Estimating Motion of Constant Acceleration from Image Sequences

Xiaoping Hu and Narendra Ahuja
Dept. of ECE and Beckman Institute, Univ. of Illinois
405 N. Mathews Avenue, Urbana, IL 61801, USA
Email: xh, ahuja@uirvld.csl.uiuc.edu

ABSTRACT

This paper presents a model-based algorithm for estimating motion from monocular image sequences. We first present a new two-view motion algorithm and then extend it to multiple views. The two-view algorithm requires generally 6 pairs of point correspondences to give unique solution of the motion parameters. However, when the used points lie on a Maybank Quadric, the algorithm requires 7 pairs of point correspondences to give double solutions. Object-centered motion representations and a motion model of constant acceleration are used to estimate motion parameters from long image sequences. The algorithm guarantees globally optimal solution. Since the algorithm does not involve structure parameters, it contains the least number of unknowns and is hence more efficient and robust than the existing ones. Experimental results with real image data are presented. The same method can be applied to solve for motions described by second or higher orders of polynomials.

1 Introduction

This paper addresses the problem of estimating motion and structure from point correspondences in a monocular image sequences. It is well-known that the motion problem is nonlinear and very sensitive to noise. One way to improve the performance of motion algorithms is to use model-based motion representations to obtain additional constraints. A good model involves fewer unknowns and, therefore, whenever the model applies, model-based motion estimation should give better estimation results than an algorithm that applies to general motion. In this paper, we consider motion having constant acceleration.

In the previous approaches [2][1][13][5], structure parameters are solved for simultaneously with the motion parameters. This not only greatly increases the computation complexity, but makes a globally optimal solution impossible. We show that the structure parameters can be eliminated in advance and present a two-step nonlinear algorithm which involves only motion parameters. First the rotation parameters are solved for nonlinearly and then the translation parameters are solved for in a closed form. Therefore a much smaller number of unknowns are involved in the nonlinear optimizations process. The same method can be applied to motions represented by any order of polynomials and guarantees globally optimal solution for motions described by second or lower orders of polynomials. The rotations are represented in the matrix form and do not introduce additional model noise which exists in some of the previous formulations [1][13][5]. Experiments with real image data are presented to show that the algorithm yields accurate estimates. Since the long sequence motion algorithm heavily relies on the two-view motion algorithm, we first discuss two-view and then multi-view motions.

Section 2 presents a very robust nonlinear two-view motion algorithm which gives double or unique solution for any rigid surface. Section 3 discusses the model and the solution for motion of constant acceleration. Section 4 presents experimental results with real image data. Section 5 summarizes the paper.

2 Two-View Motion Solution

Linear [9][14] and nonlinear [11][8] motion algorithms have been developed for motion estimation. The former rarely work for noisy data, though they have been used to produce initial guesses for the latter. The existing nonlinear algorithms all assume that the motion is uniquely determined and require good initial guesses. In this section we present a robust and efficient nonlinear two-step algorithm which solves for rotation first and then translation.

Let $\mathbf{R} = (r_{ij})$ and $\mathbf{T} = [\ t_1\ t_2\ t_3\ ]^T$ ($\mathbf{T}$ can be zero here) be the rotation matrix and translation vector between the two views. Let $\Theta_i = [\ x_i\ \ y_i\ \ 1\ ]$, $\Theta_i' = [\ x_i'\ \ y_i'\ \ 1\ ]$, i = 1,2, ..., n (n ≥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:

$$S_1 = \sum_{i=1}^{n} \{(\Theta_i' \times \mathbf{R}\Theta_i) \cdot \mathbf{T}\}^2 = \mathbf{T}^T \Pi_n^T \Pi_n \mathbf{T} = ||\Pi_n \mathbf{T}||^2, \tag{2.1}$$

where

$$\Pi_n = \begin{bmatrix} (\Theta_1' \times \mathbf{R}\Theta_1')^T \\ \vdots \\ (\Theta_n' \times \mathbf{R}\Theta_n')^T \end{bmatrix}, \quad ||\mathbf{T}||^2 = 1. \tag{2.2}$$

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:

$$S_2 = \sum_i (x_i' - p_i)^2 + (y_i' - q_i)^2, \tag{2.3}$$

where

$$p_i = \frac{r_{11}x_i + r_{12}y_i + r_{13}}{r_{31}x_i + r_{32}y_i + r_{33}},\ q_i = \frac{r_{21}x_i + r_{22}y_i + r_{23}}{r_{31}x_i + r_{32}y_i + r_{33}}. \tag{2.4}$$

The support of State of Illinois Department of Commerce and Community Affairs under grant 90-103 is gratefully acknowledged.

0-8186-2910-X/92 $3.00 © 1992 IEEE

If S_2 is close to zero (the particular threshold depends on the camera parameters), then the estimated **T** is unreliable. Otherwise, the translation is visible and hence the estimation of **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 **R** and hence only 3 unknowns are involved in the first step. To estimate **R**, we represent it by the three-angle representation

$$\mathbf{R} = \mathbf{A}_X(\omega_X)\mathbf{A}_Y(\omega_Y)\mathbf{A}_Z(\omega_Z), \tag{2.5}$$

where $\mathbf{A}_X$, $\mathbf{A}_Y$, and $\mathbf{A}_Z$ are the rotation matrices around X, Y, Z axes respectively, e.g.,

$$\mathbf{A}_Z = \begin{bmatrix} \cos\omega_Z & -\sin\omega_Z & 0 \\ \sin\omega_Z & \cos\omega_Z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.6}$$

The search of ω_X, ω_Y, and ω_Z can be fast 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 (ω_X, ω_Y, ω_Z) at (0,0,0) or at the estimate obtained from linear solution. We have experimentally found that in the neighborhood of the global minimum, different optimization algorithms (e.g., hill-climbing or gradient method) all work.

S_1 is a weighted version of the following criterion that has been used by many researchers [11][2][1][13][5]:

$$S_3 = \sum_i (x_i' - u_i)^2 + (y_i' - v_i)^2 \equiv \sum_i d_i^2, \tag{2.7}$$

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 **R** and **T** through

$$u_i = \frac{(r_{11}x_i + r_{12}y_i + r_{13})Z_i + t_1}{(r_{31}x_i + r_{32}y_i + r_{33})Z_i + t_3}, \tag{2.8}$$

$$v_i = \frac{(r_{21}x_i + r_{22}y_i + r_{23})Z_i + t_2}{(r_{31}x_i + r_{32}y_i + r_{33})Z_i + t_3}. \tag{2.9}$$

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

$$[\,u_i \quad v_i \quad 1\,]\,(\mathbf{T} \times \mathbf{R}\Theta_i) = 0. \tag{2.10}$$

Different values of Z_i makes (u_i, v_i) lie on different positions on l_i. Therefore, the optimal solution 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

$$S_3 = \sum_i \Theta_i' \cdot (\mathbf{T}\times\mathbf{R}\Theta_i)/\sqrt{a_i^2 + b_i^2}, \tag{2.11}$$

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.

The nonlinear algorithm above yields one optimal solution for any rigid surface. 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 Maybank Quadric (two planes), we have to deal with it differently from other Maybank Quadrics since in general infinitely many Maybank Quadrics are associated with a single plane [4].

The existing linear algorithms [3][12][6][10] 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 2 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 if the surface is planar, and if it is, we refine the estimation by applying plane motion algorithms.

Assume one solution $\mathbf{R}_1$ and $\mathbf{T}_1$ is obtained and minimizes S_1. Then the structure parameter Z_i is obtained (Z_i has unique solution if and only if Θ'_i is not parallel to **T**) by minimizing d_i, or by equalizing (u_i, v_i) to the intersection point of l_i and its perpendicular line passing (x_i', y_i'). Then we solve for a hypothesized plane structure $\mathbf{N}_1$ from

$$\mathbf{N}^T\mathbf{X} = [\,n_1 \quad n_2 \quad n_3\,]\Theta Z = 1. \tag{2.12}$$

The other set of motion parameters $\mathbf{R}_2$, $\mathbf{T}_2$, and $\mathbf{N}_2$ is obtained by decomposing the *plane motion matrix* **K** [3][12] defined as

$$\mathbf{K} = \mathbf{R}_1 + \mathbf{T}_1\mathbf{N}_1^T. \tag{2.13}$$

We then substitute $\mathbf{R}_2$ and $\mathbf{T}_2$ into (2.1) to check if

$$S_1(\mathbf{R}_2, \mathbf{T}_2) - S_1(\mathbf{R}_1, \mathbf{T}_1) < \varepsilon \cdot S_1(\mathbf{R}_1, \mathbf{T}_1), \tag{2.14}$$

where ϵ is a preset constant (say 0.1). If so, then the surface is planar; if not, the surface is not planar. In case **K** has unique decomposition, 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 3–D 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 algorithm [3] is used to obtain a better estimation of the motion and structure parameters.

The situation for general Maybank Quadrics is more complicated. The algorithm presented below requires 7 correspondences of points satisfying certain condition to determine the other possible solution from one known solution if the surface is Maybank Quadric.

Assume one solution $\mathbf{R}_1$ and $\mathbf{T}_1$ has been obtained. The goal below is to determine whether there is another solution which satisfy the same image data; if so, obtain an optimal solution of the alternative solution. It is well known that if another solution **R** and **T** satisfies the data, then the points used for correspondences must lie on a Maybank Quadric defined as follows:

$$\frac{1}{2}Z\Theta^T\left(\mathbf{R}_0^T\mathbf{E} + \mathbf{E}^T\mathbf{R}_0\right)\Theta + \mathbf{T}_0^T\mathbf{E}\Theta = 0, \tag{2.15}$$

where Z is the depth of the image point Θ and $\mathbf{E} = \mathbf{T}\times\mathbf{R}$. The depth Z can be solved for as described above from$\mathbf{R}_1$ and $\mathbf{T}_1$. It is well known that the Maybank Quadric passes through $Z\Theta = -\mathbf{R}^{\mathrm{T}}{}_1\mathbf{T}_1 \equiv \mathbf{O}$, where $\mathbf{O}$ is the new origin of the coordinates for camera-centered motion model. Now given n ($n \geq 7$) points Θ_i, i = 1,2, ..., n, we first fit the points Θ_i and $\mathbf{O}$ with a more general surface of the form

$$\frac{1}{2}Z\mathbf{\Theta}^{\mathrm{T}}\mathbf{A}\mathbf{\Theta}+\mathbf{B}^{\mathrm{T}}\mathbf{\Theta} = 0, \tag{2.16}$$

where $\mathbf{A}$ is symmetric matrix and $\mathbf{B}$ is a vector. $\mathbf{A}$ and $\mathbf{B}$ can be determined linearly from Equation (2.16) if and only if the resulting coefficient matrix has a row rank of 8 or above. After $\mathbf{A}$ and $\mathbf{B}$ have been solved for, we can then solve for $\mathbf{E}$ from

$$\mathbf{R}_0^{\mathrm{T}}\mathbf{E}+\mathbf{E}^{\mathrm{T}}\mathbf{R}_0=\mathbf{A}, \quad \mathbf{T}_0^{\mathrm{T}}\mathbf{E}=\mathbf{B}^{\mathrm{T}}, \tag{2.17}$$

After $\mathbf{E}$ has been solved for, we can then obtain an optimal solution of the other possible set of motion solution $\mathbf{T}_2$ and $\mathbf{R}_2$ using the method in [7].

Again, Equation (2.14) is used to determine if $\mathbf{T}_2$ and $\mathbf{R}_2$ are really a solution and if the surface is Maybank Quadric.

3 Multi-View Motion Solution

The accuracy of motion estimation can be greatly improved when a long sequence of images are used, because more evidences about motion are present.

In this section we introduce a model-based approach which uses an accurate, yet flexible object-centered motion representation. It is evident that model based motion algorithm should give better results than a general motion algorithm since a right model involves fewer unknowns and employs more constraints.

First let us describe the camera-centered and object-centered motion representations used in our approach.

The camera-centered interframe motion representation is as follows ([2]):

$$\mathbf{X}_n=\mathbf{R}_n\mathbf{X}_{n-1}+\mathbf{t}_n, \ \ or \ \ \mathbf{X}_i=\mathbf{R}_{i,j}\mathbf{X}_j+\mathbf{t}_{i,j}, \tag{3.1}$$

where

$$\mathbf{R}_{i,j}=\mathbf{R}_i\mathbf{R}_{i-1}\cdots\mathbf{R}_{j+1}, \quad \mathbf{t}_{i,j}=\mathbf{t}_i+\sum_{k=j+1}^{i-1}\mathbf{R}_{i,k}\mathbf{t}_k. \tag{3.2}$$

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

The motion of a rigid object can also be considered as that the object rotates about a center $\mathbf{O}$ which translates relative to the camera center $\mathbf{C}$. The coordinate system used is still camera-centered. We use $\mathbf{O}_n$ to denote the rotation center'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 center at time n. Then the object-centered motion representation between two consecutive frames is as follows:

$$\begin{aligned}\mathbf{X}_n&=\mathbf{R}_n[\mathbf{X}_{n-1}-\mathbf{O}_{n-1}]+\mathbf{O}_n,\\ \mathbf{O}_n&=\mathbf{O}_{n-1}+\mathbf{T}_n=\cdots=\mathbf{O}_0+\sum_{i=1}^{n}\mathbf{T}_i.\end{aligned} \tag{3.3}$$

Reordering Equation (3.3) as

$$\mathbf{X}_n-\mathbf{O}_n=\mathbf{R}_n(\mathbf{X}_{n-1}-\mathbf{O}_{n-1}) \tag{3.4}$$

and using Equation (3.2) we can obtain the following representation of motion between frame i and j (i > j):

$$\begin{aligned}\mathbf{X}_i &= \mathbf{R}_{i,j}[\mathbf{X}_j-\mathbf{O}_j]+\mathbf{O}_i\\ &= \mathbf{R}_{i,j}\mathbf{X}_j-\mathbf{R}_{i,j}\left[\mathbf{O}_0+\sum_{k=1}^{j}\mathbf{T}_k\right]+\mathbf{O}_0+\sum_{k=1}^{i}\mathbf{T}_k\\ &= \mathbf{R}_{i,j}\mathbf{X}_j+[\mathbf{I}-\mathbf{R}_{i,j}]\left[\mathbf{O}_0+\sum_{k=1}^{j}\mathbf{T}_k\right]+\sum_{k=j+1}^{i}\mathbf{T}_k\\ &\equiv \mathbf{R}_{i,j}\mathbf{X}_j+\mathbf{t}_{i,j}.\end{aligned} \tag{3.5}$$

Let us note that the above representation is significantly different from the camera-centered motion representation when the motion involves rotation. In this representation, the initial rotation center $\mathbf{O}_0$, and the motion parameters, $\mathbf{T}_n$, and $\mathbf{R}_n$, n = 1,2,3..., are the unknowns to be determined. 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..., share the same rotation axis $\mathbf{n}_{\mathbf{R}_1}$, then if $\mathbf{O}_0$ is a solution, $\mathbf{O}_0 + \alpha\mathbf{n}_{\mathbf{R}_1}$ for any constant α is also a solution. That is, any point on the rotation axis can serve as the rotation center. However, this uncertainty will not affect our understanding of the motion. And whenever such uncertainty occurs, we can remove it by enforcing $\mathbf{O}_0 \cdot \mathbf{n}_{\mathbf{R}_1} = 0$.

For the motion model of constant accelerations, we assume that the rotation axis is fixed, the rotation angle and the translation vector change with constant accelerations:

$$\mathbf{T}_n = \mathbf{T}_0 + (n-1)\mathbf{T}_a, \ \phi_n = \phi_0+(n-1)\phi_a, \tag{3.6}$$

$$\mathbf{R}_n = \mathbf{n}\mathbf{n}^{\mathrm{T}}-\left(\mathbf{n}\mathbf{n}^{\mathrm{T}}-\mathbf{I}\right)\cos\phi_n+\mathbf{n}\times\mathbf{I}\sin\phi_n, \tag{3.7}$$

where $\mathbf{T}_0$ and $\mathbf{T}_a$ are vectors, $\mathbf{n}$ 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

$$\mathbf{R}_{i,j} = \mathbf{n}\mathbf{n}^{\mathrm{T}}-\left(\mathbf{n}\mathbf{n}^{\mathrm{T}}-\mathbf{I}\right)\cos\phi_{i,j}+\mathbf{n}\times\mathbf{I}\sin\phi_{i,j}, \tag{3.8}$$

$$\mathbf{t}_{i,j}=\mathbf{U}_{i,j}\mathbf{O}_0+\mathbf{V}_{i,j}\mathbf{T}_0+\mathbf{W}_{i,j}\mathbf{T}_a, \tag{3.9}$$

from which we have

$$(\mathbf{t}_{i,j}\times\mathbf{U}_{i,j})\mathbf{O}_0+(\mathbf{t}_{i,j}\times\mathbf{V}_{i,j})\mathbf{T}_0+(\mathbf{t}_{i,j}\times\mathbf{W}_{i,j})\mathbf{T}_a=0, \tag{3.10}$$

where

$$\phi_{i,j}=\sum_{k=j}^{i}\phi_k=j\phi_0+[a_i-a_j]\phi_a, \tag{3.11}$$

$$a_k=\sum_{j=0}^{k-1}j=\frac{k(k-1)}{2}, \tag{3.12}$$

and

$$\mathbf{U}_{i,j}=\mathbf{I}-\mathbf{R}_{i,j},\ \ \mathbf{V}_{i,j}=j\mathbf{U}_{i,j}+(i-j)\mathbf{I},$$
$$\mathbf{W}_{i,j}=a_j\mathbf{U}_{i,j}+(a_i-a_j)\mathbf{I}. \tag{3.13}$$

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

$$\mathbf{n}=[\sin\alpha\cos\beta,\quad \cos\alpha,\quad \sin\alpha\sin\beta]^{\mathbf{T}}. \tag{3.14}$$

Therefore, $\mathbf{T}_0$, $\mathbf{T}_a$, α, β, ϕ_0, and ϕ_1 are the unknown parameters.

Let $\Pi_{i,j}$ be the matrix defined in the way described in (2.2). Let $\lambda_{i,j}$ be the least eigenvalue of $\Pi_{i,j}{}^{\mathbf{T}}\Pi_{i,j}$. Then the rotation parameters for each model are searched to minimize

$$\Lambda=\sum_{i>j} w_{i,j}\lambda_{i,j}, \tag{3.15}$$

where $w_{i,j}$ is a weighting factor. Since $\Pi_{i,j}{}^{\mathbf{T}}\Pi_{i,j}$ 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 guaranteed for this model.

To obtain an optimal solution of the translation parameters, first $\mathbf{t}_{i,j}$ is obtained for all i and j with i > j using the two-view motion algorithm and the rotation parameters obtained with the method above. Then $\mathbf{O}_0$, $\mathbf{T}_0$, and $\mathbf{T}_a$ are solved for from Equation (3.10) with the linear least squares method. The obtained $\mathbf{O}_0$ and $\mathbf{T}_0$ are then used to compute $\mathbf{t}_{i,j}$ using Equation (3.9). In general, $\mathbf{O}_0$, $\mathbf{T}_0$, and $\mathbf{T}_a$, cannot be obtained accurately. However, the interframe translation $\mathbf{t}_{i,j}$ computed from them is in good accuracy.

This algorithm gives unique solution whenever the motion is uniquely defined by the correspondence data. It is noteworthy that method above allows independent choices of models for rotation and translation. For example, a motion may involve a rotation with constant acceleration but a translation of second order polynomials.

4 Experimental Results

This section presents two examples with real image data. An active vision system is developed for the motion and structure estimation problem. The system is able to yield any required motion and capture a motion sequence of images while graphically controlled at the computer terminal. The images are taken with a Cohu solid state camera of 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.

In the examples provided below, only the ground truth of rotation angles are accurately recorded. Because of the difficulty in measuring the direction of the camera optical axis and the position of rotation center relative to the optical center, the rotation axis and translation direction cannot be measured accurately. Therefore the reference ground truth and results for the translation directions are not presented due to short of space and actually the "ground truth" may be more erroneous than the estimates. From our experience with simulation data, the estimation of rotation axis and translation direction is usually more reliable than that of rotation angle.

In the experiments below, the weighting factor $w_{i,j}$ is chosen as $1/N_{i,j}$, which is the number of correspondences between the *i*th and *j*th views. If $N_{i,j}$ = 0, then $w_{i,j}$ is set to zero. The correspondences are obtained with a newly developed matching algorithm.

The first example contains fifteen images. Figures 1 (a) 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 1 (c) 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 ($[0, 1, 0]^{\mathbf{T}}$). The estimated rotation parameters are: $\mathbf{n}$ = [0.0042, 0.9999, –0.0069], ϕ_0 = 0.5379°. The two-view motion algorithm gives the following results: the rotation between the first and the second views is $\mathbf{n}$ = [0.0227, 0.9986, –0.0461], ϕ = 0.6096°, and the rotation between the fourteenth and the last views is $\mathbf{n}$ = [0.3779, 0.6722, –0.6366], ϕ = 0.4783°. The arbitrary motion model gives the following results: the rotation between the first and the second views is $\mathbf{n}$ = [0.5916, 0.7918, 0.1515], ϕ = 0.5563°, and the rotation between the fourteenth and the last views is $\mathbf{n}$ = [0.2969, 0.8318, 0.4688], ϕ = 0.5209°. It is obvious that the model-based motion algorithm yields much better results.

In the fourth example, fifteen images are taken by a moving camera. Figures 2 (a) and (b) show the first and last images. No correspondences are found between these two images because of the large motion between them. Figures 2 (c) and (d) show the eighth and ninth images and the correspondences. The motion involves a rotation (tilting) of constant acceleration around the X axis ($[1\ 0\ 0]^{\mathbf{T}}$) with ϕ_0 = 0.6° per frame, ϕ_a = 0.04° per frame^2, and a constant translation along X axis ($[1\ 0\ 0]^{\mathbf{T}}$). The rotation causes also a translation along Z and Y axes, the amount of which is not accurately measured. The estimated rotation parameters are: $\mathbf{n}$ = [0.9703, –0.1975, 0.1389], ϕ_0 = 0.6371°, ϕ_0 = 0.0416°.

5 Summary

In this paper we have presented a model-based algorithm and a nonlinear two-view motion algorithm for estimating motion parameters from a long sequence of images. These algorithms utilize the state-of-art techniques developed by many motion researchers and are very flexible for use. The whole process, from images to feature points, to matching, and then to motion estimatior, is fully automated. The application of the algorithms to real image data has obtained good results, from which we can conclude that model-based methods yield much better results than those assuming arbitrary motions.

This algorithm has four good features:

1. It works for general motion (including pure rotation and translational motion) of any rigid surface and significantly involves only motion parameters as unknowns.
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. 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 two-step computing makes it ideal for real time application.

REFERENCES

[1] T.J. Broida and R. Chellapa. Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *PAMI-13*, (6):497–513, 1991.

[2] N. Cui, J. Weng, and P. Cohen. Extended structure and motion analysis from monocular image sequences. *Proc. of 3rd ICCV*, pages 222–229, 1990.

[3] X. Hu and N. Ahuja. A robust algorithm for plane motion solution. *Proc. Inter. Conf. on Automation, Robotics and Computer Vision*, 1990.

[4] X. Hu and N. Ahuja. Sufficient conditions for double or unique solution of motion and structure. *Proc. Inter. Conf. on ASSP*, 4:2445–2448, 1991.

[5] Siu-Leong Iu and Kwangyoen Wohn. Estimation of general rigid motion from a long sequence of images. *Proc. of ICPR*, pages 217–219, 1990.

[6] H.C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projection. *Proc. Royal Soc. of London, B*, 227:399–410, 1986.

[7] J. Philip. Estimation of three-dimensional motion of rigid objects from noisy observations. *PAMI-13*, (1):61–66, 1991.

[8] M. Spetsakis and J. Aloimonos. Optimal computing of structure from motion using point correspondences in two frames. *Proc. of ICCV*, pages 449–453, 1988.

[9] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *PAMI-6*, (1):13–26, 1984.

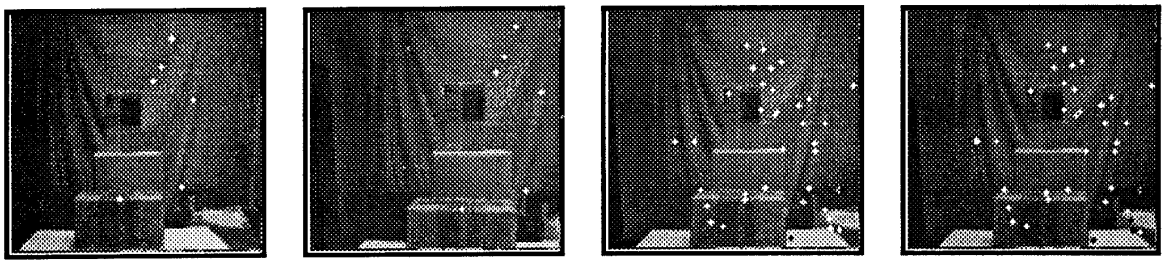
[10] R.Y. Tsai, T.S. Huang, and Wei le Zhu. Estimating three-dimensional motion parametersof a rigid planar patch, ii: Singular value decomposition. *IEEE Transactions on ASSP*, 30(4):525–534, 1982.

[11] J. Weng, N. Ahuja, and T. S. Huang. Closed-form solution + maximum likelihood: A robust approach to motion and structure estimation. *Proc. CVPR*, pages 381–386, 1988.

[12] J. Weng, N. Ahuja, and T.S. Huang. Motion and structure from point correspondences: A robust algorithm for planar case with error estimation. *Proc. Inter. Conf. Pattern Recognition*, 1988.

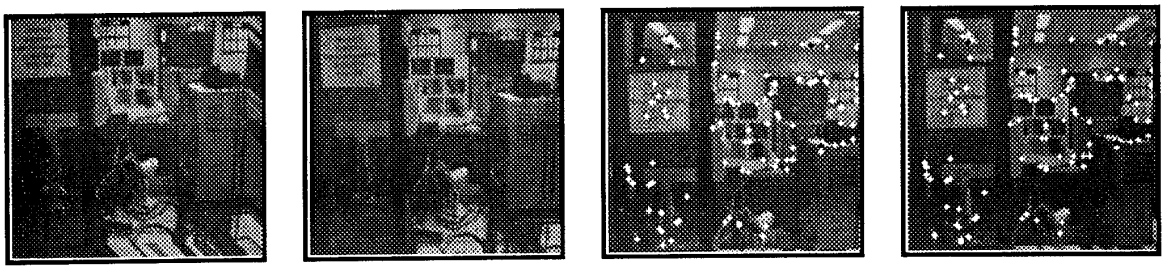
[13] G.S. Young and R. Chellapa. Monocular motion estimation using a long sequence of noisy images. *Proc. Inter. Conf. on ASSP*, 4:2437–2440, 1991.

[14] X. Zhuang, T.S. Huang, and R.M. Haralick. Two-view motion analysis: A unified algorithm. *J. of Opt. Soc. of Amer.*, 3(9):1492–1500, 1986.



(a). The first image. (b). The fifteenth image. (c). The eighth image. (d). The ninth image.

Fig. 1: Example I. Figures (a) and (b) show the first and the last images and the correspondences between them; Figures (c) and (d) show the eighth and ninth images and the correspondences between them.



(a). The first image. (b). The fifteenth image. (c). The eighth image. (d). The ninth image.

Fig. 2: Example II. Figures (a) and (b) show the first and the last images and the correspondences between them; Figures (c) and (d) show the eighth and ninth images and the correspondences between them.