Batch Recovery of Multiple Views with Missing Data Using Direct Sparse Solvers

Nicolas Guilbert[*]
Lund University
Sweden
nicolas@maths.lth.se

Adrien Bartoli[*]
INRIA Rhône-Alpes
France
adrien.bartoli@inria.fr

Abstract

Using the so-called closure constraints, it is possible to estimate the projection matrices of the cameras observing a static scene given correspondences between multiple views. We present a batch algorithm for recovering all the cameras based on the closure constraints. The approach is motivated by the necessity of including as much information as possible in the initial recovery of the motion, as is done in factorisation schemes. The main advantage of the proposed method over factorisation is that it naturally deals with missing data. Compared to other algorithms, the method is very fast and flexible in terms of the selection of input data.

1 Introduction

The problem of recovering the motion of a camera given only image feature correspondences has been extensively studied, see e.g. [5, 6, 7-17]. Most techniques are based on an initialisation phase followed by a Maximum Likelihood refinement, namely bundle adjustment [18]. Existing initialisation algorithms can be divided into two families, namely *batch* and *sequential* processes. The advantage of batch initialisation is that as much of the available information as possible is included as early as possible in the initial reconstruction step. This is not the case in sequential processing, where the global information is 'patched on' at a later stage, typically using bundle adjustment. Both families have important drawbacks, in particular the majority of existing batch algorithms require all features to be present in all images, and sequential algorithms may suffer from severe accumulation of the errors.

The method presented in this paper concerns a batch algorithm where the presence of all features in all images is *not* required. The basic idea is to compute matching tensors between the images (fundamental matrices, trifocal or quadrifocal tensors) and finally determine *all* of the camera matrices in a single computational step.

The prototypical batch algorithm for recovering structure and motion is the Tomasi-Kanade factorisation scheme in [14]. It recovers a Maximum Likelihood Estimate (MLE) [11], but it is restricted to the affine camera model. It has been extended to full perspective [13], without however maintaining the ML-property. As with all factorisation schemes,

[*] This work was accomplished within the EC-funded Visire project, IST-1999-10756.

it has difficulties handling missing data. The methods proposed in [8, 10] extend classical factorisation to handle missing data and in [1] absence of a given image feature may be accounted for by setting its associated covariance to infinity. However, the computational complexity of the abovementioned methods increases drastically with the amount of missing data. Specifically, they all *at best* involve a singular value decomposition of a $2m \times n$ matrix, m being the number of cameras and n the total number of observed 3D points. Other batch multiview reconstruction schemes include reconstruction from matching tensors [6] and these are thus limited to less than 4 views.

The notion of *closure constraints* was introduced by Triggs in [17], denoting bilinear constraints between camera parameters and matching tensors. We investigate these constraints in §3. There, closure constraints are used to construct a constraint matrix which is eventually solved for yielding the camera parameters. In [9] closure constraints for the affine camera model are derived.

This paper makes the following contributions: Firstly, we derive a simple formulation for closure constraints for the affine camera and show how a simple linear system can be built in §4. Secondly, we analyse the structure of the central design matrix for three practical families of configurations in §5. These are *(i)* the minimal cases, where the number of constraints yields an exact solution, *(ii)* the continuous motion cases where correspondences only exist among a limited number of neighbouring views, and *(iii)* the general unstructured cases which are characterised by a sparse structure of the design matrix. Thirdly, we compare our algorithm with existing ones based on simulated and real data in §6 and conclude the paper in §7.

2 Notation and Background

We denote 2D homogeneous image points by a subscripted $\mathbf{x}$, 3D homogeneous points by $\mathbf{X} = [\bar{\mathbf{X}} \quad 1]^T$ and camera matrices by $\mathbf{P}$ with

$$\lambda_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j \quad i = 1\ldots m,\ j = 1\ldots n \tag{1}$$

where m and n denote the number of views and 3D object points respectively, and λ is the *projective depth* of the given point.

The fundamental matrix $\mathbf{F}_{12}$ encapsulates the geometrical relationship between two views (for simplicity of notation we consider view 1 and 2). It constrains the positions of corresponding image points $\mathbf{x}_{1j}$ and $\mathbf{x}_{2j}$ through the relation

$$\mathbf{x}_{1j}^T\mathbf{F}_{12}\mathbf{x}_{2j} = 0, \quad j = 1\ldots n \tag{2}$$

The line defined by $\mathbf{F}_{12}\mathbf{x}_{2j}$ is the epipolar line of $\mathbf{x}_{2j}$ in image 1, i.e. the line joining the epipole $\mathbf{e}_{12}$ and $\mathbf{x}_{1j}$. For the affine camera model, the projection matrices have the form

$$\mathbf{P}_i = \begin{bmatrix} \bar{\mathbf{P}}_i & \mathbf{t}_i \\ \mathbf{0}_3^T & 1 \end{bmatrix} \tag{3}$$

The affine fundamental matrix has five non-zero entries defined up to scale, i.e. four degrees of freedom:

$$\mathbf{F}_{12} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix} \tag{4}$$

Computing the fundamental matrix thus requires at least four point correspondences, however as many points as possible should be included to minimise the effect of noise. A closed-form solution for the Maximum Likelihood Estimate of the affine fundamental matrix exists [6].

3 The Closure Constraints

In [17], Triggs introduces the so-called $\mathbf{F}-\mathbf{e}$ closure constraint, i.e. $\mathbf{F}_{12}\mathbf{P}_2+[\mathbf{e}_{21}]_\times\mathbf{P}_1=\mathbf{0}$ and the $\mathbf{e}-\mathbf{G}-\mathbf{e}$ closure. We will here give a simple derivation of an alternative closure constraint, the $\mathbf{F}$-closure: Let $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the projections of $\mathbf{X}\in\mathbb{P}^3$ onto two images. Combining equations (2) and (1) we obtain:

$$\mathbf{X}^T \underbrace{\mathbf{P}_1^T\mathbf{F}_{12}\mathbf{P}_2}_{\mathscr{S}}\mathbf{X}=0, \quad \forall\mathbf{X}\in\mathbb{P}^3 \tag{5}$$

which is a quadratic form. Consequently $\mathscr{S}$ is skew symmetric, i.e. $\mathscr{S}^T=-\mathscr{S}$.

4 A Simple Formulation of the Affine Closure Constraints

In the affine case, the structure of $\mathscr{S}$ becomes particularly simple, which stems from the structure of the camera matrices (3) and the affine fundamental matrix (4):

$$\mathscr{S}=\mathbf{P}_2^T\mathbf{F}_{21}\mathbf{P}_1=\begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{a}\\ -\mathbf{a}^T & 0\end{bmatrix} \tag{6}$$

The upper left $\mathbf{0}_{3\times3}$ matrix is the result of the corresponding bilinear term containing either a zero-coefficient from the fundamental matrix or one of the corresponding camera entries being zero. The rest of the structure of $\mathscr{S}$ is a consequence of the skew-symmetry, and *concerns only linear terms* in the entries of the camera matrices since these terms include the lower right 1 of either $\mathbf{P}_1$ or $\mathbf{P}_2$. By rearranging the equations in (6) these turn into four linear constraints on the coefficients of $\mathbf{P}_1$ and $\mathbf{P}_2$:

$$[a \quad b \quad c \quad d]\begin{bmatrix}\bar{\mathbf{P}}_1 & \mathbf{t}_1\\ \bar{\mathbf{P}}_2 & \mathbf{t}_2\end{bmatrix}=\underbrace{[\,\mathbf{0}_3 \quad -e\,]}_{\mathbf{r}_{12}} \tag{7}$$

These constraints apply for each pair of views $\{\mathbf{P}_{i_1},\mathbf{P}_{i_2}\}$, $i_1\neq i_2$, provided $\mathbf{F}_{i_1i_2}$ is defined. Affine trifocal or quadrifocal tensors could be used as well, for instance by extracting fundamental matrices, or along the lines of [9]. We construct a linear system of equations using (7) as the building block, with the form $\mathbf{S}\mathscr{P}=\mathscr{R}$:

$$\begin{bmatrix}\mathbf{s}_{12}\\ \mathbf{s}_{1i_1}\\ \vdots\\ \mathbf{s}_{i_ki_m}\end{bmatrix}\begin{bmatrix}\bar{\mathbf{P}}_1 & \mathbf{t}_1\\ \bar{\mathbf{P}}_2 & \mathbf{t}_2\\ \vdots & \vdots\\ \vdots & \vdots\\ \vdots & \vdots\\ \bar{\mathbf{P}}_m & \mathbf{t}_m\end{bmatrix}=\begin{bmatrix}\mathbf{r}_{12}\\ \mathbf{r}_{1i_1}\\ \vdots\\ \mathbf{r}_{i_ki_m}\end{bmatrix} \tag{8}$$

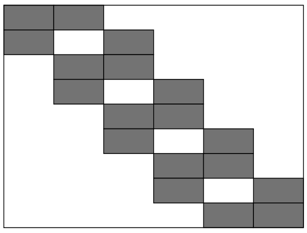


Figure 1: **Structure of the central design matrix** constraining the camera matrices. The shown structure is for a minimal configuration, i.e. $2m-3$ block-row entries for m views, before fixing the 12-parameter gauge freedom (See §5.1 for more details).

where $\mathbf{r}_{i_1 i_2}$ is the right hand side in equation (7) and $\mathbf{s}_{i_1 i_2}$ are $1 \times 2m$ row vectors

$$\mathbf{s}_{i_1 i_2} = [\ldots \quad \underbrace{a \quad b}_{\text{First block}} \quad \ldots \quad \underbrace{c \quad d}_{\text{Second block}} \quad \ldots]$$

dots indicating an adequate number of zeros. One possible structure for **S** is shown in Figure 1. Depending on the configuration of the views, in particular whether local degeneracies are present or not, and depending on other parameters such as efficiency requirements and the presence of correspondences between frames, we choose one of several approaches to solve the system.

5 Classifying and Solving Practical Cases

Below we consider three practical cases: the first concerns minimal configurations, and is appropriate when little overlap exists among the views. One scenario in which the minimal case is particularly useful is when doing a RANSAC [15] on the fundamental matrices in order to discard erroneous ones. The second case concerns sequences from continuous camera trajectories, i.e. where correspondences exist between a limited number of *neighbouring* views only. In the third case we consider general configurations with no prior structure in the overlap pattern between views, except for overall sparsity.

5.1 Minimal Configurations

In minimal configurations, the system in (8) has an exact unique solution , i.e. the matrix on the left hand side must be square and have full rank. In consequence, the gauge freedom related to the overall 12-parameter affine indeterminacy has to be eliminated. In practice, this is done by selecting two cameras for which the fundamental matrix is well defined [16], subsequently fixing 8 parameters in the first one and 4 parameters in the second. Also, since each pair of views eliminates 4 degrees of freedom, we need $(8m-12)/4 = 2m-3$ pairs to account for the remaining free parameters of the m views. One possibility is to include the relations between a given camera and the two following ones, except for the two last ones. This yields the square banded non-symmetric matrix in Figure 1 which has full rank, provided that none of the camera configurations are degenerate.

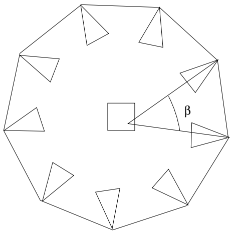


Figure 2: **The experimental setup**: *m* affine cameras pointed inwards on a circular path, observing a cubic cloud of *n* uniformly distributed points. The angle β indicates the angle between the focal axis of two neighbouring cameras and is used as a measure of the baseline

.

5.2 Continuous Motion Configurations

By continuous motion (cm) we understand a type of sequence where sufficient overlap exists between a certain number of *neighbouring* frames $n_{cm} > 3$ for fundamental matrices to be established among them (the case $n_{cm} = 3$ being a minimal configuration, see §5.1). The system is in general overdetermined. We consider the associated normal equations

$$\mathbf{S}^T\mathbf{S}\mathscr{P} = \mathbf{S}^T\mathscr{R} \tag{9}$$

where $\mathbf{S}^T\mathbf{S}$ is a band matrix with twice the bandwith as $\mathbf{S}$. The system is solved using band or profile Cholesky factorisation/back substitution, see e.g. [3] for details. The execution time of the band Cholesky factorisation is known to be $O(\text{band width} \times \text{matrix size})$, thus the execution time for the continuous motion configuration is $O(n_{cm}^2 m)$, n_{cm} being proportional to the band width and $n_{cm}m$ proportional to the matrix size. The continuous motion configuration may be extended to the case where a few matching relations exist among non-neighbouring views, in which case $\mathbf{S}^T\mathbf{S}$ may be rearranged into an *arrowhead* structure [18]. One such common case is when a sequence is closed, i.e. previously observed 3D points reappear at some point, typically the end of the sequence.

5.3 Unstructured Configurations

If each view is connected to at least two, and on average many fewer than *m*, the matrix **S** is sparse but unstructured and we solve the system using a supernodal approach [4]. Note that we could also use sparse QR decomposition and hence avoid having to form the normal equations, see e.g. [7] for an overview and further references.

6 Experiments

6.1 Synthetic Data

Experiments were performed on synthetic data using the setup shown in Figure 2. *m* cameras are laid out on a circular path, all pointing inwards. The observed object consists of *n* 3D-points uniformly distributed within a cube.

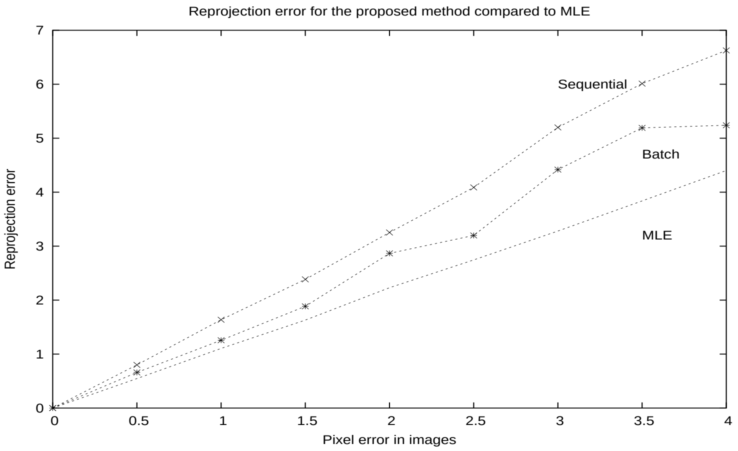


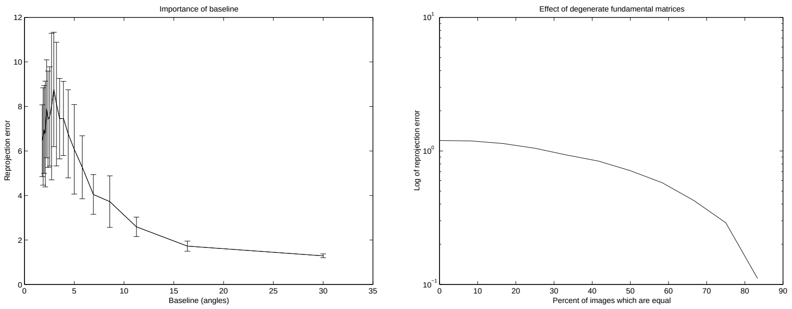
Figure 3: **Comparison** between the maximum likelihood estimate of the cameras and structure, the proposed batch method and a sequential scheme where an initial structure is computed using 6 cameras, whereupon the remaining cameras are obtained by resectioning. The graph shows the reprojection error as a function of the (gaussian) image noise. The proposed batch algorithm performs better than the resection/intersection approach and is close to the Maximum Likelihood Estimate

6.1.1 Comparison to Existing Methods

The proposed algorithm is compared to two methods: The first is the factorisation algorithm given in [14], a Maximum Likelihood Estimator for affine motion and structure. The second is a widely used reconstruction algorithm where an initial set of cameras and visible 3D points are computed from the first frames in the sequence, followed by a series of resections and intersections in order to recover the remainder of the motion and structure [2]. In the experiment, $m = 12$ cameras and $n = 100$ 3D-points were used, with gaussian noise ($\sigma = 1$ pixel) in the images, all 3D-points being visible in all views. As it can be seen from Figure 3, the presented batch algorithm performs better than the resection/intersection-approach and is close to the Maximum Likelihood Estimator.

6.1.2 The Influence of Various Parameters

Width of the camera baseline: In order to assess the importance of the width of the baseline, the experiment shown in Figure 4a was performed. Again, $m = 12$ cameras were used, positioned on the arc of the circle of Figure 2 and the angle of incidence of their focal axis β varied from 2° to 30°, i.e. the overall baseline varied from 24° to 360°. The noise in the image is white gaussian with $\sigma = 1$ pixel. Fundamental matrices were computed among each view and 10 of its neighbours. The reprojection error is seen to peak for $\beta \approx 7^\circ$. For lower values of β, the constraints imposed on the structure by the cameras are so loose that they are easily satisfied. For higher values, the computation of the fundamental matrices is getting well-conditioned, thus yielding lower reprojection errors.



(a) (b)

Figure 4: **The influence of various parameters**: **(a)** The effect of enlarging the baseline. The figure shows the decrease in error as the baseline gets larger (from $\beta = 2^\circ$ to $\beta = 30^\circ$). $m = 12$ views were used and $n = 150$ points, each camera was related to 10 neighbours using fundamental matrices. Image noise $\sigma = 1$ pixel. **(b)** The effect of degenerate camera matrices in the sequence. The graph shows the reprojection error as a function of the percentage of views which are equal in the sequence ($m = 12$, $n = 150$, $\beta = 18^\circ$, $\sigma = 1$ pixel).

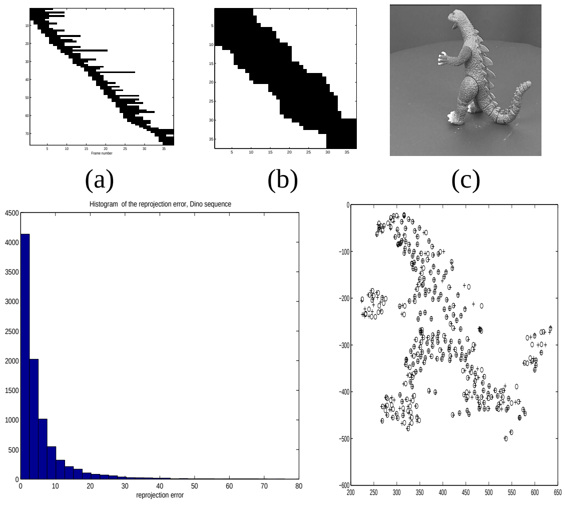
Sensitivity to degenerate matching tensors: In this experiment ($m = 12$, $n = 20$, and $\beta = 18^\circ$, image noise $\sigma = 1$ pixel) the sensitivity of the algorithm with respect to the number of deficient fundamental matrices is investigated. This is done by successively setting the images $2 \ldots m-1$ equal to the m'th,thus ensuring degenerate fundamental matrices among them. The result is shown in Figure 4b, where the reprojection error is plotted against the percentage of equal views in the sequence. As the number of equal views increase, i.e. the degeneracies become more numerous, the reprojection error is actually seen to decrease. When all the views are the same, the configuration is globally degenerated and the reprojection error is meaningless. Again, the more equal views in the sequence, the looser the constraints on the structure, eventually leading to a lower reprojection error.

6.2 Experiments on Real Data

The experiments on real data were performed on two datasets, the Hotel sequence from Carnegie Mellon University and the Dinosaur sequence from the University of Hannover.

6.2.1 Dinosaur Sequence

The sequence consists of 37 images with perspective effects, of which the first and the last are known to coincide. A total of 1888 3D-points were tracked across the sequence using the KLT-tracker [12]. In Figure 5a, the presence of the projections of the 3D points in each frame is shown. (b) shows whether a given pair of frames has sufficient correspondence overlap for its fundamental matrix to be computed, (we will call it the view interaction pattern) and (d) the histogram of all the reprojection errors. (c) and (e) show the 23rd frame in the sequence, respectively the original image and the measured/reprojected points. The



(a) (b) (c)

(d) (e)

Figure 5: **Experimental results on the Dinosaur sequence**. **(a)** feature presence in each frame. **(b)** view interaction pattern, cf. §6.2.1. **(c)** frame 23 from the sequence. **(d)** histogram of the reprojection error. **(e)** tracked and reprojected features. The mean reprojection error is 5.4 pixels.

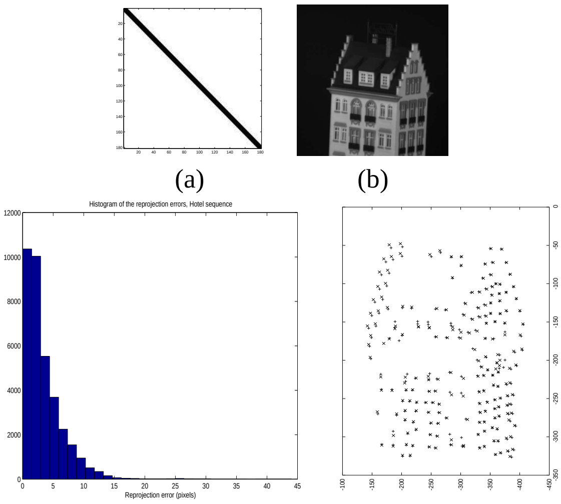
average reprojection error is 5.4 pixels and the system (7) was solved in 2.1 milliseconds on a Pentium IV@1.8 GHz.

6.2.2 Hotel Sequence

The sequence consists of 181 images obtained with a near-affine camera observing 197 3D points which are all visible in all the frames. In Figure 6a the view interaction pattern is shown. Although all the points are visible in all the views, a quasi-minimal banded configuration (with bandwith 4) is chosen in order to illustrate the good performance even with limited interactions between views. (c) shows the histogram of all the reprojection errors. (b) and (d) show the 95th frame in the sequence, respectively the original image and the tracked/reprojected points. The average reprojection error is 3.5 pixels and the execution time 23 milliseconds.

7 Conclusion

In this paper we presented a batch algorithm for recovering the camera matrices in the structure from motion problem. The method is based on linear affine closure constraints, and recovers all the camera matrices by solving a single, possibly overdetermined, linear system. Depending on the correspondence pattern in the sequence, different solvers are applied. Experimentally, the accuracy of the method was shown to lie close to that of the Maximum Likelihood Estimator as long as sufficient baseline is provided. The recovery of nearly two hundred camera matrices was performed in a few hundreds of a second.



(a) (b)

(c) (d)

Figure 6: **Experimental results on the Hotel sequence**. **(a)** view interaction pattern, cf. §6.2.1. **(b)** frame 95 in the sequence **(c)** histogram of the reprojection error. **(d)** measured and reprojected features. The mean reprojection error is 3.5 pixels.

Acknowledgements

We wish to thank the Carnegie Mellon University and the University of Hannover for sharing their data. We especially wish to thank Bill Triggs for valuable help and advice.

References

[1] H. Aanaes, R. Fisker, K. Astrom, and J.M. Carstensen. Robust factorization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(9):1215–1225, 2002.

[2] P. Beardsley, A. Zisserman, and D. W. Murray. Sequential updating of projective and affine structure from motion. *Int. Journal of Computer Vision*, 23(3):235–259, 1997.

[3] Å. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996.

[4] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3):720–755, 1999.

[5] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, Mass, 1993.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] E. J. Haunschmid and C.W.Ueberhuber. Direct solvers for sparse systems. Technical report, Institute of Applied and Numerical Mathematics, 1999.

[8] D. Jacobs. Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 206–212, 1997.

[9] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *Int. Journal of Computer Vision,* 1999.

[10] D. Martinec and T. Pajdla. Structure from many perspective images with occclusions. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, Denmark,* 2002.

[11] I. J. Reid and D.W. Murray. Active tracking of foveated feature clusrters using affine structure. *Int. Journal of Computer Vision,* 18:41–60, 1996.

[12] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94),* Seattle, June 1994.

[13] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV'96*, volume 1065 of *Lecture Notes in Computer Science*, pages 709–720. Springer Verlag, Lecture Notes in Computer Science, 1996.

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

[15] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(8):591–605, 1997.

[16] P.H.S. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding,* 71(3):312–333, September 1998.

[17] B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing,* 15(8):617–625, 1997.

[18] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS. Springer Verlag, 2000.