International Journal of Computer Vision 59(3), 285–300, 2004
© 2004 Kluwer Academic Publishers Manufactured in The Netherlands

Reconstruction of a Scene with Multiple Linearly Moving Objects

MEI HAN*
NEC Laboratories America
meihan@sv nec-labs com

TAKEO KANADE
Robotics Institute, Carnegie Mellon University
tk@cs cmu edu

Received November 2, 2001; Revised July 21, 2003; Accepted December 9, 2003

Abstract. In this paper we describe an algorithm to recover the scene structure, the trajectories of the moving objects and the camera motion simultaneously given a monocular image sequence. The number of the moving objects is automatically detected without prior motion segmentation. Assuming that the objects are moving linearly with constant speeds, we propose a unified geometrical representation of the static scene and the moving objects. This representation enables the embedding of the motion constraints into the scene structure, which leads to a factorization-based algorithm. We also discuss solutions to the degenerate cases which can be automatically detected by the algorithm. Extension of the algorithm to weak perspective projections is presented as well. Experimental results on synthetic and real images show that the algorithm is reliable under noise.

Keywords: structure from motion, motion segmentation, dynamic scene reconstruction, computer vision

1. Introduction

Many interesting problems have been discussed on monocular image sequences of scenes with multiple moving objects, such as scene reconstruction (Kumar et al., 1994; Poelman and Kanade, 1997; Han and Kanade, 1998; Irani et al., 2002), motion segmentation (Irani et al., 1992; Torr and Murray, 1993; Sawhney et al., 1999), reconstruction of motion trajectories (Avidan and Shashua, 2000), camera motion recovery (Irani et al., 1997; Costeira and Kanade, 1998) and scene synthesis (Wexler and Shashua, 2000). Most of these methods deal with the above problems separately. However, the temporal integration of information over sequences provides constraints on the scene reconstruction. We are therefore motivated to seek a one step reconstruction algorithm.

In aerial video sequences, the moving objects are often far from the camera. It is difficult to get multiple feature points from each moving object. It is a good approximation to abstract the moving objects as points. As pointed out in Avidan and Shashua (2000), recovering the locations of a moving point from a monocular image sequence is impossible without assumptions about its trajectory. We assume that the objects are moving linearly with constant speeds, and different objects can have different velocities. This assumption is for the objects in 3D scene instead of for their 2D projections. Due to the random camera motion, the images of the objects rarely move with constant velocities over the sequences. This assumption is reasonable for most moving objects, such as cars, planes and people, especially for short time intervals.

*The research described in this paper was conducted while the first author was a Ph.D. student in the Robotics Institute at Carnegie Mellon University.

We propose a unified representation of the static scene and the moving objects in which each point has an initial position and a constant velocity. Points on the static scene are defined to have zero velocity. This representation embeds the linear motion constraints within the scene structure, and naturally leads to a factorization-based algorithm which considers all the data in all the images uniformly. This algorithm reconstructs the scene structure, the trajectories of the moving objects and the camera motion simultaneously. The number of the moving objects is automatically detected without prior motion segmentation. We also discuss solutions to the degenerate cases and extension of the algorithm to weak perspective projections. Experiments on synthetic and real images are presented.

1.1. Related Work

Zelnik-Manor and Irani propose using subspace constraints on multi-frame information to compute homography (Zelnik-Manor and Irani, 1999) and optical flows (Irani, 1999). Their work demonstrates that the use of geometric constraints provides a good way to integrate information over the sequence. Avidan and Shashua recover the linear trajectory of a 3D point by line fitting (Avidan and Shashua, 1999). They assume that the object is moving along a line, but they do not require that the object is moving with constant speed. In order to perform the triangulation of lines, they assume that the camera motion is given as well as the prior motion segmentation. They do not recover the scene structure. They extend this work to conic shape trajectories in Shashua et al. (1999). Shashua and Wolf propose the concept of *Homography Tensor* to represent three views of static and moving planar points in Shashua and Wolf (2000). They also introduce the higher-dimensional mappings for the representation of various dynamic scene reconstructions (Wolf and Shashua, 2001). Bregler et al. 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 (Bregler et al., 2000). The complexity of their solution increases with the number of basis shapes.

Manning and Dyer present an algorithm for dynamic view interpolation using view morphing based on known camera-to-camera transformations and motion segmentation (Manning and Dyer, 1999). Wexler and Shashua describe the synthesis of dynamic scenes from three reference views without knowledge of camera motion or motion segmentation (Wexler and Shashua, 2000). Both of these algorithms assume that the objects are moving with constant speeds along straight lines, and they require the coplanar configuration of static and moving objects.

The algorithm presented in this paper uses the factorization technique as the basis of solution. The factorization methods, first developed by Tomasi and Kanade for orthographic views (Tomasi and Kanade, 1992) and extended by Poelman and Kanade to weak and para perspective views (Poelman and Kanade, 1997), achieve their robustness and accuracy by applying the singular value decomposition (SVD) to a large number of images and feature points, but none of these methods works on scenes with moving objects. The multibody factorization method proposed by Costeira and Kanade reconstructs the motions and shapes of independently moving objects (Costeira and Kanade, 1998). This method regards each moving object as one subspace, therefore, it requires that each object has multiple feature points.

2. Feature Points Representation

We propose a unified representation of the static scene and the moving objects (Han and Kanade, 1999). 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 feature point as a moving point with constant velocity: the static points simply have zero velocity. Any point $\mathbf{p}_{ij}$ is represented by,

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

in a world coordinate system, where $i = 1 \quad n$ and $j = 1 \quad m$. 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 moving velocity.

We first use orthographic camera model for the derivations. We describe the extension of this work to weak perspective camera models in Section 5. If a point $\mathbf{p}_{ij}$ is observed in frame i at image coordinates (u_{ij}, v_{ij}), then,

$$\begin{aligned} u_{ij} &= \mathbf{i}_i \quad \mathbf{p}_{ij} + t_{xi} \\ v_{ij} &= \mathbf{j}_i \quad \mathbf{p}_{ij} + t_{yi} \end{aligned} \tag{2}$$

$\mathbf{i}_i$ and $\mathbf{j}_i$ are the rotation axes of the ith camera. t_{xi} and t_{yi} are its translations. Therefore,

$$\begin{aligned} u_{ij} &= \mathbf{i}_i \quad \mathbf{s}_j + i\, \mathbf{i}_i \quad \mathbf{v}_j + t_{xi} \\ v_{ij} &= \mathbf{j}_i \quad \mathbf{s}_j + i\, \mathbf{j}_i \quad \mathbf{v}_j + t_{yi} \end{aligned} \tag{3}$$

We put all the feature points coordinates $(u_{ij} \;\; v_{ij})$ in a $2n \times m$ measurement matrix W,

$$W = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ v_{11} & v_{12} & & v_{1m} \\ & \vdots & \vdots & \\ u_{n1} & u_{n2} & & u_{nm} \\ v_{n1} & v_{n2} & \cdots & v_{mn} \end{bmatrix} \tag{4}$$

Each column of W contains the observations for a single point, and each row contains the observed u-coordinates or v-coordinates for a single frame. According to Eq. (3), we have,

$$W = MS + T[1 \quad 1 \quad \cdots \quad 1] \tag{5}$$

with the motion matrix,

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{x2} & \mathbf{n}_{y2} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} \end{bmatrix}^{\mathrm{I}} \tag{6}$$

where

$$\begin{aligned} \mathbf{m}_{xi} &= \mathbf{i}_i \quad \mathbf{n}_{xi} = i\, \mathbf{i}_i \\ \mathbf{m}_{yi} &= \mathbf{j}_i \quad \mathbf{n}_{yi} = i\, \mathbf{j}_i \end{aligned} \tag{7}$$

and the shape matrix,

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix} \tag{8}$$

The translation vector T is,

$$T = [t_{x1} \quad t_{y1} \quad t_{x2} \quad t_{y2} \quad \cdots \quad t_{xn} \quad t_{yn}]^{\mathrm{I}} \tag{9}$$

The constraints of the objects moving linearly with constant speeds enable the unified representation of the motion matrix M, composed of the rotation axes ($\mathbf{m}_{xi}$ and $\mathbf{m}_{yi}$) and the scaled rotation axes ($\mathbf{n}_{xi}$ and $\mathbf{n}_{yi}$), and of the shape matrix, composed of the scene structure ($\mathbf{s}_j$) and the motion velocities ($\mathbf{v}_j$).

3. Scene Reconstruction

In this section we describe the scene reconstruction algorithm based on the unified representation of the static scene and the moving objects (Han and Kanade, 2000). The algorithm factors the measurement matrix into the product of the unified motion matrix, which is a combination of the rotation and the scaled rotation axes, and the unified shape matrix, which is a combination of the initial positions of the feature points and their velocities.

3.1 *Moving World Coordinate System Location*

As a set of points are either static or moving linearly at constant speeds, the center of gravity of all the points is moving linearly at a constant speed as well. The velocity of the center of gravity is equal to the average of all the velocities ($\mathbf{v}_j$). We transform the 3D representation to a *moving* world coordinate system whose origin is at the center of gravity of all the feature points and with a fixed orientation (such as being aligned with the first camera). Therefore,

$$\sum_{j=1}^{m} \mathbf{p}_{ij} = 0 \tag{10}$$

From Eq. (2), we have,

$$\begin{aligned} \sum_{j=1}^{m} u_{ij} &= \sum_{j=1}^{m} (\mathbf{i}_i \quad \mathbf{p}_{ij} + t_{xi}) = \mathbf{i}_i \sum_{j=1}^{m} \mathbf{p}_{ij} + m t_{xi} = m t_{xi} \\ \sum_{j=1}^{m} v_{ij} &= \sum_{j=1}^{m} (\mathbf{j}_i \quad \mathbf{p}_{ij} + t_{yi}) = \mathbf{j}_i \sum_{j=1}^{m} \mathbf{p}_{ij} + m t_{yi} = m t_{yi} \end{aligned} \tag{11}$$

We can compute the translation vector directly from Eq. (11),

$$\begin{aligned} t_{xi} &= \frac{1}{m} \sum_{j=1}^{m} u_{ij} \\ t_{yi} &= \frac{1}{m} \sum_{j=1}^{m} v_{ij} \end{aligned} \tag{12}$$

3.2 *Decomposition*

Once the translation vector T is known, we subtract it from W in Eq. (5),

$$\begin{aligned}\hat{W} &= W - T[1 \quad 1 \quad \cdots \quad 1] \\ &= \hat{M}\hat{S} = \hat{M}AA^{-1}\hat{S} = MS\end{aligned} \tag{13}$$

where $M = \hat{M}A$ and $S = A^{-1}\hat{S}$. According to the representations of M and S in Eqs. (6) and (8), we know that the rank of the matrix $\hat{W}$ is at most 6 no matter how many moving objects are there. We perform a SVD on $\hat{W}$ and get the best possible rank 6 approximation of $\hat{W}$ as $\hat{M}\hat{S}$, where $\hat{M}$ is a $2n \times 6$ matrix and $\hat{S}$ is a $6 \times m$ matrix. This decomposition is not unique. Any non-singular 6×6 matrix A could be inserted between $\hat{M}$ and $\hat{S}$ to get another motion and shape pair.

3.3 *Normalization*

Metric constraints are imposed to translate the current pair of motion ($\hat{M}$) and shape ($\hat{S}$) to the Euclidean solutions through recovering the linear transformation A. This process is called *normalization*. We define,

$$A = [A_1 \quad A_2] \tag{14}$$

where A is a 6×6 matrix and A_1, A_2 are both 6×3 matrices. Since $M = \hat{M}A$,

$$\begin{aligned}M_1 &= \hat{M}A_1 = [\mathbf{m}_{x1} \quad \mathbf{m}_{y1} \quad \cdots \quad \mathbf{m}_{xn} \quad \mathbf{m}_{yn}]^{\mathrm{I}} \\ M_2 &= \hat{M}A_2 = [\mathbf{n}_{x1} \quad \mathbf{n}_{y1} \quad \cdots \quad \mathbf{n}_{xn} \quad \mathbf{n}_{yn}]^{\mathrm{I}} \\ &= N[\mathbf{m}_{x1} \quad \mathbf{m}_{y1} \quad \cdots \quad \mathbf{m}_{xn} \quad \mathbf{m}_{yn}]^{\mathrm{I}}\end{aligned} \tag{15}$$

where

$$N = \mathrm{diag}(1, 1, 2, 2, \ldots, n, n) \tag{16}$$

according to Eq. (7). We recover the 6×6 matrix A by observing that the rows of the motion matrix M consist of the rotation axes and the scaled ones (Eq. (6)),

$$|\mathbf{m}_{xi}|^2 = 1 \quad |\mathbf{m}_{yi}|^2 = 1 \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0 \tag{17}$$

$$|\mathbf{n}_{xi}|^2 = i^2 \quad |\mathbf{n}_{yi}|^2 = i^2 \quad \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} = 0 \tag{18}$$

$$\mathbf{m}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = 0 \tag{19}$$

The above equations impose linear constraints on the elements of $M_1M_1^{\mathrm{T}}$, $M_2M_2^{\mathrm{T}}$ and $M_2M_1^{\mathrm{T}}$. Since

$$\begin{aligned}M_1M_1^{\mathrm{I}} &= \hat{M}A_1A_1^{\mathrm{I}}\hat{M}^{\mathrm{I}} \\ M_2M_2^{\mathrm{I}} &= \hat{M}A_2A_2^{\mathrm{I}}\hat{M}^{\mathrm{I}} \\ M_2M_1^{\mathrm{I}} &= \hat{M}A_2A_1^{\mathrm{I}}\hat{M}^{\mathrm{I}}\end{aligned} \tag{20}$$

these constraints are linear on the elements of the matrices $Q_1 = A_1A_1^{\mathrm{T}}$, $Q_2 = A_2A_2^{\mathrm{T}}$ and $Q_3 = A_2A_1^{\mathrm{T}}$.

According to Eq. (15),

$$\hat{M}A_2 = N\hat{M}A_1 \tag{21}$$

The matrix A_2 is over constrained given A_1 and $\hat{M}$ by,

$$A_2 = KA_1 \tag{22}$$

where

$$K = \hat{M}^{-1}N\hat{M} \tag{23}$$

and $\hat{M}^{-1}$ is the pseudo inverse matrix which is $6 \times 2n$ and uniquely defined when $n \geq 3$.

From Eq. (20), we see that Eq. (17) imposes constraints on the 21 unknown elements of the 6×6 symmetric matrix Q_1, while Eq. (18) imposes constraints on the 21 elements of Q_2. From the relation of A_1 and A_2 (Eq. (22)), we have,

$$Q_2 = A_2A_2^{\mathrm{I}} = KA_1A_1^{\mathrm{I}}K^{\mathrm{I}} = KQ_1K^{\mathrm{I}} \tag{24}$$

which translates the constraints on Q_2 to the constraints on Q_1.

Equation (19) imposes constraints on Q_3 which can also be translated into constraints on Q_1,

$$Q_3 = A_2A_1^{\mathrm{I}} = KA_1A_1^{\mathrm{I}} = KQ_1 \tag{25}$$

Therefore, each frame contributes 8 constraints (Eqs. (17) to (19)) on Q_1. In total, we have $8n$ equations on the 21 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 Eq. (22), so we recover the linear transformation $A = [A_1 \quad A_2]$.

3.4. Motion and Shape Reconstruction

Once the matrix A has been found, the shape matrix is computed using $S = A^{-1}\hat{S}$ and the motion matrix is $M = \hat{M}A$. We compute the camera rotation axes as,

$$\mathbf{i}_i = \mathbf{m}_{xi} \quad \mathbf{j}_i = \mathbf{m}_{yi} \quad \mathbf{k}_i = \mathbf{m}_{xi} \times \mathbf{m}_{yi} \tag{26}$$

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 world coordinate system with the origin at the center of gravity of all the points at frame 1.

First we compute the velocity of the moving coordinate system. Since the system is moving at the average velocity of all the moving points, the static points must have the same velocity which is the negative value of the average velocity. It is often the case that there are more static points than moving points from any moving object, so we try to find a "common" velocity (denoted as $\mathbf{v}_c$) among the recovered velocities. In the presence of noise different static points have different velocities. We apply the RANSAC scheme (Fischler and Bolles, 1981) to get the "common" velocity. For each point we identify its outlier points requiring that the difference between the velocity of the outlier point and that of the point is larger than some threshold.[1] The process is repeated for all the points and the best representative point is chosen which has the minimal number of outlier points. The points which are not identified as outliers of the best representative point, including the point itself, are classified as static points. The average velocity of all the static points is the "common" velocity which is used as the negative velocity of the moving coordinate system. The static scene structure is computed as:

$$\mathbf{sc}_j = \mathbf{s}_j + \mathbf{v}_c \tag{27}$$

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

The outlier points of the best representative point are the moving points. The number of the moving objects is therefore detected. Their starting positions represented in the fixed coordinate system are:

$$\mathbf{sm}_j = \mathbf{s}_j + \mathbf{v}_c \tag{28}$$

and their velocities are:

$$\mathbf{vm}_j = \mathbf{v}_j - \mathbf{v}_c \tag{29}$$

3.5. Algorithm Outline

We summarize the reconstruction algorithm as follows:

1. Compute the camera translations T from the matrix W according to Eq. (12);
2. Subtract T from W to generate $\hat{W}$ according to Eq. (13);
3. Perform SVD on $\hat{W}$ and get $\hat{M}$ and $\hat{S}$;
4. Set up linear equations of the 21 unknown elements of the symmetric matrix Q_1 by imposing constraints in Eqs. (17) to (19);
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. Recover the shape matrix using $S = A^{-1}\hat{S}$ and the motion matrix using $M = \hat{M}A$;
9. Recover the camera rotation axes as in Eq. (26);
10. Detect the moving objects, reconstruct the scene structure and the trajectories of the moving objects according to Eqs. (27) to (29).

4. Degenerate Cases

The algorithm described in Section 3 solves the case where the "registered" measurement matrix $\hat{W}$ (the matrix generated by subtraction of translations from the original measurement matrix) has full rank 6, that is, where the static structure and the motion space of the objects are both rank 3. Equivalently, this is the case that the scene is three dimensional and the velocities of the moving objects span a three dimensional space. In this section we discuss degenerate cases.

If the scene has a degenerate shape, such as all the points lie in a plane, the plane plus parallax method (Irani et al., 1998) can detect the situation and solve for the scene structure (plane position), the camera motion and the motion segmentation (Kumar et al., 1994; Irani et al., 1997). The motion trajectories can be recovered using the method proposed by Avidan and Shashua (1999), given the reconstruction of the camera motion. Therefore, in this section we only discuss the solutions to the degenerate motion space of the objects.

We classify the degenerate situations into three classes:

1. *Rank-3 case*: The matrix $\hat{W}$ has rank 3. This corresponds to the situation where there is no moving object in the scene. The one-object factorization method (Tomasi and Kanade, 1992) is used to recover the scene structure and the camera motion.
2. *Rank-4 case*: The matrix $\hat{W}$ has rank 4. This corresponds to the situation where there is one moving object or multiple objects moving in the same and/or the opposite direction (not necessarily the same 3D line). Section 4.2 describes a linear algorithm for this case.
3. *Rank-5 case*: The matrix $\hat{W}$ has rank 5. This corresponds to the situation where the velocities of the objects lie in a two dimensional space (not necessarily the same 3D plane). Section 4.3 gives a nonlinear solution to this case.

4.1. Rank Approximation

Given tracked feature points, we first need to decide which case (full rank or one of the above three degenerate cases) is the best approximation. The rank of the matrix $\hat{W}$ is one important clue. However, finding the rank of $\hat{W}$ is not straightforward. Both inaccuracies in feature locations and approximation of perspective projection using orthographic (or weak perspective or para perspective) projections induce noises in the rank computation.

We use an algorithm similar to Boult and Brown (1991) and Irani (1999) to detect the rank of $\hat{W}$. We first estimate the noise level of the input images and approximate the rank using the singular values of $\hat{W}$ and the noise level. We refer to this method as *direct rank approximation*. In Gear (1998), Gear proposed a maximum likelihood method to estimate the grouping of points in the presence of noise. One of the core techniques of the method is rank approximation. He evaluated the grouping errors of all the possible rank values based on the statistical noise model. The rank value with the minimum error is chosen as the best rank approximation. We applied Gear's idea to the multiple motion scene reconstruction method where the rank of $\hat{W}$ can only be any value in $\{3, 4, 5, 6\}$, which is determined by the motion space of the objects and is not dependent on the number of moving objects. For each rank value in $\{3, 4, 5, 6\}$, we perform the scene reconstruction with moving objects and measure the errors in orthogonality of the recovered camera rotation matrices as well as the discrepancies of the feature points back projections. The best rank approximation is the one with the minimum error. The results show that the direct rank approximation method gives reliable estimations of the rank at most times.

4.2. Rank-4 Case

When only one moving object is in the scene, or when all moving objects travel in the same or the opposite direction, the motion space is one dimensional and the rank of the "registered" measurement matrix is 4. In this case we align the **x** direction of the world coordinate system with the motion direction. The system is still moving with the constant velocity. Therefore, the motion and shape matrices are (compared with Eqs. (6) and (8)),

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ i_{x1} & j_{x1} & 2i_{x2} & 2j_{x2} & \cdots & ni_{xn} & nj_{xn} \end{bmatrix}^{\mathrm{T}}$$

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \cdots & v_{xm} \end{bmatrix} \tag{30}$$

where i_{xi} and j_{xi} represent the x-elements of the ith rotation axes, v_{xj} denotes the x-element of the velocity of the jth feature point. We apply similar derivations as in the full rank case to the computation of T (Eq. (12)) and the decomposition of $\hat{W}$ (Eq. (13)). In this case the rank of $\hat{W}$ is 4. We perform a rank 4 matrix decomposition on $\hat{W}$ and get a $2n \times 4$ matrix $\hat{M}$ and a $4 \times m$ matrix $\hat{S}$. Now the linear transformation matrix A is 4×4. Similarly, we define,

$$A = [A_1 \quad A_2] \tag{31}$$

where A_1 is 4×3, A_2 is 4×1 and we have,

$$A_2 = K(A_1)_1 \tag{32}$$

where $(A_1)_1$ is the first column of A_1 and K is defined in Eq. (23). Since the matrix M consists of the rotation axes and only the x elements of the scaled rotation axes, the constraints in Eqs. (18) and (19) cannot be represented as linear constraints on the elements of $M_2 M_2^{\mathrm{T}}$ and $M_2 M_1^{\mathrm{T}}$. However, the constraints in Eq. (17) still hold and provide full rank linear equations on the 10 unknown elements of the symmetric 4×4 matrix $Q_1 = A_1 A_1^{\mathrm{T}}$. Least squares solutions are computed.

We then compute A_1 by rank 3 matrix decomposition of Q_1. This decomposition is up to a three dimensional rotation R since the matrix Q_1 is symmetric. When the motion space is full rank, any rotation matrix R provides a valid reconstruction with a different orientation of the world coordinate system. We usually fix the matrix R by aligning the world coordinate system with the first camera orientation. However, when the motion space is degenerate, the alignment is constrained to make the orientation of the world coordinate system consistent with the motion direction(s).

In rank-4 case, we need to align the **x** direction of the world coordinate system as the motion direction before we compute A_2 according to Eq. (32). The matrix R is determined by aligning the matrix $\hat{M}KA_1$ with the matrix $N\hat{M}A_1$.

Therefore, the linear transformation A is,

$$A = [A_1 R \quad K(A_1 R)_1] \tag{33}$$

where $()_1$ denotes the first column of the matrix. We apply a derivation similar to the one in Section 3.4 to recover the motion and shape.

4.3. *Rank-5 Case*

When the velocities of all the moving objects lie in a two dimensional space, we assume that the **x**-**y** plane of the world coordinate system is aligned with the two dimensional motion space. The system is still moving with constant velocity. Therefore, the motion and shape matrices are,

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ i_{x1} & j_{x1} & 2i_{x2} & 2j_{x2} & \cdots & ni_{xn} & nj_{xn} \\ i_{y1} & j_{y1} & 2i_{y2} & 2j_{y2} & \cdots & ni_{yn} & nj_{yn} \end{bmatrix}^{\mathrm{I}}$$

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \cdots & v_{xm} \\ v_{y1} & v_{y2} & \cdots & v_{ym} \end{bmatrix} \tag{34}$$

where i_{xi} and j_{xi} represent the x-elements of the ith rotation axes, i_{yi} and j_{yi} represent their y-elements, v_{xj} denotes the x-element of the velocity of the jth feature point and v_{yj} is its y-element. Therefore, the rank of $\hat{W}$ is 5. Similar derivations apply to the computation of T (Eq. (12)) and the decomposition of $\hat{W}$ (Eq. (13)). In this case we perform a rank 5 matrix decomposition on $\hat{W}$ and get a $2n \times 5$ matrix $\hat{M}$ and a $5 \times m$ matrix $\hat{S}$.

The linear transformation matrix A is 5×5. Similarly, we define

$$A = [A_1 \quad A_2] \tag{35}$$

where A_1 is 5×3 and A_2 is 5×2,

$$A_2 = K(A_1)_{12} \tag{36}$$

where $(A_1)_{12}$ denotes the first two columns of A_1 and K is defined in Eq. (23). Here only the constraints in Eq. (17) can be represented as linear constraints on the elements of $Q_1 = A_1 A_1^{\mathrm{T}}$. In this case the constraints are not sufficient to solve for the 15 unknown elements of the symmetric 5×5 matrix Q_1 linearly.

The constraints in Eqs. (18) and (19) can be represented as constraints on the elements of Q_1 and the five elements of the third column of A_1, which is a 5×1 vector denoted by $\mathbf{c}$. According to Eq. (36),

$$[A_2 \quad K\mathbf{c}] = KA_1 \tag{37}$$

we have,

$$\begin{aligned} A_2 A_2^{\mathrm{I}} &= KA_1 A_1^{\mathrm{I}} K^{\mathrm{I}} - K\mathbf{c}\mathbf{c}^{\mathrm{I}} K^{\mathrm{I}} \\ &= KQ_1 K^{\mathrm{I}} - K\mathbf{c}\mathbf{c}^{\mathrm{I}} K^{\mathrm{I}} \end{aligned} \tag{38}$$

and

$$\begin{aligned} [A_2 \quad i\mathbf{c}]A_1^{\mathrm{I}} &= A_2 (A_1)_{12}^{\mathrm{I}} + i\mathbf{c}\mathbf{c}^{\mathrm{I}} \\ &= KA_1 A_1^{\mathrm{T}} - K\mathbf{c}\mathbf{c}^{\mathrm{I}} + i\mathbf{c}\mathbf{c}^{\mathrm{I}} \\ &= KQ_1 - K\mathbf{c}\mathbf{c}^{\mathrm{I}} + i\mathbf{c}\mathbf{c}^{\mathrm{I}} \end{aligned} \tag{39}$$

Since,

$$\begin{aligned} \mathbf{m}_{xi}^{\mathrm{I}} &= \hat{\mathbf{m}}_x^{(i)\mathrm{I}} A_1 \quad \mathbf{n}_{xi}^{\mathrm{I}} = [\hat{\mathbf{m}}_x^{(i)\mathrm{I}} A_2 \quad i\,\hat{\mathbf{m}}_x^{(i)\mathrm{I}} \mathbf{c}] \\ \mathbf{m}_{yi}^{\mathrm{I}} &= \hat{\mathbf{m}}_y^{(i)\mathrm{I}} A_1 \quad \mathbf{n}_{yi}^{\mathrm{I}} = [\hat{\mathbf{m}}_y^{(i)\mathrm{I}} A_2 \quad i\,\hat{\mathbf{m}}_y^{(i)\mathrm{I}} \mathbf{c}] \end{aligned} \tag{40}$$

according to Eq. (15) $\hat{\mathbf{m}}_x^{(i)\mathrm{I}}$ and $\hat{\mathbf{m}}_y^{(i)\mathrm{I}}$ represent the ith x and y rows of the matrix $\hat{M}$. They are both 1×5 vectors. We translate the constraints in Eq. (18) to the constraints on Q_1 and $\mathbf{c}$ according to Eq. (38),

$$\begin{aligned} |\mathbf{n}_{xi}|^2 &= \hat{\mathbf{m}}_x^{(i)\mathrm{I}} A_2 A_2^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} + i^2 \hat{\mathbf{m}}_x^{(i)\mathrm{I}} \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} \\ &= \hat{\mathbf{m}}_x^{(i)\mathrm{I}} KQ_1 K^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} - \hat{\mathbf{m}}_x^{(i)\mathrm{I}} K\mathbf{c}\mathbf{c}^{\mathrm{I}} K^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} \\ &\quad + i^2 \hat{\mathbf{m}}_x^{(i)\mathrm{I}} \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} = i^2 \end{aligned} \tag{41}$$

and,

$$
\begin{aligned}
|\mathbf{n}_{yi}|^2 &= \hat{\mathbf{m}}_y^{(i)\mathrm{I}} K Q_1 K^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} - \hat{\mathbf{m}}_y^{(i)\mathrm{I}} K \mathbf{c}\mathbf{c}^{\mathrm{I}} K^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} \\
&\quad + i^2 \hat{\mathbf{m}}_y^{(i)\mathrm{I}} \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} = i^2 \\
\mathbf{n}_{xi} \cdot \mathbf{n}_{yi} &= \hat{\mathbf{m}}_x^{(i)\mathrm{I}} K Q_1 K^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} - \hat{\mathbf{m}}_x^{(i)\mathrm{I}} K \mathbf{c}\mathbf{c}^{\mathrm{I}} K^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} \\
&\quad + i^2 \hat{\mathbf{m}}_x^{(i)\mathrm{I}} \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} = 0
\end{aligned} \tag{42}
$$

Similarly, we translate the constraints in Eq. (19) to the constraints on Q_1 and $\mathbf{c}$ according to Eq. (39),

$$
\begin{aligned}
\mathbf{m}_{xi} \cdot \mathbf{n}_{yi} &= \hat{\mathbf{m}}_y^{(i)\mathrm{I}} [A_2 \;\; i\mathbf{c}] A_1^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} \\
&= \hat{\mathbf{m}}_y^{(i)\mathrm{I}} K Q_1 \hat{\mathbf{m}}_x^{(i)} - \hat{\mathbf{m}}_y^{(i)\mathrm{I}} K \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} \\
&\quad + i \hat{\mathbf{m}}_y^{(i)\mathrm{I}} \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_x^{(i)} = 0
\end{aligned} \tag{43}
$$

and

$$
\begin{aligned}
\mathbf{m}_{yi} \cdot \mathbf{n}_{xi} &= \hat{\mathbf{m}}_x^{(i)\mathrm{I}} K Q_1 \hat{\mathbf{m}}_y^{(i)} - \hat{\mathbf{m}}_x^{(i)\mathrm{I}} K \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} \\
&\quad + i \hat{\mathbf{m}}_x^{(i)\mathrm{I}} \mathbf{c}\mathbf{c}^{\mathrm{I}} \hat{\mathbf{m}}_y^{(i)} = 0
\end{aligned} \tag{44}
$$

Therefore, we get linear equations of the 15 unknown elements of Q_1 and the 15 unknown elements of $\mathbf{c}\mathbf{c}^{\mathrm{T}}$. Since these equations cannot provide full rank constraints on the 30 unknowns, there is no linear solutions of Q_1 and $\mathbf{c}\mathbf{c}^{\mathrm{T}}$ directly. However, the constraints are full rank on the elements of Q_1 if $\mathbf{c}\mathbf{c}^{\mathrm{T}}$ is given. That is, if $\mathbf{c}$ can be computed, we can get a linear solution of Q_1. In this way we change the problem to a small scale nonlinear optimization on the 5 elements of $\mathbf{c}$. Once the vector $\mathbf{c}$ is computed, the matrix Q_1 is computed by least squares solutions. A_1 is then calculated from Q_1.

Same to the rank-4 case, we need to align the $\mathbf{x}$-$\mathbf{y}$ plane of the world coordinate system with the two dimensional motion space before we compute A_2 according to Eq. (36). The matrix R is also determined by aligning the matrix $\hat{M}KA_1$ with the matrix $N\hat{M}A_1$. The alignment problem is solved by the least eigenvalue method.

Therefore, the linear transformation A is,

$$
A = [A_1 R \quad K(A_1 R)_{12}] \tag{45}
$$

We apply a derivation similar to the one in Section 3.4 to recover the motion and shape.

5. Scene Reconstruction under Weak Perspective Projection

Based on the unified representation of the static scene and the moving objects in Eq. (1), the image coordinates $(u_{ij} \;\; v_{ij})$ of a point $\mathbf{p}_j$ in frame i under weak perspective projection are,

$$
\begin{aligned}
u_{ij} &= \frac{\mathbf{i}_i \cdot \mathbf{p}_{ij} + t_{xi}}{z_i} \\
v_{ij} &= \frac{\mathbf{j}_i \cdot \mathbf{p}_{ij} + t_{yi}}{z_i}
\end{aligned} \tag{46}
$$

$\mathbf{i}_i$ and $\mathbf{j}_i$ are the rotation axes of the ith camera. t_{xi} and t_{yi} are the translations. z_i is the distance between the ith camera optical center and the center of gravity of all the feature points. Therefore,

$$
\begin{aligned}
u_{ij} &= \frac{\mathbf{i}_i}{z_i} \cdot \mathbf{s}_j + i \frac{\mathbf{i}_i}{z_i} \cdot \mathbf{v}_j + \frac{t_{xi}}{z_i} \\
v_{ij} &= \frac{\mathbf{j}_i}{z_i} \cdot \mathbf{s}_j + i \frac{\mathbf{j}_i}{z_i} \cdot \mathbf{v}_j + \frac{t_{yi}}{z_i}
\end{aligned} \tag{47}
$$

We put all the feature points coordinates $(u_{ij} \;\; v_{ij})$ in a $2n \times m$ measurement matrix W, and get,

$$
W = MS + T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \tag{48}
$$

with the motion matrix,

$$
M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{x2} & \mathbf{n}_{y2} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} \end{bmatrix}^{\mathrm{I}} \tag{49}
$$

where

$$
\begin{aligned}
\mathbf{m}_{xi} &= \frac{\mathbf{i}_i}{z_i} \quad \mathbf{n}_{xi} = i \frac{\mathbf{i}_i}{z_i} \\
\mathbf{m}_{yi} &= \frac{\mathbf{j}_i}{z_i} \quad \mathbf{n}_{yi} = i \frac{\mathbf{j}_i}{z_i}
\end{aligned} \tag{50}
$$

and the shape matrix,

$$
S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix} \tag{51}
$$

The translation vector T is,

$$
T = \begin{bmatrix} \frac{t_{x1}}{z_1} & \frac{t_{y1}}{z_1} & \frac{t_{x2}}{z_2} & \frac{t_{y2}}{z_2} & \cdots & \frac{t_{xn}}{z_n} & \frac{t_{yn}}{z_n} \end{bmatrix}^{\mathrm{I}} \tag{52}
$$

Now the unified representation of the motion matrix M is composed of the rotation axes scaled by the object depth z_i ($\mathbf{m}_{xi}$ and $\mathbf{m}_{yi}$) and their scaled versions by the frame number i ($\mathbf{n}_{xi}$ and $\mathbf{n}_{yi}$). The unified representation of the shape matrix is composed of the scene structure ($\mathbf{s}_j$) and the motion velocities ($\mathbf{v}_j$), which is the same as that under orthographic projection

5.1 *Moving World Coordinate System Location*

As in Section 3.1, we transform the 3D representation to a moving world coordinate system with fixed orientation and the origin at the center of gravity of all the feature points. Therefore,

$$\sum_{j=1}^{m} \mathbf{p}_{ij} = 0 \tag{53}$$

From Eq. (47), we have,

$$\begin{aligned}
\sum_{j=1}^{m} u_{ij} &= \sum_{j=1}^{m} \left(\frac{\mathbf{i}_i}{z_i} \quad \mathbf{p}_{ij} + \frac{t_{xi}}{z_i} \right) \\
&= \frac{\mathbf{i}_i}{z_i} \sum_{j=1}^{m} \mathbf{p}_{ij} + m \frac{t_{xi}}{z_i} = m \frac{t_{xi}}{z_i} \\
\sum_{j=1}^{m} v_{ij} &= \sum_{j=1}^{m} \left(\frac{\mathbf{j}_i}{z_i} \quad \mathbf{p}_{ij} + \frac{t_{yi}}{z_i} \right) \\
&= \frac{\mathbf{j}_i}{z_i} \sum_{j=1}^{m} \mathbf{p}_{ij} + m \frac{t_{yi}}{z_i} = m \frac{t_{yi}}{z_i}
\end{aligned} \tag{54}$$

We get the vector T from Eq. (55),

$$\begin{aligned}
\frac{t_{xi}}{z_i} &= \frac{1}{m} \sum_{j=1}^{m} u_{ij} \\
\frac{t_{yi}}{z_i} &= \frac{1}{m} \sum_{j=1}^{m} v_{ij}
\end{aligned} \tag{55}$$

5.2 *Decomposition*

We subtract the translation vector T from W in Eq. (48),

$$\begin{aligned}
\hat{W} &= W - T[1 \quad 1 \quad \cdots \quad 1] = \hat{M}\hat{S} \\
&= \hat{M}AA^{-1}\hat{S} = MS
\end{aligned} \tag{56}$$

where $M = \hat{M}A$ and $S = A^{-1}\hat{S}$. According to the representations of M and S in Eqs. (49) and (51), we know that the rank of the matrix $\hat{W}$ is at most 6. We perform a SVD on $\hat{W}$ and get the best possible rank 6 approximation of $\hat{W}$ as $\hat{M}\hat{S}$, where $\hat{M}$ is a $2n \times 6$ matrix and $\hat{S}$ is a $6 \times m$ matrix. This decomposition is not unique. Any non-singular 6×6 matrix A could be inserted between $\hat{M}$ and $\hat{S}$ to get another motion and shape pair.

5.3 *Normalization*

Metric constraints are imposed to translate the current pair of motion ($\hat{M}$) and shape ($\hat{S}$) to the Euclidean solutions through recovering the linear transformation A. We recover this 6×6 matrix A by observing that the rows of the motion matrix M consist of the scaled rotation axes and their corresponding scaled versions (Eq. (49)),

$$|\mathbf{m}_{xi}|^2 = |\mathbf{m}_{yi}|^2 \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0 \tag{57}$$

$$|\mathbf{n}_{xi}|^2 = i^2|\mathbf{m}_{xi}|^2 \quad |\mathbf{n}_{yi}|^2 = i^2|\mathbf{m}_{yi}|^2 \quad \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} = 0 \tag{58}$$

$$\mathbf{m}_{xi} \quad \mathbf{n}_{yi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = 0 \tag{59}$$

The above equations impose linear constraints on the elements of $M_1M_1^{\mathrm{T}}$, $M_2M_2^{\mathrm{T}}$ and $M_2M_1^{\mathrm{T}}$.

The derivations to get the linear transformation A are similar as described in Section 3.3. The same steps are also followed to solve for the degenerate cases under weak perspective projection.

5.4 *Motion and Shape Reconstruction*

Once the matrix A has been found, the shape matrix is computed using $S = A^{-1}\hat{S}$ and the motion matrix is $M = \hat{M}A$. We compute the depth z_i first,

$$z_i = \frac{1}{|\mathbf{m}_{xi}|} \tag{60}$$

then the camera rotation axes as,

$$\mathbf{i}_i = z_i\mathbf{m}_{xi} \quad \mathbf{j}_i = z_i\mathbf{m}_{yi} \quad \mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i \tag{61}$$

and the translations are,

$$t_{xi} = \frac{z_i}{m} \sum_{j=1}^{m} u_{ij} \quad t_{yi} = \frac{z_i}{m} \sum_{j=1}^{m} v_{ij} \tag{62}$$

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. The moving objects are automatically detected at the same time. This process is the same as described in Section 3.4.

6. Experiments

In this section a number of experiments are described. First some synthetic images are used to evaluate the quality of the algorithm. Then two experiments are conducted on real image sequences. The first sequence was taken by a hand-held camera of an indoor scene, and the reconstruction results are compared with the ground truth. The second sequence was taken by a small plane flying over the buildings. The weak perspective reconstruction algorithm is used in the experiments described in this section.

6.1. Synthetic Sequences

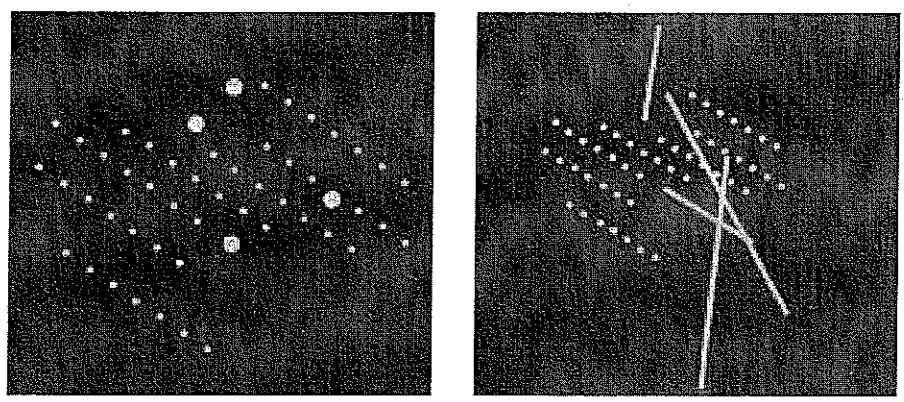
We synthesize sequences of 100 frames with 49 feature points from the static scene and 0 to 9 objects moving in random directions. The shape of the static scene is a sweep of the sin curve in the space. The camera rotates randomly at 30 to 50 degrees around the scene. The distance between the camera and the scene is 15 to 50 times the static scene size. Gaussian noise with a standard deviation of 2 pixels is added to the feature locations (the size of the image is 640×480).

Figure 1 illustrates the case where 4 objects are moving randomly in 3D space. The algorithm automatically detects the number of the moving objects as 4, reconstructs the static scene and the initial positions of the 4 moving objects, as shown in Fig. 1(a). Figure 1(b) shows the trajectories of the moving objects as well as the static scene.

We perform experiments on the case that there are two moving objects whose directions are on a plane. The algorithm detects that the rank as 5 and recovers the scene structure and the two motion trajectories correctly. We also try the case that there are three moving objects but their motion directions lie in a two dimensional space. The algorithm gets the right rank approximation (5) and the accurate reconstructions (shown in Fig. 2).

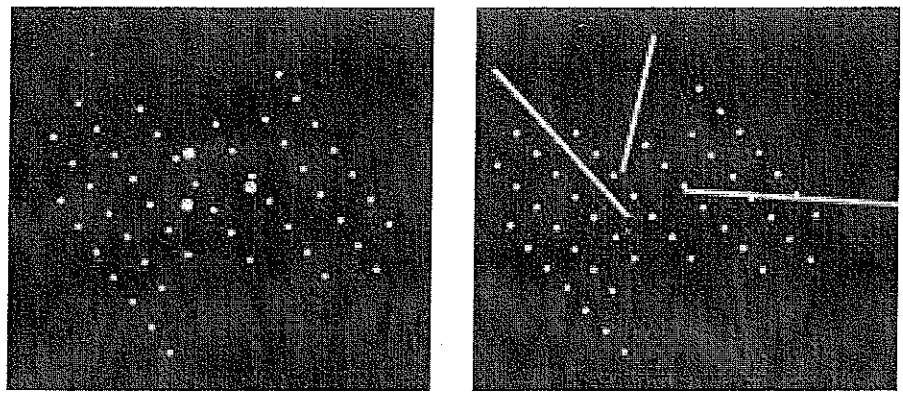
We also conduct experiments on rank-4 cases that there is only one moving object, and that there are multiple moving objects which are moving in the same or the opposite direction. The algorithm detects the rank as 4 in both cases. For the case that there is no moving object, the algorithm correctly detects the rank as 3 and recovers the scene structure.

In all cases, 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. With 2 pixel standard noise, the maximum distance between the recovered static points and their known positions is 1.0%, the maximum error of the reconstructed initial positions of the moving objects is 1.2% and the



(a) (b)

Figure 1 *Full rank case*: A scene with a three dimensional motion space. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories.



(a) (b)

Figure 2. *Rank-5 case*: A scene with three motion trajectories which lie in a two dimensional space. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories.

velocity error is less than 1.1%. We also assess the quality of the camera motion reconstruction. The maximum distance between the recovered camera locations and the ground truth values is 1.4% and the maximum angle between the recovered camera orientations and the known values is 0.1°.

6.2. *Real Sequence 1: Toy Sequence*

This sequence was taken of an indoor scene by a hand-held camera. Three objects, a car, a plane and a toy person, were moving linearly with constant speeds. The car and the person were moving on the floor, and the speed of the car was three times of the speed of the person. Their motion directions were perpendicular with each other. The plane was taking off on a slope and moved two times as fast as the car. The boxes represented the static scene. 24 images were taken. Three of them are shown in Fig. 3(a)–(c). 23 feature points were manually selected and tracked, which are overlaid on the first image shown in Fig. 3(d). We use the first 18 frames to perform the reconstruction. The shapes of the boxes, the starting positions of the moving objects and the motion velocities are recovered and demonstrated in Fig. 4(a) (with texture mapping) and (b) (with wire-frame), the motion trajectories are overlaid in the images. Figure 4(c) show the recovered camera locations and orientations.

We assess the quality of the reconstruction by comparison with the ground truth. The angle between the motion direction of the car and that of the person is 90.15°, the ratio between the speeds is 3.05 which is close to the expected value 3.0. The ratio of the speed of the plane to that of the car is 1.97. The maximum distance between the positions of the recovered static points and the ground truth positions is 2 mm. The recovered motion direction of the plane is 20° tilted upward from the floor, which is close to the expected value.

We project the motion trajectories back to the images and measure the discrepancies of the tracked objects and the back projections in the last 7 frames. The maximum discrepancy is 2 pixels.

6.3. *Real Sequence 2: Campus Sequence*

This sequence was taken by a small airplane flying over a scene with multiple moving cars. The first 80 frames of a 90 frame sequence are used, three of these frames are shown in Fig. 5(a)–(c). 35 feature points were manually selected in the first frame corresponding to the buildings and the two moving cars as shown in Fig. 5(d). These points were automatically tracked in the remaining frames. The algorithm estimates the rank of $\hat{W}$ as 4 because the two cars were moving in opposite directions. Figure 6(a) and (b) show the recovered buildings as well as the motion trajectories. Since the resolution of the input images is very low, the texture mapping is not very clear. Similar to the experiment in Section 6.2, we measure the discrepancies of the back projection cars and the tracked cars for the final 10 frames. The maximum discrepancies are 4 pixels for the white car and 5 pixels for the black car.

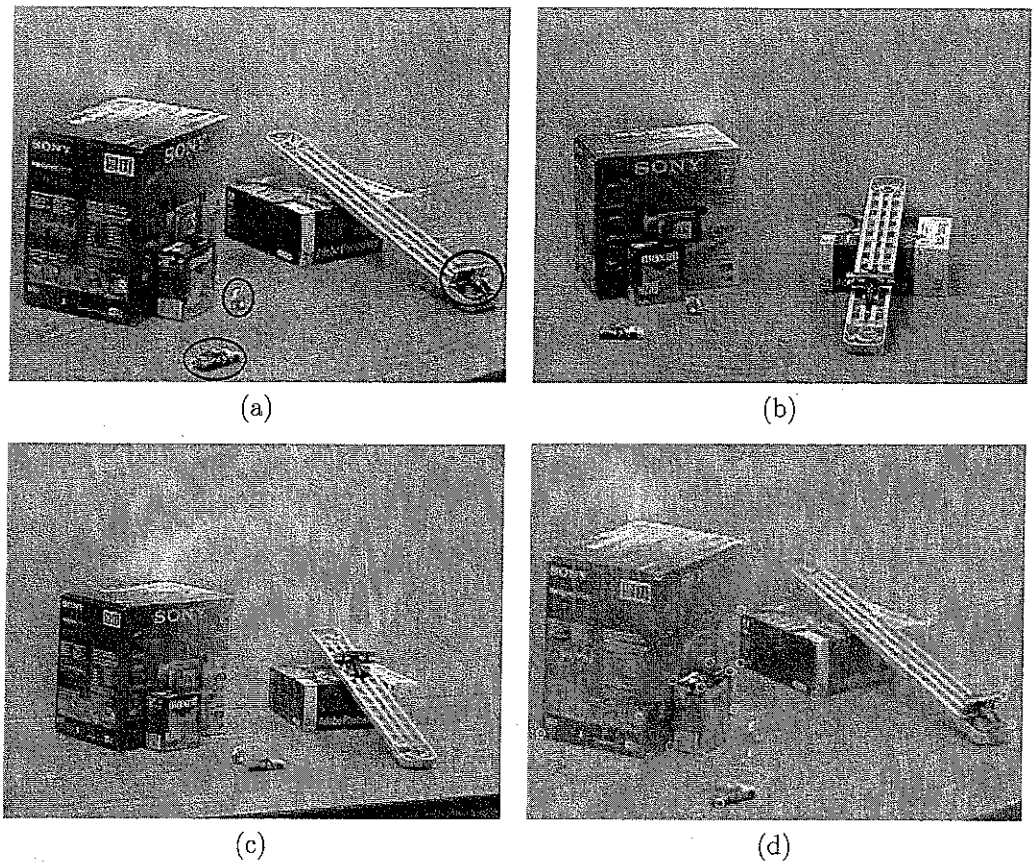


Figure 3. *Toy sequence input*: (a) 1st image. (b) 7th image, (c) 18th image of the toy sequence the moving objects are circled in the 1st image (d) 1st image of the toy sequence with the feature points overlaid

7. Discussion

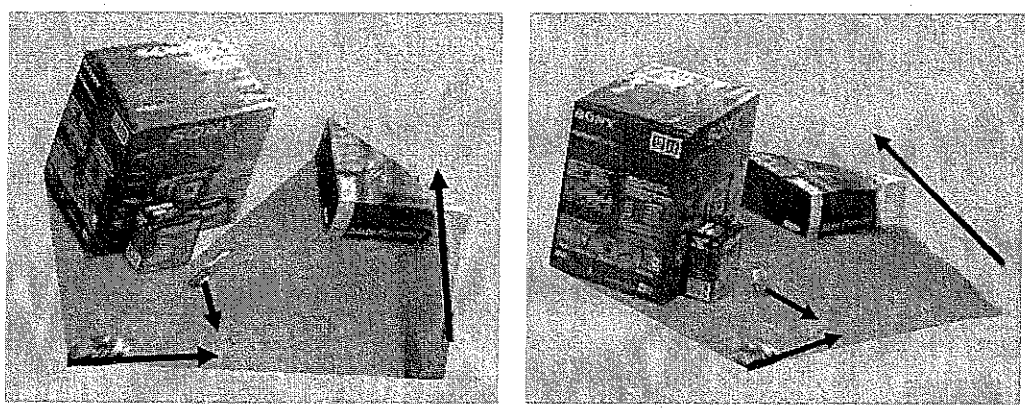
Assuming that the objects are moving linearly with constant speeds, we propose a unified geometrical representation incorporating the static scene and the moving objects. This representation enables the embedding of the motion constraints into the scene structure, that is, the shape matrix is composed of two spaces: one is the scene structure space and another is the motion space. The algorithm makes use of the constraints between the camera motion and the shape matrix to perform the reconstruction. Experiments show that the reconstruction is reliable in the presence of noises. However, analysis is necessary about the sensitivity to noise of the two spaces (the scene space and the motion space) because every point, either static or moving, contributes to the scene space and only the moving points contribute to the motion space.

We are working on extending this work to perspective camera models. Based on the same unified representation of feature points, the image coordinates $(u_{ij} \;\; v_{ij})$ of a point $\mathbf{p}_{ij}$ in frame i under perspective projection are,

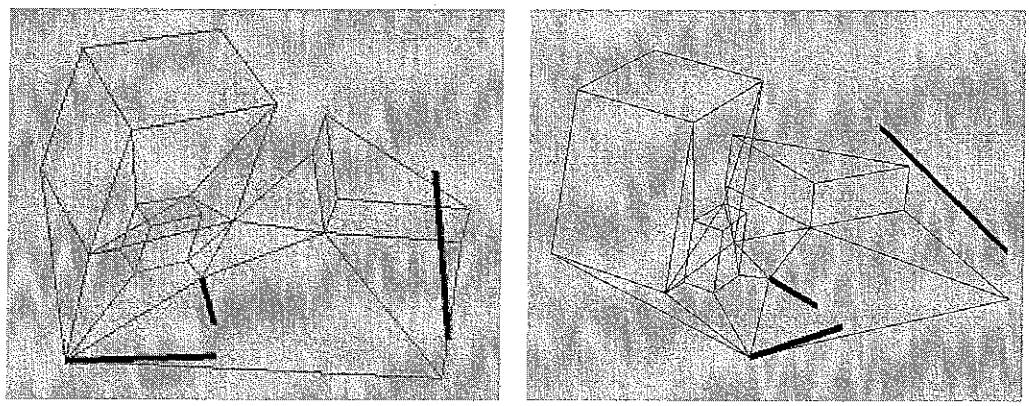
$$\begin{aligned} u_{ij} &= \frac{\mathbf{i}_i \cdot \mathbf{p}_{ij} + t_{xi}}{\mathbf{k}_i \cdot \mathbf{p}_{ij} + t_{zi}} \\ v_{ij} &= \frac{\mathbf{j}_i \cdot \mathbf{p}_{ij} + t_{yi}}{\mathbf{k}_i \cdot \mathbf{p}_{ij} + t_{zi}} \end{aligned} \tag{63}$$

$\mathbf{i}_i$, $\mathbf{j}_i$ and $\mathbf{k}_i$ are the rotation axes of the ith camera. t_{xi}, t_{yi} and t_{zi} are the translations. We divide both the numerator and the denominator of the above equations by t_{zi},

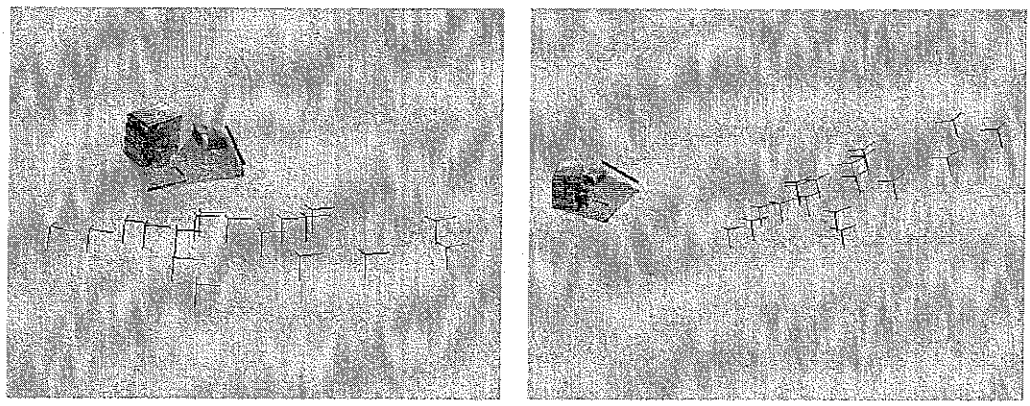
$$\begin{aligned} u_{ij} &= \frac{\frac{\mathbf{i}_i \cdot \mathbf{p}_{ij}}{t_{zi}} + \frac{t_{xi}}{t_{zi}}}{1 + \epsilon_{ij}} \\ v_{ij} &= \frac{\frac{\mathbf{j}_i \cdot \mathbf{p}_{ij}}{t_{zi}} + \frac{t_{yi}}{t_{zi}}}{1 + \epsilon_{ij}} \end{aligned} \tag{64}$$



(a)



(b)



(c)

Figure 4 *Toy sequence results*: (a) Two views of the toy sequence reconstruction with texture mapping. the black lines denote the recovered motion trajectories, the arrows show the motion directions (b) Two views of the reconstruction with wireframe. the black lines denote the recovered motion trajectories (c) Two views of the reconstruction, the 3-axis figures are the recovered cameras

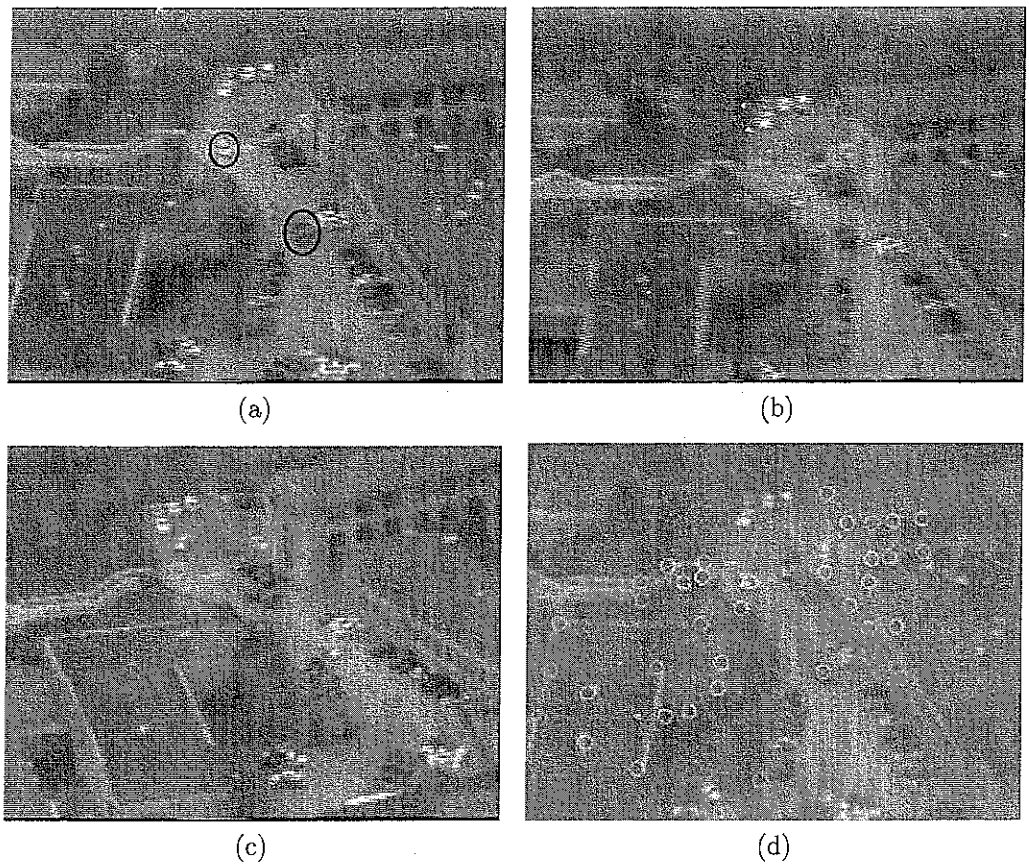


Figure 5. Campus sequence input: (a) 1st image, (b) 33th image, (c) 80th image from the campus sequence, the moving objects are circled in the 1st image (d) 1st image of the campus sequence with the feature points overlaid

where

$$\epsilon_{ij} = \frac{\mathbf{k}_i \cdot \mathbf{p}_{ij}}{t_{zi}} \tag{65}$$

Given tracked feature points, the perspective reconstruction can be regarded as non-linear parameter fitting of Eq. (64) with camera motion and scene structure as parameters. The numerators in Eq. (64) are the weak perspective projections. Christy and Horaud (1996) presented the perspective factorization method by incremental weak perspective reconstructions. Their method worked on the scenes without moving objects. We applied their idea to perspective reconstruction of scenes with moving objects. Whenever the object is at some reasonable distance from the camera, the ϵ_{ij}'s are far less than 1. We compute the parameter fitting by iterations of the weak perspective approximations starting with $\epsilon_{ij} = 0$, that is, we perform the weak perspective reconstruction algorithm described in Section 5 on the measurement matrix W, the recovered motion parameters are denoted as $\mathbf{i}'_i$, $\mathbf{j}'_i$, $\mathbf{k}'_i$ and t'_{xi}, t'_{yi}, t'_{zi}. The recovered feature points are denoted as $\mathbf{p}'_{ij}$. We then use these current parameters to generate a new measurement matrix W':

$$W' = \begin{bmatrix} u'_{11} & u'_{12} & \cdots & u'_{1m} \\ v'_{11} & v'_{12} & \cdots & v'_{1m} \\ \vdots & \vdots & & \vdots \\ u'_{n1} & u'_{n2} & \cdots & u'_{nm} \\ v'_{n1} & v'_{n2} & \cdots & v'_{nm} \end{bmatrix} \tag{66}$$

where

$$\begin{aligned} u'_{ij} &= \frac{\mathbf{i}'_i \cdot \mathbf{p}'_{ij} + t'_{xi}}{\mathbf{k}'_i \cdot \mathbf{p}'_{ij} + t'_{zi}} \\ v'_{ij} &= \frac{\mathbf{j}'_i \cdot \mathbf{p}'_{ij} + t'_{yi}}{\mathbf{k}'_i \cdot \mathbf{p}'_{ij} + t'_{zi}} \end{aligned} \tag{67}$$

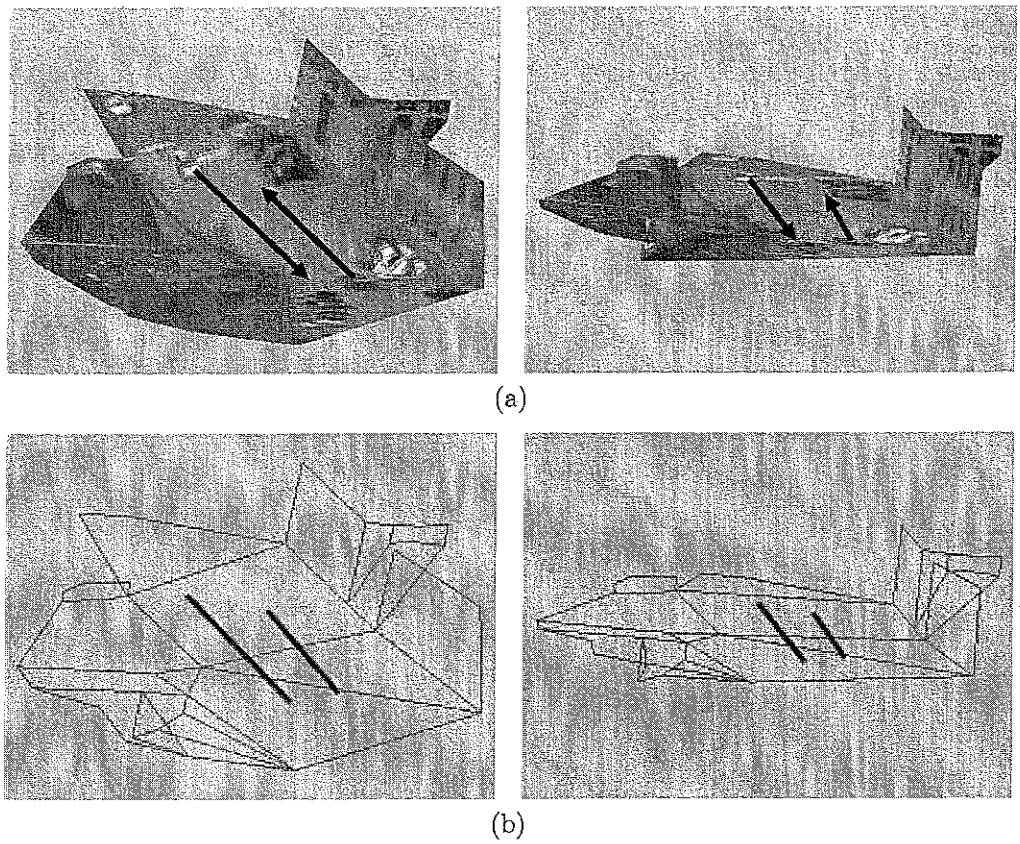


Figure 6. *Campus sequence results*: (a) Two views of the campus sequence reconstruction with texture mapping, the black lines denote the recovered motion trajectories, the arrows show the motion directions (b) Two views of the reconstruction with wireframe, the black lines denote the recovered motion trajectories.

The process of generating the new measurement matrix is equivalent to the back projection process of other non-linear optimization methods. The new measurement matrix W' provides a criterion to choose between the two ambiguous reconstructions which are up to a mirror-symmetry transformation. The difference of W' from the original measurement matrix W also gives the convergence error. A new iteration of the weak perspective reconstruction is performed on the current measurement matrix W'. The goal of the parameter fitting is to iteratively find the reconstructions which make the back projection consistent with the image measurements. The choice from mirror-symmetric shapes, the error measure and the convergence of this scheme are beyond the scope of this paper.

Note

1 In our experiments we use 5% of the point velocity as the threshold

References

Avidan, S. and Shashua, A. 1999. Trajectory triangulation of lines: Reconstruction of a 3d point moving along a line from a monocular image sequence. In *CVPR99*.

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

Boult, T. and Brown, L. G. 1991. Factorization-based segmentation of motions. In *Proceedings of the 1991 Visual Motion Workshop*, pp. 179–186.

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

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

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

Fischler, M. A. and Bolles, R. C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395.

Gear, C. W. 1998. Multibody grouping from motion images. *IJCV*, 29(2):133–150.

Han, M. and Kanade, I. 1998. Homography-based 3d scene analysis of video sequences. In *DARPA98*, pp. 154–160

Han, M. and Kanade, T. 1999. The factorization method with linear motions Technical Report CMU-RI-TR-99-23. Robotics Institute, Carnegie Mellon University

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

Irani, M. 1999. Multi-frame optical flow estimation using subspace constraints In *ICCV99* pp. 626–633

Irani, M., Anandan, P., and Cohen, M. 2002. Direct recovery of planar-parallax from multiple frames *PAMI* 24(11):1528–1534

Irani, M., Anandan, P., and Weinshall, D. 1998. From reference frames to reference planes: Multi-view parallax geometry and applications In *ECCV98*, pp. 829–845.

Irani, M., Rousso B., and Peleg, S. 1992. Detecting and tracking multiple moving objects using temporal integration In *ECCV92*, pp. 282–287

Irani, M., Rousso, B., and Peleg, S. 1997. Recovery of ego-motion using region alignment *PAMI*. 19(3):268–272.

Kumar, R., Anandan P., and Hanna, K. 1994. Direct recovery of shape from multiple views: A parallax based approach. In *ICPR94*. pp. 685–688.

Manning, R.A. and Dyer, C.R. 1999. Interpolating view and scene motion by dynamic view morphing. In *CVPR99*, pp. 388–394.

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

Sawhney, H., Guo, Y., Asmuth, J. and Kumar, R. 1999. Independent motion detection in 3d scenes In *ICCV99*. pp. 612–619

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

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

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

Torr, P.H.S. and Murray, D.W. 1993. Outlier detection and motion segmentation. *SPIE*. 2059:432–443

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

Wolf, L. and Shashua, A. 2001. On projection matrices $p^k \rightarrow p^2$, $k = 3, \ldots, 6$, and their applications in computer vision. In *ICCV01*, pp I:412–419

Zelnik-Manor, L. and Irani, M. 1999. Multi-view subspace constraints on homographies In *ICCV99*. pp. 710–715