The Multibody Trifocal Tensor: Motion Segmentation from 3 Perspective Views

Richard Hartley[1,2]
[1]Dept. of Systems Engineering
Australian National University
Richard.Hartley@anu.edu.au

and

[2]National ICT Australia

René Vidal[2,3]
[3]Center for Imaging Science
Johns Hopkins University
rvidal@cis.jhu.edu

Abstract

We propose a geometric approach to 3-D motion segmentation from point correspondences in three perspective views. We demonstrate that after applying a polynomial embedding to the correspondences they become related by the so-called multibody trilinear constraint and its associated multibody trifocal tensor. We show how to linearly estimate the multibody trifocal tensor from point-point-point correspondences. We then show that one can estimate the epipolar lines associated with each image point from the common root of a set of univariate polynomials and the epipoles by solving a plane clustering problem in $\mathbb{R}^3$ using GPCA. The individual trifocal tensors are then obtained from the second order derivatives of the multibody trilinear constraint. Given epipolar lines and epipoles, or trifocal tensors, we obtain an initial clustering of the correspondences, which we use to initialize an iterative algorithm that finds an optimal estimate for the trifocal tensors and the clustering of the correspondences using Expectation Maximization. We test our algorithm on real and synthetic dynamic scenes.

1. Introduction

One of the most important problems in visual motion analysis is that of reconstructing a 3-D scene from a collection of images taken by a moving camera. At present, the algebra and geometry of the problem is very well understood, and it is usually described in terms of the so-called bilinear, trilinear, and multilinear constraints among two, three and multiple views, respectively. Also, there are various algorithms for performing the reconstruction task, both geometric and optimization-based [4].

All the above algorithms are, however, limited by the assumption that the scene is *static*, i.e. only the camera is moving and hence there is a single motion model to be estimated from the image measurements. In practice, however, most of the scenes are *dynamic*, i.e. both the camera and multiple objects in the 3-D world are moving. Thus, one is faced with the more challenging problem of recovering multiple motion models from the image data, without knowing the assignment of data points to motion models.

Previous work on 3-D motion segmentation [2, 9] has addressed the problem using the standard probabilistic approach. Given an initial clustering of the image data, one estimates a motion model for each group using standard structure from motion algorithms. Given the motion parameters, one can easily update the clustering of the correspondences. The method then proceeds by iterating between these two steps, using the Expectation Maximization (EM) algorithm. When the probabilistic model generating the data is known, this iterative method provides an optimal estimate in the maximum likelihood sense. However, it is well-known that EM is very sensitive to initialization [8].

In order to deal with the initialization problem, recent work on 3-D motion segmentation has concentrated on the study of the geometry of multiple motion models. [10] proposed a polynomial factorization algorithm for segmenting purely translating objects. [14] derived a bilinear constraint in $\mathbb{R}^6$ which, together with a combinatorial scheme, segments two rigid-body motions from two perspective views. [13] proposed a generalization of the epipolar constraint and of the fundamental matrix to multiple rigid-body motions, which leads to a motion segmentation algorithm based on factoring products of epipolar constraints to retrieve the fundamental matrices associated with each one of the motions. [11] extended this method to most two-view motion models, such as affine, translational and planar homographies, by fitting and differentiating complex polynomials.

In this paper, we consider the problem of estimating and segmenting multiple rigid-body motions from a set of point correspondences in *three* perspective views. In Section 2 we study the three-view geometry of multiple rigid-body motions. We demonstrate that, after a suitable embedding into a higher-dimensional space, the three views are related by the so-called multibody trilinear constraint and its associated multibody trifocal tensor. In Section 3, we propose a geometric algorithm for 3-D motion segmentation that estimates the motion parameters (epipoles, epipolar lines and trifocal tensors) from the derivatives of the multibody trilinear constraint. This algebraic (non-iterative) solution is then used to initialize an optimal algorithm. To the best of our knowledge, there is no previous work addressing this problem. The only existing works on multiframe 3-D motion segmentation are for points moving on a line in three perspective views [7], and for rigid-body motions in three or more [12] and/or four or more affine views [1, 5].

2. Multibody three-view geometry

This section establishes the basic geometric relationships among three perspective views of multiple rigid-body motions. We first review the trilinear constraint and its associated trifocal tensor for the case of a single motion. We then generalize these notions to multiple motions via a polynomial embedding that leads to the so-called multibody trilinear constraint and its associated multibody trifocal tensor.

2.1. Trilinear constraint and trifocal tensor

Let $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell}' \leftrightarrow \boldsymbol{\ell}''$ be a point-line-line correspondence in three perspective views with 3×4 camera matrices

$$\mathtt{P} = [I\ 0],\ \mathtt{P}' = [R'\ \boldsymbol{e}'] \text{ and } \mathtt{P}'' = [R''\ \boldsymbol{e}''], \tag{1}$$

where $\boldsymbol{e}' \in \mathbb{P}^2$ and $\boldsymbol{e}'' \in \mathbb{P}^2$ are the epipoles in the 2^{nd} and 3^{rd} views, respectively. Then, the multiple view matrix [6]

$$\begin{bmatrix} \boldsymbol{\ell}'^\top R'\boldsymbol{x} & \boldsymbol{\ell}'^\top \boldsymbol{e}' \\ \boldsymbol{\ell}''^\top R''\boldsymbol{x} & \boldsymbol{\ell}''^\top \boldsymbol{e}'' \end{bmatrix} \in \mathbb{R}^{2\times 2} \tag{2}$$

must have rank 1, hence its determinant must be zero, i.e.

$$\boldsymbol{\ell}'^\top (R'\boldsymbol{x}\boldsymbol{e}''^\top - \boldsymbol{e}'\boldsymbol{x}^\top R''^\top)\boldsymbol{\ell}'' = 0. \tag{3}$$

This is the well-known point-line-line *trilinear constraint* among the three views [4], which we will denote as

$$\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T = 0 \tag{4}$$

where $T \in \mathbb{R}^{3\times 3\times 3}$ is the so-called *trifocal tensor*.

Notation. For ease of notation, we will drop the summation and the subscripts in trilinear expressions such as $\sum_{ijk} x_i \ell'_j \ell''_k T_{ijk}$, and write them as shown above. Similarly, we will write $\boldsymbol{x}T$ to represent the matrix whose $(jk)^{th}$ entry is $\sum_i x_i T_{ijk}$, and $\boldsymbol{x}\boldsymbol{\ell}'T$ to represent the vector whose k^{th} entry is $\sum_{ij} x_i \ell'_j T_{ijk}$. The notation is somewhat condensed, and inexact, since the particular indices that are being summed over are not specified. However, the meaning should in all cases be clear from the context.

Notice that one can linearly solve for the trifocal tensor T from the trilinear constraint (4) given at least 26 point-line-line correspondences. However, if we are given point-point-point correspondences $\boldsymbol{x} \leftrightarrow \boldsymbol{x}' \leftrightarrow \boldsymbol{x}''$, then for each point in the 2^{nd} view $\boldsymbol{x}'$, we can obtain two lines $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ passing through $\boldsymbol{x}'$, and similarly for the 3^{rd} view. Since each correspondence gives 4 independent equations on T, we only need 7 correspondences to linearly estimate T.[1]

2.2. The multibody trilinear constraint

Consider now a scene with a *known* number n of rigid-body motions with associated trifocal tensors $\{T_i \in \mathbb{R}^{3\times 3\times 3}\}_{i=1}^n$, where T_i is the trifocal tensor associated with the motion of the i^{th} object relative to the moving camera among the three views. We assume that the motions of the objects relative to the camera are such that all the trifocal tensors are different up to a scale factor. We also assume that the given images correspond to 3-D points in general configuration in $\mathbb{R}^3$, i.e. they do not all lie in any critical surface, for example.

Let $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell}' \leftrightarrow \boldsymbol{\ell}''$ be an arbitrary point-line-line correspondence associated with *any* of the n motions. Then, there exists a trifocal tensor T_i satisfying the trilinear constraint in (3) or (4). Thus, regardless of the motion associated with the correspondence, the following constraint must be satisfied by the number of independent motions n, the trifocal tensors $\{T_i\}_{i=1}^n$ and the correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell}' \leftrightarrow \boldsymbol{\ell}''$

$$\prod_{i=1}^{n} (\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_i) = 0. \tag{5}$$

The above *multibody constraint* eliminates the problem of clustering the correspondences from the motion segmentation problem by taking the product of all trilinear constraints. Although taking the product is not the only way of algebraically eliminating feature segmentation, it has the advantage of leading to a polynomial equation in $(\boldsymbol{x}, \boldsymbol{\ell}', \boldsymbol{\ell}'')$ with a nice algebraic structure. Indeed, the multibody constraint is a homogeneous polynomial of degree n in each of $\boldsymbol{x}$, $\boldsymbol{\ell}'$ or $\boldsymbol{\ell}''$. Now, suppose $\boldsymbol{x} = (x_1, x_2, x_3)^\top$. We may enumerate all the possible monomials $x_1^{n_1} x_2^{n_2} x_3^{n_3}$ of degree n in (5) and write them in some chosen order as a vector

$$\tilde{\boldsymbol{x}} = (x_1^n, x_1^{n-1}x_2, x_1^{n-1}x_3, x_1^{n-2}x_2^2, \ldots, x_3^n)^\top \ . \tag{6}$$

This vector has dimension $M_n = (n+1)(n+2)/2$. The map $\boldsymbol{x} \mapsto \tilde{\boldsymbol{x}}$ is known as the polynomial embedding of degree n in the machine learning community and as the Veronese map of degree n in the algebraic geometry community.

Now, note that (5) is a sum of terms of degree n in each of $\boldsymbol{x}$, $\boldsymbol{\ell}'$ and $\boldsymbol{\ell}''$. Thus, each term is a product of degree n monomials in $\boldsymbol{x}$, $\boldsymbol{\ell}'$ and $\boldsymbol{\ell}''$. We may therefore define a 3-dimensional tensor $\mathcal{T} \in \mathbb{R}^{M_n \times M_n \times M_n}$ containing the coefficients of each of the monomials occurring in the product (5) and write the multibody constraint (5) as

$$\boxed{\tilde{\boldsymbol{x}}\ \widetilde{\boldsymbol{\ell}'}\ \widetilde{\boldsymbol{\ell}''}\ \mathcal{T} = 0,} \tag{7}$$

where summation over all the entries of the vectors $\tilde{\boldsymbol{x}}$, $\widetilde{\boldsymbol{\ell}'}$ and $\widetilde{\boldsymbol{\ell}''}$ is implied. We call equation (7) the *multibody trilinear constraint*, as it is a natural generalization of the *trilinear constraint* valid for $n = 1$. The important point to observe is that although (7) has degree n in the entries of $\boldsymbol{x}$, $\boldsymbol{\ell}'$ and $\boldsymbol{\ell}''$, it is in fact *linear* in the entries of $\tilde{\boldsymbol{x}}$, $\widetilde{\boldsymbol{\ell}'}$ and $\widetilde{\boldsymbol{\ell}''}$.

2.3. The multibody trifocal tensor

The array $\mathcal{T}$ is called the *multibody trifocal tensor*, defined up to indeterminate scale, and is a natural generalization of the trifocal tensor. Given a point-line-line correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell}' \leftrightarrow \boldsymbol{\ell}''$, one can compute the entries of the vectors $\tilde{\boldsymbol{x}}$, $\widetilde{\boldsymbol{\ell}'}$ and $\widetilde{\boldsymbol{\ell}''}$ and use the multibody trilinear constraint (7) to obtain a linear relationship in the entries of $\mathcal{T}$. Therefore, we may estimate $\mathcal{T}$ linearly from $M_n^3 - 1$ point-line-line correspondences. That is 26 correspondences for one motion, 215 for two motions, 999 for three motions, etc.

[1] We refer the reader to [4] for further details and more robust linear methods for computing the trifocal tensor T.

Fortunately, as in the case of $n = 1$ motion, one may significantly reduce the data requirements by working with point-point-point correspondences $\boldsymbol{x} \leftrightarrow \boldsymbol{x}' \leftrightarrow \boldsymbol{x}''$. Since each point in the second view $\boldsymbol{x}'$ gives two lines $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ and each point in the third view $\boldsymbol{x}''$ gives two lines $\boldsymbol{\ell}''_1$ and $\boldsymbol{\ell}''_2$, a naive calculation would give $2^2 = 4$ constraints per correspondence. However, due to the algebraic properties of the polynomial embedding, each correspondence provides in general $(n+1)^2$ independent constraints on the multibody trifocal tensor. To see this, remember that the multibody trilinear constraint is satisfied by *all* lines $\boldsymbol{\ell}' = \boldsymbol{\ell}'_1 + \alpha\boldsymbol{\ell}'_2$ and $\boldsymbol{\ell}'' = \boldsymbol{\ell}''_1 + \beta\boldsymbol{\ell}''_2$ passing through $\boldsymbol{x}'$ and $\boldsymbol{x}''$, respectively. Therefore, for all $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ we must have

$$\prod_{i=1}^{n} \big(\boldsymbol{x}(\boldsymbol{\ell}'_1 + \alpha\boldsymbol{\ell}'_2)(\boldsymbol{\ell}''_1 + \beta\boldsymbol{\ell}''_2)T_i\big) = 0. \tag{8}$$

The above equation, viewed as a function of α, is a polynomial of degree n, hence its $n+1$ coefficients must be zero. Each coefficient is in turn a polynomial of degree n in β, whose $n+1$ coefficients must be zero. Therefore, each correspondence gives $(n+1)^2$ constraints on the multibody trifocal tensor $\mathcal{T}$, hence we need only $(M_n^3 - 1)/(n+1)^2$ point-point-point correspondences to estimate $\mathcal{T}$. That is 7, 24 and 63 correspondences for one, two and three motions, respectively. This represents a significant improvement not only with respect to the case of point-line-line correspondences, as explained above, but also with respect to the case of two perspective views which requires $M_n^2 - 1$ point-point correspondences for linearly estimating the multibody fundamental matrix [13], i.e. 8, 35 and 99 correspondences for one, two and three motions, respectively.

Given a correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{x}' \leftrightarrow \boldsymbol{x}''$, one may generate the $(n+1)^2$ linear equations in the entries of $\mathcal{T}$ by choosing $\boldsymbol{\ell}'_1$, $\boldsymbol{\ell}'_2$, $\boldsymbol{\ell}''_1$ and $\boldsymbol{\ell}''_2$ passing through $\boldsymbol{x}'$ and $\boldsymbol{x}''$, respectively, and then computing the coefficients of $\alpha^i\beta^j$ in (8). A simpler way is to choose at least $n+1$ distinct lines passing through each of $\boldsymbol{x}'$ and $\boldsymbol{x}''$ and generate the corresponding point-line-line equation. This leads to the following linear algorithm for estimating the multibody trifocal tensor.

Algorithm 1 (Estimating the multibody trifocal tensor $\mathcal{T}$) *Given $N \geq (M_n^3 - 1)/(n+1)^2$ point-point-point correspondences $\{\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i \leftrightarrow \boldsymbol{x}''_i\}_{i=1}^N$, with at least 7 correspondences per moving object, estimate $\mathcal{T}$ as follows:*

1. *Generate $N_\ell \geq (n+1)$ lines $\{\boldsymbol{\ell}'_{ij}\}_{j=1}^{N_\ell}$ and $\{\boldsymbol{\ell}''_{ik}\}_{k=1}^{N_\ell}$ passing through $\boldsymbol{x}'_i$ and $\boldsymbol{x}''_i$, respectively, for $i = 1..N$.*
2. *Compute $\mathcal{T}$, interpreted as a vector in $\mathbb{R}^{M_n^3}$, as the null vector of the matrix $A \in \mathbb{R}^{NN_\ell^2 \times M_n^3}$, whose rows are computed as $\widetilde{\boldsymbol{x}}_i \otimes \widetilde{\boldsymbol{\ell}'_{ij}} \otimes \widetilde{\boldsymbol{\ell}''_{ik}} \in \mathbb{R}^{M_n^3}$, for all $i = 1 \ldots N$ and $j, k = 1 \ldots N_\ell$, where $\otimes$ is the Kronecker product.*

Notice that Algorithm 1 is essentially the *same* as the linear algorithm for estimating the trifocal tensor T. The only differences are that we need to generate more than 2 lines per point in the second and third views $\boldsymbol{x}'$ and $\boldsymbol{x}''$, and that we need to replace the original correspondences $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell} \leftrightarrow \boldsymbol{\ell}'$ by the embedded correspondences $\tilde{\boldsymbol{x}} \leftrightarrow \widetilde{\boldsymbol{\ell}'} \leftrightarrow \widetilde{\boldsymbol{\ell}''}$ in order to build the data matrix A, whose null-space is the multibody trifocal tensor.

3. Motion Segmentation from 3 views

In this section, we present a linear algorithm for estimating and segmenting multiple rigid-body motions. More specifically, we assume we are given a set of point correspondences $\{\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j\}_{j=1}^N$, from which we can estimate the multibody trifocal tensor $\mathcal{T}$, and would like to estimate the individual trifocal tensors $\{T_i\}_{i=1}^n$ and/or the clustering of the correspondences according to the n motions.

3.1. From $\mathcal{T}$ to epipolar lines

Given the trifocal tensor T, it is well known how to compute the epipolar lines in the 2^{nd} and 3^{rd} views of a point $\boldsymbol{x}$ in the 1^{st} view [4]. Specifically, notice from (3) that the matrix

$$M_{\boldsymbol{x}} = (\boldsymbol{x}T) = (R'\boldsymbol{x}\boldsymbol{e}''^\top - \boldsymbol{e}'\boldsymbol{x}^\top R''^\top) \in \mathbb{R}^{3\times 3} \tag{9}$$

has rank 2. In fact its left null-space is $\boldsymbol{\ell}'_{\boldsymbol{x}} = \boldsymbol{e}' \times (R'\boldsymbol{x})$ and its right null-space is $\boldsymbol{\ell}''_{\boldsymbol{x}} = \boldsymbol{e}'' \times (R''\boldsymbol{x})$, i.e. the epipolar lines of $\boldsymbol{x}$ in the second and third views, respectively. In brief

Lemma 1 *The epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$ in the second view corresponding to a point $\boldsymbol{x}$ in the first view is the line such that $\boldsymbol{x}\boldsymbol{\ell}'_{\boldsymbol{x}}T = 0$. Similarly the epipolar line $\boldsymbol{\ell}''_{\boldsymbol{x}}$ in the third view is the line satisfying $\boldsymbol{x}\boldsymbol{\ell}''_{\boldsymbol{x}}T = 0$. Therefore,* $\text{rank}(\boldsymbol{x}T) = 2$.

In the case of multiple motions, we are faced with the more challenging problem of computing the epipolar lines $\boldsymbol{\ell}'_{\boldsymbol{x}}$ and $\boldsymbol{\ell}''_{\boldsymbol{x}}$ without knowing the individual trifocal tensors $\{T_i\}_{i=1}^n$ or the clustering of the correspondences. The question is then how to compute such epipolar lines from the multibody trifocal tensor $\mathcal{T}$. To this end, we notice that with each point in the first view $\boldsymbol{x}$ we can associate n epipolar lines $\{\boldsymbol{\ell}'_{i\boldsymbol{x}}\}_{i=1}^n$, each one of them corresponding to one of the n motions between the first and second views. We thus have $\boldsymbol{x}\boldsymbol{\ell}'_{i\boldsymbol{x}}T_i = 0$ which implies that for *any* line $\boldsymbol{\ell}''$ in the third view $\boldsymbol{x}\boldsymbol{\ell}'_{i\boldsymbol{x}}\boldsymbol{\ell}''T_i = 0$. Now, since the span of $\widetilde{\boldsymbol{\ell}''}$ for all $\boldsymbol{\ell}'' \in \mathbb{R}^3$ is $\mathbb{R}^{M_n}$, we have that for all $i = 1, \ldots, n$

$$\forall \boldsymbol{\ell}''\big[\prod_{k=1}^{n}(\boldsymbol{x}\boldsymbol{\ell}'_{i\boldsymbol{x}}\boldsymbol{\ell}''T_k) = (\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'_{i\boldsymbol{x}}}\widetilde{\boldsymbol{\ell}''}\mathcal{T}) = 0\big] \iff (\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'_{i\boldsymbol{x}}}\mathcal{T} = 0).$$

Since the vectors $\widetilde{\boldsymbol{\ell}'_{i\boldsymbol{x}}}$ are linearly independent when $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ are pairwise different in $\mathbb{P}^2$ (See [13]), we have the following:

Theorem 1 *If $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ and $\boldsymbol{\ell}''_{i\boldsymbol{x}}$ are the epipolar lines in the 2^{nd} and 3^{rd} views corresponding to a point $\boldsymbol{x}$ in the 1^{st} view according to the i^{th} motion, then $\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'_{i\boldsymbol{x}}}\mathcal{T} = \tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}''_{i\boldsymbol{x}}}\mathcal{T} = 0 \in \mathbb{R}^{M_n}$. Thus $\text{rank}(\tilde{\boldsymbol{x}}\mathcal{T}) \leq M_n - n$ if the epipolar lines are different.*

This result alone does not help us to find $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ according to a given motion, since any one of the n epipolar lines $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ will satisfy the above condition. This question of determining

the epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$ corresponding to a point $\boldsymbol{x}$ is not well posed as such, since the epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$ depends on which of the n motions the point $\boldsymbol{x}$ belongs to, which cannot be determined without additional information. We therefore pose the question a little differently, and suppose that we know the point $\boldsymbol{x}'$ in the second view corresponding to $\boldsymbol{x}$ and wish to find the epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$ also in the second view. This epipolar line must of course pass through $\boldsymbol{x}'$. To solve this problem, notice that $\boldsymbol{\ell}'_{\boldsymbol{x}}$ can be parameterized as

$$\boldsymbol{\ell}'_{\boldsymbol{x}} = \boldsymbol{\ell}'_1 + \alpha \boldsymbol{\ell}'_2, \tag{10}$$

where, as before, $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ are two different lines passing through $\boldsymbol{x}'$. From Theorem 1 we have that for some $\alpha \in \mathbb{R}$

$$\tilde{\boldsymbol{x}}(\widetilde{\boldsymbol{\ell}'_1 + \alpha \boldsymbol{\ell}'_2})\mathcal{T} = 0. \tag{11}$$

Each of the M_n components of this vector is a polynomial of degree n in α. These polynomials must have a common root α^* for which all the polynomials (and hence the vector) vanish. The epipolar line of $\boldsymbol{x}$ in the second view is then $\boldsymbol{\ell}'_{\boldsymbol{x}} = \boldsymbol{\ell}'_1 + \alpha^* \boldsymbol{\ell}'_2$. In practice, we do not need to consider all the M_n polynomials, but can instead find the common root of random linear combinations of these polynomials. We therefore have the following algorithm for computing epipolar lines from the multibody fundamental tensor.

Algorithm 2 (Estimating epipolar lines from $\mathcal{T}$) *Given a point-point-point correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{x}' \leftrightarrow \boldsymbol{x}''$,*

1. *Choose two different lines $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ passing through $\boldsymbol{x}'$. Choose $N_\ell \geq 2$ vectors $\{\boldsymbol{w}''_k \in \mathbb{R}^{M_n}\}_{k=1}^{N_\ell}$ and build the polynomials $q'_k(\alpha) = \tilde{\boldsymbol{x}}(\widetilde{\boldsymbol{\ell}'_1 + \alpha \boldsymbol{\ell}'_2})\boldsymbol{w}''_k\mathcal{T}$, for $k = 1, \ldots, N_\ell$. Compute the common root α^* of these N_ℓ polynomials as the root of $q'(\alpha) = \sum_{k=1}^{N_\ell} q'_k(\alpha)^2$ that minimizes $q'(\alpha)$. The epipolar line of $\boldsymbol{x}$ in the second view is given by $\boldsymbol{\ell}'_{\boldsymbol{x}} = \boldsymbol{\ell}'_1 + \alpha^* \boldsymbol{\ell}'_2$.*
2. *Determine the epipolar line of $\boldsymbol{x}$ in the third view, $\boldsymbol{\ell}''_{\boldsymbol{x}}$, in an entirely analogous way.*

We may apply the above process to all correspondences $\{\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j\}_{j=1}^N$ and obtain the set of all N epipolar lines in the second and third views according to the motion associated with each correspondence. Notice, again, that this is done from the multibody trifocal tensor only, without knowing the individual trifocal tensors or the clustering of the correspondences.

It is also useful to note that the only property of $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ that we used in the above algorithm was that the desired epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$ could be expressed as a linear combination of $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$. If instead we knew the epipoles corresponding to the required motion, then we could choose $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ to be any two lines passing through the epipole and apply Algorithm 2 to determine the epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$.

Observe therefore that, once we know the set of epipoles corresponding to the n motions, we may compute the epipolar lines corresponding to any point $\boldsymbol{x}$ in the first image. Consequently, we can determine the individual fundamental matrices and the trifocal tensors, as we will see in Section 3.3. Before proceeding, we need to show how to determine the epipoles, which we do in the next section.

3.2. From $\mathcal{T}$ to epipoles

In the case of one rigid-body motion, the epipoles in the second and third views $\boldsymbol{e}'$ and $\boldsymbol{e}''$ must lie on the epipolar lines in the second and third views, $\{\boldsymbol{\ell}'_{\boldsymbol{x}_j}\}_{j=1}^N$ and $\{\boldsymbol{\ell}''_{\boldsymbol{x}_j}\}_{j=1}^N$, respectively. Thus we can obtain the epipoles from

$$\boldsymbol{e}'^\top [\boldsymbol{\ell}'_{\boldsymbol{x}_1}, \ldots, \boldsymbol{\ell}'_{\boldsymbol{x}_N}] = 0 \text{ and } \boldsymbol{e}''^\top [\boldsymbol{\ell}''_{\boldsymbol{x}_1}, \ldots, \boldsymbol{\ell}''_{\boldsymbol{x}_N}] = 0. \tag{12}$$

Clearly, we only need 2 epipolar lines to determine the epipoles, hence we do not need to compute the epipolar lines for all points in the first view. However, it is better to use more than two lines in the presence of noise.

In the case of n motions there exist n epipole pairs, $\{(\boldsymbol{e}'_i, \boldsymbol{e}''_i)\}_{i=1}^n$, where $\boldsymbol{e}'_i$ and $\boldsymbol{e}''_i$ are epipoles in the second and third views corresponding to the i^{th} motion. Given a set of correspondences $\{\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j\}$ we may compute the multibody trifocal tensor $\mathcal{T}$ and determine the epipolar lines $\boldsymbol{\ell}'_{\boldsymbol{x}_j}$ and $\boldsymbol{\ell}''_{\boldsymbol{x}_j}$ associated with each correspondence $\{\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j\}$ by the method described in Section 3.1. Then, for each pair of epipolar lines $(\boldsymbol{\ell}'_{\boldsymbol{x}_j}, \boldsymbol{\ell}''_{\boldsymbol{x}_j})$ there exists an epipole pair $(\boldsymbol{e}'_i, \boldsymbol{e}''_i)$ such that

$$\boldsymbol{e}'^\top_i \boldsymbol{\ell}'_{\boldsymbol{x}_j} = 0 \quad \text{and} \quad \boldsymbol{e}''^\top_i \boldsymbol{\ell}''_{\boldsymbol{x}_j} = 0. \tag{13}$$

Our task is two-fold. First, we need to find the set of epipole pairs $\{(\boldsymbol{e}'_i, \boldsymbol{e}''_i)\}$. Second, we need to determine which pair of epipoles lie on the epipolar lines $(\boldsymbol{\ell}'_{\boldsymbol{x}_j}, \boldsymbol{\ell}''_{\boldsymbol{x}_j})$ derived from a given point correspondence.

If two point correspondences $\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j$ and $\boldsymbol{x}_k \leftrightarrow \boldsymbol{x}'_k \leftrightarrow \boldsymbol{x}''_k$ both belong to the same motion, then the pair of epipoles can be determined easily by intersecting the epipolar lines. If the two motions are different, then the intersection points of the epipolar lines will have no geometric meaning, and will be essentially arbitrary. This suggests an approach to determining the epipoles based on RANSAC [3] in which we intersect pairs of epipolar lines to find candidate epipoles, and determine their degree of support among the other point correspondences. This method is expected to be effective with small numbers of motions.

In reality, we used a different method based on the idea of *multibody epipoles* proposed in [13] for the case of two views, which we now extend and modify for the case of three views. Notice from (13) that, regardless of the motion associated with each pair of epipolar lines, we must have

$$\prod_{i=1}^{n} (\boldsymbol{e}'^\top_i \boldsymbol{\ell}'_{\boldsymbol{x}}) = \boldsymbol{c}'^\top \widetilde{\boldsymbol{\ell}'_{\boldsymbol{x}}} = 0, \quad \prod_{i=1}^{n} (\boldsymbol{e}''^\top_i \boldsymbol{\ell}''_{\boldsymbol{x}}) = \boldsymbol{c}''^\top \widetilde{\boldsymbol{\ell}''_{\boldsymbol{x}}} = 0,$$

where the *multibody epipoles* $\boldsymbol{c}' \in \mathbb{R}^{M_n}$ and $\boldsymbol{c}'' \in \mathbb{R}^{M_n}$ are the coefficients of the homogeneous polynomials of degree n

$$p'(\boldsymbol{\ell}'_{\boldsymbol{x}}) = \boldsymbol{c}'^{\top}\widetilde{\boldsymbol{\ell}'_{\boldsymbol{x}}} \quad \text{and} \quad p''(\boldsymbol{\ell}''_{\boldsymbol{x}}) = \boldsymbol{c}''^{\top}\widetilde{\boldsymbol{\ell}''_{\boldsymbol{x}}},$$

respectively. Similarly to (12), we may obtain the multibody epipoles from

$$\boldsymbol{c}'^{\top}[\widetilde{\boldsymbol{\ell}'_{\boldsymbol{x}_1}}, \ldots, \widetilde{\boldsymbol{\ell}'_{\boldsymbol{x}_N}}] = 0 \text{ and } \boldsymbol{c}''^{\top}[\widetilde{\boldsymbol{\ell}''_{\boldsymbol{x}_1}}, \ldots, \widetilde{\boldsymbol{\ell}''_{\boldsymbol{x}_N}}] = 0. \quad (14)$$

In order to estimate the epipoles, we notice that if a pair of epipolar lines $(\boldsymbol{\ell}'_{\boldsymbol{x}}, \boldsymbol{\ell}''_{\boldsymbol{x}})$ corresponds to the i^{th} motion, then the derivatives of p' and p'' at the pair $(\boldsymbol{\ell}_{\boldsymbol{x}'}, \boldsymbol{\ell}''_{\boldsymbol{x}})$ give the epipoles $\boldsymbol{e}'_i$ and $\boldsymbol{e}''_i$, i.e.

$$\frac{\partial}{\partial \boldsymbol{\ell}'_{\boldsymbol{x}}}(\boldsymbol{c}'^{\top}\widetilde{\boldsymbol{\ell}'_{\boldsymbol{x}}}) \sim \boldsymbol{e}'_i \quad \text{and} \quad \frac{\partial}{\partial \boldsymbol{\ell}''_{\boldsymbol{x}}}(\boldsymbol{c}''^{\top}\widetilde{\boldsymbol{\ell}''_{\boldsymbol{x}}}) \sim \boldsymbol{e}''_i. \quad (15)$$

Remark 1 (Computing derivatives) *Note that given $\mathbf{c}$ the computation of the derivatives of $p(\boldsymbol{\ell}) = \boldsymbol{c}^T\widetilde{\boldsymbol{\ell}}$ can be done algebraically, i.e. it does not involve taking derivatives of the (possibly noisy) data. For instance, one may compute $Dp(\boldsymbol{\ell})$ as $\boldsymbol{c}^T D\widetilde{\boldsymbol{\ell}} = \boldsymbol{c}^T E_n \widetilde{\boldsymbol{\ell}}^{n-1}$, where $E_n \in \mathbb{R}^{M_n \times M_{n-1}}$ is a constant matrix containing the exponents of $\widetilde{\boldsymbol{\ell}} \in \mathbb{R}^{M_n}$ and $\widetilde{\boldsymbol{\ell}}^{n-1} \in \mathbb{R}^{M_{n-1}}$ contains all monomials of degree $n-1$ in $\boldsymbol{\ell}$.*

In the case of noise-free correspondences, this means that we can immediately obtain the epipoles by evaluating the derivatives of p' and p'' at different epipolar lines. Then epipolar lines belonging to the same motion will give the same epipoles, hence we can automatically cluster all the correspondences. However, with noisy correspondences the derivatives of p' and p'' will not be equal for two pairs of epipolar lines corresponding to the same motion. Instead, we may use (15) to compute the (unit) epipoles $(\boldsymbol{e}'_{\boldsymbol{x}_j}, \boldsymbol{e}''_{\boldsymbol{x}_j})$ and $(\boldsymbol{e}'_{\boldsymbol{x}_k}, \boldsymbol{e}''_{\boldsymbol{x}_k})$ from the derivatives of (p', p'') at $(\boldsymbol{\ell}'_{\boldsymbol{x}_j}, \boldsymbol{\ell}''_{\boldsymbol{x}_j})$ and $(\boldsymbol{\ell}'_{\boldsymbol{x}_k}, \boldsymbol{\ell}''_{\boldsymbol{x}_k})$, respectively. Then the similarity measure

$$S_{jk} = \frac{1}{2}\left(\left|\boldsymbol{e}'^{\top}_{\boldsymbol{x}_j}\boldsymbol{e}'_{\boldsymbol{x}_k}\right| + \left|\boldsymbol{e}''^{\top}_{\boldsymbol{x}_j}\boldsymbol{e}''_{\boldsymbol{x}_k}\right|\right) \quad (16)$$

is approximately one for points j and k in the same group and strictly less than one for points in different groups. Given the so-defined similarity matrix $S \in \mathbb{R}^{N \times N}$, one can apply any spectral clustering technique to obtain the clustering of the correspondences. Then, one can obtain the epipoles, fundamental matrices and camera matrices from the correspondences associated with each one of the n groups. We therefore have the following algorithm for computing the epipoles and clustering the correspondences.

Algorithm 3 (Estimating epipoles from $\mathcal{T}$) *Given a set of epipolar lines $\{(\boldsymbol{\ell}'_{\boldsymbol{x}_j}, \boldsymbol{\ell}''_{\boldsymbol{x}_j})\}_{j=1}^N$,*

1. *Compute the multibody epipoles $\mathbf{c}'$ and $\mathbf{c}''$ from (14).*
2. *Compute the epipole at each epipolar line from the derivatives of the polynomials p' and p'' as in (15).*
3. *Define a pairwise similarity matrix as in (16) and apply spectral clustering to segment the epipolar lines, hence the original point correspondences.*
4. *Compute the epipoles $\mathbf{e}'_i$ and $\mathbf{e}''_i$, $i = 1, \ldots, n$, for each one of the n groups of epipolar lines as in (12).*

3.3. From $\mathcal{T}$ to trifocal tensors

The algorithm for motion segmentation that we have proposed so far computes the motion parameters (trifocal tensors, camera matrices and fundamental matrices) by first clustering the image correspondences using the geometric information provided by epipoles and epipolar lines. In this section, we demonstrate that one can estimate the individual trifocal tensors *without* first clustering the image correspondences. The key is to look at second order derivatives of the multibody trilinear constraint. Therefore, we contend that *all* the geometric information about the multiple motions is already encoded in the multibody trifocal tensor.

Let $\boldsymbol{x}$ be an arbitrary point in $\mathbb{P}^2$ (not necessarily a point in the first view). Since the i^{th} epipole $\boldsymbol{e}'_i$ is known, we can compute two lines $\boldsymbol{\ell}'_{i1}$ and $\boldsymbol{\ell}'_{i2}$ passing through $\boldsymbol{e}'_i$ and apply Algorithm 2 to compute the epipolar line of $\boldsymbol{x}$ in the second view $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ according to the i^{th} motion. In a completely analogous fashion, we can compute the epipolar line of $\boldsymbol{x}$ in the third view $\boldsymbol{\ell}''_{i\boldsymbol{x}}$ from two lines passing through $\boldsymbol{e}''_i$. Given $(\boldsymbol{\ell}'_{i\boldsymbol{x}}, \boldsymbol{\ell}''_{i\boldsymbol{x}})$, a simple calculation shows that the slices of the trifocal tensor T_i can be expressed in terms of the second derivative of the multibody epipolar constraint, as follows:

$$\left.\frac{\partial^2(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T})}{\partial \boldsymbol{\ell}' \partial \boldsymbol{\ell}''}\right|_{(\boldsymbol{x}, \boldsymbol{\ell}'_{i\boldsymbol{x}}, \boldsymbol{\ell}''_{i\boldsymbol{x}})} = M_{i\boldsymbol{x}} \sim \boldsymbol{x}T_i \in \mathbb{R}^{3\times 3}. \quad (17)$$

Thanks to (17), we can immediately outline an algorithm for computing the individual trifocal tensors.

Algorithm 4 (Estimating trifocal tensors from $\mathcal{T}$) *Let $\{\boldsymbol{e}'_i, \boldsymbol{e}''_i\}_{i=1}^n$ be the set of epipoles in the 2^{nd} and 3^{rd} views. Also let $\{\boldsymbol{x}_j\}_{j=1}^N$ be a set of $N \geq 4$ randomly chosen points.*

1. *Use Algorithm 2 to obtain the epipolar lines of $\boldsymbol{x}_j$ in the second and third views $\boldsymbol{\ell}'_{i\boldsymbol{x}_j}$ and $\boldsymbol{\ell}'_{i\boldsymbol{x}_j}$ from the epipoles $\mathbf{e}'_i$ and $\mathbf{e}''_i$, respectively.*
2. *Use (17) to obtain $M_{i\boldsymbol{x}_j}$, the slice of T_i along $\boldsymbol{x}_j$.*
3. *Solve for T_i for $i = 1, \ldots, n$ from the set of linear equations*

$$M_{i\boldsymbol{x}_j} \sim \boldsymbol{x}_j T_i \quad j = 1, \ldots, N.$$

Once the individual trifocal tensors have been computed, one may cluster the correspondences by assigning each feature to the trifocal tensor T_i which minimizes the Sampson error. Alternatively, one may first reconstruct the 3-D structure by triangulation, project those 3-D points onto the three views, and then assign points to the trifocal tensor T_i that minimizes the reprojection error. We refer the reader to [4] for details of the computation of both errors.

3.4. Iterative refinement by EM

The motion segmentation algorithm we have proposed so far is purely geometric and provably correct in the absence of noise. Since most of the steps of the algorithm involve

solving linear systems, the algorithm will also work with a moderate level of noise (as we will show in the experiments) provided that one solves each step in a least-squares fashion.

However, in order to obtain an optimal estimate for the trifocal tensors and the clustering of the correspondences in the presence of noise, we assume a generative model in which the probability of a point belonging to the i-th motion model is given by π_i, and the correspondences are corrupted with zero-mean Gaussian noise with variance σ_i^2 for the i-th model. We introduce latent variables w_{ij} representing a soft assignment of point j to motion model i, and maximize the expected complete log-likelihood

$$\sum_{j=1}^{N}\sum_{i=1}^{n} w_{ij}\left(\log(\frac{\pi_i}{\sigma_i}) - \frac{\epsilon_{ij}}{2\sigma_i^2}\right) \tag{18}$$

with respect to the w_{ij} and parameters $\theta = \{(T_i, \sigma_i, \pi_i)\}_{i=1}^{n}$ given the data $X = \{(\boldsymbol{x}_j, \boldsymbol{x}'_j, \boldsymbol{x}''_j)\}_{j=1}^{N}$. Maximation is carried out using the Expectation-Maximization (EM) algorithm. The expected complete log-likelihood (18) is maximized with respect to the w_{ij} in the E-step, and with respect to parameters θ in the M-step. In the M-step, computation of each T_i simply involves using standard structure-from-motion algorithms with correspondences weighted by w_{ij}. The EM algorithm proceeds by iterating between the E and M steps, until the estimates converge to a local maximum.

4. Experiments

In our experiments, we consider the following algorithms:

1. *Algebraic I*: this algorithm clusters the correspondences using epipoles and epipolar lines computed from the multibody trifocal tensor, as in Algorithms 1-3.
2. *Algebraic II*: this algorithm clusters the correspondences using the Sampson-distance residual to the different trifocal tensors computed as in Algorithms 1-4.
3. *K-means*: this algorithm alternates between computing (linearly) the trifocal tensors for different motion classes and clustering the point correspondences using the Sampson-distance residual to the different motions.
4. *EM*: This algorithm refines the classification and the motion parameters as described in Section 3.4. For ease of computation, in the M-step we first compute each trifocal tensor linearly as in Section 2.1. If the error (18) increases, we recompute the trifocal tensors using the linear algebraic algorithm in [4]. If the error (18) still increases, then we use Levenberg-Marquardt to solve for the trifocal tensors optimally.

Figure 1 shows three views of the Tshirt-Book-Can sequence which has two rigid-body motions, the camera and the can, for which we manually extracted a total of $N =$ 140 correspondences, 70 per motion. Figure 1 also shows the relative displacement of the correspondences between pairs of frames. We first run 1000 trials of the K-means algorithm starting from different random classifications. On average, the K-means algorithm needs 39 iterations (maximum was set to 50) to converge and yields a misclassification error of about 24.6%, as shown in Table 1. The (non-iterative) algebraic algorithms I and II, on the other hand, give a misclassification error of 24.3% and 23.6%. Running the K-means algorithm starting from the clustering produced by the second algebraic algorithm resulted in convergence after 3 iterations to a misclassification error of 7.1%. Finally, after 10 iterations of the EM algorithm, the misclassification error reduced to 1.4%.

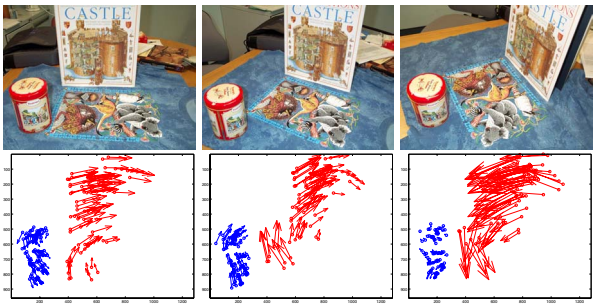

Figure 1: *Top: views 1-3 of a sequence with two rigid-body motions. Bottom: 2D displacements of the 140 correspondences from the current view ('o') to the next ('→').*

Table 1: *Percentage of misclassification of each algorithm.*

	K-means	Alg. I	Alg. II	Alg. II + K-means	Alg. II + K-means+EM
Tshirt-Book-Can	24.6%	24.3%	23.6%	7.1%	1.4%
Wilshire	39.5%	4.1%	2.5%	2.5%	0.0%

We also tested the performance of our algorithm on a sequence with transparent motions, so that the only cue for clustering the correspondences is the motion cue. Given correspondences in a sequence with one rigid-body motion, we generated a second set of correspondences by flipping the x and y coordinates of the first set of correspondences. In this way, we obtain a set of correspondences with no spatial separation and undergoing two different rigid-body motions. Figure 2 shows frames 1, 4 and 7 of the Wilshire sequence and the inter-frame displacement of the $N = 164 \times 2 = 328$ correspondences. As shown in Table 1, the K-means algorithm gives a mean misclassification error (over 1000 trials) of 35.5% with a mean number of iterations of 47.1. The algebraic algorithms I and II give an error of 4.1% and 2.5%. Following Algebraic II with K-means did not improve the classification, while following this with EM achieved a perfect segmentation.

We also tested our algorithm on synthetic data. We randomly generated two groups of 100 3-D points each with

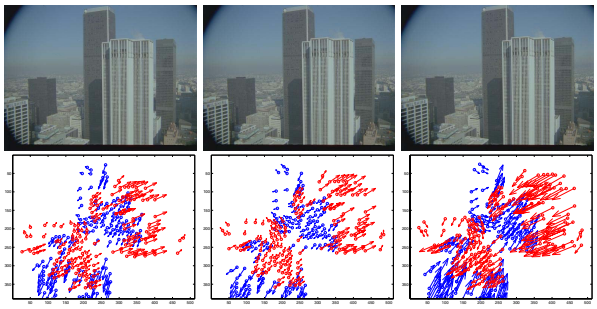

Figure 2: *Top: frames 1, 4 and 7 of the Wilshire sequence. Bottom: 2D displacement of the 328 original and flipped correspondences from current view ('$\circ$') to the next ('$\rightarrow$').*

a depth variation 100-400 units of focal length (u.f.l.). The two motions were chosen at random with an inter-frame rotation of 5° and an inter-frame translation of 30 u.f.l. We added zero-mean Gaussian noise with standard deviation between 0 and 1 pixel in an image size of 1000×1000. Figure 3 shows the percentage of misclassified correspondences and the error in the estimation of the epipoles (degrees) over 100 trials. The K-means algorithm usually converges to a local minimum due to bad initialization. The algebraic algorithms (I and II) achieve a misclassification ratio of about 20.2% and 9.1% and a rotation error of 22.4° and 11.7°, respectively, for 1 pixel noise. These errors are reduced to about 2.9% and 3.8°, respectively, by the K-means algorithm and to 2.4% and 2.8°, respectively by the EM algorithm. This is expected, as the algebraic algorithms do not enforce the nonlinear algebraic structure of the multibody trifocal tensors. The K-means algorithm improves the estimates by iteratively clustering the correspondences using the trifocal tensors. The EM algorithm further improves the estimates in a probabilistic fashion, at the expense of a higher computational cost.

5. Conclusions

The multibody trifocal tensor is effective in the analysis of dynamic scenes involving several moving objects. The algebraic method of motion classification involves computation of the multibody tensor, computation of the epipoles for different motions and classification of the points according to the compatibility of epipolar lines with the different epipoles. Our reported implementation of this algorithm was sufficiently good to provide an initial classification of points into different motion classes. This classification can be refined using a K-means or EM algorithm with excellent results. It is likely that more careful methods of computing the tensor (analogous with best methods for the single-body trifocal tensor) could give a better initialization.

The algebraic properties of the multibody trifocal tensor are in many respects analogous to those of the single-body tensor, but provide many surprises and avenues of research that we have not yet exhausted.

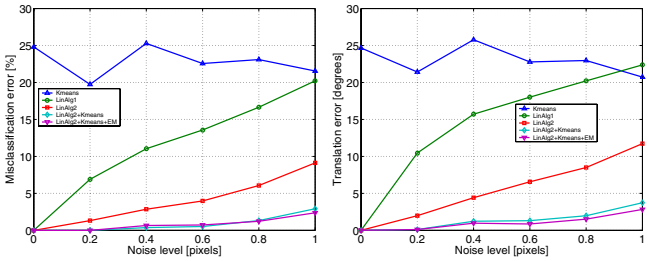

Figure 3: *Motion segmentation and motion estimation (rotation) errors as a function of noise.*

References

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

[2] X. Feng and P. Perona. Scene segmentation from 3D motion, In *CVPR*, pages 225-231, 1998.

[3] M. Fischler and R. Bolles. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communic. of the ACM*, 26:381–395, 1981.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2000.

[5] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, volume 2, pages 586–591, 2001.

[6] Y. Ma, Kun Huang, R. Vidal, J. Kosecká, and S. Sastry. Rank conditions on the multiple view matrix. *IJCV*, 2004.

[7] A. Shashua and L. Wolf. Homography tensors: on algebraic entities that represent 3 views of static or moving planar points. *ECCV*, 2000.

[8] P. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *IEEE Trans. on PAMI*, 23(3):297–303, 2001.

[9] P. H. S. Torr. Geometric motion segmentation and model selection. *Phil. Trans. Royal Society of London A*, 356(1740):1321–1340, 1998.

[10] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *CVPR*, pages 621–628, 2003.

[11] R. Vidal and Y. Ma. A unified algebraic approach to 2-D and 3-D motion segmentation. In *ECCV*, 2004.

[12] R. Vidal and R. Hartley. Motion segmentation with missing data using PowerFactorization and GPCA. In *CVPR*, 2004.

[13] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *IJCV*, 2004.

[14] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *CVPR*, pages 263–270, 2001.