Robert Insley
Dept. of Computer Science
rinsley@cs.purdue.edu

Purdue University
West Lafayette, IN 47907

Abstract

Obtaining a digital model of a real-world 3D scene is a challenging task pursued by computer vision and computer graphics. Given an initial approximate 3D model, a popular refinement process is to perform a bundle adjustment of the estimated camera position, camera orientation, and scene points. Unfortunately, simultaneously solving for both camera position and camera orientation is an ill-conditioned problem. To address this issue, we propose an improved, camera-orientation independent cost function that can be used instead of the standard bundle adjustment cost function. This yields a new bundle adjustment formulation which exhibits noticeably better numerical behavior, but at the expense of an increased computational cost. We alleviate the additional cost by automatically partitioning the dataset into smaller subsets. Minimizing our cost function for these subsets still achieves significant error reduction over standard bundle adjustment. We empirically demonstrate our formulation using several different size models and image sequences.

1. Introduction

The reconstruction of real-world environments is a challenging objective for image processing, computer vision, and virtual simulations. The ultimate goal is to obtain a digital 3D model so that, for example, students can visit famous historical sites; archeologists can capture excavation sites as they evolve over time; soldiers and fire fighters can train in simulated environments; and, people all over the world can enjoy virtual travel.

Recovering the geometry of a 3D scene is a difficult task. Several approaches, such as structure from motion (SFM) methods, attempt to extract the scene geometry from photographs acquired by a moving camera (e.g., [1][2][3]). In these situations, bundle adjustment (BA) is often used as a final refinement step. BA performs a global optimization so as to improve the coherence between the observed environment features and the reconstructed model. By exploiting the redundancy within the captured information, it has the advantage of being tolerant to missing samples and of being able to recognize and compensate for outliers in the dataset. Consequently, BA methods are able to improve the reconstruction of a 3D scene.

Bundle adjustment must, however, compensate for inconsistencies in both structure and motion (e.g., camera pose) – even if ultimately only the structure of the environment is desired. Differentiating between changes in camera position and camera orientation is unfortunately an ill-conditioned problem [4], which negatively affects the reconstruction of the geometry. Using traditional BA, the hope is that by using a batch approach, which over-constrains the problem, a reasonable solution can be obtained given a sufficiently accurate initial guess.

In this paper, we improve the robustness and convergence of BA by proposing a cost function that is independent of the camera angles. This alleviates the ill-posed aspect of bundle adjustment and results in a better solution. Our formulation uses a computational technique from invariant theory [5][6] to eliminate variables from a set of equations. Although minimizing our cost function is more computationally expensive than standard BA for a given batch of images, the improvement in accuracy allows us to subdivide the problem into smaller batches which can be computed more efficiently. The total computational time of our method can thus be kept similar to standard BA while still obtaining more accurate results. In numerical experiments, our method yields improved numerical performance that is both visibly and quantitatively better than standard bundle adjustment. We demonstrate comparisons using several objects and scenes ranging from a hundred to over 32000 scene points observed over 48 to 2644 images.

The major contributions of our work include

- a bundle adjustment method that is robust to variations in the initial model estimate,
- a degree-two polynomial formulation of the reconstruction problem that is free of any camera angles, and

- an automatic algorithm for subdividing the optimization into smaller sub-problems yielding an efficient refinement process for 3D reconstruction.

2. Related Work

Bundle adjustment has become popular in the computer vision community and is frequently used to improve upon structure-from-motion solutions [7]. Briefly, BA is the problem of improving a visual reconstruction to produce both optimal structure and optimal viewing parameter estimates. Assuming a Gaussian noise model, BA can be equated with a maximum likelihood estimator. An error minimization is performed numerically typically using a non-linear least squares method, such as Levenberg-Marquardt minimization [8]. On paper, BA may seem like an ideal algorithm.

Unfortunately, the extremely large number of parameters involved in practical problems makes finding a good solution difficult. Several approaches have been followed to reduce the number of parameters and to take advantage of the sparse solution matrix in order to improve convergence and speed-up the algorithm. Triggs [9] highlights several methods to partition or resection the data as well as interleaving methods to change what parameter group is optimized. Yet, other methods exploit the sparse nature of the problem to obtain faster algorithms [10].

But more importantly, BA attempts to solve for the camera pose, which is known to be an ill-conditioned problem [4]. This is an important issue since small errors in camera pose can lead to big errors in 3D reconstruction. Eliminating the pose, or at least the camera angles, from the cost function used in BA would thus be desirable, especially if this could be done without increasing the degree of the cost function.

The idea of eliminating camera orientation parameters to improve robustness has been previously exploited in other contexts. Tomasi and Shi [11] proposed structure-from-motion equations to compute the direction of heading of a camera -- their equations did not involve camera angles. Subsequently, they described image changes through the angles between the projection rays [12]. This approach was used to reconstruct a two-dimensional world and could theoretically be applied to a three-dimensional world although at significant additional complexity and with high-sensitivity to accurate camera calibration.

Removing camera parameters from the 3D reconstruction problem is a challenging task. This is mostly due to the complexity of variable elimination. Nevertheless, the approach of [5] describes a theoretical framework to algebraically remove camera orientation parameters. Their approach is suggested for use in several formulations of 3D reconstructions but few numerical reconstructions are performed and no comparison with other methods is provided. In this paper, we use this framework as a basis for deriving a new set of equations for bundle adjustment without dependence on camera angles. This leads to a more robust BA refinement. The downside of this new formulation is a higher computational cost, which can be alleviated by automatically partitioning the problem into smaller batches. Still, partitioning the problem yields more accurate results than BA.

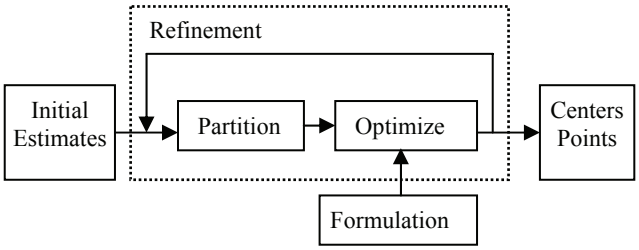


Figure 1. Algorithm Pipeline. *Given an initial estimate, our approach partitions the dataset and uses a novel angle-free formulation to efficiently refine the structure (points) and motion (centers) of a reconstructed model.*

In earlier work, we developed a related technique for improving bundle adjustment using a degree-four polynomial formulation [13]. In that method, while fewer unknowns need to be optimized, the expressions do not produce a single consistent set of both scene points and camera centers (as with typical bundle adjustment methods) and the computations are not automatically partitioned into sub-problems and thus cannot handle large datasets.

3. Invariant-Based Bundle Adjustment

We develop a new bundle adjustment formulation that is more robust to variations in the initial guess and is independent of the camera orientation. Our cost function is a polynomial of degree two in the projective coordinates of the 3D feature points and the camera centers. Note that standard BA also uses polynomials of degree two. But our cost function has significantly more terms than standard BA (i.e., $O(JN^2)$ versus $O(JN)$ for J images and N features). However, the improved resilience of our method allows us to optimize the data using disjoint subsets, rather than the whole batch of data, while still obtaining better numerical results than BA (Figure 1).

The key idea used to obtain this new mathematical formulation follows a computational approach which rewrites the SFM equations in terms of the invariants of a

group transformation. More precisely, we express all the possible reconstructions for the observed 3D scene points as the result of a group transformation parameterized by the variables to eliminate. The invariants of this group transformation are, by definition, functions whose values are unchanged by the group transformation and thus do not depend on the variables to eliminate. We know the projection of a 3D scene point onto the image plane is itself a possible 3D reconstruction. Therefore, this reconstruction can be mapped to the 3D scene point by the group transformation. This implies that the invariants have the same value when evaluated using the actual 3D scene points as when evaluated using the projections of the 3D points. By evaluating explicit expressions for the invariant functions using scene points on the image plane, we obtain a set of reconstruction equations.

3.1. Formulation

Our goal is to determine the 3D positions of some tracked features from their observed positions in an image stream. The equations relating the tracked features and their projections can be written as follows:

$$c_{ij}F_j\begin{pmatrix}P_i\\1\end{pmatrix}-\begin{pmatrix}p_{ij}\\1\end{pmatrix}=\begin{pmatrix}0\\0\\0\end{pmatrix}, \qquad (1)$$

where p_{ij} represents the 2D coordinates of the 3D feature point P_i observed on picture j, c_{ij} is a constant, and F_j is a 3-by-4 matrix containing the camera parameters corresponding to picture j. The index i is assumed to take values from 1 to N, where N is the number of tracked features, and the index j, values from 1 to J, where J is the number of pictures taken. When the camera is internally calibrated, one can assume that the matrix F_j 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. When N and J are big enough, Equations (1) form an over-determined system. In practice, this system has only approximate solutions, because the tracked features are not measured exactly. To find an approximate solution, one attempts to make the right-hand-side of these equations close to zero. In BA, this is formulated as a least-squares problem: one demands that the sum of the squares of the left-hand side expressions (the cost function) be as close as possible to zero. The camera pose is part of the cost function through the matrices F_j. In order to formulate a better cost function, we propose to eliminate the rotation matrices from Equations (1).

Eliminating parameters from Equations (1) is not straightforward, even though they can be viewed as polynomial equations. Indeed, one could think that the symbolic elimination tools developed for the case of polynomial equations (e.g., Singular [14] and Macaulay [15]) would be well suited for eliminating the rotation parameters. Unfortunately, the set of equations we are dealing with is so big and involves so many variables that these programs cannot handle the size of the problem. In contrast, the moving frame elimination method [5] can be used in a more or less straightforward manner. Using this method, we can obtain a set of equations which is equivalent to Equations (1) but does not involve any angle. This set of equations forms a "basis" for all SFM equations that do not involve the camera orientation. By "basis", we mean that any other camera orientation-free SFM equation can be written (locally) as a function of these equations.

In order to follow this approach, we first need to express the SFM problem as a group transformation where the group parameters contain the variables to be eliminated, i.e. the camera angles. Equations (1) do not represent a group transformation. However, the following equivalent set of equations does represent a group transformation:

$$C_j=R_j\begin{pmatrix}0, & 0, & -1\end{pmatrix}^T+T_j, \qquad (2)$$

$$P_j=R_j\left[\begin{pmatrix}x_{ij}\\y_{ij}\\0\end{pmatrix}+\lambda_{ij}\left[\begin{pmatrix}x_{ij}\\y_{ij}\\0\end{pmatrix}-\begin{pmatrix}0\\0\\-1\end{pmatrix}\right]\right]+T_j. \qquad (3)$$

Equation (2) describes a mapping from a canonical camera center at (0, 0, -1) looking down the +z axis to the true camera center: for every image j, there exists a 3D rotation matrix R_j and a 3D translation vector T_j such that (2) holds. Equation (3) describes a mapping from a canonical scene point projection to an actual scene point. We assume, for simplicity, that the distance from the camera center to the image plane is constant with focal length $f=1$ and that the images have been undistorted by internal camera parameters such that there is no radial distortion, no skew, and the aspect ratio is one. For $i=1...N$ and $j=1...J$, we write $p_{ij} = (x_{ij}, y_{ij})$. Then, there exists a number λ_{ij} (related to scene point depth) such that (3) holds. The parameters of the group action are R_j, T_j, and λ_{ij} for $i=1...N$ and $j=1...J$. The projective space equivalent of Equations (2) and (3) is (where W's and w's are the variables for the fourth coordinate in projective space):

$$\begin{pmatrix}W_{0j}C_j\\W_{0j}\end{pmatrix}=\begin{pmatrix}R_j & & & T_j\\0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}0\\0\\-w_{0j}\\w_{0j}\end{pmatrix}, \qquad (4)$$

$$\begin{pmatrix} W_{ij}P_{ij} \\ W_{ij} \end{pmatrix} = \begin{pmatrix} R_j & T_j \\ 0_{3\times3} & 1 \end{pmatrix} \left[\begin{pmatrix} w_{ij}x_{ij} \\ w_{ij}y_{ij} \\ 0 \\ w_{ij} \end{pmatrix} + \lambda_{ij} \left[\begin{pmatrix} w_{ij}x_{ij} \\ w_{ij}y_{ij} \\ 0 \\ w_{ij} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ -w_{0j} \\ w_{0j} \end{pmatrix} \right] \right]. \quad (5)$$

By working in projective space rather than Euclidean space, the invariants of the corresponding group action turn out to be polynomial functions, as opposed to rational in the Euclidean case. This greatly simplifies the numerical solution process.

A generating set of invariants I_i, J_i, and H_i, for $i=1...N$, of the group transformation corresponding to equations (4) and (5) was obtained using the moving frame elimination method:

$$I_i(*) = \frac{\omega_i}{\omega_i - \omega_0} \frac{\omega_1}{\omega_1 - \omega_0} Q_i \cdot Q_1$$

$$J_i(*) = \frac{\omega_i}{\omega_i - \omega_0} \frac{\omega_1}{\omega_1 - \omega_0} Q_1 \times Q_i \cdot Q_1 \times Q_2$$

$$H_i(*) = \frac{\omega_i}{\omega_i - \omega_0} \frac{\omega_1}{\omega_1 - \omega_0} Q_i \cdot Q_1 \times Q_2 \quad (6)$$

where

$$* = \left(\begin{pmatrix} \omega_0\rho_0 \\ \omega_0 \end{pmatrix}, \begin{pmatrix} \omega_1\rho_1 \\ \omega_1 \end{pmatrix}, ..., \begin{pmatrix} \omega_N\rho_N \\ \omega_N \end{pmatrix} \right)$$
$$Q_i = (\rho_i - \rho_0).$$

By construction, the invariants take constant values for all possible 3D scene points of image j, including the canonical scene point projections (because they are invariant under a group transformation and because the projections and the possible 3D scene points are related by a group transformation.) Thus, we obtain the equations

$$\begin{aligned} I_i(\sigma_j) &= I_i(s_j) \\ J_i(\sigma_j) &= J_i(s_j) \\ H_i(\sigma_j) &= H_i(s_j) \end{aligned} \quad (7)$$

for every $i=1...N$ and $j=1...J$, where σ_j represents projective coordinates of the canonical scene points and camera centers while s_j represents the actual scene points and camera centers. More precisely, σ_j and s_j are defined as

$$\sigma_j = \left(\begin{pmatrix} 0 \\ 0 \\ -w_{0j} \\ w_{0j} \end{pmatrix}, \begin{pmatrix} w_{1j}x_{1j} \\ w_{1j}y_{1j} \\ 0 \\ w_{1j} \end{pmatrix}, ..., \begin{pmatrix} w_{Nj}x_{Nj} \\ w_{Nj}x_{Nj} \\ 0 \\ w_{Nj} \end{pmatrix} \right)$$
$$s_j = \left(\begin{pmatrix} W_0C_j \\ W_0 \end{pmatrix}, \begin{pmatrix} W_{1j}P_1 \\ W_{1j} \end{pmatrix}, ..., \begin{pmatrix} W_{Nj}P_N \\ W_{Nj} \end{pmatrix} \right)$$

We can arbitrarily fix $W_i=1$ for all $i=1...N$ and $w_{0j}=W_{0j}=2$ for all $j=1...J$. Then Equations (7) become

$$\begin{aligned} \gamma_{ij}\gamma_{1j}k_{1ij} &= (P_i - C_j)\cdot(P_1 - C_j) \\ \gamma_{ij}\gamma_{2j}\gamma_{1j}^2 k_{ij} &= ((P_1 - C_j)\times(P_i - C_j))\cdot((P_1 - C_j)\times(P_2 - C_j)) \\ \gamma_{ij}\gamma_{2j}\gamma_{1j}k_{3ij} &= (P_i - C_j)\cdot((P_1 - C_j)\times(P_2 - C_j)) \end{aligned} \quad (8)$$

where,

$$k_{1ij} = ((x_{ij}, y_{ij}, -1)\cdot(x_{1j}, y_{1j}, -1)$$
$$k_{2ij} = ((x_{1j}, y_{1j}, -1)\times(x_{ij}, y_{ij}, -1))\cdot((x_{1j}, y_{1j}, -1)\times(x_{2j}, y_{2j}, -1))$$
$$k_{3ij} = (x_{ij}, y_{ij}, -1)\cdot((x_{1j}, y_{1j}, -1)\times(x_{2j}, y_{2j}, -1))$$

and the γ's are unknown variables whose general form corresponds to

$$\gamma_{ij} = \frac{w_{ij}}{w_{ij} - w_{0j}} = \frac{\|P_i - C_j\|}{\|P_{ij} - (0,0,-1)\|}.$$

This new set of SFM equations is a basis that generates all other angle-free SFM equations. In a generic case and with a large enough number of points and images, this whole system of equations contains more equations than unknowns and thus we can attempt to solve for the scene points. The system of equations contains $3NJ$-$3J$ non-trivial equations (because both the left-hand-side and right-hand-side of the first equation of (7) are zero when i is 1 or 2 and similarly for the third equation of (7) when i is 1) that are functionally independent. The number of unknowns is only $3N$+$3J$. Thus for large enough N and J, the system is over-determined.

Observe that the second and third equations of (8) are of degree four and three, respectively. Ideally, we would like to use polynomial equations of the same degree as in BA. We thus should restrict ourselves to combinations of the Equations of (8) which lead to equations of degree two only. For example, we could use the superset of equations

$$(P_{i_1} - C_j)\cdot(P_{i_2} - C_j) - \gamma_{i_1 j}\gamma_{i_2 j}K_{i_1 i_2 j} = 0 \quad (9)$$

where $K_{i_1 i_2 j} = ((x_{i_1 j}, y_{i_1 j}, -1)\cdot(x_{i_2 j}, y_{i_2 j}, -1)$

for all i_1, $i_2=1,...,N$ (not necessarily distinct) and all $j=1,...,J$. Intuitively, we seek for a solution to the 3D points and camera centers so that the angles between the vectors from the camera centers to the scene points is the same as the angles between the vectors from the camera centers to the points on the image plane.

It turns out that the above set of equations is almost equivalent to (8). The only information from (8) which is not contained in (9) is the sign of the left-hand-side of the second equation of (8). This has the consequence that there may be slightly more solutions to (9) than to the standard SFM equations. However, since a high number of pictures and features are used in practice, this is not an issue. Indeed, in most cases, we end up with an over-constrained system of equations for which the solutions will typically be the same as for (8).

We formulate a cost function by adding the squares of the left-hand-side of (9) for all images and features:

$$\sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\sum_{j=1}^{J}\left[\left(P_{i_1}-C_j\right)\cdot\left(P_{i_2}-C_j\right)-\gamma_{i_1 j}\gamma_{i_2 j}K_{i_1 i_2 j}\right]^2 . \quad (10)$$

As promised, this cost function is of minimal degree and does not involve any camera angle, so we bypass the problem of ill-condition in determining the camera motion.

To solve this equation, we use initial guesses for the camera centers C_j, 3D points P_i, and γ's (the K's are computed from the projections p_{ij}) and employ a Conjugate Gradient method. While initial guesses for the γ's are obtained using the expressions in equations (8) and we solve for them, we disregard them at the end of the optimization.

3.2. Image Sequence Partitioning

The cost function described by Equation (10) contains at most O(JN^2) terms (as opposed to O(JN) with standard BA). The actual number of terms varies depending on how many images each feature is tracked on. To compensate for the additional computational cost, we optionally partition the image sequence into disjoint subsets (i.e. each point and camera center is solved for only once). These subsets can be processed individually (and in parallel) yielding an overall reduction in computation time.

We use two strategies to subdivide the images and features depending on the acquisition style. For an inside-looking-out sequence through a large environment (e.g., a video recorded during a walkthrough), a convenient partitioning is to create disjoint but contiguous image sequences through the model. For a typical outside-looking-in sequence around an object (e.g., a video capturing a single object), a suitable partitioning is to create interleaved and disjoint sets of images observing the same portion of the scene.

4. Results and Discussion

In this section, we present results and comparisons of applying our novel formulation for bundle adjustment to several example scenes. Table 1 provides a summary of our three test datasets. Our datasets range from 96 to 32688 reconstructed points and from 48 to 2644 images. The *board* dataset consists of features tracked along an image sequence observing a chessboard of known dimensions. The images were acquired by a camera attached to a mechanically tracked arm [16]. This dataset provides us with ground truth information for both camera centers and scene points. *Giraffe* was captured using our in-house real-time acquisition system: a handheld camera acquires images and initial pose estimates are provided by tracking four landmarks (e.g., small light boxes) and triangulating position and orientation from them. *Floor* is a synthetic dataset consisting of several rooms of a radiosity-illuminated model of the single floor of a house.

Dataset	Board	Giraffe	Floor
Images	48	360	2644
Points	96	480	32688

Table 1. Datasets*. We applied our reconstruction algorithm to four datasets. "Images" is the number of captured images and "points" is the number of reconstructed 3D points.*

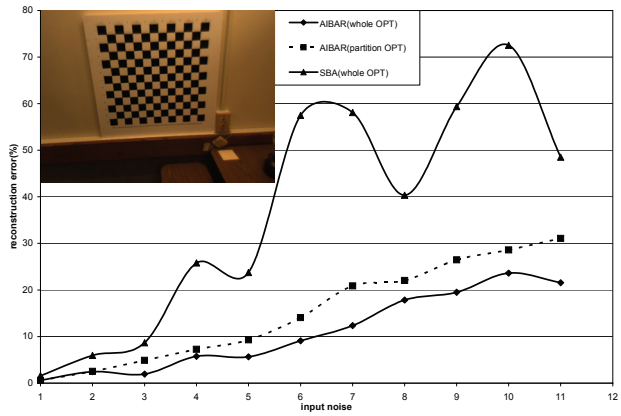


Figure 2. Angle-Independent Bundle Adjustment Refinement*. Our method (AIBAR) converges to lower error solutions than an efficient sparse bundle adjustment solution (SBA). To compensate for the additional computational time of our method, we can partition the dataset in order to obtain comparable solutions times to standard BA but with only a small increase in the amount of error.*

We have implemented our system on a Pentium IV PC using C/C++. Feature tracking is performed using an automatic algorithm based on the Kanade-Lucas-Tomasi tracking software package. Our method requires at least 6 features tracked over 3 frames. To perform all standard BA computations, we use an efficient sparse bundle adjustment package [10].

Using our novel formulation, we are able to converge to a more accurate solution than standard BA even under the presence of significant noise (or inaccuracy) in the initial solution. Figure 2 uses the board dataset to show this comparison. The horizontal axis corresponds to the amount (standard deviation) of random Gaussian noise added to the initial guess for camera pose (position and orientation) and scene points. Since we have the ground truth for this dataset, we start with zero error and progress until approximately 40% of the model diagonal in positional error and 12 degrees of rotational error. For

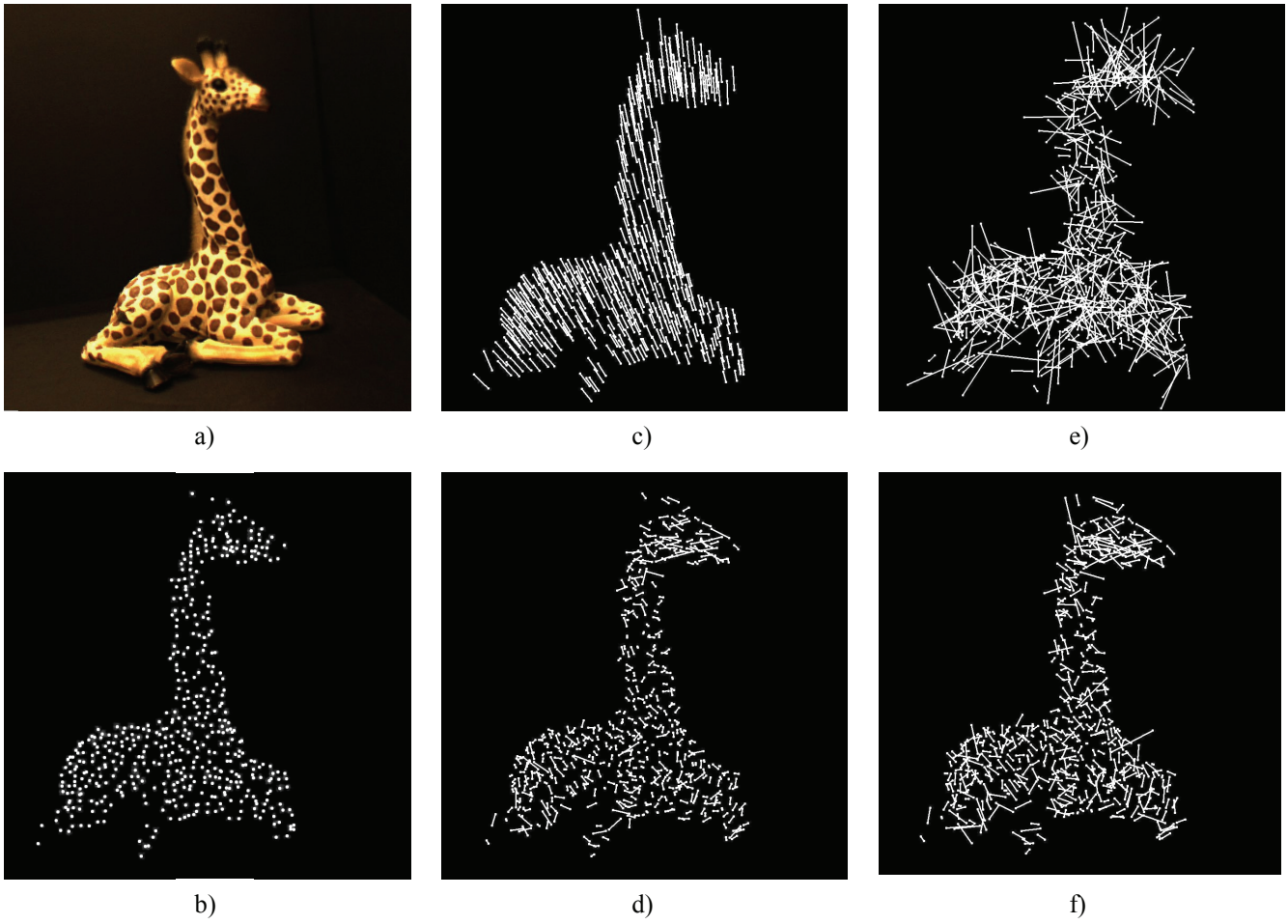
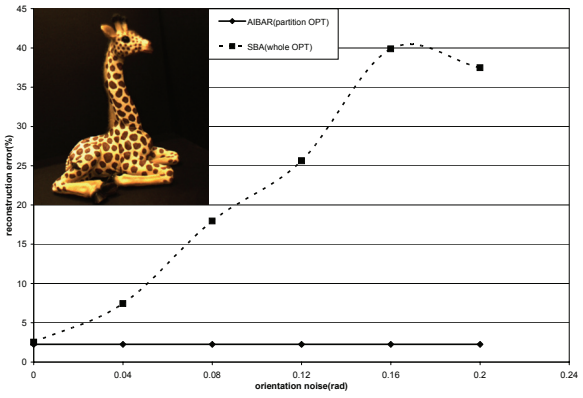


Figure 3. Reconstruction Examples*. We show several reconstructions of the giraffe dataset using our method (bottom row) and using standard BA (top row). The error of the different reconstructions is shown by drawing lines between the reconstructed points and the highest quality reconstruction (e.g., ground truth). (a) Example input image from giraffe sequence. (b) High-quality reconstruction using our method. (c) Medium-quality solution using standard BA. (d) Medium-quality solution using our method. (e) Low-quality solution using standard BA. (f) Low-quality solution using our method.*

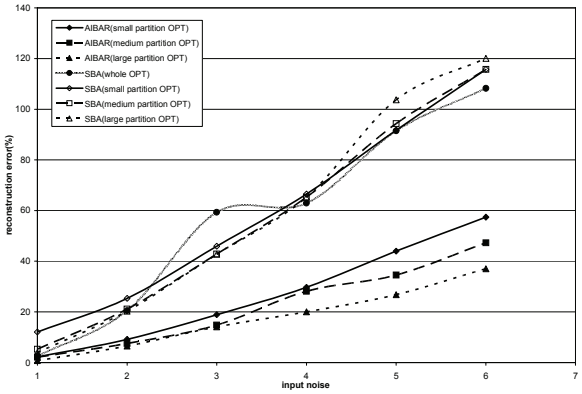
each data point of this graph (and all graphs in the paper), the vertical axis shows the average scene point reconstruction error over four reconstruction attempts, expressed as a percentage of the model diagonal. For the board dataset, the reconstruction errors produced by our formulation are, on average, about 3.3 times less than using standard BA.

Our formulation does have an additional computational cost which we can effectively combat with partitioning. On average for the board dataset, using our computing platform and formulation applied to the entire image sequence takes 838 seconds to compute as opposed to standard bundle adjustment which takes only 12 seconds to compute. Both of the methods essentially solve a nonlinear least squares problem. The increased cost is due to the additional terms that must be evaluated for our method. However, the improved numerical performance of our approach allows us to partition the problem into smaller subsets and still obtain approximately the same solution. The board sequence has 48 images which we divide into 8 interleaved datasets that are independently processed for our method. As seen in Figure 2, our partitioned solution demonstrates similar behavior to our whole solution but it only takes 14 seconds to compute, on average. Despite the partitioning, our solution is better than standard BA applied to the entire sequence. We experimented with partitioning standard BA but observed similar or larger reconstruction errors.

Figure 3 demonstrates images for various reconstructions of the giraffe dataset, an example of outside-looking-in dataset. Figure 3a contains an original image from the giraffe dataset. In place of ground truth for this dataset, we use our best estimate (Figure 3b) which is similar to BA's best estimate. Figures 3c and 3d show



(a)



(b)

Figure 4. Reconstructing Scene Points*. (a) Our approach (AIBAR) is not sensitive to orientation errors in the initial estimates while standard bundle adjustment (SBA) is sensitive. (b) Partitioning accelerates our algorithm (as well as SBA) but we consistently obtain lower-error solutions*.

reconstructions for standard bundle adjustment and for our method using a medium amount of Gaussian noise in the initial estimates. Figures 3e and 3f illustrate the corresponding reconstructions using a larger amount of Gaussian noise. In Figure 3c, BA converged to a coherent solution but offset from correct. In Figure 3e, BA did not converge. At even larger errors, BA diverges even more severely. On the other hand, our technique consistently produces lower-error reconstructions as we decrease the accuracy of the initial guesses.

Figures 4 and 5 contain the quantitative errors of our formulation using interleaved-partitioning vs. standard BA for the giraffe dataset. Figure 4a demonstrates that as compared to standard BA, our reconstructions are invariant to errors in initial camera orientation estimates. Figure 4b shows errors for the reconstructed 3D points in the presence of increasing error in the initial estimates for both camera centers and scene points. Similarly, Figure 5 shows errors for the recovered camera centers for various amounts of error in the initial estimates. In all cases, our approach shows a clear improvement over standard bundle adjustment.

We experimented with using various partitioning sizes. As seen in Figure 4b, our reconstruction errors are approximately similar albeit at very different computational costs (partitions into small, medium, and large subsets take on average 88, 253, and 961 seconds respectively). We performed similar experiments using standard bundle adjustment. While we do improve computational costs (partitions into small, medium, and large subsets take on average 7, 12, and 43 seconds respectively), the errors produced are less controlled. We believe the observed oscillations to be due to the inability of standard BA to find a truly overall improved solution because of the inherent confusion between camera position and camera orientation. As larger subsets are used, it simply latches on to one particular solution and converges to a local minimum (e.g., Figure 3c converged to a solution that is mostly translated away from the actual solution while Figure 3e did not converge well to any solution).

Using the floor dataset, we demonstrate reconstructions using contiguous partitioning in a large inside-looking-out dataset. In this example, we have partitioned the dataset of 2644 images into 155 partitions of approximately equal number of images. In Figures 6a and 6b, we show a view of the floor dataset and a representative view of the reconstruction of the scene points using our method. Figure 6c shows the reconstruction error performance of our method vs. standard BA. Our approach consistently exhibits less reconstruction error.

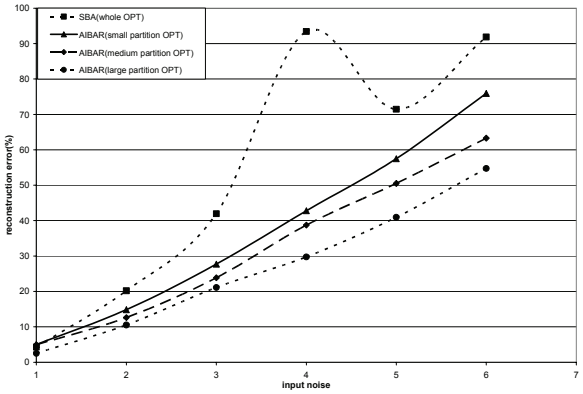


Figure 5. Recovering Camera Centers. *This graph shows how effectively our method recovers camera centers, using several partition sizes, as compared to standard bundle adjustment.*

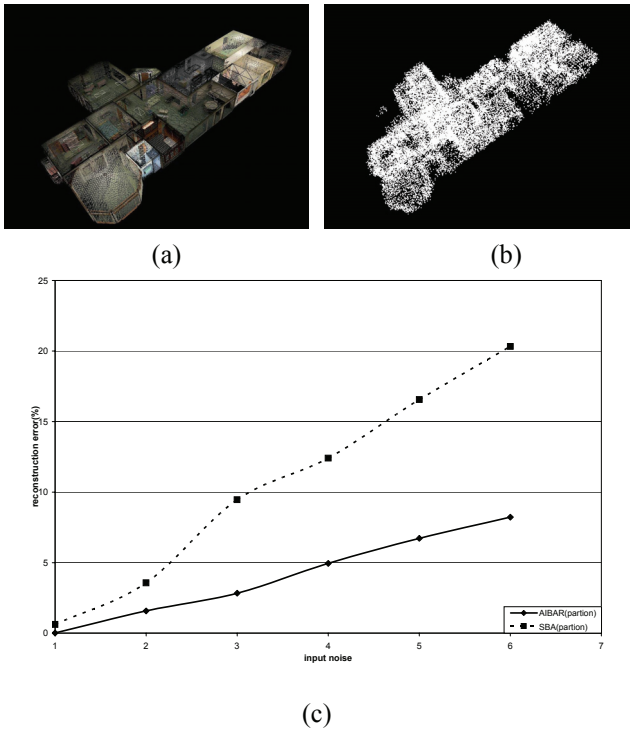


Figure 6. Outside-looking-in Example*. (a) A view of the synthetic floor dataset. (b) A reconstruction of the dataset using our method and small error Gaussian error. (c) Graph of the reconstruction error of our method vs. standard BA.*

5. Conclusions and Future Work

We have presented a degree-two polynomial formulation of structure from motion as well as an associate cost function for bundle adjustment that allows us to refine a structure from motion solution independently of camera angle estimation. By omitting camera angles from the formulation, we are able to disambiguate the inherent confusion between camera centers and camera orientation and thus obtain a more numerically robust process. Our approach introduces more terms, as compared to standard BA, into the cost function. However, we have shown how to partition the dataset into disjoint sets achieving similar computational times as standard BA but with better convergence. We have applied our method to the bundle adjustment of several models, including outside-looking-in and inside-looking-out models, and demonstrate the improved performance of our technique under varying amounts of Gaussian noise in the estimate.

Looking forward, we are currently interested in several avenues of future work. First, we are pursuing how to exploit the sparseness of the feature space over all images in order to obtain a more compact set of equations and cost terms. Second, to further improve the partitioning of the dataset, we would like to explore its cost-benefit space. Our current partition sizes are hand picked to be reasonable guesses for the datasets. Automating this selection would allow us to find a (near) optimal balance of error reduction and computation time. Third, we are investigating how to specialize our equations to perform fast and accurate camera calibration, including optimizing internal camera parameters as well during optimization. Finally, we believe the work of this paper will lead to significant improvements in bundle adjustment and in structure from motion. In addition, we are seeking for a formulation that allows us to completely bypass having to estimate camera centers on the way to recovering the structure. This and future work will significantly change how we think about reconstructing 3D scenes.

References

[1] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method", *IJCV* Vol. 9, No. 2, pp. 137-154, 1992.

[2] D. Nister, "Automatic Passive Recovery of 3D from Images and Video", *3DPVT,* pp. 438-445, 2004.

[3] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, "Visual modeling with a hand-held camera", *IJCV,* Vol. 59, No. 3*,* pp. 207-232, 2004.

[4] C. Fermüller and Y. Aloimonos, "Observability of 3D Motion", *IJCV*, Vol. 37, No. 1, pp. 43-62, 2000.

[5] P.-L. Bazin, M. Boutin, "Structure from motion: a new look from the point of view of invariant theory", *SIAM J. APPL. MATH*, Vol. 64, No. 4, pp. 1156-1174, 2004.

[6] M. Fels and P. Olver, "Moving Coframes. I. A practical algorithm", *Acta Appl. Math*, No. 51, pp. 161-213, 1998.

[7] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision", *Cambridge Univ. Press*, 2004.

[8] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, "Numerical Recipes in C", *Cambridge Univ. Press*, 1999.

[9] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis", *Vision Algorithms: Theory and Practice*. Springer-Verlag, 2000.

[10] M. I. A. Lourakis, A. A. Argyros, "The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm", *Institute of Computer Science – FORTH*, Heraklion, Greece, 2004.

[11] C. Tomasi and J. Shi, "Direction of heading from image deformations", *IEEE CVPR*, pp. 422-427, 1993.

[12] C. Tomasi, "Pictures and Trails: a New Framework for the Computation of Shape and Motion from Perspective Image Sequences", *IEEE CVPR*, pp. 913-918, 1994.

[13] J. Zhang, M. Boutin, D. Aliaga, "Robust Bundle Adjustment for Structure from Motion", *IEEE ICIP*, 2006.

[14] V. Levandovskyy G.M. Greuel and H. Schonemann, "Singular::Plural 2.1", Computer Algebra System for Noncommutative Polynomial Algebras.

[15] D. R. Grayson and M. E. Stillman, "Macaulay 2", Software system for research in algebraic geometry.

[16] Immersion Corporation, *http://www.emicroscribe.com*.