# A Multi-body Factorization Method for Motion Analysis

João Costeira Takeo Kanade

**CMU-CS-94-220**

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

email: joao@cs.cmu.edu, tk@cs.cmu.edu

September 30, 1994

**Abstract**

The structure-from-motion problem has been extensively studied in the field of computer vision. Yet, the bulk of the existing work assumes that the scene contains only a single moving object. The more realistic case where an unknown number of objects move in the scene has received little attention, especially for its theoretical treatment. In this paper we present a new method for separating and recovering the motion and shape of multiple independently moving objects in a sequence of images. The method does not require prior knowledge of the number of objects, nor is dependent on any grouping of features into an object at the image level. For this purpose, we introduce a mathematical construct of object shapes, called the shape interaction matrix, which is invariant to both the object motions and the selection of coordinate systems. This invariant structure is computable solely from the observed trajectories of image features without grouping them into individual objects. Once the matrix is computed, it allows for segmenting features into objects by the process of transforming it into a canonical form, as well as recovering the shape and motion of each object.

**Keywords:** Computer Vision, Image Understanding, 3D vision, Shape from Motion, Motin Analysis, Invariants

# 1 Introduction

A motion image sequence allows for the recovery of the three-dimensional structure of a scene. While a large amount of literature exists about this structure-from-motion problem, most previous theoretical work is based on the assumption that only a single motion is included in the image sequence; either the environment is static and the observer moves, or the observer is static and only one object in the scene is moving. More difficult and less studied is the general case of an unknown number of objects moving independently. Suppose that a set of features has been extracted and tracked in an image sequence, but it is not known which feature belongs to which object. Given a set of such feature trajectories, the question is whether we can segment and recover the motion and shape of multiple objects contained in the image sequence.

The previous approaches to the structure-from-motion problem for multiple objects can be grouped into two classes: image motion-based (2D) and three-dimensional (3D) modeling. The image-motion based approach relies mostly on spatio-temporal properties of an image sequence. For example, regions corresponding to different velocity fields are extracted by using Fourier domain analysis [18][1] or scale-space and space-time filters [3, 5, 8, 9]. These image-based methods have limited applicability either because object motions are restricted to a certain type, such as translation only, or because image-level properties, such as locality, need to be used for segmentation without assuring consistent segmentation into 3D objects.

To overcome these limitations, models of motion and scene can be introduced which provide more constraints. Representative constraints include rigidity of an object [17] and smoothness (or similarity) of motion [13, 11, 2, 4]. Then the problem becomes segmenting image events, such as feature trajectories, into objects so that the recovered motion and shape satisfy those constraints. It is now a clustering problem with constraints derived from a physical model. Though sound in theory, the practical difficulty is the cyclic dilemma: to check the constraints it is necessary to segment features and to segment it is necessary to compute constraints. So, developed methods tend to be of a "generate-and-test" nature, or require prior knowledge of the number of objects (clusters). Ullman [17] describes a computational scheme to recursively recover shape from the tracks of image features. A model of the object's shape is matched to the current position of the features, and a new model that maximizes rigidity is computed to update the shape. He suggests that this scheme could be used to segment multi-body scenes by local application of the rigidity principle. Since a single rigid body model does not fit the whole data, collections of points that could be explained by a rigid transformation would be searched and grouped into an object. Under the framework of the factorization method [16], this view of the problem is followed by Boult and Brown [4] and Gear [7], where the role of rigidity is replaced by linear dependence between feature tracks. Since the factorization produces a matrix that is related with shape, segmentation is obtained by recursively clustering columns of feature trajectories into linearly dependent groups.

This paper presents a new method for segmenting and recovering the motion and shape of multiple independently moving objects from a set of feature trajectories tracked in a sequence of images. Developed by using the framework of the factorization by Tomasi and Kanade [16], the method does not require any grouping of features into an object at the image level or prior knowledge of the number of objects. It directly computes shape information and allows segmentation into objects. This has been made possible by introducing a linear-algebraic construct of object shapes, called the shape interaction matrix. The entries of this matrix are invariant to individual object motions and yet is computable only from tracked feature trajectories without knowing their object identities (ie, segmentation). Once the matrix is computed, transforming it into the canonical form results in segmenting features as well as recovering the shape and motion of each object. We will present our theory by using the orthographic camera model. It is, however, easily seen that the theory, and thus the method, works under a broader projection model including weak perspective (scaled orthography) and paraperspective [12] up to an affine camera [10]

In the next section, we will first re-derive the factorization method in homogeneous coordinates including the translational motion component. Then, section 3 will provide geometrical interpretation of the matrices involved in factorization, which will be useful for developing the multi-body factorization method in section 4. Finally in sections 5 and 6 we present experimental results, and discuss implications of our method.

# 2 Factorization Method: A New Formulation Including Translation

The factorization method was originally introduced by Tomasi and Kanade[15, 16] for the case of single object motion. The core of the method is a procedure based on singular value decomposition that separates a matrix of measurements into the product of two matrices which represent the shape and motion of an object, respectively. The method does not need any prior assumptions about either structure or motion.

The original Tomasi-Kanade formulation addressed the case of a moving camera observing a static scene. In this section we will reformulate the method in such a way that a static camera observes a scene with a moving object. Also, whereas the translation component of motion is first eliminated in the Tomasi-Kanade formulation, we will retain that component in our formulation. Though equivalent for the single object case, the new formulation with these two changes simplifies some of the representations and allows for easy extension to the case of multiple moving objects.

## 2.1 World and Observations

Let us assume for the moment that a static camera observes a single moving object. To represent the situation we need two coordinate systems: a moving system **O** attached to the object, and a static system **C** attached to the camera as shown in figure 1.

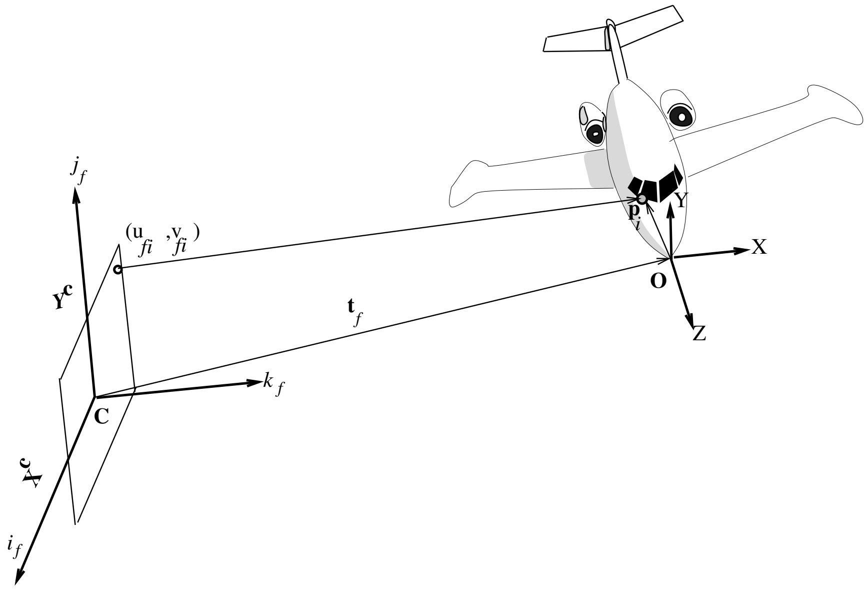



Figure 1: This figure shows the camera and the object with its coordinate system. The (unit) vectors $\mathbf{i}, \mathbf{j}$ define the image plane and $\mathbf{k}$ its normal.

Consider a point $\mathbf{p}_i$ on the object. Its position represented in the camera coordinate system at instant $f$ is given by the transformation,

$$\mathbf{p}_{fi}^{C} = \mathbf{R}_f \, \mathbf{p}_i + \mathbf{t}_f. \tag{1}$$

Here

$$\mathbf{R}_f \equiv \begin{bmatrix} \mathbf{i}_f^T \\ \mathbf{j}_f^T \\ \mathbf{k}_f^T \end{bmatrix} \tag{2}$$

is the rotation matrix whose rows $\mathbf{i}_f^T = \begin{bmatrix} i_{x_f} \, i_{y_f} \, i_{z_f} \end{bmatrix}$, $\mathbf{j}_f^T = \begin{bmatrix} j_{x_f} \, j_{y_f} \, j_z \end{bmatrix}$ and $\mathbf{k}^T = \begin{bmatrix} k_{x_f} \, k_{y_f} \, k_{z_f} \end{bmatrix}$ are the axes of the camera coordinate frame **C** expressed in the object's frame. The vector

$$\mathbf{t}_f \equiv \begin{bmatrix} t_{x_f} \\ t_{y_f} \\ t_{z_f} \end{bmatrix} \tag{3}$$

represents the position of the object's coordinate frame at instant $f$ in the camera frame. The representation (1) can be simplified if we use homogeneous coordinates,

$$\mathbf{s} \equiv \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \equiv \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4}$$

for the object's point. In the homogeneous coordinates, equation (1) can be expressed as

$$\mathbf{s}_{fi}^{C} \equiv \begin{bmatrix} \mathbf{p}_{fi}^{C} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{s}_i \tag{6}$$

The camera is modeled as an orthographic projection. It produces the image by projecting a world point parallel to the optical axis onto the image plane. The image of point $\mathbf{p}_i$ at time $f$ is then given by the first two elements of $\mathbf{p}_{fi}^{C}$:

$$\begin{bmatrix} u_{fi} \\ v_{fi} \end{bmatrix} = \left[ \begin{array}{ccc|c} i_{x_f} & i_{y_f} & i_{z_f} & t_{x_f} \\ j_{x_f} & j_{y_f} & j_{z_f} & t_{y_f} \end{array} \right] \mathbf{s}_i. \tag{7}$$

The object moves relative to the camera which acquires images. In the sequence we track feature points from frame to frame. Suppose that we track $N$ feature points over $F$ frames, and that we collect all these measurements into a single matrix

$$\mathbf{W} \equiv \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ & \vdots & \vdots & \\ u_{F1} & u_{F2} & \dots & u_{FN} \\ v_{11} & v_{12} & \dots & v_{1N} \\ & \vdots & \vdots & \\ v_{F1} & v_{F2} & \dots & v_{FN} \end{bmatrix}. \tag{8}$$

Each row of **W** lists the image coordinates $u$ or $v$ of all the feature points in each frame, and each column represents the image trajectory of one feature over the whole image sequence.

Using (7) we can represent **W** as the matrix product,

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ & \vdots & \vdots & \\ u_{F1} & u_{F2} & \cdots & u_{FN} \\ v_{11} & v_{12} & \cdots & v_{1N} \\ & \vdots & \vdots & \\ v_{F1} & v_{F2} & \cdots & v_{FN} \end{bmatrix} = \left[\begin{array}{ccc|c} i_{x_1} & i_{y_1} & i_{z_1} & t_{x_1} \\ & \vdots & & \vdots \\ i_{x_F} & i_{y_F} & i_{z_F} & t_{x_F} \\ j_{x_1} & j_{y_1} & j_{z_1} & t_{y_1} \\ & \vdots & & \vdots \\ j_{x_F} & j_{y_F} & j_{z_F} & t_{y_F} \end{array}\right] \begin{bmatrix} s_{x_1} & & s_{x_N} \\ s_{y_1} & \cdots & s_{y_N} \\ s_{z_1} & & s_{z_N} \\ 1 & & 1 \end{bmatrix}. \tag{9}$$

If we denote

$$\mathbf{M} \equiv \left[\begin{array}{ccc|c} i_{x_1} & i_{y_1} & i_{z_1} & t_{x_1} \\ & \vdots & & \vdots \\ i_{x_F} & i_{y_F} & i_{z_F} & t_{x_F} \\ j_{x_1} & j_{y_1} & j_{z_1} & t_{y_1} \\ & \vdots & & \vdots \\ j_{x_F} & j_{y_F} & j_{z_F} & t_{y_F} \end{array}\right] \tag{10}$$

$$\mathbf{S} \equiv \begin{bmatrix} s_{x_1} & & s_{x_N} \\ s_{y_1} & \cdots & s_{y_N} \\ s_{z_1} & & s_{z_N} \\ 1 & & 1 \end{bmatrix} \tag{11}$$

as the motion and shape matrices respectively, we have the compact representation

$$\mathbf{W} = \mathbf{MS}. \tag{12}$$

Compared with the original formulation in [16], note that the motion matrix contains translational components $t_{x_f}$ and $t_{y_f}$. In the original formulation they were eliminated from the bilinear equation corresponding to (9) by subtracting the means of the measurements beforehand. That procedure reflected the fact that under orthography the translation components does not provide any information about shape. In the case of a scene with multiple independent moving objects, the situation is not the same: on one hand the translation component of each object cannot be computed without segmentation, and on the other hand the translation component does provide information for segmenting the multi-body scene. For these reasons, we include the translation component in our formulation of the factorization method.

## 2.2 Solution for Shape and Motion by Factorization

We have derived the bilinear relationship (9) by modeling the imaging process. The problem of recovering the shape and motion is to start with a given matrix **W** and obtain a factorization into motion matrix $\mathbf{M}$ and shape matrix $\mathbf{S}$. By simple inspection of (9) we can see that since $\mathbf{M}$ and $\mathbf{S}$ can be at most rank 4, $\mathbf{W}$ will be at most rank 4. In real situations $\mathbf{W}$ is constructed from noisy measurements, so the rank of $\mathbf{W}$ can be higher due to noise in the feature tracking. Among all the possible matrix decompositions, singular value decomposition (SVD) is the most robust and the best rank revealing of all [14] to approximate $\mathbf{W}$ by a rank-4 matrix. With SVD, $\mathbf{W}$ is decomposed and approximated as:

$$\mathbf{W} \;=\; \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{13}$$

Matrix $\boldsymbol{\Sigma} = diag(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ is a diagonal matrix made of the four biggest singular values which reveal the most important components in the data. Matrices $\mathbf{U} \in R^{2F\times 4}$ and $\mathbf{V} \in R^{N\times 4}$ are the left and right singular matrices respectively, such that $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathcal{I}$ (the $4 \times 4$ identity matrix).

By defining,

$$\hat{\mathbf{M}} \;\equiv\; \mathbf{U}\boldsymbol{\Sigma}^{\frac{1}{2}} \tag{14}$$

$$\hat{\mathbf{S}} \;\equiv\; \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{V}^T \tag{15}$$

we have the two matrices whose product can represent the bilinear system $\mathbf{W}$. However, this factorization is not unique, since for any invertible $4 \times 4$ matrix $\mathbf{A}$, $\mathbf{M} = \hat{\mathbf{M}}\mathbf{A}$ and $\mathbf{S} = \mathbf{A}^{-1}\hat{\mathbf{S}}$ are also a possible solution because

$$\mathbf{MS} = \left(\hat{\mathbf{M}}\mathbf{A}\right)\left(\mathbf{A}^{-1}\hat{\mathbf{S}}\right) \;=\; \hat{\mathbf{M}}\hat{\mathbf{S}} = \mathbf{W}. \tag{16}$$

In other words, the singular value decomposition (13) provides a solution both for shape and motion up to an affine transformation.

The exact solution can be computed, using the fact that $\mathbf{M}$ must have certain properties. Let us denote the $4 \times 4$ matrix $\mathbf{A}$ as the concatenation of two blocks,

$$\mathbf{A} \equiv \left[\mathbf{A}_R | \mathbf{a}_t\right], \tag{17}$$

The first block $\mathbf{A}_R$ is the first $4 \times 3$ submatrix related to the rotational component and the second block $\mathbf{a}_t$ is a $4 \times 1$ vector related to translation. Now, since

$$\mathbf{M} \;=\; \hat{\mathbf{M}}\mathbf{A} = \left[\hat{\mathbf{M}}\mathbf{A}_R | \hat{\mathbf{M}}\mathbf{a}_t\right], \tag{18}$$

we can impose motion constraints, one on rotation and the other on translation, in order to solve for $\mathbf{A}$.

### 2.2.1 Rotation Constraints

Block $\mathbf{A}_R$ of $\mathbf{A}$, which is related to rotational motion, is constrained by the orthonormality of axes vectors $\mathbf{i}_f^T$ and $\mathbf{j}_f^T$: each of the $2F$ rows entries of matrix $\hat{\mathbf{M}}A_{\mathbf{R}}$ is a unit norm vector and the first and second set of $F$ rows are pairwise orthogonal. This yields a set of constraints:

$$\hat{\mathbf{m}}_i\mathbf{A}_R\mathbf{A}_R^T\hat{\mathbf{m}}_i^T \;=\; 1 \tag{19}$$

$$\hat{\mathbf{m}}_j\mathbf{A}_R\mathbf{A}_R^T\hat{\mathbf{m}}_j^T \;=\; 1 \tag{20}$$

$$\hat{\mathbf{m}}_i\mathbf{A}_R\mathbf{A}_R^T\hat{\mathbf{m}}_j^T \;=\; 0 \tag{21}$$

where $\hat{\mathbf{m}}_i, \hat{\mathbf{m}}_j$ are rows $i$ and $j$ of matrix $\hat{\mathbf{M}}$ for $i = 1 \ldots F$ and $j = F + 1 \ldots 2F$. This is an overconstrained system which can be solved for the entries of $\mathbf{A}_R \mathbf{A}_R^T$ by using least squares techniques, and subsequently solving for $\mathbf{A}_R$. See [16] for a detailed solution procedure.

### 2.2.2 Translation Constraints

In orthography, the projection of the 3D centroid of an object features into the image plane is the centroid of the feature points. The $X$ and $Y$ position of the centroid of the feature points is the average of each row of $\mathbf{W}$:

$$\begin{aligned} \overline{\mathbf{w}} &\equiv \begin{bmatrix} \frac{1}{N} \sum u_{1,i} \\ \vdots \\ \frac{1}{N} \sum v_{F,i} \end{bmatrix} && (22) \\ &= \mathbf{M}\overline{\mathbf{s}} = \left[ \hat{\mathbf{M}} \mathbf{A}_R | \hat{\mathbf{M}} \mathbf{a}_t \right] \begin{bmatrix} \overline{\mathbf{p}} \\ 1 \end{bmatrix}, && (23) \end{aligned}$$

where

$$\overline{\mathbf{p}} \equiv \frac{1}{N} \sum \mathbf{p}_i \tag{24}$$

is the centroid of the object.

The origin of the object's coordinate system is arbitrary, but we can choose to place it at the centroid of the object, that is $\overline{\mathbf{p}} = 0$. Then it follows immediately from (23) that

$$\overline{\mathbf{w}} = \hat{\mathbf{M}} \mathbf{a}_t \tag{25}$$

This expression is also an overconstrained system of equations, which can be solved for the entries of $\mathbf{a}_t$ in the least square sense. The best estimate will be given by

$$\begin{aligned} \mathbf{a}_t &= (\hat{\mathbf{M}}^T \hat{\mathbf{M}})^{-1} \hat{\mathbf{M}}^T \overline{\mathbf{w}} && (26) \\ &= \mathbf{\Sigma}^{-1/2} \mathbf{U}^T \overline{\mathbf{w}}, && (27) \end{aligned}$$

which completes the computation of all the elements of matrix $\mathbf{A}$.

# 3 Geometrical Interpretation of the Factorization

The factorization procedure developed in the previous section can be summarized as follows. Given the measurements of matrix $\mathbf{W}$, compute its singular decomposition (13)

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{28}$$

This gives recovery of the shape and motion $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ up to an affine transform. Then, by using the constraints (19)-(20) and (27), we obtain $\mathbf{A}$ which provides a unique motion and shape as

$$\begin{aligned} \mathbf{W} &= \mathbf{M}\mathbf{S} && (29)\\ \mathbf{S} &= \mathbf{A}^{-1}\hat{\mathbf{S}} = \mathbf{A}^{-1}\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T && (30)\\ \mathbf{M} &= \hat{\mathbf{M}}\mathbf{A} = \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{A}. && (31) \end{aligned}$$

All the matrix operations involved in the factorization have been so far presented from a pure numerical and algebraic point of view. It is insightful to give a geometric interpretation to these matrices.

Let us first consider the right singular matrix $\mathbf{V}^T$. From equation (30) we see that

$$\mathbf{V}^T = \mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{A}\mathbf{S}. \tag{32}$$

This equation reveals the fact that $\mathbf{V}^T$ is a linear transformation of the shape. This transformation, produced by $\mathbf{A}$ and $\mathbf{\Sigma}$, is done in such a way that the resultant $\mathbf{V}$ is orthonormal. To understand how $\mathbf{A}$ and $\mathbf{\Sigma}$ are related with shape we need to introduce a few geometric concepts first. We have previously used the centroid of the object;

$$\overline{\mathbf{s}} \equiv \frac{1}{N}\begin{bmatrix} \sum X_n \\ \sum Y_n \\ \sum Z_n \\ N \end{bmatrix} = \begin{bmatrix} \overline{\mathbf{p}} \\ 1 \end{bmatrix}. \tag{33}$$

The centroid is the first-order moment of a set of points. The second order moments of a set of points is given in homogeneous coordinates by

$$\begin{aligned} \mathbf{\Lambda} &\equiv \mathbf{S}\mathbf{S}^T && (34)\\ &= \begin{bmatrix} \sum X_n^2 & \sum X_nY_n & \sum X_nZ_n & \sum X_n \\ \sum X_nY_n & \sum Y_n^2 & \sum Y_nZ_n & \sum Y_n \\ \sum X_nZ_n & \sum Y_nZ_n & \sum Z_n^2 & \sum Z_n \\ \sum X_n & \sum Y_n & \sum Z_n & N \end{bmatrix} && (35)\\ &= N\begin{bmatrix} \mathbf{\Lambda_0} & \overline{\mathbf{p}} \\ \overline{\mathbf{p}}^T & 1 \end{bmatrix}. && (36) \end{aligned}$$

The matrix $\mathbf{\Lambda}$ consists of a submatrix $\mathbf{\Lambda_0}$ and the centroid vector $\overline{\mathbf{p}}$. The submatrix $\mathbf{\Lambda_0}$ is the matrix of the moments of inertia of the object. Its eigenvectors represent the directions of the three symmetrical axes of the ellipsoid of inertia. Using equation (30), we can show that the matrix $\mathbf{\Lambda}$ is written as

$$\begin{aligned} \mathbf{\Lambda} &= \left(\mathbf{A}^{-1}\Sigma^{\frac{1}{2}}\mathbf{V}^T\right)\left(\mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{A}^{-T}\right) && (37) \\ &= \mathbf{A}^{-1}\mathbf{\Sigma}\mathbf{A}^{-T}. && (38) \end{aligned}$$

Similarly to the moments of an object, we can think of the moments of motion. Vectors $\mathbf{i}_f$ and $\mathbf{j}_f$ of the motion matrix represent the $X$ and $Y$ axes of the camera (ie., image plane) in object coordinates. As the object moves these vectors describe trajectories in the unit sphere. The second order moments of motion vectors can be defined as

$$\begin{aligned} \mathbf{\Omega} &\equiv \mathbf{M}^T\mathbf{M} && (39) \\ &= \begin{bmatrix} \sum(i_{x_f}^2 + j_{x_f}^2) & \sum(i_{x_f} i_{y_f} + j_{x_f} j_{y_f}) & \sum(i_{x_f} i_{z_f} + j_{x_f} j_{z_f}) & \sum(i_{x_f} t_{x_f} + j_{x_f} t_{y_f}) \\ \sum(i_{x_f} i_{y_f} + j_{x_f} j_{y_f}) & \sum(i_{y_f}^2 + j_{y_f}^2) & \sum(i_{y_f} i_{z_f} + j_{y_f} j_{z_f}) & \sum(i_{y_f} t_{x_f} + j_{y_f} t_{y_f}) \\ \sum(i_{x_f} i_{z_f} + j_{x_f} j_{z_f}) & \sum(i_{y_f} i_{z_f} + j_{y_f} j_{z_f}) & \sum(i_{z_f}^2 + i_{z_f}^2) & \sum(i_{z_f} t_{x_f} + j_{z_f} t_{y_f}) \\ \sum(i_{x_f} t_{x_f} + j_{x_f} t_{y_f}) & \sum(i_{y_f} t_{x_f} + j_{y_f} t_{y_f}) & \sum(i_{z_f} t_{x_f} + j_{z_f} t_{y_f}) & \sum(t_{x_f}^2 + t_{y_f}^2) \end{bmatrix} && (40) \\ &= 2F\begin{bmatrix} \mathbf{\Omega_0} & \hat{\mathbf{t}} \\ \hat{\mathbf{t}}^T & \parallel \mathbf{t} \parallel_{\mathbf{\Sigma}}^2 \end{bmatrix}. && (41) \end{aligned}$$

The submatrix $\mathbf{\Omega_0}$ is the matrix of the "moments of inertia" of the image plane motion axes. The term $\hat{\mathbf{t}}$ is the average position of the camera origin in the object's coordinate system and $\parallel \mathbf{t} \parallel_{\mathbf{\Sigma}}^2$ the average of the norm of the translation vector.

Due to (31) we can derive an equation for $\mathbf{\Omega}$ similar to (38)

$$\mathbf{\Omega} = \mathbf{A}^T\mathbf{\Sigma}\mathbf{A} \quad (42)$$

The bilinearity of the observations is reflected in the second-order motion moments, too. By multiplying the motion moment (42) and shape moment (38), we have

$$\mathbf{\Omega}\mathbf{\Lambda} = \mathbf{A}^T\mathbf{\Sigma}^2\mathbf{A}^{-T}. \quad (43)$$

or

$$\mathbf{\Omega}\mathbf{\Lambda}\,\mathbf{A}^T = \mathbf{A}^T\,\mathbf{\Sigma}^2. \quad (44)$$

This is a standard form of a $4 \times 4$ eigensystem, where $\mathbf{\Sigma}^2$ is the diagonal matrix of the eigenvalues and $\mathbf{A}^T$ the matrix of the eigenvectors. The square of the singular values $\sigma_i^2$ of the measurements matrix $\mathbf{W}$ are the eigenvalues of the product of motion and shape moment matrices, and their eigenvectors form the rows of the transformation matrix $\mathbf{A}$. Geometrically, the eigenvectors represent space orientation, resulting from projecting the symmetry axes of motion into the symmetry axes of shape. The eigenvalues (thus singular values of $W$) represent object's "lengths" multiplied by motion moments.

# 4 The Multi-body Factorization Method

So far we have assumed that the scene contains a single moving object. If there is more than one moving object, the measurement matrix **W** will contain features (columns) which originate from different motions. One may think that solving the problem requires first sorting the columns of the measurements matrix **W** into submatrices, each of which contains features solely from one object, so that the factorization technique of the previous sections can be applied individually. We will show in this section that the multi-body problem can be solved without prior segmentation. For the sake of simplicity in presentation we will present the theory and method for the case of two bodies, but it will be clear that the method is applicable to the general case of an arbitrary unknown number of objects.

## 4.1 Multi-body Motion Recovery Problem: Its Difficulty

Suppose we have a scene in which two objects are moving and we take an image sequence of $F$ frames. The relevant coordinate systems in this case are depicted in figure 2. Suppose also that

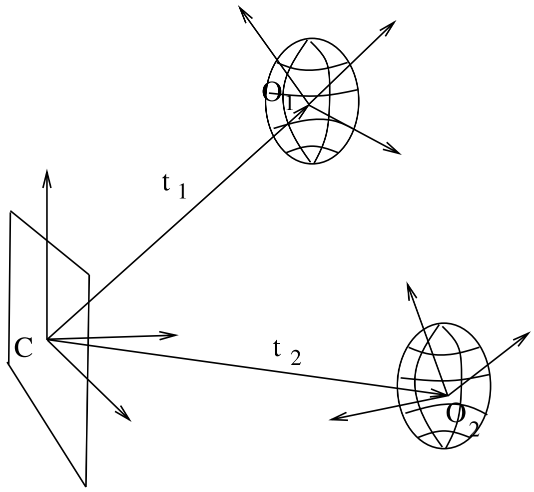



Figure 2: Two bodies: The coordinate systems

the set of features that we have observed and tracked in the image sequence actually consists of $N_1$ feature points from object 1 and $N_2$ from object 2 which are observed in an image sequence of $F$ frames.

Imagine for the moment that somehow we know the classification of features and thus could permute the columns of **W** in such a way that the first $N_1$ columns belong to object 1 followed by the $N_2$ columns from object 2. Matrix **W** would have the canonical form:

$$\mathbf{W}^* \equiv [\, \mathbf{W}_1 \mid \mathbf{W}_2 \,] \,. \tag{45}$$

Each measurement submatrix can be factorized as

$$\mathbf{W}_l = \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{V}_l^T \tag{46}$$

$$= \mathbf{M}_l\mathbf{S}_l = (\hat{\mathbf{M}}_l\mathbf{A}_l)(\mathbf{A}_l^{-1}\hat{\mathbf{S}}_l) \tag{47}$$

with $l = 1$ and 2 for object 1 and 2 respectively. Equation (45) now has the canonical factorization:

$$\mathbf{W}^* = [\mathbf{M}_1|\mathbf{M}_2]\begin{bmatrix}\mathbf{S}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{S}_2\end{bmatrix} \tag{48}$$

$$= [\mathbf{U}_1|\mathbf{U}_2]\begin{bmatrix}\mathbf{\Sigma}_1^{\frac{1}{2}} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Sigma}_2^{\frac{1}{2}}\end{bmatrix}\begin{bmatrix}\mathbf{A}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2\end{bmatrix}\begin{bmatrix}\mathbf{A}_1^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_1^{\frac{-1}{2}} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Sigma}_2^{\frac{-1}{2}}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^T & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_2^T\end{bmatrix} \tag{49}$$

By denoting

$$\mathbf{M}^* \equiv [\mathbf{M}_1|\mathbf{M}_2]\,, \quad \mathbf{S}^* \equiv \begin{bmatrix}\mathbf{S}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{S}_2\end{bmatrix}, \quad \mathbf{A}^* \equiv \begin{bmatrix}\mathbf{A}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2\end{bmatrix} \tag{50}$$

$$\mathbf{U}^* \equiv [\mathbf{U}_1|\mathbf{U}_2]\,, \quad \mathbf{\Sigma}^* \equiv \begin{bmatrix}\mathbf{\Sigma}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{\Sigma}_2\end{bmatrix}, \quad \mathbf{V}^{*T} \equiv \begin{bmatrix}\mathbf{V}_1^T & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_2^T\end{bmatrix} \tag{51}$$

we express a factorization in a similar way to a single object case, where the canonical measurement matrix relates to shape and motion by:

$$\mathbf{W}^* = \mathbf{M}^*\mathbf{S}^* \tag{52}$$

$$\mathbf{S}^* = \mathbf{A}^{*-1}\mathbf{\Sigma}^{*\frac{1}{2}}\mathbf{V}^{*T} \tag{53}$$

$$\mathbf{M}^* = \mathbf{U}^*\mathbf{\Sigma}^{*\frac{1}{2}}\mathbf{A}^* \tag{54}$$

From equation (48), we see that $\mathbf{W}^*$ (and therefore $\mathbf{W}$) will have at most rank 8, since $\mathbf{W}_1$ and $\mathbf{W}_2$ are at most rank 4. Let us consider for the remainder of this paper the non-degenerate case where the rank of $\mathbf{W}$ is in fact equal to 8; that is, the object shape is actually three-dimensional (not planar or line) and the motion vectors span 3D for both objects. The degenerate cases will be briefly touched in the last section and are discussed in more detail in [6].

In reality, we do not know which features belong to which object, and thus the columns of the given measurement matrix $\mathbf{W}$ are a mixture of features from object 1 and 2. We can still apply singular value decomposition (SVD) to the measurement matrix, and obtain

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{55}$$

Then it may appear that the remaining task is to find the linear canonical transformation $\mathbf{A}^*$ in (50) such that shape and motion will have the block structure of equations (53) and (54).

There is, however, a fundamental difficulty in doing this. The metric (rotation and translation) constraints (eq.(19)-(20) and (25)-(27)) were obtained in section 2.2 by considering that the motion matrix for one object, that is, by assuming that the measurement matrix consists of features from a single object. Those constraints are therefore applicable only after knowing the segmentation. This is exactly the mathematical manifestation of the cyclic dilemma mentioned earlier.

Faced with this difficulty, a usual approach would be to group features bit by bit so that we segment $\mathbf{W}$ into two rank-4 matrices and obtain the factorization of the form (48). For example, a most simplistic procedure would be like the following. Pick the first four columns of $\mathbf{W}$ and form a rank-4 subspace. If the fifth column belongs to the subspace (ie. is linear dependent on the first four, or "almost" linear dependent in the case of noisy measurement), then classify it to the same object as the first four columns and update the subspace representation. Otherwise, it belongs to a new object. Apply this procedure recursively to all the remaining columns. This approach is in fact essentially the one used by [4] and [7] to split matrix $\mathbf{W}$, and similar to what was suggested by Ullman [17], whose criteria for merging was local rigidity.

There are a few disadvantages in this cluster-and-test approach. First, there is no guarantee that the first four columns, which always form a rank-4 subspace, are from the same object. Second, if we use a sequential procedure like the one above or its variation, the final result is dependent on where we start the procedure, and alternatively, the search for the globally optimal segmentation most likely will be computational very expensive. Finally, the prior knowledge of the number of objects becomes very critical, since depending on the decision criteria of subspace belongingness the final number of objects may vary arbitrarily.[1]

## 4.2 Mathematical Construct of Shapes Invariant to Motions

The main difficulty in the multi-body structure-from-motion problem revealed above is that shape and motion interact. Mathematically, the equation (48) indicates that the rank-8 measurement space is originally generated by the two subspaces of rank 4 each, represented by the block-diagonal shape matrix $\mathbf{S}^*$. However, the recovered shape space $\mathbf{V}^T$, obtained by the singular value decomposition of the non-canonical $\mathbf{W}$, is in general a linear combination of the two subspaces and has lost the block-diagonal structure.

There is however a mathematical construct that preserves the original subspace structure. Let us define $\mathbf{Q}$ as $(N_1 + N_2) \times (N_1 + N_2)$ square matrix

$$\mathbf{Q} \equiv \mathbf{V}\mathbf{V}^T. \tag{56}$$

We will call this matrix the *shape interaction matrix*. Mathematically, it is the orthogonal operator that projects $N = (N_1 + N_2)$ dimensional vectors to the subspace spanned by the columns of $\mathbf{V}$. This matrix $\mathbf{Q}$ has several interesting and useful properties. First, by definition it is uniquely computable only from the measurements $\mathbf{W}$ without knowing the segmentation, since $\mathbf{V}$ is uniquely obtained by the singular value decomposition of $\mathbf{W}$.

Secondly, permuting columns of $\mathbf{W}$ does not change the set of values $\{Q_{ij}\}$ that appear in $\mathbf{Q}$ though their arrangement in $\mathbf{Q}$ does; swapping columns $l$ and $m$ of $\mathbf{W}$ results in swapping columns $l$ and $m$ of $\mathbf{V}^T$. Therefore it results in simultaneously swapping columns $l$ and $m$ and rows $l$ and $m$ in $\mathbf{Q}$, but not their entry values.

[1] While this is beyond the scope of the assumption in this section, this cluster-and-test approach also requires the prior knowledge of the ranks of objects as well. Since for example a rank-8 measurement matrix might have been generated by two line (rank-2) objects and one full 3D (rank 4) object instead of two full 3D objects, and therefore committing to find two rank-4 subspaces might be wrong.

Thirdly, each element of $\mathbf{Q}$ provides important information about whether a pair of features belong to the same object. Since the set of values do not change, let us compute $\mathbf{Q}^*$, the shape interaction matrix for the canonical measurement matrix $\mathbf{W}^*$. By substituting (53) into (56), we obtain

$$\begin{aligned}
\mathbf{Q}^* &= \mathbf{V}^*\mathbf{V}^{*T} && (57)\\
&= \mathbf{S}^{*T}\mathbf{A}^{*T}\mathbf{\Sigma}^*\,\mathbf{A}^*\,\mathbf{S}^* && (58)\\
&= \mathbf{S}^{*T}(\mathbf{A}^{*-1}\mathbf{\Sigma}^{*-1}\mathbf{A}^{*-T})^{-1}\mathbf{S}^* && (59)\\
&= \mathbf{S}^{*T}\left[(\mathbf{A}^{*-1}\mathbf{\Sigma}^{*-1/2}\mathbf{V}^{*T})(\mathbf{V}^*\mathbf{\Sigma}^{*-1/2}\mathbf{A}^{*-T})\right]^{-1}\mathbf{S}^* = \mathbf{S}^{*T}(\mathbf{S}^*\mathbf{S}^{*T})^{-1}\mathbf{S}^* && (60)\\
&= \begin{bmatrix} \mathbf{S_1}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda_1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda_2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{S_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2} \end{bmatrix} && (61)\\
&= \begin{bmatrix} \mathbf{S_1}^T\mathbf{\Lambda_1}^{-1}\mathbf{S_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2}^T\mathbf{\Lambda_2}^{-1}\mathbf{S_2} \end{bmatrix}. && (62)
\end{aligned}$$

where $\mathbf{\Lambda_1}$ and $\mathbf{\Lambda_2}$ are the $4 \times 4$ matrices of the moments of inertia of each object. This means that the canonical $\mathbf{Q}^*$ matrix for the sorted $\mathbf{W}^*$ has a very defined block-diagonal structure. Moreover, each entry has the value

$$Q^*_{ij} = \begin{cases} \mathbf{s}^T_{1_i}\mathbf{\Lambda_1}^{-1}\mathbf{s}_{1_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 1} \\ \mathbf{s}^T_{2_i}\mathbf{\Lambda_2}^{-1}\mathbf{s}_{2_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 2} \\ 0 & \text{if feature trajectory } i \text{ and } j \text{ belong to different objects.} \end{cases} \quad (63)$$

Finally and most importantly, the set of values $\{Q^*_{ij}\}$, which is the same as $\{Q_{ij}\}$ are invariant to motion. This is true since equations (63) include only $\mathbf{S}$'s, and not $\mathbf{M}$. In other words, in whatever way the objects move they will produce the same set of entries in matrix $\mathbf{Q}$.

In summary, we have shown that without knowing the segmentation of features we can compute matrix $\mathbf{Q}$ whose element $Q_{ij}$ can be interpreted as a measure of the interaction between feature $i$ and $j$: if the value is non zero, they belong to the same object, and if they don't belong to the same object, the value is zero. Also, if the features are sorted correctly into the canonical form of the measurement matrix $\mathbf{W}^*$, then the corresponding canonical shape interaction matrix $\mathbf{Q}^*$ must be block diagonal.

## 4.3 Sorting Matrix Q into Canonical Form

The problem of segmenting and recovering motion of multiple objects now has reduced to sorting the entries of matrix $\mathbf{Q}$ by swapping pairs of rows and columns until it becomes block diagonal. Once achieved, the corresponding permutations of columns of $\mathbf{W}$ will transform it its canonical form where features from one object are grouped into contiguous columns. This relationship between sorting $\mathbf{Q}$ and permuting $\mathbf{W}$ is illustrated in figure 3.

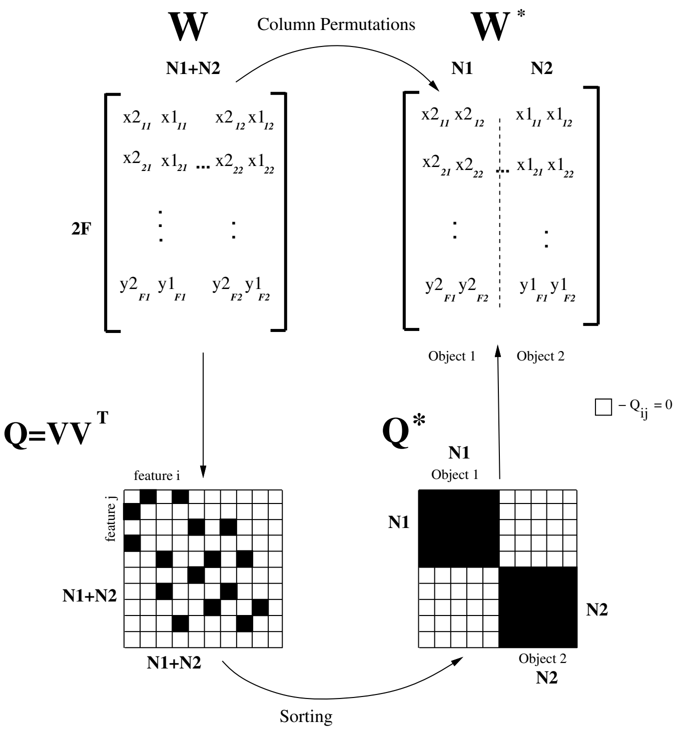



Figure 3: Segmentation process

With noisy measurements, a pair of features from different objects may exhibit a small non-zero entry in $\mathbf{Q}$. We can regard $Q_{ij}^2$ as representing the energy of the shape interaction, and the block diagonalization of $\mathbf{Q}$ can be achieved by minimizing the total energy of all possible off-diagonal blocks over all set of permutations of rows and columns of $\mathbf{Q}$. We found that a simple iterative minimization procedure suffices for our purpose. Alternatively, we can regard matrix $\{Q_{ij}^2\}$ as defining a graph of $N_1 + N_2$ nodes, where the $Q_{ij}^2$ indicates the weight of the link $(i, j)$. We also found that graph-theoretical algorithms, such as the minimum spanning tree, can be used to achieve the block diagonalization more efficiently than the energy minimization. The detailed procedures are presented in [6].

## 4.4 Summary of Algorithm

While we have presented the theory for the case of two full-3D objects, it is easy to see tht its essential part holds for more general cases. First the matrix $\mathbf{Q}^*$ has the block diagonal structure for an arbitrary number of moving objects, that is, an entry $\mathbf{Q}_{ij}$ of the $\mathbf{Q}$ matrix equals to zero if features $i$ and $j$ belong to different objects. Furthermore, this property holds even when the shape matrix of the objects has rank less than 4 (planes and lines). The computation of $\mathbf{Q}$ by (56) requires only the knowledge of the total rank of $\mathbf{W}$, which we can determine by SVD. Finally once $\mathbf{Q}^*$ is obtained, instead of permuting columns of $\mathbf{W}$ we can use the equivalent permutation of $\mathbf{V}^T$, since it is more computationally efficient.

The whole algorithm of the multi-body factorization method is now summarized as:

1. Extract and track features in the input image sequence and create matrix $\mathbf{W}$
2. Compute $r = rank(\mathbf{W})$
3. Decompose matrix $\mathbf{W}$ using SVD
4. Compute shape interaction matrix $\mathbf{Q}$ using the first $r$ rows of $\mathbf{V}^T$
5. Block-diagonalize $\mathbf{Q}$
6. Permute matrix $\mathbf{V}^T$ into submatrices, each corresponding to a single object
7. Compute $\mathbf{A}_i$ for each object, and thus its shape and motion.

## 5 Experiments

We will present two sets of experiments to demonstrate how the algorithm works. The first is an experiment with synthetically generated feature trajectories, and the second with those extracted from real images taken in the laboratory under controlled imaging conditions.

### 5.1 Synthetic Data

Figure 4 shows the 3D synthetic scene. It contains three transparent objects in front of each other moving independently. A static camera takes 100 images during the motion. The closest object to the camera is planar (rank 3) and the other two are full 3D objects (rank 4). So this is in fact a shape-degenerate case. Each object translates slightly and rotates over its own centroid in such a way that the features of all objects are completely intermingled in the image plane. This complication is intentionally introduced in order to demonstrate the fact that our motion segmentation and recovery method does not use any local information in the images. One hundred and eighteen (118) points in total on three objects are chosen: 33 features from the first object, 49 from the second, and 36 from the third. Figure 5 (a) illustrates the actual 3D motions of those 118 points.

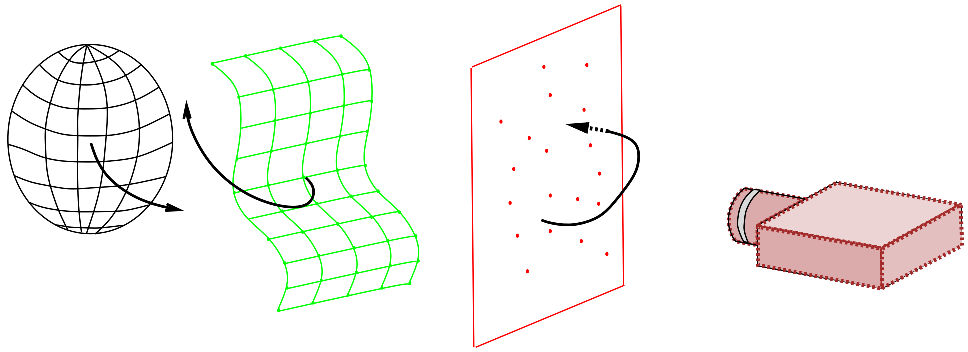
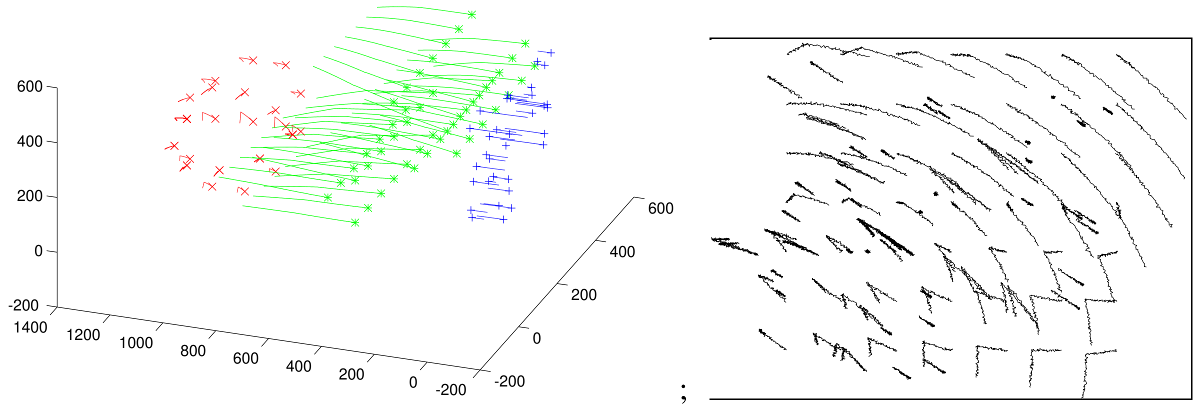


Figure 4: Synthetic scene. Three objects move transparently with arbitrary motion

The projections of 118 scene points onto the image plane during the motion, that is, the simulated trajectories of tracked image features, are shown in figure 5(b) with a different color for each object. Independently distributed Gaussian noise with one pixel of variance was added to the image feature positions for simulating errors in feature tracking. Of course, the identities of the features are assumed unknown, so the measurement matrix created by randomly ordering the features was given to the algorithm.

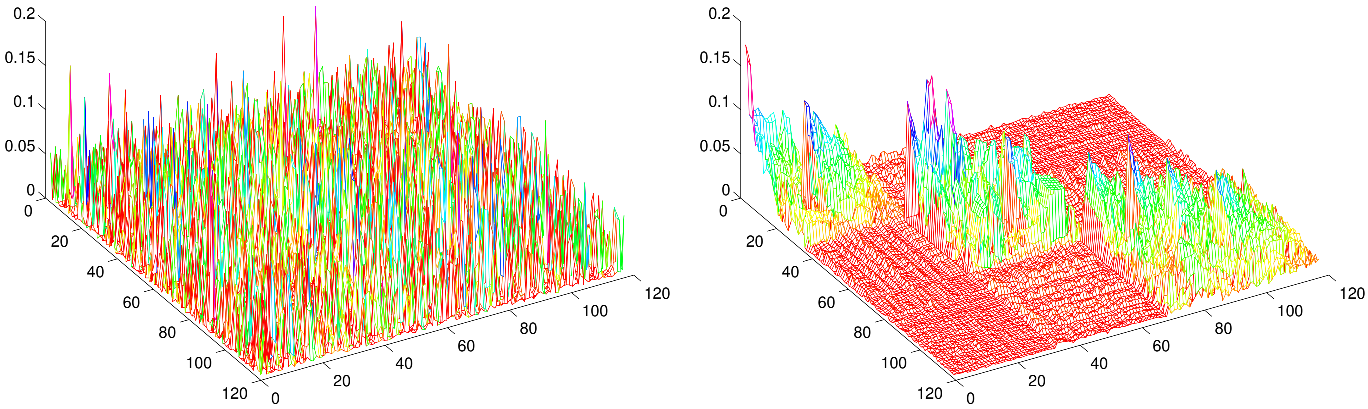
Figure 6(a) shows the shape interaction matrix $\mathbf{Q}$: the height is the square of the entry value. The result of sorting the matrix into a bloackdiagonal form is shown in figure 6(b). We can observe the three blocks corresponding to objects 3, 2 and 1: all of the 118 features are correctly classified.

Figures 7(a) (b) and (c) show one view of each of the recovered shapes of the three objects in the same order as figure 4. Figure 7(c) showing the planar object viewed from its edge indicates the correct recovery of its shape.



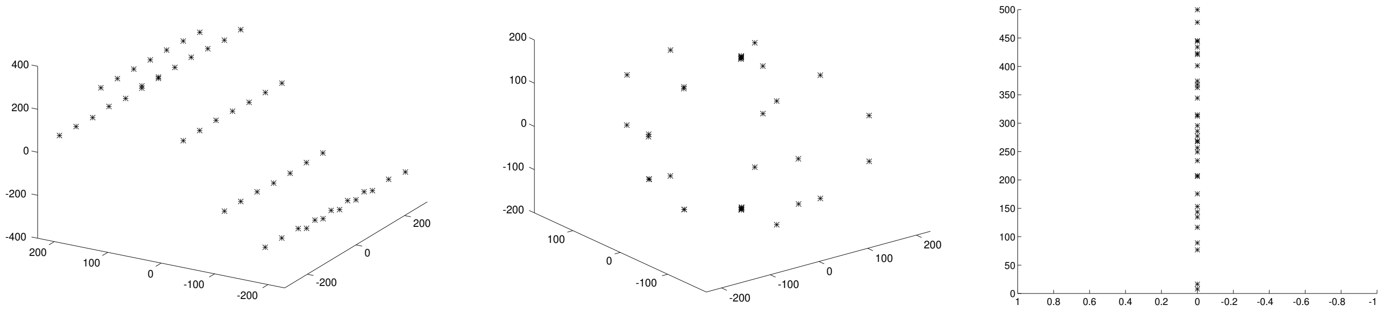

(a) (b)

Figure 5: (a) 3D trajectories of the points and (b) noisy image tracks




(a) (b)

Figure 6: The shape interaction matrix for the synthetic scene with three transparent objects: (a) Unsorted matrix $\mathbf{Q}$, and (b) sorted matrix $\mathbf{Q}^*$.




(a) (b) (c)

Figure 7: Recovered shape of the objects

## 5.2 Laboratory Data

The laboratory scene consists of two roughly cylindrical shapes made by rolling cardboard and drawing dots on the surface. The cylinder on the right tilts and rotates in the plane parallel to the image plane while the cylinder on the left hand side rotates around its axis. The 85 images were taken by a camera equipped with a telephoto lens to approximate orthographic projections, and lighting was controlled to provide the best image quality. In total, 55 features are detected and tracked throughout the sequence: 27 coming the left cylinder and 28 from the other, while the algorithm was not given that information.

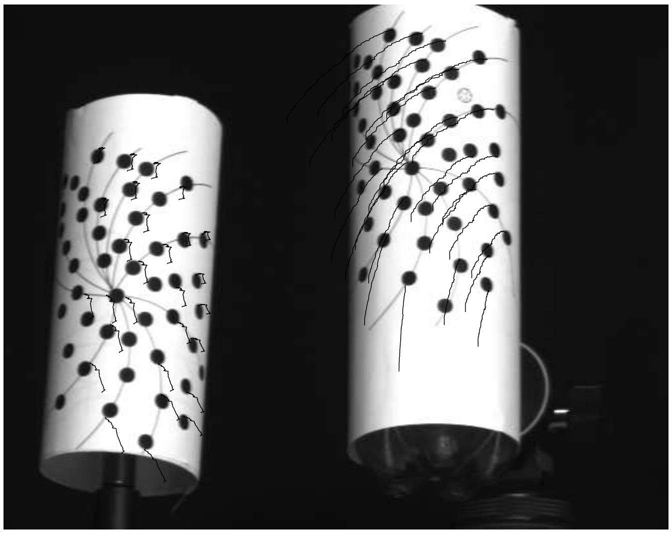
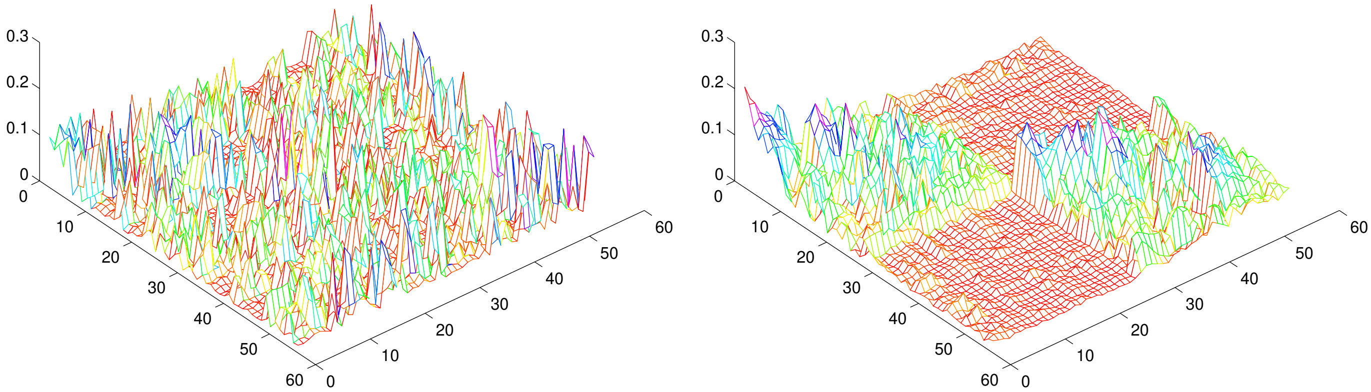


Figure 8: Image of the objects and feature tracks

Figure 8 shows the 85-th frame in the sequence with the tracks of the selected features superimposed. The scene is well approximated by orthography and the tracking was very reliable due to the high quality of the images.




(a) (b)

Figure 9: The shape interaction matrix for the lab scene: (a) Unsorted $\mathbf{Q}$; (b) block-diagonalized $\mathbf{Q}^*$

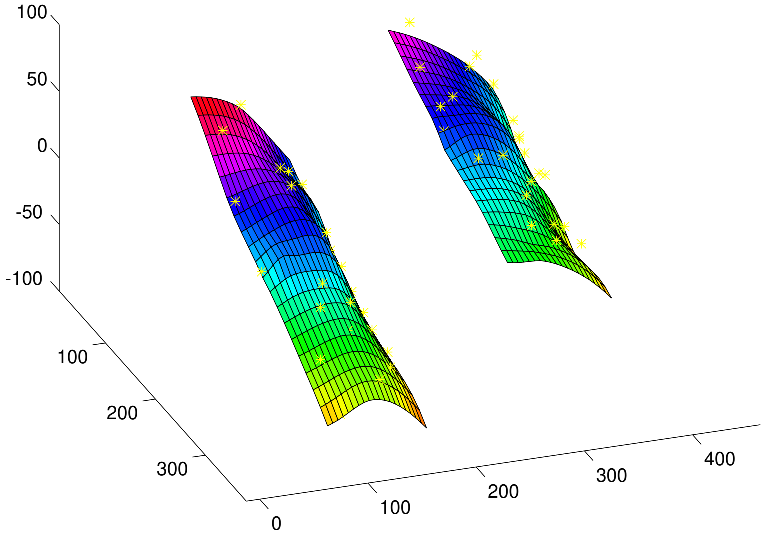



Figure 10: The recovered shape of the two cylinders

Figure 9(a) show the shape interaction matrix $\mathbf{Q}$ for the unsorted input features. The sorted block diagonal matrix $\mathbf{Q}^*$ is shown in figure 9(b), and the features are grouped accordingly for individual shape recovery. The resultant three-dimensional points are displayed in figure 10 with linearly interpolated surface in order to convey a better perception of the their shape.

## 6 Discussion and Conclusion

In this paper we have shown that the problem of multi-body structure-from-motion problem can be solved systematically by using the shape interaction matrix. The striking fact is that the method allows for segmenting or grouping image features into separate objects *based on* their shape properties *without* explicitly computing the individual shapes themselves. Also, no prior knowledge of the number of moving objects in the scene is assumed in the algorithm.

This is due to the interesting and useful invariant properties of the shape-interaction matrix $\mathbf{Q}$. We have shown that $\mathbf{Q}$ is motion invariant. Even when the matrix is computed from a different set of image-level measurements $\mathbf{W}$ generated by a different set of motions of objects, its entries will remain invariant. Each entry has the same unique value independently of the trajectories of its own and other object. The motion invariance property of $\mathbf{Q}$ means also that the degree of complexity of the solution is dependent on the scene complexity, but not on the motion complexity.

The shape interaction matrix $\mathbf{Q}$ is also invariant to the selection of individual object coordinate frames. We can easily see that by considering transforming object $k$'s shape $\mathbf{S}_k$ by a homogeneous transform $4 \times 4$ matrix $\mathbf{T}$,

$$\mathbf{S'}_k = \mathbf{T}\mathbf{S}_k. \tag{64}$$

The corresponding block-diagonal element matrix of $\mathbf{Q}^*$ will be

$$\mathbf{S'}_k^T(\mathbf{S'}_k\mathbf{S'}_k^T)^{-1}\mathbf{S'}_k \;=\; (\mathbf{T}\mathbf{S}_k)^T\left(\mathbf{T}\mathbf{S}_k\mathbf{S}_k^T\mathbf{T}^T\right)^{-1}(\mathbf{T}\mathbf{S}_k) = \mathbf{S}_k^T(\mathbf{S}_k\mathbf{S}_k^T)^{-1}\mathbf{S}_k \tag{65}$$

and therefore the entries of the shape interaction matrix remains the same.

Another interesting fact is that the shape interaction matrix can handle many degenerate cases as well, where one or more objects may not be a full 3-D object, but a line or a planar object. The synthetic example in section 5 was in fact a degenerate case where a planar object was included. More research is required for the degenerate cases including the cases where the motions are degenerate. Also, in order to achieve robustness under the presence of noise we need to relate the noise level with the thresholds necessary in some of the decision making processes. They include the identification of the rank of the measurement matrix in the singular value decomposition, and the determination of block-diagonality in sorting the shape interaction matrix. The report [6] explores some of those issues.

**Acknowledgments:** The authors wish to thank Martial Hebert, José Moura and José Bioucas for useful suggestions and comments to this work.

# References


[1] E. Adelson and J. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, 1985.

[2] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

[3] J. Bergen, P. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proceedings of the IEEE International Conference on Computer Vision*, December 1990.

[4] T. Boult and L. Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*, October 1991.

[5] P. Burt, R. Hingorani, and R. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on visual motion*, pages 187–193, October 1991.

[6] J. Costeira and T. Kanade. A multibody factorization method for motion analysis; Degenerate cases. Technical report, School of Computer Science, Carnegie Mellon University, to be printed.

[7] C. W. Gear. Feature groupin in moving objects. In *Proceedings of the workshop on motion of non-rigid and articulated objects*, Austin, Texas, 1994.

[8] M. Irani, R. Benny, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16, February 1994.

[9] R. S. Jasinschi, A. Rosenfeld, and K. Sumi. Perceptual motion transparency: the role of geometrical information. *Journal of the Optical Society of America*, 9(11):1–15, November 1992.

[10] J. Koendering and A. van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 12(1):5–16, February 1992.

[11] N. Navab and Z. Zhang. From multiple objects motion analysis to behavior-based object recognition. In *Proc. 10th European Conference on Artificial Intelligence, EC AI 92*, 1992.

[12] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In *Proc. Third European Conference on Computer Vision*, pages 97–108. Stockholm, Sweden, May 1994. Part of this was originally published as CMU Technical Report TR-CMU-CS-93-219, December 1993.

[13] D. Sinclair. Motion segmentation and local structure. In *Proceedings of the 4$^{th}$ International Conference on Computer Vision*, 1993.

[14] G. Stewart. On the early history of the singular value decomposition. Technical Report TR-92-31, University of Maryland,Institute for Advanced Computer Studies, March 1992.

[15] C. Tomasi and T. Kanade. Shape and motion without depth. In *International Conference on Computer Vision*, pages 91–95. IEEE Computer Society, 1990.

[16] C. Tomasi and T. Kanade. Shape from motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992. Originally published as CMU Technical Report CMU-CS-90-166, September 1990.

[17] S. Ullman. Maximizing rigidity: The incremental recovery of 3-d structure from rigid and rubbery motion. Technical Report A.I. Memo No. 721, MIT, June 1983.

[18] A. Watson and A. Ahumada. A look at motion in the frequency domain. Technical Report TM-84352, NASA, 1983.