Journal of Mathematical Imaging and Vision
© 2006 Springer Science + Business Media, LLC. *Manufactured in The Netherlands.*
DOI: 10.1007/s10851-006-8286-z

A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation and Estimation*

RENÉ VIDAL
Center for Imaging Science, Department of Biomedical Engineering, Johns Hopkins University, 308B Clark Hall, 3400 N. Charles St., Baltimore, MD 21218, USA
rvidal@cis.jhu.edu

YI MA
Electrical & Computer Engineering Department, University of Illinois at Urbana-Champaign, 1406 West Green Street, Urbana, IL 61801, USA
yima@uiuc.edu

Published online: 1 September 2006

Abstract. In this paper, we present an analytic solution to the problem of estimating an unknown number of 2-D and 3-D motion models from two-view point correspondences or optical flow. The key to our approach is to view the estimation of multiple motion models as the estimation of a single *multibody motion model*. This is possible thanks to two important algebraic facts. First, we show that all the image measurements, regardless of their associated motion model, can be *fit* with a single real or complex *polynomial*. Second, we show that the parameters of the individual motion model associated with an image measurement can be obtained from the *derivatives* of the polynomial at that measurement. This leads to an algebraic motion segmentation and estimation algorithm that applies to most of the two-view motion models that have been adopted in computer vision. Our experiments show that the proposed algorithm out-performs existing algebraic and factorization-based methods in terms of efficiency and robustness, and provides a good initialization for iterative techniques, such as Expectation Maximization, whose performance strongly depends on good initialization.

Keywords: multibody structure from motion, motion segmentation, multibody epipolar constraint, multibody fundamental matrix, multibody homography, and Generalized PCA (GPCA)

1. Introduction

An important problem in computer vision is to estimate a model for the motion of a scene from the trajectories of a set of 2-D feature points. This problem is well understood when the relative motion between the scene and the camera can be described with a single rigid-body motion [14, 24]. For example, it is well-known that two views of a scene are related by the *epipolar constraint* [23] and that multiple views are related by the *multilinear constraints* [15]. Such constraints can be used to estimate a motion model using linear techniques such as the eight-point algorithm and its variations.

In this paper we address the more general case of motion segmentation and estimation for *dynamic*

*This paper is an extended version of [34]. The authors thank Sampreet Niyogi for his help with the experimental section of the paper. This work was partially supported by Hopkins WSE startup funds, UIUC ECE startup funds, and by grants NSF CAREER IIS-0347456, NSF CAREER IIS-0447739, NSF CRS-EHS-0509151, NSF-EHS-0509101, NSF CCF-TF-0514955, ONR YIP N00014-05-1-0633 and ONR N00014-05-1-0836.

scenes in which both the camera and an unknown number of objects with unknown 3-D structure can move independently. Thus, different regions of the image may obey different 2-D or 3-D motion models due to depth discontinuities, perspective effects, multiple motions, etc. More specifically, we consider the following problem:

Problem 1 (Multiple-Motion Segmentation and Estimation)

Given a set of image measurements $\{(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j)\}_{j=1}^N$ taken from two views of a motion sequence related by a collection of n motion models $\{\mathcal{M}_i\}_{i=1}^n$, estimate the number of motion models and their parameters without knowing which image measurements correspond to which motion model.

Depending on whether one is interested in understanding the motion in the 2-D image, or the motion in 3-D space, the motion segmentation and estimation problem can be divided into two main categories. 2-*D motion segmentation* refers to the estimation of the 2-D motion field in the image plane (optical flow), while 3-*D motion segmentation* refers to the estimation of the 3-D motion (rotation and translation) of multiple rigidly moving objects relative to the camera. When the scene is static, one can model its 2-D motion with a mixture of 2-D motion models such as translational, affine or projective. Even though a single 3-D motion is present, multiple 2-D motion models arise, because of perspective effects, depth discontinuities, occlusions, transparent motions, etc. In this case, the task of 2-D motion segmentation is to estimate these models from the image data. When the scene is dynamic one can still model its motion with a mixture of 2-D motion models. Some of these models are due to independent 3-D motions, e.g., when the motion of an object relative to the camera can be well approximated by the affine motion model. Others are due to perspective effects and/or depth discontinuities, e.g., when some of the 3-D motions are broken into different 2-D motions. The task of 3-D motion segmentation is to obtain a collection of 3-D motion models, in spite of perspective effects and/or depth discontinuities.

1.1. Related Literature

Classical approaches to 2-D motion segmentation separate the image flow into different regions by looking for flow discontinuities [3, 28], fitting a mixture of parametric models through successive computation of dominant motions [16], or using a layered representation of the motion field [6]. The problem has also been formalized in a maximum likelihood framework [2, 17, 42, 43] in which the estimation of the motion models and their regions of support is done by alternating between the segmentation of the image measurements and the estimation of the motion parameters using Expectation Maximization (EM). EM-like approaches provide robust motion estimates by combining information over large regions in the image. However, their convergence to the optimal solution strongly depends on good initialization [27, 30]. Existing initialization techniques obtain a 2-D motion representation from local patches and cluster this representation using *K*-means [41] or normalized cuts [27]. The drawback of these approaches is that they are based on a local computation of 2-D motion, which is subject to the aperture problem and to the estimation of a single model across motion boundaries. Some of these problems can be partially solved by incorporating multiple frames and a local process that forces the clusters to be connected [22].

The 3-D motion segmentation problem has received relatively less attention. Existing work [31] solves this problem by successive computation of dominant motions using methods from robust statistics. Such methods fit a single motion model to all the image measurements using RANSAC [8]. The measurements that fit this motion model well (inliers) are removed from the data set, and RANSAC is re-applied to the remaining points to obtain a second motion model. This process is repeated until most of the measurements have been assigned to a model. Alternative approaches to 3-D motion segmentation [7] first cluster the features corresponding to the same motion using e.g., *K*-means or spectral clustering, and then estimate a single motion model for each group. This can also be done in a probabilistic framework by alternating between feature clustering and single-body motion estimation using the EM algorithm. In order to deal with the initialization problem of EM-like approaches, recent work has concentrated on the study of the geometry of dynamic scenes, including the analysis of multiple points moving linearly with constant speed [10, 26] or in a conic section [1], multiple points moving in a plane [29], multiple translating planes [44], self-calibration from multiple motions [9, 11], multiple moving objects seen by an affine camera [4, 18, 20, 21, 33, 46, 47], and two-object segmentation from two perspective views [45].

Table 1. 2-D and 3-D motion models considered in this paper.

Motion models	Model equations	Model parameters	Segmentation of
2-D translational	$x_2 = x_1 + T_i$	$\{T_i \in \mathbb{R}^2\}_{i=1}^n$	Hyperplanes in $\mathbb{C}^2$
2-D similarity	$x_2 = \lambda_i R_i x_1 + T_i$	$\{R_i \in SO(2), \lambda_i \in \mathbb{R}^+\}_{i=1}^n$	Hyperplanes in $\mathbb{C}^3$
2-D affine	$x_2 = A_i \left[\begin{smallmatrix} x_1 \\ 1 \end{smallmatrix}\right]$	$\{A_i \in \mathbb{R}^{2\times 3}\}_{i=1}^n$	Hyperplanes in $\mathbb{C}^4$
3-D translational	$0 = x_2^T \widehat{T}_i x_1$	$\{T_i \in \mathbb{R}^3\}_{i=1}^n$	Hyperplanes in $\mathbb{R}^3$
3-D rigid-body	$0 = x_2^T F_i x_1$	$\{F_i \in \mathbb{R}^{3\times 3}\}_{i=1}^n$	Bilinear forms in $\mathbb{R}^6$
3-D homography	$x_2 \sim H_i x_1$	$\{H_i \in \mathbb{R}^{3\times 3}\}_{i=1}^n$	Bilinear forms in $\mathbb{C}^5$

The case of multiple moving objects seen by two perspective views was recently studied in [38–40], which proposed a generalization of the 8-point algorithm based on the so-called *multibody epipolar constraint* and its associated *multibody fundamental matrix*. The method simultaneously recovers multiple fundamental matrices using polynomial fitting and differentiation, and can be extended to three perspective views via the so-called *multibody trifocal tensor* [13]. To the best of our knowledge, the only existing works on 3-D motion segmentation from omnidirectional cameras are [25, 32].

1.2. Contributions of This Paper

In this paper, we address the initialization of iterative approaches to motion estimation and segmentation by proposing a *non-iterative* algebraic solution to Problem 1 that applies to most 2-D and 3-D motion models in computer vision, as detailed in Table 1.

The key to our approach is to view the estimation of multiple motion models as the estimation of a *single*, though more complex, *multibody motion model* that is then factored into the original models. This is achieved by (1) algebraically eliminating the feature segmentation problem, (2) fitting a single multibody motion model to all the image measurements, and (3) segmenting the multibody motion model into its individual components. More specifically, our approach proceeds as follows:

1. *Eliminate Feature Segmentation*: Find an algebraic equation that is satisfied by all the image measurements, regardless of the motion model associated with each measurement. For the motion models considered in this paper, the ith motion model will be typically defined by an algebraic equation of the form $f(x_1, x_2, \mathcal{M}_i) = 0$, for $i = 1, \ldots, n$. Therefore, an algebraic equation that is satisfied by all the data is

   $$p_n(x_1, x_2, \mathcal{M}) \\ = f(x_1, x_2, \mathcal{M}_1) \cdot \cdots \cdot f(x_1, x_2, \mathcal{M}_n) = 0. \quad (1)$$

   Such an equation represents a single *multibody motion model* whose parameters $\mathcal{M}$ encode those of the original motion models $\{\mathcal{M}_i\}_{i=1}^n$.
2. *Multibody Motion Estimation*: Estimate the parameters $\mathcal{M}$ of the multibody motion model from the given image measurements. For the motion models considered in this paper, the parameters $\mathcal{M}$ will correspond to the coefficients of a real or complex polynomial p_n of degree n. We will show that the number of motions n and the coefficients $\mathcal{M}$ can be obtained *linearly* after properly embedding the image measurements into a higher-dimensional space.
3. *Motion Segmentation*: Recover the parameters of the original motion models from the parameters of the multibody motion model $\mathcal{M}$, that is,

   $$\mathcal{M} \mapsto \{\mathcal{M}_i\}_{i=1}^n. \quad (2)$$

   We will show that the individual motion parameters $\mathcal{M}_i$ can be computed from the *derivatives* of p_n evaluated at a collection of n image measurements that can be obtained automatically from the data.

This new approach to motion segmentation offers two important technical advantages over previously known algebraic solutions to the segmentation of 3-D translational [36] and rigid-body motions (fundamental matrices) [40] based on homogeneous *polynomial factorization*:

1. It is based on *polynomial differentiation* rather than *polynomial factorization*, which greatly improves the efficiency, accuracy and robustness of the algorithm.

2. It applies to either feature point correspondences or optical flow and includes most of the two-view motion models in computer vision: 2-D translational, 2-D similarity, 2-D affine, 3-D translational, 3-D rigid-body motions (fundamental matrices), or 3-D motions of planar scenes (homographies), as shown in Table 1. The unification is achieved by embedding some of the motion models into the complex domain, which resolves cases such as 2-D affine motions and 3-D homographies that could not be easily handled in the real domain.

With respect to extant probabilistic methods, our approach has the advantage of providing a global, non-iterative solution that does not need initialization. Therefore, our method can be used to initialize any iterative or optimization based technique, such as EM, or else in a layered (multiscale) or hierarchical fashion at the user's discretion.

Although the derivation of the algorithm will assume noise free data, the algorithm is designed to work with a moderate level of noise, as we will point out shortly. However, in its present form the algorithm does not consider the presence of outliers in the data. Nevertheless, as a key step in our algorithm is to estimate the multibody motion model, one can improve the robustness of the estimate by using one of many existing robust (covariance) estimators, such as the M-estimators, multivariate trimming (MVT), and influence function.[1] However, a detailed account is beyond the scope of this paper.

2. Segmenting Hyperplanes in $\mathbb{C}^K$

As we will see shortly, most 2-D and 3-D motion segmentation problems are equivalent or can be reduced to clustering data lying in multiple hyperplanes in $\mathbb{R}^3$, $\mathbb{C}^2$, $\mathbb{C}^3$, or $\mathbb{C}^4$. Rather than solving this problem for each particular case, we present in this section a unified solution to the common mathematical problem of segmenting hyperplanes in $\mathbb{C}^K$ with an arbitrary K by adapting the Generalized PCA algorithm of [35, 37] to the complex domain.

To that end, let $\boldsymbol{z}$ be a vector in $\mathbb{C}^K$ and let $\boldsymbol{z}^T$ be its transpose *without* conjugation.[2] A homogeneous polynomial of degree n in $\boldsymbol{z}$ is a polynomial $p_n(\boldsymbol{z})$ such that $p_n(\lambda \boldsymbol{z}) = \lambda^n p_n(\boldsymbol{z})$ for all λ in $\mathbb{C}$. The space of all homogeneous polynomials of degree n in K variables, S_n, is a vector space of dimension $M_n(K) \doteq \binom{n+K-1}{K-1} = \binom{n+K-1}{n}$. A particular basis for S_n is obtained by considering all the monomials of degree n in K variables, that is $\boldsymbol{z}^I \doteq z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K}$ with $0 \leq n_j \leq n$ for $j = 1, \ldots, K$, and $n_1 + n_2 + \cdots + n_K = n$. To represent S_n, it is convenient to define the *Veronese map* $\nu_n : \mathbb{C}^K \to \mathbb{C}^{M_n(K)}$ of degree n as [12]

$$\nu_n : [z_1, \ldots, z_K]^T \mapsto [\ldots, \boldsymbol{z}^I, \ldots]^T$$

with the index I chosen in the degree-lexicographic order. The Veronese map is also known as the *polynomial embedding* in the machine learning community. Using this notation, each polynomial $p_n(\boldsymbol{z}) \in S_n$ can be written as a linear combination of the monomials $\boldsymbol{z}^I$ as

$$p_n(\boldsymbol{z}) = \boldsymbol{c}^T \nu_n(\boldsymbol{z}) = \sum c_{n_1,n_2,\ldots,n_K} z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K}, \quad (3)$$

where $\boldsymbol{c} \in \mathbb{C}^{M_n(K)}$ is the vector of coefficients.

Assume now that we are given a set of sample points $\boldsymbol{Z} \doteq \{\boldsymbol{z}^j \in \mathbb{C}^K\}_{j=1}^N$ drawn from $n \geq 1$ different hyperplanes $\{\mathcal{P}_i \subseteq \mathbb{C}^K\}_{i=1}^n$ of dimension $K-1$. Without knowing which points belong to which hyperplane, we would like to determine the number of hyperplanes, a basis for each hyperplane, and the segmentation of the data points.

Notice that every $(K-1)$-dimensional hyperplane $\mathcal{P}_i \subset \mathbb{C}^K$ can be represented by its *normal* vector $\boldsymbol{b}_i \in \mathbb{C}^K$ as

$$\mathcal{P}_i \doteq \big\{\boldsymbol{z} \in \mathbb{C}^K : \boldsymbol{b}_i^T \boldsymbol{z} = b_{i1} z_1 + b_{i2} z_2 + \cdots + b_{iK} z_K = 0\big\}. \quad (4)$$

We assume that the hyperplanes are different from each other, and hence the normal vectors $\{\boldsymbol{b}_i\}$ are pairwise linearly independent. For uniqueness, we also assume that either the norm or the last entry of each $\boldsymbol{b}_i$ is 1.

2.1. Eliminating Data Segmentation

We first notice that each point $\boldsymbol{z} \in \boldsymbol{Z}$, regardless of which one of the n hyperplanes $\{\mathcal{P}_i\}_{i=1}^n$ it is associated with, must satisfy the following homogeneous polynomial of degree n in K complex variables

$$\begin{aligned} p_n(\boldsymbol{z}) &\doteq \prod_{i=1}^n \big(\boldsymbol{b}_i^T \boldsymbol{z}\big) = \boldsymbol{c}^T \nu_n(\boldsymbol{z}) \\ &= \sum c_{n_1,n_2,\ldots,n_K} z_1^{n_1} z_2^{n_2} \cdots z_K^{n_K} = 0, \end{aligned} \quad (5)$$

because we must have $\boldsymbol{b}_i^T \boldsymbol{z} = 0$ for one of the $\boldsymbol{b}_i$. In the context of motion segmentation, the vectors

$\boldsymbol{b}_i$ represent the parameters of each individual motion model, and the coefficient vector $\boldsymbol{c} \in \mathbb{C}^{M_n(K)}$ represents the *multibody motion parameters*. We call n the *degree of the multibody motion model.*

2.2. Estimating the Degree and Parameters of the Multibody Motion Model

Since the polynomial $p_n(z) = \boldsymbol{c}^T \nu_n(z)$ must be satisfied by all the data points $\boldsymbol{Z} = \{z^j \in \mathbb{C}^K\}_{j=1}^N$, we have that $\boldsymbol{c}^T \nu_n(z^j) = 0$ for all $j = 1, \ldots, N$. Therefore, we obtain the following linear system on $\boldsymbol{c}$

$$L_n \boldsymbol{c} = 0 \quad \in \mathbb{C}^N, \tag{6}$$

where $L_n \doteq [\nu_n(z^1), \nu_n(z^2), \ldots, \nu_n(z^N)]^T \in \mathbb{C}^{N \times M_n(K)}$. In order to solve for $\boldsymbol{c}$, we first need to know the number of hyperplanes n. The following lemma allows one to compute n from the image measurements.

Lemma 1. *Given a sufficient number of sample points in general position on n hyperplanes in $\mathbb{C}^K$, we have*

$$n = \arg\min_i \{\operatorname{rank}(L_i) = M_i(K) - 1\}, \tag{7}$$

and the equation $L_n \boldsymbol{c} = 0$ determines the vector $\boldsymbol{c}$ up to a nonzero scale.

Proof: The proof of this lemma can be found in [37] and is a consequence of the basic algebraic fact that there is a one-to-one correspondence between a polynomial and its zero set. Therefore, there is no polynomial of degree $i < n$ that vanishes on all the points of the n hyperplanes, hence we must have $\operatorname{rank}(L_i) = M_i(K)$ for $i < n$. Conversely, there are multiple polynomials of degree $i > n$, namely any multiple of $p_n(z)$, which are satisfied by all the data, hence rank $(L_i) < M_i(K) - 1$ for $i > n$. Therefore, the case $i = n$ is the only one in which system $L_n \boldsymbol{c} = 0$ has a unique solution (up to scale). □

According to this lemma, the number of hyperplanes (or motions) can be uniquely determined from the smallest i such that L_i drops rank. Furthermore, if the last entry of each $\boldsymbol{b}_i$ is equal to one, so is the last entry of $\boldsymbol{c}$, hence one can solve for $\boldsymbol{c}$ uniquely in this case.

In the presence of noise, we cannot directly estimate n from (7), because the matrix L_i may be full rank for all $i \geq 1$. By borrowing tools from the model selection literature [19], we may determine the number of hyperplanes (or motions) from noisy data as

$$n = \arg\min_i \left\{ \frac{\sigma^2_{M_i(K)}(L_i)}{\sum_{k=1}^{M_i(K)-1} \sigma_k^2(L_i)} + \kappa M_i(K) \right\}, \tag{8}$$

where $\sigma_k(L_i)$ is the kth singular value of L_i and $\kappa > 0$ is a (weighting) parameter. Once n is determined, we can solve for $\boldsymbol{c}$ in a least-squares sense as the singular vector of L_n associated with its smallest singular value. One can normalize $\boldsymbol{c}$ so that its last entry is 1, whenever appropriate.

2.3. Segmenting the Multibody Motion Model

Given $\boldsymbol{c}$, we now present an algorithm for computing the parameters $\boldsymbol{b}_i$ of each individual hyperplane (or motion) from the derivatives of p_n. To that end, we consider the derivative of $p_n(z)$,

$$Dp_n(z) \doteq \frac{\partial p_n(z)}{\partial z} = \sum_{i=1}^{n} \prod_{\ell \neq i} \left(\boldsymbol{b}_\ell^T z\right) \boldsymbol{b}_i, \tag{9}$$

and notice that if we evaluate $Dp_n(z)$ at a point $z = \boldsymbol{y}_i$ that belongs to only the ith hyperplane (or motion), i.e. if $\boldsymbol{y}_i$ is such that $\boldsymbol{b}_i^T \boldsymbol{y}_i = 0$, then we have $Dp_n(\boldsymbol{y}_i) \sim \boldsymbol{b}_i$. Therefore, given $\boldsymbol{c}$ we can obtain the hyperplane (motion) parameters as

$$\boldsymbol{b}_i = \left. \frac{Dp_n(z)}{e_K^T Dp_n(z)} \right|_{z=\boldsymbol{y}_i} \quad \text{or} \quad \boldsymbol{b}_i = \left. \frac{Dp_n(z)}{\|Dp_n(z)\|} \right|_{z=\boldsymbol{y}_i} \tag{10}$$

depending on whether $\boldsymbol{e}_K^T \boldsymbol{b}_i = 1$ or $\|\boldsymbol{b}_i\| = 1$, where $e_K = [0, \ldots, 0, 1]^T \in \mathbb{C}^K$ and $\boldsymbol{y}_i \in \mathbb{C}^K$ is a nonzero vector such that $\boldsymbol{b}_i^T \boldsymbol{y}_i = 0$.

The rest of the problem is to find one point $\boldsymbol{y}_i \in \mathbb{C}^K$ in each one of the hyperplanes $\mathcal{P}_i = \{z \in \mathbb{C}^K : \boldsymbol{b}_i^T z = 0\}$ for $i = 1, \ldots, n$. To that end, notice that we can always choose a point $\boldsymbol{y}_n$ lying in one of the hyperplanes as any of the points in the data set $\boldsymbol{Z}$. However, in the presence of noise and outliers an arbitrary point in $\boldsymbol{Z}$ may be far from all the hyperplanes. In order to choose points close to the hyperplanes, we need to be able to compute the *distance* from each data point to its closest hyperplane, *without* knowing the normals to the hyperplanes. The following lemma allows us to compute a first order approximation to such a distance.[3]

Lemma 2. *Let $\tilde{z} \in \mathcal{P}_i$ be the projection of a point $z \in \mathbb{C}^K$ onto its closest hyperplane $\mathcal{P}_i$. Also let $\Pi \doteq [I_{K-1}\ 0] \in \mathbb{R}^{(K-1)\times K}$ or $I_K \in \mathbb{R}^{K\times K}$, depending on whether $e_K^T \boldsymbol{b}_i = 1$ or $\|\boldsymbol{b}_i\| = 1$ for $i = 1, \ldots, n$, respectively. Then the Euclidean distance from z to $\mathcal{P}_i$ is given by*

$$\|z - \tilde{z}\| = n\frac{|p_n(z)|}{\|\Pi D p_n(z)\|} + O(\|z - \tilde{z}\|^2). \qquad (11)$$

Proof: It follows as a corollary of Lemma 1 in [37]. □

Thanks to Lemma 2, we can choose a point in the data set close to one of the subspaces as:

$$\boldsymbol{y}_n = \arg\min_{z\in \boldsymbol{Z}} \frac{|p_n(z)|}{\|\Pi D p_n(z)\|}, \qquad (12)$$

and then compute the normal vector at $\boldsymbol{y}_n$ as

$$\boldsymbol{b}_n = Dp_n(\boldsymbol{y}_n)/\big(e_K^T Dp_n(\boldsymbol{y}_n)\big) \quad \text{or}$$
$$\boldsymbol{b}_i = Dp_n(\boldsymbol{y}_i)/\|Dp_n(\boldsymbol{y}_i)\|.$$

In order to find a point $\boldsymbol{y}_{n-1}$ in one of the remaining hyperplanes, we could just remove the points in $\mathcal{P}_n$ from $\boldsymbol{Z}$ and compute $\boldsymbol{y}_{n-1}$ similarly to (12), but minimizing over $\boldsymbol{Z}\setminus\mathcal{P}_n$, and so on. However, this process is not very robust in the presence of noise, as it depends on the choice of a threshold in order to determine which points belong to $\mathcal{P}_n$. Therefore, we propose an alternative solution that penalizes choosing a point from $\mathcal{P}_n$ in (12) by dividing the objective function by the distance from z to $\mathcal{P}_n$, namely $|\boldsymbol{b}_n^T z|/\|\Pi \boldsymbol{b}_n\|$. That is, we can choose a point in or close to $\bigcup_{i=1}^{n-1}\mathcal{P}_i$ as

$$\boldsymbol{y}_{n-1} = \arg\min_{z\in \boldsymbol{Z}} \frac{\frac{|p_n(z)|}{\|\Pi D p_n(z)\|} + \delta}{\frac{|\boldsymbol{b}_n^T z|}{\|\Pi \boldsymbol{b}_n\|} + \delta}, \qquad (13)$$

where $\delta > 0$ is a small positive number chosen to avoid cases in which both the numerator and the denominator are zero (e.g., with perfect data). By repeating this process for the remaining hyperplanes, we obtain the Polynomial Differentiation Algorithm (Algorithm 1) for segmenting hyperplanes in $\mathbb{C}^K$.

Algorithm 1 (Polynomial Differentiation Algorithm (PDA) for clustering Multiple Hyperplanes in $\mathbb{C}^K$)

```
Given data points $\boldsymbol{Z} = \{\boldsymbol{z}^j \in \mathbb{C}^K\}_{j=1}^N$:
– construct the embedded data matrix
      $L_i = [\nu_i(\boldsymbol{z}^1), \nu_i(\boldsymbol{z}^2), \ldots, \nu_i(\boldsymbol{z}^N)]^T \in \mathbb{C}^{N\times M_i(K)}$,
– determine the number of hyperplanes $n$ from
      $n = \arg\min_i \left\{ \frac{\sigma^2_{M_i(K)}(L_i)}{\sum_{k=1}^{M_i(K)-1} \sigma_k^2(L_i)} + \kappa\, M_i(K) \right\}$;
– solve for $\boldsymbol{c} \in \mathbb{C}^{M_n(K)}$ from the linear system
      $L_n \boldsymbol{c} = 0$;
– set $p_n(\boldsymbol{z}) = \boldsymbol{c}^T \nu_n(\boldsymbol{z})$;
– for $i = n : 1$,
      $\boldsymbol{y}_i = \arg\min_{\boldsymbol{z}\in\boldsymbol{Z}} \frac{\frac{|p_n(\boldsymbol{z})|}{\|\Pi D p_n(\boldsymbol{z})\|} + \delta}{\frac{|\boldsymbol{b}_{i+1}^T \boldsymbol{z}|\cdots|\boldsymbol{b}_n^T \boldsymbol{z}|}{\|\Pi \boldsymbol{b}_{i+1}\|\cdots\|\Pi \boldsymbol{b}_n\|} + \delta}$;
      $\boldsymbol{b}_i = \begin{cases} \frac{Dp_n(\boldsymbol{y}_i)}{\|Dp_n(\boldsymbol{y}_i)\|} & \text{if } \Pi = I_K \\ \frac{Dp_n(\boldsymbol{y}_i)}{e_K^T Dp_n(\boldsymbol{y}_i)} & \text{otherwise} \end{cases}$;
– end.
```

Notice that one could also choose the points $\boldsymbol{y}_i$ in a purely algebraic fashion, e.g., by intersecting a random line with the hyperplanes [38], or else by dividing the polynomial $p_n(z)$ by $\boldsymbol{b}_n^T z$ [37]. However, we have chosen to present the simpler Algorithm 1 instead, because it has a better performance with noisy data and is not very sensitive to the choice of δ.

3. 2-D Motion Segmentation by Segmenting Hyperplanes in $\mathbb{C}^K$

This section considers the problem of segmenting a collection of 2-D motion models from point correspondences in two frames of a video sequence, or from optical flow measurements at each pixel. We show that when the image measurements are related by a collection of 2-D translational, 2-D similarity or 2-D affine motion models, the motion segmentation and estimation problem (Problem 1) is equivalent to segmenting hyperplanes in $\mathbb{C}^2$, $\mathbb{C}^3$, or $\mathbb{C}^4$, respectively. We solve this segmentation problem using the algebraic algorithm presented in the previous section.

3.1. Segmentation of 2-D Translational Motions: Segmenting Hyperplanes in $\mathbb{C}^2$

3.1.1. The Case of Feature Points. Under the 2-D translational motion model the two images are related by one out of n possible 2-D translations $\{T_i \in \mathbb{R}^2\}_{i=1}^n$. That is, for each feature pair $\boldsymbol{x}_1 \in \mathbb{R}^2$ and $\boldsymbol{x}_2 \in \mathbb{R}^2$ there exists a 2-D translation $T_i \in \mathbb{R}^2$ such that

$$\boldsymbol{x}_2 = \boldsymbol{x}_1 + T_i. \tag{14}$$

Therefore, if we interpret the displacement of the features $(\boldsymbol{x}_2 - \boldsymbol{x}_1)$ and the 2-D translations T_i as complex numbers $(\boldsymbol{x}_2 - \boldsymbol{x}_1) \in \mathbb{C}$ and $T_i \in \mathbb{C}$, then we can re-write equation (14) as

$$\boldsymbol{b}_i^T \boldsymbol{z} \doteq [T_i \;\; 1]\begin{bmatrix} 1 \\ -(\boldsymbol{x}_2 - \boldsymbol{x}_1) \end{bmatrix} = 0 \quad \in \mathbb{C}^2. \tag{15}$$

This equation corresponds to a hyperplane in $\mathbb{C}^2$ whose normal vector $\boldsymbol{b}_i$ encodes the 2-D translational motion T_i. Therefore, the segmentation of 2-D translational motions from a set of point correspondences $\{(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j)\}_{j=1}^N$ is equivalent to clustering data $\{z^j \in \mathbb{C}^2\}_{j=1}^N$ lying in n hyperplanes in $\mathbb{C}^2$ with normal vectors $\{\boldsymbol{b}_i \in \mathbb{C}^2\}_{i=1}^n$. As such, we can obtain the motion parameters $\{\boldsymbol{b}_i \in \mathbb{C}^2\}_{i=1}^n$ by applying Algorithm 1 with $K = 2$ and $\Pi = [1 \;\; 0]$ to a collection of $N \geq M_n(2) - 1 = n$ image measurements $\{z^j \in \mathbb{C}^2\}_{j=1}^N$ in general position on the n hyperplanes. The original *real* motion parameters are then given as

$$T_i = [\mathrm{Re}(\boldsymbol{b}_{i1}), \mathrm{Im}(\boldsymbol{b}_{i1})]^T, \quad \text{for } i = 1, \ldots, n. \tag{16}$$

3.1.2. The Case of Translational Optical Flow. Imagine now that rather than a collection of feature points we are given the optical flow $\{\boldsymbol{u}^j \in \mathbb{R}^2\}_{j=1}^N$ between two consecutive views of a video sequence. If we assume that the optical flow is piecewise constant, i.e. the optical flow of every pixel in the image takes only n possible values $\{T_i \in \mathbb{R}^2\}_{i=1}^n$, then at each pixel $j \in \{1, \ldots, N\}$ there exists a motion T_i such that

$$\boldsymbol{u}^j = T_i. \tag{17}$$

The problem is now to estimate the n motion models $\{T_i\}_{i=1}^n$ from the optical flow measurements $\{\boldsymbol{u}^j\}_{j=1}^N$. This problem can be solved using the same technique as in the case of feature points after replacing $\boldsymbol{x}_2 - \boldsymbol{x}_1 = \boldsymbol{u}$.

3.2. Segmentation of 2-D Similarity Motions: Segmenting Hyperplanes in $\mathbb{C}^3$

3.2.1. The Case of Feature Points. In this case, we assume that for each feature point $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ there exists a 2-D rigid-body motion $(R_i, T_i) \in SE(2)$ and a scale $\lambda_i \in \mathbb{R}^+$ such that

$$\boldsymbol{x}_2 = \lambda_i R_i \boldsymbol{x}_1 + T_i = \lambda_i \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \boldsymbol{x}_1 + T_i. \tag{18}$$

If we interpret the rotation as a unitary complex number $R_i = \exp(\theta_i \sqrt{-1}) \in \mathbb{C}$, and the translation vector and the image features as points in the complex plane T_i, $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{C}$, then we can write the 2-D similarity motion model as the following hyperplane in $\mathbb{C}^3$:

$$\boldsymbol{b}_i^T \boldsymbol{z} \doteq [\lambda_i R_i \quad T_i \quad 1] \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \\ -\boldsymbol{x}_2 \end{bmatrix} = 0. \tag{19}$$

Therefore, the segmentation of 2-D similarity motions is equivalent to segmenting hyperplanes in $\mathbb{C}^3$. As such, we can apply Algorithm 1 with $K = 3$ and $\Pi = [I_2 \;\; 0]$ to a collection of $N \geq M_n(3) - 1 \sim O(n^2)$ image measurements $\{z^j \in \mathbb{C}^3\}_{j=1}^N$ in general position on the hyperplanes to obtain the motion parameters $\{\boldsymbol{b}_i \in \mathbb{C}^3\}_{i=1}^n$. The original *real* motion parameters are then given as

$$\lambda_i = |\boldsymbol{b}_{i1}|, \;\; \theta_i = \angle \boldsymbol{b}_{i1}, \;\; \text{and } T_i = [\mathrm{Re}(\boldsymbol{b}_{i2}), \mathrm{Im}(\boldsymbol{b}_{i2})]^T, \quad \text{for } i = 1, \ldots, n. \tag{20}$$

3.2.2. The Case of Optical Flow. Let $\{\boldsymbol{u}^j \in \mathbb{R}^2\}_{j=1}^N$ be N measurements of the optical flow at the N pixels $\{\boldsymbol{x}^j \in \mathbb{R}^2\}_{j=1}^N$. We assume that the optical flow can be modeled as a collection of n 2-D similarity motion models as $\boldsymbol{u} = \lambda_i R_i \boldsymbol{x} + T_i$. Therefore, the segmentation of 2-D similarity motions from optical flow measurements can be solved as in the case of feature points, after replacing $\boldsymbol{x}_2 = \boldsymbol{u}$ and $\boldsymbol{x}_1 = \boldsymbol{x}$.

3.3. Segmentation of 2-D Affine Motions: Segmenting Hyperplanes in $\mathbb{C}^4$

3.3.1. The Case of Feature Points. In this case, we assume that the images are related by a collection of n 2-D affine motion models $\{A_i \in \mathbb{R}^{2\times 3}\}_{i=1}^n$. That is, for each feature pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ there exists a 2-D affine motion A_i such that

$$\boldsymbol{x}_2 = A_i \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}_i \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \end{bmatrix}. \quad (21)$$

Therefore, if we interpret $\boldsymbol{x}_2$ as a complex number $\boldsymbol{x}_2 \in \mathbb{C}$, but we still think of $\boldsymbol{x}_1$ as a vector in $\mathbb{R}^2$, then we have

$$\boldsymbol{x}_2 = \boldsymbol{a}_i^T \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \end{bmatrix} = \Big[a_{11}+a_{21}\sqrt{-1}, a_{12}+a_{22}\sqrt{-1}, a_{13}+a_{23}\sqrt{-1}\Big]_i \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \end{bmatrix}. \quad (22)$$

This equation represents the following hyperplane in $\mathbb{C}^4$

$$\boldsymbol{b}_i^T z = \begin{bmatrix} \boldsymbol{a}_i^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ 1 \\ -\boldsymbol{x}_2 \end{bmatrix} = 0, \quad (23)$$

where the normal vector $\boldsymbol{b}_i \in \mathbb{C}^4$ encodes the affine motion parameters and the data point $z \in \mathbb{C}^4$ encodes the image measurements $\boldsymbol{x}_1 \in \mathbb{R}^2$ and $\boldsymbol{x}_2 \in \mathbb{C}$. Therefore, the segmentation of 2-D affine motion models is equivalent to segmenting hyperplanes in $\mathbb{C}^4$. As such, we can apply Algorithm 1 with $K = 4$ and $\Pi = [I_3\ 0]$ to a collection of $N \geq M_n(4) - 1 \sim O(n^3)$ image measurements $\{z^j \in \mathbb{C}^4\}_{j=1}^N$ in general position to obtain the motion parameters $\{\boldsymbol{b}_i \in \mathbb{C}^4\}_{i=1}^n$. The original affine motion models are then obtained as

$$A_i = \begin{bmatrix} \mathrm{Re}(\boldsymbol{b}_{i1}) & \mathrm{Re}(\boldsymbol{b}_{i2}) & \mathrm{Re}(\boldsymbol{b}_{i3}) \\ \mathrm{Im}(\boldsymbol{b}_{i1}) & \mathrm{Im}(\boldsymbol{b}_{i2}) & \mathrm{Im}(\boldsymbol{b}_{i3}) \end{bmatrix} \in \mathbb{R}^{2\times 3}, \quad (24)$$

for $i = 1, \ldots, n$.

3.3.2. The Case of Affine Optical Flow.

In this case, the optical flow $\boldsymbol{u}$ is modeled as being generated by a collection of n affine motion models $\{A_i \in \mathbb{R}^{2\times 3}\}_{i=1}^n$ of the form

$$\boldsymbol{u} = A_i \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}. \quad (25)$$

Therefore, the segmentation of 2-D affine motions can be solved as in the case of feature points, after replacing $\boldsymbol{x}_2 = \boldsymbol{u}$ and $\boldsymbol{x}_1 = \boldsymbol{x}$.

4. 3-D Motion Segmentation

This section considers the problem of segmenting a collection of 3-D motion models from measurements of either the position of a set of feature points in two frames of a video sequence, or optical flow measurements at each pixel. We show that for the 3-D translational, 3-D rigid and 3-D homography motion models, the motion segmentation problem is equivalent to segmenting hyperplanes or bilinear forms in $\mathbb{R}^3$, $\mathbb{R}^6$ or $\mathbb{C}^5$, respectively. We develop extensions of Algorithm 1 to deal with the bilinear cases.

4.1. Segmentation of 3-D Translational Motions: Segmenting Hyperplanes in $\mathbb{R}^3$

4.1.1. The Case of Feature Points.

In this case, we assume that the scene can be modeled as a mixture of purely translational motion models, $\{T_i \in \mathbb{R}^3\}_{i=1}^n$, where T_i represents the translation (calibrated case) or the *epipole* (uncalibrated case) of object i relative to the camera between the two frames. We assume that the epipoles are different in a projective sense, i.e. they are different up to a nonzero scalar.

Given the images $\boldsymbol{x}_1 \in \mathbb{P}^2$ and $\boldsymbol{x}_2 \in \mathbb{P}^2$ of a point in object i in the first and second frames, the images and the 3-D translational motion are related by the well-known epipolar constraint for linear motions

$$-\boldsymbol{x}_2^T \widehat{T}_i \boldsymbol{x}_1 = T_i^T(\boldsymbol{x}_2 \times \boldsymbol{x}_1) = T_i^T \boldsymbol{\ell} = 0, \quad (26)$$

where $\boldsymbol{\ell} = (\boldsymbol{x}_2 \times \boldsymbol{x}_1) \in \mathbb{R}^3$ is known as the *epipolar line* associated with the image pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ and $\widehat{T} \in so(3)$ denotes the skew-symmetric matrix generating the cross product by T.

Therefore, the segmentation of 3-D translational motions is equivalent to clustering data (epipolar lines) lying in a collection of hyperplanes in $\mathbb{R}^3$ whose normal vectors are the n epipoles $\{T_i\}_{i=1}^n$. As such, we can apply Algorithm 1 with $K = 3$ and $\Pi = I_3$ to $N \geq M_n(3) - 1 \sim O(n^2)$ epipolar lines $\boldsymbol{\ell}^j = \boldsymbol{x}_1^j \times \boldsymbol{x}_2^j\}_{j=1}^N$ in general position to estimate the epipoles $\{T_i\}_{i=1}^n$ from the derivatives of the polynomial $p_n(\boldsymbol{\ell}) = (T_1^T\boldsymbol{\ell}) \cdots (T_n^T\boldsymbol{\ell})$ as

$$T_i = Dp_n(\boldsymbol{y}_i)/\|Dp_n(\boldsymbol{y}_i)\|, \qquad i = 1, \ldots, n. \quad (27)$$

Note that when choosing the points $\boldsymbol{y}_i$ in Algorithm 1 we take $\Pi = I_3$. This is because in the case of 3-D translational motions the last entry of each epipole is *not* constrained to be equal to one. In fact, the amount of translation $\|T_i\|$ is lost under perspective projection and cannot be recovered from the image measurements. Hence, we assume the norm of each epipole to be one.

An alternative method for computing the n epipoles from the N epipolar lines is to first evaluate the epipole associated with each epipolar line $\{\boldsymbol{\ell}^j\}_{j=1}^N$ as $Dp_n(\boldsymbol{\ell}^j)$ and then apply any clustering algorithm that deals with projective data to the points $\{Dp_n(\boldsymbol{\ell}^j)\}_{j=1}^N$. For example, one can apply spectral clustering using the absolute value of the angle between $Dp_n(\boldsymbol{\ell}^j)$ and $Dp_n(\boldsymbol{\ell}^{j'})$ as a pairwise distance between image pairs j and j'.

4.1.2. The Case of Optical Flow. In the case of optical flow generated by purely translating objects, we have that $\boldsymbol{u}^T\widehat{T_i}\boldsymbol{x} = 0$, where the optical flow $\boldsymbol{u}$ is augmented as a three-dimensional vector as $\boldsymbol{u} = [\mathrm{u}, \mathrm{v}, 0]^T \in \mathbb{R}^3$. Therefore, one can estimate the translations $\{T_i \in \mathbb{R}^3\}_{i=1}^n$ as before by replacing $\boldsymbol{x}_2 = \boldsymbol{u}$ and $\boldsymbol{x}_1 = \boldsymbol{x}$.

4.2. Segmentation of 3-D Rigid-Body Motions: Segmenting Bilinear Forms in $\mathbb{R}^6$

In this section, we consider the problem of segmenting multiple 3-D rigid-body motions from point correspondences in two perspective views. That is, we assume that the motion of the objects relative to the camera between the two views can be modeled as a mixture of 3-D rigid-body motions $\{(R_i, T_i) \in SE(3)\}_{i=1}^n$, where $R_i \in SO(3)$ is the relative rotation and $T_i \in \mathbb{R}^3$ is the relative translation. We assume that $T_i \neq 0$, so that we can represent each motion with a nonzero rank-2 *fundamental matrix* $F_i = \widehat{T_i}R_i \in \mathbb{R}^{3\times 3}$. We also assume that the fundamental matrices are different from each other in a projective sense.

Recall that given an image pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, there exists a motion i such that the following epipolar constraint [23] is satisfied

$$\boldsymbol{x}_2^T F_i \boldsymbol{x}_1 = 0. \tag{28}$$

Therefore, the following *multibody epipolar constraint* [38] must be satisfied by the number of independent motions n, the fundamental matrices $\{F_i\}_{i=1}^n$ and the image pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, regardless of the object to which the image pair belongs

$$p_n(\boldsymbol{x}_1, \boldsymbol{x}_2) \doteq \prod_{i=1}^n \left(\boldsymbol{x}_2^T F_i \boldsymbol{x}_1\right) = 0. \tag{29}$$

As shown in [38], this constraint can be written in bilinear form as

$$\nu_n(\boldsymbol{x}_2)^T \mathcal{F} \nu_n(\boldsymbol{x}_1) = 0, \tag{30}$$

where $\mathcal{F} \in \mathbb{R}^{M_n(3)\times M_n(3)}$ is the so-called *multibody fundamental matrix*.

When the number of motions n is known, one can linearly estimate $\mathcal{F}$ from $N \geq M_n(3)^2 - 1 \sim O(n^4)$ image pairs in general position by solving the linear system $L_n \boldsymbol{f} = 0$, where $\boldsymbol{f} \in \mathbb{R}^{M_n(3)^2}$ is the stack of the rows of $\mathcal{F}$ and $L_n \in \mathbb{R}^{N\times M_n(3)^2}$ is a matrix whose jth row is $(\nu_n(\boldsymbol{x}_2^j) \otimes \nu_n(\boldsymbol{x}_1^j))^T$ with $\otimes$ the Kronecker product. When n is unknown, one can estimate n as [38]

$$n = \min\{i : \operatorname{rank}(L_i) = M_i(3)^2 - 1\}, \tag{31}$$

where L_i is computed using the Veronese map of degree i. However, in the presence of noise in the image measurements, we cannot directly estimate n from (31), because the matrix L_i may be full rank for all $i \geq 1$. Following (8), we determine the number of motions from noisy data as

$$n = \arg\min_i \left\{ \frac{\sigma^2_{M_i^2(3)}(L_i)}{\sum_{k=1}^{M_i^2(3)-1} \sigma_k^2(L_i)} + \kappa\, M_i^2(3) \right\}, \tag{32}$$

where $\sigma_k(L_i)$ is the kth singular value of L_i and $\kappa > 0$ is a (weighting) parameter. Given n, we compute $\mathcal{F}$ as the least-squares solution to $L_n \boldsymbol{f} = 0$.

Given n and $\mathcal{F}$, we now show how to estimate the individual fundamental matrices $\{F_i\}_{i=1}^n$ by taking derivatives of the multibody epipolar constraint. Recall that, given a point $\boldsymbol{x}_1 \in \mathbb{P}^2$ in the first image frame, the epipolar lines associated with it are defined as $\boldsymbol{\ell}_i \doteq F_i\boldsymbol{x}_1 \in \mathbb{R}^3$, $i = 1, \ldots, n$. Therefore, if the image pair $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ corresponds to motion i, i.e. if $\boldsymbol{x}_2^T F_i \boldsymbol{x}_1 = 0$, then

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}_2} \nu_n(\boldsymbol{x}_2)^T \mathcal{F} \nu_n(\boldsymbol{x}_1) &= \sum_{i=1}^n \prod_{\ell\neq i} \left(\boldsymbol{x}_2^T F_\ell \boldsymbol{x}_1\right)(F_i\boldsymbol{x}_1)\\ &= \prod_{\ell\neq i} \left(\boldsymbol{x}_2^T F_\ell \boldsymbol{x}_1\right)(F_i\boldsymbol{x}_1) \sim \boldsymbol{\ell}_i.\end{aligned} \tag{33}$$

In other words, the partial derivative of the multibody epipolar constraint with respect to $\boldsymbol{x}_2$ evaluated at $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is proportional to *the* epipolar line associated with $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ in the second view. Similarly, the partial derivative of the multibody epipolar constraint with respect to $\boldsymbol{x}_1$ evaluated at $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is proportional to *the* epipolar line associated with $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ in the first view. Therefore, given a set of image pairs $\{(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j)\}_{j=1}^N$ and the multibody fundamental matrix $\mathcal{F} \in \mathbb{R}^{M_n(3)\times M_n(3)}$, we can estimate a collection of epipolar lines $\{\boldsymbol{\ell}^j\}_{j=1}^N$ associated with each image pair.[4] As described in Section 4.1, this collection of epipolar lines must pass through the n epipoles $\{T_i\}_{i=1}^n$. Therefore, if the n epipoles are different in a projective sense,[5] we can apply Algorithm 1 with $K = 3$ and $\Pi = I_3$ to the epipolar lines $\{\boldsymbol{\ell}^j\}_{j=1}^N$ to obtain the n epipoles $\{T_i\}_{i=1}^n$ up to a scale factor, as in Eq. (27). We can then compute the n fundamental matrices $\{F_i\}_{i=1}^n$ by assigning the image pair $(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j)$ to group i if $i = \arg\min_{\ell=1,\ldots,n}(T_i^T \boldsymbol{\ell}^j)^2$ and then applying the eight-point algorithm to the image pairs in group $i = 1, \ldots, n$.

4.3. Segmentation of 3-D Homographies: Segmenting Bilinear Forms in $\mathbb{C}^5$

The motion segmentation scheme described in the previous section assumes that the displacement of each object between the two views relative to the camera is nonzero, i.e. $T_i \neq 0$. Otherwise, the individual fundamental matrices are zero, hence the motions cannot be segmented. Furthermore, the segmentation scheme also requires that the 3-D points be in general configuration. Otherwise, one cannot uniquely recover each fundamental matrix from its epipolar constraint. The latter case occurs, for example, in the case of a planar structure, i.e. when the 3-D points lie in a plane, as shown in [14].

Both in the case of a purely rotating object (with respect to the camera center) or in the case of a planar 3-D structure, the motion model between the two views $\boldsymbol{x}_1 \in \mathbb{P}^2$ and $\boldsymbol{x}_2 \in \mathbb{P}^2$ can be described with a *homography* matrix $H \in \mathbb{R}^{3\times 3}$ that results in the following *homography constraint* [14]

$$\boldsymbol{x}_2 \sim H\boldsymbol{x}_1 \doteq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \boldsymbol{x}_1. \tag{34}$$

Therefore, in this section we consider the problem of segmenting a scene whose 3-D motion can be modeled with n different homographies $\{H_i\}_{i=1}^n$. Note that in this case the n homographies do *not* necessarily correspond to n different rigid-body motions. This is because it could be the case that one rigidly moving object consists of two or more planes, hence its rigid-body motion will lead to two or more homographies. Therefore, the n homographies can represent anything from 1 up to n rigid-body motions.

An important difference between segmentation of fundamental matrices and segmentation of homography matrices is that we cannot take the product of the individual homography constraints, as we did in (29) with the epipolar constraints, because (34) yields two linearly independent equations per image pair. In principle, one could resolve this difficulty by considering a line $\boldsymbol{\ell}_2$ passing through the image point in the second view $\boldsymbol{x}_2$, i.e. $\boldsymbol{\ell}_2^T \boldsymbol{x}_2 = 0$, so that the homography constraint can be rewritten as a single equation $\boldsymbol{\ell}_2^T H \boldsymbol{x}_1 = 0$. This approach indeed leads to a method for computing a multibody homography $\mathcal{H}$ analogous to the multibody fundamental matrix $\mathcal{F}$. However, it is unclear how to factorize such $\mathcal{H}$ into the individual homographies $\{H_i\}_{i=1}^n$. In this section, we resolve this difficulty by working in the complex domain.

4.3.1. Complexification of Homographies. We interpret $\boldsymbol{x}_2 \in \mathbb{P}^2$ as a point in $\mathbb{CP}$ by considering the first two coordinates of $\boldsymbol{x}_2$ as a complex number and appending a one to it. However, we still think of $\boldsymbol{x}_1$ as a point in $\mathbb{P}^2$. With this interpretation, we can rewrite (34) as

$$\begin{aligned} &\boldsymbol{x}_2 \sim H^{1,2}\boldsymbol{x}_1 \\ &\doteq \begin{bmatrix} h_{11}+h_{21}\sqrt{-1} & h_{12}+h_{22}\sqrt{-1} & h_{13}+h_{23}\sqrt{-1} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \boldsymbol{x}_1, \end{aligned} \tag{35}$$

where $H^{1,2} \in \mathbb{C}^{2\times 3}$ now represents a *complex homography*[6] obtained by complexifying the first two rows of H (as indicated by the superscripts). Let $\boldsymbol{w}_2$ be the vector in $\mathbb{CP}$ perpendicular to $\boldsymbol{x}_2$, i.e. if $\boldsymbol{x}_2 = [\begin{smallmatrix} z \\ 1 \end{smallmatrix}]$ then $\boldsymbol{w}_2 = [\begin{smallmatrix} 1 \\ -z \end{smallmatrix}]$. Then we can rewrite (35) as the following *complex* bilinear constraint

$$w_2^T H^{1,2} \boldsymbol{x}_1 = 0, \tag{36}$$

which we call the *complex homography constraint*.

Thanks to (36), we can interpret the motion segmentation problem as one in which we are given image data $\{\boldsymbol{x}_1^j \in \mathbb{P}^2\}_{j=1}^N$ and $\{\boldsymbol{w}_2^j \in \mathbb{CP}\}_{j=1}^N$ related by a collection of n complex homographies $\{H_i^{1,2} \in \mathbb{C}^{2\times 3}\}_{i=1}^n$. Then each image pair $(\boldsymbol{x}_1, \boldsymbol{w}_2)$ has to satisfy the *multibody homography constraint*

$$\prod_{i=1}^n \left(\boldsymbol{w}_2^T H_i^{1,2} \boldsymbol{x}_1\right) = \nu_n(\boldsymbol{w}_2)^T \mathcal{H}^{1,2} \nu_n(\boldsymbol{x}_1) = 0, \tag{37}$$

regardless of which complex homography is associated with the image pair. We call the matrix $\mathcal{H}^{1,2} \in \mathbb{C}^{M_n(2)\times M_n(3)}$ the *multibody homography*.

Now, since the multibody homography constraint (37) is linear in the multibody homography $\mathcal{H}^{1,2}$, when n is known we can linearly solve for $\mathcal{H}^{1,2}$ from (37) given $N \geq M_n(2)M_n(3) - (M_n(3)+1)/2 \sim O(n^3)$ image pairs in general position.[7] When n is unknown, one can estimate it as[8]

$$n = \min\{i : \operatorname{rank}(L_i) = M_i(2)M_i(3) - 1\}, \quad (38)$$

where $L_i \in \mathbb{C}^{N\times M_i(3)M_i(2)}$ is a matrix whose jth row is $(\nu_i(\boldsymbol{w}_2^j) \otimes \nu_i(\boldsymbol{x}_1^j))^T$. As before, in the presence of noise in the image measurements, we determine the number of motions from noisy data as

$$n = \arg\min_i \left\{ \frac{\sigma^2_{M_i(2)M_i(3)}(L_i)}{\sum_{k=1}^{M_i(2)M_i(3)-1} \sigma_k^2(L_i)} + \kappa\, M_i(2)M_i(3) \right\}, \quad (39)$$

where $\sigma_k(L_i)$ is the kth singular value of L_i and $\kappa > 0$ is a parameter.

4.3.2. Decomposition of the Multibody Homography. Given the multibody homography $\mathcal{H}^{1,2} \in \mathbb{C}^{M_n(2)\times M_n(3)}$, the rest of the problem is to recover the individual homographies $\{H_i^{1,2}\}_{i=1}^n$ or $\{H_i\}_{i=1}^n$. In the case of fundamental matrices discussed in Section 4.2, the key for solving the problem was the fact that fundamental matrices are of rank 2, hence one can cluster epipolar lines based on the epipoles. Note that here we cannot do the same with real homographies $H_i \in \mathbb{R}^{3\times3}$, because in general they are full rank. Nevertheless, if we work with the complex homographies $H_i^{1,2} \in \mathbb{C}^{2\times3}$ instead, they automatically have rank 2. We call the only vector in the kernel of a complex homography $H^{1,2}$ its *complex epipole*, and denote it by $\boldsymbol{e}^{1,2} \in \mathbb{C}^3$. That is, we have $H^{1,2}\boldsymbol{e}^{1,2} = 0$.

For the sake of simplicity, let us first consider the case in which the complex homographies $H_i^{1,2}$ have different complex epipoles. Once the multibody homography matrix $\mathcal{H}^{1,2}$ is obtained, similarly to the case of epipolar lines of fundamental matrices (33), we can associate a *complex epipolar line*

$$\boldsymbol{\ell}^j \sim \left.\frac{\partial \nu_n(\boldsymbol{w}_2)^T \mathcal{H}^{1,2} \nu_n(\boldsymbol{x}_1)}{\partial \boldsymbol{x}_1}\right|_{(\boldsymbol{x}_1,\boldsymbol{w}_2)=(\boldsymbol{x}_1^j,\boldsymbol{w}_2^j)} \in \mathbb{CP}^2 \quad (40)$$

with each image pair $(\boldsymbol{x}_1^j, \boldsymbol{w}_2^j)$. Given this set of $N \geq M_n(3)-1$ complex epipolar lines $\{\boldsymbol{\ell}^j\}_{j=1}^N$ in general position, we can apply Algorithm 1 with $K = 3$ and $\Pi = I_3$ to estimate the n complex epipoles $\{\boldsymbol{e}_i^{1,2} \in \mathbb{C}^3\}_{i=1}^n$ up to a scale factor. Since the n complex epipoles are different, we can cluster the original image measurements by assigning image pair $(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j)$ to group i if $i = \arg\min_{\ell=1,\dots,n} |\boldsymbol{e}_\ell^T \boldsymbol{\ell}^j|^2$. Once the image pairs have been clustered, the estimation of each homography becomes a simple linear problem.

Remark 1 (*Direct Extraction of Homographies from* $\mathcal{H}^{1,2}$). There is yet another way to obtain individual H_i from $\mathcal{H}^{1,2}$ without segmenting the image pairs first. Once the complex epipoles $\boldsymbol{e}_i^{1,2}$ are known, one can compute the following linear combination of the rows of $H_i^{1,2}$ (up to scale) from the derivatives of the multibody homography constraint at $\boldsymbol{e}_i^{1,2}$

$$\boldsymbol{w}^T H_i^{1,2} \sim \left.\frac{\partial \nu_n(\boldsymbol{w})^T \mathcal{H}^{1,2} \nu_n(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{e}_i^{1,2}} \in \mathbb{CP}^2, \forall \boldsymbol{w} \in \mathbb{C}^2. \quad (41)$$

In particular, if we take $\boldsymbol{w} = [1, 0]^T$ and $\boldsymbol{w} = [0, 1]^T$ we obtain the first and second row of $H_i^{1,2}$ (up to scale), respectively. By choosing additional $\boldsymbol{w}$'s one obtains more linear combinations from which the rows of H_i can be linearly and uniquely determined.

4.3.3. Epipoles of Complex Homographies. The algorithm presented in the previous subsection assumes that the n complex epipoles are different. However, two different real homographies may have the same complex epipole (see Example 1). In fact, one can show that the set of complex homographies that share the same epipole $\boldsymbol{e}^{1,2}$ is a five-dimensional subset (hence a zero-measure subset) of all real homography matrices. We then want to know under what conditions the complex epipoles are guaranteed to be different. The following lemma gives a condition.

Lemma 3. *If the third rows of two real non-singular homography matrices H_1 and $H_2 \in \mathbb{R}^{3\times3}$ are different (in a projective sense) then the associated complex epipoles $\boldsymbol{e}_1^{1,2}$ and $\boldsymbol{e}_2^{1,2} \in \mathbb{C}^3$ must be different (in a projective sense).*

Proof: Let H_1 and $H_2 \in \mathbb{R}^{3\times3}$ be two different homographies and let $\boldsymbol{h}_1^T$ and $\boldsymbol{h}_2^T \in \mathbb{R}^3$ be their respective third rows. Suppose that the two homographies share the same complex epipole $\boldsymbol{e}$, i.e. $H_1^{1,2}\boldsymbol{e} = H_2^{1,2}\boldsymbol{e} = 0$. Then, the complexifications of the first two rows of H_1 and H_2 are orthogonal to $\boldsymbol{e}$, hence they must be in the (complex) plane spanned by $\boldsymbol{h}_1^T$ and $\boldsymbol{h}_2^T$. Therefore, all the three rows of H_1 or H_2 are linearly dependent on $\boldsymbol{h}_1^T$ and $\boldsymbol{h}_2^T$. This contradicts the assumption that H_1 and H_2 are non-singular. □

Example 1 (One-Motion/Multi-Planes–Multi-Motions/One-Plane). A homography is generally of the form $H = R + T\pi^T$, where (R, T) is the camera motion and π is the plane normal. If the homographies come from different planes (different π) undergoing the same rigid-body motion with $T_z \neq 0$, then the associated complex epipoles will always be different since their third rows depend on π_i^T. However, if one plane with the normal vector $\pi = [0, 0, 1]^T$ undergoes different translational motions of the form $T_i = [T_{ix}, T_{iy}, T_{iz}]^T$, then all the complex epipoles are equal to $\boldsymbol{e}_i^{1,2} = [\sqrt{-1}, -1, 0]^T$. To avoid this problem, one can complexify the first and third rows of H instead of the first two. The new complex epipoles will be $\boldsymbol{e}_i^{1,3} = [T_{ix} + T_{iz}\sqrt{-1}, T_{iy}, -1]^T$, which in general are different for different translational motions.

Unfortunately, the condition in Lemma 3 is sufficient, but not necessary, as shown by the following example.

Example 2 (Complex Epipole of a Rotational Homography). Suppose that a homography H is induced from a rotation, i.e. $H = R = [\boldsymbol{r}_1^T; \boldsymbol{r}_2^T; \boldsymbol{r}_3^T] \in SO(3)$. The complexification gives two row vectors $\boldsymbol{r}_1^T + \sqrt{-1}\,\boldsymbol{r}_2^T$ and $\boldsymbol{r}_3^T$. It is easy to check that the complex epipole is $\boldsymbol{e} = \boldsymbol{r}_1 + \sqrt{-1}\,\boldsymbol{r}_2$, which is orthogonal to both vectors. This shows that Lemma 3 is only sufficient but not necessary, because rotations in the XY-plane share the same last row [0, 0, 1] but in general they lead to different complex epipoles.

In order to find a condition that is both necessary and sufficient, let $H^{1,2}, H^{2,3}, H^{1,3} \in \mathbb{C}^{2\times3}$ be the three different complex homographies associated with a real homography matrix $H \in \mathbb{R}^{3\times3}$ obtained by complexifying rows (1, 2), (2, 3), and (1, 3), respectively. Let $\boldsymbol{e}^{1,2}, \boldsymbol{e}^{2,3}, \boldsymbol{e}^{1,3} \in \mathbb{C}^3$ be the three corresponding complex epipoles. We have the following result.

Theorem 1 *(Complex Epipoles of Real Homographies). Two non-singular real homography matrices H_1 and $H_2 \in \mathbb{R}^{3\times3}$ are different (in a projective sense) if and only if they have different sets of complex epipoles $(\boldsymbol{e}^{1,2}, \boldsymbol{e}^{2,3}, \boldsymbol{e}^{1,3})$.*

Proof: The sufficiency is obvious according to the definition of the complex epipoles. We only have to show the necessity and we show it by contradiction. Assume that the two sets of complex epipoles are the same up to scale. According to Lemma 3, each of the three rows of H_1 and H_2 must be equal up to a (probably different) scale. That is $H_2 = DH_1$ for some diagonal matrix $D \doteq \text{diag}\{d_1, d_2, d_3\} \in \mathbb{R}^{3\times3}$. Let $\boldsymbol{h}_1^T, \boldsymbol{h}_2^T, \boldsymbol{h}_3^T$ be the three rows of H_1. If $d_1 \neq d_2$, the two vectors $(d_1\boldsymbol{h}_1^T + \sqrt{-1}d_2\boldsymbol{h}_2^T, \boldsymbol{h}_3^T)$ span a different plane in $\mathbb{C}^3$ from that spanned by $(\boldsymbol{h}_1^T + \sqrt{-1}\,\boldsymbol{h}_2^T, \boldsymbol{h}_3^T)$. Otherwise, we have

$$d_1\boldsymbol{h}_1^T + \sqrt{-1}d_2\boldsymbol{h}_2^T = \alpha\big(\boldsymbol{h}_1^T + \sqrt{-1}\boldsymbol{h}_2^T\big) + \beta\boldsymbol{h}_3^T$$

for some α, β not identically zero. This gives $\alpha = d_2$ and $(d_1 - d_2)\boldsymbol{h}_1^T = \beta\boldsymbol{h}_3^T$, which contradicts that the matrix H_1 is non-singular. Thus, the two epipoles $\boldsymbol{e}_1^{1,2}$ and $\boldsymbol{e}_2^{1,2}$ must be different. Therefore, in order for the sets of epipoles to coincide, we must have $d_1 = d_2 = d_3$. That is, H_1 and H_2 are equal in the projective sense. □

Theorem 1 guarantees that two different homographies will have two different epipoles for some complexification. However, if we are given $n \geq 3$ different homograhies, it could still be the case that none of the three complexifications results in n different complex epipoles. In order to handle this rare degenerate case, we can first apply our motion segmentation algorithm to each one of the three complexifications, thus obtaining three possible groupings of the image measurements. The number of groups may be strictly less than n for each one of the three groupings. In the case of perfect data, the correct grouping into n motions can be obtained by assigning two image pairs to the same motion if and only if they belong to the same group for each one of the three groupings. In the case of noisy image measurements, one needs to combine multiple segmentations into a single one, e.g., by merging the probabilities of membership to each group using Bayes rule.[9]

5. Experiments on Real and Synthetic Images

In this section, we evaluate our motion segmentation algorithms on both real and synthetic data. We compare our results with those of existing algebraic motion segmentation methods and use our algorithms to initialize iterative techniques.

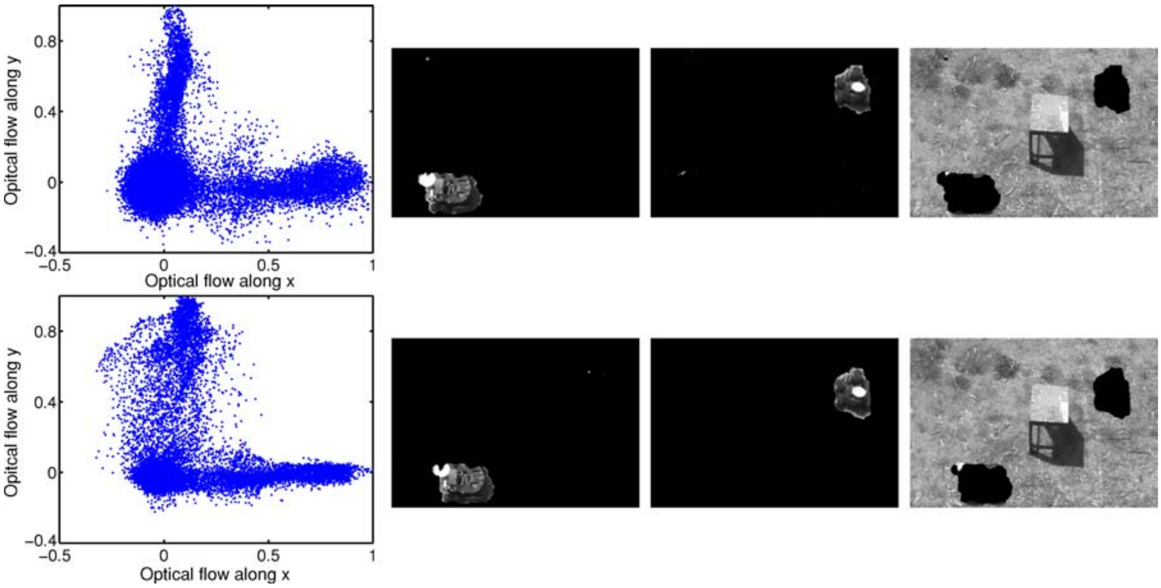
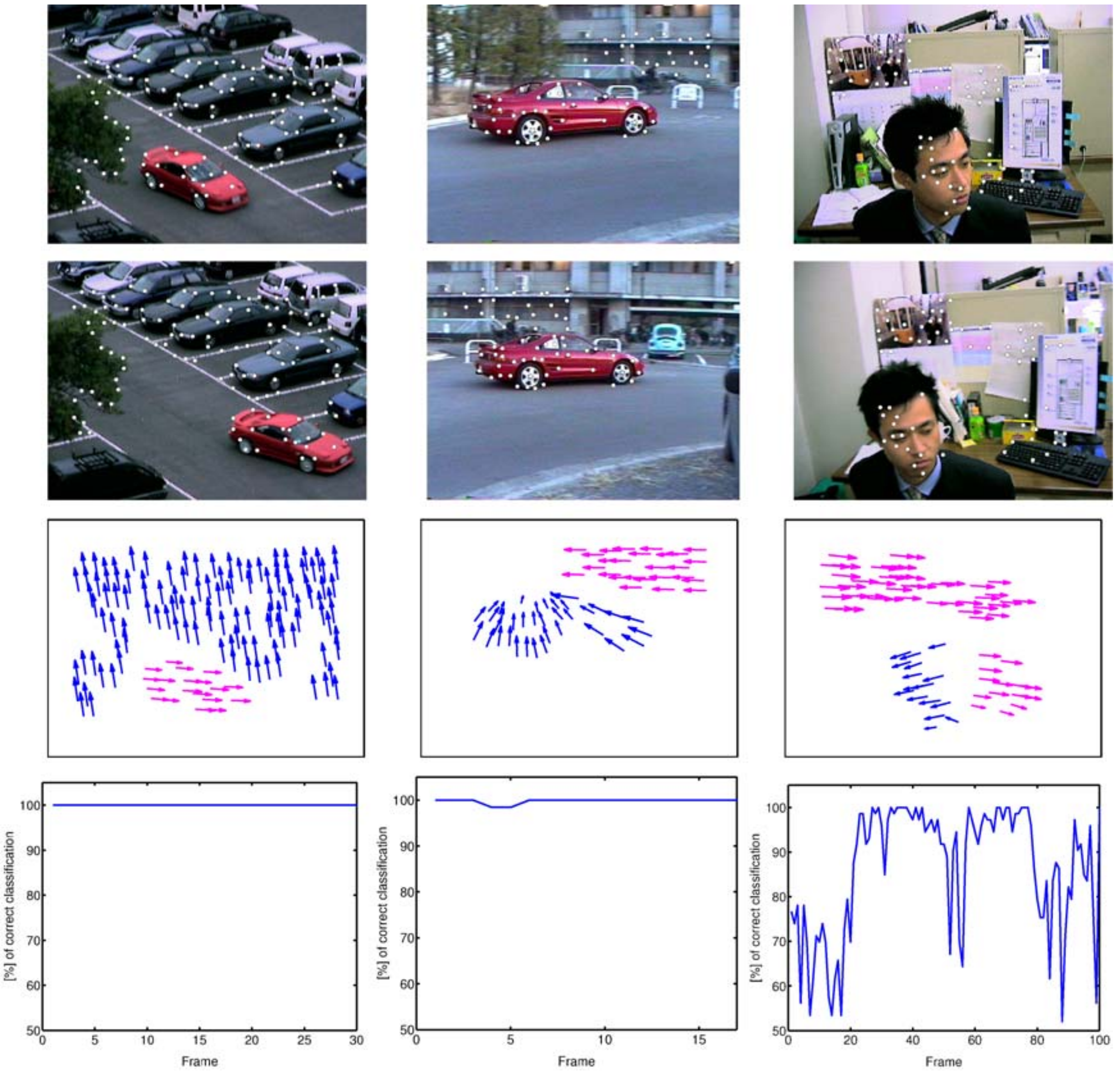


Figure 1. Segmenting the optical flow of the two-robot sequence by clustering lines in $\mathbb{C}^2$.

5.1. 2-D Translational Motions

We first test our polynomial differentiation algorithm (PDA) on a 12-frame video sequence consisting of an aerial view of two robots moving on the ground. The robots are purposely moving slowly, so that it is harder to distinguish their optical flow from the noise. At each frame, we apply Algorithm 1 with $K = 2$, $\Pi = [1\ \ 0]$, $\kappa = 10^{-6}$ and $\delta = 0.02$ to the optical flow of all $N = 240 \times 352$ pixels in the image. We compute the optical flow using Black's code, which is available at `http://www.cs.brown.edu/people/black/ignc.html`. The leftmost column of Fig. 1 displays the x and y coordinates of the optical flow for frames 4 and 10, showing that it is not so simple to distinguish the three clusters from the raw data. The remaining columns of Fig. 1 show the segmentation of the image pixels into three 2-D translational motion models. The motion of the two robots and that of the background are correctly segmented.

We also test our algorithm on two outdoor sequences taken by a moving camera tracking a car moving in front of a parking lot and a building (sequences A and B), and one indoor sequence taken by a moving camera tracking a person moving his head (sequence C), as shown in Fig. 2. The data for these sequences are taken from [21] and consist of point correspondences in multiple views, which are available at `http://www.suri.it.okayama-u.ac.jp/data.html`. For each pair of consecutive frames we apply Algorithm 1 with $K = 2$, $\Pi = [1\ \ 0]$ and $\delta = 0.02$ to the point correspondences. For all sequences and for every pair of frames the number of motions is correctly estimated as $n = 2$ for all values of $\kappa \in [2, 20]\, 10^{-7}$. For sequence A, our algorithm gives a perfect segmentation for all pairs of frames. For sequence B, our algorithm gives a perfect segmentation for all pairs of frames, except for 2 frames in which one point is misclassified. The average percentage of correct classification over the 17 frames is 99.8%. For sequence C, however, our algorithm has poor performance during the first and last 20 frames. This is because for these frames the camera and head motions are strongly correlated, and the interframe motion is just a few pixels. Therefore, it is very difficult to tell the motions apart from local information. However, if we combine all pairwise segmentations into a single segmentation,[9] our algorithms gives a percentage of correct classification of 100.0% for all three sequences as shown in Table 2. Table 2 also shows results reported in [21] from existing *multiframe* algorithms for motion segmentation. The comparison is somewhat unfair, because our algorithm uses only two views at a time and a simple 2-D translational motion model, while the other algorithms use multiple frames and a rigid-body motion



(a) Sequence A

(b) Sequence B

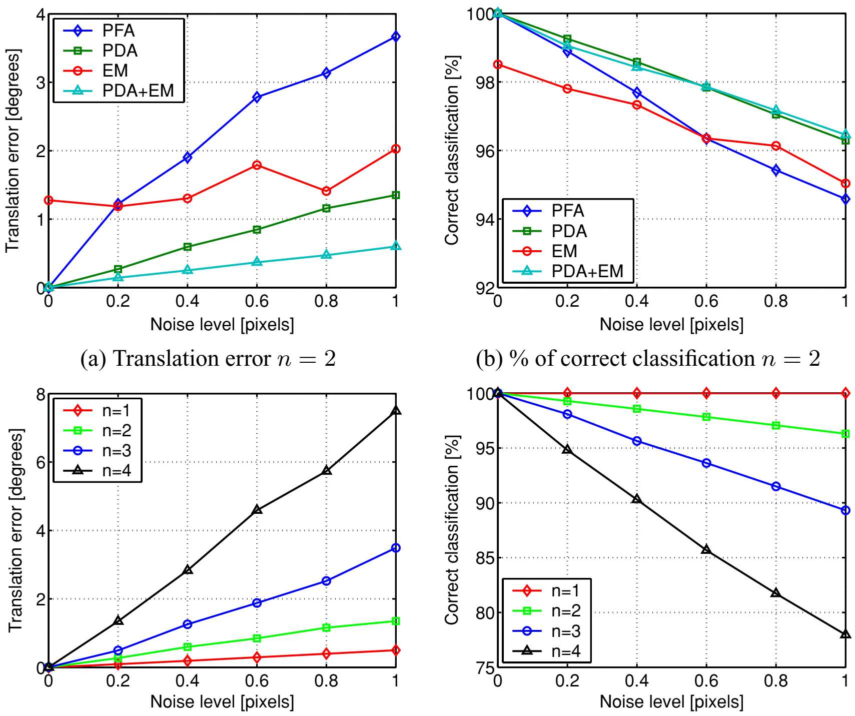
(c) Sequence C

Figure 2. Segmenting the point correspondences of sequences A, B and C for each pair of consecutive frames by clustering lines in $\mathbb{C}^2$. First row: first frame of the sequence with point correspondences superimposed. Second row: last frame of the sequence with point correspondences superimposed. Third row: displacement of the correspondences between first and last frames. Fourth row: percentage of correct classification for each pair of consecutive frames.

model for affine cameras. Furthermore, our algorithm is purely algebraic, while the others use iterative refinement to deal with noise. Nevertheless, the only algorithm having a comparable performance to ours is Kanatani's multi-stage optimization algorithm [21], which is based on solving a series of EM-like iterative optimization problems, at the expense of a significant increase in computation.

5.2. 3-D Translational Motions

In this section, we compare our polynomial differentiation algorithm (PDA) with the polynomial factorization algorithm (PFA) of [36] and a variation of the Expectation Maximization algorithm (EM) for segmenting hyperplanes in $\mathbb{R}^3$. For an image size of 500×500 pixels, we randomly generate two sets of points in



(a) Translation error $n = 2$

(b) % of correct classification $n = 2$

(c) Translation error n =1,...,4

(d) % of correct classification n =1,...,4

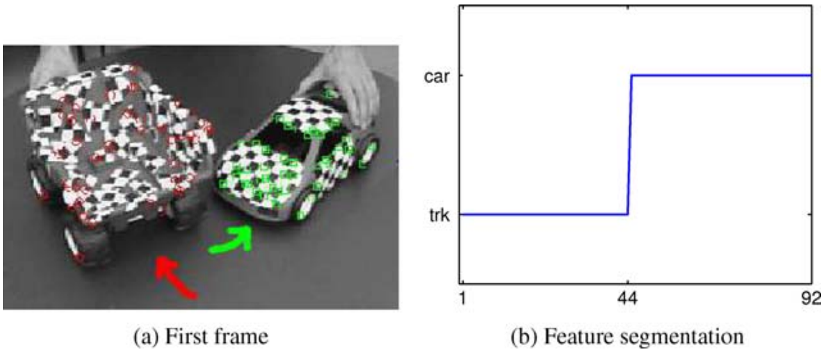
Figure 3. Segmenting 3-D translational motions by clustering planes in $\mathbb{R}^3$. Top: comparing our algorithm with PFA and EM as a function of noise in the image features. Bottom: performance of PFA as a function of the number of motions for different levels of noise.

3-D space related by multiple randomly chosen 3-D translational motions. These two sets of 3-D points are then projected onto the image plane to generate a set of point correspondences, which are then corrupted with zero-mean Gaussian noise with a standard deviation between 0 and 1 pixels.

Table 2. A comparison of the percentage of correct classification given by our two-view algebraic algorithm (PDA) with respect to that of extant multiframe optimization-based algorithms for sequences A, B, C.

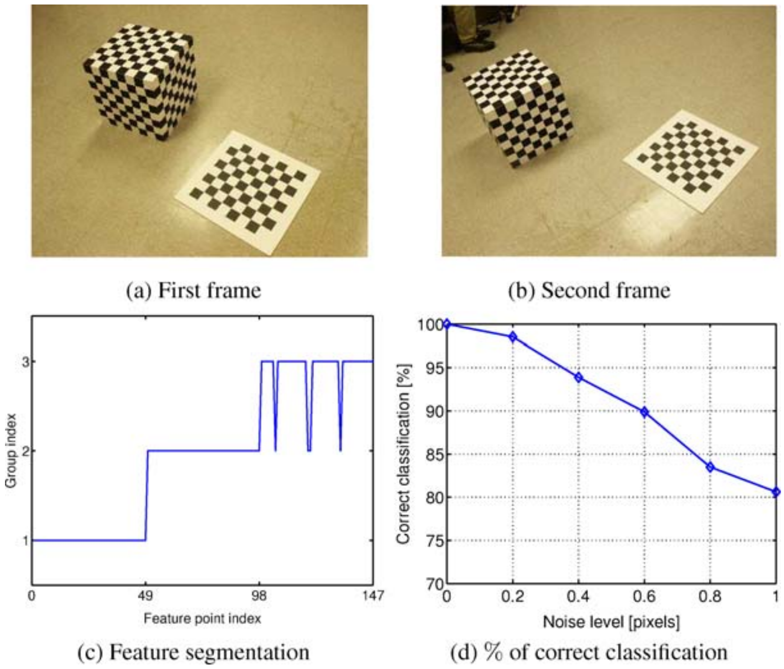
Sequence	A	B	C
Number of points	136	63	73
Number of frames	30	17	100
Costeira-Kanade	60.3%	71.3%	58.8%
Ichimura	92.6%	80.1%	68.3%
Kanatani: subspace separation	59.3%	99.5%	98.9%
Kanatani: affine subspace separation	81.8%	99.7%	67.5%
Kanatani: multi-stage optimization	100.0%	100.0%	100.0%
PDA: mean over consecutive pairs of frames	100.0%	99.8%	86.4%
PDA: including all frames	100.0%	100.0%	100.0%

Figures 3(a) and (b) show the performance of all the algorithms as a function of the level of noise for $n = 2$ moving objects. The performance measures are the mean error between the estimated and the true epipoles (in degrees), and the mean percentage of correctly segmented point correspondences using 1000 trials for each level of noise. Notice that PDA gives a translation error of less than 1.3° and a percentage of correct classification of over 96%. Therefore, PDA reduces the translation error to approximately 1/3 and improves the classification performance by about 2% with respect to PFA. Notice also that EM with the epipoles initialized at random yields a nonzero error in the noise free case, because it frequently converges to a local minimum. In fact, PDA outperforms EM when a single random initialization for EM is used. However, if we use PDA to initialize EM (PDA + EM), the performance of both EM and PDA improves, showing that our algorithm can be effectively used to initialize iterative approaches to motion segmentation. Furthermore, the number of iterations of PDA + EM is approximately 50% with respect to EM randomly initialized, hence



(a) First frame (b) Feature segmentation

Figure 4. Segmenting two 3-D translational motions by clustering planes in $\mathbb{R}^3$.



(a) First frame (b) Second frame

(c) Feature segmentation (d) % of correct classification

Figure 5. Segmenting 3-D homographies by clustering complex bilinear forms in $\mathbb{C}^5$.

there is also a gain in computing time. We also evaluate the performance of PDA as a function of the number of moving objects for different levels of noise, as shown in Figs. 3(c) and (d). As expected, the performance deteriorates with the number of moving objects, though the translation error is still below 8° and the percentage of correct classification is over 78%.

We also test the performance of PDA on a 320 × 240 video sequence containing a truck and a car undergoing two 3-D translational motions, as shown in Fig. 4(a). We apply Algorithm 1 with $K = 3$, $\Pi = I_3$ and $\delta = 0.02$ to the (real) epipolar lines obtained from a total of $N = 92$ point correspondences, 44 in the truck and 48 in the car, obtained using the tracking algorithm in [5]. The number of motions is correctly estimated as $n = 2$ for all $\kappa \in [4, 90] \cdot 10^{-4}$. Notice that PDA gives a perfect segmentation of the correspondences, as shown in Fig. 4(b). The two epipoles are estimated with an error of 5.9° for the truck and 1.7° for the car.

5.3. 3-D Homographies

In this section, we test the performance of our algorithm for segmenting rigid-body motions of planar 3-D structures, as described in Section 4.3. Figures 5(a) and (b) show two frames of a 2048×1536 video sequence with two moving objects: a cube and a checkerboard. Notice that although there are only *two* rigid-body motions, the scene contains *three* different homographies, each one associated with each one of the three visible planar structures. Furthermore, notice that the top side of the cube and the checkerboard have approximately the same normals. We manually tracked a total of $N = 147$ features: 98 in the cube (49 in each of the two visible sides) and 49 in the checkerboard. We applied our algorithm in Section 4.3 to segment the image data and obtained a 97% of correct classification, as shown in Fig. 5(c).

In order to test the performance of the algorithm as a function of noise, we further added zero-mean Gaussian noise with standard deviation between 0 and 1 pixels to the features, after rectifying the features in the second view in order to simulate the noise free case. Figure 5(d) shows the mean percentage of correct classification for 1000 trials per level of noise. The percentage of correct classification of our algorithm is between 80% and 100%, which gives a very good initial estimate for any of the existing iterative/optimization/EM based motion segmentation schemes.

6. Conclusions

We have presented a unified algebraic approach to 2-D and 3-D motion segmentation from feature point correspondences or optical flow. Contrary to extant methods, our approach does not iterate between feature segmentation and motion estimation. Instead, it computes a single multibody motion model that is satisfied by all the image measurements and then extracts the original motion models from the derivatives of the multibody one. Experiments showed that our algorithm not only outperforms existing algebraic and factorization-based methods, but also provides a good initialization for iterative techniques, such as EM, which are strongly dependent on good initialization.

Notes

1. Our ongoing work has shown that the multivariate trimming method is the most effective robust statistical technique for the estimation and segmentation of multiple motions.
2. For simplicity, we will not follow the standard definition of Hermitian transpose, which involves conjugating the entries of z.
3. This first order approximation is known in the computer vision community as the Sampson distance to an implicit surface [14].
4. Remember from Section 4.1 that in the case of purely translating objects the epipolar lines were readily obtained as $\boldsymbol{x}_1 \times \boldsymbol{x}_2$. Here the calculation is more involved because of the rotational component of the rigid-body motions.
5. Notice that this is not a strong assumption. If two individual fundamental matrices share the same (left) epipoles, one can consider the right epipoles (in the first image frame) instead, because it is extremely rare that two motions give rise to the same left and right epipoles. In fact, this happens only when the rotation axes of the two motions are equal to each other and parallel to the translation direction [38].
6. Strictly speaking, we embed each real homography matrix into an affine complex matrix.
7. The multibody homography constraint gives two equations per image pair, and there are $(M_n(2) - 1)M_n(3)$ complex entries in $\mathcal{H}^{1,2}$ and $M_n(3)$ real entries (the last row).
8. The proof is analogous to that of Lemma 1.
9. A simple, though not necessarily optimal, way of combining multiple segmentations into a single one is to compute for each segmentation the probability that an image measurement belongs to each one of the motions. Such probabilities of membership can be combined into a single one using the Bayes rule. A point is then assigned to the group yielding the maximum probability of membership. We used this method in our experiments.

References

1. S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, pp. 348–357, 2000.
2. S. Ayer and H. Sawhney, "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding," in *IEEE International Conference on Computer Vision*, 1995, pp. 777–785.
3. M. Black and P. Anandan, "Robust dynamic motion estimation over time," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 296–302.
4. T.E. Boult and L.G. Brown, "Factorization-based segmentation of motions," in *Proc. of the IEEE Workshop on Motion Understanding*, 1991, pp. 179–186.
5. A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Motion and structure causally integrated over time," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 4, pp. 523–535, 2002.
6. T. Darrel and A. Pentland, "Robust estimation of a multilayered motion representation," in *IEEE Workshop on Visual Motion*, 1991, pp. 173–178.
7. X. Feng and P. Perona, "Scene segmentation from 3D motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 225–231.
8. M.A. Fischler and R. C. Bolles, "RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 26, pp. 381–395, 1981.
9. A. Fitzgibbon and A. Zisserman, "Multibody structure and motion: 3D reconstruction of independently moving objects," in *European Conference on Computer Vision*, 2000, pp. 891–906.
10. M. Han and T. Kanade, "Reconstruction of a scene with multiple linearly moving objects," in *IEEE Conference on Com-*

puter Vision and Pattern Recognition, Vol. 2, 2000, pp. 542–549.

11. M. Han and T. Kanade, "Multiple motion scene reconstruction from uncalibrated views," in *IEEE International Conference on Computer Vision*, Vol. 1, pp. 163–170, 2001.
12. J. Harris, *Algebraic Geometry: A First Course*, Springer-Verlag, 1992.
13. R. Hartley and R. Vidal, "The multibody trifocal tensor: Motion segmentation from 3 perspective views," in *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. I, pp. 769–775, 2004.
14. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge, 2004.
15. A. Heyden and K. Åström, "Algebraic properties of multilinear constraints," *Mathematical Methods in Applied Sciences*, Vol. 20, No. 13, pp. 1135–1162, 1997.
16. M. Irani, B. Rousso, and S. Peleg, "Detecting and tracking multiple moving objects using temporal integration," in *European Conference on Computer Vision*, 1992, pp. 282–287.
17. A. Jepson and M. Black, " Mixture models for optical flow computation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1993, pp. 760–761.
18. K. Kanatani, "Motion segmentation by subspace separation and model selection," in *IEEE International Conference on Computer Vision*, 2001, Vol. 2, pp. 586–591.
19. K. Kanatani, "Evaluation and selection of models for motion segmentation," in *Asian Conference on Computer Vision*, 2002, pp. 7–12.
20. K. Kanatani and C. Matsunaga, "Estimating the number of independent motions for multibody motion segmentation," in *European Conference on Computer Vision*, 2002, pp. 25–31.
21. K. Kanatani and Y. Sugaya, "Multi-stage optimization for multi-body motion segmentation," in *Australia-Japan Advanced Workshop on Computer Vision*, 2003, pp. 335–349.
22. Q. Ke and T. Kanade, "A robust subspace approach to layer extraction," in *IEEE Workshop on Motion and Video Computing*, 2002, pp. 37–43.
23. H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, Vol. 293, pp. 133–135, 1981,
24. Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3D Vision: From Images to Geometric Models*, Springer Verlag, 2003.
25. O. Shakernia, R. Vidal, and S. Sastry, "Multi-body motion estimation and segmentation from multiple central panoramic views," in *IEEE International Conference on Robotics and Automation*, 2003, Vol. 1, pp. 571–576.
26. A. Shashua and A. Levin, "Multi-frame infinitesimal motion model for the reconstruction of (dynamic) scenes with multiple linearly moving objects," in *IEEE International Conference on Computer Vision*, 2001, Vol. 2, pp. 592–599.
27. J. Shi and J. Malik, " Motion segmentation and tracking using normalized cuts," in *IEEE International Conference on Computer Vision*, 1998, pp. 1154–1160.
28. A. Spoerri and S. Ullman, "The early detection of motion boundaries," in *IEEE International Conference on Computer Vision*, 1987, pp. 209–218.
29. P. Sturm, " Structure and motion for dynamic scenes - the case of points moving in planes," in *European Conference on Computer Vision*, 2002, pp. 867–882.
30. 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*, Vol. 23, No. 3, pp. 297–303, 2001.
31. P.H.S. Torr, "Geometric motion segmentation and model selection," *Phil. Trans. Royal Society of London*, Vol. 356, No. 1740, No. 1321–1340, 1998.
32. R. Vidal, *Imaging Beyond the Pinhole Camera*, chapter Segmentation of Dynamic Scenes Taken by a Central Panoramic Camera, LNCS. Springer Verlag, 2006.
33. R. Vidal and R. Hartley, "Motion segmentation with missing data by PowerFactorization and Generalized PCA," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, Vol. 2, pp. 310–316.
34. R. Vidal and Y. Ma, "A unified algebraic approach to 2-D and 3-D motion segmentation," in *European Conference on Computer Vision*, 2004, pp. 1–15.
35. R. Vidal, Y. Ma, and J. Piazzi, " A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004, Vol. 1, pp. 510–517.
36. R. Vidal, Y. Ma, and S. Sastry, "Generalized Principal Component Analysis (GPCA)," in *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. I, pp. 621–628, 2003.
37. R. Vidal, Y. Ma, and S. Sastry, "Generalized Principal Component Analysis (GPCA)," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 12, pp. 1–15, 2005.
38. R. Vidal, Y. Ma, S. Soatto, and S. Sastry, "Two-view multibody structure from motion," *International Journal of Computer Vision*, Vol. 68, No. 1, 2006.
39. R. Vidal and S. Sastry, "Optimal segmentation of dynamic scenes from two perspective views," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, Vol. 2, pp. 281–286.
40. R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "Segmentation of dynamic scenes from the multibody fundamental matrix," in *ECCV Workshop on Visual Modeling of Dynamic Scenes*, 2002.
41. J. Wang and E. Adelson, "Layered representation for motion analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1993, pp. 361–366.
42. Y. Weiss, "A unified mixture framework for motion segmentation: incoprporating spatial coherence and estimating the number of models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996, pp. 321–326.
43. Y. Weiss, "Smoothness in layers: Motion segmentation using nonparametric mixture estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 520–526.
44. L. Wolf and A. Shashua, "Affine 3-D reconstruction from two projective images of independently translating planes," in *IEEE International Conference on Computer Vision*, 2001, pp. 238–244.
45. L. Wolf and A. Shashua, "Two-body segmentation from two perspective views," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 263–270.
46. Y. Wu, Z. Zhang, T.S. Huang, and J.Y. Lin, " Multibody grouping via orthogonal subspace decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, Vol 2, pp. 252–257.
47. L. Zelnik-Manor and M. Irani, "Degeneracies, dependencies and their implications in multi-body and multi-sequence

factorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, Vol. 2, pp. 287–293.

René Vidal received his B.S. degree in Electrical Engineering (highest honors) from the Universidad Católica de Chile in 1997 and his M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California at Berkeley in 2000 and 2003, respectively. In 2004, he joined The Johns Hopkins University as an Assistant Professor in the Department of Biomedical Engineering and the Center for Imaging Science. He has co-authored more than 70 articles in biomedical imaging, computer vision, machine learning, hybrid systems, robotics, and vision-based control. Dr. Vidal is recipient of the 2005 NFS CAREER Award, the 2004 Best Paper Award Honorable Mention at the European Conference on Computer Vision, the 2004 Sakrison Memorial Prize, the 2003 Eli Jury Award, and the 1997 Award of the School of Engineering of the Universidad Católica de Chile to the best graduating student of the school.

Yi Ma received his two bachelors' degree in Automation and Applied Mathematics from Tsinghua University, Beijing, China in 1995. He received an M.S. degree in Electrical Engineering and Computer Science (EECS) in 1997, an M.A. degree in Mathematics in 2000, and a PhD degree in EECS in 2000 all from the University of California at Berkeley. Since August 2000, he has been on the faculty of the Electrical and Computer Engineering Department of the University of Illinois at Urbana-Champaign, where he is now an associate professor. In fall 2006, he is a visiting faculty at the Microsoft Research in Asia, Beijing, China. He has written more than 40 technical papers and is the first author of a book, entitled "An Invitation to 3-D Vision: From Images to Geometric Models," published by Springer in 2003. Yi Ma was the recipient of the David Marr Best Paper Prize at the International Conference on Computer Vision in 1999 and Honorable Mention for the Longuet-Higgins Best Paper Award at the European Conference on Computer Vision in 2004. He received the CAREER Award from the National Science Foundation in 2004 and the Young Investigator Program Award from the Office of Naval Research in 2005.