Efficient Bundle Adjustment with Virtual Key Frames: A Hierarchical Approach to Multi-frame Structure from Motion

Heung-Yeung Shum
Microsoft Research, China
Beijing 100080, PRC
hshum@microsoft.com

Qifa Ke
Carnegie Mellon University
Pittsburgh, PA 15213, USA
qifa.ke@cs.cmu.edu

Zhengyou Zhang
Microsoft Research
Redmond, WA 98052, USA
zhang@microsoft.com

Abstract

In this paper we present an efficient hierarchical approach to structure from motion for long image sequences. There are two key elements to our approach: accurate 3D reconstruction for each segment and efficient bundle adjustment for the whole sequence. The image sequence is first divided into a number of segments so that feature points can be reliably tracked across each segment. Each segment has a long baseline to ensure accurate 3D reconstruction. To efficiently bundle adjust 3D structures from all segments, we reduce the number of frames in each segment by introducing "*virtual key frames*". The virtual frames encode the 3D structure of each segment along with its uncertainty but they form a small subset of the original frames. Our method achieves significant speedup over conventional bundle adjustment methods.

1 Introduction

Structure from motion (SFM) simultaneously recovers the 3D structure of a scene or object and the camera motion (rotation and translation) associated with the images. SFM has been studied extensively because of its applications in robotics, video editing, and image-based modeling and rendering.

Methods for structure from motion include fatorization (affine [14] and perspective [11]), sequential or recursive estimation (e.g.,[7]) and bundle adjustment [12, 6]. Bundle adjustment is optimal in terms of minimizing reprojection error by varying the structure and camera motion. There are, however, two problems with conventional bundle adjustment. First, because bundle adjustment is a nonlinear minimization process, it is critical to have a good initial estimate of both 3D structure and camera motion. Second, bundle adjustment is computationally expensive because it involves all input frames and features.

All SFM methods depend on accurate feature correspondence. Traditionally, SFM employs an optical flow based point tracker. Because optical flow is based on brightness constancy and a translational motion model, it is accurate when the motion is small (or the baseline between frames is short). Unfortunately, the epipolar constraint (and 3D reconstruction) only becomes reliable when the baseline is sufficiently long.

This trade-off between the photometric (optical flow) and the geometric (epipolar) constraint leads us to the efficient hierarchical SFM approach proposed in this paper. We divide a long image sequence into a number of segments such that feature points are not only reliably tracked across each segment, but also form a sufficiently long baseline. Two main aspects of our approach are accurate 3D reconstruction for each segment and efficient bundle adjustment for the whole sequence.

For each segment, we initialize its bundle adjustment process by first solving a few long-baseline two-frame SFM problems, and then interpolating motion parameters of in-between frames. After merging local 3D models from segments onto a common coordinate system, a final bundle adjustment can be applied to the whole sequence. Instead of using all image frames to generate a complete 3D reconstruction, however, we use only representative frames (called *virtual key frames*) from each segment so that bundle adjustment can be done efficiently.

Another interesting feature of our SFM system is our tracker. We interactively select regions of interests, number of features, and add new features at any time of the tracking process. Features are then generated automatically in each selected region so that they are more uniformly distributed in 3D space, leading to a more accurate reconstruction.

This paper is organized as follows. We introduce our hierarchical multi-frame SFM system in Section 2. We show how to initialize local SFM for each segment and how to merge all local 3D models to obtain a final 3D model. After a brief review of bundle adjustment and complexity analysis in Section 3, we present in Section 4 how to perform efficient bundle adjustment using *virtual key frames* for each segment. In Section 5, we introduce our interactive point tracker and discuss how to segment the input sequence using

0-7695-0149-4/99 $10.00 © 1999 IEEE

both photometric and geometric constraints. We close with experimental results and discussion.

2 Hierarchical multi-frame SFM

2.1 Multi-frame SFM: previous work

Given observed corresponding image features $\{\mathbf{u}_{ik} = (u_{ik}, v_{ik}, 1)\}$, structure from motion can be formulated as the recovery of a set of 3D structure parameters $\{\mathbf{x}_i = (X_i, Y_i, Z_i)\}$ and time-varying motion parameters $\{(\mathbf{R}_k, \mathbf{t}_k)\}$.

There has been a large body of literature on SFM. We refer the reader to [3] for an introduction to structure and motion from image correspondence, and [16] for a review of two-frame SFM.

In general, multi-frame SFM works in a sequential manner because of the sequential process of establishing correspondence and merging local structures [2]. Among all SFM methods, bundle adjustment is known to be optimal in terms of minimizing reprojection error in (measurable) image space. While bundle adjustment technique is now used in most SFM systems (perhaps as post-processing after sequential methods) to obtain optimal 3D models, little work has been done on its complexity analysis because most systems use relatively short or sparse image sequence. In practice, however, structure from motion is often applied to a long video sequence.

To optimally distribute reconstruction errors over the sequence, Fitzgibbon and Zisserman [4] recently presented a hierarchical approach by building local structures from image triplets. They assume that the sequence is sparse and each triplet forms a long baseline. Unfortunately, for a dense sequence, triplets with short baselines will result in unreliable 3D models.

2.2 Hierarchical SFM: overview

In this section, we present our hierarchical approach to multi-frame structure from motion, constructing local model with a 2-frame SFM algorithm, merging local models to a partial model for each segment, and combining partial models into a complete 3D model. As shown in Figure 1, our approach consists of the following steps:

- divide the whole sequence into several segments;
- for each segment:
  1. initialize local models with a 2-frame SFM algorithm;
  2. combine multiple local models by eliminating scale ambiguity;
  3. bundle adjust to obtain a partial 3D model for the segment;
- merge all partial models into a common coordinate frame;

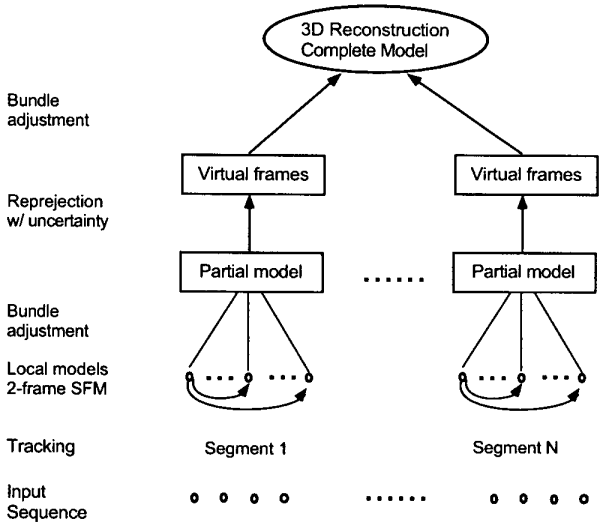

Figure 1: System overview: from image sequence to 3D reconstruction.

- extract virtual key frames for each segment;
- bundle adjust all segments to obtain a complete 3D model.

2.3 Local model from two-frame SFM

For each segment, its first frame is selected as the base frame or the reference frame. The simplest way to initialize multi-frame SFM for each segment is to solve a two-frame SFM between the base frame and the last frame and interpolate motion parameters for all in-between frames. Robust two-frame SFM [16] is used in our system but other robust techniques such as RANSAC [2] can be used as well. It is important to note that the two-frame SFM result (e.g., camera translation) has an unknown scale.

For a long segment, we may want to generate a few more local models for some intermediate frames to get better interpolation (initialization). The baseline for any two-frame SFM should be sufficiently long in order to obtain an accurate 3D reconstruction.

2.4 Partial model for each segment

To combine two local models (from two different two-frame SFM's but with the same base frame), we need to resolve the unknown scale between them.

For example, assume there are 21 frames which are divided into 2 intervals. For each interval, we compute local models of $\{0, 10\}$ and $\{0, 20\}$. The scale between the first model $\{0, 10\}$ and the second model $\{0, 20\}$ can be computed by:

$$\min_s \sum_i (\mathbf{x}_i^{(1)} - s\mathbf{x}_i^{(2)})^2,$$

where $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are the corresponding 3D points of the first model and second model.

Linearly interpolating motion parameters for in-between frames completes the initialization step before bundle adjusting a partial model for the segment. This avoids using unreliable short baseline two-frame SFM. Good initialization is very important for bundle adjustment due to the nature of nonlinear optimization.

2.5 Complete model by merging segments

To merge all partial models into a complete model, we need to solve the following optimization problem

$$\min_{s_{jk},\mathbf{R}_{jk},\mathbf{t}_{jk}} \sum_i (\mathbf{x}_i^{(j)} - s_{jk}(\mathbf{R}_{jk}\mathbf{x}_i^{(k)} + \mathbf{t}_{jk}))^2$$

where s_{jk} is scale, R_{jk}, t_{jk} is the rotation and translation between the two partial models of local segments j and k.

3 Bundle adjustment

Bundle adjustment performs simultaneous optimization of 3D point and camera placements by minimizing the squared error between estimated and measured image feature locations. Bundle adjustment has been studied extensively in Photogrammetry (e.g., [8]) where accurate correspondence is obtained mostly manually. There are two general approaches to bundle adjustment. The first interleaves structure and motion estimation stages [13], while the second simultaneously optimizes for structure and motion [15, 12, 6]. The first approach has the advantage that each point (or frame) reconstruction problem is decoupled from the other problems, thereby solving much smaller systems but having a slow convergence rate. The second approach needs to solve a minimization problem over a large parameter space. However, the computation efficiency can be considerably improved by observing that 3D structure estimation is independent for each point, given a motion estimation [15, 6, 16]. This is the technique we adopt here.

3.1 Estimating structure

In this paper, we adopt the interleaving approach. To project the ith 3-D point $\mathbf{x}_i$ into the kth frame at location $\mathbf{u}_{ik}, i = 1, ..., m, k = 1, ..., n$, we write

$$\mathbf{u}_{ik} \sim \mathbf{V}_k\mathbf{R}_k(\mathbf{x}_i - \mathbf{t}_k), \quad (1)$$

where $\sim$ indicates equality up to a scale, $\mathbf{R}_k$ is the rotation matrix for camera k, $\mathbf{t}_k$ is the location of its optical center, and $\mathbf{V}_k$ is its camera internal matrix (usually assumed to be upper triangular or some simpler form, e.g., diagonal). The location of a 3D point corresponding to an observed image feature is

$$\mathbf{x}_i = w_{ik}\mathbf{R}_k^{-1}\mathbf{V}_k^{-1}\mathbf{u}_{ik} + \mathbf{t}_k, \quad (2)$$

where w_{ik} is an unknown scale factor.

To reconstruct a 3D point location, we minimize

$$\sum_{k=1}^{n}\left[\left(u_{ik} - \frac{\mathbf{p}_{k0}^T\tilde{\mathbf{x}}_i}{\mathbf{p}_{k2}^T\tilde{\mathbf{x}}_i}\right)^2 + \left(v_{ik} - \frac{\mathbf{p}_{k1}^T\tilde{\mathbf{x}}_i}{\mathbf{p}_{k2}^T\tilde{\mathbf{x}}_i}\right)^2\right] \quad (3)$$

where $\mathbf{p}_{kr}$ are the three rows of the *camera or projection matrix*

$$\mathbf{P}_k = \mathbf{V}_k[\mathbf{R}_k| - \mathbf{t}_k]$$

and $\tilde{\mathbf{x}}_i = [\mathbf{x}_i|1]$, i.e., the homogeneous representation of $\mathbf{x}_i$. As pointed out by [16], this is equivalent to solving the following overconstrained linear system,

$$\begin{aligned} D_{ik}^{-1}(\mathbf{p}_{k0} - u_{ik}\mathbf{p}_{k2})^T\tilde{\mathbf{x}}_i &= 0 \\ D_{ik}^{-1}(\mathbf{p}_{k1} - v_{ik}\mathbf{p}_{k2})^T\tilde{\mathbf{x}}_i &= 0, \end{aligned} \quad (4)$$

where the weights are given by $D_{ik} = \mathbf{p}_{k2}^T\tilde{\mathbf{x}}_i$ (these are set to $D_{ik} = 1$ in the first iteration).

Because the structure of each point can be estimated independently given the motion parameters, we need only to solve m linear systems of size $2n \times 3$.

3.2 Estimating motion

Unfortunately, estimation of camera motion parameters is not independent. We must conduct a minimization problem over $6n$-D space[1] with $2nm$ constraints, i.e.,

$$\min_{\{\mathbf{m}_k\}}\sum_{i=1}^{m}\min_{\mathbf{x}_i}\sum_{k=1}^{n}\left[\left(u_{ik} - \frac{\mathbf{p}_{k0}^T\tilde{\mathbf{x}}_i}{\mathbf{p}_{k2}^T\tilde{\mathbf{x}}_i}\right)^2 + \left(v_{ik} - \frac{\mathbf{p}_{k1}^T\tilde{\mathbf{x}}_i}{\mathbf{p}_{k2}^T\tilde{\mathbf{x}}_i}\right)^2\right] \quad (5)$$

At each iteration for motion parameters, we perform the optimization of the structure parameters, as described in the last subsection.

To update $\mathbf{P}_k$, we apply a linearized least squares to reconstruct Euclidean motion parameters ($\mathbf{V}_k[\mathbf{R}_k| - \mathbf{t}_k]$). Let us assume the following updates

$$\mathbf{R}_k \leftarrow \mathbf{R}_k(\mathbf{I} + [\omega_k]_\times), \quad \mathbf{t}_k \leftarrow \mathbf{t}_k + \delta\mathbf{t}_k. \quad (6)$$

The rotation estimate can then be updated using Rodriguez's formula [1].

3.3 Complexity of bundle adjustment

Lemma 1 *The complexity of interleaving bundle adjustment at each iteration step is $O(mn^3)$ where m is the number of feature points, and n is the number of frames.*

For the SFM problem with m feature points, and n frames, there are $N = 3m + 6(n-1) - 1$ unknown parameters and $M = 2nm$ constraints because each feature point provides two constraints in every frame (see Equations 4). To solve for 3D structure, we have to solve m linear least

[1] It is $6(n-1) - 1$ to be exact.

squares problems of size $2n \times 3$; to solve for camera motion, we solve a linear system size $2nm \times (6n-7)$.

The complexity for solving a least squares problem $M \times N (M > N)$ using LU decomposition is $O(MN^2 - N^3/3)$ (p.102 of [5]). Therefore, the complexity of interleaving bundle adjustment is $O(mn^3)$.[2]

3.4 Efficient bundle adjustment

Therefore, to speed up bundle adjustment, we should either reduce the number of features or the number of frames. In particular, reducing the number of frames in a long sequence will result in significant speedup. If we reduce the number of frames from n to n', the complexity changes from $O(mn^3)$ to $O(mn'^3)$. For example, if we reduce the number of frames from 500 to 100 (e.g., using 2 virtual frames in each of 50 segments), we will have more than 100-fold speedup.

4 Virtual key frames

How can we reduce the number of frames? Simply subsampling each segment is not a good idea because it throws away useful information.

After performing bundle adjustment for each segment as described in Section 3, we obtain an optimal estimate of camera motion and structure. Furthermore, we can compute their uncertainty, in terms of covariance matrices under first order approximation, as shown in Appendix A. Let $\mathbf{m}_k = [\omega_k^T, \mathbf{t}_k^T]^T$ be the motion of the k-th camera and $\mathbf{\Lambda}_{\mathbf{m}_k}$, its covariance matrix. Let $\mathbf{x}_i$ be the i-th reconstructed 3D point and $\mathbf{\Lambda}_{\mathbf{x}_i}$ be its covariance matrix. Under first order approximation, those quantities capture all the information contained in each segment. When we deal with the whole sequence, why not make use of this concise information already available? This leads to the idea of *virtual key frames*, described below.

4.1 Representation of virtual key frames

We use two virtual frames to represent each segment because at least two frames are needed for 3D reconstruction. The position of a virtual frame can coincide with one real frame, say k. A virtual frame contains the projection of 3D reconstructed points, denoted by $\bar{\mathbf{u}}_{ik} = [\bar{u}_{ik}, \bar{v}_{ik}]$, and more importantly its covariance matrix $\mathbf{\Lambda}_{\bar{\mathbf{u}}_{ik}}$. They are given by

$$\bar{\mathbf{u}}_{ik} = \frac{1}{\mathbf{p}_{k2}^T \tilde{\mathbf{x}}_i} \begin{bmatrix} \mathbf{p}_{k0}^T \tilde{\mathbf{x}}_i \\ \mathbf{p}_{k1}^T \tilde{\mathbf{x}}_i \end{bmatrix} \equiv \mathbf{f}(\mathbf{m}_k, \mathbf{x}_i) \quad (7)$$

$$\mathbf{\Lambda}_{\mathbf{u}_{ik}} = \frac{\partial \mathbf{f}}{\partial \mathbf{m}} \mathbf{\Lambda}_{\mathbf{m}_k} \frac{\partial \mathbf{f}}{\partial \mathbf{m}}^T + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{\Lambda}_{\mathbf{x}_i} \frac{\partial \mathbf{f}}{\partial \mathbf{x}}^T \quad (8)$$

where the derivatives $\partial \mathbf{f}/\partial \mathbf{m}$ and $\partial \mathbf{f}/\partial \mathbf{x}$, are evaluated at $\mathbf{m}_k$ and $\mathbf{x}_i$. In computing $\mathbf{\Lambda}_{\bar{\mathbf{u}}_{ik}}$ as shown above, we have assumed that the uncertainty in $\mathbf{m}_k$ and $\mathbf{x}_i$ is independent, and indeed we have experimentally found that their correlation is negligible. The reason is that the motion is computed from many data points.

[2] On the other hand, simultaneous bundle adjustment (solving structure and motion all together) has the complexity of $O(mn(m+2n)^2)$.

4.2 Bundle adjust with virtual frames

Bundle adjustment as formulated in Section 3 assumes that all image points are corrupted by independent and identically distributed noise. When we incorporate virtual frames together with real ones, we must reformulate the problem in order to account for different noise distribution. Maximum likelihood estimation is equivalent to minimizing the following functional:

$$\sum_k \sum_i (\bar{\mathbf{u}}_{ik} - \hat{\mathbf{u}}_{ik})^T \mathbf{\Lambda}_{\bar{\mathbf{u}}_{ik}}^{-1} (\bar{\mathbf{u}}_{ik} - \hat{\mathbf{u}}_{ik}) , \quad (9)$$

where $\bar{\mathbf{u}}_{ik}$ and $\mathbf{\Lambda}_{\bar{\mathbf{u}}_{ik}}$ are given by Equations 8 and 9 if they are from virtual frames, or simply the detected image points if they are from real images; and $\hat{\mathbf{u}}_{ik}$ is the projection of the estimated 3D point $\mathbf{x}_i$ at image k.

This is simply the weighted version of the standard bundle adjustment. The technique presented in Section 3, after some simple modifications, can be used to solve this problem.

5 Tracking

In our multi-frame SFM system, we select good features based on minimum eigenvalues of Hessian matrix of motion estimation [10]. Moreover, we interactively select regions of interest and number of good features if necessary, track features with respect to the base frame, instead of the previous frame, and divide the sequence into segments based on the quality of the epipolar constraint.

Because automatic feature selection may not always select all the features that we are interested in the images, we design our feature tracker as an interactive system. We first select regions that we want to generate features, and specify the number of features. Features are automatically generated in these regions. Additional features can be added at any frame. In this way, we obtain more uniformly distributed feature points which are important for accurate 3D reconstruction.

We always track features with respect to the base frame because we want to keep feature points from drifting away. Drifting is more likely to happen if we track features with respect to the previous frame because of error accumulation. The purpose of tracking here is to have very accurate correspondence that is crucial for 3D reconstruction.

In general, tracking error increases with the length of the tracking sequence. On the other hand, the shorter the tracking sequence is, the more uncertain the estimated epipolar geometry (therefore, 3D reconstruction) would be. We

(a)

(b)

Figure 2: Basketball sequence: (a) input images; (b) two views of 3D reconstruction.

need to decide for how long we should keep tracking the features. A good indication is the number of trackable feature points given a threshold of tracking error. Any frame in a segment should have at least 60%[3] of the trackable features identified in the base frame. A better indication is the percentage of the trackable features that satisfy the epipolar constraint. Specifically, the distance between the features and the epipolar lines (computed through fundamental matrix) shows how much we can trust the tracking result.

6 Experimental results

We have used our SFM system to reconstruct 3D models for several long sequences. The first example is a basketball sequence shown in Figure 2. As expected for the fixation type of sequence, structure from motion works well. Two views of the basketball model show good reconstruction in Figure 2. No camera calibration was used except a rough estimation of focal length was made. Better 3D reconstruction can be expected if focal length is calibrated using methods such as [9]. The second example is the MPEG garden sequence with large camera translation. Two different views of the reconstructed garden model are shown in Figure 3. Note that three different parts of the garden scene are reconstructed well: tree in front, flower bed in the middle, and house at the back.

[3]This number is ad-hoc. It depends on tracking error threshold and distribution of features.

(a)

(b)

Figure 3: Garden sequence: (a) input images; (b) two views of 3D reconstruction.

7 Discussion

In this paper we have presented a preliminary study on an efficient hierarchical approach to structure from motion for long image sequences. To efficiently bundle adjust 3D structures from all segments, we reduce the number of frames in each segment by introducing *virtual key frames*. The virtual frames encode the 3D structure of each segment along with its uncertainty but are a small subset of the original frames. It has been shown from complexity analysis of bundle adjustment that our method achieves significant speedup compared to the conventional bundle adjustment for a typical dense long sequence.

We are currently working on optimally segmenting image sequences by comparing optical flow and epipolar geometry constraints. The epipolar constraint, if estimated reliably, could be used to further constrain the feature tracking as well, much like the process of stereo matching. The difficulty is how we combine those two different kinds of constraints.

We also plan to have quantative study with synthetic data to verify if the proposed algorithm is indeed accurate and robust, The topics under study include: Does the new algorithm converge to the same result of bundle adjusting all frames? How does the reprojection error change for each frame at each stage of the algorithm? How does the new algorithm compare with the original bundle adjustment, and with simple subsampling of frames, in terms of reconstruction error?

8 Acknowledgements

The authors thank Rick Szeliski for his help on implementing two-frame SFM and multi-frame SFM, and for his many technical discussions.

A Parameter uncertainty

Briefly, we describe how to compute uncertainty of the parameters obtained from nonlinear least squares. More details can be found in [16].

Let $\hat{\mathbf{y}}$ be the parameter vector which minimizes $C(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is the data vector. Its covariance matrix is then given by

$$\mathbf{\Lambda_y} = \mathbf{H}^{-1} \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}} \mathbf{\Lambda_x} \left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}} \right)^T \mathbf{H}^{-T} \qquad (10)$$

where $\mathbf{\Phi} = \partial C(\mathbf{x}, \hat{\mathbf{y}}) / \partial \mathbf{y}$ and $\mathbf{H} = \partial \mathbf{\Phi} / \partial \mathbf{y}$.

If $C(\mathbf{x}, \mathbf{y})$ is in the form of sum of squares, i.e., $\sum_{i=1}^{n} C_i^2(\mathbf{x}, \mathbf{y})$, and if $C_i(\mathbf{x}, \mathbf{y})$ can be considered as zero mean and independent identical distributed errors, then we have the following simple form

$$\mathbf{\Lambda_y} = \frac{2C(\mathbf{x}, \hat{\mathbf{y}})}{n - p} \mathbf{H}^{-1} \qquad (11)$$

where p is the parameter vector dimension, and $\mathbf{H}$ can be approximated under first order approximation as

$$\mathbf{H} = 2 \sum_i \left(\frac{\partial C(\mathbf{x}, \hat{\mathbf{y}})}{\partial \mathbf{y}} \right)^T \frac{\partial C(\mathbf{x}, \hat{\mathbf{y}})}{\partial \mathbf{y}} .$$

In Eq.(8), $\mathbf{\Lambda_{m_k}}$ is computed using the above simple form Eq.(11) because motion parameters are estimated from many points (e.g., hundreds), while $\mathbf{\Lambda_{x_i}}$ is computed using the general form Eq. (10).

References

[1] N. Ayache. *Vision Stereoscopique et Perception Multisensorielle.* InterEditions., Paris, 1989.

[2] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Fourth European Conference on Computer Vision (ECCV'96)*, volume 2, pages 683–695, Cambridge, England, April 1996. Springer-Verlag.

[3] O. Faugeras. *Three-dimensional computer vision: A geometric viewpoint.* MIT Press, Cambridge, Massachusetts, 1993.

[4] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed and open image sequences. In *Fifth European Conference on Computer Vision (ECCV'98)*, pages 311–326, Freiburg, Germany, June 1998. Springer-Verlag.

[5] G. Golub and C. F. Van Loan. *Matrix Computation, third edition.* The John Hopkins University Press, Baltimore and London, 1996.

[6] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In *Second European Workshop on Invariants*, pages 187–202, Ponta Delgada, Azores, October 1993. Springer-Verlag.

[7] P. F. McLauchlan, I. D. Reid, and D. W. Murray. Recursive affine structure and motion from image sequences. In *Third European Conference on Computer Vision (ECCV'94)*, volume 1, pages 217–224, Stockholm, Sweden, May 1994. Springer-Verlag.

[8] F. H. Moffitt and E. M. Mikhail. *Photogrammetry.* Harper & Row, New York, 3 edition, 1980.

[9] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 90–95, Bombay, January 1998.

[10] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, Washington, June 1994. IEEE Computer Society.

[11] P. Sturm and W. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Fourth European Conference on Computer Vision (ECCV'96)*, volume 1, pages 709–720, Cambridge, England, April 1996. Springer-Verlag.

[12] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, March 1994.

[13] R. Szeliski and P. Torr. Geometrically constrained structure from motion: Points on planes. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE)*, pages 171–186, Freiburg, Germany, June 1998.

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

[15] J. Weng, N. Ahuja, and T. S. Huang. Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):864–884, September 1993.

[16] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998.