The Multibody Trifocal Tensor: Motion Segmentation from 3 Perspective Views

PAPER ID #583

Abstract

We propose a new 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 $\mathcal{T}$. We first show how to linearly estimate $\mathcal{T}$ from point-point-point correspondences. Given $\mathcal{T}$, we show that the derivatives of the multibody trilinear constraint at a correspondence enable us to transfer points and lines from two views to the other. 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 $\mathbb{R}^3$. 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, one can immediately obtain an initial clustering of the correspondences. We use this initial clustering to initialize an iterative algorithm that finds an optimal estimate for the trifocal tensors and the segmentation of the correspondences using Expectation Maximization. We test our algorithm on various 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 we 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 3D motion segmentation [8] has addressed the problem using the standard probabilistic framework. Given an initial clustering of the image data, one estimates a motion model for each group using standard structure from motion algorithms [4]. Given the motion parameters, one can easily update the clustering of the image data. The algorithm then proceeds by iterating between these two steps, using the Expectation Maximization algorithm [2]. When the probabilistic model generating the data is known, the above iterative scheme indeed provides an optimal estimate in the maximum likelihood sense. Unfortunately, it is well-known that the EM algorithm is very sensitive to initialization [7].

In order to deal with the initialization problem, recent work on 3D motion segmentation has concentrated on the study of the geometry of multiple motion models. [11] derived a bilinear constraint in $\mathbb{R}^6$ which, together with a combinatorial scheme, segments two rigid-body motions from two perspective views. [10] 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. The algorithms of [11] and [10] are algebraic, hence they do not require initialization.

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 show that one can use the multibody trifocal tensor to transfer points and lines from two views to the other. In Section 4, we propose a geometric algorithm for 3D 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 work on multiframe 3D motion segmentation is for the case of affine cameras [1, 5], and requires a minimum of four views.

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 (3) 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 feature segmentation stage from the motion segmentation problem by taking the product of all trilinear constraints. Although taking products 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 (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 27 correspondences for one motion, 215 for two motions, 999 for three motions, etc.

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

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

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

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

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 only 7 correspondences for one motion, 24 for two motions, 63 for three motions, etc. 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 [10], 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 $i = 1 \dots N$ and $j, k = 1 \dots N_\ell$, 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. Multibody transfer properties of $\mathcal{T}$

An important property of the trifocal tensor T is that of transferring points and lines from a pair of views to the other [4]. For example, if $\boldsymbol{\ell}'$ and $\boldsymbol{\ell}''$ are corresponding lines in the second and third views, then $\boldsymbol{\ell} = \boldsymbol{\ell}'\boldsymbol{\ell}''T$ is a corresponding line in the first view. Similarly, if $\boldsymbol{x}$ is a point in the first view and $\boldsymbol{\ell}'$ is a corresponding line in the second view, then $\boldsymbol{x}'' = \boldsymbol{x}\boldsymbol{\ell}'T$ is the corresponding point in the third view. Likewise, $\boldsymbol{x}' = \boldsymbol{x}\boldsymbol{\ell}''T$ is the point in the second view corresponding to $(\boldsymbol{x}, \boldsymbol{\ell}'')$.

In this section, we discuss the transfer properties of the multibody trifocal tensor $\mathcal{T}$. While in principle these properties are natural generalizations of the corresponding properties of the individual trifocal tensors $\{T_i\}_{i=1}^n$, in the multibody case the situation is more complex, because $\mathcal{T}$ incorporates information about *all* the motions at the same time. In order to obtain information about a specific motion, say the i^{th} motion, without yet knowing its trifocal tensor T_i, we exploit the algebraic properties of the multibody trilinear constraint. In particular, we show that information about individual motions is encoded in its derivatives.

We begin by considering the derivative of the multibody trilinear constraint with respect to its first argument

$$\frac{\partial}{\partial \boldsymbol{x}}(\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T}) = \frac{\partial}{\partial \boldsymbol{x}}\prod_{i=1}^{n}(\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_i) = \sum_{i=1}^{n}(\boldsymbol{\ell}'\boldsymbol{\ell}''T_i)\prod_{k\neq i}(\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_k).$$

We notice that if we evaluate this derivative at a correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{\ell}' \leftrightarrow \boldsymbol{\ell}''$ associated with the i^{th} motion, i.e. the correspondence is such that $\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_i = 0$, then all the terms in the above summation but the i^{th} vanish. Thus we obtain

$$\frac{\partial}{\partial \boldsymbol{x}}(\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T})\bigg|_{\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_i=0} = (\boldsymbol{\ell}'\boldsymbol{\ell}''T_i)\prod_{k\neq i}(\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_k) \sim (\boldsymbol{\ell}'\boldsymbol{\ell}''T_i),$$

which from the properties of the trifocal tensor T_i gives a line $\boldsymbol{\ell}$ in the first view. Notice that this line $\boldsymbol{\ell}$ in the first view is *transferred* from the two lines in the second and third views according to the *unknown* i^{th} trifocal tensor T_i. That is, the multibody trifocal tensor enables us to transfer corresponding lines according to their own motion, without having to know the motion with which the correspondence is associated. In a similar fashion, one may obtain point transfer properties, by considering derivatives of the multibody trilinear constraint with respect to its second and third arguments. We therefore have the following results.

Theorem 1 (Line transfer from corresponding lines in the second and third views to the first) *The derivative of the multibody trilinear constraint with respect to its first argument evaluated at a correspondence $(\boldsymbol{x}, \boldsymbol{\ell}', \boldsymbol{\ell}'')$ gives a line $\boldsymbol{\ell}$ in the first view passing though $\boldsymbol{x}$, i.e.*

$$\boldsymbol{\ell} = \frac{\partial}{\partial \boldsymbol{x}}(\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T}) \quad \textit{and} \quad \boldsymbol{\ell}^\top\boldsymbol{x} = 0.$$

Theorem 2 (Point transfer from first to second and third views) *The derivative of the multibody trilinear constraint with respect to its second and third arguments evaluated at a correspondence $(\boldsymbol{x}, \boldsymbol{\ell}', \boldsymbol{\ell}'')$ gives the corresponding point in the second and third view $\boldsymbol{x}'$ and $\boldsymbol{x}''$, respectively, i.e.*

$$\frac{\partial}{\partial \boldsymbol{\ell}'}(\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T}) \sim \boldsymbol{x}' \textit{ and } \frac{\partial}{\partial \boldsymbol{\ell}''}(\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T}) \sim \boldsymbol{x}''.$$

4. 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 segmentation of the correspondences according to the n motions.

Our algorithm proceeds as follows. In Section 4.1 we show how to the estimate epipolar lines in the second and third views, $\boldsymbol{\ell}'_{\boldsymbol{x}}$ and $\boldsymbol{\ell}''_{\boldsymbol{x}}$, respectively, associated with each point $\boldsymbol{x}$ in the first view by solving for the common root of a set of univariate polynomials. In Section 4.2 we show how to estimate the epipoles in the second and third views, $\{\boldsymbol{e}'_i\}_{i=1}^n$ and $\{\boldsymbol{e}''_i\}_{i=1}^n$, respectively, by solving a plane clustering problem using a combination of Generalized Principal Component Analysis (GPCA) with spectral clustering. Given epipolar lines and epipoles one may immediately cluster the correspondences into n groups and then estimate individual trifocal tensors and camera matrices from the data associated with each group. However, we also show in Section 4.3 that one may recover the individual trifocal tensors directly from the second order derivatives of the multibody trilinear constraint. In Section 4.4 we show how to refine the estimates of the linear algorithm by applying the EM algorithm to a mixture of trifocal tensors model.

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

Given the trifocal tensor T, it is well known how to compute the epipolar lines in the second and third views of a point $\boldsymbol{x}$ in the first view [4]. Specifically, notice that the matrix

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

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,* $\operatorname{rank}(\boldsymbol{x}T) = 2$.

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 segmentation 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 each one of the n motions between the first and second views. One of such n epipolar lines, $\boldsymbol{\ell}'_{\boldsymbol{x}}$, is *the true* epipolar line corresponding to $\boldsymbol{x}$ according to its own motion. 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 $i = 1, \ldots, n$

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

We have shown the following result.

Theorem 3 *If $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ and $\boldsymbol{\ell}''_{i\boldsymbol{x}}$ are the epipolar lines in the second and third views corresponding to a point $\boldsymbol{x}$ in the first view according to the i^{th} motion, then $\tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'_{i\boldsymbol{x}}}\mathcal{T} = \tilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}''_{i\boldsymbol{x}}}\mathcal{T} = 0 \in \mathbb{R}^{M_n}$. Therefore,* $\operatorname{rank}(\tilde{\boldsymbol{x}}\mathcal{T}) \leq M_n - n$.

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, we begin by noticing that $\boldsymbol{\ell}'_{\boldsymbol{x}}$ can be parameterized as

$$\boldsymbol{\ell}'_{\boldsymbol{x}} = \boldsymbol{\ell}'_1 + \alpha\boldsymbol{\ell}'_2 \qquad (10)$$

where, as before, $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ are two different lines passing through $\boldsymbol{x}'$. From Theorem 3 we have that

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

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) vanishes. 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 $\boldsymbol{x}$ in the first view,*

1. *Choose two different lines $\boldsymbol{\ell}'_1$ and $\boldsymbol{\ell}'_2$ passing through its corresponding point $\boldsymbol{x}'$ in the second view. Choose $N_\ell \geq 2$ vectors $\{\boldsymbol{w}''_k \in \mathbb{R}^{M_n}\}_{k=1}^{N_\ell}$ and build the polynomials $q'_k(\alpha) = \tilde{\boldsymbol{x}}(\widetilde{\boldsymbol{\ell}'_1 + \alpha\boldsymbol{\ell}'_2})\boldsymbol{w}''_k\mathcal{T}$, $k = 1, \ldots, N_\ell$. 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. *Given a correspondence $\boldsymbol{x} \leftrightarrow \boldsymbol{x}''$, determine its epipolar line 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 segmentation 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. Algorithm 2 could then be used to determine the epipolar line $\boldsymbol{\ell}'_{\boldsymbol{x}}$.

Observe therefore, that once we know the set of epipoles corresponding to the n motion, we may compute the epipolar lines corresponding to any point $\boldsymbol{x}$ in the first image. Consequently, we can determine the fundamental matrices and, as we will see in Section 4.3, the individual trifocal tensors. Before proceeding, we need to show how the epipoles may be determined, which we do in the next section.

4.2. 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

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

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. Now, given a set of correspondences $\{\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j\}$ we may compute the multibody trifocal tensor, and then for each correspondence $\{\boldsymbol{x}_j \leftrightarrow \boldsymbol{x}'_j \leftrightarrow \boldsymbol{x}''_j\}$ determine its epipolar lines $\boldsymbol{\ell}'_{\boldsymbol{x}_j}$ and $\boldsymbol{\ell}''_{\boldsymbol{x}_j}$ by the method described in Section 4.1. 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

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

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 from 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 [10] 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 $p'(\boldsymbol{\ell}'_{\boldsymbol{x}}) = \boldsymbol{c}'^\top\widetilde{\boldsymbol{\ell}'_{\boldsymbol{x}}}$ and $p''(\boldsymbol{\ell}''_{\boldsymbol{x}}) = \boldsymbol{c}''^\top\widetilde{\boldsymbol{\ell}''_{\boldsymbol{x}}}$, 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, similarly to our results in Section 3, we notice that if the 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 $(\boldsymbol{\ell}'_x,\boldsymbol{\ell}''_{\boldsymbol{x}})$ give the epipoles $\boldsymbol{e}'_i$ and $\boldsymbol{e}''_i$, i.e.

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

In the noise free case, 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. With noisy measurements, however, the derivatives of p' and p'' will not be equal for two pairs of epipolar lines corresponding to the same motion. However, 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 segmentation of the correspondences, and then the epipoles, fundamental matrices and camera matrices for each group. 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* $\boldsymbol{c}'$ *and* $\boldsymbol{c}''$ *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* $\boldsymbol{e}'_i$ *and* $\boldsymbol{e}''_i$, $i = 1,\ldots,n$, *for each one of the* n *groups of epipolar lines as in (12).*

4.3. 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 $\mathcal{P}^2$ (not necessarily a point in the first view). Also let $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ and $\boldsymbol{\ell}''_{i\boldsymbol{x}}$ be its corresponding epipolar lines in the second and third views according to the i^{th} motion. Then, we have the following result.

Theorem 4 (Slices of the trifocal tensors from second order derivatives of the multibody trilinear constraint) *The second order derivative of the multibody trilinear constraint with respect to the second and third argument evaluated at* $(\boldsymbol{x}, \boldsymbol{\ell}'_{i\boldsymbol{x}}, \boldsymbol{\ell}''_{i\boldsymbol{x}})$ *gives the matrix* $M_{i\boldsymbol{x}} \sim \boldsymbol{x}T_i \in \mathbb{R}^{3\times 3}$*, i.e.*

$$\left.\frac{\partial^2(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{\ell}'}\widetilde{\boldsymbol{\ell}''}\mathcal{T})}{\partial\boldsymbol{\ell}'\partial\boldsymbol{\ell}''}\right|_{(\boldsymbol{x},\boldsymbol{\ell}'_{i\boldsymbol{x}},\boldsymbol{\ell}''_{i\boldsymbol{x}})} = M_{i\boldsymbol{x}}. \qquad (17)$$

Proof. A simple calculation shows that

$$\frac{\partial^2(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{\ell}}'\tilde{\boldsymbol{\ell}}''\mathcal{T})}{\partial\boldsymbol{\ell}'\partial\boldsymbol{\ell}''} = \sum_{j=1}^{n}(\boldsymbol{x}T_j)\prod_{k\neq j}(\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_k) + \sum_{j=1}^{n}(\boldsymbol{x}\boldsymbol{\ell}'T_j)\sum_{k\neq j}(\boldsymbol{x}\boldsymbol{\ell}''T_k)\prod_{\ell\neq k}(\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_\ell)$$

Since $\boldsymbol{\ell}'_{i\boldsymbol{x}}$ and $\boldsymbol{\ell}''_{i\boldsymbol{x}}$ are epipolar lines associated with the i^{th} motion, then $\boldsymbol{x}\boldsymbol{\ell}'_{i\boldsymbol{x}}T_i = \boldsymbol{x}\boldsymbol{\ell}''_{i\boldsymbol{x}}T_i = 0$. Therefore,

$$\left.\frac{\partial^2(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{\ell}}'\tilde{\boldsymbol{\ell}}''\mathcal{T})}{\partial\boldsymbol{\ell}'\partial\boldsymbol{\ell}''}\right|_{(\boldsymbol{x},\boldsymbol{\ell}'_{i\boldsymbol{x}},\boldsymbol{\ell}''_{i\boldsymbol{x}})} = (\boldsymbol{x}T_i)\prod_{j\neq i}(\boldsymbol{x}\boldsymbol{\ell}'\boldsymbol{\ell}''T_j) \sim (\boldsymbol{x}T_i)$$

■

Thanks to Theorem 4, we can immediately outline an algorithm for computing the individual trifocal tensors. We just take $\boldsymbol{x} = (1,0,0)^\top$, $\boldsymbol{x} = (0,1,0)^\top$ and $\boldsymbol{x} = (0,0,1)^\top$ and we immediately obtain the slices of the i^{th} trifocal tensor T_i according to its first coordinate. Unfortunately, the above procedure suffers from the following two problems:

1. Since the point $\boldsymbol{x}$ is not necessarily a point in the first view, we cannot compute its epipolar lines $(\boldsymbol{\ell}'_{\boldsymbol{x}}, \boldsymbol{\ell}''_{\boldsymbol{x}})$ using Algorithm 2, because we do not know the corresponding points in the other views. Furthermore, (17) needs the epipolar lines according to all n motions.
2. The slice $\boldsymbol{x}T_i$ of the i^{th} trifocal tensor is only obtained up to a scale factor. Thus, we need an additional procedure to obtain the relative scales among the slices.

The first problem can be easily solved, because the epipolar lines we are looking for must pass through the epipoles, which are already known. Therefore, for each epipole in the second view, $\boldsymbol{e}'_i$, we compute two lines $\boldsymbol{\ell}'_{i1}$ and $\boldsymbol{\ell}'_{i2}$ passing through $\boldsymbol{e}'_i$ and apply Algorithm 2 to compute $\boldsymbol{\ell}'_{i\boldsymbol{x}}$. Similarly, we compute two lines passing to $\boldsymbol{e}''_i$ and then apply Algorithm 2 to compute $\boldsymbol{\ell}''_{i\boldsymbol{x}}$. By repeating this process for the n epipoles, we obtain the epipolar lines $\{\boldsymbol{\ell}'_{i\boldsymbol{x}}\}_{i=1}^n$ and $\{\boldsymbol{\ell}''_{i\boldsymbol{x}}\}_{i=1}^n$ for any $\boldsymbol{x} \in \mathcal{P}^2$, whether an image in the first view or not. The second problem can also be easily solved. Let $e_1 = (1,0,0)^\top$, $e_2 = (0,1,0)^\top$ and $e_3 = (0,0,1)^\top$ be the standard basis for $\mathbb{R}^3$. Also let $\boldsymbol{x} = (x_1, x_2, x_3)^\top$ be any point in $\mathcal{P}^2$ not equal to e_1, e_2 or e_3. Then with the above procedure we know how to compute the four 3×3 matrices $M_{i,e_1} = \lambda_1^{-1}(e_1 T_i)$, $M_{i,e_2} = \lambda_2^{-1}(e_2 T_i)$, $M_{i,e_3} = \lambda_3^{-1} e_3 T_i$ and $M_{i\boldsymbol{x}} = \lambda^{-1}(\boldsymbol{x}T_i)$ up to unknown scale factors λ_1, λ_2, λ_3 and λ, respectively. We therefore have

$$\lambda M_{i\boldsymbol{x}} = \lambda_1 M_{i,e_1} x_1 + \lambda_2 M_{i,e_2} x_2 + \lambda_3 M_{i,e_3} x_3 \qquad (18)$$

which gives a total of 9 linear equations in the 4 unknown scales. After solving for the scales $\lambda_1, \lambda_2, \lambda_3$ from (18), one can recover the individual trifocal tensors as:

$$T_i = [\lambda_1 M_{i,e_1}\ \lambda_2 M_{i,e_2}\ \lambda_3 M_{i,e_3}] \quad i = 1, \ldots, n. \qquad (19)$$

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 3D structure by triangulation, project those 3D 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.

4.4. 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.

In order to obtain an optimal estimate for the trifocal tensors and the segmentation of the correspondences, we assume a generative model in which rigid-body motions occur with probabilities $\{0 \leq \pi_i \leq 1\}_{i=1}^n$, $\sum_{i=1}^n \pi_i = 1$, and the correspondences are corrupted with zero-mean i.i.d. Gaussian noise with variance $\{\sigma_i^2\}_{i=1}^n$. Let $z_{ij} = 1$ denote the event that the j^{th} correspondence belongs to the i^{th} motion. Also let $\epsilon_{ij} = RepErr(\boldsymbol{x}_j, \boldsymbol{x}'_j, \boldsymbol{x}''_j, T_i)$ be the reprojection error for correspondence j according to motion i. Then the complete log-likelihood (neglecting constant factors) of the data $(\boldsymbol{x}_j, \boldsymbol{x}'_j, \boldsymbol{x}''_j)$ and the latent variables z_{ij} is given by

$$\log\prod_{j=1}^{N}\prod_{i=1}^{n}\left(\frac{\pi_i}{\sigma_i}\exp\left(\frac{-\epsilon_{ij}}{2\sigma_i^2}\right)\right)^{z_{ij}} = \sum_{j=1}^{N}\sum_{i=1}^{n} z_{ij}(\log(\frac{\pi_i}{\sigma_i}) - \frac{\epsilon_{ij}}{2\sigma_i^2}).$$

In order to obtain a maximum likelihood estimate for the parameters $\theta = \{(T_i, \sigma_i, \pi_i)\}_{i=1}^n$ given the data $X = \{(\boldsymbol{x}_j, \boldsymbol{x}'_j, \boldsymbol{x}''_j)\}_{j=1}^N$, we maximize the above function using the Expectation-Maximization algorithm [2] as follows:

E-step: Computing the expected log-likelihood. Given a current estimate for the parameters, we can compute the expected value of the latent variables

$$w_{ij} \doteq E[z_{ij}|X,\theta] = P(z_{ij} = 1|X,\theta) = \frac{\frac{\pi_i}{\sigma_i}\exp(-\frac{\epsilon_{ij}}{2\sigma_i^2})}{\sum_{i=1}^n \frac{\pi_i}{\sigma_i}\exp(-\frac{\epsilon_{ij}}{2\sigma_i^2})}.$$

Then the expected complete log-likelihood is given by

$$\sum_{j=1}^N \sum_{i=1}^n w_{ij}(\log(\pi_i) - \log(\sigma_i)) - w_{ij}\frac{\epsilon_{ij}}{2\sigma_i^2}. \qquad (20)$$

M-step: Maximizing the expected log-likelihood. After differentiating (20) w.r.t. the trifocal tensors, we notice that the optimization problem can be decomposed into n optimization problems of the form $\min \sum_{j=1}^n w_{ij}\epsilon_{ij}$, which can be solved using standard structure from motion algorithms with the correspondences weighted according to w_{ij}. Then, we can solve for π_i using Lagrange multipliers as

$$\sum_{i=1}^n \sum_{j=1}^N w_{ij}\log(\pi_i) + \lambda(1 - \sum_{i=1}^n \pi_i) \implies \pi_i = \frac{\sum_{j=1}^N w_{ij}}{N}.$$

Finally, after differentiating (20) w.r.t. σ_i we obtain

$$\sigma_i^2 = \frac{\sum_{j=1}^N w_{ij}\epsilon_{ij}}{\sum_{j=1}^N w_{ij}}.$$

The EM algorithm proceeds by iterating between the E-step and the M-step, until the estimates converge to a local maxima.

5. Experiments

In this paper, 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 first computes the epipoles using Algorithms 1-3, the trifocal tensors as in Section 4.3, and then clusters the point correspondences according to their motion classes using the Sampson-distance residual to the different motions.
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 4.4. For ease of computation, in the M-step we first compute each trifocal tensor linearly as in Section 2.1. If the error (20) increases, we recompute the trifocal tensors using the linear algebraic algorithm in [4]. If the error (20) 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 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%.

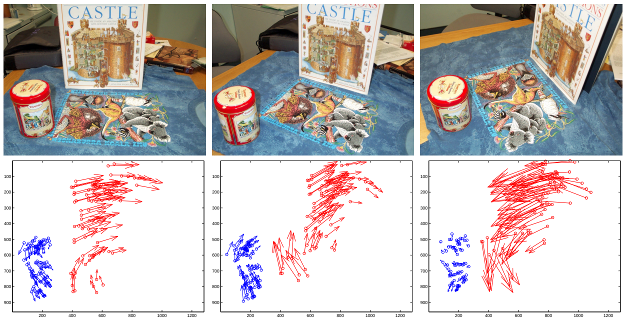


Figure 1: Top: views 1-3 of a sequence with two rigid-body motions. Bottom: 2D displacements of the 140 correspondences from the current view ('o') to the next ('→').

We also tested the performance of our algorithm on two sequences with transparent motions, so that the only cue for clustering the correspondences is the motion cue. Given a set of correspondences of 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 give an error of 4.1% and 2.5%. Following this with the K-means did not improve the classification, while following this with EM achieved a perfect segmentation. Figure 3 shows two out of three frames from the Tea-Tins sequence and the inter-frame displacement of the $N = 42 \times 2 = 84$ correspondences. As shown in Table 1, the K-means algorithm gives a mean error of 32.0% with a mean number of iterations of 49.8. The algebraic algorithms give an error of 15.4% and 10.7%, which is reduced to 9.5% by K-means, and to 0% by EM.

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%
Tea-Tins	32.0%	15.4%	10.7%	9.5%	4.8%

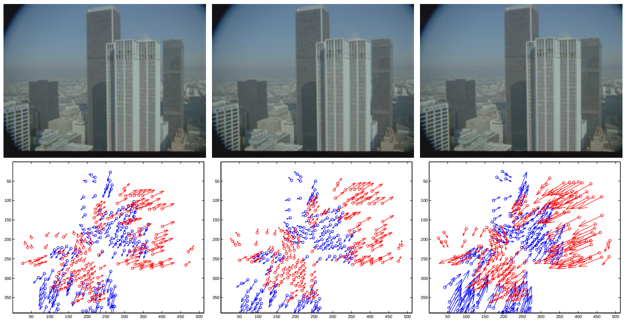
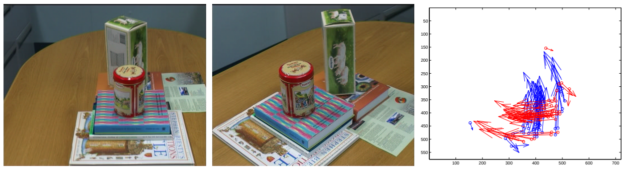


Figure 2: Top: frames 1, 4 and 7 of the Wilshire sequence. Bottom: 2D displacement of the 328 original and flipped correspondences from current view ('○') to the next ('→').



Frame 83 Frame 94 2D disp. 83–94

Figure 3: Two (out of three) views from the Tea-Tins sequence and the 2D displacement of the 84 original and flipped correspondences.

We also tested our algorithm on synthetic data. We randomly generated two groups of 100 3D points each with a depth variation 100-400 units of focal length (u.f.l.). The two rigid-body 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 i.i.d. Gaussian noise with s.t.d. of $[0, 1]$ pixels for an image size of 1000×1000. Figure 4 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 minima due to bad initialization. The algebraic algorithms (I and II) achieve a misclassification ratio of about 20.2% and 9.1% and a translation 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 directly 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.

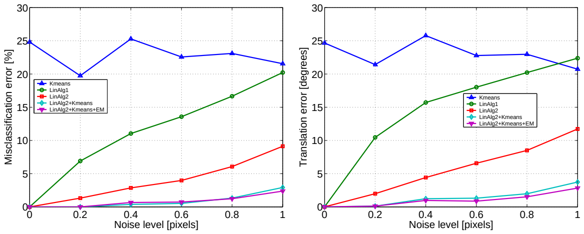


Figure 4: Motion segmentation and motion estimation (translation) errors as a function of noise.

6. 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.

References

[1] J. Costeira and T. Kanade. Multi-body factorization methods for motion analysis. In *ICCV*, pages 1071–1076, 1995.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38, 1977.

[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. To appear.

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

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

[9] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *CVPR*, 2003.

[10] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 2004.

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