A Closed Form Solution to Direct Motion Segmentation

René Vidal and Dheeraj Singaraju
Center for Imaging Science, Johns Hopkins University
301 Clark Hall, 3400 N Charles St., Baltimore, MD, 21218, USA

Abstract

We present a closed form solution to the problem of segmenting multiple 2-D motion models of the same type directly from the partial derivatives of an image sequence. We introduce the multibody brightness constancy constraint (MBCC), a polynomial equation relating motion models, image derivatives and pixel coordinates that is independent of the segmentation of the image measurements. We first show that the optical flow at a pixel can be obtained analytically as the derivative of the MBCC at the corresponding image measurement, without knowing the motion model associated with that pixel. We then show that the parameters of the multiple motion models can be obtained from the cross products of the derivatives of the MBCC at a set of image measurements that minimize a suitable distance function. Our approach requires no feature tracking, point correspondences or optical flow, and provides a global non-iterative solution that can be used to initialize more expensive iterative approaches to motion segmentation. Experiments on real and synthetic sequences are also presented.

1. Introduction

Motion segmentation refers to the problem of fitting a collection of motion models to the image data, without knowing which data belong to which model. The mathematical nature of this problem depends largely on whether the scene is static or dynamic, on the type of image measurements that are available (image derivatives, optical flow, point correspondences), and on the type of motion and camera models relating such measurements. In this paper, we consider the segmentation of both static and dynamic scenes from measurements of the image partial derivatives related by multiple 2-D translational or 2-D affine motion models.

When the scene is static, i.e. when either the camera or a single rigid object moves, one can model the 2-D motion of the scene as 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. Various iterative and probabilistic approaches to solving this problem have been proposed in the past, such as looking for flow discontinuities [11, 2], fitting a mixture of parametric models through successive computation of dominant motions [7], clustering local motion profiles using K-means [18], or fitting a mixture of probabilistic models iteratively using EM [4, 8, 1, 19, 12]. The drawback of most 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 [9]. Another problem with iterative approaches is that their solution depends strongly on good initialization. This issue has been addressed using algebraic methods that solve the motion segmentation problem globally. [10] deals with segmenting two transparent motions by factorizing a second order homogeneous polynomial. The method needs high order derivatives of the image sequence, which cannot be computed reliably even with moderate noise. [17] deals with segmenting n affine motions by factorizing a bi-homogeneous polynomial of degree n. The method only requires first order derivatives of the image sequence, but the factorization is sensitive to noise.

When the scene is dynamic, i.e. when both the camera and multiple objects move, one can still model the scene 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 induce multiple 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. Recently, the 3-D motion segmentation has been enjoying a lot of attention in the literature, including probabilistic approaches [13] as well as geometric approaches for both affine [3, 14] and perspective [20, 15, 16, 5] cameras. However, all existing methods assume that point correspondences or optical flow measurements are available. Obtaining such measurements can be rather challenging in the presence of multiple motions.

To the best of our knowledge, although there is a lot of work on direct methods for static scenes (see [6]), our work is the first one to give an algebraic solution to 2-D motion segmentation of both static and dynamic scenes *directly* from the first-order derivatives of the image sequence.

1.1. Paper contributions

In this paper, we propose a unified algebraic approach to direct 2-D motion segmentation of static and dynamic scenes. We show that one can estimate the number of motion models, the optical flow, and the parameters of each motion model directly from the image intensities, without any need for feature tracking, point correspondences, optical flow or prior segmentation. We introduce the *multibody brightness constancy constraint* (MBCC), a polynomial equation relating the motion models, the image derivatives, and the pixel coordinates. This constraint is satisfied by all the pixels, regardless of which motion model is associated with each pixel. We show that one can compute the number of motion models as the degree of the MBCC and the optical flow at a pixel from the derivatives of the MBCC at the image measurement corresponding to that pixel. The parameters of the multiple motion models are then obtained from the cross products of the derivatives of the MBCC at a set of image measurements that minimize a suitable distance function.

This new approach to motion segmentation offers various important technical advantages over the state-of-the-art.

1. With respect to local methods, our approach has the advantage of using all the image data simultaneously to fit all motion models. Therefore, it is less sensitive to the aperture problem and to the estimation of a single motion model across motion boundaries.
2. With respect to the direct algebraic approach of [17], our approach is based on polynomial differentiation rather than polynomial factorization, which greatly improves the efficiency, accuracy and robustness of the algorithm. Furthermore, our approach also applies to other motion models, such as 2-D translational.
3. With respect to the feature-based algebraic method of [15], our approach does not need require feature tracking, point correspondences or optical flow. Indeed, optical flow is automatically computed in closed form, without knowing the motion model associated with each pixel. Furthermore, our approach does not require the image measurements to be embedded in the complex domain, which greatly simplifies the complexity of the algorithm.
4. 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. In its present form, however, the algorithm does not consider the presence of outliers in the data.

2. Direct motion segmentation

2.1. Multibody brightness constancy constraint

Consider a motion sequence taken by a moving camera observing an *unknown* number m of independently and rigidly moving objects. The 3-D motion of each object relative to the camera induces a 2-D motion field in the image plane. Because of perspective effects, depth discontinuities, occlusions, transparent motions, etc., each 3-D motion induces one or more 2-D motions. Therefore, we assume that the 2-D motion of the scene is generated from an unknown number $n \geq m$ of 2-D motion models $\{\mathcal{M}_i\}_{i=1}^n$ of the form

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_i(\boldsymbol{x}) \quad \text{if} \quad \boldsymbol{x} \in \mathcal{R}_i, \quad i = 1, \ldots, n, \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{x}) = [\mathrm{u}, \mathrm{v}, 1]^T \in \mathbb{P}^2$ is the optical flow at pixel $\boldsymbol{x} = [x_1, x_2, 1]^T \in \mathbb{P}^2$, and $\mathcal{R}_i \subset \mathbb{P}^2$ is the region of the image where the ith motion model holds.

Under the assumption that all surfaces in the scene are Lambertian, for each pixel $\boldsymbol{x}$ there exist a motion model $\mathcal{M}_i$ such that its optical flow $\boldsymbol{u}_i(\boldsymbol{x})$ can be related to the image partial derivatives $\boldsymbol{y} = [I_{x_1}, I_{x_2}, I_t]^T \in \mathbb{R}^3$ at $\boldsymbol{x}$ by the well-known *brightness constancy constraint* (BCC)

$$\mathrm{BCC}_i(\boldsymbol{x}, \boldsymbol{y}) \doteq \boldsymbol{y}^T \boldsymbol{u}_i(\boldsymbol{x}) = I_{x_1}\mathrm{u}_i + I_{x_2}\mathrm{v}_i + I_t = 0. \tag{2}$$

Therefore, the following *multibody brightness constancy constraint* (MBCC) holds at every pixel in the image

$$\mathrm{MBCC}(\boldsymbol{x}, \boldsymbol{y}) \doteq \prod_{i=1}^n (\boldsymbol{y}^T \boldsymbol{u}_i(\boldsymbol{x})) = 0. \tag{3}$$

Note that if we fix $\boldsymbol{x}$, then the MBCC is a homogeneous polynomial of degree n in $\boldsymbol{y}$, which can be written as a linear combination of the monomials $I_{x_1}^{n_1} I_{x_2}^{n_2} I_t^{n_3}$ with $n_1+n_2+n_3 = n$. If we stack these $M_n = (n+1)(n+2)/2$ independent monomials into a vector $\nu_n(\boldsymbol{y}) \in \mathbb{R}^{M_n}$, we get

$$\mathrm{MBCC}(\boldsymbol{x}, \boldsymbol{y}) = \nu_n(\boldsymbol{y})^T \mathcal{U}(\boldsymbol{x}) = \sum \mathcal{U}_{n_1,n_2,n_3}(\boldsymbol{x}) I_{x_1}^{n_1} I_{x_2}^{n_2} I_t^{n_3}.$$

The vector $\mathcal{U}(\boldsymbol{x}) \in \mathbb{R}^{M_n}$ is called the *multibody optical flow*, and $\nu_n : \mathbb{R}^3 \mapsto \mathbb{R}^{M_n}$ is called the *Veronese map* of degree n.

In the following subsections, we will demonstrate that in the case of 2-D translational motion models $\{\boldsymbol{u}_i \in \mathbb{P}^2\}_{i=1}^n$ or 2-D affine motion models $\{A_i \in \mathbb{R}^{3\times3}\}_{i=1}^n$

$$\boldsymbol{u} = \boldsymbol{u}_i \;\text{ or }\; \boldsymbol{u} = A_i \boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_{i1}^T \\ \boldsymbol{a}_{i2}^T \\ 0, 0, 1 \end{bmatrix} \boldsymbol{x} \quad i = 1, \ldots, n, \tag{4}$$

respectively, the MBCC can be expressed linearly in terms of a multibody motion model $\mathcal{M}$. By exploiting the algebraic properties of $\mathcal{M}$, we will derive an algebraic closed form solution to the following problem:

Problem 1 (Direct multiple-motion segmentation) *Given the partial derivatives $\{(I_x^j, I_y^j, I_t^j)\}_{j=1}^N$ of a motion sequence generated from n 2-D translational or 2-D affine motion models, estimate the number of motion models n, the optical flow $\boldsymbol{u}(\boldsymbol{x})$ at each pixel $\{\boldsymbol{x}^j\}_{j=1}^N$, and the model parameters $\{\mathcal{M}_i\}_{i=1}^n$, without knowing which image measurements correspond to which model.*

2.2. Computing the multibody motion model and the number of motion models

In this subsection, we show that the MBCC can be expressed linearly in terms of a multibody motion model $\mathcal{M}$ and derive a rank constraint on the image measurements, from which one can linearly estimate n and $\mathcal{M}$.

In the case of 2-D translational motions, the optical flow $\boldsymbol{u}_i(\boldsymbol{x})$ does not depend on the pixel coordinates $\boldsymbol{x}$, hence the multibody optical flow is a constant vector $\mathcal{U}$. Therefore, after evaluating the MBCC $\nu_n(\boldsymbol{y})^T\mathcal{U}=0$ at each measurement $\{\boldsymbol{y}^j\}_{j=1}^N$, we obtain the following linear system on $\mathcal{U}$

$$L_n^{\mathcal{U}}\mathcal{U} = \begin{bmatrix}\nu_n(\boldsymbol{y}^1) & \nu_n(\boldsymbol{y}^2) & \cdots & \nu_n(\boldsymbol{y}^N)\end{bmatrix}^T\mathcal{U}=0. \quad (5)$$

In the case of 2-D affine motions, the optical flow is linear in $\boldsymbol{x}\in\mathbb{P}^2$. Therefore, the MBCC is a homogeneous polynomial in each of $\boldsymbol{x}$ and $\boldsymbol{y}$ that can be written as [17]

$$\text{MBCC}(\boldsymbol{x},\boldsymbol{y}) = \nu_n(\boldsymbol{y})^T\mathcal{A}\nu_n(\boldsymbol{x}) = 0, \quad (6)$$

where $\mathcal{A}\in\mathbb{R}^{M_n\times M_n}$ is called the *multibody affine matrix*. Since equation (6) holds for all $(\boldsymbol{x}^j,\boldsymbol{y}^j)$, we obtain the following linear system on $\boldsymbol{a}$ (the stack of the columns of $\mathcal{A}$)

$$L_n^{\mathcal{A}}\boldsymbol{a} = \left[\nu_n(\boldsymbol{y}^1)\otimes\nu_n(\boldsymbol{x}^1)\cdots\nu_n(\boldsymbol{y}^N)\otimes\nu_n(\boldsymbol{x}^N)\right]^T\boldsymbol{a}=0.$$

In addition, note that $A_{(n_1,n_2,n_3),(m_1,m_2,m_3)}=0$ when $0\le m_3<n_3\le n$, because the entries (3,1) and (3,2) of each A_i are zero. After enforcing these equations we obtain

$$\tilde{L}_n^{\mathcal{A}}\tilde{\boldsymbol{a}} = 0, \quad (7)$$

where $\tilde{\boldsymbol{a}}\in\mathbb{R}^{M_n^2-Z_n}$ is equal to $\boldsymbol{a}$ without the zero entries, $Z_n = n(n+1)(n+2)(3n+5)/24$ is the number of zero entries in $\mathcal{A}$, and $\tilde{L}_n^{\mathcal{A}}\in\mathbb{R}^{N\times(M_n^2-Z_n)}$ are the columns of $L_n^{\mathcal{A}}\in\mathbb{R}^{N\times M_n^2}$ that do not correspond to the zero entries.

Since equations (5) and (7) depend explicitly on the number of motions n, in order to compute the multibody motion model $\mathcal{M}=\mathcal{U}$ or $\mathcal{M}=\mathcal{A}$ we must first determine n. To this end, we assume that the image measurements are non-degenerate, i.e. they do not satisfy any homogeneous polynomial of degree less than or equal to n other than the MBCC. This assumption is analogous to the standard assumption in structure from motion that the measurements do not live in a critical surface. Under this non-degeneracy assumption we have that:

1. There is no polynomial of degree $i<n$ that is satisfied by every data point, hence the embedded data matrices of degree i, $L_i^{\mathcal{U}}$ and $\tilde{L}_i^{\mathcal{A}}$, are of full column rank;
2. There is only one polynomial of degree n, namely the MBCC, that is satisfied by all the data, hence $L_n^{\mathcal{U}}$ and $\tilde{L}_n^{\mathcal{A}}$ are of rank M_n-1 and $M_n^2-Z_n-1$ respectively;
3. There are two or more polynomials of degree $i>n$, namely any multiple of the MBCC, that are satisfied by all the data points, hence the null space of $L_i^{\mathcal{U}}$ and $\tilde{L}_i^{\mathcal{A}}$ is at least two-dimensional.

Table 1. Minimum number of image measurements as a function of the number of motion models.

n	1	2	3	4	5	10
2-D Translational	2	5	9	14	20	65
2-D Affine	6	24	64	139	265	2430

Therefore, if $N\ge M_n-1$ or $N\ge M_n^2-Z_n-1$ image measurements are given, then the number of 2-D translational and 2-D affine motion models can be obtained as

$$n = \min\{i\in\mathbb{N}:\text{rank}(L_i^{\mathcal{U}}) = M_i-1\} \quad \text{and} \quad (8)$$

$$n = \min\{i\in\mathbb{N}:\text{rank}(\tilde{L}_i^{\mathcal{A}}) = M_i^2-Z_i-1\}, \quad (9)$$

respectively. Table 1 gives numeric values for the minimum number of points needed to estimate the number of motions. Notice that for 10 motion models less than 2500 pixels are needed, which is feasible even with a 100 × 100 image.

Note that formulae (8) and (9) are valid only for noise-free measurements, because with noisy data the matrices $L_i^{\mathcal{U}}$ and $\tilde{L}_i^{\mathcal{A}}$ may be full rank even if $i\ge n$. In this case, we use model selection to determine the number of models. Let $L_i\in\mathbb{R}^{N\times r_i}$ be $L_i^{\mathcal{U}}$ or $L_i^{\mathcal{A}}$, with $r_i=M_i$ or $M_i^2-Z_i$, respectively. We determine the number of motions as

$$n = \arg\min_{i=1,2,\dots}\left\{\frac{\sigma_{r_i}^2(L_i)}{\sum_{k=1}^{r_i-1}\sigma_k^2(L_i)}+\kappa r_i\right\} \quad (10)$$

where $\sigma_k(L_i)$ is the kth eigenvalue of L_i, and $\kappa>0$ is a parameter. The first term in (10) measures how close the matrix L_i is to dropping rank by one, and the second term penalizes choosing a large number of motions.

Once n is known, we can solve for $\mathcal{U}$ uniquely from (5) by enforcing its M_nth entry to be one, because the last entry of each $\boldsymbol{u}_i$ is one. Similarly, we can solve for $\mathcal{A}$ uniquely from (7) by enforcing its (M_n,M_n)th entry to be one.

2.3. Computing the optical flow at each pixel

Given n and $\mathcal{M}$, we can easily compute the optical flow $\boldsymbol{u}(\boldsymbol{x})$ at each pixel in closed form, without knowing which motion model is associated with each pixel. To this end, notice that for each pixel $\boldsymbol{x}$ there is a $k=1,\dots,n$ such that $\boldsymbol{y}^T\boldsymbol{u}_k(\boldsymbol{x})=0$, hence $\prod_{\ell\ne i}(\boldsymbol{y}^T\boldsymbol{u}_\ell(\boldsymbol{x}))=0$ for all $i\ne k$. Therefore, if pixel $\boldsymbol{x}$ is associated with the kth motion model only, its optical flow can be obtained from

$$\frac{\partial\text{MBCC}(\boldsymbol{x},\boldsymbol{y})}{\partial\boldsymbol{y}} = \sum_{i=1}^n\boldsymbol{u}_i(\boldsymbol{x})\prod_{\ell\ne i}(\boldsymbol{y}^T\boldsymbol{u}_\ell(\boldsymbol{x}))\sim\boldsymbol{u}_k(\boldsymbol{x}), \quad (11)$$

after normalizing its third entry to be equal to one. Note that (11) does not apply to pixels coming from two or more motion models, as in this case the MBCC has a repeated factor, hence its derivative is zero. Notice also that (11) computes the optical flow *globally* from $\mathcal{M}$, thus eliminating the aperture problem. Even if $\mathcal{M}$ is computed locally, our method can deal with multiple motions, thus it does not suffer from estimating a single motion model across motion boundaries.

2.4. Segmenting the multibody motion model

In the case of 2-D translational motions, we can obtain the n motion models $\{\boldsymbol{u}_i\}_{i=1}^n$ by computing the optical flow at every pixel using (11), and then applying any 2-D clustering algorithm to obtain n values for the optical flow. Unfortunately, with noisy data we may not be able to estimate the optical flow reliably at every pixel, which may seriously affect the clustering results, thus the estimation of the models.

An alternative method is to first choose n pixels $\{\boldsymbol{x}_i\}_{i=1}^n$ with reliable optical flow and then evaluate the optical flow at these n pixels. Under the assumption of zero-mean Gaussian noise in $\boldsymbol{y}$ with covariance $\Lambda \in \mathbb{R}^{3\times 3}$ we can choose $\boldsymbol{x}_n$ as the pixel that minimizes the negative log-likelihood of the associated generative model. A first-order approximation of the negative log-likelihood is given by[1]

$$d_n^2(\boldsymbol{x},\boldsymbol{y}) = \frac{|\mathrm{MBCC}(\boldsymbol{x},\boldsymbol{y})|^2}{\|\Lambda \frac{\partial \mathrm{MBCC}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{y}}\|^2}. \tag{12}$$

We choose the remaining $n-1$ pixels recursively as proposed in [15]. Assume we have computed pixels $\boldsymbol{x}_n,\ldots,\boldsymbol{x}_i$ by minimizing $d_n^2,\ldots,d_i^2$, respectively. A pixel $\boldsymbol{x}_{i-1}$ associated with one of the remaining $i-1$ models is chosen by minimizing

$$d_{i-1}^2(\boldsymbol{x},\boldsymbol{y}) = \frac{d_i^2(\boldsymbol{x},\boldsymbol{y})}{\frac{|\boldsymbol{y}^T\boldsymbol{u}(\boldsymbol{x}_i)|^2}{\|\Lambda \boldsymbol{u}(\boldsymbol{x}_i)\|^2}}. \tag{13}$$

Note that in choosing the pixels there is no optimization involved. We just need to evaluate the distance functions at each pixel and choose the one giving the minimum distance. Once the n pixels and their models have been obtained, we assign pixel j to model $i = \arg\min_{\ell=1,\ldots,n} \frac{(\boldsymbol{u}_\ell^T\boldsymbol{y}^j)^2}{\|\Lambda\boldsymbol{u}_\ell\|^2}$.

In the case of 2-D affine motion models, we have

$$\nu_n(\boldsymbol{y})^T\mathcal{A}\nu_n(\boldsymbol{x}) = (\boldsymbol{y}^TA_1\boldsymbol{x})(\boldsymbol{y}^TA_2\boldsymbol{x})\cdots(\boldsymbol{y}^TA_n\boldsymbol{x}). \tag{14}$$

Since $\mathcal{A}$ is known, we can see that computing the affine matrices $\{A_i\}_{i=1}^n$ is equivalent to factoring the MBCC, a bi-homogeneous polynomial of degree n, into a product of n bilinear factors. A factorization algorithm can be found in [17]. However, with noisy data the linear estimate of $\mathcal{A}$ may not necessarily factor as a product of bilinear forms. Even if $\mathcal{A}$ is factorizable, the factorization process is sensitive to noise, especially when two or more factors are similar.

In this section we propose a new solution to the factorization of $\mathcal{A}$ that does not require polynomial factorization. Instead, we exploit the geometric properties of $\mathcal{A}$ to obtain the following purely geometric solution for computing $\{A_i\}_{i=1}^n$

1. Compute derivatives of the MBCC with respect to $\boldsymbol{x}$ to obtain linear combinations of the rows of each A_i.
2. Obtain the rows of each A_i up to a scale factor from the cross products of these linear combinations.
3. Solve linearly for the scales from the optical flow.

For step 1, note that if the image measurement $(\boldsymbol{x},\boldsymbol{y})$ comes from the ith motion model, i.e. if $\boldsymbol{y}^TA_i\boldsymbol{x}=0$, then

$$\frac{\partial \mathrm{MBCC}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}} \sim \boldsymbol{y}^TA_i. \tag{15}$$

That is, the derivatives of the MBCC with respect to $\boldsymbol{x}$ give linear combinations of the rows of the affine model at $\boldsymbol{x}$. Now, since the optical flow $\boldsymbol{u} = [\mathrm{u},\mathrm{v},1]^T$ at pixel $\boldsymbol{x}$ is known, we can compute the vectors $\boldsymbol{y}_1 = [1,0,-\mathrm{u}]^T$ and $\boldsymbol{y}_2 = [0,1,-\mathrm{v}]^T$. Although these vectors are not actual image measurements, they do satisfy $\boldsymbol{y}_1^T\boldsymbol{u} = \boldsymbol{y}_2^T\boldsymbol{u} = 0$. Hence, we can use them to obtain the following linear combination of the rows of the affine model A_i at $(\boldsymbol{x},\boldsymbol{y})$

$$\boldsymbol{g}_{i1} \sim \boldsymbol{a}_{i1} - \mathrm{u}e_3 \quad \text{and} \quad \boldsymbol{g}_{i2} \sim \boldsymbol{a}_{i2} - \mathrm{v}e_3, \tag{16}$$

where $e_3 = [0,0,1]^T$, from the derivatives of the MBCC at $(\boldsymbol{x},\boldsymbol{y}_1)$ and $(\boldsymbol{x},\boldsymbol{y}_2)$, respectively.

For step 2, notice that $\boldsymbol{b}_{i1} = \boldsymbol{g}_{i1}\times e_3 \sim \boldsymbol{a}_{i1}\times e_3$ and $\boldsymbol{b}_{i2} = \boldsymbol{g}_{i2}\times e_3 \sim \boldsymbol{a}_{i1}\times e_3$ are vectors orthogonal to $\boldsymbol{a}_{i1}$ and $\boldsymbol{a}_{i2}$, respectively. As a consequence, even though the pairs $(\boldsymbol{b}_{i1},e_1)$ and $(\boldsymbol{b}_{i2},e_2)$, where $e_1 = [1,0,0]^T$ and $e_2 = [0,1,0]^T$, are not actual image measurements, they do satisfy $e_1^TA_i\boldsymbol{b}_{i1} = \boldsymbol{a}_{i1}^T\boldsymbol{b}_{i1} = 0$ and $e_2^TA_i\boldsymbol{b}_{i2} = \boldsymbol{a}_{i2}^T\boldsymbol{b}_{i2} = 0$. Therefore we can immediately compute the rows of A_i up to scale factors λ_{i1} and λ_{i2} as

$$\tilde{\boldsymbol{a}}_{i1}^T = \lambda_{i1}^{-1}\boldsymbol{a}_{i1}^T = \left.\frac{\partial \mathrm{MBCC}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x},\boldsymbol{y})=(\boldsymbol{b}_{i1},e_1)}, \tag{17}$$

$$\tilde{\boldsymbol{a}}_{i2}^T = \lambda_{i2}^{-1}\boldsymbol{a}_{i2}^T = \left.\frac{\partial \mathrm{MBCC}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x},\boldsymbol{y})=(\boldsymbol{b}_{i2},e_2)}. \tag{18}$$

For step 3, from the optical flow equations $\boldsymbol{u} = A_i\boldsymbol{x}$ we have that $\mathrm{u} = \lambda_{i1}\tilde{\boldsymbol{a}}_{i1}^T\boldsymbol{x}$ and $\mathrm{v} = \lambda_{i2}\tilde{\boldsymbol{a}}_{i2}^T\boldsymbol{x}$, hence the unknown scales are automatically given by

$$\lambda_{i1} = \frac{\mathrm{u}}{\tilde{\boldsymbol{a}}_{i1}^T\boldsymbol{x}} \text{ and } \lambda_{i2} = \frac{\mathrm{v}}{\tilde{\boldsymbol{a}}_{i2}^T\boldsymbol{x}}. \tag{19}$$

By applying steps 1-3 to all N pixels in the image, we can effectively compute one affine matrix A for each pixel, without yet knowing the segmentation of the image measurements. Since in our model we only have $n \ll N$ different affine matrices, we only need to apply steps 1-3 to n pixels corresponding to each one of the n models. We can automatically choose the n pixels at which to perform the computation using the same methodology proposed for 2-D translational motions. Once the $\{A_i\}_{i=1}^n$ are calculated we can cluster the data by assigning $(\boldsymbol{x}^j,\boldsymbol{y}^j)$ to group $i = \arg\min_{\ell=1,\ldots,n}\frac{|\boldsymbol{y}^{jT}A_\ell\boldsymbol{x}^j|^2}{\|\Lambda A_\ell\boldsymbol{x}\|^2}$. We can then refine the affine motion model parameters by solving the linear equation $\boldsymbol{y}^TA_i\boldsymbol{x}=0$ for each separate cluster.

Remark 1 (Spatial smoothness) *Note that our algorithm computes one model per pixel without enforcing that nearby pixels belong to the same group. We can incorporate spatial regularization by applying any smoothing filter, e.g., an average or median filter, to $\{\boldsymbol{u}_i\}_{i=1}^n$ or $\{A_i\}_{i=1}^n$.*

[1] Recall that for any surface $f(\boldsymbol{y}) = 0$, a first order approximation to the geometric distance to the surface is given by $|f(\boldsymbol{y})|/\|\nabla f(\boldsymbol{y})\|$.

3. Experimental results

Synthetic data. We first test our 2-D affine algorithm on synthetic data. We randomly pick $n = 2$ collections of $N = 300$ pixel coordinates and apply a different (randomly chosen) affine motion model to each collection of pixels to generate their optical flow. From the optical flow associated with each pixel, we generate a random vector $\boldsymbol{y}$ of spatial and temporal image derivatives satisfying the BCC (2). The coordinates of $\boldsymbol{y}$ are constrained to be in $[-1, 1]$ to simulate image intensities in the $[0, 1]$ range. Zero-mean Gaussian noise with standard deviation $\sigma \in [0, 0.02]$ is added to the partial derivatives $\boldsymbol{y}$. We run 5,000 trials for each noise level. For each trial the error between the true affine motions $\{A_i\}_{i=1}^n$ and the estimates $\{\hat{A}_i\}_{i=1}^n$ is computed as

$$\text{Affine error} = \frac{1}{n}\sum_{i=1}^{n}\frac{\|A_i - \hat{A}_i\|}{\|A_i\|} \quad (\%). \tag{20}$$

We compare our method against the following algorithms:

1. *K-means*: starting from an initial set of affine matrices, it alternates between assigning data to clusters and computing $\{A_i\}_{i=1}^n$ linearly for each motion class.
2. *Factorization [17]*: it solves for $\{A_i\}_{i=1}^n$ by applying homogeneous polynomial factorization to the MBCC.
3. *Our algorithm + K-means*: it uses our algorithm's output to initialize the K-means algorithm.

Figure 1 plots the mean affine error as a function of σ for all algorithms. Notice that K-means has a nonzero error with perfect data, showing that it usually converges to a local minima when a single random initialization is used. The average number of iterations needed for convergence is 12. The factorization algorithm performs better than K-means for a small level of noise. However, its performance deteriorates quickly as σ increases. In fact, when the multibody affine matrix $\mathcal{A}$ is not factorizable due to noise, the factorization algorithm gives *complex* results. Our algorithm's estimates are always *real* and within 5% of the true affine motions, thus outperforming the K-means and factorization algorithms. The best results are obtained by using our algorithm to initialize K-means, which reduces the error to about 1% and the average number of iterations to 3. Figure 1 also shows the percentage of misclassification. Our algorithm's misclassification rate is 6.5%, even for a noise level of 2% in the image derivatives. The percentage reduces to about 2% by following our algorithm with K-means.

Figure 2 shows the mean error in estimation of optical flow as a function of σ. Note that the error in either component is less than 0.35 pixels for σ = 0.02.

Real sequences. Figure 3 shows segmentation results for a 240×320 sequence of a person's head rotating from right to left in front of a lab background using two 2-D translational motions. The top row shows the pixels associated with the camera's fronto-parallel motion and the bottom row shows the pixels associated with the head motion. In each row, pixels that do not correspond to the group are colored red.

Figure 4 shows another example on the segmentation of a 240×320 sequence of a car leaving a parking lot using two 2-D translational motions. The top row shows the pixels associated with the camera's downward motion and the bottom row shows the pixels associated with the car's right-downward motion. In each row, pixels that do not correspond to the group are colored black. Figure 5 shows segmentation results for the same sequence using two 2-D affine motion models.

Figure 6 shows the segmentation of a sequence taken by a static camera observing a moving car and a moving box using two 2-D affine motions. The top row shows the segmentation of the box motion while the bottom row shows the segmentation of the car motion. In each row the pixels that do not correspond to the group are colored black.

The segmentation results in Figures 3-6 are encouraging. Although we are using a simple mixture of two 2-D translational or two 2-D affine motion models for the entire scene, the two motion models are segmented accurately to a great extent. Most of the errors occur at regions with low texture, e.g., the black sweater and white wall regions in Figure 3, parts of the body of each car and the road in Figures 4 and 5, as well as pixels in highly specular regions where the BCC is not satisfied. In Figure 6 the discrepancies arise from the fact that the 2-D affine motion model gives a rough approximation to the motion of the two objects, because this scene contains noticeable perspective effects. Note also that for the parking lot sequence using 2-D affine motion models gives better segmentation results than using 2-D translational models, as can be seen by comparing Figures 4 and 5. Overall, about 85% of the image pixels are correctly classified with respect to ground truth manual segmentation. These results can be used as an initial segmentation for any more computationally intense nonlinear iterative refinement scheme.

4. Summary and Conclusions

We have presented a closed form solution to direct motion segmentation from the image derivatives. Our approach fits a multibody brightness constancy constraint (MBCC) to all image measurements, and computes the optical flow and the parameters of each motion model from the derivatives of the MBCC. Our algorithm deals properly with moderate amounts of noise, and can be used to initialize any iterative refinement scheme, e.g., EM, to deal with larger amounts of noise. Open research avenues include dealing with outliers in the image measurements as well as dealing with motion models of different type.

Acknowledgments

Work funded by Johns Hopkins Whiting School of Engineering startup funds and NSF CAREER Award ISS-0447739.

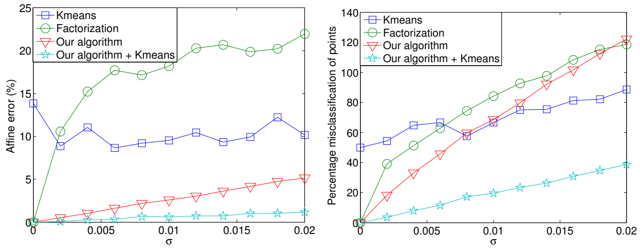

Figure 1. **Left:** Error in affine parameters as a function of noise. **Right:** Percentage of misclassification as a function of noise.

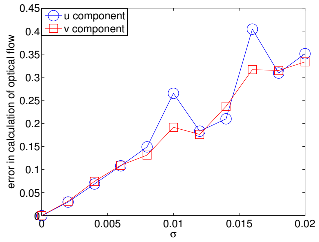

Figure 2. Error in the estimated optical flow as a function of noise.

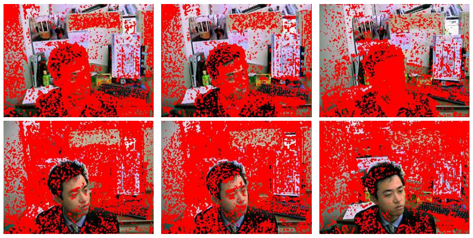

Figure 3. 2-D translational segmentation of head and lab sequence.

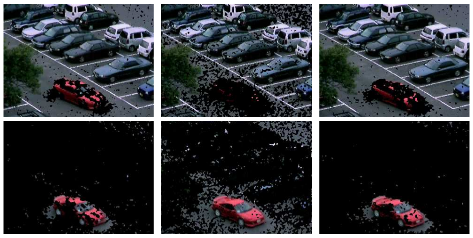

Figure 4. 2-D translational segmentation of parking lot sequence.

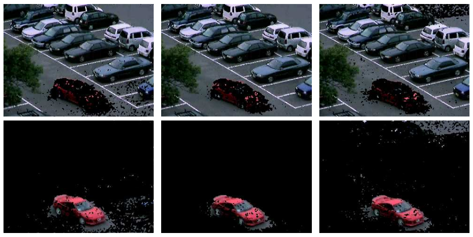

Figure 5. 2-D affine segmentation of parking lot sequence.

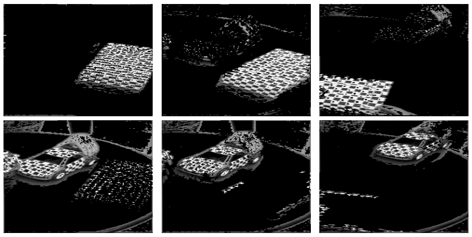

Figure 6. 2-D affine segmentation of car and box sequence.

References

[1] 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*, pages 777–785, 1995.

[2] M. Black and P. Anandan. Robust dynamic motion estimation over time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–302, 1991.

[3] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[4] T. Darrel and A. Pentland. Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, pages 173–178, 1991.

[5] R. Hartley and R. Vidal. The multibody trifocal tensor: Motion segmentation from 3 perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 769–775, 2004.

[6] M. Irani and P. Anandan. About direct methods. In *Workshop on Vision Algorithms*, pages 267–277, 1999.

[7] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *European Conference on Computer Vision*, pages 282–287, 1992.

[8] A. Jepson and M. Black. Mixture models for optical flow computation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, 1993.

[9] Q. Ke and T. Kanade. A robust subspace approach to layer extraction. In *IEEE Workshop on Motion and Video Computing*, pages 37–43, 2002.

[10] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 289–295, 1991.

[11] A. Spoerri and S. Ullman. The early detection of motion boundaries. In *IEEE International Conference on Computer Vision*, pages 209–218, 1987.

[12] 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.

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

[14] R. Vidal and R. Hartley. Motion segmentation with missing data by PowerFactorization and Generalized PCA. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 310–316, 2004.

[15] R. Vidal and Y. Ma. A unified algebraic approach to 2-D and 3-D motion segmentation. In *European Conference on Computer Vision*, pages 1–15, 2004.

[16] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *International Journal of Computer Vision*, 2005.

[17] R. Vidal and S. Sastry. Segmentation of dynamic scenes from image intensities. In *IEEE Workshop on Motion and Video Computing*, pages 44–49, 2002.

[18] J. Wang and E. Adelson. Layered representation for motion analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–366, 1993.

[19] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 520–526, 1997.

[20] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 263–270, 2001.