

Motion and structure estimation using long sequence motion models

Xiaoping Hu and Narendra Ahuja

This paper presents several algorithms for estimating motion and structure parameters from long monocular image sequences by using the most appropriate of a set of long sequence motion models. We first present a new two-view motion algorithm and then extend it to long sequence motion analysis. The two-view motion algorithm generally requires six pairs of point correspondences to give a unique solution of the motion parameters. However, when the points used for correspondences lie on a Maybank Quadric, the algorithm requires seven pairs of point correspondences to give all possible double solutions. Object-centred motion representations and models of motion described by up to the second order polynomials are analysed. Two long sequence algorithms are presented, one using interframe matches, and the other using point trajectories. The long sequence algorithms automatically find the proper model that applies to an image sequence and gives the globally optimal solution for the motion and structure parameters under the chosen model. Experimental results with several real image sequences undergoing different motions are presented to demonstrate the performance of the algorithm.

Keywords: estimation, motion, polynomial motion models, structure

This paper addresses the problem of estimating motion and depth structure from a sequence of monocular images under perspective projection (*motion problem* hereafter). After a decade's effort of many researchers¹⁻²¹, significant progress has been made in

Beckman Institute and Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 405 N. Mathews Avenue, Urbana, IL 61801, USA

Paper received: 21 May 1992; revised paper received: 1 March 1993

both theory and algorithms. However, performance on real image data is less than satisfactory in many cases for several reasons. First, it is not easy to obtain reliable, accurate correspondence data from real images. Second, the existing motion algorithms are not robust enough to obtain good results with the state-of-the-art camera resolution. Another reason is probably that the existing algorithms either focus on special cases (e.g. constant motion) or address general cases at the expense of special cases. In the former situation, the algorithm needs prior knowledge about motion and is very restrictive; in the latter, the algorithm assumes arbitrary motion and does not exploit the smoothness of motion. Other factors may significantly affect the accuracy of estimation; for example, the angle of visual field must be large enough to distinguish translation and rotation.

The goal of this paper is to present an approach that integrates different motion algorithms into a system of methods. First, model-based motion estimation algorithms (including algorithms for estimating arbitrary motions) are developed to estimate smooth motions. A strategy for automatically determining the proper model is implemented to adapt the algorithms to different motions. Second, the state-of-art results about two-view motion estimation are integrated to yield a robust, adaptive two-view motion algorithm which provides necessary initial guesses for the long sequence motion algorithms. Third, a global, yet efficient non-linear search of the motion parameters is carried out to ensure globally optimal solution of the motion parameters. We also present the experimental results with real image data which show that the algorithms yield very good estimates for a diversity of image sequences undergoing different motions.

Since the long sequence motion algorithms rely heavily on the two-view motion algorithm and can be more easily described in terms of two-view motion, we first present extensive discussion about two-view

motion and then extend our discussion to multiview motion.

The following section summarizes the current understanding about two view-motion and introduces some notation. We then present a robust nonlinear two-view motion algorithm which gives double or unique solution for any rigid surface. Long sequence motion models and the solutions of motion and structure from long sequences are discussed, and experimental results with real image data presented.

ADVANCED RESULTS ON TWO-VIEW MOTION; A SUMMARY

The two-view motion problem generally involves two aspects:

1. With how many point matches and under what conditions is the solution of motion and structure unique?
2. Given the correspondences, which algorithm gives the most robust estimation of motion and structure?

In this section we discuss the first aspect only, and leave the second to the next section.

A robust algorithm generally requires a profound understanding of the problem. Therefore, to develop a robust algorithm for two-view motion solution, it is necessary to investigate the conditions under which the algorithms work or fail.

The uniqueness problem of two-view motion estimation can be stated as follows: given n pairs of point correspondences $(x_i, y_i), (x'_i, y'_i), i = 1, 2, \dots, n$, under what situations there exist finite, double, or unique solution of the rotation matrix \mathbf{R} , the translation vector \mathbf{T} , and the (positive) depths Z_i, Z'_i , from the motion equation:

$$Z'_i [x'_i \ y'_i \ 1]^T = Z_i \mathbf{R} [x_i \ y_i \ 1]^T + \mathbf{T}, \quad i = 1, 2, \dots, n? \quad (1)$$

This problem has been attacked by many researchers^{1-17,22} and is now almost completely solved. However, a clear statement of the necessary and sufficient conditions for a unique solution of motion and structure is still not available, though the maximum number of solutions and the relationships between the solutions have been known^{6,14,22}. In this section, we summarize several existing sufficient conditions for a finite or a unique solution of motion parameters. These conditions are the theoretic foundations of the unified nonlinear two-view motion algorithm that is introduced in this paper. We so far cannot express a necessary and sufficient condition for this problem in a nice, closed form, though we can show that the conditions presented here are very close to being necessary and sufficient.

For planar surface (with the surface shape known in advance), we now know that in general at most two

solutions can result^{3-5,10,11,17}, although in rare situations there may be an infinite number of solutions^{5,11}. Most researchers who investigated the plane motion problem also proposed algorithms for estimating motion and structure parameters.

For a nonplanar surface, we now know under what situations ambiguous solutions arise, how the spurious solution are related to the true motion and structure¹⁴, and how many solutions are admissible^{6,22}. We also have sufficient conditions for double or unique solution of motion and structure²². There have also been many algorithms developed for estimating motion and structure of a general surface^{2,4,7,8,23-27}.

Although in theory multiple solutions arise only for Maybank Quadrics, in practice we have found multiple solutions exist for general surfaces, especially when the object is small or far away. This is due to the fact that when the object is small or far away, it is almost always possible to fit a Maybank Quadric (or even a planar patch) to the object surface. For example, a human face at a distance can be fitted very well by an elliptical sphere, an hyperbolic paraboloid, or even a planar patch. Therefore, linear algorithms hardly work for small objects or objects at large distances. Even nonlinear algorithms require a sufficiently large visual angle to estimate the motion parameters with reasonable accuracy. It is thus important to have algorithms that require the least stringent conditions for unique solution of motion and yield all possible solutions when multiple solutions exist. The following are the most advanced results in the uniqueness problem, and comprise the backbone of our two-view motion algorithm in the next section.

We assume that \mathbf{R}_0 and \mathbf{T}_0 are the true motion parameters and \mathbf{R} and \mathbf{T} are the other possible set of solution. We assume that $\mathbf{T}_0 \neq 0$. When $\mathbf{T}_0 = 0$, the motion problem is generally more determined²².

First, let us introduce two matrix operations, \otimes and rs , called *Kronecker product* and *row stretch*, respectively, that will simplify notation used in this paper. Let $\mathbf{A} = (a_{ij})_{m \times n}$ and $\mathbf{B} = (b_{ij})$ be any two matrices. Then the Kronecker product is defined as:

$$\mathbf{A} \otimes \mathbf{B} \equiv (a_{ij} \mathbf{B}) = \begin{bmatrix} a_{11} \mathbf{B} & \cdots & a_{1n} \mathbf{B} \\ \cdots & \cdots & \cdots \\ a_{m1} \mathbf{B} & \cdots & a_{mn} \mathbf{B} \end{bmatrix} \quad (2)$$

The row stretch of \mathbf{A} is a vector obtained by placing all rows of \mathbf{A} into one column in the row order:

$$rs \mathbf{A} = [a_{11} \cdots a_{1n} \cdots a_{m1} \cdots a_{mn}]^T \quad (3)$$

Let

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (4)$$

be any essential matrix of the form $\mathbf{T} \times \mathbf{R}$ ^{12,28}. Then all information for the motion parameters available from

point correspondences is the motion epipolar equations:

$$\begin{bmatrix} \Theta_1'^T \otimes \Theta_1^T \\ \Theta_2'^T \otimes \Theta_2^T \\ \dots \\ \Theta_n'^T \otimes \Theta_n^T \end{bmatrix} (rs\mathbf{E}) \equiv \mathbf{A}_n \mathbf{h} = \mathbf{A}_n \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_9 \end{bmatrix} = 0 \quad (5)$$

and the constraint equation for the essential matrix^{13,28}:

$$(\mathbf{e}_1 \cdot \mathbf{e}_2)(\mathbf{e}_1 \cdot \mathbf{e}_3)\mathbf{e}_1 + (\mathbf{e}_1 \cdot \mathbf{e}_2)(\mathbf{e}_2 \cdot \mathbf{e}_3)\mathbf{e}_2 + (\mathbf{e}_2 \cdot \mathbf{e}_3)(\mathbf{e}_1 \cdot \mathbf{e}_3)\mathbf{e}_3 = 0 \quad (6)$$

where $\Theta_i = [x_i \ y_i \ 1]^T$, $\Theta_i' = [x_i' \ y_i' \ 1]^T$, $i = 1, 2, \dots, n$ are n pairs of point correspondences. \mathbf{A}_n will be called the *coefficient matrix* hereafter.

For $n = 5$, Netravali *et al.*¹³ and Faugeras *et al.*¹² have shown that there are at most 10 sets of motion solutions. However, not all five correspondences assure this result. A necessary and sufficient condition is that there are five linearly independent epipolar equations, or equivalently, that the matrix \mathbf{A}_5 defined by the five pairs of points has full row rank 5. The maximum number of physically feasible solutions from five correspondence that has been numerically found is 8²⁹. For $n = 6$, if the coefficient matrix \mathbf{A}_6 has a full row rank 6, then the motion parameters can be uniquely determined with probability 1^{13,22}. For $n = 7$, if the coefficient matrix \mathbf{A}_7 has a full row rank 7, then the maximum number of solutions is 3²⁹, and the probability for two or three solutions to exist is zero. For $n \geq 8$, as is well known, if the coefficient matrix \mathbf{A}_n has a row rank of 8, then the motion is uniquely determined. Multiple solutions exist only if the points used to lie on a Maybank Quadric. However, if the Maybank Quadric is uniquely defined by six or more space points, then at most two physically feasible solutions can be sustained²².

Here we see that the row rank of \mathbf{A}_n determines the maximum number of possible solutions; but the actual number of physically acceptable solutions depends on the positions of all image points. From these results we know that given $n \geq 6$ point correspondences, if \mathbf{A}_n has row rank of 6 then the motion can be uniquely determined with probability 1.

Now the problems that remain are:

1. How to obtain an optimal solution of the motion parameters when there is only one solution?
2. How to know whether the solution is unique?
3. How to obtain other solutions when there are multiple solutions?

The last two problems may appear insignificant since when six or more correspondences are available, multiple solutions rarely exist. However, as we have pointed out earlier, when the object is small or far away, the points can often be fitted by a Maybank Quadric and hence multiple solutions are admitted.

Therefore, to make the motion algorithm more robust and applicable, it is desirable that the algorithm requires the least stringent conditions for unique solution and yields all solutions when multiple solutions exist.

AN OPTIMAL NONLINEAR TWO-VIEW MOTION ALGORITHM

A number of linear^{2,7} and nonlinear^{8,23,25-27,30} motion algorithms have been developed for motion estimation. Linear algorithms rarely work for noisy data, though they have been used to produce initial guesses for nonlinear algorithms by many authors. The existing nonlinear algorithms all assume that the motion is uniquely determined and require good initial guesses since they do not do global search. When the motion is a pure rotation or involves a small translation, the existing algorithms often fail to produce good estimates of the rotation parameters. We have implemented many algorithms^{2,25,26}, and find that the following two-stage nonlinear algorithm performs most robustly and applies to most general situations. Since the algorithm solves for rotation parameters first and ten translation parameters, the computing efficiency is also much higher than other nonlinear algorithms.

Outline of the algorithm

Let $\mathbf{R} = (r_{ij})$ and $\mathbf{T} = [t_1 \ t_2 \ t_3]^T$ (\mathbf{T} can be zero in this algorithm) be the rotation matrix and translation vector between the two views. Let $\Theta_i = [x_i \ y_i \ 1]^T$, $\Theta_i' = [x_i' \ y_i' \ 1]^T$, $i = 1, 2, \dots, n (n \geq 5)$, be n pairs of correspondences, where (x_i, y_i) and (x_i', y_i') denote the image coordinates in the first and the second views. Then the algorithm will minimize the sum of squared residues of the motion epipolar line equations:

$$S_1 = \sum_{i=1}^n \{(\Theta_i' \times \mathbf{R} \Theta_i) \cdot \mathbf{T}\}^2 = \mathbf{T}^T \Pi_n^T \Pi_n \mathbf{T} = \|\Pi_n \mathbf{T}\|^2 \quad (7)$$

where

$$\Pi_n = \begin{bmatrix} (\Theta_1' \times \mathbf{R} \Theta_1)^T \\ \vdots \\ (\Theta_n' \times \mathbf{R} \Theta_n)^T \end{bmatrix}, \quad \|\mathbf{T}\|^2 = 1 \quad (8)$$

For a given estimate \mathbf{R} , the optimal estimate of \mathbf{T} corresponds to the eigenvector of $\Pi_n^T \Pi_n$ associated with $\Pi_n^T \Pi_n$'s least eigenvalue λ_m , which is just the minimum value of S_1 . Therefore, minimizing S_1 can be reduced to minimizing λ_m , which is a function of only the rotation matrix. The algorithm is thus divided into two steps: (1) first search for \mathbf{R} to minimize the least eigenvalue λ_m of $\Pi_n^T \Pi_n$; (2) then estimate \mathbf{T} in a closed form by solving for the eigenvector of $\Pi_n^T \Pi_n$.

associated with the smallest eigenvalue λ_m . The sign of \mathbf{T} is to be determined such that the depths are positive.

If the motion is a pure rotation, then the resulting \mathbf{T} in the above algorithm could be anything but zero. To determine if the translation is zero, we use the following confidence measure:

$$S_2 = \sum_i \left(x'_i - \frac{r_{11}x_i + r_{12}y_i + r_{13}}{r_{31}x_i + r_{32}y_i + r_{33}} \right)^2 + \left(y'_i - \frac{r_{21}x_i + r_{22}y_i + r_{23}}{r_{31}x_i + r_{32}y_i + r_{33}} \right)^2 \quad (9)$$

If S_2 is close to zero (the particular threshold depends on the camera parameters), then the estimated \mathbf{T} is unreliable. Otherwise, the translation is visible and hence the estimation of \mathbf{T} is trustworthy.

The first step of the algorithm is nonlinear, but the second step is linear and gives a closed form. Therefore most of the computation is spent on the first step. Fortunately, only the rotation matrix \mathbf{R} and hence only three unknowns are involved in the first step. Thus even an exhaustive search algorithm can be applied to estimate \mathbf{R} . To do this, we represent the rotation matrix by the three-angle representation:

$$\mathbf{R} = \mathbf{A}_X(\omega_X) \mathbf{A}_Y(\omega_Y) \mathbf{A}_Z(\omega_Z) \quad (10)$$

where

$$\mathbf{A}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega_X & -\sin \omega_X \\ 0 & \sin \omega_X & \cos \omega_X \end{bmatrix}$$

$$\mathbf{A}_Y = \begin{bmatrix} \cos \omega_Y & 0 & \sin \omega_Y \\ 0 & 1 & 0 \\ -\sin \omega_Y & 0 & \cos \omega_Y \end{bmatrix}$$

$$\mathbf{A}_Z = \begin{bmatrix} \cos \omega_Z & -\sin \omega_Z & 0 \\ \sin \omega_Z & \cos \omega_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Then there are only three unknowns in \mathbf{R} . If we exhaustively search each angle ω_X , ω_Y or ω_Z to within 1° in the range of $(-100^\circ, 100^\circ)$, then the computation time is proportional to 8×10^6 . For real images of small motion, the search range can often be much smaller. After the rough minimum has been found, the estimates can be refined in the neighbourhood of the rough minimum to improve the accuracy to any prerequisite level of precision.

The search of ω_X , ω_Y and ω_Z can be accelerated if an optimization technique and/or an initial estimate is used. For rotation angle of less than 10° , it often suffices to start the search of $(\omega_X, \omega_Y, \omega_Z)$ at $(0, 0, 0)$ or at the estimate obtained from linear solution. To avoid the dependence of the algorithm on the initial guess, the search should start at all points in a predetermined grid, e.g. one point in every $10^\circ \times 10^\circ \times 10^\circ$ cube. We have experimentally found that

in the neighbourhood of the global minimum, different optimization algorithms (e.g. Hooke-Jeeves method³¹, or gradient method³²) all work.

S_1 is a weighted version of the following criterion used by many researchers^{18-20,26,33}:

$$S_3 = \sum_i (x'_i - u_i)^2 + (y'_i - v_i)^2 \equiv \sum_i d_i^2 \quad (12)$$

where u_i and v_i are the expected position of x_i and y_i after the motion, u_i and v_i are related to the depth Z_i and the motion parameters \mathbf{R} and \mathbf{T} through:

$$u_i = \frac{(r_{11}x_i + r_{12}y_i + r_{13}) Z_i + t_1}{(r_{31}x_i + r_{32}y_i + r_{33}) Z_i + t_3}$$

$$v_i = \frac{(r_{21}x_i + r_{22}y_i + r_{23}) Z_i + t_2}{(r_{31}x_i + r_{32}y_i + r_{33}) Z_i + t_3} \quad (13)$$

For a given estimate of \mathbf{R} and \mathbf{T} , (u_i, v_i) is on the motion epipolar line l_i defined by the motion and (x_i, y_i) :

$$[u_i \ v_i \ 1](\mathbf{T} \times \mathbf{R} \Theta_i) = 0 \quad (14)$$

The value of Z_i determines the position of (u_i, v_i) on l_i . Therefore, the optimal solution of Z_i is that which makes d_i the shortest distance between point (x'_i, y'_i) and line l_i . Let $\mathbf{T} \times \mathbf{R} \Theta_i = (a_i, b_i, c_i)^T$, then S_3 is equivalent to:

$$S_3 = \sum_i \{ \Theta_i \cdot (\mathbf{T} \times \mathbf{R} \Theta_i) \}^2 / (a_i^2 + b_i^2) \quad (15)$$

which indicates that S_1 is a weighted version of S_3 . Hence, by choosing S_1 instead of S_3 as the optimality criterion, the motion algorithm can be significantly simplified without loss of performance.

Iterative procedure

Here we present an iterative procedure for searching for the global minimum. This procedure converges very quickly, and works very well when the rotation angles are smaller than 20° , although, like most iteration procedures there is no guarantee that it always converges to the global minimum.

First, note that S_1 is a function of the estimates of \mathbf{R} and \mathbf{T} . For an estimated \mathbf{R} , the optimal solution of \mathbf{T} that minimizes S_1 can be obtained in closed form, as discussed above. We then fix \mathbf{T} at its most recently updated value and search for the angles of \mathbf{R} such that S_1 is minimized. Thus, we alternatively update the estimates of \mathbf{R} and \mathbf{T} until the algorithm converges to a fixed point which is the optimal solution. The details follow.

In the small rotation case:

$$\mathbf{R} \approx \mathbf{I} + \Omega \times = \mathbf{I} + \begin{bmatrix} 0 & -\omega_Z & \omega_Y \\ \omega_Z & 0 & -\omega_X \\ -\omega_Y & \omega_X & 0 \end{bmatrix} \quad (16)$$

If we rewrite the motion epipolar equation:

$$(\Theta'_i \times \mathbf{R} \Theta_i) \cdot \mathbf{T} = 0 \quad (17)$$

as:

$$(\mathbf{T} \times \Theta'_i) \cdot \mathbf{R} \Theta_i = 0 \quad (18)$$

we then have:

$$\Psi_i \cdot (\mathbf{I} + \Omega \times) \Theta_i = 0 \quad (19)$$

or:

$$(\Theta_i \times \Psi_i)^T \Omega = \Theta_i^T \Psi_i \quad (20)$$

where $\Psi_i = \mathbf{T} \times \Theta'_i$. We can then obtain a linear least squares solution of $\Omega = [\omega_x \ \omega_y \ \omega_z]^T$ from (20) for a given \mathbf{T} .

Equations (20) and (17) constitute an iteration pair. That is, after we have obtained an estimate of \mathbf{T} , we can use (20) to solve for Ω , and then use (17) to obtain a refined estimate of \mathbf{T} .

Special surfaces

The nonlinear algorithm above yields one optimal solution for all kinds of surfaces. However, for special surfaces like planes and Maybank Quadrics, the solution is generally not unique.

First, let us discuss planar surface. Although planar surface can be a branch of a special form of Maybank Quadric (two planes), we have to deal with it differently from other Maybank Quadrics since planes are more often seen than other Maybank Quadrics, and in general we cannot determine the Maybank Quadrics associated with a plane²², since the other branch has one degree of freedom (we only know that it passes the two origins of the coordinates before and after motion).

Although there are many linear algorithms that give closed-form solution of plane motion and structure^{3-5,10,11}, these linear algorithms need *a priori* knowledge of the surface shape before they can be applied. To determine whether the surface is planar or not, a general algorithm is still needed. For planar surfaces, there are generally two solutions for the motion and structure, and the nonlinear algorithm above will converge to any one of the solutions. We now describe the method with which we determine whether the surface is planar, and if it is, we refine the estimation by applying plane motion algorithms.

If the projection of a plane in the image plane is not a line, then the plane must have an equation:

$$\mathbf{N}^T \mathbf{X} = [n_1 \ n_2 \ n_3] \Theta Z = [n_1 \ n_2 \ n_3] \begin{bmatrix} x \\ y \\ l \end{bmatrix} Z = 1 \quad (21)$$

or:

$$x n_1 + y n_2 + n_3 = 1/Z \quad (22)$$

where $\mathbf{X} = \Theta Z$ is a point on the planar surface with correspondence $\mathbf{X}' = \Theta' Z'$. After the rotation \mathbf{R} and translation \mathbf{T} are known, the depths Z and Z' can be obtained from:

$$\begin{bmatrix} Z \\ Z' \end{bmatrix} = \begin{bmatrix} \Theta^T \Theta & -\Theta^T \mathbf{R}^T \Theta' \\ -\Theta^T \mathbf{R}^T \Theta' & \Theta'^T \Theta' \end{bmatrix}^{-1} \begin{bmatrix} -\Theta^T \mathbf{R}^T \\ \Theta'^T \end{bmatrix} \mathbf{T} \quad (23)$$

if and only if Θ is not parallel to \mathbf{T} , minimizing the squared residue of the motion representation (1). Depth Z can also be solved for by minimizing (x', y') to the epipolar line l as defined by (18), or by equalizing (u, v) defined as in (13) to the intersection point of l and its perpendicular line passing (x', y') . Then with three or more pairs of correspondences of points noncolinear in the image plane we can obtain a linear least squares estimation of \mathbf{N} from equation (22).

If the nonlinear algorithm converges to rotation \mathbf{R}_1 and translation \mathbf{T}_1 , the associated structure parameters \mathbf{N}_1 of the plane can be obtained by a linear least squares solution of equation (22). The other set of motion and structure parameters can be obtained by a plane motion decomposition of the *plane motion matrix*:

$$\mathbf{K} = \mathbf{R}_1 + \mathbf{T}_1 \mathbf{N}_1^T \quad (24)$$

using the method, for example, in Hu and Ahuja¹¹ or Wang *et al.*¹⁰.

There are many ways for determining if the surface is planar or not. The following method is used in our work and gives very accurate judgement. Assume we have obtained one solution \mathbf{R}_1 and \mathbf{T}_1 which minimizes S_1 . Then we solve for a hypothesized plane structure \mathbf{N}_1 from (22). We then obtain the other set of motion parameters $\mathbf{R}_2, \mathbf{T}_2$ and \mathbf{N}_2 by decomposing \mathbf{K} defined in (24). We then substitute \mathbf{R}_2 and \mathbf{T}_2 into (7) to check if:

$$S_1(\mathbf{R}_2, \mathbf{T}_2) - S_1(\mathbf{R}_1, \mathbf{T}_1) < \epsilon \cdot S_1(\mathbf{R}_1, \mathbf{T}_2) \quad (25)$$

where ϵ is a preset constant (e.g. 0.1). If so, the surface is planar; if not, the surface is not planar. In case \mathbf{K} has unique decomposition, then the above method may not work. However, in this case the motion and the depths are uniquely determined. We can also use the sum of the distances of the estimated 3D points to the hypothesized plane to determine if the surface is planar or not. When the surface is determined to be planar, then the plane motion algorithm¹¹ is used to obtain a better estimation of the motion and structure parameters.

One question associated with the above method is: what if there exist two motion solutions such that one leads to planar surface and the other nonplanar surface? If this situation occurs then the above method will fail irrespective of whether the nonlinear algorithm, converges to the planar solution or nonplanar solution. However, this is unlikely. As we have pointed out²², when the surface is planar, six correspondences of points that do not lie on a quadratic curve in the image plane suffice to exclude motion solutions that

lead to nonplanar surfaces. We now briefly illustrate why.

Let \mathbf{R}_1 , \mathbf{T}_1 and \mathbf{N}_1 be a planar motion and structure solution and \mathbf{K} defined in equation (24) be the corresponding plane motion matrix. Then for each correspondence pair Θ and Θ' we have:

$$\gamma\Theta' = \mathbf{K}\Theta \tag{26}$$

where:

$$\gamma = \sqrt{\frac{\Theta^T \mathbf{K}^T \mathbf{K} \Theta}{\Theta'^T \Theta'}} \tag{27}$$

Assume \mathbf{R} and \mathbf{T} is another set of solution leading to nonplanar surface and $\mathbf{E} = \mathbf{T} \times \mathbf{R}$ is the essential matrix. Then Θ and Θ' must satisfy the motion epipolar equation:

$$\Theta'^T \mathbf{E} \Theta = 0 \tag{28}$$

Now substituting equation (26) into the above equation, we obtain:

$$\Theta^T \mathbf{K}^T \mathbf{E} \Theta = 0 \tag{29}$$

Equation (29) indicates that the points used must lie on a quadratic curve in the image plane. Exception may occur when $\mathbf{K}^T \mathbf{E}$ is skew-symmetric, but in this case we can prove that²² $\mathbf{K} = \mathbf{R} + \mathbf{T}\mathbf{N}^T$ for some vector \mathbf{N} . That is, solution \mathbf{R} and \mathbf{T} leads to planar surface. Since five points may uniquely define a quadratic curve in the image plane, six correspondences of points that do not lie on a quadratic curve in the image plane suffice to exclude motion solutions that leads to nonplanar surfaces.

The situation for general Maybank Quadrics is more complicated. The algorithm presented below requires seven correspondences of points that satisfy some condition to determine the other possible solution from one known solution if the surface is Maybank Quadric.

Assume one solution $\mathbf{R}_0 = (r_{ij})$ and $\mathbf{T}_0 = [t_1 \ t_2 \ t_3]^T$ has been obtained with the nonlinear search algorithm. The goal below is to determine whether there is another solution which satisfies the same image data; if so, obtain an optimal solution of the alternative solution. It is well known that if another solution \mathbf{R} and \mathbf{T} satisfies the data, then the points used for correspondences must lie on a Maybank Quadric defined as follows:

$$\frac{1}{2} Z \Theta^T (\mathbf{R}_0^T \mathbf{E} + \mathbf{E}^T \mathbf{R}_0) \Theta + \mathbf{T}_0^T \mathbf{E} \Theta = 0 \tag{30}$$

where Z is the depth of the image point Θ and $\mathbf{E} = \mathbf{T} \times \mathbf{R}$. The depth Z can be solved as described earlier, since \mathbf{R}_0 and \mathbf{T}_0 are known. It is well known that the Maybank Quadric passes through $Z\Theta = -\mathbf{R}_0^T \mathbf{T}_0 \equiv \mathbf{O} = [o_x \ o_y \ o_z]^T$, where \mathbf{O} is the new origin of the coordinates for camera-centred motion model. Now given $n (n \geq 7)$ points $\Theta_i, i = 1, 2, \dots, n$, with their depths solved from equation (23), we first fit the points

with a more general surface of the form:

$$\frac{1}{2} Z \Theta^T \mathbf{A} \Theta + \mathbf{B}^T \Theta = 0 \tag{31}$$

where \mathbf{A} is symmetric and:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_4 & a_5 \\ a_4 & a_2 & a_6 \\ a_5 & a_6 & a_3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{32}$$

From equation (31) we can determine \mathbf{A} and \mathbf{B} linearly as:

$$\begin{bmatrix} \frac{o_x^2}{2} & \frac{o_y^2}{2} & \frac{o_z^2}{2} & o_x o_y & o_x o_z & o_y o_z & o_x & o_y & o_z \\ \frac{x_1^2 Z_1}{2} & \frac{y_1^2 Z_1}{2} & \frac{Z_1}{2} & x_1 y_1 Z_1 & x_1 Z_1 & y_1 Z_1 & x_1 & y_1 & 1 \\ \cdot & \cdot \\ \frac{x_n^2 Z_n}{2} & \frac{y_n^2 Z_n}{2} & \frac{Z_n}{2} & x_n y_n Z_n & x_n Z_n & y_n Z_n & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{Q}_n \mathbf{C} = 0 \tag{33}$$

if and only if \mathbf{Q}_n has a row rank of 8 or above. After \mathbf{A} and \mathbf{B} have been solved, we can then solve \mathbf{E} from:

$$\mathbf{R}_0^T \mathbf{E} + \mathbf{E}^T \mathbf{R}_0 = \mathbf{A}, \quad \mathbf{T}_0^T \mathbf{E} = \mathbf{B}^T \tag{34}$$

or from:

$$\begin{bmatrix} 2r_{11} & 0 & 0 & 2r_{12} & 0 & 0 & 2r_{13} & 0 & 0 \\ 0 & 2r_{21} & 0 & 0 & 2r_{22} & 0 & 0 & 2r_{23} & 0 \\ 0 & 0 & 2r_{31} & 0 & 0 & 2r_{32} & 0 & 0 & 2r_{33} \\ r_{21} & r_{11} & 0 & r_{22} & r_{12} & 0 & r_{23} & r_{13} & 0 \\ r_{31} & 0 & r_{11} & r_{32} & 0 & r_{12} & r_{33} & 0 & r_{13} \\ 0 & r_{31} & r_{21} & 0 & r_{32} & r_{22} & 0 & r_{33} & r_{23} \\ t_1 & 0 & 0 & t_1 & 0 & 0 & t_1 & 0 & 0 \\ 0 & t_2 & 0 & 0 & t_2 & 0 & 0 & t_2 & 0 \\ 0 & 0 & t_3 & 0 & 0 & t_3 & 0 & 0 & t_3 \end{bmatrix} r_s \mathbf{E} = \mathbf{P} \quad r_s \mathbf{E} = \mathbf{C} \tag{35}$$

Equations (33) and (35) can be merged into one:

$$\mathbf{Q}_n \mathbf{P} r_s \mathbf{E} = 0 \tag{36}$$

Since there are at most two solutions of \mathbf{E} that satisfy equation (34)¹⁴, matrix $\mathbf{Q}_n \mathbf{P}$ must have a row rank of at least 8 as long as \mathbf{R}_0 is a rotation matrix and $\mathbf{T}_0 \neq 0$. Therefore with seven or more points that make $\mathbf{Q}_n \mathbf{P}$ have row rank of 8 or above, \mathbf{E} can be linearly determined from equation (36). The linear least squares solution of $r_s \mathbf{E}$ is just the eigenvector of $\mathbf{P}^T \mathbf{Q}_n^T \mathbf{Q}_n \mathbf{P}$ (subject to $\|r_s \mathbf{E}\| = 1$) associated with the least eigenvalue. After \mathbf{E} has been solved, we can then obtain an optimal solution of \mathbf{T} and \mathbf{R} using the method in Philip³⁴ or the simpler linear decomposition algorithm presented in Appendix B.

To determine whether the surface is Maybank Quadric or not, we use the following method. If the surface is indeed a Maybank Quadric, then \mathbf{E} obtained from equation (36) will be an essential matrix and thus the decomposed \mathbf{T} and \mathbf{R} will satisfy the image data

very well. Otherwise, \mathbf{E} will not be an essential matrix and thus the decomposed \mathbf{T} and \mathbf{R} will not satisfy the image data. Even for a Maybank Quadric, \mathbf{E} obtained from equation (36) may not be exactly essential if the data is noisy. Therefore, after \mathbf{T} and \mathbf{R} are obtained, we need to search around them to minimize criterion S_1 . Again, equation (25) is used to determine whether \mathbf{T} and \mathbf{R} are really a solution, and whether the surface is Maybank Quadric. In case that two solutions are admitted, the positive depth constraint can be employed to identify the spurious solution if one solution yields depths of opposite signs. The ambiguity of the motion may also be removed by the long sequence motion algorithm that is to be introduced in the next section.

Summary of the two-view motion algorithm

Before we summarize the nonlinear motion algorithm, let us discuss how to properly use the correspondence data to make the algorithms more robust. When a large number of correspondences is available, a proper selection of the correspondence data for motion estimation may significantly improve the accuracy of the estimates of motion parameters. It has been found that the motion estimation is very sensitive to the size of the camera visual field. If the visual field is too small, motion estimation will suffer from the *motion aperture* problem because motion is intrinsically ambiguous for a small object. Therefore, to make maximum use of the camera resolution, the correspondence points should be distributed over the whole image plane. The points must be *uniformly* distributed lest they are crowded into some region and again make motion algorithms suffer from the motion aperture problem. Since image coordinates of far away points are less sensitive to translational motion, the correspondences of such points yield more accurate estimation of rotation parameters. On the other hand, image correspondences of nearby object points yield more accurate estimation of translation vector. Therefore, a minimum interpoint distance (e.g. 10 pixels) is chosen to select some of the correspondence for motion estimation. Further, points of small disparity values are more heavily weighted (the weight is proportional to the disparity value) than those of large disparity values in the solution of rotation and the opposite is true in the solution of translation.

The algorithm in this section can be summarized as follows:

Step 1. Check the interpoint distances of the correspondences and remove some from those crowded regions. Weight each correspondence pair according to its disparity value for the solution of rotation and translation as described above.

Step 2. Search (exhaustively, iteratively or starting from an initial estimate) ω_X , ω_Y and ω_Z to within a predetermined coarse level of resolution (e.g. 1°) to minimize the least eigenvalue λ_m of $\Pi_n^T \Pi_n$. Let the optimal estimates of ω_X , ω_Y and ω_Z be ω_1 , ω_2 and ω_3 .

Step 3. Use any optimal searching procedure to search around ω_1 , ω_2 and ω_3 to any prerequired precision (e.g. 0.001°) to minimize λ_m again. Let the optimal estimates of ω_X , ω_Y and ω_Z be ω_X^* , ω_Y^* and ω_Z^* , and let the resulted rotation matrix be \mathbf{R} .

Step 4. Substitute ω_X^* , ω_Y^* and ω_Z^* into equation (8) to compute $\Pi_n^T \Pi_n$. If all eigenvalues of $\Pi_n^T \Pi_n$ are close to zero, or if the confidence measure defined by S_2 is low, then $\mathbf{T} = 0$; otherwise, \mathbf{T} equals the eigenvector of $\Pi_n^T \Pi_n$ associated with λ_m .

Step 5. If the translation is zero and the points (six or more) are not on a quadratic curve in the image plane²², then the motion is unique; otherwise, estimate the hypothesized plane normal \mathbf{N} and decompose the plane motion matrix $\mathbf{K} (= \mathbf{R} + \mathbf{T}\mathbf{N}^T)$ to obtain the other set of motion \mathbf{R}_2 , \mathbf{T}_2 . Check if \mathbf{R}_2 , \mathbf{T}_2 also minimizes criterion S_1 using equation (25). If so, the surface is planar, if not, the surface is nonplanar.

Step 6. If the surface is not planar and the translation is not zero, then compute the hypothesized Maybank Quadric parameters \mathbf{A} and \mathbf{B} . An alternative set of motion solution \mathbf{R}_2 , \mathbf{T}_2 is then obtained from equation (36) and is locally refined by searching. If inequality (25) is satisfied, then the surface is Maybank Quadric and two solutions result. Otherwise, \mathbf{R} and \mathbf{T} consist of the unique solution.

This algorithm has four good features:

1. It works for general motion (including pure rotation and translational motion) of objects of planar or general surfaces, and significantly outperforms the existing algorithms in efficiency and robustness.
2. It does not depend on initial guesses of the motion parameters, and gives a robust and globally optimal solution for the motion parameters for the given criterion S_1 .
3. Irrespective of how many correspondences are used, it gives unique solution as long as the motion is uniquely determined by the correspondence data. Therefore this algorithm requires the least stringent condition for unique solution of motion parameters.
4. The search space of \mathbf{R} is small enough for real time application.

MOTION ESTIMATION FROM LONG SEQUENCE OF IMAGES

The accuracy of motion estimation can be greatly improved when a long sequence of images is used³⁵. This is because more evidence about the motion is present. In the previous approaches^{18-21,35,36}, either arbitrary motion, or restricted motion such as constant motion, or velocity decomposition models are assumed. Debrunner and Ahuja³⁷ and Tomasi and Kanade^{38,39} obtained impressive results by using orthographic projection and the factorization method³⁷. They assume arbitrary shape and need trajectories of points as input. Debrunner and Ahuja analysed the constant motion case, and Tomasi and

Kanade analysed the arbitrary motion case. The motion problem is much simpler under orthographic projection since the motion can be reduced to pure rotations. Even under perspective projection, the solution of a pure rotation or a pure translation is much more accurate than the solution of a motion involving both rotation and translation, and two views suffice to yield good results.

In this section we introduce a model-based approach to motion estimation under perspective projection which uses an accurate, yet flexible, object-centred motion representation, and optimally adapts to different motions. It is evident that a model-based motion algorithm should give better results than a general motion algorithm, since a correct model involves fewer unknowns and employs more constraints. We first present the algorithms for several motion models, and then present a strategy for automatically determining the proper model to be used.

Camera-centred and object-centred motion representations

First let us describe the camera- and object-centred motion representations, both of which are used in our approach. For different motion models, we find that different representations are more convenient.

The camera-centred interframe motion representation is as follows¹⁸:

$$\mathbf{X}_n = \mathbf{R}_n \mathbf{X}_{n-1} + \mathbf{t}_n \quad (37)$$

A general representation for motion between the i th and j th ($i > j$) frames is:

$$\mathbf{X}_i = \mathbf{R}_{ij} \mathbf{X}_j + \mathbf{t}_{ij} \quad (38)$$

where:

$$\mathbf{R}_{ij} = \mathbf{R}_i \mathbf{R}_{i-1} \dots \mathbf{R}_{j+1} \quad (39)$$

$$\mathbf{t}_{i,j} = \mathbf{t}_i + \sum_{k=j+1}^{i-1} \mathbf{R}_{i,k} \mathbf{t}_k \quad (40)$$

This representation gives a purely mathematical description of the relationships between multi-frame motion and interframe motion.

From the point of view of mechanics, the motion of a rigid object can also be considered as rotation about a centre \mathbf{O} which translates relative to the camera centre \mathbf{C} . The coordinate system used is still camera-centred. We use \mathbf{O}_n to denote the rotation centre's position and \mathbf{X}_n any point on the object at time n . \mathbf{R}_n and \mathbf{T}_n denote the rotation matrix of the object and the translation velocity of the rotation centre at time n . Then the object-centred motion representation between two consecutive frames is as follows:

$$\begin{aligned} \mathbf{X}_n &= \mathbf{R}_n [\mathbf{X}_{n-1} - \mathbf{O}_{n-1}] + \mathbf{O}_n \\ \mathbf{O}_n &= \mathbf{O}_{n-1} + \mathbf{T}_n = \dots = \mathbf{O}_0 + \sum_{i=1}^n \mathbf{T}_i \end{aligned} \quad (41)$$

Reordering equation (41) as:

$$\mathbf{X}_n - \mathbf{O}_n = \mathbf{R}_n (\mathbf{X}_{n-1} - \mathbf{O}_{n-1}) \quad (42)$$

and using equation (39) we can obtain the following representation of motion between frame i and j ($i > j$):

$$\begin{aligned} \mathbf{X}_i &= \mathbf{R}_{i,j} [\mathbf{X}_j - \mathbf{O}_j] + \mathbf{O}_i \\ &= \mathbf{R}_{i,j} \mathbf{X}_j - \mathbf{R}_{i,j} \left[\mathbf{O}_0 + \sum_{k=1}^j \mathbf{T}_k \right] + \mathbf{O}_0 + \sum_{k=1}^i \mathbf{T}_k \\ &= \mathbf{R}_{i,j} \mathbf{X}_j + [\mathbf{I} - \mathbf{R}_{i,j}] \left[\mathbf{O}_0 + \sum_{k=1}^j \mathbf{T}_k \right] + \sum_{k=j+1}^i \mathbf{T}_k \\ &= \mathbf{R}_{i,j} \mathbf{X}_j + \mathbf{t}_{i,j} \end{aligned} \quad (43)$$

Let us note that the above representation is significantly different from the camera-centred motion representation when the motion involves rotation. However, for translational motion where $\mathbf{R}_n = \mathbf{I}$ for all n , the object-centred representation and the camera-centred representation are identical. In this representation, the initial rotation centre \mathbf{O}_0 , and the motion parameters \mathbf{T}_n and \mathbf{R}_n , $n = 1, 2, 3, \dots$, are the unknowns and to be determined. It is evident that a scale constant is involved in the translation vectors. Also, \mathbf{O}_0 may not always be determined uniquely. For example, when \mathbf{R}_n , $n = 1, 2, 3, \dots$, share the same rotation axis $\mathbf{n}_{\mathbf{R}}$, then if \mathbf{O}_0 is a solution, $\mathbf{O}_0 + \alpha \mathbf{n}_{\mathbf{R}}$ for any constant α , is also a solution, due to the fact that:

$$(\mathbf{I} - \mathbf{R}_{n+i,n})(\mathbf{O}_0 + \alpha \mathbf{n}_{\mathbf{R}}) = (\mathbf{I} - \mathbf{R}_{n+i,n})\mathbf{O}_0 \quad (44)$$

That is, any point on the rotation axis can serve as the rotation centre. However, this uncertainty will not affect our understanding of the motion. Whenever such uncertainty occurs, we can remove it by enforcing the following equation:

$$\mathbf{O}_0 \cdot \mathbf{n}_{\mathbf{R}} = 0 \quad (45)$$

Motion models

To take advantage of a long-sequence of images, it is often assumed that the motion parameters change slowly between frames. We now examine several special cases. The solutions for these cases are presented in the following sections. In the models below, the choice of models for rotation and translation can be independent, for example, the rotation can be constant but the translation may have constant acceleration, and *vice versa*. We first discuss the rotation models and then the translation models.

Rotation models

The models approximate a rotation by a polynomial of increasing order.

1. Constant rotation

In this case, the object rotates about a centre at a

constant velocity, that is $\mathbf{R}_n = \mathbf{R}$ for all n . The rotation matrix $\mathbf{R}_{i,j}$ reduces to the following form:

$$\mathbf{R}_{i,j} = \mathbf{R}^{i-j} \quad (46)$$

\mathbf{R} is expressed in the three-angle representation (10) during solution process and hence only three unknowns are involved.

2. Rotation with constant acceleration

In this case, the rotation velocity changes at a constant rate about a fixed axis, that is:

$$\mathbf{R}_n = \mathbf{nn}^T - (\mathbf{nn}^T - \mathbf{I}) \cos \phi_n + \mathbf{n} \times \mathbf{I} \sin \phi_n \quad (47)$$

with

$$\phi_n = \phi_0 + n\phi_a \quad (48)$$

where \mathbf{n} is the rotation axis, ϕ_n the rotation angle, and \mathbf{I} the identity matrix. Since the rotation axis \mathbf{n} is fixed, we have:

$$\mathbf{R}_{i,j} = \mathbf{nn}^T - (\mathbf{nn}^T - \mathbf{I}) \cos \phi_{i,j} + \mathbf{n} \times \mathbf{I} \sin \phi_{i,j} \quad (49)$$

where:

$$\phi_{i,j} = \sum_{k=j}^i \phi_k = (i-j)\phi_0 + [a_i - a_j]\phi_a \quad (50)$$

and:

$$a_k = \sum_{j=0}^k j = \frac{k(k+1)}{2} \quad (51)$$

For the purpose of estimation, we represent the rotation axis \mathbf{n} by two angles α and β in the form of:

$$\mathbf{n} = \begin{bmatrix} \sin \alpha \cos \beta \\ \cos \alpha \\ \sin \alpha \sin \beta \end{bmatrix} \quad (52)$$

Therefore, α , β , ϕ_0 and ϕ_a are the unknown rotation parameters.

3. Rotation described by second order polynomials

In this case, the rotation velocity between two consecutive frames is approximated by second order polynomials in time, i.e.:

$$\Omega_n = \begin{bmatrix} \omega_X^n \\ \omega_Y^n \\ \omega_Z^n \end{bmatrix} = \Omega_0 + n\Omega_a + n^2\Omega_b \quad (53)$$

and:

$$\mathbf{R}_n = \mathbf{A}_X(\omega_X^n) \mathbf{A}_Y(\omega_Y^n) \mathbf{A}_Z(\omega_Z^n) \quad (54)$$

The rotation between two nonconsecutive frames is still expressed by equation (39). Three vectors Ω_0 , Ω_a and Ω_b are the unknowns to be estimated.

4. Arbitrary rotation model

In this case, no smoothness constraint about the rotation is enforced during the solution. Therefore, all rotation matrices \mathbf{R}_i , $i = 1, 2, \dots, n$, are considered to be independent and unknown. The three-angle representation is used for each rotation matrix, and rotations between two nonconsecutive frames are expressed in the form of equation (39).

Translation models and the corresponding solutions

Similarly, a translation is also approximated by polynomials over time.

1. Constant translation

In this case, the rotation centre translates at a constant speed, i.e. $\mathbf{T}_n = \mathbf{T}$ for all n . Then from equation (43) we have:

$$\mathbf{t}_{i,j} = \mathbf{U}_{i,j}\mathbf{O}_0 + \mathbf{V}_{i,j}\mathbf{T}_0 \quad (55)$$

from which we have:

$$(\mathbf{t}_{i,j} \times \mathbf{U}_{i,j})\mathbf{O}_0 + (\mathbf{t}_{i,j} \times \mathbf{V}_{i,j})\mathbf{T}_0 = 0 \quad (56)$$

where

$$\mathbf{U}_{i,j} = \mathbf{I} - \mathbf{R}_{i,j}, \quad \mathbf{V}_{i,j} = j\mathbf{U}_{i,j} + (i-j)\mathbf{I} \quad (57)$$

2. Translation with constant acceleration

In this case, the rotation centre's translation changes at a constant acceleration, i.e.:

$$\mathbf{T}_n = \mathbf{T}_0 + n\mathbf{T}_a \quad (58)$$

Then, the translation between i th and j th frame ($i > j$) is:

$$\mathbf{t}_{i,j} = \mathbf{U}_{i,j}\mathbf{O}_0 + \mathbf{V}_{i,j}\mathbf{T}_0 + \mathbf{W}_{i,j}\mathbf{T}_a \quad (59)$$

from which we have:

$$(\mathbf{t}_{i,j} \times \mathbf{U}_{i,j})\mathbf{O}_0 + (\mathbf{t}_{i,j} \times \mathbf{V}_{i,j})\mathbf{T}_0 + (\mathbf{t}_{i,j} \times \mathbf{W}_{i,j})\mathbf{T}_a = 0 \quad (60)$$

where:

$$\mathbf{W}_{i,j} = a_j\mathbf{U}_{i,j} + (a_i - a_j)\mathbf{I} \quad (61)$$

and $\mathbf{U}_{i,j}$ and a_i are the same as defined above.

3. Translation described by second order polynomials

In this case, the rotation centre's translation is approximated by a second order polynomial, i.e.:

$$\mathbf{T}_n = \mathbf{T}_0 + n\mathbf{T}_a + n^2\mathbf{T}_b \quad (62)$$

Then, the interframe translation between the i th and j th frame ($i > j$) is:

$$\mathbf{t}_{i,j} = \mathbf{U}_{i,j}\mathbf{O}_0 + \mathbf{V}_{i,j}\mathbf{T}_0 + \mathbf{W}_{i,j}\mathbf{T}_a + \mathbf{Y}_{i,j}\mathbf{T}_b \quad (63)$$

from which we have:

$$(\mathbf{t}_{i,j} \times \mathbf{U}_{i,j}) \mathbf{O}_0 + (\mathbf{t}_{i,j} \times \mathbf{V}_{i,j}) \mathbf{T}_0 + (\mathbf{t}_{i,j} \times \mathbf{W}_{i,j}) \mathbf{T}_a + (\mathbf{t}_{i,j} \times \mathbf{Y}_{i,j}) \mathbf{T}_b = 0 \quad (64)$$

where:

$$\mathbf{Y}_{i,j} = b_j \mathbf{U}_{i,j} + (b_i - b_j) \mathbf{I} \quad (65)$$

$$b_k = \sum_{i=0}^k j^2 = \frac{1}{6} k(k+1)(2k+1) \quad (66)$$

and other variables are the same as defined before.

4. Arbitrary translation model

In this case, no smoothness constraint about the rotation centre's translation is enforced during the solution. Therefore, all translation vectors \mathbf{T}_i , $i = 1, 2, \dots, n$, are considered independent unknowns. Then, the interframe translation between the i th and j th frame ($i > j$) is:

$$\mathbf{t}_{i,j} = \sum_{k=j+1}^i \mathbf{R}_{i,k} \mathbf{t}_k \quad (67)$$

from which we can obtain:

$$\sum_{k=j+1}^i (\mathbf{t}_{i,j} \times \mathbf{R}_{i,k}) \mathbf{t}_k = 0 \quad (68)$$

where we assume $\mathbf{R}_{i,i} = \mathbf{I}$, and $\mathbf{R}_{i,j}$ is represented by equation (39).

Solution using interframe matches

The following algorithm applies to the worst situations in which only interframe matches are present. The solution is based on the motion models presented above.

The algorithm is divided into three steps: (1) first solve for the rotation parameters nonlinearly; (2) solve for the translation parameters in a closed form; (3) estimate the depths using all related correspondence data for each point.

The solutions for the above rotation models are all the same, though different parameters are searched for to minimize the same criterion. Let $\Theta_p^{i,j} = [x_p^{i,j} \ y_p^{i,j} \ 1]^T$, $\Theta_p^{j,i} = [x_p^{j,i} \ y_p^{j,i} \ 1]^T$, $p = 1, 2, \dots, P_{i,j}$, be $P_{i,j}$ pairs of correspondences between the i th and j th frames, where $(x_p^{i,j}, y_p^{i,j})$ and $(x_p^{j,i}, y_p^{j,i})$ denote the image coordinates in the i th and j th frames. Let:

$$\Pi_{i,j} = \begin{bmatrix} (\Theta_1^{i,j} \times \mathbf{R}_{i,j} \Theta_1^{j,i})^T \\ \vdots \\ (\Theta_{P_{i,j}}^{i,j} \times \mathbf{R}_{i,j} \Theta_{P_{i,j}}^{j,i})^T \end{bmatrix} \quad (69)$$

Let $\lambda_{i,j}$ be the least eigenvalue of $\Pi_{i,j}^T \Pi_{i,j}$. The rotation parameters for each model are searched for to minimize:

$$\Lambda = \sum_{i>j} w_{i,j} \lambda_{i,j} \quad (70)$$

Since $\Pi_{i,j}^T \Pi_{i,j}$ is a 3×3 matrix, its eigenvalues can be obtained in a closed form. With good initial guesses from the two-view algorithms, globally optimal solutions are generally obtained for models of second or lower orders.

After rotation parameters are obtained, the translation parameters of the first three models are estimated in a closed form. First $\mathbf{t}_{i,j}$ is obtained for all i and j with $i > j$ using the two-view motion algorithm and the estimated rotation parameters. Then \mathbf{O}_0 and \mathbf{T}_0 (also $\mathbf{T}_a, \mathbf{T}_b$) are solved for from equations (56), (60) or (64) with the linear least squares method. The obtained \mathbf{O}_0 and \mathbf{T}_0 (also $\mathbf{T}_a, \mathbf{T}_b$) are then used to compute $\mathbf{t}_{i,j}$ using again equations (56), (60) or (64). The solution for the arbitrary model is also the same as that for the polynomial models, except that equation (68) instead of (56), (60) or (64) is used.

After the motion is solved, we can integrate all information available about a point to get an integrated solution of the structure about the point. Consider the points in the first frame. Let $\Theta_1 = [x_1, y_1, 1]^T$ be a point in the first frame that has correspondences with points in any other views. Then we track Θ_1 in the image sequence to get its trajectories in a subsequence. Reorder the subsequence into a sequence and assume there are F frames in the sequence for Θ_1 . Therefore, the problem is reduced to estimating the structure with known motions from a point trajectory over a sequence. We now consider the reduced problem.

Assume we have a point trajectory over F frames with the point position in the i th view being $\Theta_i = [x_i, y_i, 1]^T$. Let $\mathbf{R}_{i,j} = (r_{mn}^{i,j})$, $\mathbf{t}_{i,j} = [t_1^{i,j}, t_2^{i,j}, t_3^{i,j}]^T$ ($i > j$) be the known rotation and translation between i th and j th views. We now need to solve for the depth Z_1 of the point at time 1. This is done by searching Z_1 such that the following error:

$$S_4 = \sum_{i=2}^F \left(x_i - \frac{a_i Z_1 + t_1^{i1}}{c_i Z_1 + t_3^{i1}} \right)^2 + \left(y_i - \frac{b_i Z_1 + t_2^{i1}}{c_i Z_1 + t_3^{i1}} \right)^2 \quad (71)$$

is minimized, where:

$$\begin{aligned} a_i &= r_{11}^{i1} x_1 + r_{12}^{i1} y_1 + r_{13}^{i1}, \\ b_i &= r_{21}^{i1} x_1 + r_{22}^{i1} y_1 + r_{23}^{i1}, \\ c_i &= r_{31}^{i1} x_1 + r_{32}^{i1} y_1 + r_{33}^{i1} \end{aligned} \quad (72)$$

S_4 is the sum of squared distances between the observed point positions and the estimated positions assuming a depth value Z_1 . A good initial guess of Z_1 is

obtained in a closed-form by minimizing:

$$S_4 = \sum_{i=2}^F [x_i(c_i Z_1 + t_3^{i1}) - (a_i Z_1 + t_1^{i1})]^2 + [y_i(c_i Z_1 + t_3^{i1}) - (b_i Z_1 + t_2^{i1})]^2 \quad (73)$$

Solution using point trajectories

The solution above needs to consider all interframe motions, and hence the computation complexity is very high. In case point trajectories are available, we can integrate information about the depths in the first frame and the motions between the first and other frames to solve only the depths and motions related to the first frame. This is the best situation for motion and structure estimation in the sense that an integrated solution of motion and structure can be obtained simultaneously. Requiring both motion consistence and structure consistence places a stronger constraint on the motion and structure parameters than requiring only motion consistence, as in the previous method. Therefore, the accuracy of the motion and structure parameters in this method should be much better if the number of trajectories is comparable to the number of interframe correspondences.

Nonlinear solution

If the depths and motion parameters are considered simultaneously, the number of unknowns can be too large to handle. Therefore, we need a formulation which involves both structure and motion but allows stepwise solution. The following formulation satisfies this purpose.

Let $\mathbf{M}_i = \mathbf{R}_{i1}$ be the rotation and $\mathbf{V}_i = \mathbf{t}_{i1}$ the translation between the i th and the first views. Let $\Theta_p^f = [x_p^f, y_p^f, 1]^T$ be the position of the p th point in the f th image. Let $\mathbf{X}_p^f = Z_p^f \Theta_p^f$, where Z_p^f is the depth. Then, the following motion equation holds:

$$Z_p^f \Theta_p^f = Z_p^1 \mathbf{M}_f \Theta_p^1 + \mathbf{V}_f, \quad f = 2, \dots, F \quad (74)$$

Equation (74) describes the motion equation between the f th frame and the first frame for point p .

Consider equation (74) for $p = i$ and $p = j$. Subtracting the equation for i from the equation for j , we have:

$$Z_i^f \Theta_i^f - Z_j^f \Theta_j^f = \mathbf{M}_f \mathbf{S}_{i,j}, \quad i \neq j, \quad i, j = 1, \dots, P \quad f = 2, \dots, F \quad (75)$$

where:

$$\mathbf{S}_{ij} = Z_i^1 \Theta_i^1 - Z_j^1 \Theta_j^1 \quad (76)$$

is the vector connecting the i th point and the j th point. Equation (75) indicates that vectors $Z_i^f \Theta_i^f$, $Z_j^f \Theta_j^f$, and $\mathbf{M}_f \mathbf{S}_{ij}$ must be coplanar. Therefore we have:

$$(\Theta_i^f \times \Theta_j^f) \cdot \mathbf{M}_f \mathbf{S}_{ij} = 0, \quad i \neq j, \quad i, j = 1, \dots, P, \quad f = 2, \dots, F \quad (77)$$

Equations (76) and (77) yield the following equation:

$$\begin{bmatrix} (\Theta_i^1 \times \Theta_j^1)^T \\ (\Theta_i^2 \times \Theta_j^2)^T \mathbf{M}_2 \\ \vdots \\ (\Theta_i^F \times \Theta_j^F)^T \mathbf{M}_F \end{bmatrix} \mathbf{S}_{ij} \equiv \Phi_{ij} \mathbf{S}_{ij} = 0, \quad i \neq j, \quad i, j = 1, \dots, P \quad (78)$$

The solution of the motion and structure parameters is based on the above equation.

Clearly, for each i and j , to make the structure \mathbf{S}_{ij} have a consistent solution, we must have:

$$\text{rank}(\Phi_{ij}) = 2 \quad (79)$$

However, for noisy data, it is possible that no rotation parameters \mathbf{M}_f , $f = 2, \dots, F$, can make every Φ_{ij} have a rank of 2. We now consider a two-step least-squares solution. For any given Φ_{ij} , the optimal solution of \mathbf{S}_{ij} subject to $\|\mathbf{S}_{ij}\| = 1$ that minimizes:

$$\epsilon_{ij} = \|\Phi_{ij} \mathbf{S}_{ij}\|^2 = \mathbf{S}_{ij}^T \Phi_{ij}^T \Phi_{ij} \mathbf{S}_{ij} \quad (80)$$

is the eigenvector of $\Phi_{ij}^T \Phi_{ij}$ associated with $\Phi_{ij}^T \Phi_{ij}$'s least eigenvalue λ_{ij} , which is the minimum value of ϵ_{ij} for a given Φ_{ij} . Now, the problem that remains is to estimate the rotation parameters and minimize λ_{ij} . However, to solve for \mathbf{M}_f , $f = 2, \dots, F$, uniquely, we need to consider all the points simultaneously. Let:

$$S_5 = \sum_{i=1}^P \sum_{j=1}^{P-1} w_{ij} \lambda_{ij} = \sum_{i>j} w_{ij} \lambda_{ij} \quad (81)$$

where λ_{ij} is the least eigenvalue of $\Phi_{ij}^T \Phi_{ij}$ and w_{ij} a weighting factor. Then, the optimal solution of \mathbf{M}_f , $f = 2, \dots, F$, is that which minimizes S_5 defined above.

In general, an optimal solution of \mathbf{M}_f , $f = 2, \dots, F$, that minimizes S_5 requires a nonlinear search of the rotation parameters. If the polynomial rotation models are used, many fewer unknowns will be involved for long-sequence motion than when the arbitrary motion model is used.

After \mathbf{M}_f , $f = 2, \dots, F$, have been found, the structure vectors \mathbf{S}_{ij} , $i > j$, can all be determined to within a scalar and chosen as the eigenvector of $\Phi_{ij}^T \Phi_{ij}$ associated with the least eigenvalue. Let the scaled solution be \mathbf{S}_{ij} be $\hat{\mathbf{S}}_{ij}$. Then, because of equation (76) we have the following equation:

$$Z_i^1 \Theta_i^1 - Z_j^1 \Theta_j^1 = \alpha_{ij} \hat{\mathbf{S}}_{ij} \quad (82)$$

where α_{ij} is some constant to be determined. Using $\hat{\mathbf{S}}_{ij}$ to cross-multiply both sides of equation (82) we obtain the following equation for the depths:

$$Z_i^1 (\Theta_i^1 \times \hat{\mathbf{S}}_{ij}) - Z_j^1 (\Theta_j^1 \times \hat{\mathbf{S}}_{ij}) = 0, \quad i > j, \quad i, j = 1, \dots, P \quad (83)$$

The above equation gives $3P(P-1)/2$ linear equations for the P unknown depths. We can reorganize equation (83) into the following matrix form:

$$\begin{bmatrix} \beta_{21} & \gamma_{21} & 0 & \cdots & 0 \\ \beta_{31} & 0 & \gamma_{31} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{p1} & 0 & 0 & \cdots & \gamma_{p1} \\ 0 & \beta_{32} & \gamma_{32} & \cdots & 0 \\ 0 & \beta_{42} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \beta_{p2} & 0 & \cdots & \gamma_{p2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \beta_{p,p-1} & \gamma_{p,p-1} \end{bmatrix}_{3P(P-1)(2) \times P} \begin{bmatrix} Z_1^1 \\ Z_2^1 \\ \vdots \\ Z_p^1 \end{bmatrix} = \mathbf{H}\mathbf{Z} = 0 \tag{84}$$

where:

$$\beta_{ij} = -\Theta_j^1 \times \hat{\mathbf{S}}_{ij}, \quad \gamma_{ij} = \Theta_i^1 \times \hat{\mathbf{S}}_{ij}, \tag{85}$$

$i > j, \quad i, j = 1, \dots, P$

The depths can be determined to within a scalar from the above equation using the least-squares method.

After the rotations and depths are known, there are many ways of determining the translation vectors. For example, the two-view motion algorithm discussed earlier can be used to determine the direction of \mathbf{V}_f after \mathbf{M}_f is known. However, to determine the relative scales of $\mathbf{V}_f, f = 2, \dots, F$, the constancy of object size still needs to be enforced. Therefore, we present a different algorithm which uses the depths in other views. Let \mathbf{S}_{ij} be computed as in equation (76), and let $\mathbf{X}_{ij}^f = \mathbf{M}_f \mathbf{S}_{ij}$. Then equation (75) gives:

$$Z_i^f \Theta_i^f - Z_j^f \Theta_j^f = \mathbf{X}_{ij}^f, \quad i, j = 1, \dots, P \tag{86}$$

Using Θ_j^f to take the cross-product with both sides of the above equation, we obtain the following equation for Z_i^f :

$$Z_i^f (\Theta_i^f \times \Theta_j^f) = \mathbf{X}_{ij}^f \times \Theta_j^f, \quad j = 1, \dots, P \tag{87}$$

Depth Z_i^f can be solved for linearly from the above equation by a least-squares method minimizing

$$\sum_{j=1}^P \| Z_i^f (\Theta_i^f \times \Theta_j^f) - \mathbf{X}_{ij}^f \times \Theta_j^f \|^2.$$

The solution is given in closed form by:

$$Z_i^f = \frac{\mathbf{A}_i^f \cdot \mathbf{B}_i^f}{\mathbf{A}_i^f \cdot \mathbf{A}_i^f}, \quad i = 1, \dots, P, \quad f = 2, \dots, F \tag{88}$$

where

$$\mathbf{A}_i^f = \begin{bmatrix} \Theta_i^f \times \Theta_1^f \\ \vdots \\ \Theta_i^f \times \Theta_p^f \end{bmatrix}, \quad \mathbf{B}_i^f = \begin{bmatrix} \mathbf{X}_{i1}^f \times \Theta_1^f \\ \vdots \\ \mathbf{X}_{ip}^f \times \Theta_p^f \end{bmatrix}. \tag{89}$$

After the depths are all obtained, the optimal solution of translation vector \mathbf{V}_f that minimizes

$$\sum_{p=1}^P \| Z_p^f \Theta_p^f - Z_p^1 \mathbf{M}_f \Theta_p^1 - \mathbf{V}_f \|^2$$

is then given by:

$$\mathbf{V}_f = \frac{1}{P} \sum_{p=1}^P (Z_p^f \Theta_p^f - Z_p^1 \mathbf{M}_f \Theta_p^1) \tag{90}$$

Since translations can be solved in closed form with well-defined optimality criterion after the rotations are known, in general, there is not much need to use the translation models if rotations are estimated with good accuracy. However, if a suitable model applies, the results can still be improved by applying the estimates of $\mathbf{V}_f = \mathbf{t}_{f1}$ obtained as above to the model-based estimation method for translation vectors discussed in the last section.

Iterative solution for arbitrary motion model

In the algorithm discussed above, if the motion cannot be described by a low-order polynomial, then an arbitrary motion model has to be used. Therefore, a robust and efficient algorithm for solving arbitrary motion is desired. Here, we describe an alternative algorithm which solves motion and structure iteratively.

As has been shown above, if the rotation are known, the depths and translations can be solved in closed form. On the other hand, if the depths and translations are known, the rotations can be solved in closed form as follows.

Let:

$$\mathbf{S}_{ij}^f = Z_i^f \Theta_i^f - Z_j^f \Theta_j^f \tag{91}$$

From equation (75) we have:

$$\mathbf{S}_{ij}^f = \mathbf{M}_f \mathbf{S}_{ij}, \quad i \neq j, \quad i, j = 1, \dots, P \tag{92}$$

If \mathbf{S}_{ij}^f and $\mathbf{S}_{ij}, p = 1, \dots, P$, are all known, \mathbf{M}_f can be determined uniquely using the methods of Faugeras and Hebert⁴⁰ or Arun *et al.*⁴¹. Or, \mathbf{M}_f can be searched around the previous estimate to minimize:

$$\sum_{i>j} \{ (\Theta_i^f \times \Theta_j^f) \cdot \mathbf{M}_f \mathbf{S}_{ij} \}^2 \tag{93}$$

Since only three unknowns are involved and the earlier estimate of \mathbf{M}_f can be used as an initial guess, any optimization method should work; \mathbf{M}_f is solved for independently for each f .

After $\mathbf{M}_f, f = 2, \dots, F$, have been obtained, $\mathbf{S}_{ij}, p = 1, \dots, P$, can be updated using equation (78) and then refined using equation (76) after the depths are computed from (84). Therefore, the iteration algorithm consists of two iteration steps:

1. Each rotation matrix $\mathbf{M}_f, f = 2, \dots, F$, is updated by solving equation (92) or by minimizing

$$\sum_{i>j} \{ (\Theta_i^f \times \Theta_j^f) \cdot \mathbf{M}_f \mathbf{S}_{ij} \}^2,$$

using the current estimates of \mathbf{S}_{ij} (and \mathbf{S}_{ij}^f , if needed), $i > j$, $i, j = 1, \dots, \mathbf{P}$.

- Each shape vector \mathbf{S}_{ij} is first updated by the eigenvector of $\Phi_{ij}^T \Phi_{ij}$ defined in (78) associated with the least eigenvalue of $\Phi_{ij}^T \Phi_{ij}$. The depths are computed from equation (84) using the updated shape vectors \mathbf{S}_{ij} , $i > j$. Finally, \mathbf{S}_{ij} is recomputed using equation (76).

The algorithm iteratively repeats the above two steps until it converges to a solution at which S_5 has a minimum value. The initial estimates of \mathbf{M}_f , $f = 2, \dots, F$, can be obtained using the two-view motion algorithm discussed earlier.

Automatic selection of models

Since models of different orders use the same optimality criterion, in general, a higher-order model gives a smaller objective function value, but the correct model gives the best estimation results. For real applications, it is important to select the correct model for the given data. We implemented the following method which does not require any heuristic information about motion, but only the correspondence data. Here we consider only the algorithm using interframe matches.

In the above models, the polynomial models terminate at the second order. Any third- or higher-order polynomial models can be similarly built with increasing computation complexity. However, our experience shows that the solutions of third order or higher order models are not robust, and in general a globally optimal solution is not guaranteed. Therefore, for a very long sequence of images whose motion cannot be approximated by a second-order polynomial, it is better to divide the sequence into subsequences so that second- or lower-order models can be applied. Therefore, in the following discussion about how to automatically determine the correct model, we are concerned with only the first three models (polynomials of up to the second order) for both rotation and translation.

There seems to be no efficient way except to apply all models to the same data and see what happens. In general, a model of higher order needs a larger number of views to stabilize the solution. For a sequence of a few images, probably only low-order models apply. In the following, we assume that a sufficient number (e.g. 10) of frames is available so that all models can be used.

Consider the rotation parameters first. In general, the two consecutive models give comparable results about lower-order parameters. Then a decision is made about which model is the right one according to the results. This is done as follows:

- If the minimum objective function value Λ_1 obtained by the first model is smaller than those obtained by the second and third models, then rotation is constant and the first model is used.
- Otherwise, obtain the rotation vector $\Omega^n = (\omega_x^n \ \omega_y^n \ \omega_z^n)^T$ for each time instant n for the third and the second models. If the distance between the two solutions at any time instant is

larger than a threshold (e.g. 0.3°) determined according to the system resolution, then the third model is used.

- Else, obtain the rotation vector Ω^n for each time instant n for the second and the first models. If the distance between the two solutions at any time instant is larger than a threshold (e.g. 0.3°), then the second model is used. Otherwise, the first model is used.

A somewhat different procedure is used for determining the translation model. The numerical behaviour of the solutions for translation parameters is much worse than that of rotation parameters. Therefore, for small translation a wrong selection of the translation models may occur. Fortunately, the direction of the translation is still quite accurate, even if a wrong model is chosen. The procedure is listed formally as follows:

- First the translation parameters estimated from different models are properly scaled so that the magnitudes of the solutions are comparable.
- At any time instant n , the interframe translation vector $\mathbf{t}_{i,i+1}$ for solutions from the third and the second models are obtained. If the two vectors differ in direction by more than a threshold (e.g. 5°), or if the two vectors differ in magnitude by larger than another threshold (e.g. 10%), then the third model is used.
- Otherwise, at any time instant n the interframe translation vector $\mathbf{t}_{i,i+1}$ for solutions from the second and the first models are obtained. Again, if the two vectors differ in direction by more than a threshold (e.g. 5°), or if the two vectors differ in magnitude by more than another threshold (e.g. 10%), then the second model is used. Otherwise, the first model is used.

Using the above methods, the algorithm can automatically select the correct model to solve for the motion parameters. It is worth noting that the algorithm is so robust that even if a mistake* is made about the model, the resulting estimation is still acceptable.

Uniqueness condition

For two-view motion there are many theoretical results about the uniqueness of the solution. However, for multiview motion, it is very difficult to analyse the uniqueness problem, because there exist too many situations and unknowns. As pointed out earlier²², in the noisy case, the uniqueness problem of motion solution is often changed to the uniqueness problem of the global optimum of a given criterion. Therefore, for the algorithm using interframe matches, if for a given model, the globally optimal solution minimizing Λ is unique, then the motion is uniquely determined. Otherwise, the algorithm yields only one of the solutions. Fortunately, even for two-view motion, when six or more correspondences are available, there are, in

*In case of noise, the algorithm tends to choose a polynomial of higher degree. This mistake is not serious and makes the estimates only somewhat less accurate.

general, at most two solutions. Therefore, the multi-view motion can, in general, be uniquely determined, since multiview motion is more determined than two-view motion. Thus, as long as six or more correspondences are available for any two consecutive views then, practically there is no need to worry about the multisolution problem.

The algorithm using point trajectories, in general, gives a unique solution of motion and structure when four point trajectories over at least three views, with the points noncolinear in space, are available. A necessary and sufficient condition in general cannot be expressed in closed form. However, if the solution is uniquely defined by the image data, the algorithm will generally give the correct solution. If multiple solutions are defined, then the algorithm will give one of the solutions. It can be shown⁴² that four is the minimum number of point trajectories over three views from which a unique solution can be obtained. Therefore, the algorithm above requires the least stringent conditions for a unique solution of motion and structure.

EXPERIMENTAL RESULTS

In this section we present some experimental results with real image data using the algorithm that uses interframe matches. Experimental results with real image data for the algorithm using trajectories are still not available because we have not obtained a robust trajectory-finding algorithm. The images are acquired by the University of Illinois active vision system which can yield required motion and capture a motion sequence. The images are taken with a Cohu solid state camera having wide angle lens (Vicon V10-100M). The maximum visual field of the camera is about 50°. Cameras of such wide angle must be carefully calibrated. A method for calibrating wide angle lens distortion is presented in Appendix A.

In the examples provided below, only the ground truth for rotation angles is accurately recorded. The accuracy of the rotation angles is 0.001° for pan and 0.002° for tilt determined by the step motors. As we will see soon, the accuracy of the estimates of rotation angles for short range scenes is comparable. Because of the difficulty in measuring the direction of the camera's optical axis and the position of rotation centre relative to the position of optical centre, we are not able to measure the rotation axis and translation direction accurately. It is therefore meaningless to present the reference ground truth for the translation directions since the *ground truth* may contain larger errors than the estimates do. From our experience with simulation data, the estimation of rotation axis is usually more reliable than that of rotation angle. The same is true for translation direction if translation causes large enough changes in the images. However, we can still get a sense of the accuracy of the algorithm from these rough measurements.

In the experiments below, the weighting factor w_{ij} is chosen as the inverse of the number $N_{i,j}$ of correspondences between the i th and j th views. If $N_{i,j} = 0$, then $w_{i,j}$ is set to zero.

The first example contains a sequence of 15 images. *Figures 1a* and *b* show the first and last images and the correspondences obtained between the two images. The white points on a dark background and the black points on a bright background are the matched feature points. *Figures 1c* and *d* show the eighth and ninth images and the correspondences. The motion involves a constant rotation of 0.55° per frame around the Y axis ($[0 \ 1 \ 0]^T$) and a translation along the Z axis ($[0 \ 0 \ 1]^T$) with parameters:

$$t_X^0 = 1.0, \quad t_X^a = 0, \quad t_X^b = 0.015 \quad (94)$$

The estimated rotation parameters are:

$$\mathbf{n} = [0.004216 \quad 0.999967 \quad -0.006964], \\ \phi_0 = 0.537975^\circ \quad (95)$$

Since the ground truth for the rotation centre and translation parameters is not accurate, we provide here only some estimated interframe translation vectors. The algorithm automatically approximated the translation with the constant acceleration model. The normalized translation vector between the first and the second views is:

$$\mathbf{t}_{1,2} = [0.643008 \quad 0.034012 \quad 0.765104] \quad (96)$$

which is inaccurate. However, the translation direction improves from the third view. For example, the normalized translation vector between the third and the fourth views is:

$$\mathbf{t}_{3,4} = [0.185518 \quad 0.010620 \quad 0.982584] \quad (97)$$

and the normalized translation vector between the fourteen and the last views is:

$$\mathbf{t}_{14,15} = [0.117105 \quad 0.001030 \quad 0.993119] \quad (98)$$

The small translation along the X axis caused by rotation cannot be accurately measured. Therefore, we do not know how accurate the translation directions are. However, we can compare the above results with those obtained with the two-view motion algorithm or arbitrary motion model. Here we give a few examples. The two-view motion algorithm gives the following results: the rotation between the first and the second views is:

$$\mathbf{n} = [0.022721 \quad 0.998677 \quad -0.046138], \\ \phi = 0.609610^\circ \quad (99)$$

and the rotation between the fourteenth and the last views is:

$$\mathbf{n} = [0.377904 \quad 0.672219 \quad -0.636640], \\ \phi_0 = 0.478300^\circ \quad (100)$$

The arbitrary motion model gives the following results:

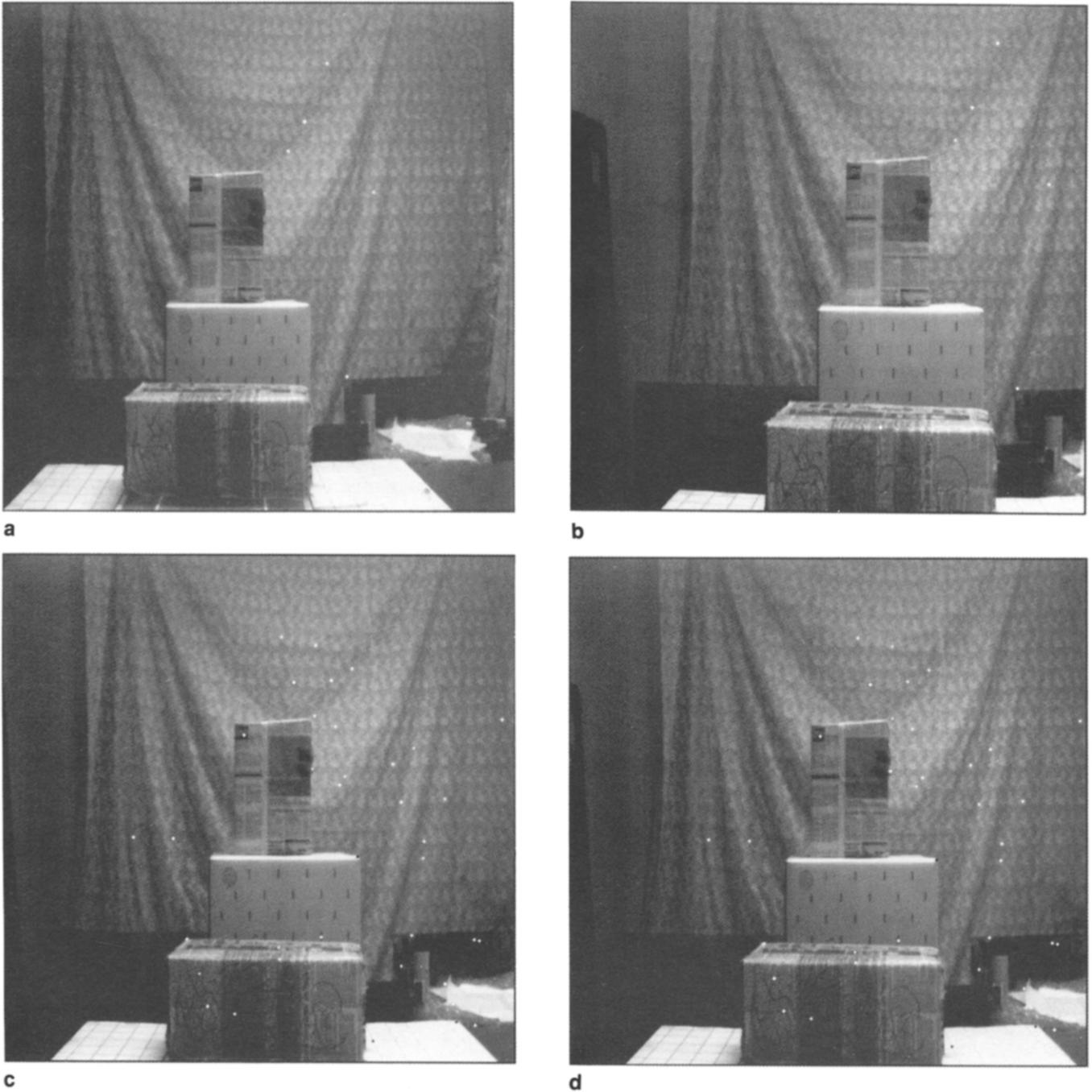


Figure 1 Example 1. Some images and the correspondences obtained. (a) (b) first and last images and the correspondences obtained between them; (c) (d) eighth and ninth images and the correspondences obtained between them

the rotation between the first and the second views is:

$$\mathbf{n} = [0.591695 \quad 0.791808 \quad 0.151541],$$

$$\phi_0 = 0.556398^\circ (101)$$

and the rotation between the fourteenth and the last views is:

$$\mathbf{n} = [0.296980 \quad 0.831835 \quad 0.468886],$$

$$\phi_0 = 0.520983^\circ \quad (102)$$

It is obvious that the model-based motion algorithm yields much better results.

The second example contains a sequence of 20 images. *Figures 2a* and *b* show the first and last images. *Figures 2c* and *d* show the tenth and eleventh images and the correspondences. The motion involves a rotation of second order polynomial around the X axis ($[1 \ 0 \ 0]^T$) with:

$$\omega_X^0 = 0.42^\circ, \quad \omega_X^a = -0.08^\circ, \quad \omega_X^b = 0.008^\circ \quad (103)$$

and a translation of constant acceleration along the Z axis ($[0 \ 0 \ 1]^T$) with parameters:

$$t_Z^0 = 1.0, \quad t_Z^a = -0.06, \quad t_Z^b = 0.01 \quad (104)$$

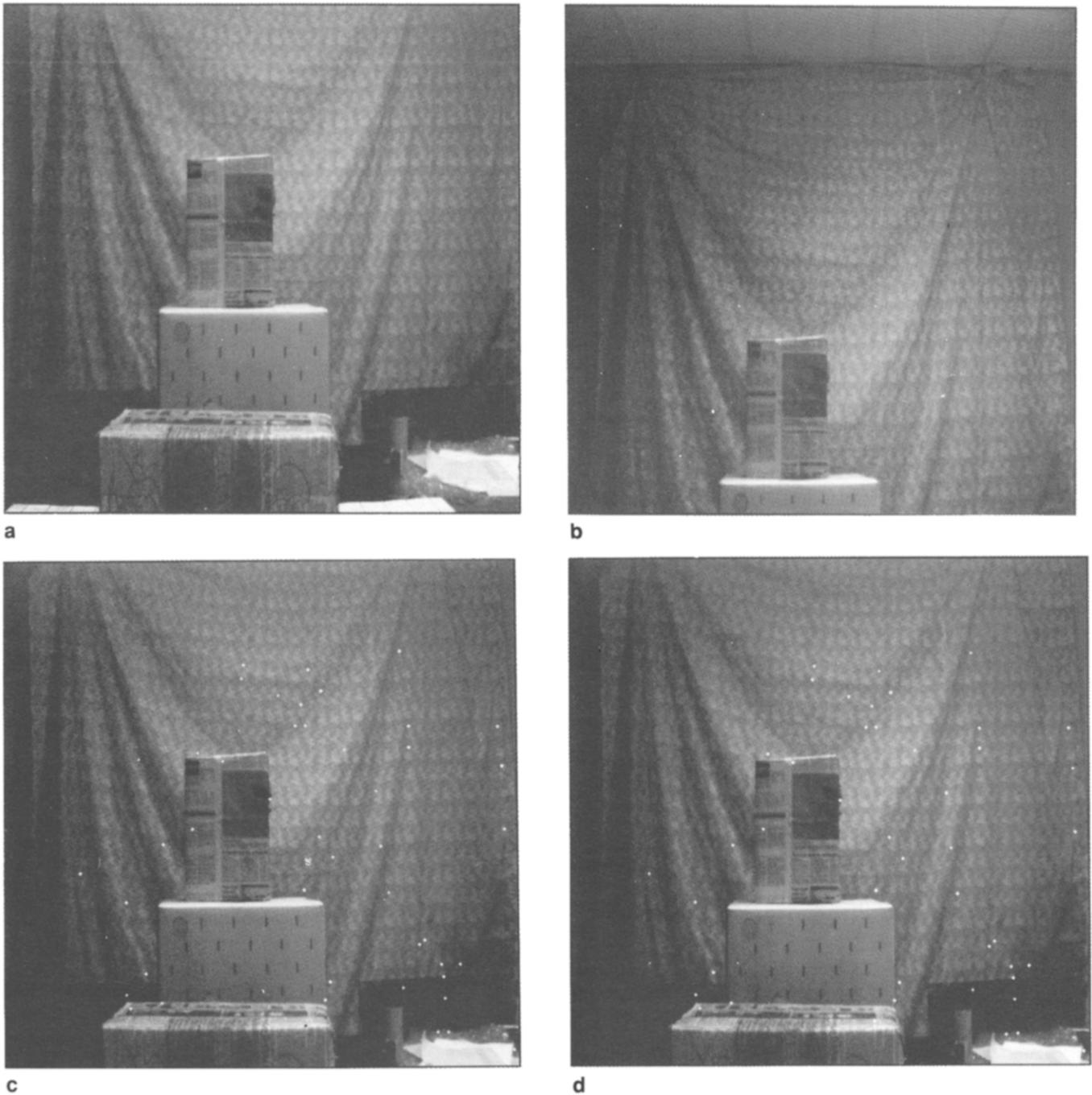


Figure 2 Example 2. Some images and the correspondences obtained. (a) (b) first and last images; (c) (d) tenth and eleventh images and the correspondences obtained between them

The rotation also causes a small translation along the Y axis since the rotation centre is not at the optical centre. The estimated rotation parameters are:

$$\Omega_0 = \begin{bmatrix} 0.426033 \\ -0.015811 \\ 0.022964 \end{bmatrix}, \quad \Omega_a = \begin{bmatrix} -0.081521 \\ 0.004798 \\ -0.004323 \end{bmatrix},$$

$$\Omega_b = \begin{bmatrix} 0.008151 \\ -0.000314 \\ 0.000300 \end{bmatrix} \quad (105)$$

and the interframe translation direction vectors are, for example:

$$\mathbf{t}_{1,2} = [0.034507 \quad 0.066768 \quad 0.997172] \quad (106)$$

$$\mathbf{t}_{10,11} = [0.027467 \quad 0.123701 \quad 0.991939] \quad (107)$$

$$\mathbf{t}_{19,20} = [0.008584 \quad 0.177239 \quad 0.984130] \quad (108)$$

Only two interframe translations are not accurate, which are:

$$\mathbf{t}_{7,8} = [0.342731 \quad 0.311376 \quad 0.886330] \quad (109)$$

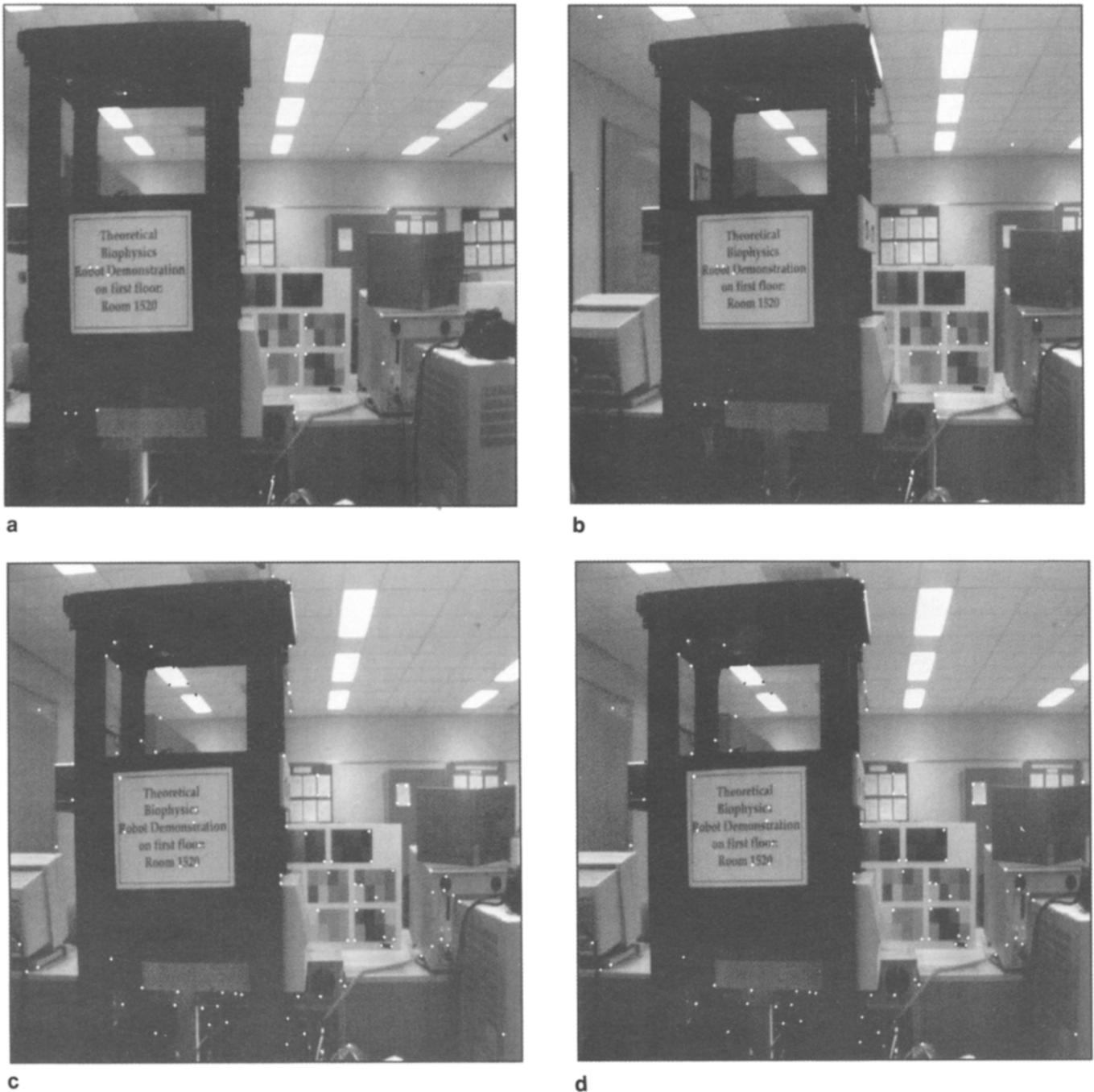


Figure 3 Example 3. Some images and the correspondences obtained. (a) (b) first and last images and the correspondences obtained between them; (c) (d) tenth and eleventh images and the correspondences obtained between them

and:

$$\mathbf{t}_{17,18} = [-0.259094 \quad -0.682176 \quad 0.683744] \quad (110)$$

The estimate of interframe translation vectors is more vulnerable to noise in the image data. We are finding methods to improve the accuracy of estimation of translation.

The third example also contains a sequence of 20 images. *Figures 3a* and *b* show the first and last images and the correspondences obtained between the two images. *Figures 3c* and *d* show the tenth and eleventh images and the correspondences. The motion involves a rotation of second order polynomial around the Y

axis ($[0 \ 1 \ 0]^T$) with:

$$\omega_Y^0 = 0.5^\circ, \quad \omega_Y^a = 0.04^\circ, \quad \omega_Y^b = -0.03^\circ \quad (111)$$

and a translation of constant acceleration along X axis ($[1 \ 0 \ 0]^T$) with parameters:

$$t_X^0 = 1.0, \quad t_X^a = -0.04286 \quad (112)$$

However, in this example, the rotation causes a translation along the Z axis that is comparable with the translation along the X axis. Therefore, we do not know the ground truth of the translation directions.

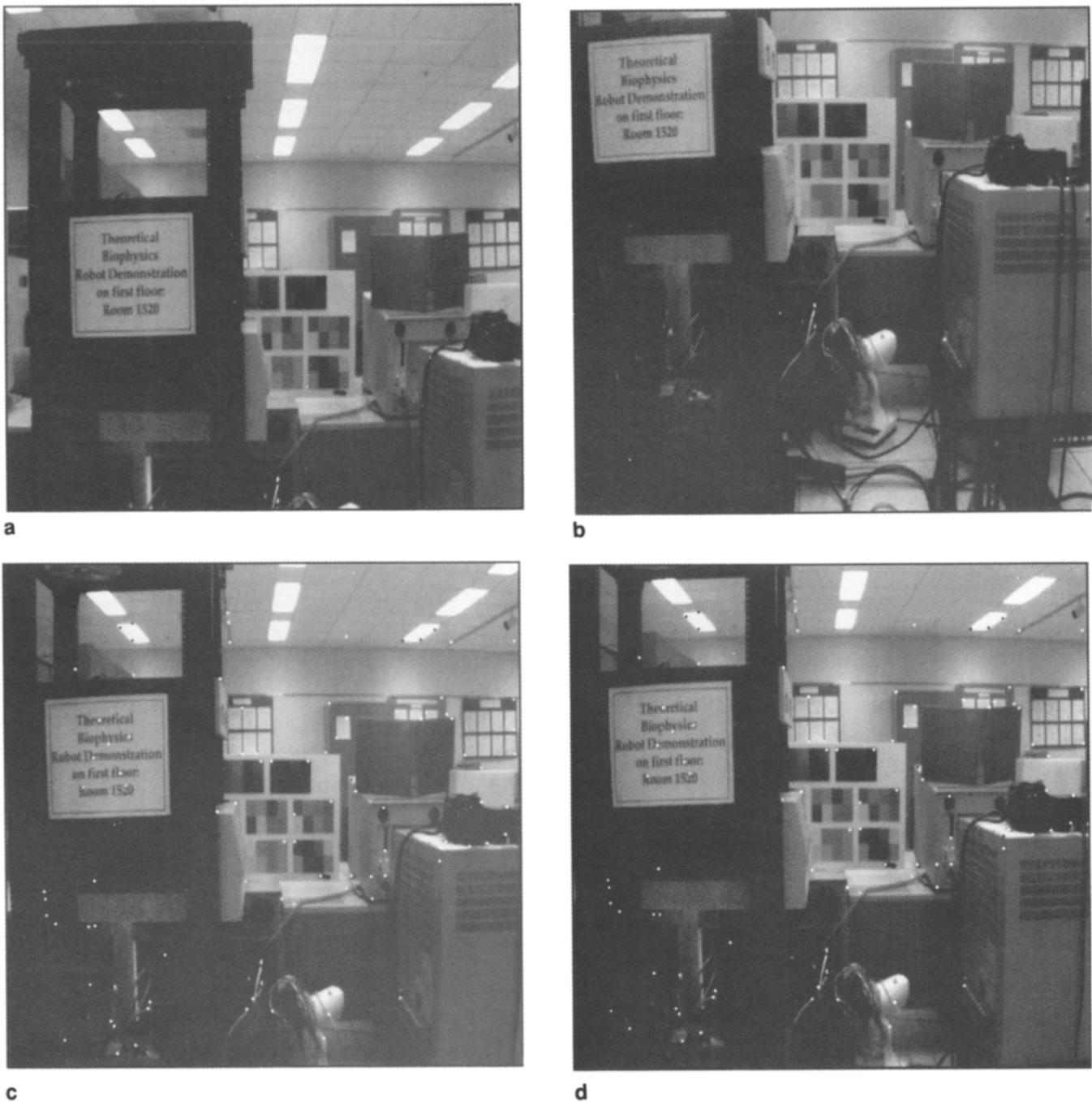


Figure 4 Example 4. Some images and the correspondences obtained. (a) and (b) first and last images; (c) (d) eighth and ninth images and the correspondences obtained between them

The estimated rotation parameters are:

$$\Omega_0 = \begin{bmatrix} 0.019174 \\ 0.370683 \\ 0.000170 \end{bmatrix}, \quad \Omega_a = \begin{bmatrix} -0.004321 \\ 0.038607 \\ -0.002828 \end{bmatrix},$$

$$\Omega_b = \begin{bmatrix} 0.017744 \\ -0.002894 \\ 0.000046 \end{bmatrix} \quad (113)$$

The estimates of the interframe translations are omitted here for this example.

The accuracy of this and the next example is worse than the first two examples. The main reason is probably due to the fact that the objects in this and the next example are further away, and the translation contains a component along the X axis which can be easily confused with rotation. However, for most time instants, the estimated rotation parameters are very close to the ground truth.

In the fourth example, 15 images are taken by a moving camera. *Figures 4a* and *b* show the first and last images. No correspondences are found between these two images because of the large motion between them. *Figures 4c* and *d* show the eighth and ninth images and

the correspondences. The motion involves a rotation (tilt) with constant acceleration around the X axis ($[1\ 0\ 0]^T$) with $\phi_0 = 0.6^\circ$ per frame and $\phi_a = 0.04^\circ$ per frame, and a constant translation along the X axis ($[1\ 0\ 0]^T$). The rotation also causes a translation along Z and Y axes, the amount of which is not accurately measured. The estimated rotation parameters are:

$$\mathbf{n} = [0.970392 \quad -0.197587 \quad 0.138917] \quad (114)$$

$$\phi_0 = 0.637189^\circ, \quad \phi_a = 0.041694^\circ \quad (115)$$

Due to image noise, the algorithm chooses a second order polynomial to estimate the translation parameters. Here we present a few interframe translation direction vectors:

$$\mathbf{t}_{1,2} = [0.416507 \quad -0.741442 \quad -0.526104] \quad (116)$$

$$\mathbf{t}_{7,8} = [0.520486 \quad -0.782628 \quad -341450] \quad (117)$$

and:

$$\mathbf{t}_{14,15} = [0.649293 \quad -0.758120 \quad 0.060604] \quad (118)$$

Clearly, the translation vectors in this example are much less accurate than those in the first two examples. We are currently developing methods to evaluate and improve this accuracy.

SUMMARY

In this paper, we have presented a long sequence motion algorithm using the most appropriate of a set of motion models and a nonlinear two-view motion algorithm for estimating motion parameters from a

long sequence of images. These algorithms utilize the state-of-the-art techniques developed by many motion researchers and are very easy to use. The whole process, from feature detection to matching and then to motion estimation, is fully automated. The application of the algorithms to real image data has yielded good results, from which we can conclude that model-based methods yield much better results than those assuming arbitrary motions.

The algorithms in this paper can use either interframe matches or point trajectory points as input. Because the number of structure parameters does not increase with the number of image frames, trajectories contain much more information than interframe correspondences. Therefore, the accuracy of motion algorithms should be better if trajectories of points are used. Nevertheless, the results obtained using interframe matches over no more than 20 image frames are good enough for many applications.

APPENDIX A: CALIBRATING A WIDE-ANGLE LENS

In this appendix we present a simple method for calibrating wide angle lens cameras. This method is based on the assumption that the distortion is symmetric around the optical centre. *Figure Aa* is a grid image taken with an uncalibrated camera, where we see straight lines are distorted into curves. The problem of calibration is to restore the curves to straight lines. To do this, we use a *stretching function*, defined below, to stretch the warped lines:

$$r' = r(1 + \alpha r + \beta r^2) \quad (A1)$$

where r is the distance of a point to the optical centre (x_0, y_0) before calibration, and r' is the calibrated

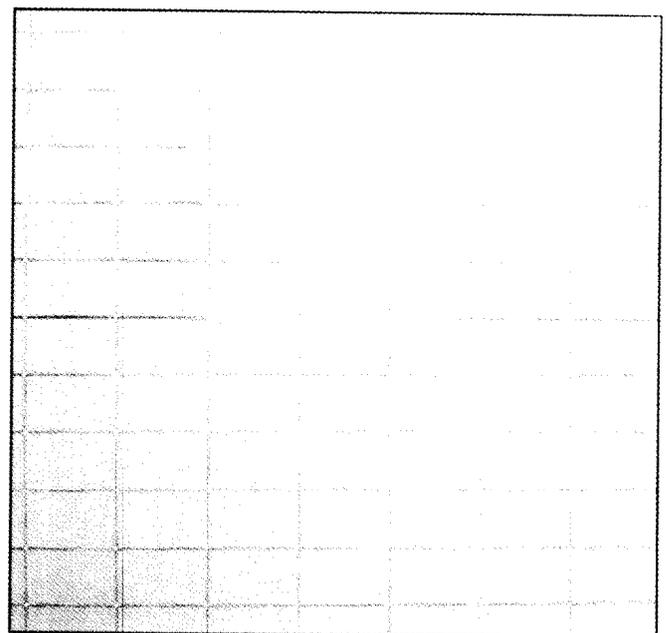
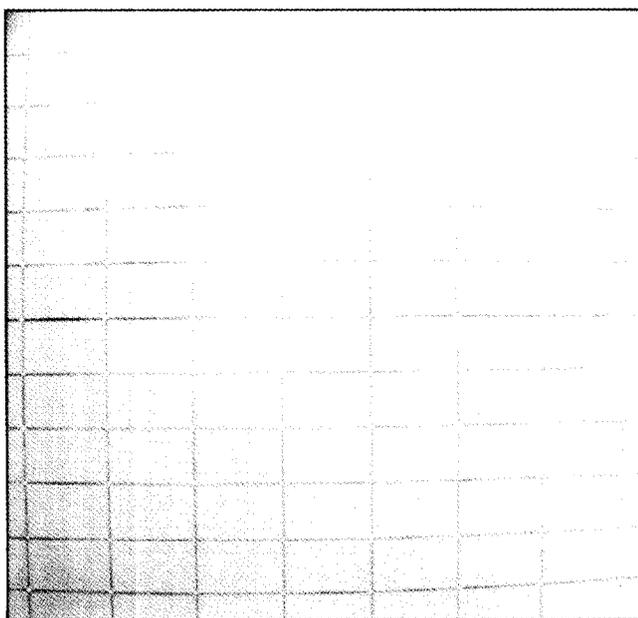


Figure A1 Calibration of a wide angle lens camera. (a) Grid image of straight lines taken with uncalibrated camera; (b) same image after calibration

distance. We need to optimize four unknowns, α , β , x_0 and y_0 such that the warped lines become mostly straight. The objective function used in the optimization process follows. Assume there are N horizontal and M vertical lines.

First, the $N \times M$ intersection points of the lines in the uncalibrated image are identified. Then, the positions of the grid points on each horizontal line are fitted by a quadratic function:

$$y = y_h + a_1x + a_2x^2 \tag{A2}$$

where x is the horizontal coordinate of a point and y the vertical coordinate. y_h , a_1 and a_2 are optimized for each horizontal line. Similarly, a quadratic function of the form:

$$x = x_v + b_1y + b_2y^2 \tag{A3}$$

is used to fit grid points on vertical lines. The $N \times M$ intersection points of the quadratic curves are the refined positions of the grid points and are to be used for calibrating the parameters α , β , x_0 and y_0 .

For each choice of α , β , x_0 and y_0 we find the least squares line fit to the refined grid points on horizontal or vertical curves. That is, for each curve (horizontal or vertical), we find a line:

$$[l_1 \ l_2 \ l_3][x \ y \ 1]^T = 0 \tag{A4}$$

such that the sum of its squared distances to the grid points related to that curve is minimized. The optimal solution of α , β , x_0 and y_0 is that which minimizes the $N + M$ sums of squared distances.

Figure A1b is the calibrated image of Figure A1a, where we see that the lines are almost straight. This simple method is easy to apply and gives reasonable accuracy.

It is unfortunate that symmetry assumption is not true for all cases. Therefore, the results would be better if the image is divided into regions and the calibration is done piecewise. But this requires a more complicated method and more parameters to search for so that a globally optimal estimation of the parameters is often impossible. For the problem of motion estimation, we find that as long as a wide angle lens camera is used, the estimation accuracy is not so sensitive to the camera parameters. Therefore, the calibration method presented here should satisfy many purposes.

APPENDIX B: LINEAR TWO-VIEW MOTION ALGORITHMS

Although linear solutions cannot be superior to nonlinear solutions, linear algorithms may give a very good starting point for nonlinear algorithms.

When the surface is not a Maybank Quadric, we can first solve the essential matrix $\mathbf{E} = \mathbf{T} \times \mathbf{R}$ linearly and then decompose \mathbf{E} into \mathbf{T} and \mathbf{R} . Among the linear algorithms^{2,7,34}, Philip's Singular Value Decomposition (SVD) algorithm is optimal in the sense of least squares. Here we present several simpler solutions.

First, we rewrite equations (5) or (7) as:

$$S_1 = \sum_{i=1}^n \{ \Theta_i^T (\mathbf{T} \times \mathbf{R}) \Theta_i \}^2 = \mathbf{h}^T \mathbf{A}_n^T \mathbf{A}_n \mathbf{h} \tag{B1}$$

where $\mathbf{h} = r s \mathbf{E}$. In the above equation we force $\|\mathbf{h}\|^2 = 2$ so that the corresponding solution of \mathbf{T} below satisfies $\|\mathbf{T}\|^2 = 1$. The optimal solution of \mathbf{h} that minimizes S_1 corresponds to the eigenvector of $\mathbf{A}_n^T \mathbf{A}_n$ associated with its smallest eigenvalue. Although the sign of \mathbf{h} is uncertain, this uncertainty will not affect the decomposition discussed below. Then the goal below is to estimate a unit translation vector \mathbf{T} and a rotation matrix \mathbf{T} such that $\|\mathbf{E} - \mathbf{T} \times \mathbf{R}\|^2$ is minimized, where the (Frobenius or Euclid) norm of a matrix is defined as the sum of its squared elements. In general, an essential matrix will have two decompositions of the form $\mathbf{T} \times \mathbf{R}$, which constitute a twisted pair². However, only one decomposition yields positive depths for the correspondence data. The following method finds the solution that yields positive depths for the correspondences.

It can be shown³⁴ that if (\mathbf{T}, \mathbf{R}) is the optimal decomposition of \mathbf{E} , then \mathbf{T} corresponds to the eigenvector of $\mathbf{E}^T \mathbf{E}$ associated with its least eigenvalue subject to $\|\mathbf{T}\|^2 = 1$. Therefore, \mathbf{T} is the linear least squares solution of the following equation:

$$\mathbf{E}^T \mathbf{T} = \mathbf{E}^T [t_1 \ t_2 \ t_3]^T = 0 \tag{B2}$$

To determine if \mathbf{T} or $-\mathbf{T}$ is the right solution, we need make use of the correspondence data. Let:

$$\mathbf{T} \times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} = \mathbf{G} = \begin{bmatrix} \mathbf{g}_1^T \\ \mathbf{g}_2^T \\ \mathbf{g}_3^T \end{bmatrix} \tag{B3}$$

Then from the motion equation:

$$\Theta'_i Z'_i = \mathbf{R} \Theta_i Z_i + \mathbf{T} \tag{B4}$$

we would have:

$$\mathbf{T} \times (\Theta'_i Z'_i) = (\mathbf{T} \times \mathbf{R}) \Theta_i Z_i, \text{ or } \frac{Z'_i}{Z_i} \mathbf{G} \Theta'_i = \mathbf{E} \Theta_i \tag{B5}$$

Irrespective of whether \mathbf{T} or $-\mathbf{T}$ is the right solution, Z'_i/Z_i can be determined by:

$$\frac{Z'_i}{Z_i} = \sqrt{\frac{\Theta_i^T \mathbf{E}^T \mathbf{E} \Theta_i}{\Theta_i^T \mathbf{G}^T \mathbf{G} \Theta_i}} \tag{B6}$$

Let:

$$L_1 = \sum_{i=1}^n \left\| \mathbf{E} \Theta_i - \frac{Z'_i}{Z_i} \mathbf{G} \Theta'_i \right\| \tag{B7}$$

$$L_2 = \sum_{i=1}^n \left\| \mathbf{E} \Theta_i + \frac{Z'_i}{Z_i} \mathbf{G} \Theta'_i \right\| \tag{B8}$$

then if $L_1 < L_2$, \mathbf{T} is the correct solution; else we must have $L_2 < L_1$ and $-\mathbf{T}$ is the correct solution. We denote the correct solution as \mathbf{T} in the below.

After \mathbf{T} is determined, we need to estimate \mathbf{R} such that $\|\mathbf{E} - \mathbf{T} \times \mathbf{R}\|^2$ or $\|\mathbf{E}^T - \mathbf{R}^T \mathbf{G}\|^2$ is minimized. The algorithms^{40,41} can then be applied to obtain the optimal solution of \mathbf{R} . For motion of small rotations, the rotation angles can also be solved for from equation (20) in a closed form.

ACKNOWLEDGEMENTS

The support of National Science Foundation and Defense Advanced Research Projects Agency under grant IRI-89-02728, and the US Army Advanced Construction Technology Center under grant DAAL 03-87-K-0006, is gratefully acknowledged.

REFERENCES

- Longuet-Higgins, H 'A computer algorithm for reconstructing a scene from two projections', *Nature*, Vol 293 (1981) pp 133-135
- Tsai, R and Huang, T 'Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces', *IEEE Trans. PAMI*, Vol 6 No 1 (1984) pp 13-26
- Tsai, R, Huang, T and le Zhu, W 'Estimating three-dimensional motion parameters of a rigid planar patch, II: Singular value decomposition', *IEEE Trans. Account., Speech and Signal Process.*, Vol 30 No 4 (1982) pp 525-534
- Longuet-Higgins, H 'The visual ambiguity of a moving plane', *Proc. Royal Soc. Lond., Series B*, Vol 223 (1985) pp 165-175
- Longuet-Higgins, H 'The reconstruction of a plane surface from two perspective projection', *Proc. Royal Soc. Lond., Series B*, Vol 227 (1986) pp 399-410
- Longuet-Higgins, H 'Multiple interpretations of a pair of images of a surface', *Proc. Royal Soc. Lond., Series B*, Vol 223 (1988) pp 165-175
- Zhuang, X, Huang, T and Haralick, R 'Two-view motion analysis: A unified algorithm', *Opt. Soc. Amer.*, Vol 3 No 9 (1986) pp 1492-1500
- Horn, B 'Relative orientation', *Proc. Image Understanding Workshop*, Irvine, CA (1989) pp 826-837
- Horn, B 'Motion fields are hardly ever ambiguous', *Int. J. Computer Vision*, Vol 1 (October 1987)
- Weng, J, Ahuja, N and Huang, T 'Motion and structure from point correspondences: A robust algorithm for planar case with error estimation', *Proc. Int. Conf. on Pattern Recognition*, Paris, France (1988)
- Hu, X and Ahuja, N 'A robust algorithm for plane motion solution', *Proc. Int. Conf. on Automation, Robotics and Computer Vision*, Singapore (1990)
- Faugeras, O and Maybank, S 'Motion from point matches: multiplicity of solutions', *Proc. IEEE Workshop on Visual Motion*, Irvine, CA (1989) pp 248-255
- Netravali, A, Huang, T, Krishnakumar, A and Holt, R 'Algebraic methods in 3-D motion estimation from two-view point correspondence', *Int. J. Imaging Syst. & Technol.*, Vol 1 (1989) pp 78-99
- Negahdaripour, S 'Multiple interpretations of the shape and motion of objects from two perspective images', *IEEE Trans. PAMI*, Vol 12 No 11 (1990) pp 1025-1039
- Negahdaripour, S 'Ambiguities of a motion field', *Proc. 1st Int. Conf. Comput. Vision*, London, UK (1987) pp 607-611
- Negahdaripour, S 'Critical surface pairs and triplets', *Int. J. Comput. Vision*, Vol 3 (1989)
- Negahdaripour, S 'Closed-form relationship between the two interpretations of a moving plane', *J. Opt. Soc. Amer. A*, Vol 7 (1990)
- Cui, N, Weng, J and Cohen, P 'Extended structure and motion analysis from monocular image sequences', *Proc. 3rd ICCV*, Osaka, Japan (1990) pp 222-229
- Broida, T and Chellapa, R 'Estimating the kinematics and structure of a rigid object from a sequence of monocular images', *IEEE Trans. PAMI*, Vol 13 No 6 (1991) pp 497-513
- Young, G and Chellapa, R 'Monocular motion estimation using a long sequence of noisy images', *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Toronto, Canada, Vol 4 (1991) pp 2437-2440
- Young, G and Chellapa, R '3-D motion estimation using a sequence of noisy stereo images: models, estimation, and uniqueness results', *IEEE Trans. PAMI*, Vol 12 No 8 (1990) pp 735-759
- Hu, X and Ahuja, N 'Sufficient conditions for double or unique solution of motion and structure', *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Toronto, Canada, Vol 4 (1991) pp 2445-2448 (To appear in *CVGIP: Image Understanding*, July 1993)
- Roach, J and Aggarwal, J K 'Determining the movement of objects from a sequence of images', *J. ACM*, Vol 6 (1980) pp 554-562
- Yen, B and Huang, T 'Determining the 3-D motion and structure of a rigid object using spherical projections', *Comput. Vision, Graph. & Image Process.*, Vol 21 (1983) pp 21-32
- Spetsakis, M and Aloimonos, J 'Optimal computing of structure from motion using point correspondences in two frames', *Proc. 1st ICCV*, Tampa, FL (1988) pp 449-453
- Weng, J, Ahuja, N and Huang, T S 'Closed-form solution + maximum likelihood: A robust approach to motion and structure estimation', *Proc. CVPR*, Ann Arbor, MI (1988) pp 381-386
- Jerian, C and Jain, R 'Polynomial methods for structure from motion', *Proc. 1st ICCV*, Tampa, FL (1988) pp 197-206
- Huang, T and Faugeras, O 'Some properties of the E matrix in two-view motion estimation', *IEEE Trans. PAMI*, Vol 11 No 12 (1989) pp 1310-1312
- Faugeras, O and Maybank, S 'Motion from point matches: multiplicity of solutions', *Int. J. Computer Vision*, Vol 4 (1990) pp 225-246
- Weng, J, Huang, T and Ahuja, N 'Motion and structure from two perspective views: Algorithms, error analysis, and error estimation', *IEEE Trans. PAMI*, Vol 11 No 5 (1989) pp 451-476
- Hooke, R and Jeeves, T 'Direct Search Solution', of numerical and statistical problems', *J. ACM* (1961) pp 212-229
- Wilder, D and Beightler, C *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, NJ (1967)
- Iu, S-L and Wohn, K 'Estimation of general rigid motion from a long sequence of images', *Proc. ICPR*, Atlantic City, NJ (1990) pp 217-219
- Philip, J 'Estimation of three-dimensional motion of rigid objects from noisy observations', *IEEE Trans. PAMI*, Vol 13 No 1 (1991) pp 61-66
- Aloimonos, J 'Purposive and qualitative active vision', *Proc. Int. Joint Conf. on Pattern Recognition* (1990) pp 346-360
- Weng, J, Huang, T and Ahuja, N '3-D motion estimation, understanding, and prediction from noisy image sequences', *IEEE Trans. PAMI*, Vol 9 No 3 (1987) pp 370-389
- Debrunner, C and Ahuja, N 'A direct data approximation based motion estimation algorithm', *Proc. 10th Int. Conf. on Pattern Recognition*, Atlantic City, NJ (1990) pp 384-389
- Tomasi, C and Kanade, T 'Shape and motion without depth', *Proc. 3rd ICCV*, Osaka, Japan (1990) pp 91-95
- Tomasi, C and Kanade, T 'The factorization method for the recovery of shape and motion from image streams', *Image Understanding Workshop*, San Diego, CA (1992) pp 459-472
- Faugeras, O and Hebert, M 'A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces', *Int. Joint Conf. on Artif. Intell.* Karlsruhe, Germany (1983) pp 996-1001
- Arun, K, Huang, T and Blostein, S 'Least squares fitting of two 3-D point sets', *IEEE Trans. PAMI*, Vol 11 (1987) pp 698-700
- Hu, X *Perception of Shape and Motion*, PhD dissertation, Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL (1993)