International Journal of Computer Vision 29(3), 159–179 (1998)
© 1998 Kluwer Academic Publishers. Manufactured in The Netherlands.

A Multibody Factorization Method for Independently Moving Objects

JOÃO PAULO COSTEIRA*
Instituto de Sistemas e Robótica, Instituto Superior Técnico, Av. Rovisco Pais, 1096 Lisboa CODEX, Portugal
jpc@isr.ist.utl.pt

TAKEO KANADE†
Carnegie Mellon University, 5000 Forbes Av., Pittsburgh, PA 15213, USA
tk@cs.cmu.edu

Received December 12, 1995; Accepted July 30, 1997

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. The theory works under a broad set of projection models (scaled orthography, paraperspective and affine) but they must be linear, so it excludes projective "cameras".

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 (Adelson, 1985) or scale-space and space-time filters (Bergen, 1990; Irani, 1994;

*Partially funded by project PRAXIS /3/3.1/TPR/23/94.
†Partially funded by NATO Collaborative Research Grant N. 920226.

Jasinschi, 1992). 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 (Ullman, 1983) and smoothness (or similarity) of motion (Sinclair, 1993, Boult, 1991). 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 (1983) 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 multibody 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 (Tomasi, 1990), this view of the problem is followed by Boult and Brown (1991), 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. More recently, Gear (1994) introduced a new method using the reduced row echelon form of the track matrix which finds those linearly dependent groups of tracks by choosing the best set of features that span the subspaces of each object.

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 (1990), the method does not require any grouping of features into an object at the image level or prior knowledge of the number of objects. Furthermore, the method does not rely on any particular set of features from which all other are generated. Insetad it directly computes shape information and allows segmentation into objects 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 (i.e., 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 (Poelman, 1993) up to an affine camera (Koenderink, 1993).

2. Factorization Method: A New Formulation Including Translation

The factorization method was originally introduced by Tomasi and Kanade (1990) for the case of single static object viewd by a moving camera. Here, 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.

2.1. World and Observations

The object moves relative to the camera which acquires images. In the sequence we track feature points from frame to frame. The position of an object point $\mathbf{p}_i^T = [X_i Y_i Z_i]^T$ expressed in homogeneous coordinates in the camera frame, is given by

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

$$= \begin{bmatrix} \mathbf{R}_f & \mathbf{t}_f \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{s}_i \tag{2}$$

where R_f and t_f are, respectively, the rotation and translation components. Suppose that we track N feature points over F frames, and that we collect all these measurements into a single matrix:

$$\begin{bmatrix} u_{11} & \dots & u_{1N} \\ \vdots & & \vdots \\ u_{F1} & \dots & u_{FN} \\ v_{11} & \dots & v_{1N} \\ \vdots & & \vdots \\ v_{F1} & \dots & v_{FN} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{i}_1^T & t_{x_1} \\ \vdots & \vdots \\ \mathbf{i}_F^T & t_{x_F} \\ \mathbf{j}_1^T & t_{y_1} \\ \vdots & \vdots \\ \mathbf{j}_f^T & t_{y_F} \end{array}\right] [\mathbf{s}_1 \ \dots \ \mathbf{s}_N] \tag{3}$$

$$\mathbf{W} = \mathbf{MS}. \tag{4}$$

where (u_{fi}, v_{fi}) are the feature image position, vectors $\mathbf{i}_f^T = [i_{x_f}\, i_{y_f}\, i_{z_f}]$, $\mathbf{j}_f^T = [j_{x_f}\, j_{y_f}\, j_z]^T$, $(f = 1, \dots, F)$ are the first two rows of the rotation matrix at instant f, and $(\mathbf{t}_{x_f}, \mathbf{t}_{y_f})$ are the X and Y coordinates of the position of the object's coordinate frame, in the camera frame, at the same instant.

2.2. *Solution for Shape and Motion by Factorization*

Recovering the shape and motion is equivalent to start with a given matrix $\mathbf{W}$ and obtain a factorization into motion matrix $\mathbf{M}$ and shape matrix $\mathbf{S}$. By simple inspection of (4) we can see that since $\mathbf{M}$ and $\mathbf{S}$ can be at most rank 4, $\mathbf{W}$ will be at most rank 4. Using Singular Value Decomposition (SVD), $\mathbf{W}$ is decomposed and approximated as

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{5}$$

Matrix $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ 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 4×4 identity matrix).

By defining,

$$\hat{\mathbf{M}} \equiv \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}, \quad \hat{\mathbf{S}} \equiv \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T \tag{6}$$

we have the two matrices whose product can represent the bilinear system $\mathbf{W}$. However, this factorization is not unique, since for any invertible 4×4 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

$$\mathbf{MS} = (\hat{\mathbf{M}}\mathbf{A})(\mathbf{A}^{-1}\hat{\mathbf{S}}) = \hat{\mathbf{M}}\hat{\mathbf{S}} = \mathbf{W}. \tag{7}$$

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

$$\mathbf{A} \equiv [\mathbf{A}_R \mid \mathbf{a}_t], \tag{8}$$

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

$$\mathbf{M} = \hat{\mathbf{M}}\mathbf{A} = [\hat{\mathbf{M}}\mathbf{A}_R \mid \hat{\mathbf{M}}\mathbf{a}_t], \tag{9}$$

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 $\hat{\mathbf{M}}\mathbf{A}_R$ is a unit norm vector and the first and second set of F rows are pairwise orthogonal. This yields a set of constraints:

$$\hat{\mathbf{m}}_i\mathbf{A}_R\mathbf{A}_R^T\hat{\mathbf{m}}_i^T = 1 \quad \hat{\mathbf{m}}_j\mathbf{A}_R\mathbf{A}_R^T\hat{\mathbf{m}}_j^T = 1 \tag{10}$$

$$\hat{\mathbf{m}}_i\mathbf{A}_R\mathbf{A}_R^T\hat{\mathbf{m}}_j^T = 0 \tag{11}$$

where $\hat{\mathbf{m}}_i, \hat{\mathbf{m}}_j$ are rows i and j of matrix $\hat{\mathbf{M}}$ for $i = 1, \dots, F$ and $j = F+1, \dots, 2F$. 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 (Tomasi, 1990) 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 $\mathbf{W}$:

$$\bar{\mathbf{w}} \equiv \begin{bmatrix} \frac{1}{N}\sum u_{1,i} \\ \vdots \\ \frac{1}{N}\sum v_{F,i} \end{bmatrix} = \mathbf{M}\bar{\mathbf{s}} \tag{12}$$

$$= [\hat{\mathbf{M}}\mathbf{A}_R \mid \hat{\mathbf{M}}\mathbf{a}_t]\begin{bmatrix} \bar{\mathbf{p}} \\ 1 \end{bmatrix}, \tag{13}$$

where $\bar{\mathbf{p}} \equiv \frac{1}{N}\sum \mathbf{p}_i$ is the centroid of the object. The origin of the object's coordinate system is arbitrary, so we can choose to place it at the centroid of the object, that is $\bar{\mathbf{p}} = 0$. Then it follows immediately from (13) that

$$\bar{\mathbf{w}} = \hat{\mathbf{M}}\mathbf{a}_t \tag{14}$$

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

$$\mathbf{a}_t = (\hat{\mathbf{M}}^T\hat{\mathbf{M}})^{-1}\hat{\mathbf{M}}^T\bar{\mathbf{w}} \tag{15}$$

$$= \mathbf{\Sigma}^{-1/2}\mathbf{U}^T\bar{\mathbf{w}}, \tag{16}$$

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

3. The Multibody Factorization Method

Until now we have assumed that the scene contains a single moving object. If there is more than one moving object, the measurements matrix $\mathbf{W}$ will contain features (columns) which are produced by different motions. One may think that solving the problem requires first sorting the columns of the measurements matrix $\mathbf{W}$ into submatrices, each of which contains features from one object only, so that the factorization technique of the previous sections can be applied individually. In fact this is exactly the approach taken by Boult (1991) and Gear (1994). We will show in this section that the multibody problem can be solved without prior segmentation. For the sake of simplicity we will present the theory and the method for the two body case, but it will be clear that the method is applicable to the general case of an arbitrary unknown number of bodies.

3.1. The Multibody 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. Suppose also that 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. For the moment, imagine that somehow we knew the classification of features and thus could permute the columns of $\mathbf{W}$ in such a way that the first N_1 columns belong to object 1 and the following N_2 columns to object 2. Matrix $\mathbf{W}$ would have the canonical form:

$$\mathbf{W}^* = [\mathbf{W}_1 \mid \mathbf{W}_2]. \tag{17}$$

Each measurements submatrix can be factorized as

$$\mathbf{W}_l = \mathbf{U}_l\mathbf{\Sigma}_l\mathbf{V}_l^T \tag{18}$$

$$= \mathbf{M}_l\mathbf{S}_l = (\hat{\mathbf{M}}_l\mathbf{A}_l)\big(\mathbf{A}_l^{-1}\hat{\mathbf{S}}_l\big) \tag{19}$$

with $l = 1$ and 2 for object 1 and 2, respectively. Equation (17) has now the canonical factorization:

$$\mathbf{W}^* = [\mathbf{M}_1 \mid \mathbf{M}_2]\begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix} \tag{20}$$

$$= [\hat{\mathbf{M}}_1 \mid \hat{\mathbf{M}}_2]\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix}\begin{bmatrix} \mathbf{A}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2^{-1} \end{bmatrix}\begin{bmatrix} \hat{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{S}}_2 \end{bmatrix} \tag{21}$$

By denoting

$$\hat{\mathbf{M}}^* = [\hat{\mathbf{M}}_1 \mid \hat{\mathbf{M}}_2] \tag{22}$$

$$\hat{\mathbf{S}}^* = \begin{bmatrix} \hat{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{S}}_2 \end{bmatrix} \tag{23}$$

$$\mathbf{A}^* = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \tag{24}$$

$$\mathbf{U}^* = [\mathbf{U}_1 \mid \mathbf{U}_2] \tag{25}$$

$$\mathbf{\Sigma}^* = \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 \end{bmatrix} \tag{26}$$

$$\mathbf{V}^{*T} = \begin{bmatrix} \mathbf{V}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2^T \end{bmatrix}, \tag{27}$$

we obtain a factorization similar to a single object case, where the canonical measurements matrix relates to shape and motion according to:

$$\mathbf{W}^* = \mathbf{M}^*\mathbf{S}^* \tag{28}$$

$$\mathbf{S}^* = \mathbf{A}^{*-1}\hat{\mathbf{S}}^* = \mathbf{A}^{*-1}\mathbf{\Sigma}^{*\frac{1}{2}}\mathbf{V}^{*T} \tag{29}$$

$$\mathbf{M}^* = \hat{\mathbf{M}}^*\mathbf{A}^* = \mathbf{U}^*\mathbf{\Sigma}^{*\frac{1}{2}}\mathbf{A}^* \tag{30}$$

From Eq. (20), we see that $\mathbf{W}^*$ (and therefore $\mathbf{W}$) will have at most rank 8; $\mathbf{W}_1$ and $\mathbf{W}_2$ are at most rank 4. For the remainder of this section let us consider non-degenerate cases where the rank of $\mathbf{W}$ is in fact equal to 8; that is, the object shape is actually full three dimensional (excluding planes and lines) and the motion vectors span a four-dimensional space for both objects. Degenerate cases will be discussed later on.

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

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{31}$$

Then it appears that the remaining task is to find the linear transformation $\mathbf{A}$ such that shape and motion will have the block structure of Eqs. (29) and (30).

There is, however, a fundamental difficulty in doing this. The metric (rotation and translation) constraints (Eqs. (10), (11) and (14)–(16)) were obtained in Section 2.2 by considering the motion matrix for one object, that is, by assuming that the measurements matrix consists of features from a single object. These constraints are therefore only applicable when the segmentation is known. This is exactly the mathematical evidence of the cyclic dilemma mentioned earlier.

Faced with this difficulty, the 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 as in Eq. (20). For example, a simplistic procedure would be as follows: pick the first four columns of $\mathbf{W}$ and span a rank-4 subspace. If the fifth column belongs to the subspace (i.e., is linear dependent on the first four, or almost linear dependent in the case of noisy measurements), then classify it as belonging 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 that used by Boult (1991) and Gear (1994) to split matrix $\mathbf{W}$, and similar to that suggested by Ullman (1983), where the criterion for merging was local rigidity.

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

3.2. *A Mathematical Construct of Shapes Invariant to Motions*

In the multibody structure-from-motion problem, the main difficulty, revealed just above, is due to the fact that shape and motion interact. Mathematically, as shown in (20), 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, is in general a linear combination of the two subspaces and does not exhibit a block-diagonal structure.

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

$$\mathbf{Q} = \mathbf{V}\mathbf{V}^T. \tag{32}$$

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 $\mathbf{W}$, since $\mathbf{V}$ is uniquely obtained by the singular value decomposition of $\mathbf{W}$.

Secondly, each element of $\mathbf{Q}$ provides important information about whether a pair of features belong to the same object. Since $\mathbf{W}^*$ is formed applying a set of column permutations to $\mathbf{W}$, $\mathbf{V}^{*T}$ will also result by permuting the same set of columns of $\mathbf{V}^T$.[2] Thus, the canonical $\mathbf{Q}^*$ will result by permuting columns and rows of $\mathbf{Q}$ (the order of each operation is irrelevant) so that both matrices have the same entry values but in different locations. Then, let us compute $\mathbf{Q}^*$ for the canonical form of $\mathbf{W}^*$. By inserting (29) into the canonical version of (32) we can obtain the following derivation:

$$\mathbf{Q}^* = \mathbf{V}^*\mathbf{V}^{*T} \tag{33}$$

$$= \mathbf{S}^{*T}\mathbf{A}^{*T}\mathbf{\Sigma}^{*-}\mathbf{A}^*\ \mathbf{S}^* \tag{34}$$

$$= \mathbf{S}^{*T}(\mathbf{A}^{*-1}\mathbf{\Sigma}^*\mathbf{A}^{*-T})^{-1}\mathbf{S}^* \tag{35}$$

$$= \mathbf{S}^{*T}[(\mathbf{A}^{*-1}\mathbf{\Sigma}^{*-1/2}\mathbf{V}^{*T})(\mathbf{V}^*\mathbf{\Sigma}^{*-1/2}\mathbf{A}^{*-T})]^{-1}\mathbf{S}^*$$

$$= \mathbf{S}^{*T}(\mathbf{S}^*\mathbf{S}^{*T})^{-1}\mathbf{S}^* \tag{36}$$

$$= \begin{bmatrix} \mathbf{S_1}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda_1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda_2}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{S_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2} \end{bmatrix} \tag{37}$$

$$= \begin{bmatrix} \mathbf{S}_1^T \boldsymbol{\Lambda}_1^{-1}\mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2^T \boldsymbol{\Lambda}_2^{-1}\mathbf{S}_2 \end{bmatrix}. \tag{38}$$

where $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ are the 4×4 matrices of the moments of inertia of each object. This means that the canonical $\mathbf{Q}^*$ matrix for the sorted $\mathbf{W}^*$ has a well-defined block-diagonal structure. Also, each entry has the value

$$Q^*_{ij} = \begin{cases} \mathbf{s}_{1_i}^T \boldsymbol{\Lambda}_1^{-1} \mathbf{s}_{1_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 1} \\ \mathbf{s}_{2_i}^T \boldsymbol{\Lambda}_2^{-1} \mathbf{s}_{2_j} & \text{if feature trajectory } i \text{ and } j \text{ belong to object 2} \\ 0 & \text{if feature trajectory } i \text{ and } j \text{ belong to different objects.} \end{cases} \tag{39}$$

Properties of $\mathbf{Q}^*$

Invariant Structure to the Number of Objects. Even though expression (38) was derived for the case of two objects, it is now clear that its structure is the same for any number of objects. In fact, if the scene has M moving objects $\mathbf{Q}^*$ would still have the block diagonal form:

$$\mathbf{Q}^* = \begin{bmatrix} \mathbf{S}_1^T \boldsymbol{\Lambda}_1^{-1}\mathbf{S}_1 & 0 & 0 & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \mathbf{S}_\mathbf{k}^T \boldsymbol{\Lambda}_\mathbf{k}^{-1}\mathbf{S}_\mathbf{k} & 0 & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & 0 & \mathbf{S}_\mathbf{M}^T \boldsymbol{\Lambda}_\mathbf{M}^{-1}\mathbf{S}_\mathbf{M} \end{bmatrix}. \tag{40}$$

If any features i and j belong to different objects their entry Q^*_{ij} will be zero. This property also holds in $\mathbf{Q}$, that is, regardless the way features are sorted in $\mathbf{W}$, the shape interaction matrix contains all the necessary information about the multibody scene.

Invariant to Object Motion. Most importantly, entries Q^*_{ij} are invariant to motion. This is true since Eqs. (39) include only shape variable $\mathbf{S}_k$, and not $\mathbf{M}$. In other words, no matter how the objects move, they will produce the same set of entries in matrix $\mathbf{Q}^*$.

Invariant to Image Scale. Image scaling consists of multiplying the image coordinates at frame f by a constant c_f. For each object, feature coordinates are generated by a scaled version of Eq. (4):

$$\begin{bmatrix} u_{11} & \dots & u_{1N} \\ \vdots & & \vdots \\ u_{F1} & \dots & u_{FN} \\ v_{11} & \dots & v_{1N} \\ \vdots & & \vdots \\ v_{F1} & \dots & v_{FN} \end{bmatrix} = \left[\begin{array}{c|c} c_1\mathbf{i}_1^T & c_1 t_{x_1} \\ \vdots & \vdots \\ c_F\mathbf{i}_F^T & c_F t_{x_F} \\ c_1\mathbf{j}_1^T & c_1 t_{y_1} \\ \vdots & \vdots \\ c_F\mathbf{j}_f^T & c_F t_{y_F} \end{array}\right] [\mathbf{s}_1 \ \dots \ \mathbf{s}_N]$$

$$\tilde{\mathbf{W}} = \mathbf{CMS},$$

$$\mathbf{C}_{2F\times 2F} = \text{diag}(c_1, \dots, c_F, c_1, \dots, c_F),$$

or, in the multibody case

$$\tilde{\mathbf{W}} = [\mathbf{C}_1\mathbf{M}_1 \mid \mathbf{C}_2\mathbf{M}_2] \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix}. \tag{41}$$

From Eqs. (33–38) we can see that both $\mathbf{V}$ and $\mathbf{A}$ change but $\mathbf{Q}$ will be the same. The rows of $\tilde{W}$ are still linear combinations of the rows of $\mathbf{S}$. Then, the subspace spanned by the rows of $\tilde{W}$ is the same of the nonscaled $\mathbf{W}$, consequently the orhogonal projector is the same. This property is, in fact, a corollary from the motion invariance property of $\mathbf{Q}$. The shape interaction matrix is invariant to any pre- or post-multiplication of the motion matrices, since its only dependence is on the shape matrix. However, we higlight this property because it confers invariance to perspective effects that can be modeled by image scaling.

Invariant to Coordinate System. The shape interaction matrix is invariant to the coordinate system in which we represent the shape. Suppose we transform the shape, $\mathbf{S}$, of object k by the general transformation $\mathbf{T} \in R^{4\times 4}$:

$$\mathbf{S}' = \mathbf{TS}. \tag{42}$$

The corresponding block-diagonal element matrix will be

$$\begin{aligned} \mathbf{S}'^T(\mathbf{S}'\mathbf{S}'^T)^{-1}\mathbf{S}' &= (\mathbf{TS})^T \, [(\mathbf{TS})(\mathbf{TS})^T]^{-1} \, (\mathbf{TS}) \\ &= \mathbf{S}^T(\mathbf{SS}^T)^{-1}\mathbf{S} \end{aligned} \tag{43}$$

which remains the same.

Invariant to Shape Rank. Finally, the shape interaction matrix is also invariant to the type of object. The rank of the shape matrix **S** can be 2 for a line, 3 for a plane and 4 for a full 3D object. However, the entries of **Q*** will have the same general expression. For degenerate shapes (lines and planes), the difference will be the number of rows and columns of matrices **S** and **Λ**. Since **Q** is invariant to the coordinate system, if object k is a line, $\mathbf{S}_k$ can be represented as a $2 \times N_k$ matrix ($3 \times N_k$ for a plane); therefore, $\mathbf{\Lambda}_k$ will be 2×2 (3×3 for a plane). In both cases, the total rank of **Q*** changes but not its structure nor its entries.

3.3. Sorting Matrix Q into Canonical Form

In the previous section we have shown that we can compute matrix **Q** without knowing the segmentation of the features. Each element Q_{ij} can be interpreted as a measure of the interaction between features i and j: if its value is nonzero, then the features belong to the same object, otherwise they belong to different objects if the value is zero. Also, if the features are sorted correctly into the canonical form of the measurement matrix **W***, then the corresponding canonical shape interaction matrix **Q*** must be block diagonal.

Now, the problem of segmenting and recovering motion of multiple objects has been reduced to that of sorting the entries of matrix **Q**, by swapping pairs of rows and columns until it becomes block diagonal. Once this is achieved, applying the corresponding permutations to the columns of **W** will transform it to the canonical form where features from one object are grouped into adjacent columns. This equivalence between sorting **Q** and permuting **W** is illustrated in Fig. 1.

With noisy measurements, a pair of features from different objects may exhibit a small nonzero entry. We can regard Q_{ij}^2 as representing the energy of the shape interaction between features i and j. Then, the block diagonalization of **Q** can be achieved by minimizing the total energy of all possible off-diagonal blocks over all sets of permutations of rows and columns of **Q**. This is a computationally overwhelming task since the number of possibilities is factorial with the number of features.

Alternatively, since matrix $\{Q_{ij}^2\}$ is symmetric and all elements are positive, it defines the incidence matrix of a graph of $N_1 + N_2$ nodes, where the Q_{ij}^2 indicates the weight of the link (i, j). Several graph-theoretical algorithms (Thomas, 1986), such as the minimum spanning tree (MST), can be used to achieve block diagonalization much more efficiently than energy minimization.

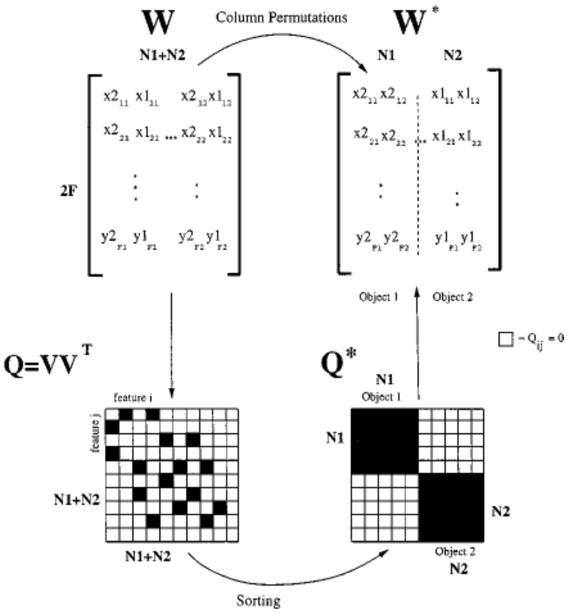

Figure 1. Segmentation process.

The importance of these methods lie in the interesting interpretation of the shape interaction matrix (or the square of its elements). In noise-free environments **Q** is, in fact, a forest: a graph made of several nonconnected subgraphs, and segmentation reduces to looking for the connected components. In the presence of noise **Q** is interpreted as a single fully connected graph from which the noisy links have to be removed. We can use the MST to achieve a minimum representation of **Q** where the noisy links can be easily removed. However, a single spike of noise can be understood by the sorting algorithm as a link, jeopardizing the entire segmentation. Because of this, and also because of the difficulty of coding prior knowledge in the MST algorithm we have devised another algorithm that explores the global constraints on **Q**, allowing a much more efficient and robust sorting.

4. Segmentation Algorithm

The algorithm we propose here segments a multibody scene in two steps: In the first step rows and columns of **Q** are iteratively permuted in such a way that features of the same object are arranged adjacently into blocks, transforming **Q** into the canonical shape interaction matrix **Q***. Though sorted, **Q*** alone does not

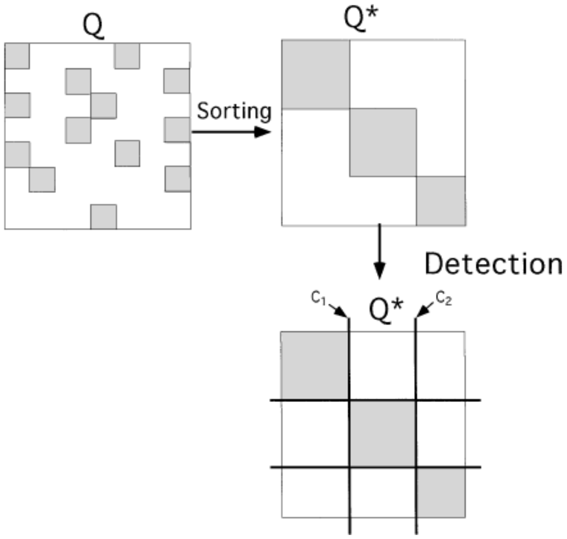

Figure 2. The segmentation algorithm: Sorting matrix **Q** and detecting the blocks.

provide information about the size and location of each block; therefore, a second step is required to compute the rows/columns that limit the blocks corresponding to features of a single object.

Consider an example where three objects move independently. Figure 2 depicts the two steps of the algorithm graphically. First, the sorting process transforms the original **Q** into $\mathbf{Q}^*$. Then the detection process determines columns c_1 and c_2 which isolate the blocks corresponding to each object. This detection process is quite relevant mainly for two reasons: on one hand, the off-diagonal elements are nonzero and we have no prior knowledge about either the signal or the noise models, hence we are unable to detect the limiting columns based on local information alone. In other words, we cannot compute optimal thresholds to classify the elements of **Q** as either noise or signal (VanTrees). Also, the detection process must take into consideration shape degeneracy issues, that is, cases where objects have less than three independent dimensions (lines and planes). Fortunately, using the properties of **Q**, the block diagonal structure is invariant with shape rank; therefore, we have been able to develop a detection algorithm that robustly handles any shape degeneracy.

4.1. Sorting

As already stated, sorting **Q** is equivalent to minimizing the energy of the off-diagonal blocks, over the set of all permutations of rows and columns. A straightforward inspection shows that this type of optimization leads to a combinatorial explosion. Instead, we can considerably reduce the search effort by using suboptimal strategies without jeopardizing performance.

Our algorithm uses a gready or also known as hill-climbing search strategy. By hill-climbing we mean a search procedure that, at each search level (or iteration), chooses the best path without taking into account past decisions.

At each iteration, say k, the current state is represented by a $k \times k$ submatrix $\mathbf{Q}^{*k}$ which contains the features sorted so far. A set of operations expands the current state, producing candidates (features) to be included in the current sorted $\mathbf{Q}^{*k}$.

Figure 3 shows iteration k where feature $k+1$ is to be selected from among the $N-k$ candidates. The candidates are features $k+1$ to N whose columns and rows are not included in the current segmentation. The cost, C_j^k of each candidate is given by the energy of the first k elements

$$C_j^k = \sum_{i=1}^{k} Q_{i,j}^2 \quad \text{for } (j = k+1, \dots, N), \tag{44}$$

which represents the total energy of interaction between each of the candidate features and the set of already sorted features.[3] By maximizing the cost function C_j^k, our search strategy selects the feature whose global energy of interaction with the current segmentation, is the largest.

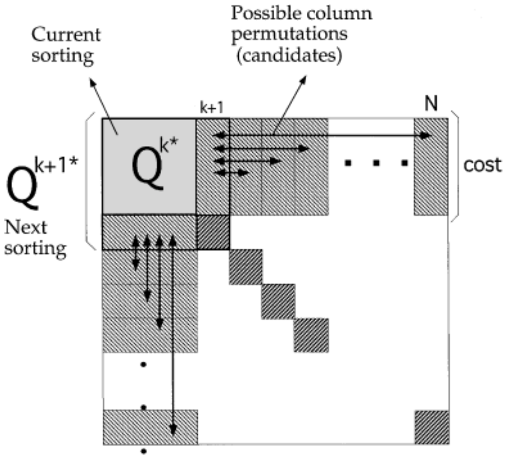

Figure 3. Sorting algorithm: At iteration k, columns $k+1$ to N are permuted and the column with the highest norm is selected to form $\mathbf{Q}^{*k+1}$.

The updated state Q^{*k+1} is obtained by augmenting Q^{*k} with the column and the row of the best feature. The column corresponding to this feature is first permuted with column $k+1$, followed by a permutation of rows with the same indices. Matrix $(\mathbf{Q}^{*k+1})$ is then formed with the first $(k+1) \times (k+1)$ elements of the permuted shape interaction matrix. As a result of this maximization strategy, submatrix Q^{*k+1} has maximal energy among all possible $(k+1) \times (k+1)$ submatrices of $\mathbf{Q}$. Unless the noise energy is similar to that of the signal, for all features in a set, this sorting procedure groups features by the strength of their coupling. Even though this procedure may look like a blind search, in the next section we will show that this maximization relates the energy maximization to rank properties of operators $\mathbf{Q}^{*k}$, thus taking into account the structure of the problem.

4.2. Block Detection

Having sorted matrix $\mathbf{Q}$ into canonical form, the matrix $\mathbf{Q}^*$ for an arbitrary number of objects M has the block form:

$$\mathbf{Q}^* = \begin{bmatrix} \mathbf{S_1^T \Lambda_1^{-1} S_1} & 0 & 0 & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \mathbf{S_K^T \Lambda_K^{-1} S_K} & 0 & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & 0 & \mathbf{S_M^T \Lambda_M^{-1} S_M} \end{bmatrix} \tag{45}$$

$$= \begin{bmatrix} \mathbf{Q}_1 & 0 & 0 & 0 & 0 \\ & \ddots & & & \\ 0 & 0 & \mathbf{Q}_K & 0 & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & 0 & \mathbf{Q}_M \end{bmatrix}. \tag{46}$$

Since noise induces a small nonzero value in the off-diagonal elements, instead of detecting zeros we must detect the transition between blocks (signal) and the off-diagonal elements (noise). Even assuming correct sorting, this transition is quite hard to detect, based on local values alone, due to the lack of knowledge about noise characteristics. In other words, it is not possible to set up a threshold below which an entry could be considered zero. The threshold is determined by an optimality criterion involving the noise probability distribution function (VanTrees).

However, there are global constraints that can be applied to $\mathbf{Q}^*$. First the rank of each block is constrained to be 2 (a line), 3 (a plane) or 4 (a full 3D object). Second, we can relate the rank of a block to its energy: in fact, Eq. (45) shows that the rank of each $\mathbf{Q}_K$ is the same as the rank of the shape matrix of object K. Also, the square of the Frobenius norm (F-norm) of matrix $\mathbf{Q}_K$ relates to the block energy and to its singular values σ_{K_i} according to

$$\|\mathbf{Q}_K\|_F^2 = \sum_{i=1}^{N_K} \sum_{j=1}^{N_K} Q_{K_{ij}}^2 = \sigma_{K_1}^2 + \cdots + \sigma_{K_R}^2, \tag{47}$$

where N_K is the number of features of each block and R its rank. The number of nonzero singular values is $R = 2, 3, 4$ depending whether the object is a line, a plane or a 3D object respectively. Since $\mathbf{Q}_K$ is orthonormal, all singular values, σ_{K_i}, are equal to 1 and hence for each type of object, the sum (47) adds to 2 (line), 3 (plane) or 4 (3D object). Then, we can relate the energy of each block with its rank by

$$\|\mathbf{Q}_K\|_F^2 = \sum_{i=1}^{N_K} \sum_{j=1}^{N_K} Q_{ij}^2 \tag{48}$$

$$= \sigma_{K_1}^2 + \cdots + \sigma_{K_R}^2 = \operatorname{rank}(\mathbf{Q}_K) \tag{49}$$

Instead of considering an individual block, let us compute the sum (47) for the first m columns/rows of $\mathbf{Q}^*$, defined by the function $\varepsilon(\cdot)$:

$$\varepsilon(m) = \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbf{Q}_{ij}^{*2}, \tag{50}$$

for $m = 1, \ldots, N$. Then, columns for which the integer part of ε increases one unit are potential block limiting columns, provided the block rank constraints are satisfied. Consider Fig. 4 which illustrates one possible shape of the function ε for the case of two objects with rank-4 shape.

The vertical dashed lines indicate rank jumps, that is, columns where ε is a whole number or its integer part increases by one (under noisy conditions, except for $m = N$, the function ε may never be a whole number). Given the indices of the columns of integer crossing by ε, segmentation consists in finding the blocks that match these rank jumps and satisfy the constraints. The solution is not unique, as our examples illustrate in

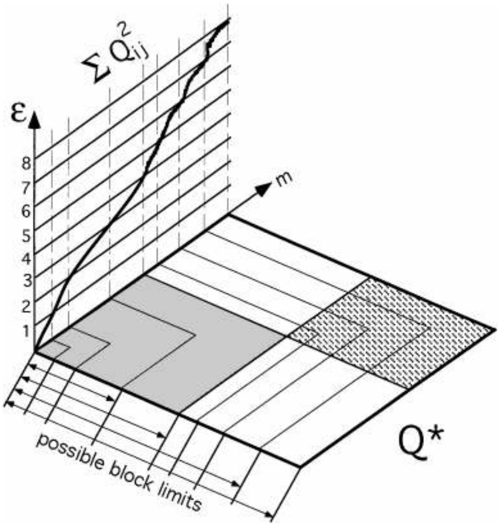

Figure 4. Evolution of the norm of **Q***.

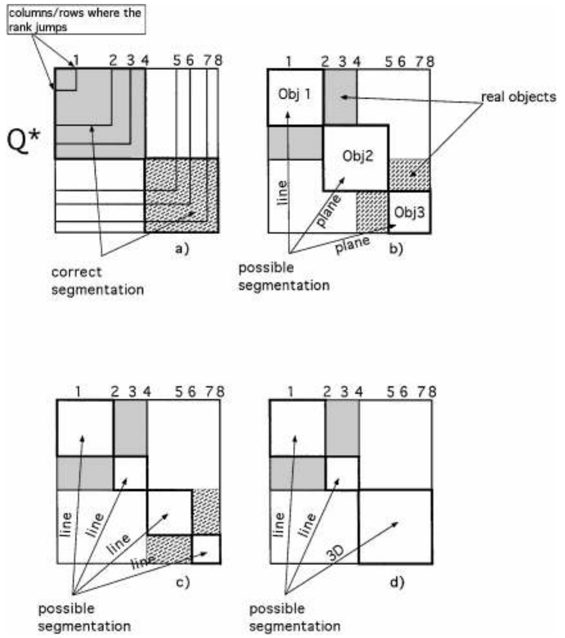

Figure 5. Possible $\mathbf{Q}_K^*$'s for a rank 8 **Q***: (a) One line and two planes, (b) four lines, (c) two 3D objects, and (d) one full 3D object and two lines.

Fig. 5. For a rank 8 matrix **Q***, we have eight possible segmentations, obtained by considering all possible combinations of rank 2, 3 or 4 blocks whose sum is 8.

In Fig. 5 we show only four possible solutions for this example. The numbers in Fig. 5(a) represent the columns and rows that limit submatrices of **Q*** for which the rank jumps by one. In the same figure, the shaded rectangles represent the correct feature segmentation. The superimposed white blocks in Figs. 5(b), (c) and (d) show other possible segmentations that also match the rank jumps and satisfy the block rank constraint. Among the eight possible configurations, Fig. 5 considers the following scene segmentations for the rank-8 **Q***:

5(a) Two rank-4 objects: assuming the scene is formed by two, rank 4, objects there is only one block configuration. The first four rank jumps form object one and the remaining four the second object. This is also the correct solution.

5(b) One object with rank 2 and two objects with rank 3. In other words, the scene is made of one moving line and two planes. These objects also form a shape interaction matrix with rank 8. In the figure we show one possible configuration, where the first block has rank 2 (a line) and the other two blocks have rank 3 (planes).

5(c) Four lines. Each of the blocks has rank 2 and represents one line. In a rank-8 **Q** we can have four of these blocks.

5(d) One 3D object and two lines. In this configuration the first block has rank 3, the second has rank 2 and the third rank 4. With these three blocks two more combinations are possible.

Considering all possible combinations, the correct solution is easily determined through the energy maximization of the blocks. Since the total energy of **Q*** is equal to the constant

$$\varepsilon(N) = \|\mathbf{Q}^*\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}^* = \text{rank}(\mathbf{Q}), \quad (51)$$

we can divide it into the energy of the blocks and the energy of the off-diagonal. The best solution is then the one which concentrates most energy in the blocks. In summary, the detection process is a constrained optimization process which maximizes the energy of the blocks subject to the constraint that each block represents a physical object (line, plane or full 3D).

4.3. Interpretation of the Cost Function

The algorithm described in the previous sections has an interesting interpretation. This interpretation will support the decision reduce the search effort by using a hill-climbing strategy. We will show that the hill-climbing strategy finds a solution that represents the whole class of all possible solutions that make **Q** block diagonal.

Assuming that we have a correctly sorted **Q**, let us recall the definition of function $\varepsilon(\cdot)$ as in (47):

$$\varepsilon(m) = \sum_{i=1}^{m}\sum_{j=1}^{m} Q_{ij}^{*2}. \tag{52}$$

Now let us compute a family of functions $\varepsilon^{O}(\cdot)$ where O represents the set of all "correct" shape interaction matrices. By "correct" we mean all possible block diagonal matrices that result from permutations of rows and columns of $\mathbf{Q}^*$. For segmentation purposes these matrices are, in fact, indistinguishable. In short, these permutations switch columns and rows of features belonging to the same object.

Figure 6 illustrates how the set ε^{O} might look for the example considered throughout this section. In a noise-free environment, if $\mathbf{Q}^*$ contains M blocks, each of the functions ε^i, $(i \in O)$, has the same value for columns, say $C = \{c_1, \ldots, c_K, \ldots, c_M\}$, that limits each of the blocks (see Figs. 2 and 6). At each of these columns the function $\varepsilon^i(c_K)$ represents the total energy of the first K blocks:

$$\varepsilon^i(c_K) = \sum_{i=1}^{c_K}\sum_{j=1}^{c_K} Q_{ij}^{*2} = \sum_{n=1}^{K} \text{rank}(\mathbf{Q}_n). \tag{53}$$

Values $\varepsilon^i(c_K)$ are invariant to "correct" permutations of $\mathbf{Q}^*$ due to its block diagonal structure. In fact, they are the only important points for detecting the limits of the blocks. Among the whole family of functions, we denoted ε^* as,

$$\varepsilon^* = \max_{\forall_{i\in O}}(\varepsilon^i), \tag{54}$$

which bounds the whole set, and is represented by the thick curve in Fig. 6. Then, function ε^* maximizes the energy of any submatrix of $\mathbf{Q}^*$ formed by its first m columns and rows. Since values $\varepsilon^i(K)$ contain all the information needed for block detection, and are invariant to permutations that block diagonalize **Q**, all functions $\varepsilon^i(\ )$ can be represented by ε^* without any loss of information. As we showed in Section 4.1, function $\varepsilon^*(\ )$ can be computed by a hill-climbing search, thus reducing the search space to polynomial complexity.

Due to noise, the energy of the block will not be a whole number at the block's border. As Fig. 7 shows, at the block limiting columns, the value of the energy

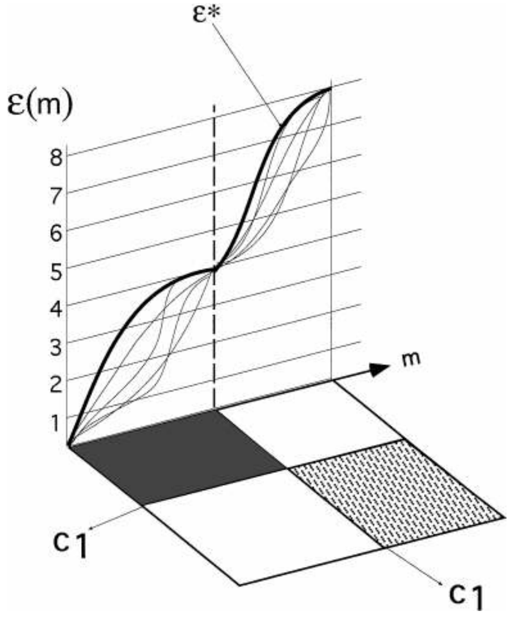

Figure 6. Several possibilities for energy functions. Each curve represents the energy function for a different set of permutations of the columns of **Q**. Function ε^* is an upper bound of the all set of ε^i.

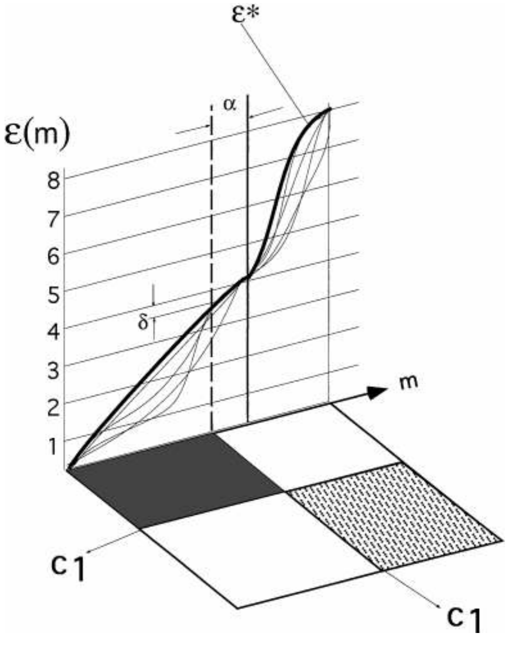

Figure 7. Noisy $\mathbf{Q}^*$. Funtion ε^* deviates due to the energy spread in the off-diagonal.

will exhibit a small difference δ from the integer crossing, which is the energy of the off-diagonal elements. Since we do not have a statistical description of noise, we cannot compute an estimate of δ and use it to modify the threshold of the possible block limiting columns. However, this limitation can be easily overcome. The energy of noise induces an uncertainty α in the position of the block limiting column (see Fig. 7). In other words, we do not know whether feature c_K, for which $\varepsilon^*(c_K)$ is integer, belongs to the previous block or the following one. By testing the linear dependence between feature c_K and the neighbouring ones, we can determine to which of the blocks it is closer.

Finally, recall that one of the reasons we did not use graph theoretical algorithms was because they rely on local information. Hence, the segmentation is done based on relationships between individual features, making the sorting quite sensitive to noisy features. As a consequence, due to a single strong noise spike between two features belonging to different objects, the MST algorithm joins two different objects through that link.

Instead, with our algorithm, the effect of a single noisy feature is smoothed.

Assume that there is one feature whose energy of interaction with features of a different object is high. Then, since we are sorting features by the strength of their coupling, the noisy elements will be understood (and sorted) as signal. This is illustrated in Fig. 8, where the correct column and row of a feature has been swapped with the column and row of a feature belonging to another object. If the block has $N_k \times N_k$ elements, the number of elements that have been wrongly swapped is $2N_k - 1$ (one row and one column), that is the ratio of noisy elements over the total size of the block is $(2N_k - 1)/N_k^2$. If $N_k \gg 1$, the influence of noise in the function $\varepsilon^*(\cdot)$ is of the order of $1/N_k$ of the noise to signal ratio. The drawback of gready strategies is its lack of memory, in other words, it could be the case that a particularly bad feature could be segmented to the wrong block in the early stages of the algorithm when only a few features have been processed but unless the scene is ubsurdly noisy the gready strategy is perfectly adequate.

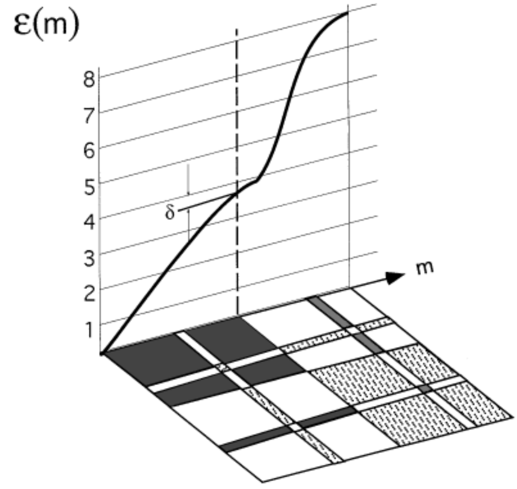

Figure 8. Noisy **Q** with misclassification of two features.

In summary, the fact that we use global constraints to sort the matrix **Q** by maximization of the energy of all its submatrices, produces a smoothing effect on noise, making the process more reliable against individual noise spikes.

5. Summary of Algorithm

Now the algorithm can be summarized as the sequence of the following steps:

1. Run the tracking process and create matrix **W**
2. Compute $r = \text{rank}(\mathbf{W})$
3. Decompose the matrix **W** using SVD, and yielding $\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$
4. Compute the shape interaction matrix **Q** using the first r rows of $\mathbf{V}^T$
5. Block-diagonalize **Q**
6. Permute matrix $\mathbf{V}^T$ into submatrices corresponding to a single object each
7. Compute $\mathbf{A}_i$ for each object, and obtain the corresponding shape and motion.

It should be clear by now that the segmentation algorithm presented above is independent of the number of objects, that is, the block diagonal structure of $\mathbf{Q}^*$ is valid for an arbitrary number of moving objects. Furthermore, this property holds also when the shape matrix of the objects has rank less than four (planes and lines) so that the total rank of **W** is the only required prior knowledge. Finally, note that instead of permuting columns of **W** in step 6 we permute columns of $\mathbf{V}^T$, which is equivalent.

6. Experiments

We will present two sets of experiments to demonstrate how the algorithm works. The first is an experiment

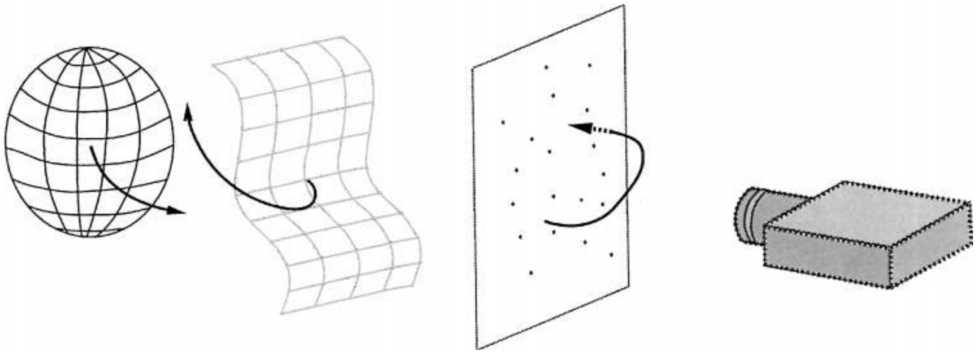

Figure 9. Synthetic scene. Three objects move transparently with arbitrary motion.

with synthetically generated feature trajectories, and the second with those extracted from real images taken in the laboratory under controlled imaging conditions.

6.1. Synthetic Data

Figure 9 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 10(a) illustrates the actual 3D motions of those 118 points.

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 Fig. 10(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 11(a) shows the shape interaction matrix **Q**: the height is the square of the entry value. The result of sorting the matrix into a bloackdiagonal form is shown in Fig. 11(b). We can observe the three blocks corresponding to objects 3, 2 and 1: all of the 118 features are correctly classified.

Figures 12(a), (b) and (c) show one view of each of the recovered shapes of the three objects in the same

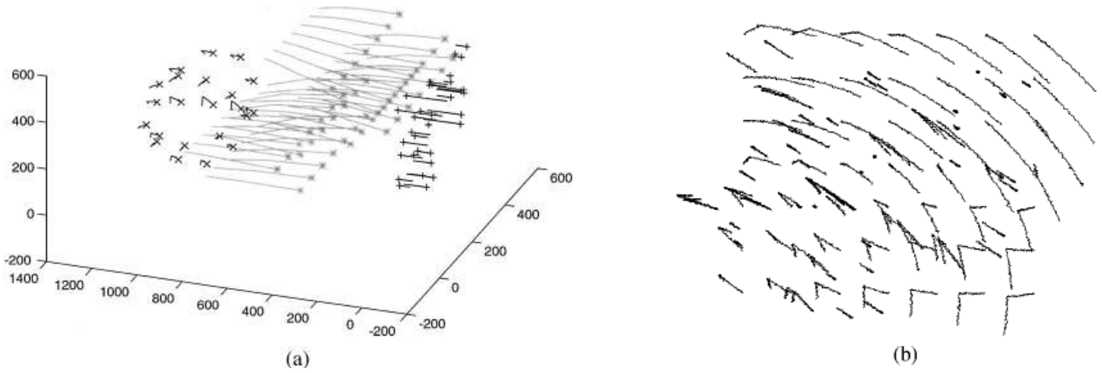

Figure 10. (a) 3D trajectories of the points and (b) noisy image tracks.

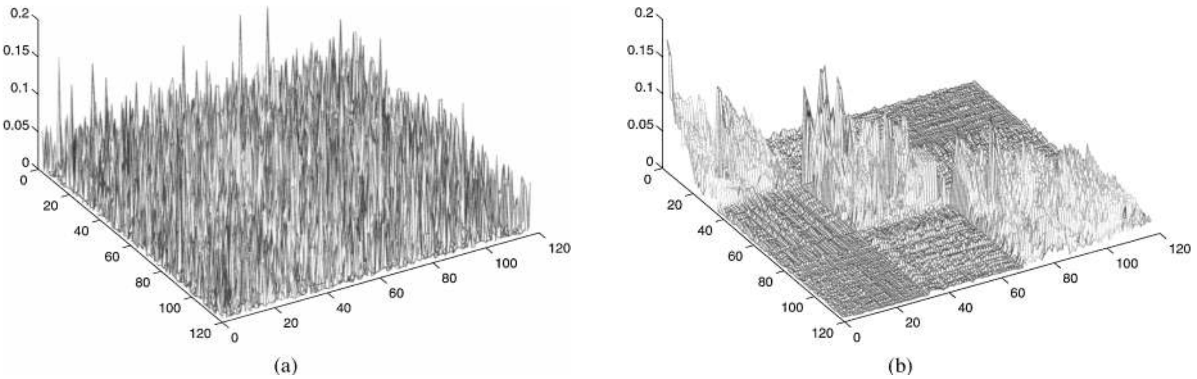

Figure 11. The shape interaction matrix for the synthetic scene with three transparent objects: (a) Unsorted matrix **Q**, and (b) sorted matrix **Q***.

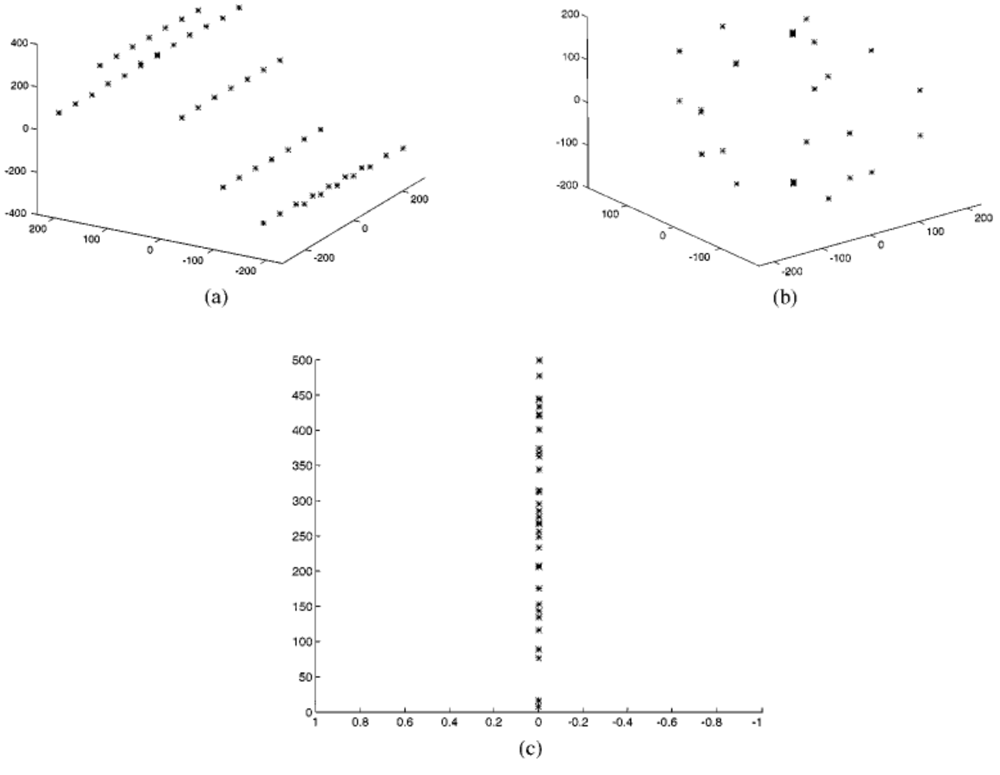

Figure 12. Recovered shape of the objects.

order as Fig. 4. Figure 12(c) showing the planar object viewed from its edge indicates the correct recovery of its shape.

6.2. 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

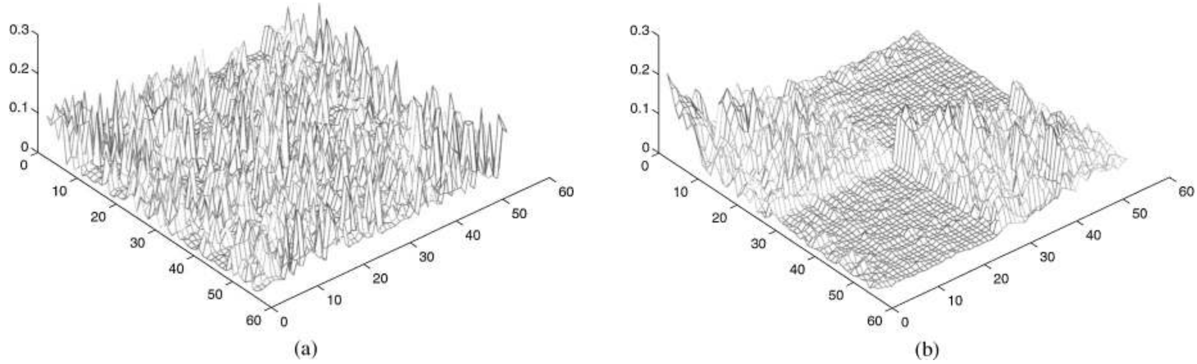

Figure 13. The shape interaction matrix for the lab scene: (a) Unsorted **Q** and (b) block-diagonalized $\mathbf{Q}^*$.

cylinder and 28 from the other, while the algorithm was not given that information.

Figure 14 shows the 85th 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.

Figure 13(a) show the shape interaction matrix **Q** for the unsorted input features. The sorted block diagonal matrix $\mathbf{Q}^*$ is shown in Fig. 13(b), and the features are grouped accordingly for individual shape recovery. The resultant three-dimensional points are displayed in Fig. 15 with linearly interpolated surface in order to convey a better perception of the their shape.

Figure 16 shows the profile of function ε^*. The rank detection algorithm, to be described later, computed a total rank of 6 (possibly due to a negligible translation component). The total energy of the off-diagonal block (noisy block) was 0.036, two orders of magnitude below the signal energy.

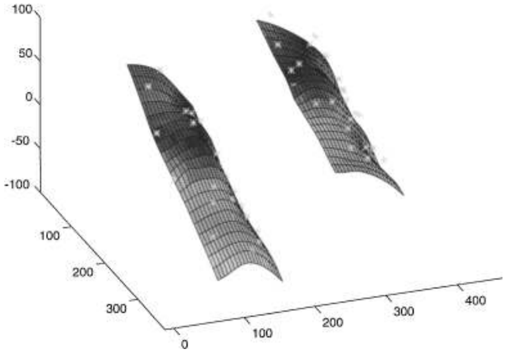

Figure 15. The recovered shape of the two cylinders.

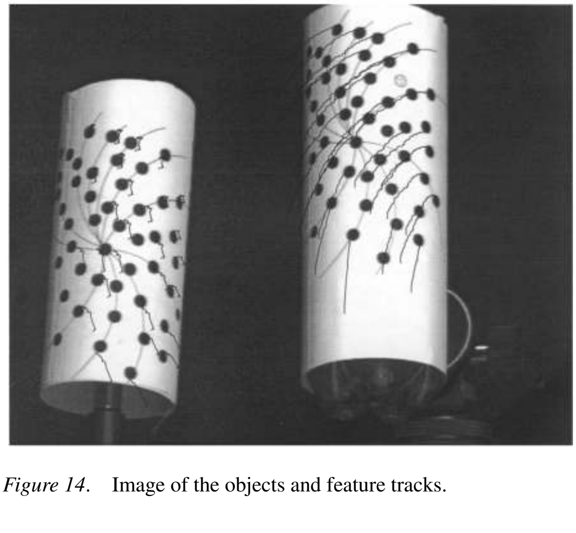

Figure 14. Image of the objects and feature tracks.

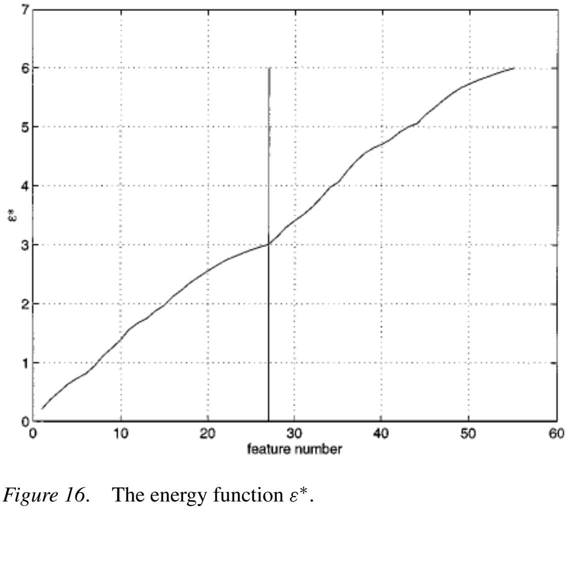

Figure 16. The energy function ε^*.

6.3. Noisy Outdoor Scene

In this section we show some tests done with images taken in an outdoor scene. The main difficulty in the analysis of this type of scenes is the lack of tracking reliability. Also, the objects in this scene move in a particular fashion highligting the shape and motion degeneracy problems.

Figure 17(a) shows the first of the 72 images of the sequence. The scene is formed by a moving face in the foreground and a still background. The unreliability of the tracking can be observed on the building in the background. Since they are still, the tracks should look like a dot or a small cloud. However, we observe that some tracks have a small linear motion, considerably greater than the one pixel variance which is the maximum that can be expected in laboratory environments. Note that the face moves essencially with translational motion and also the disproportionate number of fetaures between foreground and background. The shape interaction matrix for this scene can be observed unsorted in Fig. 17(b) and sorted in Fig. 18(a), with one block made of four features and another with the rest.

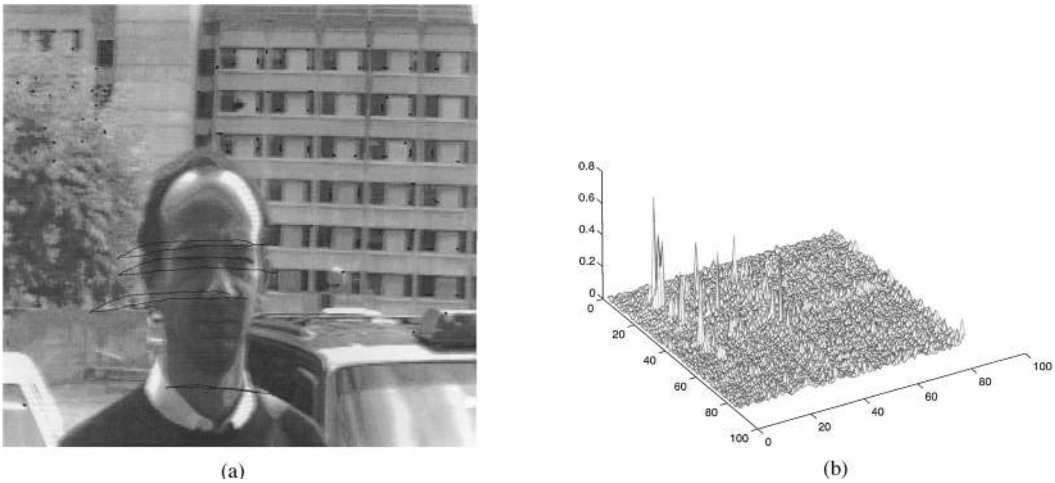

Figure 17. (a) First image with tracks and (b) the unsorted shape interaction matrix for the outdoor scene.

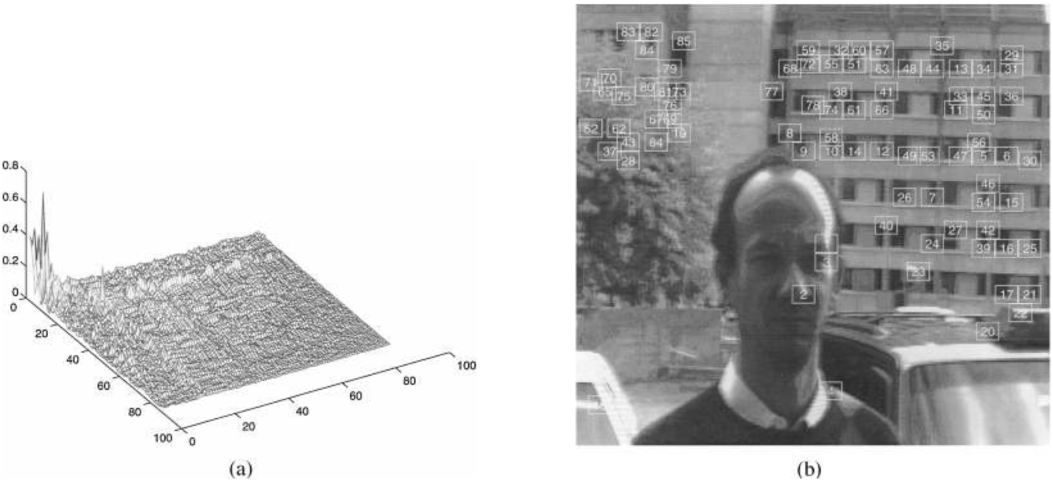

Figure 18. (a) The sorted shape interaction matrix for the outdoor scene and (b) list of the sorted features.

Notice the few peaks in the upper left corner of the matrix in contrast with the generally flat pattern. This is due to the fact that the total energy of the matrix is constant and the number of features is unbalanced between the two objects. Recall from previous discussions that, for features i and j belonging to the same object, Q_{ij} is given (again for a particular configuration of the object's coordinate system) by

$$Q_{ij} = S_i \mathbf{\Lambda}^{-1} S_j \tag{55}$$

$$\mathbf{\Lambda} = \mathbf{S}\mathbf{S}^T \tag{56}$$

$$= \begin{bmatrix} \sum X_n^2 & 0 & 0 & 0 \\ 0 & \sum Y_n^2 & 0 & 0 \\ 0 & 0 & \sum Z_n^2 & 0 \\ 0 & 0 & 0 & N \end{bmatrix} \tag{57}$$

The norm of matrix $\mathbf{\Lambda}$ increases, in general, with the number of points. The value of Q_{ij}, which depends on the inverse of $\mathbf{\Lambda}$, decreases with the same number. This is one of the drawbacks of the methodology for large numbers of features. In Section 9 we will see that noise energy grows with the size of the measurements matrix, and since the total energy of $\mathbf{Q}$ is constant, for larger numbers of features the noise influence becomes more important. A more detailed analysis on this experimental data can be found in (Costeira, 1997).

7. Tolerance to Perspective Distortion

The multibody factorization method assumes that the scene is viewed by an orthographic camera. The general requirement here is that the object is small compared to its distance from the camera. In the cases where the perspective effect can be accounted by a scale effect, the method still holds. Consider the perspective projection of feature i at frame f, described by a pinhole camera (see also Eq. (2)):

$$u_{if} = F_d \frac{\mathbf{i}_f^T \mathbf{p}_i + t_{x_f}}{\mathbf{k}_f^T \mathbf{p}_i + t_{z_f}} \tag{58}$$

$$v_{if} = F_d \frac{\mathbf{j}_f^T \mathbf{p}_i + t_{y_f}}{\mathbf{k}_f^T \mathbf{p}_i + t_{z_f}} \tag{59}$$

where $\mathbf{k}$ is the third row of the rotation matrix, t_{z_f} is the translation component along the Z axis and F_d is the focal distance. The observation matrix $\mathbf{W}$ can be written in the following form:

$$\begin{bmatrix} u_{11}\left(\mathbf{k}_1^T \mathbf{p}_1 + t_{z_1}\right) & \cdots & u_{1N}\left(\mathbf{k}_1^T \mathbf{p}_N + t_{z_1}\right) \\ \vdots & & \vdots \\ u_{F1}\left(\mathbf{k}_F^T \mathbf{p}_1 + t_{z_F}\right) & \cdots & u_{FN}\left(\mathbf{k}_F^T \mathbf{p}_N + t_{z_F}\right) \\ v_{11}\left(\mathbf{k}_1^T \mathbf{p}_1 + t_{z_1}\right) & \cdots & v_{1N}\left(\mathbf{k}_1^T \mathbf{p}_N + t_{z_1}\right) \\ \vdots & & \vdots \\ v_{F1}\left(\mathbf{k}_F^T \mathbf{p}_1 + t_{z_F}\right) & \cdots & v_{FN}\left(\mathbf{k}_F^T \mathbf{p}_N + t_{z_F}\right) \end{bmatrix}$$

$$= F_d \left[\begin{array}{c|c} \mathbf{i}_1^T & t_{x_1} \\ \vdots & \vdots \\ \mathbf{i}_F^T & t_{x_F} \\ \mathbf{j}_1^T & t_{y_1} \\ \vdots & \vdots \\ \mathbf{j}_f^T & t_{y_F} \end{array}\right] [\mathbf{s}_1 \quad \cdots \quad \mathbf{s}_N]\,. \tag{60}$$

Dividing each row by the translation t_{z_f} we obtain the perspective observation model:

$$\begin{bmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & & \vdots \\ u_{F1} & \cdots & u_{FN} \\ v_{11} & \cdots & v_{1N} \\ \vdots & & \vdots \\ v_{F1} & \cdots & v_{FN} \end{bmatrix} + \begin{bmatrix} u_{11}\frac{\mathbf{k}_1^T\mathbf{p}_1}{t_{z_1}} & \cdots & u_{1N}\frac{\mathbf{k}_1^T\mathbf{p}_N}{t_{z_1}} \\ \vdots & & \vdots \\ u_{F1}\frac{\mathbf{k}_F^T\mathbf{p}_1}{t_{z_F}} & \cdots & u_{FN}\frac{\mathbf{k}_F^T\mathbf{p}_N}{t_{z_F}} \\ v_{11}\frac{\mathbf{k}_1^T\mathbf{p}_1}{t_{z_1}} & \cdots & v_{1N}\frac{\mathbf{k}_1^T\mathbf{p}_N}{t_{z_1}} \\ \vdots & & \vdots \\ v_{F1}\frac{\mathbf{k}_F^T\mathbf{p}_1}{t_{z_F}} & \cdots & v_{FN}\frac{\mathbf{k}_F^T\mathbf{p}_N}{t_{z_F}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{F_d}{t_{z_1}} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \frac{F_d}{t_{z_F}} \end{bmatrix} \left[\begin{array}{c|c} \mathbf{i}_1^T & t_{x_1} \\ \vdots & \vdots \\ \mathbf{i}_F^T & t_{x_F} \\ \mathbf{j}_1^T & t_{y_1} \\ \vdots & \vdots \\ \mathbf{j}_f^T & t_{y_F} \end{array}\right] [\mathbf{s}_1 \quad \cdots \quad \mathbf{s}_N] \tag{61}$$

$$\mathbf{W} + \tilde{\mathbf{W}} = K\mathbf{M}\mathbf{S}. \tag{62}$$

The orthographic model is adequate if $\tilde{\mathbf{W}}$ is small. One possible criteria is to consider an upper limit of the Frobenius norm of matrix **W**:

$$\|\tilde{\mathbf{W}}\|_F^2 \leq \|\tilde{\mathbf{W}}_{\max}\|_F^2 = \frac{u_{\max}^2}{t_{z_{\min}}^2} \sum_{f=1}^{2F} \sum_{p=1}^{N} \mathbf{k}_f \mathbf{p}_p \mathbf{p}_p^T \mathbf{k}_f^T$$
$$= \frac{u_{\max}^2}{t_{z_{\min}}^2} \sum_{f=1}^{2F} \mathbf{k}_f \Lambda \mathbf{k}_f^T \qquad (63)$$

where $u_{\max} = \max\{u_{fp}\}$ and $t_{z_{\min}}$ is the object's (centroid) minimum distance to the camera. Equation (63) shows that as long as the object is far away from the camera and/or the object's ellipsoid of inertia has a small projection in the camera axis, the subspace structure of **W** is well approximated. The scaling due to perspective affects greatly the shape reconstruction but not the segmentation. With the above equation we see that it is not possible to represent **W** as a product of an orthonornal matrix (motion) by a shape matrix but the rows of **W** are still a linear combination of the rows of **S**. Error analysis has been done in (Poelman, 1995) for the case of paraperspective factorization reconstruction for various types of scenes. In our experiments a 1 m long object at 6 m distance with a 70 mm lens, the perspective effect is negligible.

8. Shape and Motion Degeneracies

As said before, the shape interaction matrix keeps its structure when objects have degenerate shape, in other words, when the shape matrix **S** has rank less than four. The same is valid to motion degeneracies where the motion matrix **M** is rank deficient. Using a trivial example, suppose one object is moving with translational motion. The motion matrix will have the form

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & t_{x_1} \\ & \vdots & & \\ 1 & 0 & 0 & t_{x_1} \\ 0 & 1 & 0 & t_{y_1} \\ & \vdots & & \\ 0 & 1 & 0 & t_{y_1} \end{bmatrix}. \qquad (64)$$

In this case, there is a shape ambiguity since the product **MS** multiplyes the third row of **S** by 0 (the Z component); therefore, we can remove the third column of **M** and the third row of **S** and interpret the observations as resulting from a moving plane. This interpretation is valid for any motion degeneracy where matrix **M** has some columns that are linearly dependent on others. This poses no problem to the multibody segmentation since the properties of **Q** will hold. A different case is when the motion matrix is linearly dependent on another object's motion. Here, the block-diagonal properties of **Q** vanish and the algorithm wouldn't find the correct block transitions.

9. Computing the Rank of Matrix W

In the previous theoretical developments we assumed that the rank of the measurements matrix, **W**, was known. In order to build the shape interaction matrix, **Q**, this knowledge is essential since the rank specifies the number of singular vectors **V** from which **Q** is created. Due to camera noise and other tracking errors, the rank of matrix **W** will, in general, differ from the correct one. Therefore, we need a rank determination procedure which selects the significant singular values and vectors.

Several approaches have been developed regarding the subject of signal/noise separation (e.g., the MUSIC algorithm (Bienvenu, 1979; Schmidt, 1980)) but since we use SVD to build our constructs, and given its rank revealing properties and numerical stability, we closely follow the approach of (Stewart, 1992), formalizing the rank determination problem under the SVD framework and including uncertainty models of the feature tracking process. We present the final result but a full derivation together with tracking uncertainty can be seen in (Costeira, 1997).

The rank of the observations matrix, **W**, is determined by an approximation criterion for which the noise model is required. We model the noisy imaging process as the result of additive noise to the projection equations (2) (Wilson, 1994). Then, the ith feature position, in frame f, is given by

$$\begin{bmatrix} \tilde{u}_{f,i} \\ \tilde{v}_{f,i} \end{bmatrix} = \begin{bmatrix} u_{f,i} \\ v_{f,i} \end{bmatrix} + \begin{bmatrix} \mathcal{E}_{u_{f,i}} \\ \mathcal{E}_{v_{fi}} \end{bmatrix}, \qquad (65)$$

where $\mathcal{E}_{u_{fi}}$ and $\mathcal{E}_{v_{fi}}$ are the additive noise components in the X and Y directions of the image plane. From Eq. (65) we foresee the need for feature tracking modeling, namely the characterization of the feature position.

The $2F \times N$ noisy measurement matrix, $\tilde{\mathbf{W}}$, in this case, will be given by the matrix sum

$$\tilde{\mathbf{W}} = \mathbf{MS} + \begin{bmatrix} \mathcal{E}_u \\ \mathcal{E}_v \end{bmatrix} \qquad (66)$$

$$\tilde{\mathbf{W}} = \mathbf{W} + \mathcal{E}, \tag{67}$$

where $\mathcal{E}_u$ and $\mathcal{E}_v$ are two $F \times N$ matrices and $\mathbf{W}$ is the noise-free measurement matrix, whose rank we want to estimate. It is now clear from (67) that the rank of $\tilde{\mathbf{W}}$ can be at most N. Then, the singular value decomposition will be given by

$$\tilde{\mathbf{W}} = \mathbf{U}\boldsymbol{\Sigma}V^T \tag{68}$$

$$\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_N) \tag{69}$$

$$\mathbf{U} \in R^{2F \times N} \tag{70}$$

$$\mathbf{V} \in R^{N \times N}. \tag{71}$$

In a noise free environment the rank of $\mathbf{W}$ is determined by the number of nonzero singular values σ_i, whereas in the noisy case we will have to compute a threshold that separates the noisy singular values from the significant ones. From the SVD decomposition of $\tilde{\mathbf{W}}$ we have to estimate the number of columns of $\mathbf{U}$, $\mathbf{V}$ and the singular values σ_i that represent an estimate $\hat{\mathbf{W}}$ of $\mathbf{W}$.

The singular value decomposition of $\tilde{\mathbf{W}}$ "spreads" the noise components over all the elements of $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\Sigma}$. In other words, it is not possible to isolate the noise-free components of these matrices or even directly estimate noise influence. Then, we will seek an estimate $\hat{\mathbf{W}}$, with the same rank of $\mathbf{W}$, that approximates $\tilde{\mathbf{W}}$ in the least squares sense (Stewart, 1992; Demmel, 1987; Golub, 1987).

In other words, we must solve two problems here: we have to determine the "real" rank of $\tilde{\mathbf{W}}$ and also obtain an approximation of the "real" observations matrix $\mathbf{W}$. Then, following the minimum error criterion, if r is the rank of the noise-free measurement matrix $\mathbf{W}$, for all possible $2F \times N$ matrices $\mathbf{Y}$ with $\operatorname{rank}(\mathbf{Y}) \leq r$, we seek an estimate $\hat{\mathbf{W}}$ that minimizes the error:

$$\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2 = \min_{\operatorname{rank}(\mathbf{Y}) \leq r} \|\tilde{\mathbf{W}} - \mathbf{Y}\|_F^2. \tag{72}$$

Fischer's theorem (Stewart, 1992) states that a solution $\hat{\mathbf{W}}$ exists, and the error (72) is explicitly given by:

$$\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2 = \sigma_{r+1}^2 + \cdots + \sigma_N^2. \tag{73}$$

Expression (73) is equivalent to saying that the approximation given by

$$\hat{\mathbf{W}} = \mathbf{U}_{2F \times r} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \mathbf{V}_{N \times r}^T \tag{74}$$

is the minimum error approximation of $\tilde{\mathbf{W}}$ by a rank r matrix. Comparing with the "correct" equations, we have the following correspondence:

$$\tilde{\mathbf{W}} = \mathbf{W} + \mathcal{E} \tag{75}$$

$$\tilde{\mathbf{W}} = \hat{\mathbf{W}} + \hat{\mathcal{E}} \tag{76}$$

$$\begin{aligned} \tilde{\mathbf{W}} = {} & \mathbf{U}_{2F,1:r} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \mathbf{V}_{N,1:r}^T \\ & + \mathbf{U}_{2F,r+1:N} \begin{bmatrix} \sigma_{r+1} & & \\ & \ddots & \\ & & \sigma_N \end{bmatrix} \mathbf{V}_{N,r+1:N}^T \end{aligned} \tag{77}$$

where the notation $_{i:j}$ denotes column or row range. Since $\hat{\mathbf{W}}$ is the closest matrix to $\tilde{\mathbf{W}}$ with rank r, its error is minimum; in particular, it is smaller than the error in the real measurements $\mathbf{W}$, that is:

$$\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2 \leq \|\tilde{\mathbf{W}} - \mathbf{W}\|_F^2 = \|\mathcal{E}\|_F^2. \tag{78}$$

Using (73) in (78) yields:

$$\sigma_{r+1}^2 + \cdots + \sigma_N^2 \leq \|\mathcal{E}\|_F^2 = \sum_{i=1}^{2F} \sum_{j=1}^{N} \mathcal{E}_{ij}^2. \tag{79}$$

Using this relation and knowing the magnitude (norm) of the noise matrix, we can define the rank of $\mathbf{W}$ as the smallest integer, r, such that inequality (79) holds, or equivalently, the smallest integer r, for which the sum of the last $N - r$ singular values of the noisy matrix is less than or equal to the norm of the noise matrix.

However, there is one problem with this strategy: the entries of matrix $\mathcal{E}$ are stochastic variables; therefore, we do not know their absolute values (realizations). If the feature tracking provided a statistical description of the feature positions we could have a decision based on mean values of the noise component. Then, if we compute the mean of Eq. (79) we obtain the relation:

$$\begin{aligned} E\big[\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_F^2\big] &= E\big[\sigma_{k+1}^2\big] + \cdots + E\big[\sigma_N^2\big] \\ &\leq E\big[\|\mathcal{E}\|_F^2\big] \qquad (80) \\ &\leq \sum_{i=1}^{N} \sum_{j=1}^{2F} E\big[\mathcal{E}_{ij}^2\big]. \qquad (81) \end{aligned}$$

The terms $\mathcal{E}_{ij}$ in (81) are the statistical second moment (variance) of the feature noise. Using the notation as in (65), the covariance of the noise is given by

$$\sum_{i=1}^{N}\sum_{j=1}^{2F} E\left[\mathcal{E}_{ij}^{2}\right] = \sum_{i=1}^{N}\sum_{f=1}^{F}\left(E\left[\mathcal{E}_{u_{f,i}}^{2}\right] + E\left[\mathcal{E}_{v_{f,i}}^{2}\right]\right). \tag{82}$$

In general, for each sampling time, the uncertainty of each feature point is characterized by a 2 × 2 covariance matrix

$$\mathbf{\Pi}_{f,i} = E\left(\begin{bmatrix}\mathcal{E}_{u_{fi}}\\ \mathcal{E}_{v_{fi}}\end{bmatrix}\begin{bmatrix}\mathcal{E}_{u_{fi}} & \mathcal{E}_{v_{fi}}\end{bmatrix}\right) \tag{83}$$

$$= \begin{bmatrix} E\left[\mathcal{E}_{u_{fi}}^{2}\right] & E\left[\mathcal{E}_{u_{fi}}\ \mathcal{E}_{v_{fi}}\right] \\ E\left[\mathcal{E}_{u_{fi}}\ \mathcal{E}_{v_{fi}}\right] & E\left[\mathcal{E}_{v_{fi}}\right] \end{bmatrix} \tag{84}$$

$$= \begin{bmatrix} \pi_{u_{fi}}^{2} & \pi_{uv_{fi}} \\ \pi_{uv_{fi}} & \pi_{v_{fi}}^{2} \end{bmatrix}. \tag{85}$$

This uncertainty is provided by a statistically modeled feture tracker (see Faugeras (1994)). The error (81) is finally expressed as

$$E\left[\|\tilde{\mathbf{W}} - \hat{\mathbf{W}}\|_{F}^{2}\right] = E\left[\sigma_{k+1}^{2}\right] + \cdots + E\left[\sigma_{N}^{2}\right]$$
$$\leq \sum_{i=1}^{N}\sum_{f=1}^{F}\left(\pi_{u_{f,i}}^{2} + \pi_{v_{fi}}^{2}\right). \tag{86}$$

From (86) we can finally describe the procedure to detect the rank of $\mathbf{W}$:

1. Decompose the real measurements matrix $\tilde{\mathbf{W}}$ using SVD.
2. Compute the sum of the last $N - r$ singular values $(\sigma_r + \cdots + \sigma_N)$.
3. Find the rank as the value of r for which $\sigma_r + \cdots + \sigma_N \leq \mathcal{T}\sum_{p=1}^{N}\sum_{f=1}^{F}(\pi_{u_{f,p}}^{2} + \pi_{v_{fp}}^{2})$, where $\mathcal{T}$ is an experimental constant used to adjust for unaccounted deviations and for the difference between the actual value and the expected value of σ_i.

10. Discussion and Conclusion

In this paper we have shown that the problem of multibody 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. 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. Another interesting fact is that the shape interaction matrix can handle many degenerate cases as well, where objects may be full 3D object but also linear or planar. More research is required for the degenerate cases including the cases where the motions are coupled.

Acknowledgments

The authors wish to thank Martial Hebert, José Moura and José Dias for useful suggestions and comments to this work.

Notes

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, since for example a rank-8 measurements 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 hence attempting to find two rank-4 subspaces could be wrong.
2. $\mathbf{V}$ and $\mathbf{V}^{*}$ may still differ up to an orthonormal transformation, but this is irrelevant to our derivations.
3. Note that the diagonal elements are not included in the cost since they do not measure any interaction among features. Also note that the matrix is symmetric so we only need to perform half the computations.

References

1. Adelson, E. and Bergen, J. 1985. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299.
2. Bergen, J., Burt, P., Hingorani, R., and Peleg, S. 1990. Computing two motions from three frames. In *Proceedings of the IEEE International Conference on Computer Vision*.
3. Bienvenu, G. and Kopp, L. 1979. Principe de la noniometre passive adaptive. In *Proc. 7'eme Colloque GRETSI*, Nice, France, pp. 106/1–106/10.

4. Boult, T. and Brown, L. 1991. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*.
5. Cormen, T.H., Leiserson, C.E., and Rivest, R.L. 1986. *Introduction to Algorithms*. The MIT Press.
6. Costeira, J. and Kanade, T. 1997. A multi-body factorization method for independently moving objects: Full report. Technical Report CMU-RI-TR-97-30, Robotics Institute, Carnegie Mellon University. Also available at http://www.isr.ist.utl.pt/˜jpc.
7. Demmel, J. 1987. The smallest perturbation of a submatrix which lowers the rank and constrained total least squares problems. *SIAM Journal of Numverical Analysis*, 24(1).
8. Faugeras, O. 1994. *Three Dimensional Computer Vision*. MIT Press: Cambridge, MA.
9. Gear, C.W. 1994. Feature grouping in moving objects. In *Proceedings of the Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, Texas.
10. Golub, G., Hoffman, A., and Stewart, G. 1987. A generalization of the eckart-young-mirsky approximation theorem. *Linear Algebra Applications*.
11. Irani, M., Benny, R., and Peleg, S. 1994. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1):5–16.
12. Jasinschi, R.S., Rosenfeld, A., and Sumi, K. 1992. Perceptual motion transparency: the role of geometrical information. *Journal of the Optical Society of America*, 9(11):1–15.
13. Koenderink, J. and van Doorn, A. 1991. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385.
14. Poelman, C. and Kanade, T. 1993. A paraperspective factorization method for shape and motion recovery. Technical Report CS-93-219, School of Computer Science, Carnegie Mellon University.
15. Poelman, C. 1995. The paraperspective and projective factorization method for recovering shape and motion. Technical Report Also SCS Report CMU-CS-95-173, School of Computer Science, Carnegie Mellon University.
16. Schmidt, R. 1980. A signal subspace approach to multiple emitter location and spectral estimation. PhD Thesis, Stanford University, CA.
17. Sinclair, D. 1993. Motion segmentation and local structure. In *Proceedings of the 4th International Conference on Computer Vision*.
18. Stewart, G.W. 1992. Determining rank in the presence of error. In *Proceedings of the NATO Workshop on Large Scale Linear Algebra*, Leuven, Belgium. Also University of Maryland Tech. Report.
19. Tomasi, C. and Kanade, T. 1992. Shape from motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154. Originally published as CMU Technical Report CMU-CS-90-166, September 1990.
20. Ullman, S. 1983. Maximizing rigidity: The incremental recovery of 3D structure from rigid and rubbery motion. Technical Report A.I. Memo No. 721, MIT.
21. Van Trees. H. 1968. *Detection, Estimation, and Modulation Theory*, vol. 1. Wiley: New York.
22. Wilson, R. 1994. Modeling and calibration of automated zoom lenses. PhD Thesis, ECE, Carnegie Mellon University.