Recursive Affine Structure and Motion from Image Sequences *

Philip F. McLauchlan, Ian D. Reid and David W. Murray

Department of Engineering Science,
University of Oxford,
Parks Road, Oxford, OX1 3PJ, UK.
Email pm|ian|dwm@robots.oxford.ac.uk

Abstract. This paper presents a new algorithm for structure from motion from an arbitrary number of tracked features over an arbitrary number of images, which possesses several advantages over previous formulations. First, it is recursive, so the time complexity is independent of the number of images. The complexity is linear with the number of tracked features. The algorithm allows newly appeared features to be included, stale features to be discarded, and missing data to be handled naturally. Dynamic outlier elimination is achieved without recourse to heuristic segmentation strategies. Lastly, the algorithm can employ different kinds of tracked features, e.g. edges and corners, in the same framework.
The actual structure from motion recovered is affine, which assumes limited depth variation within the field of view, but the recovery is based on a more general recursive estimation algorithm, known as the variable state dimension filter (VSDF), which we devised and applied earlier to active camera calibration.
Results are presented for real image sequences, and timings for the algorithm demonstrate the feasibility for real-time implementation.

Keywords: Structure from motion, affine invariance, recursive filter.

1 Introduction

Recent developments in the computation of structure from motion have demonstrated the clear advantages of, first, considering structure in an *object centred* reference frame rather than as a set of depths [4, 12, 2]; and, secondly, using extended sequences of closely-spaced views of an object in order to alleviate correspondence problems but at the same time to provide a large baseline for the structure computation [16, 15]. This paper draws on these two ideas, on our previous work on active fixation in dynamic scenes [10, 13, 1], and on our development of a recursive filter with variable state-dimension, to devise a system for automated acquisition of three-dimensional models of objects which are tracked over extended periods.

The work described here differs from other recent work in the same area in a number of fundamental ways:

* This work was supported by SERC Grants GR/G30003 and GR/J65372. The authors are grateful to Larry Shapiro and Paul Beardsley for discussions.

Lecture Notes in Computer Science, Vol. 800
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

- the algorithm is *recursive*, meaning that the solution for frame $k+1$ is determined from the solution at frame k and the data at frame $k+1$, with no overhead as more frames are added — in this respect it has a distinct advantage over "batch" approaches such as [16];
- the time complexity of the recursive update is *linear* with the number of currently tracked features, making the algorithm scale benignly;
- the method allows newly appeared features to be included, old features to be discarded, and temporarily occluded or otherwise missing features to be treated naturally;
- dynamic outlier detection and elimination are achieved without recourse to heuristic segmentation strategies;
- there is no reliance on a reference or basis set of features as in [4, 17, 18], the choice of which can bias results drastically, particularly if one of these chosen points is erroneous. All valid features contribute to the structure and motion estimation, improving its stability.
- the method can simultaneously embed different kinds of tracked features, such as edges and corners, within the one esimation process.

These improvements are achieved by posing structure from motion as a parameter estimation problem. The typically non-linear measurement equation relating scene structure, motion and image feature position is linearised about the latest estimates of structure and motion, in an analogous way to the extended Kalman filter. This allows the use of a recursive least-squares estimation algorithm known as the variable state dimension filter (VSDF) which we devised earlier and have previously applied to active camera calibration [7, 6]. Any structure from motion problem that can be encapsulated as a measurement equation (the projection from scene to image) can be solved using the VSDF.

2 The Variable State Dimension Filter (VSDF)

The VSDF is a general way of using a set of features $\mathbf{z}_i(j)$ observed at timestep j to estimate a *global* state vector $\mathbf{x}$, associated with all the features, along with *local* state vectors $\mathbf{y}_i$ associated with individual features. The different state vectors are coupled by linearisable measurement equations.

In [7] we applied the method to camera calibration of an active camera system, in which the global state $\mathbf{x}$ is the calibration parameter vector and is constant over time, and the $\mathbf{y}_i$ relate to points observed in the scene.

In structure from motion applications we again identify the $\mathbf{y}_i$ with individual points so that $\mathbf{y}_i$ relates to the structure, whereas $\mathbf{x}$ describes the motion. However, we also allow $\mathbf{x}$ to be time-varying and so the independent value of $\mathbf{x}$ at frame j is written as $\mathbf{x}(j)$.

A non-linear measurement for each feature i then has the form

$$\mathbf{z}_i(j) = \mathbf{h}_i(j; \mathbf{x}(j), \mathbf{y}_i) + \mathbf{w}_i(j)$$

where $\mathbf{h}_i$ is a vector-valued function (which may change over time) and $\mathbf{w}_i(j)$ is a zero-mean Gaussian distributed vector with covariance $R_i(j)$. The local state

vectors $\mathbf{y}_i$ are taken to be constant over time. $\mathbf{z}_i(j)$ will typically be the projected position of feature i in image j. Note that each measurement depends on the global state vector $\mathbf{x}(j)$ and *one* local state vector $\mathbf{y}_i$. It is this restriction, that the local state vectors are not directly coupled by the measurement equations, that allows the state vector estimation problem to be solved in linear time with the number of tracked features (local states).

Let us consider n features $i = 1, \ldots, n$ as they are tracked over k frames. Let $\mathbf{x}(1)$, the value of the global state vector at the first frame, be set arbitrarily. This serves to fix an initial frame of reference. Each $\mathbf{y}_i$ is provided with an initial estimate $\hat{\mathbf{y}}_i(0)$ and covariance T_{i0}. The maximum likelihood estimators at time step k for $\mathbf{x}(k)$ and $\mathbf{y}_1, \ldots, \mathbf{y}_n$ given a zero-confidence estimate $\hat{\mathbf{x}}^*(k)$ of $\mathbf{x}(k)$ and previous estimates $\hat{\mathbf{y}}_i(k-1)$ of $\mathbf{y}_i$, obtained by expanding $\mathbf{h}_i(k)$ to first order about the $\hat{\mathbf{x}}^*$ and $\hat{\mathbf{y}}_i(k-1)$, are [8]:

$$\hat{\mathbf{x}}(k) = \begin{cases} \mathbf{x}(1) & \text{for } k = 1 \\ \hat{\mathbf{x}}^*(k) + [A - \sum_{i=1}^n B_i C_i^{-1} B_i^T)^{-1} & \\ \quad \times \sum_{i=1}^n (D_i(k)^T - B_i C_i^{-1} E_i(k)^T) R_i(k)^{-1} (\mathbf{z}_i(k) - \mathbf{h}_i(k)] & \text{for } k > 1 \end{cases}$$

$$\hat{\mathbf{y}}_i(k) = \hat{\mathbf{y}}_i(k-1) + C_i^{-1}(E_i(k)^T R_i(k)^{-1}(\mathbf{z}_i(k) - \mathbf{h}_i(k)) - B_i^T(\hat{\mathbf{x}}(k) - \hat{\mathbf{x}}^*(k))) \quad (1)$$

where $D_i(k)$, $E_i(k)$ are the Jacobian matrices for each feature

$$D_i(k) = \frac{\partial \mathbf{h}_i(k)}{\partial \mathbf{x}(k)} \quad , \quad E_i(k) = \frac{\partial \mathbf{h}_i(k)}{\partial \mathbf{y}_i}$$

with $\mathbf{h}_i(k)$, $D_i(k)$, $E_i(k)$ being evaluated at $\hat{\mathbf{x}}^*(k)$, $\hat{\mathbf{y}}_i(k-1)$. Matrices A, B_i, C_i are defined by:

$$A = \sum_{i=1}^n D_i(k)^T R_i(k)^{-1} D_i(k) \quad , \quad B_i = \sum_{i=1}^n D_i(k)^T R_i(k)^{-1} E_i(k) \quad ,$$
$$C_i = T_{i0}^{-1} + \sum_{j=1}^k E_i(j)^T R_i(j)^{-1} E_i(j) \quad . \quad (2)$$

Missing measurements for local states are incorporated by simply ignoring the corresponding terms in the above formulae. Note that all the above update rules have computation time proportional to n.

2.1 Obtaining the Global State Estimate

The simplest algorithm for generating $\hat{\mathbf{x}}^*(k)$ is to set it to $\hat{\mathbf{x}}(k-1)$. However $\mathbf{x}(k)$ may change markedly between time steps k, giving rise to linearisation error in the update equation 1. One way around this problem would be to iterate the global state vector update part of equation 1. There is a short cut, however. Let us consider fitting $\hat{\mathbf{x}}(k)$ to the observations at time step k independently of the $\mathbf{y}_i$. To achieve this we minimise

$$J = \sum_{i=1}^n (\mathbf{z}_i(k) - \mathbf{h}_i(k; \hat{\mathbf{x}}^*(k), \hat{\mathbf{y}}_i(k-1)))^T R_i(k)^{-1} (\mathbf{z}_i(k) - \mathbf{h}_i(k; \hat{\mathbf{x}}^*(k), \hat{\mathbf{y}}_i(k-1))) \quad (3)$$

over $\hat{\mathbf{x}}^*(k)$. The first-order update formula that achieves the minimisation is

$$\hat{\mathbf{x}}^*(k) = \hat{\mathbf{x}}(k-1) + A(k)^{-1} \sum_{i=1}^{n} D_i(k)^T R_i(k)^{-1} (\mathbf{z}_i(k) - \mathbf{h}_i(k)) \ .$$

Here $\mathbf{h}_i(k)$, $D_i(k)$ (and hence $A(k)$) are evaluated at $\hat{\mathbf{x}}(k-1)$, $\hat{\mathbf{y}}(k-1)$. Outliers are removed at this stage by testing the residual J using a χ^2 residual test. If the χ^2 test is failed, points with highest contribution to the residual are removed until the test is passed. Recursive formulae for updating $\hat{\mathbf{x}}^*$ and J when a point is removed are given in [8]. Calculating $\hat{\mathbf{x}}^*(k)$ in this way is much quicker than using equation 1 and removes most of the error in a single step.

3 Affine Structure from Motion

An affine projection from scene to image at frame j is described by the following projection equation [9]:

$$\mathbf{z}_i(j) = M(j)\mathbf{X}_i + \mathbf{t}(j) \tag{4}$$

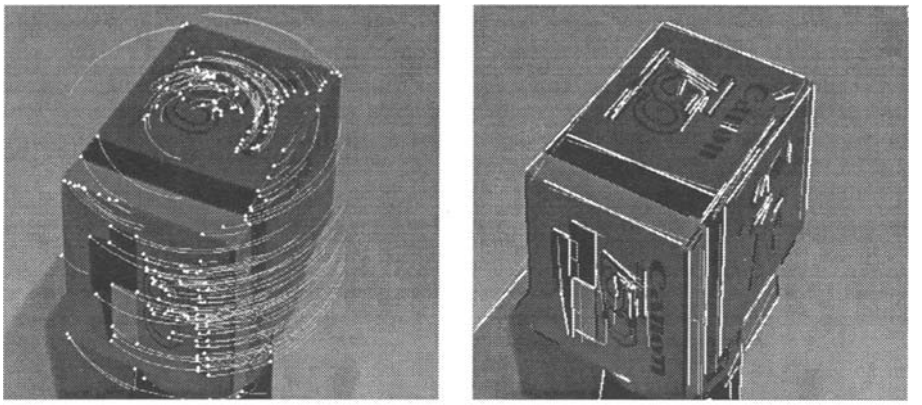
where $\mathbf{X}_i = (X_i, Y_i, Z_i)^T$, $i = 1 \ldots n$ are the 3D positions of n scene points and $\mathbf{z}_i(j) = (x_i(j), y_i(j))^T$ is the projected position of the i^{th} point in the j^{th} frame. $M(j)$ is a 2×3 matrix and $\mathbf{t}(j)$ is a 2×1 translation vector in the image. This projection equation is valid for a small field of view and limited variation in scene depth. The object then is to estimate $\mathbf{X}_i$, $M(j)$ and $\mathbf{t}(j)$ given measurements $\mathbf{z}_i(j)$, $i = 1 \ldots n$, $j = 1 \ldots k$ of the n points in k frames. We consider isolated points for now, located in the image using a corner detector, but show how line segments may be incorporated in [8].

Koenderink and van Doorn [4] have shown that $M(j)$ and $\mathbf{t}(j)$ may be determined by labelling four arbitrary points to define a set of basis vectors. This method was used in [13] to define a fixation point for redirecting gaze onto a moving target during tracking. In practice, for the purpose of determining 3D structure this method is problematic since the four chosen points fix the frame for determining the structure of all the other points, and so any errors in the four points will be amplified in the others. A better method is to optimise the choice of $M(j)$ and $\mathbf{t}(j)$ by using *all* the points, and this is achieved by the VSDF algorithm. The feature positions $\mathbf{z}_i(j)$ are considered as measurements of state vectors $M(j)$, $\mathbf{t}(j)$ and $\mathbf{X}_i$. Let us bundle up the state vectors as follows:

$$\begin{aligned} \mathbf{x}(j) &= (M_{11}(j) \ M_{12}(j) \ M_{13}(j) \ M_{21}(j) \ M_{22}(j) \ M_{23}(j) \ t_1(j) \ t_2(j))^T \\ \mathbf{y}_i &= (X_i \ Y_i \ Z_i)^T \ . \end{aligned} \tag{5}$$

We now substitute the affine camera projection equation 4 into the VSDF formulae using the definitions 5, obtaining

$$\begin{aligned} \mathbf{h}_i(j; \mathbf{x}(j), \mathbf{y}_i) &= M(j)\mathbf{X}_i + \mathbf{t}(j) \\ D(j; \mathbf{x}(j), \mathbf{y}_i) &= \begin{pmatrix} X_i & Y_i & Z_i & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & X_i & Y_i & Z_i & 0 & 1 \end{pmatrix} \ , \\ E(j; \mathbf{x}(j), \mathbf{y}_i) &= \begin{pmatrix} M_{11}(j) & M_{12}(j) & M_{13}(j) \\ M_{21}(j) & M_{22}(j) & M_{23}(j) \end{pmatrix} \ , \end{aligned} \tag{6}$$



a b

Fig. 1. Feature matching for an image sequence of a rotating box. a) Corner matching. The latest corner positions are shown as white blobs, the trajectory of the corner from the start of its "history" in grey. b) Line segment matching, the latest position being shown in white and a line between the previous and current midpoints in grey.

The measurement covariance $R_i(j)$ is determined from the properties of the feature detector. The Plessey corner detector that we have used in our experiments [3] does not have sub-pixel precision, and we have estimated the corner position error standard deviation $\sigma = 0.7$ pixels, independent in the x and y directions. Thus $R_i(j)$ is diagonal with entries σ^2. With R, D and E derived, we can evaluate A, B and C and hence perform the update cycle. To obtain the initial motion and structure $\mathbf{x}(1)$ and $\hat{\mathbf{y}}(0)$ we use the method of Tomasi and Kanade [16], a batch algorithm that we apply over a small number of initial frames. See below for comparison of the two algorithms when applied over a complete image sequence.

4 Results

In Fig. 1 we show the trajectories of corner and line features tracked through a sequence of 20 images of a rotating box, the projection of the rotation axis being aligned with the image y-axis. The corner/line feature detectors and matchers are described in [8].

Three orthographic views of the recovered 3D structure are shown in Fig. 2. Both corner and line segment matches have contributed to the computation. Note the projective distortion of the structure: lines that should be parallel appear to converge towards a vanishing point. Since affine transformations preserve parallelism, this is not an affine distortion, and occurs because the box covers most of the field of view of the camera (25^o), stretching the validity of the affine approximation. For foveal images in our active vision applications this problem will be greatly reduced. In order to demonstrate that 3D structure has been recovered, the last image is back-projected onto the structure. Figure 3 shows the same views of the structure as Fig. 2 rendered in this manner.

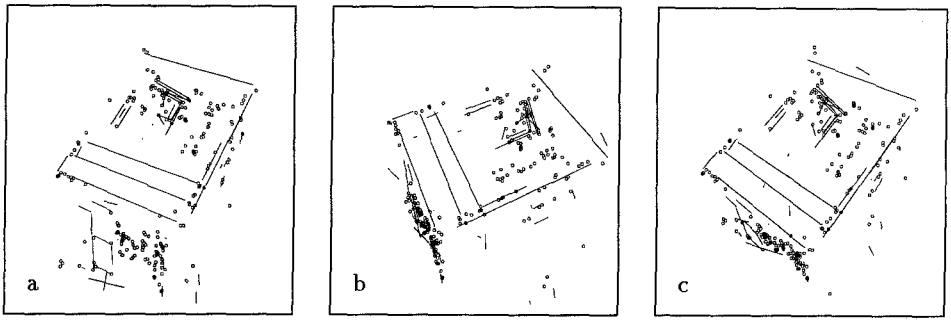


Fig. 2. Three views of the 3D structure obtained by the VSDF algorithm. a) From the left hand side. b) From the right hand side. c) Top view.

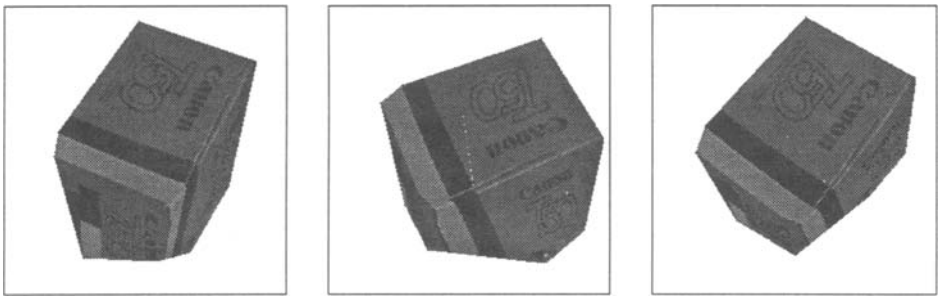


Fig. 3. The same three views of the 3D structure, this time rendered by back-projecting the final image onto the structure.

We have also obtained results on simulated data that demonstrate the efficiency improvement that the VSDF over the measurement matrix factorisation (SVD) method of Tomasi and Kanade [16]. We do not consider here the approach of Szeliski and Kang [15], since the Levenberg-Marquardt minimisation algorithm they used [11] is a slow batch process when a realistic number of images and features are used. In our experiments using 50 or so frames (only 2 seconds worth of video rate data), each iteration of their algorithm required many minutes. The method of Tomasi and Kanade has a number of drawbacks: it too is a batch algorithm, and requires a completely full measurement matrix, i.e. with no missing data [2]. Furthermore the SVD algorithm has complexity $\mathcal{O}(n^2(k+n))$ as opposed to $\mathcal{O}(n)$ for the VSDF. Such questions as the *relative* efficiency and accuracy of different algorithms can be satisfactorily investigated using simulated data. Figure 4 shows results for a set of thirty randomly generated points lying inside the unit sphere, and projected parallel to the Z axis onto the unit circle on the image plane, with additive Gaussian noise of standard deviation 0.005. Between each frame a rotation of five degrees was performed around an oblique

[2] The "hallucination" approach that Tomasi and Kanade describe for filling in missing data from the surrounding data dilutes the main benefit of the algorithm — that it is optimal for the case of affine projection.

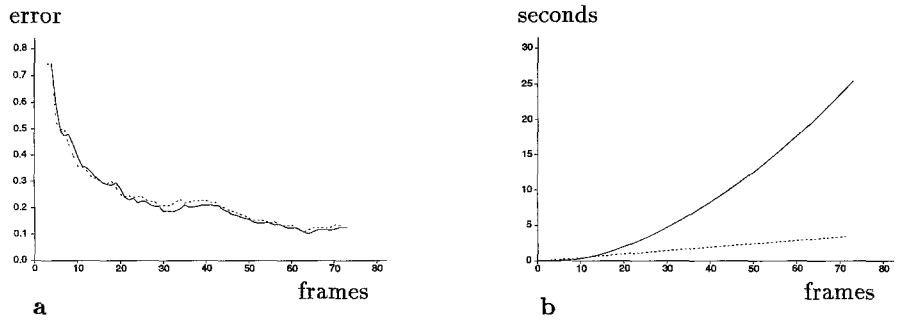


Fig. 4. Comparison of the VSDF and SVD algorithms. a) Measure of accuracy of structure computation over time. b) Timings. The results for the SVD are indicated by the solid line, those for the VSDF by the dotted line.

axis. The SVD was performed at each frame k on all data up to frame k, in order to provide the optimal structure and motion at each time step, to compare with the VSDF. The error is measured by calculating the affine transformation that takes the computed structure closest to the known true structure, as measured in the frame of the new structure. The SVD provides the best possible structure and motion computation on this data, but the difference between the two methods is negligible. We have been unable to create data which causes the two algorithms to diverge. On the other hand, the timings for the two algorithms (on a Sparc-2) demonstrate the efficiency advantage of the VSDF, which is especially marked for longer sequences and/or more points.

5 Conclusions

We have demonstrated the accuracy and efficiency of the VSDF affine structure motion motion algorithm in a number of experiments including direct comparison another method. Both corner and edge token data have been successfully incorporated. Contrary to a recent claim by Weng *et al.* [19] that recursive methods are inherently unstable, the experiments demonstrate stability in both the structure and motion computations, a result facilitated greatly by the natural identification and treatment of outliers within the framework. Furthermore the algorithm is currently being ported to our real-time motion detection and tracking architecture Yorick/Horatio [14, 5]. When this is complete we will be well placed to explore many of the problems which present themselves when considering allocation of attention in dynamic scenes. The VSDF software is available as part of the Horatio vision libraries [5].

References

1. K. J. Bradshaw, P. F. McLauchlan, I. D. Reid, and D. W. Murray. Saccade and pursuit on an active head/eye platform. In J. Illingworth, editor, *Proc. 4th British Machine Vision Conf., Guildford.* BMVA Press, 1993. to appear in Image and Vision Computing Special Issue on BMVC '93.

2. S. Demey, A. Zisserman, and P. Beardsley. Affine and projective structure from motion. In D. Hogg and R. Boyle, editors, *Proc. 3rd British Machine Vision Conf., Leeds*, pages 49–58. Springer-Verlag, September 1992.
3. C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf., Manchester*, pages 147–151, 1988.
4. J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.
5. P. F. McLauchlan. Horatio: libraries for vision applications. Technical Report OUEL 1967/92, Dept. Engineering Science, University of Oxford, October 1992.
6. P. F. McLauchlan and D. W. Murray. Variable state dimension filter applied to active camera calibration. In *Proc SPIE Sensor Fusion VI, Boston MA*, pages 14–25, September 1993.
7. P.F. McLauchlan and D.W. Murray. Active camera calibration for a head/eye platform using a variable state dimension filter. Oxford University Engineering Library report number OUEL 1975/93. Submitted to PAMI, 1993.
8. P.F. McLauchlan, I.D. Reid, and D.W. Murray. Recursive structure and motion from image sequences. Technical Report OUEL report, in preparation, Dept. Engineering Science, University of Oxford, 1994.
9. J. L. Mundy and A. P. Zisserman, editors. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, MA, 1992.
10. D. W. Murray, P. F. McLauchlan, I. D. Reid, and P. M. Sharkey. Reactions to peripheral image motion using a head/eye platform. In *Proc. 4th Int'l Conf. on Computer Vision, Berlin*, pages 403–411, Los Alamitos, CA, 1993. IEEE Computer Society Press.
11. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.
12. L. Quan and R. Mohr. Towards structure from motion for linear features through reference points. In *Proc. IEEE Workshop on Visual Motion*, 1991.
13. I. D. Reid and D. W. Murray. Tracking foveated corner clusters using affine structure. In *Proc. 4th Int'l Conf. on Computer Vision, Berlin*, pages 76–83, Los Alamitos, CA, 1993. IEEE Computer Society Press.
14. P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics*, 3(4):517–535, 1993.
15. R. Szeliski and S.B. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. Technical Report CRL 93/3, DEC Cambridge Research Lab, March 1993.
16. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, 1992.
17. S. Vinther and R. Cipolla. Towards 3D object model acquisition and recognition using 3D affine invariants. In J. Illingworth, editor, *Proc. 4th British Machine Vision Conf., Guildford.* BMVA Press, 1993.
18. D. Weinshall and C. Tomasi. Linear and incremental acquisition of invariant shape models from image sequences. In *Proc. 4th Int'l Conf. on Computer Vision, Berlin*, pages 675–682, Los Alamitos, CA, 1993. IEEE Computer Society Press.
19. 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.