Computing Motion and Structure of Multiple Objects from Image Trajectories

Tina Yu Tian and Mubarak Shah
Computer Science Department
University of Central Florida
Orlando, FL 32816

Abstract

We present an approach to determine 3D structure and motion of *multiple* objects. 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 provided by a linear multiple motion SFM algorithm from 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. Applications of the proposed method to both synthetic and real image sequences are demonstrated with promising results.

Key Words: Motion, Shape representation and recovery, Trajectories, multiple object motion.

1 Introduction

Determining 3D motion and structure from monocular images has been a challenging problem for decades. All the existing structure-from-motion algorithms are highly sensitive to errors in the image measurements, since this sensitivity is inherent in the problem. In two-frame approaches, 3-D structure at each point is only determined locally from a single displacement vector or a pair of motion correspondence points. Thus, even a small amount of input noise makes its structure estimate inaccurate. Since a long sequence is almost always available in motion analysis, extended image sequence with large number of correspondence points or displacement vectors can be used to provide more reliable 3D structure estimates.

Either batch methods or recursive methods, such as Iterated Extended Kalman Filtering (IEKF), can be used for the motion sequences to solve the SFM problem. A batch method iteratively adjusts the estimate based on all the data until the minimum of the objective function is reached, while a recursive method starts with an initial estimate and refines its estimate according to the new data at each time instant. For highly nonlinear systems, batch methods are more stable and provide better estimates when the initial guess is inaccurate and high level of noise is present, since more information is extracted from the data. A theoretical comparison between batch approach and IEKF approach is given in [25].

Most existing multi-frame SFM algorithms [6, 14, 15, 16, 23, 26] deal with a single moving object (ego-motion). However, motion in an image sequence can be produced by a camera moving in an environment and/or several independently moving objects. Multiple motions are more realistic in the real world. In this paper, we attempt to solve the SFM problem for multiple moving objects using multiple frames.

For multiple-motion analysis, the essential part is segmentation of independently moving objects. One approach to segmentation is based on the set of coherent 2D motion parameters, independent of depth values. This approach [2, 4, 11, 22] exploits 2D parametric motion approximations, such as affine transformation and projective transformation. However, affine transformation is not always valid when the moving object is relatively large and close to the camera. Projective transformation [1] is based on the assumption that the entire scene consists of piecewise planar surfaces, which is not always true. These parametric models ignore the higher-order optical flow information, and thus may yield incorrect motion segmentation (e.g. over-segmentation). Moreover, using a 2D motion model to segment a 3D scene might lead to ambiguities.

In this paper, we propose a method of recovering structure and motion of multiple objects from image trajectories. Instead of segmenting the scene based on 2D trajectory information, we perform a 3D segmentation based on 3D motion parameters, which can be applied to arbitrary motions and also can reduce the ambiguity inherently present in 2D segmentation of the 3D scene. The initial estimate is provided by a linear two-frame multiple motion SFM algorithm based on the first two frames, in order to avoid being trapped into a local minimum and to speed up the convergence. On the other hand, unlike the approaches in [23, 15] which integrate the instantaneous estimates of every two consecutive frames over time, we do not heavily rely on the initial guess obtained from two frame estimation. In our approach, the estimate is refined over time. For a single trajectory fitting, rather than fitting a specific model, (for example, ballistic model, or conic model [16]), we fit the general motion model to allow an arbitrary motion.

In the next section, we review some previous work on multi-frame structure-from-motion for a single moving object and 3D motion segmentation in multi-motions analysis. Then we present our method and describe each module in successive sections. In Section 8, we demonstrate the proposed method on both synthetic and real images. Finally, we conclude in Section 9.

2 Previous Work

In this section, previous work on structure-from-motion for a single moving object using multiple-frames as well as 3D motion segmentation is reviewed.

2.1 Multiple-frame Methods for Structure-from-motion

Weng et al. [23] and Soatto et al. [15] computed the instantaneous motion and structure from two consecutive frames, then integrated the instantaneous estimates over time by fitting a processional model and Extended Kalman Filter (EKF), respectively. Both methods suffer from the inherent instabilities of two-frame estimates. Broida et al. [5, 6], Sawhney et al. [16] and Azarbayejani and Pentland [14] assumed that the object is rigid and dealt with trajectories tracked over multiple-frames from image correspondence of feature points. In Broida et al.'s methods an object-centered coordinate system is used to define the structure of the 3-D feature points. The object motion is simply decomposed into the rotation occurring about the origin of this system, and the translation of its origin with respect to the camera coordinate system. In [5], the EKF formulation was applied to the SFM problem, while in [6] the objective function was formulated as the error between the expected positions of observed and measured features, and the Conjugate Gradient algorithm was used to obtain motion parameters and object structure. The latter method is more stable, since batch approaches are more powerful than sequential approaches for nonlinear optimization.

Azarbayejani and Pentland [14] also used an object-centered coordinate system in their formulation, and applied EKF formulation to determine object structure. The results obtained from both Broida et al. and Azarbayejani and Pentland's methods are quite good. However, both methods consider a highly nonlinear minimization problem, in which the initial estimate is crucial. If the initial estimate is not within the convergence area of the minimization function, the algorithm may converge to a local minimum. Consequently, the accuracy of their method depends on the initial estimates.

In Sawhney et al.'s method [16], the motion is restricted to rotation only. They proposed a two-stage algorithm to this restricted motion. First they applied the conic fitting to each trajectory, then grouped similar trajectories, and finally computed, in a close form, the 3-D structure of the object from the image trajectories. However, their method cannot be applied to general motion including translation.

Xiong and Shafer [26] presented an EKF-based method for recovering dense structure from a dense optical flow sequence. Most traditional feature-based approaches (e.g. [6], [14]) cannot be extended to compute dense structure due to impractical computational complexities. In this method, by decomposing structure uncertainty information into independent and correlated parts and using principal component analysis, these complexities are reduced from $O(N^2)$ to $O(N)$, where N is the number of image pixels. The initial motion of every frame is computed independently to allow discontinuous motion. Dynamic motion parameterization is proposed to make the optical flow equally sensitive to each motion parameter, and to stabilize numerical computations.

2.2 3-D Motion Segmentation

The previous work on 3D motion segmentation is divided into the methods assuming: orthographic projection, and perspective projection.

Orthographic Projection:

Thompson et al. [18] combined an orthographic structure-from-motion algorithm and a least median squares (LMedS) method to solve for the relative motion of camera and background, rejecting outliers corresponding to moving objects. Since LMedS regression assumes that at least half of the data points are inliers, this scheme cannot be applied to any scene containing more than two moving objects. Tomasi and Kanade [20] proposed that under orthographic projection the measurement matrix, $\mathbf{W}$, of feature trajectories can be factored into motion matrix and shape matrix, $\mathbf{V}$, using singular value decomposition. Under this framework, Gear [10] and Costeira and Kanade [8] developed the segmentation methods. Each column of $\mathbf{W}$ represents a single feature point tracked over M frames. The set of feature points associated with a single object lies in a four- (or less) dimensional subspace of the $2M$-d space of column vectors. Thus the task of segmentation becomes identifying these subspaces and the column vectors that lie in them. This corresponds to finding reduced row echelon form of $\mathbf{W}$, by applying Gauss-Jordan elimination to the rows of $\mathbf{W}$ with partial pivoting, or QR reduction with column pivoting [10]. In [8], a "shape interaction matrix", $\mathbf{Q}$, is constructed as $\mathbf{V}\mathbf{V}^T$. $\mathbf{Q_{ij}}$, element of $\mathbf{Q}$, is zero only when feature trajectories i and j belong to different objects. Thus, segmentation is performed by putting $\mathbf{Q}$ in block-diagonal form, where each block represents a moving object.

Perspective Projection:

Torr and Murray [21] proposed a stochastic approach to segmentation using Fundamental Matrix (F) encapsulating the epipolar constraint. The method begins with generating hypothetical clusters by randomly sampling a subset of feature-correspondences and using them to calculate a solution F. All feature-pairs consistent with the solution are included in the cluster, by a *t-test*. Solutions are then pruned or merged. Finally, using an *integer programming* technique, the clusters are partitioned to form a segmentation. MacLean et al. [13] used mixture models to model multiple motion processes and applied EM algorithm to segmentation from linear constraints on 3-D translation and bilinear constraints on 3-D

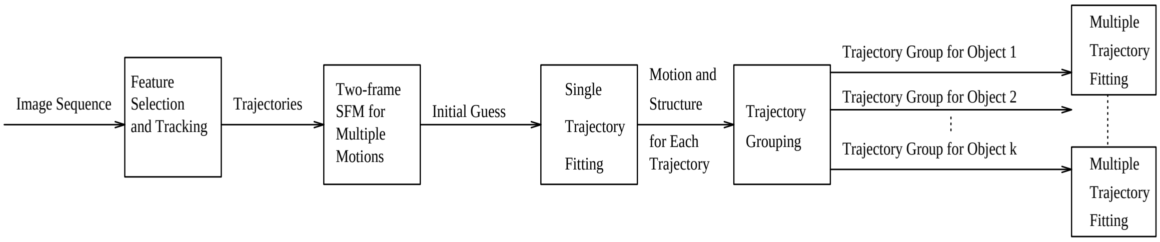


Figure 1: System Structure.

translation and rotation [12]. The result of clustering from the linear constraints are then used as an initial guess for parameter fitting using bilinear constraints. It is difficult to determine the number of clusters and the initial parameters for the EM algorithm. Some researchers [1, 3, 7] suggested applying the Hough transform to motion estimation and segmentation. The advantages of the Hough transform are that it is relatively insensitive to noise and more robust being a global approach, and the multiple local maxima in the parameter space naturally correspond to independently moving objects. However, it is a two-frame approach.

3 Our Method

Our method consists of five modules shown in Figure 1. 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 [24]), in order to avoid being trapped into a local minimum and to converge faster. The third module fits a general motion model to each trajectory, the fourth module groups 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 scene. Finally, the fifth module applies the multiple trajectory fitting algorithm to each individual group of trajectories to reject noise and obtain more accurate estimates of motion and structure.

Next sections deal with the description of several modules in our approach.

4 Single Trajectory Fitting

4.1 Models and Parametrization

In this work, the motion of a 3D point is formulated as a rotation around an arbitrary rotation axis in a camera-centered coordinate system, followed by a translation of the point with respect to the camera. The structure of the 3-D point to be estimated is defined as $\mathbf{X}_0 = (\mathrm{X}_0, \mathrm{Y}_0, \mathrm{Z}_0)^\top$, the 3-D coordinates at time t_0 in the camera-centered coordinate system. It is assumed that motion is constant over time; rotation occurs about $\mathbf{X}_0$, and translation is a constant $\mathbf{T} = (\mathrm{T}_1, \mathrm{T}_2, \mathrm{T}_3)^\top$, where $\top$ denotes transpose. Rotation matrix $\mathbf{R}$ can be expressed in terms of quaternion $\mathbf{q} = (\mathrm{q}_0, \mathrm{q}_1, \mathrm{q}_2, \mathrm{q}_3)^\top$ in the following form:

$$\mathbf{R} = \begin{pmatrix} q_0^2 - q_1^2 - q_2^2 + q_3^2 & 2(q_0q_1 + q_2q_3) & 2(q_0q_2 - q_1q_3) \\ 2(q_0q_1 - q_2q_3) & -q_0^2 + q_1^2 - q_2^2 + q_3^2 & 2(q_1q_2 + q_0q_3) \\ 2(q_0q_2 + q_1q_3) & 2(q_1q_2 - q_0q_3) & -q_0^2 - q_1^2 + q_2^2 + q_3^2 \end{pmatrix}, \tag{1}$$

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

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

Assuming angular velocity $\mathbf{w} = (w_\mathrm{x}, w_\mathrm{y}, w_\mathrm{z})^\top$ is constant, quaternion $\mathbf{q}(\mathrm{t})$ propagates in time as follows:

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

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.

A central projection is used for the imaging model. A 3D point, $\mathbf{X} = (\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3)$, 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 whether $\mathbf{T} = \mathbf{0}$ can be easily checked, it is assumed that the scale factor is the translational speed $\parallel \mathbf{T} \parallel$, and $\mathbf{T}$ is a unit vector, denoting translational direction, represented by spherical coordinates in terms of slant, θ_T, and tilt, ϕ_T:

$$\mathbf{T} = (\sin\theta_\mathrm{T}\cos\phi_\mathrm{T}, \sin\theta_\mathrm{T}\sin\phi_\mathrm{T}, \cos\theta_\mathrm{T})^\top. \tag{4}$$

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

For a given 3-D point, there are six unknown parameters: five motion parameters (θ_T and ϕ_T for translation, and w_x, w_y, w_z for rotation), and one depth parameter, Z_0. Other structure parameters at time t_0, X_0 and Y_0, can be obtained from Z_0 and image feature coordinates. Let $\mathbf{a}$ denote these unknowns: $\mathbf{a} = (\theta_\mathrm{T}, \phi_\mathrm{T}, \mathrm{w_x}, \mathrm{w_y}, \mathrm{w_z}, \mathrm{Z_0})^\top$. Now, the problem is to determine $\mathbf{a}$, given $\{\mathbf{x}_\mathrm{s}, \mathbf{x}_{\mathrm{s}+1}, \cdots, \mathbf{x}_{\mathrm{s}+\mathrm{N}}\}$, where $\mathbf{x}_\mathrm{t} = (\mathrm{x_t}, \mathrm{y_t})^\top, \mathrm{t} = \mathrm{s}, \cdots, \mathrm{s}+\mathrm{N}$, $s \geq 0$, a 2-D trajectory of a 3-D point undergoing translation and rotation.

To facilitate the minimization, our objective function is defined as the error between the expected 3-D positions of imaged and measured features, rather than as 2-D projection error as in [6]. Let $\mathbf{h}_\mathrm{t}(\mathbf{a})$ be the 3-D coordinates of a point at time t, given parameter $\mathbf{a}$. $\mathbf{h}_\mathrm{t}(\mathbf{a})$ is given by:

$$\mathbf{h}_\mathrm{t}(\mathbf{a}) = \mathbf{R}(\mathbf{q}(\mathrm{t}))\mathbf{X}_0 + \mathrm{t}\mathbf{T}. \tag{5}$$

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

$$\mathbf{X}_\mathrm{t} = \mathbf{p}\ \mathbf{h}_\mathrm{t}(\mathbf{a}) \begin{pmatrix} x_t/f \\ y_t/f \\ 1 \end{pmatrix}.$$

The objective function to be minimized is given by:

$$E^2(\mathbf{a}) = \sum_{\mathrm{t}=\mathrm{s}}^{\mathrm{s}+\mathrm{N}} \parallel \mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a}) \parallel^2, \tag{6}$$

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

4.2 Minimization

The Levenberg-Marquadt (L-M for short) algorithm is used to minimize the nonlinear function (6) . Let $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6)^\top$. The first and second partial derivative of the objective function with respect to parameter $\mathbf{a}_\mathrm{i}$, $i = 1, 2, \cdots, 6$, must be computed.

Let $\mathbf{R}_\mathrm{ij}$ denote the element in the ith row and the jth column of $\mathbf{R}$. First, the partial derivatives of the function $\mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a})$ in Equation (6) with respect to the parameter $\mathbf{a}$ are computed as follows:

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

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

$$\frac{\partial}{\partial \mathbf{a}_6}(\mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a})) = (\mathbf{R}_{31}\frac{\mathrm{x}_0}{\mathrm{f}} + \mathbf{R}_{32}\frac{\mathrm{y}_0}{\mathrm{f}} + \mathbf{R}_{33}) \begin{pmatrix} x_t/f \\ y_t/f \\ 1 \end{pmatrix} - \mathbf{R} \begin{pmatrix} x_0/f \\ y_0/f \\ 1 \end{pmatrix}, \tag{9}$$

where

$$\frac{\partial \mathbf{T}}{\partial \mathbf{a}_1} = \frac{\partial \mathbf{T}}{\partial \theta_T} = \begin{pmatrix} \cos\theta_T\ \cos\phi_T \\ \cos\theta_T\ \sin\phi_T \\ -\sin\theta_T \end{pmatrix}, \qquad \frac{\partial \mathbf{T}}{\partial \mathbf{a}_2} = \frac{\partial \mathbf{T}}{\partial \phi_T} = \begin{pmatrix} -\sin\theta_T\ \sin\phi_T \\ \sin\theta_T\ \cos\phi_T \\ 0 \end{pmatrix},$$

$$\frac{\partial \mathbf{R}}{\partial \mathbf{a}_3} = \frac{\partial \mathbf{R}}{\partial w_x} = \sum_{k=0}^{3} D_k \frac{\partial q_k}{\partial w_x}, \quad \frac{\partial \mathbf{R}}{\partial \mathbf{a}_4} = \frac{\partial \mathbf{R}}{\partial w_y} = \sum_{k=0}^{3} D_k \frac{\partial q_k}{\partial w_y}, \quad \frac{\partial \mathbf{R}}{\partial \mathbf{a}_5} = \frac{\partial \mathbf{R}}{\partial w_z} = \sum_{k=0}^{3} D_k \frac{\partial q_k}{\partial w_z},$$

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

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

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

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

$\frac{\partial q_0}{\partial w_x} = (\frac{1}{w} - \frac{w_x^2}{w^3}) \sin\frac{wt}{2} + \frac{t}{2}\frac{w_x^2}{w^2}\cos\frac{wt}{2}$, $\frac{\partial q_1}{\partial w_x} = -\frac{w_x w_y}{w^3}\sin\frac{wt}{2} + \frac{t}{2}\frac{w_x w_y}{w^2}\cos\frac{wt}{2}$,

$\frac{\partial q_2}{\partial w_x} = -\frac{w_x w_z}{w^3}\sin\frac{wt}{2} + \frac{t}{2}\frac{w_x w_z}{w^2}\cos\frac{wt}{2}$, $\frac{\partial q_3}{\partial w_x} = -\frac{t}{2}\frac{w_x}{w}\sin\frac{wt}{2}$,

$\frac{\partial q_0}{\partial w_y} = -\frac{w_x w_y}{w^3}\sin\frac{wt}{2} + \frac{t}{2}\frac{w_x w_y}{w^2}\cos\frac{wt}{2}$, $\frac{\partial q_1}{\partial w_y} = (\frac{1}{w} - \frac{w_y^2}{w^3}) \sin\frac{wt}{2} + \frac{t}{2}\frac{w_y^2}{w^2}\cos\frac{wt}{2}$,

$\frac{\partial q_2}{\partial w_y} = -\frac{w_y w_z}{w^3}\sin\frac{wt}{2} + \frac{t}{2}\frac{w_y w_z}{w^2}\cos\frac{wt}{2}$, $\frac{\partial q_3}{\partial w_y} = -\frac{t}{2}\frac{w_y}{w}\sin\frac{wt}{2}$,

$\frac{\partial q_0}{\partial w_z} = -\frac{w_x w_z}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_x w_z}{w^2} \cos \frac{wt}{2}$, $\frac{\partial q_1}{\partial w_z} = -\frac{w_y w_z}{w^3} \sin \frac{wt}{2} + \frac{t}{2} \frac{w_y w_z}{w^2} \cos \frac{wt}{2}$,
$\frac{\partial q_2}{\partial w_z} = (\frac{1}{w} - \frac{w_z^2}{w^3}) \sin \frac{wt}{2} + \frac{t}{2} \frac{w_z^2}{w^2} \cos \frac{wt}{2}$, $\frac{\partial q_3}{\partial w_z} = -\frac{t}{2} \frac{w_z}{w} \sin \frac{wt}{2}$.

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

$$\frac{\partial E^2}{\partial \mathbf{a}_\mathrm{i}} = 2 \sum_{t=s}^{s+N} (\mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a}))^\top \frac{\partial}{\partial \mathbf{a}_\mathrm{i}} (\mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a})), \quad \mathrm{i} = 1, 2, \cdots, 6. \tag{10}$$

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

$$\frac{\partial^2 E^2}{\partial \mathbf{a}_\mathrm{i} \partial \mathbf{a}_\mathrm{j}} \approx 2 \sum_{t=s}^{s+N} \frac{\partial}{\partial \mathbf{a}_\mathrm{i}} (\mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a}))^\top \frac{\partial}{\partial \mathbf{a}_\mathrm{j}} (\mathbf{X}_\mathrm{t} - \mathbf{h}_\mathrm{t}(\mathbf{a})), \quad \mathrm{i}, \mathrm{j} = 1, 2, \cdots, 6. \tag{11}$$

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

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

where m is the number of unknown parameters. Here $m = 6$. The algorithm for single trajectory fitting is summarized as follows:

Algorithm: TrajFit

1. Compute $E^2(\mathbf{a})$ in Equation (6). Set $\lambda = 0.001$.
2. Compute β_i and α_{ij}, where $i, j = 1, 2, \cdots, 6$, 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 Equation (12) for $\delta(\mathbf{a})$ and evaluate $E^2(\mathbf{a} + \delta\mathbf{a})$.
5. If $E^2(\mathbf{a} + \delta\mathbf{a}) \geq \mathrm{E}^2(\mathbf{a})$, 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.

5 Two-Frame SFM Algorithm for Multiple Motions

Since nonlinear optimization is involved in single trajectory fitting, an initial guess is obtained first by applying a closed-form SFM algorithm to avoid being trapped into a local minimum and to speed up the convergence. There are a number of algorithms which can provide a closed-form solution for motion and structure estimation. In this work, Weng et al.'s linear algorithm [24] based on epipolar constraint is chosen because of its simplicity. Their method is extended to deal with multiple motions. An image is subdivided 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 [24] 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.

$$\mathbf{A} = \begin{pmatrix} u_1 u_1' & u_1 v_1' & u_1 f & v_1 u_1' & v_1 v_1' & v_1 f & u_1' f & v_1' f & f^2 \\ u_2 u_2' & u_2 v_2' & u_2 f & v_2 u_2' & v_2 v_2' & v_2 f & u_2' f & v_2' f & f^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n u_n' & u_n v_n' & u_n f & v_n u_n' & v_n v_n' & v_n f & u_n' f & v_n' f & f^2 \end{pmatrix}, \qquad (13)$$

   where $\mathbf{x}_\mathrm{i} = (\mathrm{u_i}, \mathrm{v_i}, \mathrm{f})^\top$, $\mathbf{x}_\mathrm{i}' = (\mathrm{u_i'}, \mathrm{v_i'}, \mathrm{f})^\top, \mathrm{i} = 1, 2, \cdots, \mathrm{n}$, 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, $\mathbf{m}_\mathrm{i}$, 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 $\{\mathbf{m}_\mathrm{i}\}$.

(c) Convert the feature vectors at centroids of K clusters in the original scale, and compute motion parameters for K moving objects.

3. Determine 3-D structure for K moving objects.

For each feature point i in the image,

(a) For each set of motion parameters $(\mathbf{T}_\mathrm{j}, \mathbf{R}_\mathrm{j})$, $j = 0, \cdots, K$,

- Compute the structure $(\mathbf{X}'_\mathrm{i}, \mathbf{X}_\mathrm{i})$ at t_{k+1} and t_k, such that $\| \mathbf{X}'_\mathrm{i} - \mathbf{R}_\mathrm{j}\mathbf{X}_\mathrm{i} - \mathbf{T}_\mathrm{j} \| \rightarrow$ min.
- Compute the corresponding residual error $Res_j = \| \mathbf{X}'_\mathrm{i} - \mathbf{R}_\mathrm{j}\mathbf{X}_\mathrm{i} - \mathbf{T}_\mathrm{j} \|$.

(b) Assign feature point i to Segment s, where $Res_s = min\{Res_j, j = 1, \cdots, K\}$, and $(\mathbf{X}'_\mathrm{i}, \mathbf{X}_\mathrm{i})$ is given by the corresponding $(\mathbf{T}_\mathrm{s}, \mathbf{R}_\mathrm{s})$.

(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:

$$\mathbf{X}(\mathrm{t}_0) = \mathbf{R}^\top(\mathrm{t}_\mathrm{k})(\mathbf{X}(\mathrm{t}_\mathrm{k}) - \mathrm{t}_\mathrm{k}\mathbf{T}),$$

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

In the k-means algorithm, all the feature dimensions should be independent. A six dimensional feature space $\{(speed, \theta_T, \phi_T, w_x, w_y, w_z)\}$ is chosen, 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.

6 Trajectory Grouping

After performing trajectory fitting for each trajectory,the trajectories will be segmented into groups corresponding to moving objects based on the computed motion parameters. Moreover, when the objects move, some features may disappear due to self-occlusion or mutual occlusion among multiple objects, while new features may appear. The 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, the "broken" trajectory segments must be merged, 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, $\mathbf{m}_{\mathrm{i}}$, 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 $Res_j = \| \mathbf{X}_{\mathrm{i0}} - \mathbf{X}_{\mathrm{j0}} \|$, where $\mathbf{X}_{\mathrm{i0}}$ and $\mathbf{X}_{\mathrm{j0}}$ denote the structure of feature i and j at t_0, respectively.
   - Record the minimal residual error and the corresponding Trajectory k, such that $Res_k = min\{Res_j, j = 1, \cdots, K\}$.
   - If Res_k is large, record Trajectory i belongs to a new feature, otherwise, link Trajectory i to Trajectory k.

7 Multiple Trajectory Fitting

In order to reject noise and obtain more accurate estimates of motion and structure of the objects, the multiple trajectory fitting algorithm is applied to each group of trajectories. The objective function to be minimized is given by:

$$E^2 = \sum_{t=0}^{N} \sum_{i=1}^{M} \| \mathbf{X}_{\mathrm{i}}(\mathrm{t}) - \mathbf{h}_{\mathrm{t}}(\mathbf{m}, \mathbf{X}_{\mathrm{i0}}) \|^2, \tag{14}$$

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

$$\mathbf{h}_{\mathrm{t}}(\mathbf{m}, \mathbf{X}_{\mathrm{i0}}) = \mathbf{R}(\mathbf{q}(\mathrm{t}))\mathbf{X}_{\mathrm{i0}} + \mathrm{t}\mathbf{T}, \tag{15}$$

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

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

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

For this problem, there are $M + 5$ unknown parameters; five are motion parameters, and one depth parameter for each of M feature points. Let $\mathbf{a}$ denote these unknowns: $\mathbf{a} = (\mathbf{a}_1, \cdots, \mathbf{a}_{\mathrm{M+5}})^\top = (\theta_{\mathrm{T}}, \phi_{\mathrm{T}}, w_{\mathrm{x}}, w_{\mathrm{y}}, w_{\mathrm{z}}, \mathrm{Z}_{10}, \cdots, \mathrm{Z}_{\mathrm{M0}},)^\top$. L-M method is used for this minimization also.

8 Results

This section presents results on both synthetic data and real images. The simulation results are presented first.

8.1 Synthetic Images

Two objects, namely the Upper Object, O_U, and the Lower Object, O_L, are assumed to be 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, the objects undergo rotation $w_{\mathrm{U}} = (1.5^0,\ -8^0,\ 1^0)^\top$, and $w_{\mathrm{L}} = (0^0,\ 0^0,\ 3^0)^\top$, followed by translation $\mathbf{T}_{\mathrm{U}} = (0.354, 0.612, 0.707)^\top$, $\mathbf{T}_{\mathrm{L}} = (0.296, 0.171, 0.940)^\top$, respectively. Twenty-five feature points from each object are tracked over five frames. The trajectories are shown in Figure 2. 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

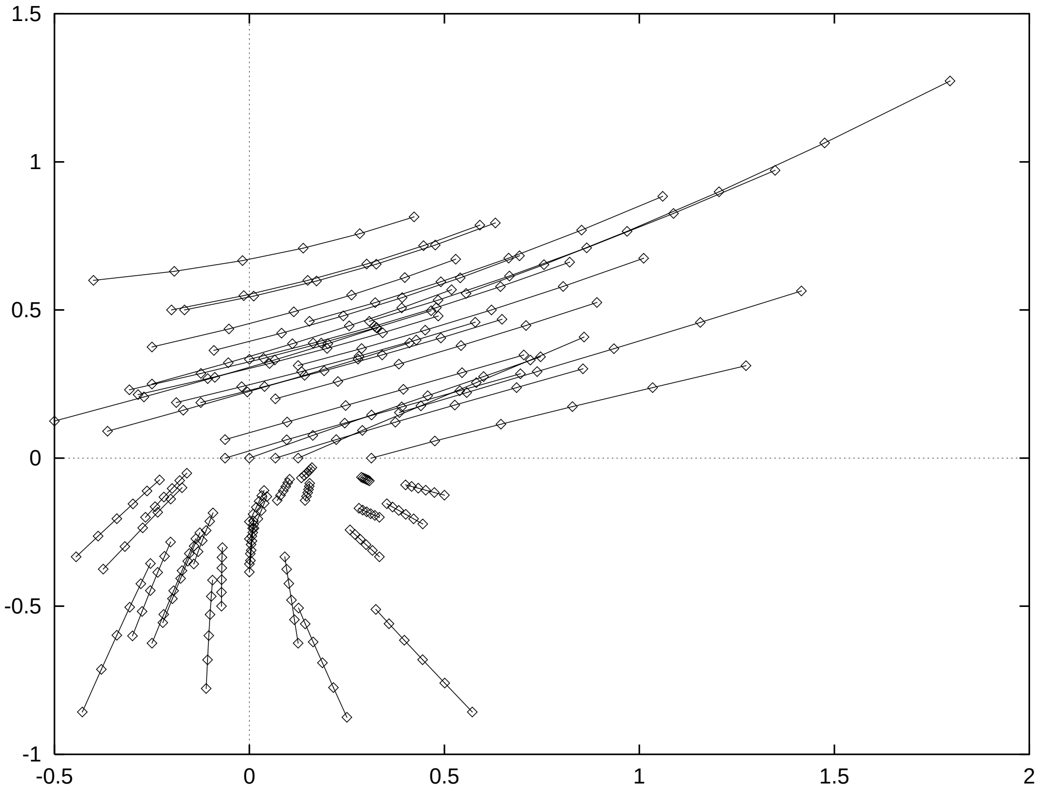
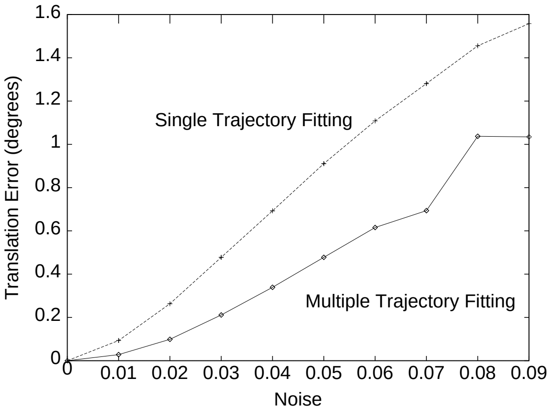


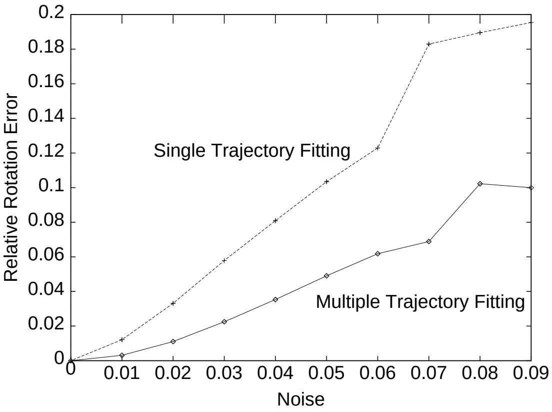
Figure 2: Generated trajectories of two moving objects.

evaluated against the noise also. In this experiment, only a single moving object is studied, 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. Noise with uniform distribution is added 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.

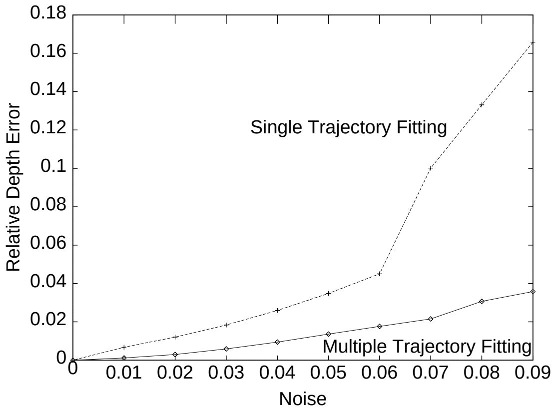
The errors associated with translation, rotation, and depth were evaluated. Translation error is defined as the angle between the actual and estimated translational direction, and the rotation (or depth) error is defined as the relative error, i.e. the norm of the error vector (or scalar value) divided by the norm of the actual vector (or scalar value). Both single trajectory fitting and multiple trajectory fitting are evaluated. 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 3 illustrates the estimate



(a)



(b)



(c)

Figure 3: Errors in motion and depth estimates against different noise levels.

errors as a function of the noise level. It is from the experiments that providing initial guess by two-frame SFM algorithm indeed improved the results. The reason is that the single trajectory fitting is easily trapped into a local minimum instead of the global minimum for the noisy data. Also, the good initial estimate will help converge faster to the correct estimate, especially for the nonlinear minimization. Figure 3 also shows that the multiple trajectory fitting improves the estimates further.

8.2 Real Images

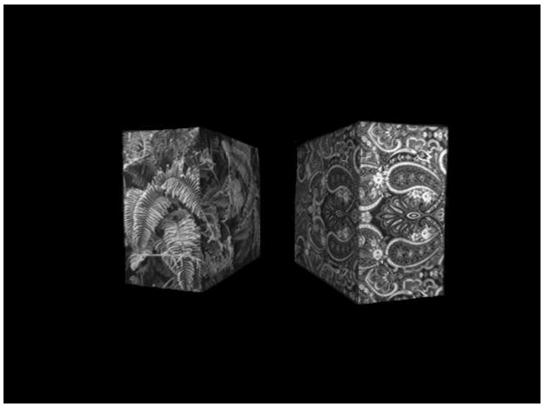
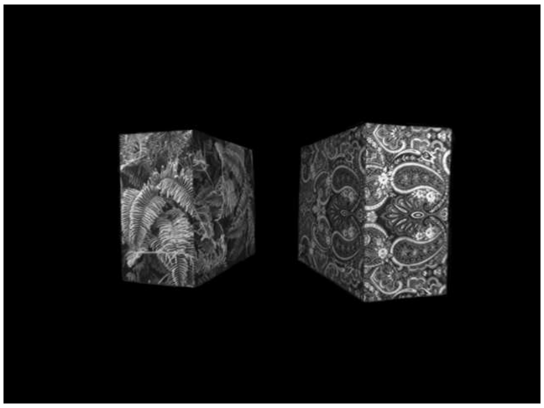
A standard Pulnix CCD camera with a 2/3” sensor (corresponding to an actual sensor size of 6.6 x 9.8 mm) and a high quality Schneider Cinegon 1.8/4.8mm lens are used. Because of the wide field of view (105 degrees along diagonal), distortion is unavoidable and calibration is crucial. In our experiments, radial distortion is removed by adapting the procedure described in [9] and fitting a fifth-degree polynomial to the coordinates measured on an image of a calibration target made of equidistant stripes. Frames are acquired by a Galileo Video frame grabber for an Indigo 2. The algorithm is tested with two real image sequences: the Cubes Sequence and the Cylinders Sequence, where motion is controlled by a DMC-1500 robot. Since the algorithm uses a camera-centered coordinate system, and no attempt has been made to align the robot coordinate system with the camera coordinate system, true 3-D motion parameters are unknown. For feature tracking, the Tomasi-Kanade method [19] is used, which allows for the automatic selection of image features.

The Cubes Sequence: Some frames in this sequences are shown in Figure 4, where both objects move along all axes. The left cube moves to the left, right cube moves to the right, and both move away from camera. The speed of the left cube is twice of the right cube. As mentioned earlier, the speed is the scale factor that cannot be determined from monocular sequence. The ratio of the speed of these two motions can be used to determine the relative size of objects (shown in Figure 6(c)). Figure 5 shows the features automatically selected in the first frame and trajectories throughout 60 frames. The features are grouped for individual structure recovery. Figures 6(a) and (b) show the segmentation result, top view of recovered structure of two cubes. The recovered 3-D points are displayed in Figure 6(c) with linearly interpolated surface to convey a better perception of their structure.

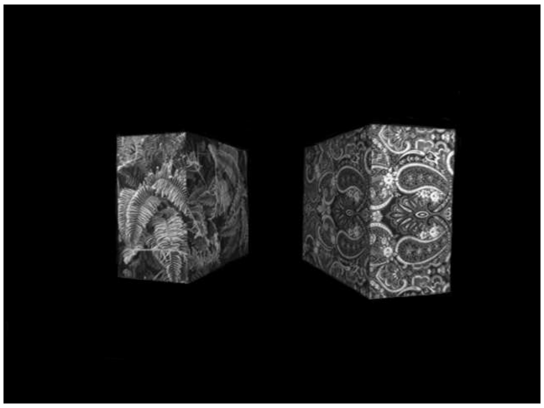
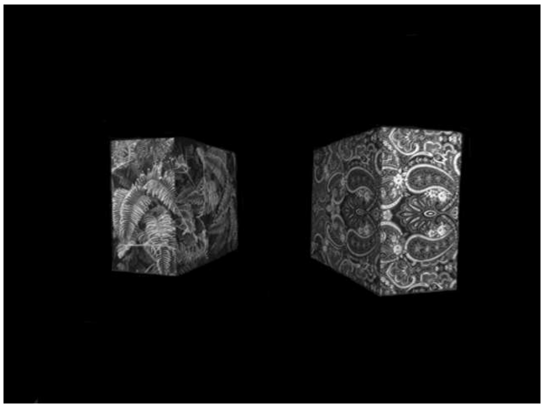
The Cylinders Sequence: Figure 7 shows some of 60 frames in this sequences. The scene consists of two cylindrical objects moving along all axes at the approximately same speed, and away from the camera. The left object moves down and to the left while the right object moves to the right. Figure 8 shows the features automatically selected in the first frame and trajectories throughout 60 frames. Figure 9(a) and (b) show the segmentation result, top view of recovered structure of two cylinders. The recovered 3-D points are displayed in Figure 9(c) with linearly interpolated surface.

9 Summary

In this paper, a multi-frame algorithm is proposed to determine 3D motion and structure of multiple objects. In this method, segmentation is performed based on 3D motion parameters to reduce the ambiguity inherently present in 2D segmentation of the 3D scene. The initial

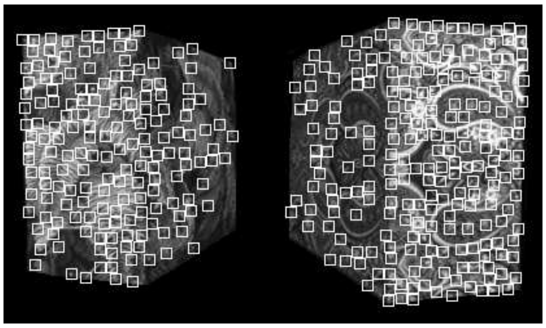
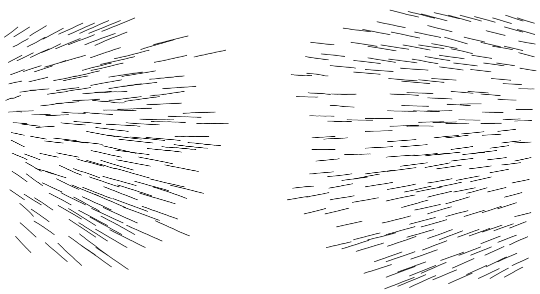



1 15

30 60

Figure 4: The Cubes Sequence: four of the 60 frames.

(a) (b)

Figure 5: The Cubes Sequence: (a) the features automatically selected in the first frame. (b) trajectories of 381 features.

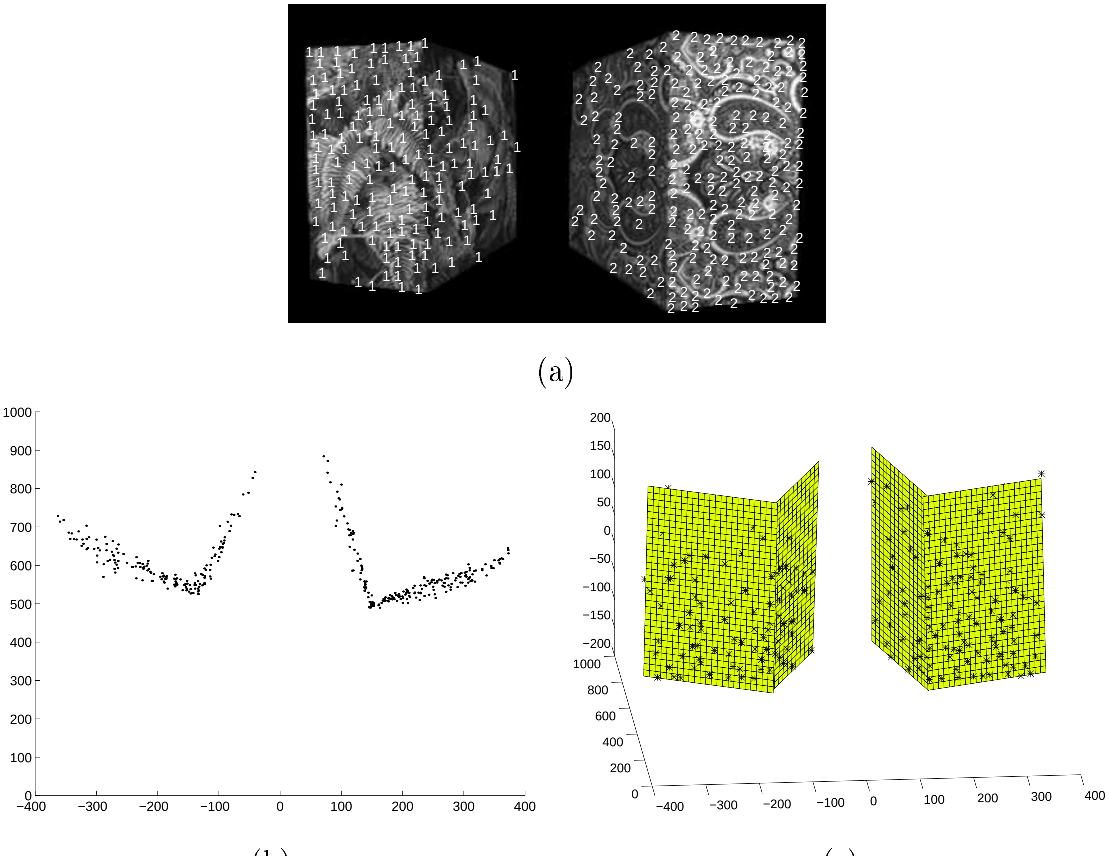
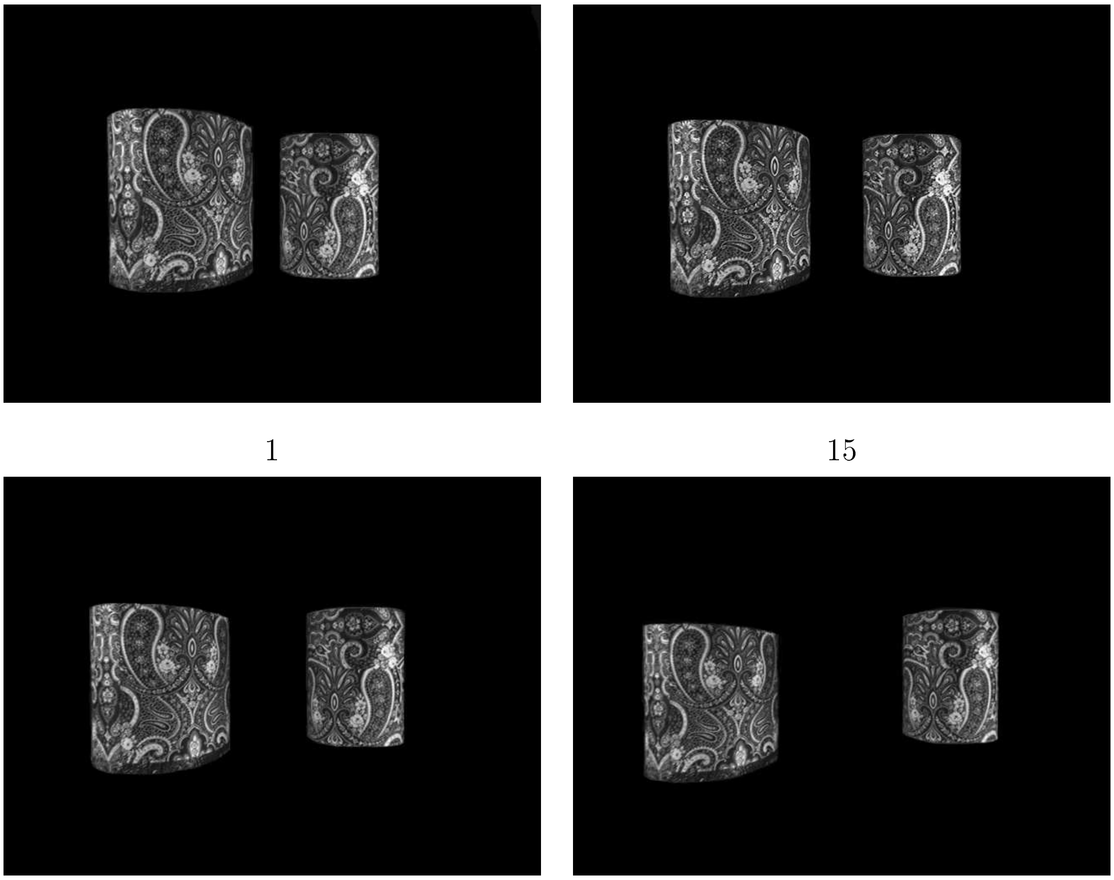


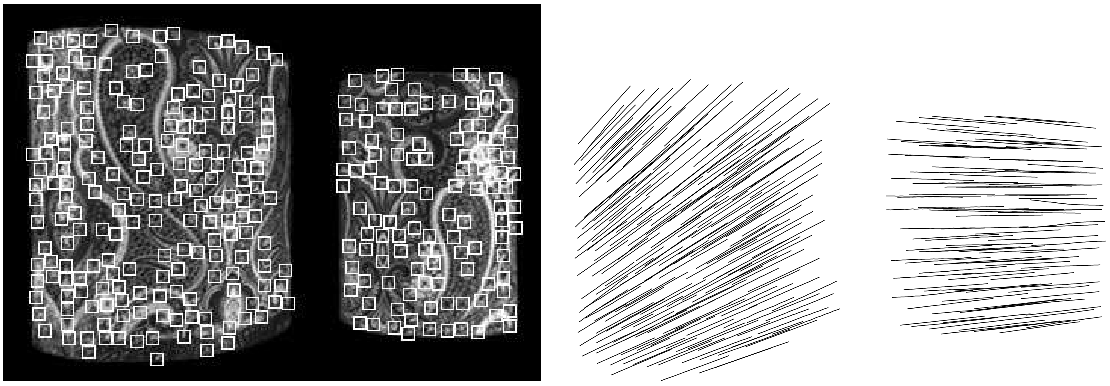
Figure 6: Results of the Cubes Sequence. (a) segmentation result. (b) top view and (c) recovered structure of two cubes.



1 15

30 60

Figure 7: The Cylinders Sequence: four of the 60 frames.



(a) (b)

Figure 8: The Cylinders Sequence: (a) the features automatically selected in the first frame. (b) trajectories of 300 features.

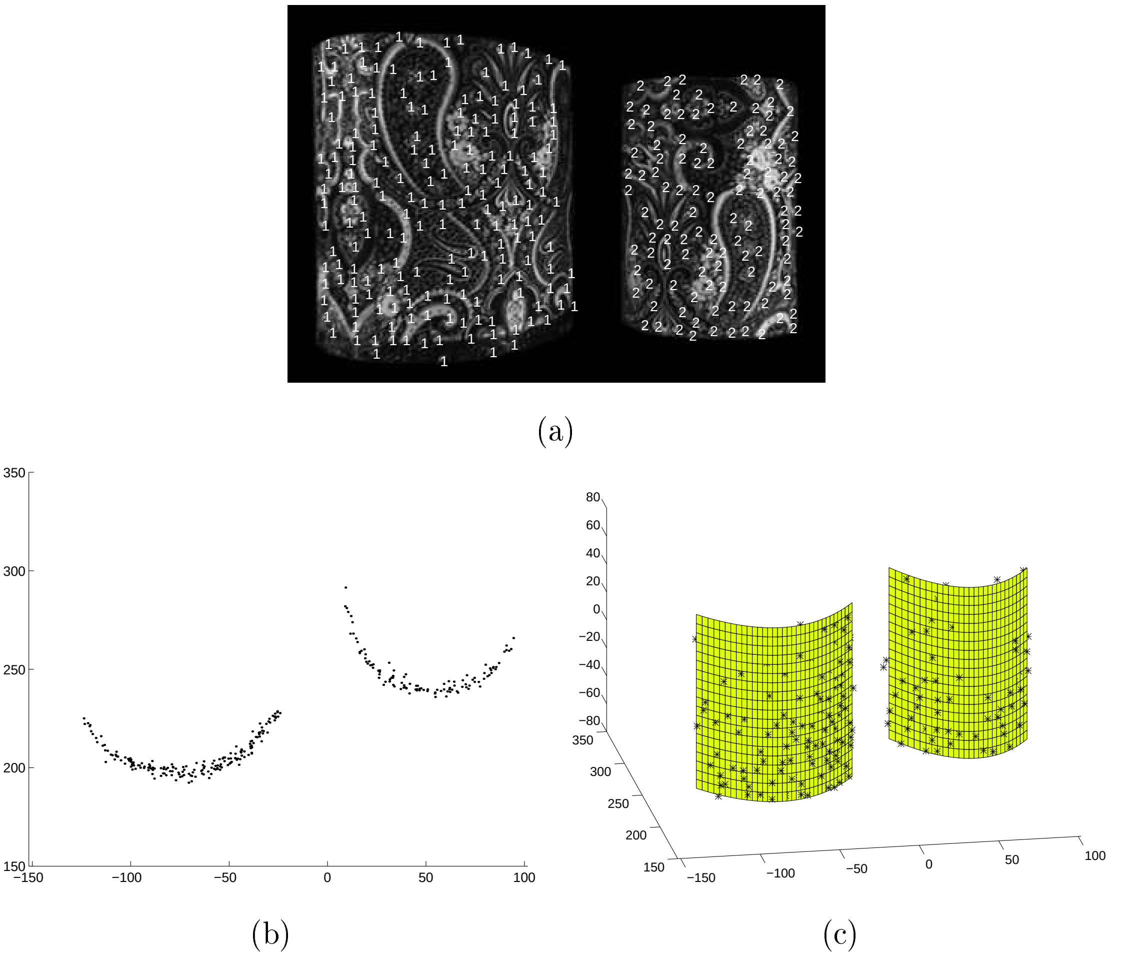


Figure 9: Results of the Cylinders Sequence. (a) segmentation result. (b) top view and (c) recovered structure of two cylinders.

estimate is provided by a linear two-frame multiple motion SFM algorithm based on the first two frames, and is refined over time. The general motion model is fitted to a single trajectory to allow an arbitrary motion. The multiple trajectory fitting is applied to each group of trajectories corresponding to each moving object in order to improve the estimates further. The method is demonstrated on synthetic and real images.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:384–401, 1985.

[2] S. Ayer, P. Schroeter, and J. Bigun. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Proceedings of European Conference on Computer Vision*, pages 316–327, 1994.

[3] D. Ballard and O. Kimball. Rigid body motion from depth and optical flow. *Computer Vision, Graphics, and Image Processing*, 22:95–115, 1983.

[4] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of International Conference on Computer Vision*, pages 374–381, 1995.

[5] T. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26:639–656, 1990.

[6] T. Broida and R. Chellappa. Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:497–513, 1991.

[7] M. Bober and J. Kittler. Estimation of complex multimodal motion: An approach based on robust statistics and hough transform. *Image and Vision Computing*, 12:661–668, 1994.

[8] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proceedings of International Conference on Computer Vision*, pages 1071–1076, 1995.

[9] M. Fleck. Shape and wide-angle image. *Tech. Rep. 94-04*, University of Iowa, Computer Science, 1994.

[10] C. Gear. Feature grouping in moving objects. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 214–219, 1994.

[11] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proceedings of European Conference on Computer Vision*, pages 282–287, 1992.

[12] A. Jepson and D. Heeger. Linear subspace methods for recovering translation direction. In L Harris and M Jenkin, editors, *Spatial Vision in Humans and Robots*, pages 39–62. Cambridge University Press, New York, 1993.

[13] W. MacLean, A. Jepson, and R. Frecker. Recovery of egomotion and segmentation of independent object motion using the em algorithm. In *Proceedings of the British Machine Vision Conference*, pages 13–16, 1994.

[14] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:602–605, 1993.

[15] S. Soatto, P. Perona, R. Frezza, and G. Picci. Recursive motion and structure estimation with complete error characterization. In *Proceedings of Computer Vision and Pattern Recognition*, pages 428–433, 1993.

[16] H. Sawhney, J. Oliensis, and A. Hanson. Image description and 3-d reconstruction from image trajectories of rotational motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:885–898, 1993.

[17] W. Thompson, K. Mutch, and V. Berzins. Dynamic occlusion analysis in optical flow fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:374–383, 1985.

[18] W. Thompson, P. Lechleider, and E. Stuck. Detecting moving objects using the rigidity constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:162–166, 1993.

[19] C. Tomasi and T. Kanade. Detection and tracking of point features, shape and motion from image streams: a factorization method – part 3. In *Tech. Rep. CMU-CS-91-132*, 1991.

[20] C. Tomasi and T. Kanade. Shape from motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[21] P. Torr and D. Murray. Stochastic motion clustering. In *Proceedings of European Conference on Computer Vision*, pages 328–337, 1994.

[22] J. Wang and E. Adelson. Layer representation for motion analysis. In *Proceedings of Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[23] J. Weng, T. Huang, and N. Ahuja. 3d motion estimation understanding and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:370–389, 1987.

[24] J. Weng, T. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:451–476, 1989.

[25] J. Weng, N. Ahuja, and T. Huang. Optimal motion and structure estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:864–884, 1993.

[26] Y. Xiong and S. Shafer. Dense Structure from a Dense Optical Flow Sequences. In *International Symposium on Computer Vision*, pages 21–23, 1995.