

A General Approach for Determining 3D Motion and Structure of Multiple Objects from Image Trajectories*

Tina Yu Tian and Mubarak Shah
Computer Vision Lab, Computer Science Department
University of Central Florida, Orlando, FL 32816
email: {tian, shah}@cs.ucf.edu

Abstract

We present a general approach to determine 3D motion and structure of multiple objects undergoing arbitrary motions. We segment the scene based on 3D motion parameters. First, the general motion model is fitted to each single trajectory. For this nonlinear fitting, initial estimates are obtained by a linear multiple motion SFM algorithm using the first two frames. Next, trajectories are clustered into groups corresponding to different moving objects. In our approach, discontinuous trajectories, resulting from occlusion, are also allowed. Finally, the multiple trajectory fitting is applied to each trajectory group to improve the estimates further. Our simulation results show that the proposed method is robust.

1 Introduction

Multiple motions are ubiquitous in the real world. Determining 3D motion and structure of multiple objects has been a challenging problem. In two-frame approach, since each moving object only occupies a small field of view and perspective effect is not obvious, orthographic or weak perspective projection model is more appropriate for motion and shape recovery. Since a long sequence is almost always available in motion analysis, extended image sequence with a large number of correspondence points can be used to provide more reliable 3D structure estimates and motion parameters as well. However, in a long motion sequence, each object can occupy a relatively large field of view over frames, since orthographic projection cannot deal with general motions, (excluding the motion translating towards or away from objects), we propose to use perspective projection model in a multi-frame approach.

Most of the existing multi-frame SFM algorithms (e.g. [3, 6, 2]) deal with a single moving object (ego-motion). For multiple motions, a straightforward method is to segment the displacement field or 2D trajectories first, and then apply an ego-motion SFM algorithm to each moving object. However, segmentation using displacement field itself cannot distinguish between real motion boundaries and depth discontinuities. Another approach for segmentation is based on the set of coherent motion parameters, independent of depth values. This approach (e.g. [1, 5, 7]) exploits 2D parametric motion approximations, ignoring the higher-order information of the displacement vector, and thus yields incorrect motion segmentation. Moreover, using a 2D motion model to segment a 3D scene can lead to ambiguities. Methods in [4] and [9] belong to multi-frame approaches for multiple motions. In [4], an orthographic factorization-based method was used to compute motion and structure for a single object, and split and merge processes were repeatedly performed to partition trajectories into groups corresponding to different objects. In [9], 3D line segments obtained from stereo were used as tokens, which greatly simplifies the problem. A kinematic model was fitted for each token by extended Kalman filter, and tokens were grouped into objects based on Mahalanobis distance of kinematic parameters.

In this paper, we present a general approach to determine 3D motion and structure of multiple objects from monocular image sequence. Our method consists of five modules. The first module generates trajectories from image correspondence of feature points by tracking over multiple frames. The second module computes the initial guess for the single trajectory fitting module by using two-frame multiple motion SFM algorithm (extended from [8]), in order to avoid being trapped into a local minimum and to speed up the convergence. The third module fits a general motion model to each trajectory, the fourth module groups

*The research reported in this paper was supported by NSF grants CDA-9122006 and IRI-9220768.

trajectories into sets corresponding to each moving object and merges the “broken” trajectories, caused by occlusion. Segmentation is performed based on 3D motion parameters to reduce the ambiguity inherently present in 2D segmentation of the 3D scenes. Finally, the fifth module applies the multiple trajectory fitting algorithm to each individual group of trajectories to reject noise and obtain more refined and accurate estimates of motion and structure. In this paper, we assume feature selection and tracking has been performed, so it is omitted for brevity.

2 Single Trajectory Fitting

2.1 Models and Parametrization

In our formulation, the motion of a 3D point is formulated as a rotation around an arbitrary axis in a camera-centered coordinate system, followed by a translation of the point with respect to the camera. The structure of the 3D point to be estimated is defined as $\mathbf{X}_0 = (X_0, Y_0, Z_0)^\top$, the 3D coordinates at time t_0 in the camera-centered coordinate system. We assume that motion is constant over time; rotation occurs about \mathbf{X}_0 , and translation is a constant $\mathbf{T} = (T_1, T_2, T_3)^\top$, where \top denotes transpose. Rotation matrix \mathbf{R} can be expressed in terms of quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)^\top$ in the following form:

$$\mathbf{R} = \begin{pmatrix} q_0^2 - q_1^2 - q_2^2 + q_3^2 & 2(q_0 q_1 + q_2 q_3) & 2(q_0 q_2 - q_1 q_3) \\ 2(q_0 q_1 - q_2 q_3) & -q_0^2 + q_1^2 - q_2^2 + q_3^2 & 2(q_1 q_2 + q_0 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_1 q_2 - q_0 q_3) & -q_0^2 - q_1^2 + q_2^2 + q_3^2 \end{pmatrix} \quad (1)$$

where \mathbf{R} is the function of quaternion \mathbf{q} , which is propagated in time. In (1), we omit the parameter t for clarity. The motion of a feature point is given by

$$\mathbf{X}(t) = \mathbf{R}(\mathbf{q}(t))\mathbf{X}_0 + t\mathbf{T}. \quad (2)$$

Assuming angular velocity $\mathbf{w} = (w_x, w_y, w_z)^\top$ is constant, quaternion $\mathbf{q}(t)$ propagates in time as follows:

$$\mathbf{q}(t) = \left(\frac{w_x}{w} \sin \frac{wt}{2}, \frac{w_y}{w} \sin \frac{wt}{2}, \frac{w_z}{w} \sin \frac{wt}{2}, \cos \frac{wt}{2} \right)^\top, \quad (3)$$

where the magnitude of the angular velocity is given by $w = \sqrt{w_x^2 + w_y^2 + w_z^2}$, and $w \neq 0$. If $w = 0$, the motion is pure translation.

For the imaging model, a central projection model is used. A 3D point $\mathbf{X} = (X_1, X_2, X_3)$ is projected to 2D point (x, y) in the image plane: $x = f \frac{X_1}{X_3}$, $y = f \frac{X_2}{X_3}$, where f is focal length. There is always

a scale factor which cannot be determined from a monocular sequence. Since we can easily check if $\mathbf{T} = \mathbf{0}$, we assume the scale factor is the translational speed $\|\mathbf{T}\|$, and \mathbf{T} is a unit vector, denoting translational direction, represented by spherical coordinates in terms of slant, θ_T , and tilt, ϕ_T :

$$\mathbf{T} = (\sin\theta_T \cos\phi_T, \sin\theta_T \sin\phi_T, \cos\theta_T)^\top. \quad (4)$$

The estimated structure is the relative structure, $\mathbf{X}_0 / \|\mathbf{T}\|$.

For a given 3D point, we have eight unknown parameters: five motion parameters (θ_T and ϕ_T for translation, and w_x, w_y, w_z for rotation), and three structure parameters, (X_0, Y_0 , and Z_0 for structure at time t_0). Let \mathbf{a} denote these unknowns: $\mathbf{a} = (\theta_T, \phi_T, w_x, w_y, w_z, X_0, Y_0, Z_0)^\top$. Now, the problem is to determine \mathbf{a} , given $\{\mathbf{x}_s, \mathbf{x}_{s+1}, \dots, \mathbf{x}_{s+N}\}$, where $\mathbf{x}_t = (x_t, y_t)^\top$, $t = s, \dots, s+N$, $s \geq 0$, a 2D trajectory of a 3D point undergoing translation and rotation.

To facilitate the minimization, we define our objective function as the error between the expected 3D positions of imaged and measured features, rather than as 2D projection error as in [3]. Let $\mathbf{h}_t(\mathbf{a})$ be the 3D coordinates of a point at time t , given parameter \mathbf{a} . Therefore, $\mathbf{h}_t(\mathbf{a})$ is given by:

$$\mathbf{h}_t(\mathbf{a}) = \mathbf{R}(\mathbf{q}(t))\mathbf{X}_0 + t\mathbf{T}. \quad (5)$$

Given image coordinates at time t , \mathbf{x}_t , and depth Z_t , its corresponding 3D coordinates are $\mathbf{X}_t = (Z_t x_t / f, Z_t y_t / f, Z_t)^\top$. We can think of Z_t as the depth at time t propagated in time from Z_0 . Let $\mathbf{p} = (0, 0, 1)$, then we have $Z_t = \mathbf{p} \mathbf{h}_t(\mathbf{a})$. Now, \mathbf{X}_t becomes

$$\mathbf{X}_t = \mathbf{p} \mathbf{h}_t(\mathbf{a}) \begin{pmatrix} x_t / f \\ y_t / f \\ 1 \end{pmatrix},$$

where (x_t, y_t) is known. The objective function to be minimized is given by:

$$E^2(\mathbf{a}) = \sum_{t=s}^{s+N} \|\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})\|^2, \quad (6)$$

where $\|\mathbf{X}\|$ denotes the Euclidean norm of vector \mathbf{X} .

2.2 Minimization

The Levenberg-Marquardt (L-M for short) algorithm is used to minimize the nonlinear function (6). Let $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7, \mathbf{a}_8)^\top$. The first and second partial derivative of the objective function with

respect to parameter \mathbf{a}_i , $i = 1, 2, \dots, 8$, must be computed.

Let \mathbf{R}_j denote the j th column of \mathbf{R} and \mathbf{R}_{3j} denote the element in the third row and the j th column of \mathbf{R} . We first compute the partial derivatives of the function $\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})$ in (6) with respect to the parameter \mathbf{a} as follows:

$$\frac{\partial}{\partial \mathbf{a}_i}(\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})) = t\mathbf{p} \frac{\partial \mathbf{T}}{\partial \mathbf{a}_i} \begin{pmatrix} x_t/f \\ y_t/f \\ 1 \end{pmatrix} - t \frac{\partial \mathbf{T}}{\partial \mathbf{a}_i}, \quad i = 1, 2, \quad (7)$$

$$\frac{\partial}{\partial \mathbf{a}_i}(\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})) = \mathbf{p} \frac{\partial \mathbf{R}}{\partial \mathbf{a}_i} \mathbf{X}_0 \begin{pmatrix} x_t/f \\ y_t/f \\ 1 \end{pmatrix} - \frac{\partial \mathbf{R}}{\partial \mathbf{a}_i} \mathbf{X}_0, \quad i = 3, 4, 5, \quad (8)$$

$$\frac{\partial}{\partial \mathbf{a}_{5+i}}(\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})) = \mathbf{R}_{3i} \begin{pmatrix} x_t/f \\ y_t/f \\ 1 \end{pmatrix} - \mathbf{R}_i, \quad i = 1, 2, 3, \quad (9)$$

where

$$\frac{\partial \mathbf{T}}{\partial \mathbf{a}_1} = \frac{\partial \mathbf{T}}{\partial \theta_T} = \begin{pmatrix} \cos \theta_T & \cos \phi_T \\ \cos \theta_T & \sin \phi_T \\ -\sin \theta_T & \end{pmatrix},$$

$$\frac{\partial \mathbf{T}}{\partial \mathbf{a}_2} = \frac{\partial \mathbf{T}}{\partial \phi_T} = \begin{pmatrix} -\sin \theta_T & \sin \phi_T \\ \sin \theta_T & \cos \phi_T \\ 0 & \end{pmatrix},$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{a}_3} = \frac{\partial \mathbf{R}}{\partial w_x} = \sum_{k=0}^3 D_k \frac{\partial q_k}{\partial w_x},$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{a}_4} = \frac{\partial \mathbf{R}}{\partial w_y} = \sum_{k=0}^3 D_k \frac{\partial q_k}{\partial w_y},$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{a}_5} = \frac{\partial \mathbf{R}}{\partial w_z} = \sum_{k=0}^3 D_k \frac{\partial q_k}{\partial w_z},$$

where $D_k = \partial \mathbf{R} / \partial q_k$, $k = 0, 1, 2, 3$, are given by

$$\mathbf{D}_0 = 2 \begin{pmatrix} q_0 & q_1 & q_2 \\ q_1 & -q_0 & q_3 \\ q_2 & -q_3 & -q_0 \end{pmatrix}, \quad \mathbf{D}_1 = 2 \begin{pmatrix} -q_1 & q_0 & -q_3 \\ q_0 & q_1 & q_2 \\ q_3 & q_2 & -q_1 \end{pmatrix},$$

$$\mathbf{D}_2 = 2 \begin{pmatrix} -q_2 & q_3 & q_0 \\ -q_3 & -q_2 & q_1 \\ q_0 & q_1 & q_2 \end{pmatrix}, \quad \mathbf{D}_3 = 2 \begin{pmatrix} q_3 & q_2 & -q_1 \\ -q_2 & q_3 & q_0 \\ q_1 & -q_0 & -q_3 \end{pmatrix}.$$

Now, $\partial q_k / \partial \mathbf{a}_i$, $k = 0, 1, 2, 3$, $i = 3, 4, 5$, are given by

$$\frac{\partial q_0}{\partial w_x} = \left(\frac{1}{w} - \frac{w^2}{w^3}\right) \sin \frac{wt}{2} + \frac{t}{2} \frac{w^2}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_1}{\partial w_x} = -\frac{w_x w_y}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_x w_y}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_2}{\partial w_x} = -\frac{w_x w_z}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_x w_z}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_3}{\partial w_x} = -\frac{t}{2} \frac{w_x}{w} \sin \frac{wt}{2},$$

$$\frac{\partial q_0}{\partial w_y} = -\frac{w_x w_y}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_x w_y}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_1}{\partial w_y} = \left(\frac{1}{w} - \frac{w^2}{w^3}\right) \sin \frac{wt}{2} + \frac{t}{2} \frac{w^2}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_2}{\partial w_y} = -\frac{w_y w_z}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_y w_z}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_3}{\partial w_y} = -\frac{t}{2} \frac{w_y}{w} \sin \frac{wt}{2},$$

$$\frac{\partial q_0}{\partial w_z} = -\frac{w_x w_z}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_x w_z}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_1}{\partial w_z} = -\frac{w_y w_z}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_y w_z}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_2}{\partial w_z} = \left(\frac{1}{w} - \frac{w^2}{w^3}\right) \sin \frac{wt}{2} + \frac{t}{2} \frac{w^2}{w^2} \cos \frac{wt}{2},$$

$$\frac{\partial q_3}{\partial w_z} = -\frac{t}{2} \frac{w_z}{w} \sin \frac{wt}{2}.$$

Then the first partial derivatives of the function E^2 with respect to \mathbf{a}_i are given by:

$$\frac{\partial E^2}{\partial \mathbf{a}_i} = 2 \sum_{t=s}^{s+N} (\mathbf{X}_t - \mathbf{h}_t(\mathbf{a}))^\top \frac{\partial}{\partial \mathbf{a}_i} (\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})), \quad i = 1, 2, \dots, 8. \quad (10)$$

The second partial derivatives are obtained by ignoring the second derivatives of the model function:

$$\frac{\partial^2 E^2}{\partial \mathbf{a}_i \partial \mathbf{a}_j} \approx 2 \sum_{t=s}^{s+N} \frac{\partial}{\partial \mathbf{a}_i} (\mathbf{X}_t - \mathbf{h}_t(\mathbf{a}))^\top \frac{\partial}{\partial \mathbf{a}_j} (\mathbf{X}_t - \mathbf{h}_t(\mathbf{a})), \quad i, j = 1, \dots, 8. \quad (11)$$

Let $\beta_i \stackrel{\text{def}}{=} -\frac{1}{2} \frac{\partial E^2}{\partial \mathbf{a}_i}$, and $\alpha_{ij} \stackrel{\text{def}}{=} \frac{1}{2} \frac{\partial E^2}{\partial \mathbf{a}_i \partial \mathbf{a}_j}$, the elements in matrix $[\alpha]$. The minimization problem is reduced to iteratively solving the following linear equation:

$$\sum_{l=1}^m \alpha_{kl} \delta a_l = \beta_k, \quad (12)$$

where m is the number of unknown parameters. Here $m = 8$. The algorithm for single trajectory fitting is summarized as follows:

Algorithm: TrajFit

1. Compute $E^2(\mathbf{a})$ in (6). Set $\lambda = 0.001$.
2. Compute β_i and α_{ij} , where $i, j = 1, 2, \dots, 8$, using equations (10) and (11), respectively.
3. Compute matrix $[\alpha]$ by augmenting its diagonal elements: $\alpha'_{jj} = \alpha_{jj}(1 + \lambda)$, and $\alpha'_{jk} = \alpha_{jk}$.
4. Solve (12) for $\delta(\mathbf{a})$ and evaluate $E^2(\mathbf{a} + \delta \mathbf{a})$.
5. If $E^2(\mathbf{a} + \delta \mathbf{a}) \geq E^2(\mathbf{a})$, increase λ by a factor and go back to 2.
6. If $E^2(\mathbf{a} + \delta \mathbf{a}) < E^2(\mathbf{a})$, decrease λ by a factor, update the trial solution $\mathbf{a} \leftarrow \mathbf{a} + \delta \mathbf{a}$, and go back to 2.

3 Two-Frame SFM Algorithm for Multiple Motions

Since nonlinear optimization is involved in single trajectory fitting, we first apply a closed-form SFM algorithm to obtain an initial guess to ensure the

faster convergence. There are a number of algorithms which can provide a closed-form solution for motion and structure estimation. We have chosen Weng et al.’s linear algorithm [8] based on epipolar constraint, because of its simplicity, and have extended their method to deal with multiple motions. We subdivide the image into overlapping patches. The algorithm is as follows:

Algorithm: InitGuess

1. For each overlapping patch,
 - (a) Count the number of feature points, n , inside the patch.
 - (b) If $n \geq 8$, apply Weng et al.’s algorithm [8] to the set of feature points in the patch, and compute motion parameters.
 - (c) If the smallest eigenvalue of $\mathbf{A}^\top \mathbf{A}$, (representing the residual error of the optimization function $\mathbf{A}^\top \mathbf{A}$, where \mathbf{A} is given in the following form is large (this implies that the patch contains multiple motions), eliminate the patch from further analysis.

$$\mathbf{A} = \begin{pmatrix} u_1 u'_1 & u_1 v'_1 & u_1 f & v_1 u'_1 & v_1 v'_1 & v_1 f & u'_1 f & v'_1 f & f^2 \\ u_2 u'_2 & u_2 v'_2 & u_2 f & v_2 u'_2 & v_2 v'_2 & v_2 f & u'_2 f & v'_2 f & f^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n u'_n & u_n v'_n & u_n f & v_n u'_n & v_n v'_n & v_n f & u'_n f & v'_n f & f^2 \end{pmatrix}$$

where $\mathbf{x}_i = (u_i, v_i, f)^\top$, $\mathbf{x}'_i = (u'_i, v'_i, f)^\top$, $i = 1, 2, \dots, n$, denote the n motion correspondences at t_k and t_{k+1} , and f is focal length.

- (d) Save the motion parameters into feature vector, \mathbf{m}_i , where i is patch index.
2. Segment the scene into different moving objects based on the computed motion parameters.
 - (a) Normalize each component of feature vector to ensure each component has equal range. (This is a necessary preprocessing step for k-means clustering technique because k-means clustering is based on Euclidean distance.)
 - (b) Apply k-means clustering algorithm to the set of $\{\mathbf{m}_i\}$.
 - (c) Convert the feature vectors at centroids of K clusters in the original scale, and compute motion parameters for K moving objects.
3. Determine 3D structure for K moving objects.

For each feature point i in the image,

- (a) For each set of motion parameters $(\mathbf{T}_j, \mathbf{R}_j)$, $j = 0, \dots, K$, compute the structure $(\mathbf{X}'_i, \mathbf{X}_i)$ at t_{k+1} and t_k , such that $\|\mathbf{X}'_i - \mathbf{R}_j \mathbf{X}_i - \mathbf{T}_j\| \rightarrow \min$, then compute the corresponding residual error $Res_j = \|\mathbf{X}'_i - \mathbf{R}_j \mathbf{X}_i - \mathbf{T}_j\|$.
- (b) Assign feature point i to Segment s , where $Res_s = \min\{Res_j, j = 1, \dots, K\}$, and $(\mathbf{X}'_i, \mathbf{X}_i)$ is given by the corresponding $(\mathbf{T}_s, \mathbf{R}_s)$.
- (c) If $k > 0$, compute the structure at t_0 from the motion parameters and the structure at t_k by performing the inverse transform:

$$\mathbf{X}(t_0) = \mathbf{R}^\top(t_k)(\mathbf{X}(t_k) - t_k \mathbf{T}),$$

where $\mathbf{R}(t_k)$ is rotation matrix propagated at t_k , and $\mathbf{R}^{-1}(t_k) = \mathbf{R}^\top(t_k)$.

In the k-means algorithm, all the feature dimensions should be independent. We chose six dimensional feature space $\{\text{speed}, \theta_T, \phi_T, w_x, w_y, w_z\}$, where translational *speed* could be 0 or 1, (denoting pure rotation and nonzero translation, respectively). θ_T and ϕ_T represent translation direction, and w_x, w_y , and w_z represent angular rotation velocity. The mapping from quaternion \mathbf{q} to (w_x, w_y, w_z) , and translation \mathbf{T} to (θ_T, ϕ_T) are according to equations (3) and (4), respectively.

4 Trajectory Grouping

After performing trajectory fitting for each trajectory, starting at any time instant, we segment the trajectories into groups corresponding to moving objects based on the computed motion parameters. When the objects move, some features disappear due to self-occlusion or mutual occlusion among multiple objects, while new features appear. Our task is to identify whether a trajectory segment, starting at instant $t > t_0$, belongs to a new feature or a feature which has been occluded for a while. If it is not a new feature, we must merge the “broken” trajectory segments, and the merged trajectory will be used to refine the estimates of motion and structure at the next stage. The trajectory grouping algorithm is as follows:

Algorithm: TrajGroup

1. Fit the general motion model to each trajectory, and save the motion parameters in feature vector, \mathbf{m}_i , where i is trajectory index.
2. Cluster the trajectories in groups corresponding to different moving objects based on the computed motion parameters, following the same procedure in Step 2 of Algorithm InitGuess.

3. Merge the “broken” trajectories.

For each trajectory, say i , starting at $t > t_0$,

- For each trajectory, say j , which belongs to the same group as Trajectory i , and either starts at t_0 or has been declared as a new feature, compute the residual error $Res_j = \| \mathbf{X}_{i0} - \mathbf{X}_{j0} \|$, where \mathbf{X}_{i0} and \mathbf{X}_{j0} denote the structure of feature i and j at t_0 , respectively.
- Record the minimal residual error and the corresponding Trajectory k , such that $Res_k = \min\{Res_j, j = 1, \dots, K\}$.
- If Res_k is large, record Trajectory i belongs to a new feature, otherwise, link Trajectory i to Trajectory k .

5 Multiple Trajectory Fitting

In order to reject noise and obtain more accurate estimates of motion and structure of the objects, we apply the multiple trajectory fitting algorithm to each group of trajectories.

The objective function to be minimized is given by:

$$E^2 = \sum_{t=0}^N \sum_{i=1}^M \| \mathbf{X}_i(t) - \mathbf{h}_t(\mathbf{m}, \mathbf{X}_{i0}) \|^2, \quad (13)$$

where N is the number of the frames and M is the number of the features, and

$$\mathbf{h}_t(\mathbf{m}, \mathbf{X}_{i0}) = \mathbf{R}(\mathbf{q}(t))\mathbf{X}_{i0} + t\mathbf{T}, \quad (14)$$

where $\mathbf{X}_{i0} = (X_{i0}, Y_{i0}, Z_{i0})^\top$ denotes structure of feature point i at t_0 , and

$$\mathbf{X}_i(t) = \mathbf{p} \mathbf{h}_t(\mathbf{m}, \mathbf{X}_{i0}) \begin{pmatrix} x_{it}/f \\ y_{it}/f \\ 1 \end{pmatrix},$$

where (x_{it}, y_{it}) is image coordinates.

For this problem, we have $3M + 5$ unknown parameters; five are motion parameters for this moving object, and three structure parameters for each of M feature points. Let \mathbf{a} denote these unknowns: $\mathbf{a} = (\mathbf{a}_1, \dots, \mathbf{a}_{3M+5})^\top = (\theta_T, \phi_T, w_x, w_y, w_z, X_{10}, Y_{10}, Z_{10}, \dots, X_{M0}, Y_{M0}, Z_{M0})^\top$. L-M method is used for this minimization also.

6 Results

In this section, we present the simulation results. We first demonstrate the proposed algorithm for determining motion and structure of multiple moving

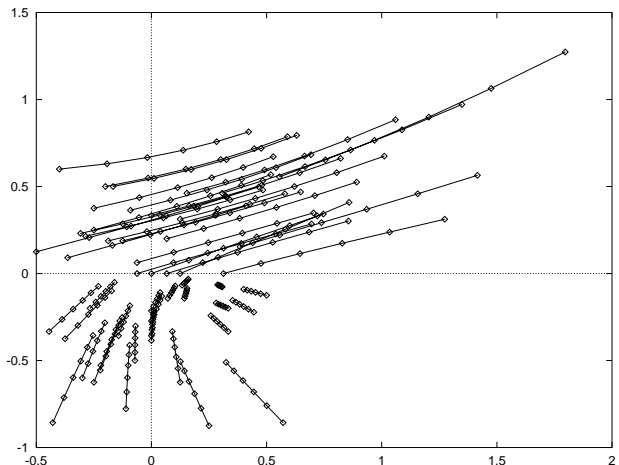


Figure 1: Generated trajectories of two moving objects.

objects, then show the performance of the algorithm for noisy data.

We assume two objects, namely the Upper Object, O_U , and the Lower Object, O_L , are present in the scene. The feature points of each object are generated randomly according to the uniform distribution in each transparent cube of $10 \times 7 \times 10$. The centers of the cubes are the centers of the objects at time t_0 , which are at $(0, 3, 11)$ and $(0, -4, 11)$, respectively. Focal length is assumed to be 1. In this experiment, we assume the two objects undergo rotation $w_U = (1.5^0, -8^0, 1^0)^\top$, and $w_L = (0^0, 0^0, 3^0)^\top$, followed by translation $\mathbf{T}_U = (0.354, 0.612, 0.707)^\top$, $\mathbf{T}_L = (0.296, 0.171, 0.940)^\top$, respectively. Twenty-five feature points from each object are tracked over five frames. The trajectories are shown in Figure 1. The image size is 2×2 . The overlapping patches of 0.5×0.5 are used in the two-frame SFM algorithm (InitGuess). The algorithm follows the steps indicated by the five modules. For noise-free data, the correct segmentation, motion and structure of each object are obtained.

The proposed algorithm was evaluated against the noise also. In this experiment, we only studied a single moving object, since the segmentation results will be correct if the single trajectory fitting results are close to the ground truth. A set of feature points was generated randomly from a transparent cube of $10 \times 10 \times 10$, centered at $(0, 0, 11)$. The object underwent rotation $w = (1.5^0, -2^0, 0^0)^\top$ and translation $\mathbf{T} = (0.707, 1.225, 1.414)^\top$. Twelve features were

tracked over one hundred frames. We added uniform random noise to each of the feature points. In this experiment, the noise level 0.01 corresponds to adding ± 0.87 pixel noise, and the noise level 0.09 corresponds to ± 8 pixel noise.

The errors associated with translation, rotation, and depth were evaluated. Translation error is defined as the angle between the actual and estimated translational direction, and the rotation (or depth) error is defined as the relative error, i.e. the norm of the error vector (or scalar value) divided by the norm of the actual vector (or scalar value). We evaluated both single trajectory fitting and multiple trajectory fitting. All the errors in the single trajectory fitting and the depth error in the multiple trajectory fitting are calculated as the average errors over all the trajectories in the entire image. Figure 2 illustrates the estimate errors as a function of the noise level. We observed in these experiments that providing initial guess from two-frame SFM algorithm indeed improved the results. The reason is that the single trajectory fitting algorithm is easily trapped into a local minimum with the noisy data. Also, the good initial estimate speeds up the convergence to the correct estimate, especially for the nonlinear minimization. Figure 2 also shows that the multiple trajectory fitting improves the estimates further. The experiments demonstrated that the proposed algorithm is a general and robust approach, in which all the modules are indispensable.

7 Conclusion

We present a general approach to determine 3D motion and structure of multiple objects undergoing arbitrary motions. Our simulation results show that the proposed method is robust.

Acknowledgements: We would like to thank Prof. David Heeger for helpful discussions.

References

- [1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *PAMI*, 7:384–401, 1985.
- [2] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *PAMI*, 15:602–605, 1993.
- [3] T. Broida and R. Chellappa. Estimating the kinematics and structure of a rigid object from a se-

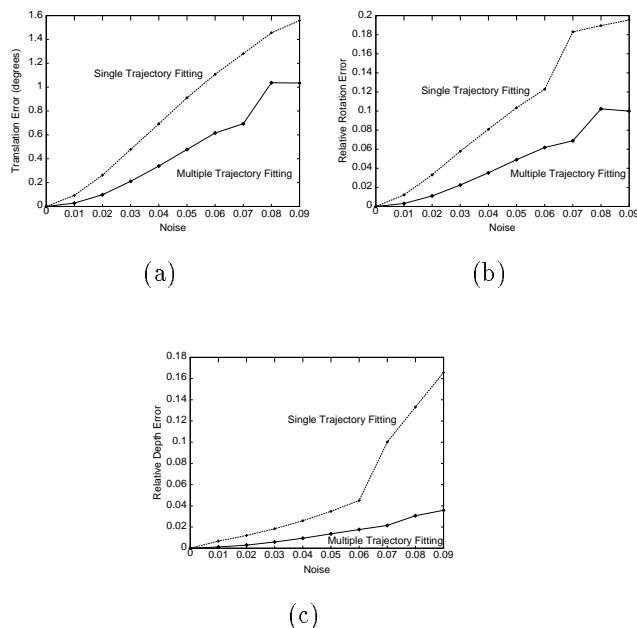


Figure 2: Errors in motion and depth estimates against different noise levels. (a) Translation error. (b) Relative rotation error. (c) Relative depth error.

quence of monocular images. *PAMI*, 13:497–513, 1991.

- [4] C. Debrunner, and N. Ahuja. Motion and Structure Factorization and Segmentation of Long Multiple Motion Image Sequences. In *ECCV'92*, pp. 217–221, 1992.
- [5] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV'92*, pp. 282–287, 1992.
- [6] H. Sawhney, J. Oliensis, and A. Hanson. Image description and 3-d reconstruction from image trajectories of rotational motion. *PAMI*, 15:885–898, 1993.
- [7] J. Wang and E. Adelson. Layer representation for motion analysis. In *CVPR'93*, pp. 361–366, 1993.
- [8] J. Weng, T. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *PAMI*, 11:451–476, 1989.
- [9] Z. Zhang and O. Faugeras. Three-Dimensional Motion Computation and Object Segmentation in a Long Sequence of Stereo Frames. *International Journal of Computer Vision*, 7:3, 211–241, 1992.