MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

Long Image Sequence Motion Analysis Using Polynomial Motion Models

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

ABSTRACT

This paper presents two algorithms for estimating motion and structure from long monocular image sequences: one using interframe correspondences, the other using point trajectories. Object-centered motion representations and motion models described by up to the second order polynomials are used to estimate motion parameters from long image sequences. The algorithms automatically find the proper model that applies to an image sequence and give the globally optimal solution for the motion parameters under the chosen model. The selection of motion model for rotation is independent from that for translation. Since the algorithms first solve for rotation parameters nonlinearly and then solve for translation and structure parameters in closed forms, they invlove less unknowns in the nonlinear search and are hence more robust and efficient than existing long sequence algorithms. Experimental results with real image data are presented.

1 Introduction

This paper presents model based algorithms for estimating motion and structure from point correspondences or trajectories in a monocular image sequence. Arbitrary motion and motion desribed by up to the second order polynomials are considered.

The previous approaches [1] [2] [3] [4] solve for motion and structure parameters simultaneously, involving infeasible computation complexity and making a globally optimal solution impossible. Earlier [5] we have presented a stepwise algorithm for solving motion of constant acceleration using interframe correspondences. We now extend the method to arbitrary motion described by up to the second oder polynomial models. Experiments with real image data are presented to demonstrate the performance of the algorithm.

The existing approaches have exploited only the motion consistency and smoothness properties. When trajectories of points are available, structure consistency can be enforced to further improve estimation accuracy. A nonlinear formulation and a corresponding stepwise solution for motion and structure estimation using point trajectories are also presentd in this paper. This formulation can also make combined use of both interframe correspondences and point trajectories [6].

2 Motion Representations

Let $\mathbf{X}_n$ be the position of a space point at time n, $\mathbf{R}_n$ and $\mathbf{t}_n$ be the rotation and translation between *n*th and *n-1*th frames, and

$$\mathbf{R}_{i,j}=\mathbf{R}_i\mathbf{R}_{i-1}\cdots\mathbf{R}_{j+1}, \quad \mathbf{t}_{i,j}=\mathbf{t}_i+\sum_{k=j+1}^{i-1}\mathbf{R}_{i,k}\mathbf{t}_k. \tag{2.1}$$

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.

The camera-centered motion representation is ([1][5][6]):

$$\mathbf{X}_n=\mathbf{R}_n\mathbf{X}_{n-1}+\mathbf{t}_n, \ \ or \ \ \mathbf{X}_i=\mathbf{R}_{i,j}\mathbf{X}_j+\mathbf{t}_{i,j}\,. \tag{2.2}$$

In object-centered motion representation, everything is the same except that $\mathbf{t}_{i,j}$ is represented as ([5][6])

$$\mathbf{t}_{i,j}=[\mathbf{I}-\mathbf{R}_{i,j}]\left[\mathbf{O}_0+\sum_{k=1}^{j}\mathbf{T}_k\right]+\sum_{k=j+1}^{i}\mathbf{T}_k, \tag{2.3}$$

where $\mathbf{T}_k$ is the translation of rotation center $\mathbf{O}$ between the *n*th and *n-1*th frames, whose position at time 0 is $\mathbf{O}_0$.

3 Rotation Models

Polynomial and arbitrary rotation models are considered.

i. **Constant Rotation**

In this case, $\mathbf{R}_n = \mathbf{R}$ for all n and $\mathbf{R}_{i,j}$ reduces to

$$\mathbf{R}_{i,j} = \mathbf{R}^{i-j}\,. \tag{3.1}$$

$\mathbf{R}$ is expressed in the three-angle representation ([5]) during solution process and hence involves only three unknowns.

ii. **Rotation with Constant Acceleration**

In this case, the rotation angle changes at a constant rate about a fixed axis:

$$\mathbf{R}_n=\mathbf{n}\mathbf{n}^T-\left(\mathbf{n}\mathbf{n}^T-\mathbf{I}\right)\cos\phi_n+\mathbf{n}\times\mathbf{I}\sin\phi_n, \tag{3.2}$$

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

$$\mathbf{R}_{i,j}=\mathbf{n}\mathbf{n}^T-\left(\mathbf{n}\mathbf{n}^T-\mathbf{I}\right)\cos\phi_{i,j}+\mathbf{n}\times\mathbf{I}\sin\phi_{i,j}, \tag{3.3}$$

where

$$\phi_{i,j}=\sum_{k=j}^{i}\phi_k=j\phi_o+[a_i-a_j]\phi_a, \quad a_k=\sum_{j=o}^{k-1}j=\frac{k(k-1)}{2}. \tag{3.4}$$

To ease the solution, $\mathbf{n}$ is represented by two angles α and β as

$$\mathbf{n}=[\sin\alpha\cos\beta \quad \cos\alpha \quad \sin\alpha\sin\beta]^T. \tag{3.5}$$

α, β, ϕ_0, and ϕ_1 are the unknown rotation parameters.

iii. **Rotation Described by Second Order Polynomials**

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

$$\mathbf{\Omega}_n=[\,\omega_X^n \quad \omega_Y^n \quad \omega_Z^n\,]^T=\mathbf{\Omega}_o+n\mathbf{\Omega}_a+n^2\mathbf{\Omega}_b, \tag{3.6}$$

and

$$\mathbf{R}_n = \mathbf{A}_X(\omega_X^n)\mathbf{A}_Y(\omega_Y^n)\mathbf{A}_Z(\omega_Z^n). \tag{3.7}$$

The rotation between two nonconsecutive frames is still expressed by Equation (2.1). Three vectors Ω_0, Ω_a, and Ω_b are the unknowns to be estimated.

iv. **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,...,n, are considered to be independent and unknown, and are represented by the three-angle representation. Rotations between two nonconsecutive frames are expressed in the form of (2.1).

4 Translation Models

Similar models are considered for translation.

i. **Constant Translation**
In this case, the rotation center translates at a constant velocity. That is, $\mathbf{T}_n = \mathbf{T}_0 = \mathbf{T}$ for all n. Equation (2.3) then reduces to

$$\mathbf{t}_{i,j} = \mathbf{U}_{i,j}\mathbf{O}_o + \mathbf{V}_{i,j}\mathbf{T}_o, \tag{4.1}$$

from which we have

$$(\mathbf{t}_{i,j} \times \mathbf{U}_{i,j})\mathbf{O}_o + (\mathbf{t}_{i,j} \times \mathbf{V}_{i,j})\mathbf{T}_o = 0, \tag{4.2}$$

where

$$\mathbf{U}_{i,j} = \mathbf{I} - \mathbf{R}_{i,j}, \quad \mathbf{V}_{i,j} = j\mathbf{U}_{i,j} + (i-j)\mathbf{I}. \tag{4.3}$$

ii. **Translation with Constant Acceleration**
In this case, the rotation center's translation changes at a constant acceleration, that is,

$$\mathbf{T}_n = \mathbf{T}_o + (n-1)\mathbf{T}_a. \tag{4.4}$$

Then, the translation between *i*th and *j*th frame (i > j)

$$\mathbf{t}_{i,j} = \mathbf{U}_{i,j}\mathbf{O}_o + \mathbf{V}_{i,j}\mathbf{T}_o + \mathbf{W}_{i,j}\mathbf{T}_a, \tag{4.5}$$

from which we have

$$(\mathbf{t}_{i,j} \times \mathbf{U}_{i,j})\mathbf{O}_o + (\mathbf{t}_{i,j} \times \mathbf{V}_{i,j})\mathbf{T}_o + (\mathbf{t}_{i,j} \times \mathbf{W}_{i,j})\mathbf{T}_a = 0, \tag{4.6}$$

where

$$\mathbf{W}_{i,j} = a_j\mathbf{U}_{i,j} + (a_i - a_j)\mathbf{I}. \tag{4.7}$$

iii. **Translation Described by Second Order Polynomials**
In this case, the rotation center's translation is approximated by a second order polynomial. That is,

$$\mathbf{T}_n = \mathbf{T}_o + n\mathbf{T}_a + n^2\mathbf{T}_b. \tag{4.8}$$

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

$$\mathbf{t}_{i,j} = \mathbf{U}_{i,j}\mathbf{O}_o + \mathbf{V}_{i,j}\mathbf{T}_o + \mathbf{W}_{i,j}\mathbf{T}_a + \mathbf{Y}_{i,j}\mathbf{T}_b, \tag{4.9}$$

from which we have

$$(\mathbf{t}_{i,j} \times \mathbf{U}_{i,j})\mathbf{O}_o + (\mathbf{t}_{i,j} \times \mathbf{V}_{i,j})\mathbf{T}_o + (\mathbf{t}_{i,j} \times \mathbf{W}_{i,j})\mathbf{T}_a + (\mathbf{t}_{i,j} \times \mathbf{Y}_{i,j})\mathbf{T}_b = 0 \tag{4.10}$$

where

$$\mathbf{Y}_{i,j} = b_j\mathbf{U}_{i,j} + (b_i - b_j)\mathbf{I}, \quad b_k = \sum_{j=0}^{k-1} j^2 = \frac{1}{6}k(k-1)(2k-1). \tag{4.11}$$

iv. **Arbitrary Translation Model**
In this case, no smoothness constraint about the translation is assumed. Therefore, all translation vectors, $\mathbf{T}_i$, i = 1,2,...,n, are considered independent unknowns. Then, the translation between *i*th and *j*th frame (i > j) is

$$\mathbf{t}_{i,j} = \sum_{k=j+1}^{i} \mathbf{R}_{i,k}\mathbf{t}_k, \tag{4.12}$$

from which we can obtain

$$\sum_{k=j+1}^{i} (\mathbf{t}_{i,j} \times \mathbf{R}_{i,k})\mathbf{t}_k = 0, \tag{4.13}$$

where we assume $\mathbf{R}_{i,i} = \mathbf{I}$, and $\mathbf{R}_{i,j}$ is represented by (2.1).

5 Solution Using Interframe Matches

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) compute the structure parameters.

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, ..., $P_{i,j}$, 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

$$\Pi_{i,j} = \begin{bmatrix} (\Theta_1^{i,j} \times R_{i,j}\Theta_1^{j,i})^T \\ \vdots \\ \left(\Theta_{P_{i,j}}^{i,j} \times R_{i,j}\Theta_{P_{i,j}}^{j,i}\right)^T \end{bmatrix}. \tag{5.1}$$

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

$$\Lambda = \sum_{i>j} w_{i,j}\lambda_{i,j}. \tag{5.2}$$

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 [5] 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 (4.2), (4.6), or (4.10) 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 (4.2), (4.6), or (4.10). The solution for the arbitrary model is also the same as that for the polynomial models except that Equation (4.13) instead of (4.2), or (4.6), or (4.10) is used.

After the motion is solved for, 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 $\Theta_1 = (x_1, y_1, 1)$ 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 ith view being $\Theta_i = [x_i, y_i, 1]^T$. Let $\mathbf{R}_{ij} = (r_{mn}^{ij})$, $\mathbf{t}_{ij} = [t_1^{ij}, t_2^{ij}, t_3^{ij}]^T$ $(i > j)$ be the known rotation and translation between ith and jth 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

$$S = \sum_{i=2}^{F}\left(x_i - \frac{a_i Z_1 + t_1^{i1}}{c_i Z_1 + t_3^{i1}}\right)^2 + \left(y_i - \frac{b_i Z_1 + t_1^{i1}}{c_i Z_1 + t_3^{i1}}\right)^2 \qquad (5.3)$$

is minimized, where

$$a_i = r_{11}^{i1}x_1 + r_{12}^{i1}y_1 + r_{13}^{i1}, \quad b_i = r_{21}^{i1}x_1 + r_{22}^{i1}y_1 + r_{23}^{i1}, \quad c_i = r_{31}^{i1}x_1 + r_{32}^{i1}y_1 + r_{33}^{i1}. \qquad (5.4)$$

S 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

$$S' = \sum_{i=2}^{F}\left[x_i\left(c_i Z_1 + t_3^{i1}\right) - \left(a_i Z_1 + t_1^{i1}\right)\right]^2 + \left[y_i\left(c_i Z_1 + t_3^{i1}\right) - \left(b_i Z_1 + t_2^{i1}\right)\right]^2 \qquad (5.5)$$

6 Solution Using Point Trajectories

Let $\mathbf{M}_i = \mathbf{R}_{i1}$ and $\mathbf{V}_i = \mathbf{t}_{i1}$. Let $\Theta_p^f = [x_p^f, y_p^f, 1]^T$ be the position of pth point in the fth frame. Let $\mathbf{X}_p^f = Z_p^f \Theta_p^f$, where Z_p^f is the depth. Then, the motion equation between the fth frame and the first frame for point p is given by

$$Z_p^f \Theta_p^f = Z_p^1 \mathbf{M}_f \Theta_p^1 + \mathbf{V}_f. \qquad (6.1)$$

Subtracting (6.1) for p = i from the equation for p=j, we get

$$Z_i^f \Theta_i^f - Z_j^f \Theta_j^f = \mathbf{M}_f \mathbf{S}_{ij}, \; i, j = 1, \ldots, P, f = 2, \ldots, F, \qquad (6.2)$$

where

$$\mathbf{S}_{ij} = Z_i^1 \Theta_i^1 - Z_j^1 \Theta_j^1 \qquad (6.3)$$

is the vector connecting the ith and the jth points. Equation (6.2) 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 the following equation which constitutes the basic equation for solution

$$\begin{bmatrix} (\Theta_i^1 \times \Theta_j^1)^T \\ (\Theta_i^2 \times \Theta_j^2)^T \mathbf{M}_2 \\ \vdots \\ (\Theta_i^F \times \Theta_j^F)^T \mathbf{M}_F \end{bmatrix} \mathbf{S}_{ij} \equiv \mathbf{\Phi}_{ij} \mathbf{S}_{ij} = 0, \; i \neq j. \qquad (6.4)$$

For any given Φ_{ij}, the optimal solution of $\mathbf{S}_{ij}$ subject to $\|\mathbf{S}_{ij}\| = 1$ that minimizes

$$\epsilon_{ij} = \|\mathbf{\Phi}_{ij}\mathbf{S}_{ij}\|^2 = \mathbf{S}_{ij}^T \mathbf{\Phi}_{ij}^T \mathbf{\Phi}_{ij} \mathbf{S}_{ij} \qquad (6.5)$$

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}. The problem that remains is to estimate the rotation parameters and minimize λ_{ij}. However, in order to solve for $\mathbf{M}_f$, f = 2, ..., F, uniquely, we need to consider all the points simultaneously. Let

$$\Lambda = \sum_{i=j+1}^{P} \sum_{j=1}^{P-1} w_{ij}\lambda_{ij} = \sum_{i>j} w_{ij}\lambda_{ij}, \qquad (6.6)$$

where w_{ij} is a weighting factor. An optimal solution of $\mathbf{M}_f$, f = 2, ..., F, that minimizes Λ requires nonlinear search of the rotation parameters. If the polynomial rotation models are used, much less unknowns will be involved for long sequence motion than when the arbitrary motion model is used.

After $\mathbf{M}_f$, f = 2, ..., F, have been obtained somehow, depths and translation vectors can be determined in a closed form. First consider the depths in the first view. After $\mathbf{M}_f$, f = 2, ..., F, have been solved for, 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 of $\mathbf{S}_{ij}$ be $\hat{\mathbf{S}}_{ij}$. Then, because of equation (6.3) we have the following equation

$$Z_i^1 \Theta_i^1 - Z_j^1 \Theta_j^1 = \alpha_{ij} \hat{\mathbf{S}}_{ij}, \qquad (6.7)$$

where α_{ij} is some constant to be determined. Using $\hat{\mathbf{S}}_{ij}$ to cross-multiply both sides of Equation (6.7) we get the following equation for the depths

$$Z_i^1\left(\Theta_i^1 \times \hat{\mathbf{S}}_{ij}\right) - Z_j^1\left(\Theta_j^1 \times \hat{\mathbf{S}}_{ij}\right) = 0, \quad i > j, \qquad (6.8)$$

which allows a linear least squares solution of P depths to within a scalar from 3P(P-1)/2 equations:

$$\begin{bmatrix} \beta_{21} & \gamma_{21} & 0 & \cdots & 0 \\ \beta_{31} & 0 & \gamma_{31} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_{P1} & 0 & 0 & & \gamma_{P1} \\ 0 & \beta_{32} & \gamma_{32} & \cdots & 0 \\ 0 & \beta_{42} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \beta_{P2} & 0 & \cdots & \gamma_{P2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \beta_{P,P-1} & \gamma_{P,P-1} \end{bmatrix} \begin{bmatrix} Z_1^1 \\ Z_2^1 \\ \vdots \\ Z_P^1 \end{bmatrix} \equiv \mathbf{H}\mathbf{Z} = 0, \qquad (6.9)$$

where

$$\beta_{ij} = -\Theta_j^1 \times \hat{\mathbf{S}}_{ij}, \quad \gamma_{ij} = \Theta_i^1 \times \hat{\mathbf{S}}_{ij}, \quad i > j. \qquad (6.10)$$

Now consider depths in other views. Let $\mathbf{S}_{ij}$ be computed as in Equation (6.3), and let $\mathbf{X}_{ij}^f = \mathbf{M}_f \mathbf{S}_{ij}$. Then Equation (6.2) gives

$$Z_i^f \Theta_i^f - Z_j^f \Theta_j^f = \mathbf{X}_{ij}^f, \quad i, j = 1, \ldots, P. \qquad (6.11)$$

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

$$Z_i^f\left(\Theta_i^f \times \Theta_j^f\right) = \mathbf{X}_{ij}^f \times \Theta_j^f, \quad j = 1, \ldots, P. \tag{6.12}$$

Z_i^f can be solved for linearly from the above equation by a least squares method minimizing $\sum_{j=1}^{P} \left\|Z_i^f\left(\Theta_i^f \times \Theta_j^f\right) - \mathbf{X}_{ij}^f \times \Theta_j^f\right\|^2$. The solution is given in closed form by

$$Z_i^f = \frac{\mathbf{A}_i^f \cdot \mathbf{B}_i^f}{\mathbf{A}_i^f \cdot \mathbf{A}_i^f}, \quad i = 1, \ldots, P, \; f = 2, \ldots, F. \tag{6.13}$$

where

$$\mathbf{A}_i^f = \begin{bmatrix} \Theta_i^f \times \Theta_1^f \\ \vdots \\ \Theta_i^f \times \Theta_P^f \end{bmatrix}, \quad \mathbf{B}_i^f = \begin{bmatrix} \mathbf{X}_{i1}^f \times \Theta_1^f \\ \vdots \\ \mathbf{X}_{iP}^f \times \Theta_P^f \end{bmatrix}. \tag{6.14}$$

After the depths are all obtained, the optimal solution of translation vector $\mathbf{V}_f$ that minimizes $\sum_{p=1}^{P} \left\|Z_p^f \Theta_p^f - Z_p^1 \mathbf{M}_f \Theta_p^1 - \mathbf{V}_f\right\|^2$ is then given by

$$\mathbf{V}_f = \frac{1}{P}\sum_{p=1}^{P}\left(Z_p^f \Theta_p^f - Z_p^1 \mathbf{M}_f \Theta_p^1\right). \tag{6.15}$$

Since translations can be solved for 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 in good accuracy. However, if a suitable model applies, the results can still be improved by applying the estimates of $\mathbf{V}_f = \mathbf{t}_{i1}$ obtained as above to the model based estimation method for translation vectors discussed in the last section. An alternative solution of the translation vectors which allows polynomial motion models as well as arbitrary motion model is possible [6]. In that formulation, translation vectors instead of depths are solved for first.

7 Automatic Selection of Motion Models

A higher order model gives a smaller objective function value and needs a larger number of frames to stabilize the solution, but the correct model gives the best estimation results. For real application, it is important to select the right model for the given data. There seems to be no efficient way except to apply all models to the same data and see what happens. For a sequence of a few images, probably only low order models apply. In the following, we assume a sufficient number (say 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 estimation results.

i. If the minimum objective function value Λ obtained by the first model is smaller than those obtained by other models, the first model is used.

ii. Otherwise, obtain $\Omega_n = (\omega_X{}^n, \omega_Y{}^n, \omega_Z{}^n)$ for each n for the third and second models. If the distance between the two solutions for any n is larger than a threshold (say 0.3°, determined according to the system resolution), the third model is used.

iii. Else, obtain Ω_n for each n for the second and first models. If the distance between the two solutions at any time is larger than a threshold (say 0.3°), the second model is used. Otherwise, the first model is used.

A similar but somewhat different procedure is used for determining the translation model [6]. For translation, two thresholds are used: one on translation direction, and the other on magnitude of translation.

Using the above methods, the algorithm can automatically select the right model to solve for the motion parameters. It is worthwhile to note that the algorithm is so robust that even if a mistake[2] is made about the model, the resulting estimation is still acceptable.

8 Experimental Results

This section presents two examples with real image data for the algorithm using interframe matches. The system setup and algorithm parameters are described in [5][6]. 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 first example contains a sequence of 20 images. Figs. 1 (a) and (b) show the first and last images and the correspondences obtained. Figs. 1 (c) 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 ω_X^0=0.42°, ω_X^a=−0.08°, ω_X^b=0.008°, and a translation of constant acceleration along Z axis ($[0\ 0\ 1]^T$) with parameters t_Z^0=1.0, t_Z^a=−0.06, t_Z^b=0.01. The rotation also causes a small translation along Y axis since the rotation center is not at the optical center. The estimated rotation parameters are:

$$\Omega_0 = \begin{bmatrix} 0.42603 \\ -0.01581 \\ 0.02296 \end{bmatrix}, \; \Omega_a = \begin{bmatrix} 0.08152 \\ -0.00479 \\ 0.00432 \end{bmatrix}, \; \Omega_b = \begin{bmatrix} 0.00815 \\ -0.00031 \\ 0.00030 \end{bmatrix},$$

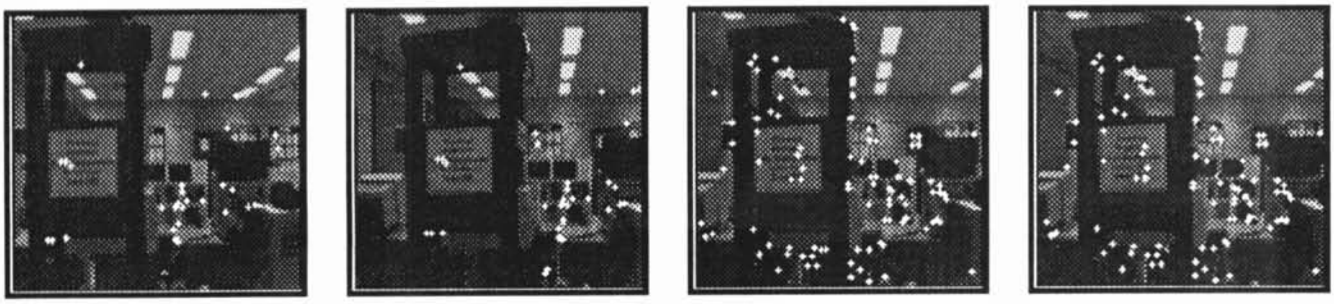
and the interframe translation direction vectors are, for example, $\mathbf{t}_{1,2}$ = [-0.03450, −0.06676, −0.99717], $\mathbf{t}_{10,11}$ = [-0.02747, −0.177239, −0.99194], $\mathbf{t}_{19,20}$ = [-0.00858, −0.17724, −0.98413]. Only two interframe translations are not accurate, which are $\mathbf{t}_{7,8}$ = [-0.34273, −0.31138, −0.88633], and $\mathbf{t}_{17,18}$ = [0.25909, 0.68218, −0.68374]. The estimation of interframe translation vectors is more vulnerable to noise in the image data.

The second example also contains a sequence of 20 images. Figs. 2 (a) and (b) show the first and last images and the correspondences obtained between them. Figs. 2 (c) 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 ω_Y^0 = 0.5°, ω_Y^a = 0.04°, ω_Y^b =

2 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.

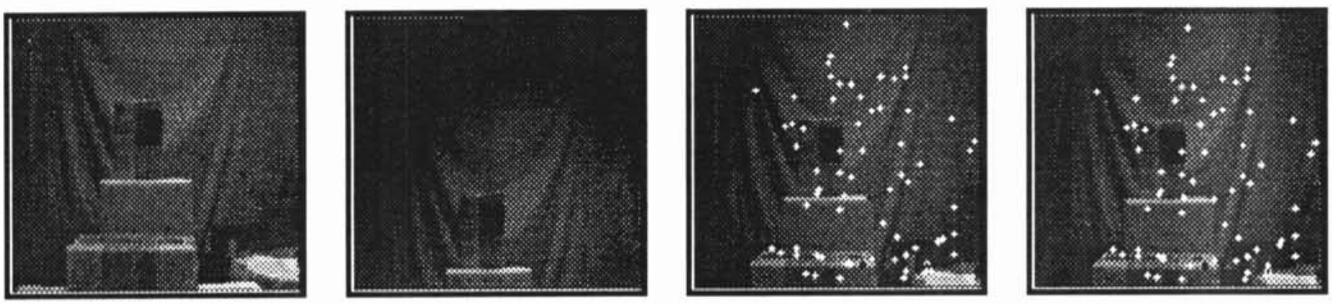
-0.003^o, and a translation of constant acceleration along X axis ($[1\ 0\ 0]^T$) with parameters $t_X^0 = 1.0$, $t_X^a = -0.04286$. However, in this example, the rotation causes a translation along Z axis that is comparable with the translation along X axis. Therefore, we do not know the ground truth of the translation directions. The estimated rotation parameters are:

$$\Omega_0 = \begin{bmatrix} 0.01917 \\ 0.37068 \\ 0.00017 \end{bmatrix}, \quad \Omega_a = \begin{bmatrix} -0.00432 \\ 0.03860 \\ -0.00282 \end{bmatrix}, \quad \Omega_b = \begin{bmatrix} 0.01774 \\ -0.00289 \\ -0.00004 \end{bmatrix}.$$



(a). The first image. (b). The nineteenth image. (c). The tenth image. (d). The eleventh 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 nineteenth image. (c). The tenth image. (d). The eleventh 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.

9 Summary

In this paper we have presented model-based algorithms for estimating motion parameters from a long sequence of images. The whole process, from images to feature points, to matching, and then to motion estimation, is fully automated. The algorithm automatically finds the proper model for a given motion and obtains optimal solution for the chosen model. The application of the algorithm using interframe matches 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. When point trajectories are available, the accuracy of the results should increase substantially.

REFERENCES

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

[2] T. Broida and R. Chellapa, "Estimating the kinematics and structure of a rigid object from a sequence of monocular images," *PAMI-13*, no. 6, pp. 497–513, 1991.

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

[4] S.-L. Iu and K. Wohn, "Estimation of general rigid motion from a long sequence of images," *Proc. of ICPR*, pp. 217–219, 1990.

[5] X. Hu and N. Ahuja, "Estimating motion of constant acceleration from image sequences," *Proc. 11th Inter. Conf. on Pattern Recognition*, Hague, Netherlands, 1992.

[6] X. Hu, *Perception of Shape and Motion*. Ph.d. thesis, Dept. of Electrical and Computer Engineering, University of Illinois, Urbana, IL, 1992.

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

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

[9] X. Hu and N. Ahuja, "Motion estimation under orthographic projection," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 848–853, 1991.