Multiple Motion Scene Reconstruction from Uncalibrated Views

Mei Han [1]
C & C Research Laboratories
NEC USA, Inc.
meihan@ccrl.sj.nec.com

Takeo Kanade
Robotics Institute
Carnegie Mellon University
tk@cs.cmu.edu

Abstract

We describe a reconstruction method of multiple motion scenes, which are the scenes containing multiple moving objects, from uncalibrated views. Assuming that the objects are moving with constant velocities, the method recovers the scene structure, the trajectories of the moving objects, the camera motion and the camera intrinsic parameters (except skews) simultaneously. The number of the moving objects is automatically detected without prior motion segmentation. The method is based on a unified geometrical representation of the static scene and the moving objects. It first performs a projective reconstruction using a bilinear factorization algorithm, and then converts the projective solution to a Euclidean one by enforcing metric constraints. Experimental results on synthetic and real images are presented.

1. Introduction

Structure from Motion (SFM), which is recovering camera motion and scene structure from image sequences, has various applications, such as scene modeling, robot navigation, object recognition and virtual reality. Most of previous research on SFM requires simplifying assumptions on the camera or the scene. Common assumptions are a) the camera intrinsic parameters, such as focal lengths, are known or unchanged throughout the sequence, and/or b) the scene does not contain moving objects. In practice, these are unrealistic assumptions. In this paper we propose a factorization-based method for multiple motion scene reconstruction from uncalibrated views. The method reconstructs the scene structure, the trajectories of the moving objects, the camera motion and the camera intrinsic parameters (assuming zero skews) simultaneously. The number of the moving objects is automatically detected without prior motion segmentation.

When the moving objects are far from the camera, it is difficult to get multiple feature points from every moving object throughout the image sequences. It is a good approximation to abstract the moving objects as points. However, recovering the locations of a moving point from a monocular image sequence is impossible without assumptions about its trajectory [2]. In this paper we assume that the objects are moving linearly with constant speeds. This assumption is reasonable for most moving objects, such as cars, planes and people, especially for short time intervals.

1.1. Related work

The multibody factorization method proposed by Costeira and Kanade [7] reconstructs the motions and shapes of independently moving objects, but requires that each object have multiple feature points. Avidan and Shashua [2] recover the trajectory of a 3D point by line fitting. They assume that the object is moving along a line, but they do not require the object to move with constant speed. They assume that the camera positions and the prior motion segmentation are given. They extend this work to conic shape trajectories in [18]. Bregler et al. [4] describe a technique to recover non-rigid 3D model based on the representation of 3D shape as a linear combination of a set of basis shapes. The complexity of their solution increases with the number of basis shapes. There is much work done on planar points as well [23, 19, 20]. Han and Kanade present a multiple motion scene reconstruction method under affine camera models in [11], assuming that the objects are moving with constant velocities. The method requires that the camera intrinsic parameters are known.

Whether cameras are intrinsically pre-calibrated or uncalibrated differentiates various solutions. The above multiple motion scene reconstruction methods all assume that the cameras are intrinsically pre-calibrated. When nothing is known about the camera calibration parameters or the scene, it is only possible to compute a reconstruction up to an unknown projective transformation [8]. There has been considerable progress on projective reconstruction ([3, 22, 13, 17]). Some additional information about either the camera or the object is needed to obtain a Euclidean

[1] The research described in this paper was conducted while the first author was a Ph.D. student in the Robotics Institute at CMU.

0-7695-1143-0/01 $10.00 © 2001 IEEE

reconstruction from the projective reconstruction. Tremendous work has been done in this area ([12, 14, 1, 16, 10]). However, most of these methods assume that the scenes do not contain moving objects.

The method presented here uses the factorization technique as the basis of solution. The factorization method, first developed by Tomasi and Kanade [21] for orthographic views and extended by Poelman and Kanade [15] to weak and para perspective views, achieves its robustness and accuracy by applying the singular value decomposition (SVD) to a large number of images and feature points. Christy and Horaud [5, 6] describe a method for the perspective camera model by incrementally performing reconstructions with either a weak or a para perspective camera model. One major limitation with most previous factorization methods is that they require the use of intrinsically calibrated cameras. The method described in this paper, however, works on uncalibrated cameras.

1.2. Representation

We propose a unified representation of the static scene and the moving objects in [11]. Assuming that m feature points are tracked over n images, some of them static and the others moving linearly with constant speeds, we regard every point as a moving point with constant velocity: the static points simply have zero velocity. Any point is represented by a 3×1 vector $\mathbf{p}_j$,

$$\mathbf{p}_j = \mathbf{s}_j + i\mathbf{v}_j \tag{1}$$

in a world coordinate system, where $i = 1 \cdots n$ and $j = 1 \cdots m$, n is the number of frames and m is the number of feature points. $\mathbf{s}_j$ is the point position at frame 0 (i.e., when the 0th frame is taken) and $\mathbf{v}_j$ is its motion velocity. The method presented in this paper is built on the same unified representation of feature points.

2. Projective reconstruction

Given tracked feature points from uncalibrated views, we first perform a projective reconstruction. Perspective projection P_i, $i = 1 \cdots n$, is represented by a 3×4 matrix,

$$P_i \sim K_i \begin{bmatrix} R_i & \mathbf{t}_i \end{bmatrix} \tag{2}$$

where

$$K_i = \begin{bmatrix} f_i & 0 & u_{0i} \\ 0 & \alpha_i f_i & v_{0i} \\ 0 & 0 & 1 \end{bmatrix} \quad R_i = \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \\ \mathbf{k}_i^{\mathrm{T}} \end{bmatrix} \quad \mathbf{t}_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix}$$

The upper triangular calibration matrix K_i encodes the intrinsic parameters of the ith camera: f_i represents the focal length, (u_{0i}, v_{0i}) is the principal point and α_i is the aspect ratio. We assume that the cameras have zero skews. R_i is the ith rotation matrix with $\mathbf{i}_i$, $\mathbf{j}_i$ and $\mathbf{k}_i$ denoting the rotation axes. $\mathbf{t}_i$ is the ith translation vector. Feature point $\mathbf{x}_j$, $j = 1 \cdots m$, is represented by homogeneous coordinates,

$$\mathbf{x}_j \sim \begin{bmatrix} \mathbf{p}_j^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \tag{3}$$

where $\mathbf{p}_j$ is defined as the unified representation of point in Equation (1). The image coordinates are represented by (u_{ij}, v_{ij}) and the following hold,

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} \sim P_i \mathbf{x}_j \text{ or } \lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = \mathrm{P_i x_j} \tag{4}$$

where λ_{ij} is a non-zero scale factor called projective depth. According to Equations (2) and (3),

$$\begin{aligned} P_i \mathbf{x}_j &\sim K_i \left(R_i \, \mathbf{p}_j + \mathbf{t}_i \right) \\ &= K_i \left(R_i \, \mathbf{s}_j + i \, R_i \, \mathbf{v}_j + \mathbf{t}_i \right) \\ &= K_i \begin{bmatrix} R_i & iR_i & \mathbf{t}_i \end{bmatrix} \begin{bmatrix} \mathbf{s}_j^{\mathrm{T}} & \mathbf{v}_j^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \\ &\sim \tilde{P}_i \tilde{\mathbf{x}}_j \end{aligned} \tag{5}$$

where

$$\tilde{P}_i \sim K_i \begin{bmatrix} R_i & iR_i & \mathbf{t}_i \end{bmatrix} \quad \tilde{\mathbf{x}}_j \sim \begin{bmatrix} \mathbf{s}_j^{\mathrm{T}} & \mathbf{v}_j^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \tag{6}$$

$\tilde{P}_i$ is a 3×7 matrix which is the product of the ith calibration matrix and the unified motion matrix composed of the camera rotation, the scaled camera rotation by the frame number and the camera translation. $\tilde{\mathbf{x}}_j$ is a 7×1 vector which is the homogeneous representation of the unified scene structure including the initial point position and its velocity. The equivalent matrix form is,

$$W_s = \begin{bmatrix} \lambda_{11} \begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1m} \begin{bmatrix} u_{1m} \\ v_{1m} \\ 1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \lambda_{n1} \begin{bmatrix} u_{n1} \\ v_{n1} \\ 1 \end{bmatrix} & \cdots & \lambda_{nm} \begin{bmatrix} u_{nm} \\ v_{nm} \\ 1 \end{bmatrix} \end{bmatrix} \tag{7}$$

$$= \begin{bmatrix} \tilde{P}_1 \\ \vdots \\ \tilde{P}_n \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_1 & \cdots & \tilde{\mathbf{x}}_m \end{bmatrix} = \tilde{P}\tilde{X} \tag{8}$$

where W_s is the *scaled measurement matrix*. We call the $3n \times 7$ matrix $\tilde{P}$ as motion matrix and the $7 \times m$ matrix $\tilde{X}$ as shape matrix. The constraint of the objects moving with constant velocities enables the unified representation of the motion matrix $\tilde{P}$ and the shape matrix $\tilde{X}$. They are both at most rank 7, therefore, the rank of the scaled measurement

matrix W_s is at most 7 (instead of rank 4 when the scene does not contain moving objects).

We apply the following bilinear factorization algorithm to get the projective reconstruction. The algorithm is similar to the iterative algorithm presented in [22] with the difference that a rank 7 matrix factorization is performed at step 3. It iteratively applies factorization to the current scaled measurement matrix.

Iterative Projective Factorization Algorithm

1. Set $\lambda_{ij} = 1$, for $i = 1 \cdots n$ and $j = 1 \cdots m$;
2. Compute the current scaled measurement matrix W_s by Equation (7);
3. Perform **rank** 7 factorization on W_s, generate the projective motion $\hat{P}$ and shape $\hat{X}$;
4. Reset $\lambda_{ij} = \hat{P}_i^{(3)}\hat{\mathbf{x}}_j$, where $\hat{P}_i^{(3)}$ denotes the third row of the projection matrix $\hat{P}_i$;
5. If λ_{ij}'s are the same as the previous iteration, stop; else go to step 2.

The goal of the projective reconstruction process is to estimate the values of the projective depths (λ_{ij}'s) which make Equation (8) consistent. The reconstruction results are iteratively improved by back projecting the current projective reconstruction to refine the depth estimates.

3. Euclidean reconstruction

The factorization of Equation (8) recovers the motion and shape up to a 7×7 linear projective transformation H,

$$W_s = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = \tilde{P}\tilde{X} \tag{9}$$

where $\tilde{P} = \hat{P}H$ and $\tilde{X} = H^{-1}\hat{X}$. $\hat{P}$ and $\hat{X}$ are referred to as the projective motion and the projective shape. Any non-singular 7×7 matrix could be inserted between $\hat{P}$ and $\hat{X}$ to get another motion and shape pair. The goal of the Euclidean reconstruction is to impose metric constraints on the projective motion and shape in order to recover the linear transformation H, from which we can simultaneously reconstruct the intrinsic parameters and the Euclidean motion and shape. This is the *normalization* process.

In this section we present a linear normalization algorithm for the case that only the focal lengths are unknown and varying. It is straightforward to derive the normalization algorithms for the other cases where more or all of the intrinsic parameters are unknown (except skews) following the same line of work presented in this paper.

When the focal lengths are the only unknown intrinsic parameters, we have,

$$u_{0i} = 0 \quad v_{0i} = 0 \quad \alpha_i = 1 \tag{10}$$

therefore, according to Equation (6),

$$\tilde{P} = \begin{bmatrix} M & T \end{bmatrix} \tag{11}$$

where

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \\ \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{z1} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} & \mathbf{n}_{zn} \end{bmatrix}^{\mathrm{T}}$$

$$T = \begin{bmatrix} T_{x1} & T_{y1} & T_{z1} & \cdots & T_{xn} & T_{yn} & T_{zn} \end{bmatrix}^{\mathrm{T}}$$

and

$$\begin{array}{lll} \mathbf{m}_{xi} = \mu_i f_i \mathbf{i}_i & \mathbf{n}_{xi} = i\mu_i f_i \mathbf{i}_i & T_{xi} = \mu_i f_i t_{xi} \\ \mathbf{m}_{yi} = \mu_i f_i \mathbf{j}_i & \mathbf{n}_{yi} = i\mu_i f_i \mathbf{j}_i & T_{yi} = \mu_i f_i t_{yi} \\ \mathbf{m}_{zi} = \mu_i \mathbf{k}_i & \mathbf{n}_{zi} = i\mu_i \mathbf{k}_i & T_{zi} = \mu_i t_{zi} \end{array} \tag{12}$$

The shape matrix is represented by,

$$\tilde{X} \sim \begin{bmatrix} S \\ \mathbf{1} \end{bmatrix} \tag{13}$$

where

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix}$$

and

$$\tilde{\mathbf{x}}_j = \begin{bmatrix} \nu_j \mathbf{s}_j^{\mathrm{T}} & \nu_j \mathbf{v}_j^{\mathrm{T}} & \nu_j \end{bmatrix}^{\mathrm{T}}$$

μ_i and ν_i are the scale factors for the homogeneous representations in Equations (2) and (3).

3.1. Moving coordinate system location

As the points are either static or moving linearly with constant speeds, the center of gravity of all the points is also moving linearly with constant speed. So is the center of gravity of the scaled feature points ($\nu_j \mathbf{p}_j$). Here we transform the 3D representations to a moving world coordinate system with fixed orientation (such as being aligned with the first camera) and the origin at the center of gravity of all the scaled points. Therefore,

$$\sum_{j=1}^{m} \nu_j \mathbf{p}_j = 0 \tag{14}$$

We get,

$$\begin{aligned} \sum_{j=1}^{m} \lambda_{ij} u_{ij} &= \sum_{j=1}^{m} (\mathbf{m}_{xi} \cdot \nu_j \mathbf{s}_j + \mathbf{n}_{xi} \cdot \nu_j \mathbf{v}_j + \nu_j T_{xi}) \\ &= \sum_{j=1}^{m} (\mathbf{m}_{xi} \cdot \nu_j \mathbf{s}_j + i\mathbf{m}_{xi} \cdot \nu_j \mathbf{v}_j + \nu_j T_{xi}) \\ &= \mathbf{m}_{xi} \cdot \sum_{j=1}^{m} \nu_j (\mathbf{s}_j + i\mathbf{v}_j) + T_{xi} \sum_{j=1}^{m} \nu_j \\ &= \mathbf{m}_{xi} \cdot \sum_{j=1}^{m} \nu_j \mathbf{p}_j + T_{xi} \sum_{j=1}^{m} \nu_j \\ &= T_{xi} \sum_{j=1}^{m} \nu_j \end{aligned} \tag{15}$$

Similarly,

$$\sum_{j=1}^{m} \lambda_{ij} v_{ij} = T_{yi} \sum_{j=1}^{m} \nu_j \quad \sum_{j=1}^{m} \lambda_{ij} = T_{zi} \sum_{j=1}^{m} \nu_j \tag{16}$$

Define the 7×7 projective transformation H as,

$$H = \begin{bmatrix} A & B \end{bmatrix} \tag{17}$$

where A is 7×6 and B is 7×1.
Since $\tilde{P} = \hat{P}H$,

$$\begin{bmatrix} M & T \end{bmatrix} = \hat{P} \begin{bmatrix} A & B \end{bmatrix} \tag{18}$$

we have,

$$T_{xi} = \hat{P}_{xi} B \quad T_{yi} = \hat{P}_{yi} B \quad T_{zi} = \hat{P}_{zi} B \tag{19}$$

From Equations (15) and (16),

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} u_{ij}}{\sum_{j=1}^{m} \lambda_{ij}} \quad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} v_{ij}}{\sum_{j=1}^{m} \lambda_{ij}} \tag{20}$$

we set up $2n$ linear equations of the 7 unknown elements of the matrix B. Least squares solutions are then computed.

3.2. Normalization

We recover the 7×6 matrix A by observing that the rows of the matrix M consist of $\mathbf{m}_i$, which are the scaled rotation axes by μ_i and the focal length f_i, and $\mathbf{n}_i$, which are the scaled $\mathbf{m}_i$ by the frame number i (Equation (12)), orthogonality of $\mathbf{m}_i$:

$$\begin{gathered} |\mathbf{m}_{xi}|^2 = |\mathbf{m}_{yi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0 \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} = 0 \end{gathered} \tag{21}$$

orthogonality of $\mathbf{n}_i$:

$$\begin{gathered} |\mathbf{n}_{xi}|^2 = |\mathbf{n}_{yi}|^2 \\ \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{n}_{xi} \cdot \mathbf{n}_{zi} = 0 \quad \mathbf{n}_{yi} \cdot \mathbf{n}_{zi} = 0 \end{gathered} \tag{22}$$

relationship of $\mathbf{m}_i$ and $\mathbf{n}_i$:

$$\begin{gathered} |\mathbf{n}_{xi}|^2 = i^2 |\mathbf{m}_{xi}|^2 \quad |\mathbf{n}_{yi}|^2 = i^2 |\mathbf{m}_{yi}|^2 \quad |\mathbf{n}_{zi}|^2 = i^2 |\mathbf{m}_{zi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{m}_{xi} \cdot \mathbf{n}_{zi} = 0 \\ \mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{n}_{zi} = 0 \\ \mathbf{m}_{zi} \cdot \mathbf{n}_{xi} = 0 \quad \mathbf{m}_{zi} \cdot \mathbf{n}_{yi} = 0 \end{gathered} \tag{23}$$

The above equations impose linear constraints on the elements of MM^{T}. Since $M = \hat{P}A$,,

$$MM^{\mathrm{T}} = \hat{P} A A^{\mathrm{T}} \hat{P}^{\mathrm{T}} \tag{24}$$

these constraints are linear on the elements of the symmetric matrix $Q = AA^{\mathrm{T}}$. Define,

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{25}$$

where A is a 7×6 matrix and A_1, A_2 are both 7×3 matrices. We get,

$$\begin{aligned} \hat{P}A_1 &= [\mathbf{m}_{x1}\ \mathbf{m}_{y1}\ \mathbf{m}_{z1} \cdots \mathbf{m}_{xn}\ \mathbf{m}_{yn}\ \mathbf{m}_{zn}]^{\mathrm{T}} \\ \hat{P}A_2 &= [\mathbf{n}_{x1}\ \mathbf{n}_{y1}\ \mathbf{n}_{z1} \cdots \mathbf{n}_{xn}\ \mathbf{n}_{yn}\ \mathbf{n}_{zn}]^{\mathrm{T}} \\ &= N [\mathbf{m}_{x1}\ \mathbf{m}_{y1}\ \mathbf{m}_{z1} \cdots \mathbf{m}_{xn}\ \mathbf{m}_{yn}\ \mathbf{m}_{zn}]^{\mathrm{T}} \end{aligned} \tag{26}$$

where

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & n \end{bmatrix} \tag{27}$$

according to Equation (12). Therefore,

$$\hat{P}A_2 = N\hat{P}A_1 \tag{28}$$

A_2 is over constrained given A_1 and $\hat{P}$,

$$A_2 = KA_1 \tag{29}$$

where

$$K = \hat{P}^{-1} N \hat{P} \tag{30}$$

and $\hat{P}^{-1}$ is the generalized inverse matrix which is $7 \times 3n$ and uniquely defined when $n \geq 3$.

From Equation (26), we see that Equation (21) imposes linear constraints on the 28 unknown elements of the 7×7 symmetric matrix $Q_1 = A_1 A_1^{\mathrm{T}}$, while Equation (22) imposes constraints on the 28 unknown elements of $Q_2 = A_2 A_2^{\mathrm{T}}$. From Equation (29) we have,

$$Q_2 = A_2 A_2^{\mathrm{T}} = KA_1 A_1^{\mathrm{T}} K^{\mathrm{T}} = KQ_1 K^{\mathrm{T}} \tag{31}$$

which translates the constraints on Q_2 to constraints on Q_1. Equation (23) imposes constraints on $Q_3 = A_2 A_1^{\mathrm{T}}$ which can also be translated into constraints on Q_1,

$$Q_3 = A_2 A_1^{\mathrm{T}} = KA_1 A_1^{\mathrm{T}} = KQ_1 \tag{32}$$

Therefore, each frame contributes 17 constraints (Equations (21) to (23)) on Q_1. In total, we get $17n$ linear equations on the 28 unknown elements of the symmetric matrix Q_1. Linear least squares solutions are computed. We then compute the matrix A_1 from Q_1 by rank 3 matrix decomposition and A_2 by Equation (29), so we recover the linear transformation A.

3.3. Shape reconstruction and camera calibration

Once the matrices A and B have been found, the projective transformation H is $[A\ B]$. The shape matrix is computed as $\tilde{X} = H^{-1}\hat{X}$ and the motion matrix as $\tilde{P} = \hat{P}H$. We first compute the scale factors μ_i,

$$\mu_i = |\mathbf{m}_{zi}| \tag{33}$$

We then compute the focal lengths as,

$$f_i = \frac{|\mathbf{m}_{xi}| + |\mathbf{m}_{yi}|}{2\mu_i} \tag{34}$$

Therefore, the camera motion parameters are,

$$\begin{array}{lll} \mathbf{i}_i = \frac{\mathbf{m}_{xi}}{\mu_i f_i} & \mathbf{j}_i = \frac{\mathbf{m}_{yi}}{\mu_i f_i} & \mathbf{k}_i = \frac{\mathbf{m}_{zi}}{\mu_i} \\ t_{xi} = \frac{T_{xi}}{\mu_i f_i} & t_{yi} = \frac{T_{yi}}{\mu_i f_i} & t_{zi} = \frac{T_{zi}}{\mu_i} \end{array} \tag{35}$$

The shape matrix consists of the scene structure and the velocities represented in the moving world coordinate system. We need to transform the representation back to a fixed coordinate system with the origin at the center of gravity of all the points at frame 1.

First the velocity of the moving coordinate system is computed. Since the system is moving at the average velocity of all the scaled moving points, the static points share the same velocity which is the negative value of the average velocity. It is often the case that there are more static points than the points from any moving object, so we let every point vote for a "common" velocity (denoted as $\mathbf{v}_c$). The velocity with the most votes is taken as the negative velocity of the moving coordinate system. The points with the "common" velocity are automatically classified as static and the scene structure is computed as,

$$\mathbf{sc}_j = \mathbf{s}_j + \mathbf{v}_c \tag{36}$$

where $\mathbf{sc}_j$ denotes the scene point position represented in the fixed coordinate system. According to Equation (1), $\mathbf{s}_j$ is the point position at frame 0.

The points which do not have the "common" velocity are the moving points. The number of the moving objects is therefore detected. Their starting positions represented in the fixed coordinate system are,

$$\mathbf{sm}_j = \mathbf{s}_j + \mathbf{v}_c \tag{37}$$

and their velocities are,

$$\mathbf{vm}_j = \mathbf{v}_j - \mathbf{v}_c \tag{38}$$

3.4. Algorithm outline

We summarize the algorithm as follows:

1. Perform SVD on W_s, get the projective motion $\hat{P}$ and the projective shape $\hat{X}$;
2. Sum up each row of W_s and compute the ratios between them as in Equation (20);
3. Set up $2n$ linear equations of the 7 unknown elements of the matrix B based on the ratios from step 2 and compute B;
4. Set up $17n$ linear equations of the 28 unknown elements of the symmetric matrix Q_1 by imposing constraints in Equations (21) to (23);
5. Factor Q_1 to get A_1 from $Q_1 = A_1 A_1^{\mathrm{T}}$;
6. Compute A_2 from $A_2 = KA_1$;
7. Combine A_1 and A_2 to generate the linear transformation matrix $A = [A_1\ A_2]$;
8. Put the matrices A and B together and get the projective transformation $H = [A\ B]$;
9. Recover the shape matrix using $\tilde{X} = H^{-1}\hat{X}$ and motion matrix using $\tilde{P} = \hat{P}H$;
10. Recover the focal lengths, the camera rotation axes and the translation vectors according to Equations (34) and (35).
11. Reconstruct the scene structure and the trajectories of the moving objects according to Equations (36) to (38).

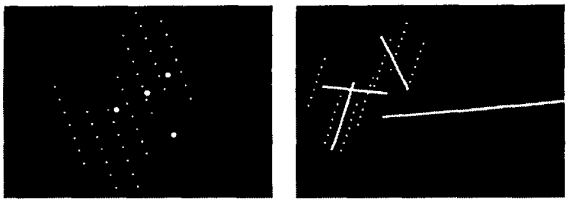
4. Experiments

In this section the experimental results on synthetic and real images are presented. The first set of experiments use synthetic images to evaluate the method quantitatively. The second experiment is conducted on a real image sequence taken by a hand-held camera of an indoor scene, and the reconstruction results are compared with the ground truth values.

4.1. Synthetic data

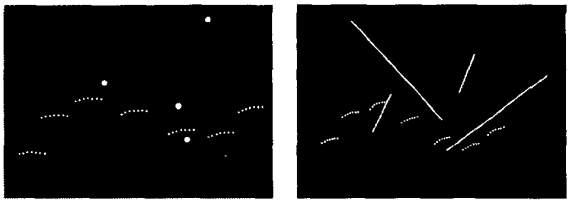
We generate 100 image sequences of the scene with 8 to 49 static feature points and 3 to 8 points moving in random directions. The frame number is 4 to 60. The shape of the static scene is a sweep of the sin curve in the space. The camera is rotating randomly through 30 to 40 degrees for each of roll, pitch and yaw. The distance between the camera and the center of gravity of all the static points is varied from 4 to 20 times the object size. We add 2 pixel standard noise to the feature locations from 640 × 480 images.

Figure 1 illustrates the case where 4 objects are moving randomly in 3D space. The method automatically detects the number of the moving objects as 4, reconstructs the



(a) (b)

Figure 1. Reconstruction of a scene with four moving objects by the method. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories.



(a) (b)

Figure 3. Reconstruction of a scene with four moving objects by the weak perspective method [11]. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories. The distortions are caused by the approximation of perspective cameras with weak perspective cameras.

static scene and the initial positions of the 4 moving objects, as shown in Figure 1(a). Figure 1(b) shows the trajectories of the moving objects as well as the static scene. There are 49 points from the static scene and 60 frames are taken.

Figure 2 plots the focal lengths recovered by the method and their ground truth values. The maximum error is 7.2% of the true value.

To compare the results, we apply the multiple motion scene reconstruction method for weak perspective cameras [11] to the same sequence using the true values of the focal lengths. The results are shown in Figure 3. It is easy to see that the reconstruction results have distortions which are caused by the approximation of perspective cameras with weak perspective cameras.

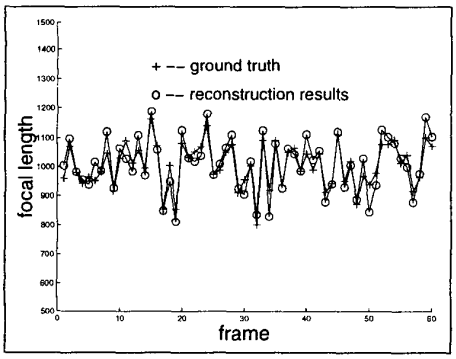


Figure 2. Comparison of the focal lengths recovered by the method and their ground truth values for the synthetic sequence. The maximum error is 7.2% of the true value.

To evaluate the quality of the reconstruction method, we measure the reconstruction error by comparison with the ground truth. Since the reconstruction from monocular image sequences is up to scale, we assume that the size of the static shape is 1. The maximum distance between the recovered static points and their known positions is 3.2%, the maximum error of the reconstructed initial positions of the moving objects is 4.1% and the velocity error is less than 1.9%. We also assess the quality of the camera motion reconstruction. The maximum distance between the recovered camera locations and the ground truth values is 5.4% and the maximum angle between the recovered camera orientations and the known values is 0.12°. The maximum reconstruction error of the focal lengths is 8.11% of the ground truth values.

4.2. Real data

This real data sequence was taken by a hand-held camera. There were three objects moving in the scene, including a toy car, a toy bird and a toy person. The objects were moving linearly with constant speeds. The car and the person were moving on the table. The speed of the car was $3.5cm$ per frame and the speed of the person was $2.5cm$ per frame. The bird was climbing the pole and moved $3.0cm$ per frame. The books and the box represented the static scene. The camera was zoomed out at the beginning and gradually zoomed in as it moved around the scene. The focal length was changed every two frames. 10 images were taken. Three of them are shown in Figure 4(a). 29 feature points were manually selected and tracked. Each moving object had one feature point selected.

The shapes of the books and the box, the starting positions of the toys and the motion velocities are recovered and demonstrated in Figure 4(b), the motion trajectories are overlaid in the images. Figure 4(c) shows the recovered camera locations and orientations. Figure 5 plots the recovered focal lengths, which shows that the focal lengths are changing with the camera motion as we expected. The largest focal length almost doubles the smallest one, which is correct for the $2\times$ optical lens.

We assess the quality of the reconstruction by comparison with the ground truth. The ratio between the speeds of

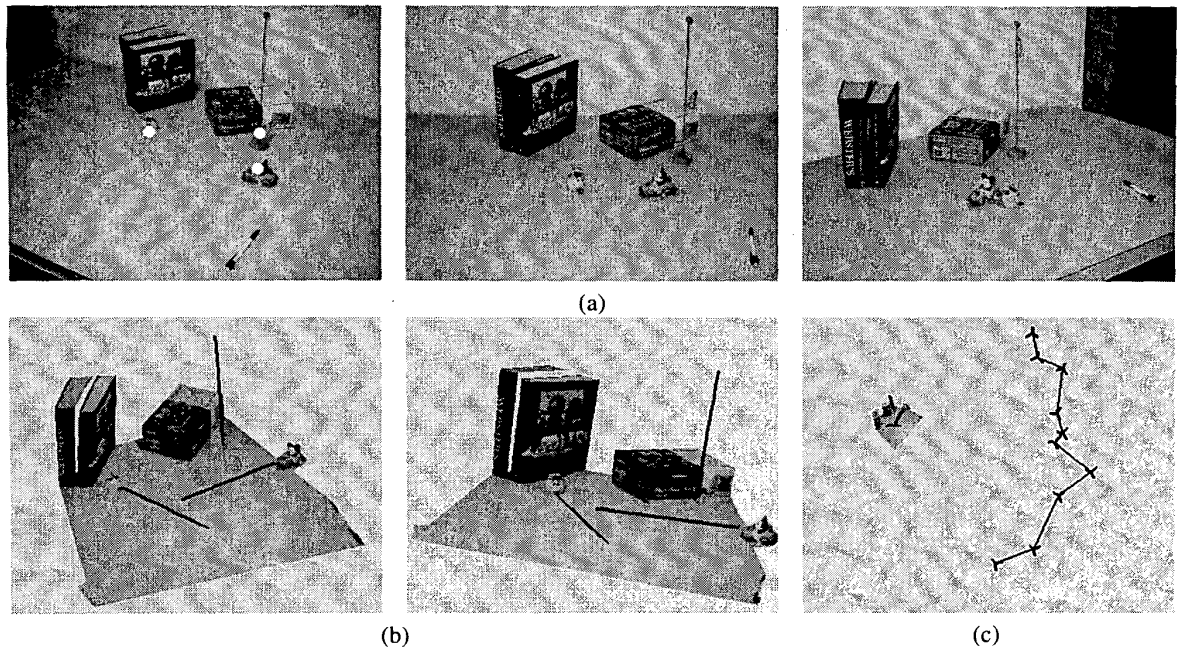


Figure 4. (a) 1st, 5th and 10th images of the real data, the white circles show the feature points selected on the moving objects in the 1st image. (b) Two views of the scene reconstruction with texture mapping, the black lines denote the recovered motion trajectories. (c) Reconstruction of the scene and the camera positions/orientations, the 3-axis figures are the recovered cameras.

the moving toys are 2.5 : 3.77 : 2.91 which are close to the expected value 2.5 : 3.5 : 3.0. The maximum distance between the positions of the recovered static points and the ground truth positions is 5mm. The angle between the recovered motion direction of the bird and the floor is 91.2°, which is close to the expected value.

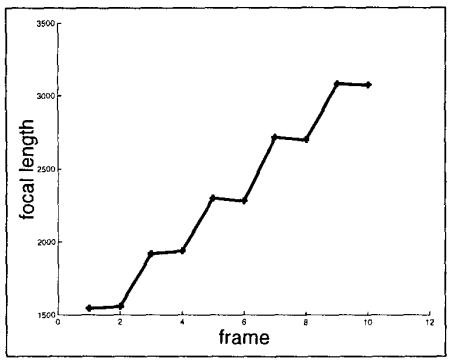


Figure 5. Focal lengths of the real data recovered by the method. The recovered values change every two frames as expected.

5. Minimum data requirement

The main advantage of the factorization-based method is using the heavily redundant information from multiple image features and views. However, it is equally important to compute the minimum data requirement of the method to analyze its practicality and reliability.

The low bound of data requirement is determined by the number of degrees of freedom of the reconstruction and the number of constraints given by each feature in each view. The input to the reconstruction method are the feature correspondences. Therefore, the number of constraints is $2nm$, where n is the number of views and m is the number of feature points. The output of the reconstruction consists of the scene structure ($3m$), the trajectories of the moving objects ($3m$), the camera motion ($6n$) and the camera intrinsic parameters (n). Euclidean reconstruction from monocular image sequences is up to a rigidity transformation which has 6 degrees of freedom, and the scale. This number should be subtracted from the total number of degrees of freedom. Therefore, the constraint is $2nm \geq 7n + 6m - 7$. The minimum data requirement is $n = 4, m = 11$.

The minimum data is also constrained by the solution process. It requires that the number of equations is larger than the number of variables. The multiple motion scene re-

construction method decouples the reconstruction into projective and Euclidean reconstruction. A total of $2nm$ measurements are available to estimate the projective motion and shape. Each camera projection is represented by a 3×7 matrix which has 20 variables because of the homogeneous representation. Each feature point is a 7×1 vector which has 6 variables. Since the projective reconstruction is up to an unknown 7×7 transformation, the total number of variables is $20n + 6m - 48$. In order to convert the projective reconstruction to the Euclidean one, the normalization algorithm sets up $17n$ equations to solve the 28 unknowns. Therefore, there are two constraints, $2nm \geq 20n + 6m - 48$ and $17n \geq 28$, for the reconstruction process. The minimum data required is $n = 4, m = 16$.

The above two computations of the minimum data only provide necessary conditions to carry out the reconstruction. We conduct a number of synthetic experiments to determine the minimum number of views and features required for reasonably accurate reconstructions. The empirical results show that the minimum data requirement is $n = 4, m = 16$.

6. Discussion

The method described in this paper solves the full rank case where the static structure and the motion space of the objects are both rank 3. In other words, the scene is three dimensional and the velocities of the moving objects span a three dimensional space. Degenerate cases, however, exist because either or both of shape and motion spaces are degenerate. The shape space is degenerate, for example, when all the points lie in a plane. The motion space of the moving objects is degenerate, when:

1. There is no moving object in the scene.
2. There is one moving object or multiple objects moving in the same and/or the opposite direction (not necessarily the same 3D line).
3. The velocities of the objects lie in a two dimensional space (not necessarily the same 3D plane).

When cameras are intrinsically calibrated, there are solutions to these degenerate cases as shown in [11]. Following the same line of work presented in this paper, it is easy to design the reconstruction algorithms for degenerate cases with uncalibrated cameras. However, the rank of the measurement matrix can not be used as a clue about which case is the best approximation under perspective projections. The measurement matrix is always full rank. Therefore, we assume that the rank approximation information is given though there is no requirement for prior motion segmentation and the rank does not depend on how many objects are moving. Detailed derivations can be found in [9].

References

[1] L. d. Agapito, R. Hartley, and E. Hayman. Linear self-calibration of a rotating and zooming camera. In *CVPR99*, pages 15–21, 1999.

[2] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *PAMI*, 22(4):348–357, April 2000.

[3] P. Beardsley, P. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. In *ECCV96*, pages II:683–695, 1996.

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR00*, pages II:690–696, 2000.

[5] S. Christy and R. Horaud. Euclidean reconstruction: From paraperspective to perspective. In *ECCV96*, pages II:129–140, 1996.

[6] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *PAMI*, 18(11):1098–1104, November 1996.

[7] J. Costeira and T. Kanade. A multibody factorization method for independently moving-objects. *IJCV*, 29(3):159–179, 1998.

[8] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV92*, pages 563–578, 1992.

[9] M. Han. Linear and bilinear subspace methods for structure from motion. *Ph.D. Thesis*, February 2001.

[10] M. Han and T. Kanade. Creating 3d models with uncalibrated cameras. In *WACV00*, pages 178–185, 2000.

[11] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *CVPR00*, pages II:542–549, 2000.

[12] R. Hartley. Euclidean reconstruction from uncalibrated views. In *CVPR94*, pages 908–912, 1994.

[13] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *SCIA97*, 1997.

[14] A. Heyden and K. Astrom. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *CVPR97*, pages 438–443, 1997.

[15] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 19(3):206–218, 1997.

[16] M. Pollefeys, R. Koch, and L. VanGool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *IJCV*, 32(1):7–25, August 1999.

[17] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *PAMI*, 19(8):834–845, August 1997.

[18] A. Shashua, S. Avidan, and M. Werman. Trajectory triangulation over conic sections. In *ICCV99*, pages 330–336, 1999.

[19] A. Shashua and L. Wolf. Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *ECCV00*, 2000.

[20] A. Shashua and L. Wolf. On the structure and properties of the quadrifocal tensor. In *ECCV00*, 2000.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, 1992.

[22] B. Triggs. Factorization methods for projective structure and motion. In *CVPR96*, pages 845–851, 1996.

[23] Y. Wexler and A. Shashua. On the synthesis of dynamic scenes from reference views. In *CVPR00*, pages II:576–581, 2000.