Direct Segmentation of Multiple 2-D Motion Models of Different Types

Dheeraj Singaraju and René Vidal

Center for Imaging Science, Johns Hopkins University, Baltimore MD 21218
{dheeraj,rvidal}@cis.jhu.edu

Abstract. We propose a closed form solution for segmenting mixtures of 2-D translational and 2-D affine motion models directly from the image intensities. Our approach exploits the fact that the spatial-temporal image derivatives generated by a mixture of these motion models must satisfy a bi-homogeneous polynomial called the multibody brightness constancy constraint (MBCC). We show that the degrees of the MBCC are related to the number of motions models of each kind. Such degrees can be automatically computed using a one-dimensional search. We then demonstrate that a sub-matrix of the Hessian of the MBCC encodes information about the type of motion models. For instance, the matrix is rank-1 for 2-D translational models and rank-3 for 2-D affine models. Once the type of motion model has been identified, one can obtain the parameters of each type of motion model at every image measurement from the cross products of the derivatives of the MBCC. We then demonstrate that accounting for a 2-D translational motion model as a 2-D affine one would result in erroneous estimation of the motion models, thus motivating our aim to account for different types of motion models. We apply our method to segmenting various dynamic scenes.

1 Introduction

Recently, finding effective solutions to the motion segmentation problem has become an important issue in numerous emerging applications. This has motivated the development of various algorithms for motion segmentation. [1] fits a mixture of 2-D parametric models through successive computation of dominant motions. [2] clusters locally estimated 2-D motion models using K-means. 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.

Global methods deal with such problems by fitting a mixture of motion models to the entire scene. [9] fits a mixture of parametric models by minimizing a Mumford-Shah-like cost functional. [3–8] fit a mixture of probabilistic models iteratively using the Expectation Maximization algorithm (EM). The drawback of such iterative approaches is that they are very sensitive to correct initialization and are computationally expensive.

To overcome these difficulties, more recent work [10–13] proposes to solve the problem globally by fitting a polynomial to all the image measurements and

then factorizing this polynomial to obtain the parameters of each 2-D motion model. These approaches have been shown to be effective at finding a good initial estimate for iterative approaches as shown in [12] where the method has been extended to most 2-D and 3-D motion models. [14] integrates the algorithms of [13] and [2] to solve the motion segmentation problem. It applies [13] to a window around every pixel in the scene and thus can account for multiple motions in every such window. K-means is performed on a subset of these locally estimated motion model parameters to get the motion model parameters describing the entire scene.

Unfortunately, all the aforementioned approaches to motion segmentation assume that the scene can be modeled as a mixture of motion models of *the same type*. In practice, this is a significant limitation, because most dynamic scenes exhibit different types of motions. For instance, in the sequence shown in Figure 1 the background is a translating image of a robot on a floor, where as the foreground has a rotating patch that undergoes an affine motion. One could argue that the 2-D translational model is a particular case of a 2-D affine model, hence the problem could be solved by fitting a mixture of 2-D affine motion models. In practice, however this results in poor performance, because in most cases the data associated with simpler models is not rich enough to accurately define the parameters of a more complex model. In fact, as we shall demonstrate later, it is not valid to use algebraic methods such as [13] to estimate a 2-D translational model as a special case of a 2-D affine motion model.

Fig. 1. Sequence consisting of a 2-D translational and a 2-D affine motion model.

We are therefore faced with the problem of fitting multiple models of *different type* to the image data without knowing which pixels correspond to which model. There are many reasons why this problem is significantly more challenging than fitting motion models of the same type.

1. The number of parameters defining each motion model is not the same, hence one cannot directly apply methods based on clustering in the space of parameters, such as K-means, as the parameters to be clustered live in spaces of different dimensions.
2. The number of data points needed to fit a model is not the same, hence it may be difficult to fit one model at a time, e.g., with RANSAC [15], without knowing how many points to use. One could use the maximum number of points needed to define the more complex model, but this may lead to poor performance, as argued before.

1.1 Paper Contributions

We propose a closed form solution to the problem of fitting an unknown number of 2-D motion models of different type to the image derivatives, without knowing which pixels move according to the same motion model. To the best of our knowledge, there is no prior work other than [16] addressing this problem in a purely algebraic setting. However, [16] is a feature-based method, while ours is a direct method. As such, finding a general methodology that solves the motion segmentation problem for all kinds of motion models is at this point elusive. Therefore, in this paper we restrict our attention to 2-D translational and 2-D affine motion models, and propose an algebraic method that solves simultaneously for the type of motion model at every image measurement, the parameters of each motion model, and the segmentation of the image data.

Our algorithm proceeds as follows. We fit a bi-homogeneous polynomial called the multibody brightness constancy constraint (MBCC) to the image measurements. We show that the degrees of the MBCC are related to the number of translational and affine motions and that such degrees can be automatically computed using a one-dimensional search. We then inspect the rank of a sub-matrix of the Hessian of the MBCC at every pixel, and show it encodes information about the type of motion model associated with the pixel. More specifically, this matrix is rank-1 for 2-D translational models and rank-3 for 2-D affine models. We demonstrate that for any given image measurement, we can obtain the parameters of each type of motion model at that measurement, from the cross products of the derivatives of the MBCC, using an extension of the method reported in [13]. We also explain why the method of [13] cannot be used to estimate a 2-D translational model as a degenerate case of a 2-D affine motion model, thus emphasizing the need to account for multiple types of motion models.

2 Segmenting Motions of Different Types

Consider a motion sequence taken by a moving camera observing an *unknown* number of independently and rigidly moving objects. Assume that each one of the surfaces in the scene is Lambertian, so that the optical flow $\boldsymbol{u}(\boldsymbol{x}) = [\mathrm{u}, \mathrm{v}, 1]^\top \in \mathbb{P}^2$ of pixel $\boldsymbol{x} = [x, y, 1]^\top \in \mathbb{P}^2$ is related to the spatial-temporal image derivatives at pixel $\boldsymbol{x}$, $\boldsymbol{y}(\boldsymbol{x}) = [I_x, I_y, I_t]^\top \in \mathbb{R}^3$, by the well-known *brightness constancy constraint* (BCC)

$$\boldsymbol{y}^\top \boldsymbol{u} = I_x \mathrm{u} + I_y \mathrm{v} + I_t = 0. \tag{1}$$

We assume that the optical flow in the scene is generated by n_t 2-D translational motion models $\{\boldsymbol{u}_i \in \mathbb{P}^2\}_{i=1}^{n_t}$

$$\boldsymbol{u} = \boldsymbol{u}_i \quad i = 1, \ldots n_t \tag{2}$$

and by n_a 2-D affine motion models $\{A_i \in \mathbb{R}^{3\times 3}\}_{i=1}^{n_a}$

$$\boldsymbol{u} = A_i \boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_{i1}^\top \\ \boldsymbol{a}_{i2}^\top \\ 0, 0, 1 \end{bmatrix} \boldsymbol{x} \quad i = 1, \ldots, n_a. \tag{3}$$

After combining the 2-D translational and 2-D affine motion models with the BCC (1) we obtain

$$\boldsymbol{y}^\top \boldsymbol{u}_i = 0 \quad \text{and} \quad \boldsymbol{y}^\top A_i \boldsymbol{x} = 0 \tag{4}$$

respectively. Notice that the total number of motion models $n = n_t + n_a$ may be larger than the number of independent rigid-body motions because of perspective effects, depth discontinuities, occlusions, transparent motions, etc.

In the presence of $n = 1$ motion, the above motion constraints are either linear or bilinear on the image measurements $(\boldsymbol{x}, \boldsymbol{y})$ and linear on the motion parameters $\boldsymbol{u}_1$ or A_1. Therefore, if the type of motion model is known, one can estimate the motion model linearly from a collection of N image measurements $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^N$ using one of the equations in (4). In the presence of $n_t + n_a$ motion models, we cannot solve the problem linearly because we do not know

1. The type of motion model associated with each image measurement $(\boldsymbol{x}, \boldsymbol{y})$.
2. The parameters of the motion model associated with each image measurement $(\boldsymbol{x}, \boldsymbol{y})$, or equivalently the segmentation of the data.
3. The number of translational and affine motion models.

Therefore, we are faced with the following problem:

Problem 1 (Segmenting motion models of different types). Given the spatial-temporal derivatives $\{(I_{xj}, I_{yj}, I_{tj}))\}_{j=1}^N$ of a motion sequence generated from n_t translational and n_a affine motion models, estimate the number of motion models (n_a, n_t), the optical flow $\boldsymbol{u}(\boldsymbol{x}_j)$ and the type of motion model at each pixel $\{\boldsymbol{x}_j\}_{j=1}^N$, and the motion parameters of the $n_t + n_a$ models, without knowing which image measurements correspond to which motion model.

2.1 Multibody Brightness Constancy Constraint for Motions of Different Types

Let $(\boldsymbol{x}, \boldsymbol{y})$ be an image measurement associated with any of the motion models. According to the BCC (1) there exists a motion model M_k whose optical flow $\boldsymbol{u}_k(\boldsymbol{x})$ satisfies $\boldsymbol{y}^\top \boldsymbol{u}_k(\boldsymbol{x}) = 0$. Therefore, the following *multibody brightness constancy constraint* (MBCC) must be satisfied by every pixel in the image

$$\text{MBCC}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{n_t} (\boldsymbol{y}^\top \boldsymbol{u}_i) \prod_{j=1}^{n_a} (\boldsymbol{y}^\top A_j \boldsymbol{x}) = 0. \tag{5}$$

From equation (5) we can see that if $n_a = 0$, the MBCC is a homogeneous polynomial of degree n_t in $\boldsymbol{y} = [y_1, y_2, y_3]^\top$ which can be written as a linear combination of the monomials $y_1^{n_1} y_2^{n_2} y_3^{n_3}$ with coefficients $\mathcal{U}_{n_1,n_2,n_3}$. By stacking all the monomials in a vector $\nu_{n_t}(\boldsymbol{y}) \in \mathbb{R}^{M_{n_t}}$ and the coefficients in a *multibody optical flow* vector $\mathcal{U} \in \mathbb{R}^{M_{n_t}}$, where $M_{n_t} = \frac{(n_t+1)(n_t+2)}{2}$, we can express the MBCC as [13]

$$MBCC(\boldsymbol{x}, \boldsymbol{y}) = \nu_{n_t}(\boldsymbol{y})^\top \mathcal{U} = \prod_{i=1}^{n_t} (\boldsymbol{y}^\top \boldsymbol{u}_i). \tag{6}$$

The vector $\nu_{n_t}(\boldsymbol{y}) \in \mathbb{R}^{M_{n_t}}$ is also known as the Veronese map of $\boldsymbol{y}$ of degree n_t.

Similarly, if $n_t = 0$, the MBCC is a bi-homogeneous polynomial of degree n_a in $(\boldsymbol{x}, \boldsymbol{y})$. The coefficients of this polynomial can be stacked into a *multibody affine matrix* $\mathcal{A} \in \mathbb{R}^{M_{n_a} \times M_{n_a}}$, so that the MBCC can be written as [13]

$$MBCC(\boldsymbol{x}, \boldsymbol{y}) = \nu_{n_a}(\boldsymbol{y})^\top \mathcal{A} \nu_{n_a}(\boldsymbol{x}) = \prod_{j=1}^{n_a} (\boldsymbol{y}^\top A_j \boldsymbol{x}). \tag{7}$$

In the case of n_t translational and n_a affine motion models, if we let the (m_1, m_2, m_3)th row of $\mathcal{A}$ be $\boldsymbol{a}^T_{m_1,m_2,m_3}$, we can write the MBCC as

$$\begin{aligned}
&\big(\nu_{n_t}(\boldsymbol{y})^\top \mathcal{U}\big)\big(\nu_{n_a}(\boldsymbol{y})^\top \mathcal{A} \nu_{n_a}(\boldsymbol{x})\big) \\
&\qquad = \Big(\sum y_1^{n_1} y_2^{n_2} y_3^{n_3} \mathcal{U}_{n_1,n_2,n_3}\Big) \Big(\sum y_1^{m_1} y_2^{m_2} y_3^{m_3} \boldsymbol{a}^T_{m_1,m_2,m_3} \nu_{n_a}(\boldsymbol{x})\Big) \\
&\qquad = \sum y_1^{n_1+m_1} y_2^{n_2+m_2} y_3^{n_3+m_3} \mathcal{U}_{n_1,n_2,n_3} \boldsymbol{a}^T_{m_1,m_2,m_3} \nu_{n_a}(\boldsymbol{x}) \\
&\qquad = \nu_{n_a+n_t}(\boldsymbol{y})^\top \mathcal{M} \nu_{n_a}(\boldsymbol{x}) = 0.
\end{aligned}$$

We call $\mathcal{M} \in \mathbb{R}^{M_{n_a+n_t} \times M_{n_a}}$ the *multibody motion matrix*, because it contains information about all the motion models $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ and $\{A_i\}_{i=1}^{n_a}$. Note that when $n_a = 0$, $\mathcal{M}$ is equivalent to the multibody optical flow $\mathcal{U}$ and when $n_t = 0$, $\mathcal{M}$ is equivalent to the multibody affine matrix $\mathcal{A}$.

2.2 Computing the Multibody Motion Matrix

In order to compute the multibody motion matrix $\mathcal{M}$, note that the MBCC holds at every image measurement $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^N$. Therefore, we can compute $\mathcal{M}$ by solving the linear system,

$$L_{(n_a,n_t)} \boldsymbol{m} = 0, \tag{8}$$

where the jth row of $L_{(n_a,n_t)} \in \mathbb{R}^{N \times M_{n_a+n_t} M_{n_a}}$ is given as $(\nu_{n_a+n_t}(\boldsymbol{y}_j) \otimes \nu_{n_a}(\boldsymbol{x}_j))^\top$ and $\boldsymbol{m}$ is the stack of the columns of $\mathcal{M}$.

If $n_a > 0$, notice that some entries of $\mathcal{M}$ are zero, because the entries $(3,1)$ and $(3,2)$ of each A_i are zero. Therefore, we can obtain a more robust estimate of $\mathcal{M}$ in the presence of noise by solving the linear system

$$\tilde{L}_{(n_a,n_t)} \tilde{\boldsymbol{m}} = 0, \tag{9}$$

where $\tilde{\boldsymbol{m}} \in \mathbb{R}^{M_{n_t+n_a} M_{n_a} - Z_{(n_a,n_t)}}$ is the same as $\boldsymbol{m}$, but with the corresponding $Z_{(n_a,n_t)}$ zero entries removed, and $\tilde{L}_{(n_a,n_t)} \in \mathbb{R}^{N \times (M_{n_a+n_t} M_{n_a} - Z_{(n_a,n_t)})}$ is the same as $L_{(n_a,n_t)}$, but with $Z_{(n_a,n_t)}$ columns removed. We solve for $\tilde{\boldsymbol{m}}$ in a least-squares sense as the singular vector of $L_{(n_a,n_t)}$ associated with its smallest singular value. The scale of $\mathcal{M}$ is obtained from $\mathcal{M}(M_{n_a+n_t}, M_{n_a}) = 1$, as $\boldsymbol{u}_i(3) = A_j(3,3) = 1$.

2.3 Computing the Number of Motion Models

Note that in order to solve for $\mathcal{M}$ from the linear system $\tilde{L}_{(n_a,n_t)}\tilde{\boldsymbol{m}} = 0$, we need to know the number of translational and affine models, n_t and n_a, respectively. This problem is indeed more challenging than estimating the number of motion models when all the models are of the same type. This is because there can be multiple possible combinations of (n_a, n_t) for a given data set, as we shall elucidate later in this section.

In order to determine the number of models, we assume that the image measurements $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ are non-degenerate, i.e. they do not satisfy any homogeneous polynomial in $(\boldsymbol{x}, \boldsymbol{y})$ of degree less than n_a in $\boldsymbol{x}$ or less than $n_t + n_a$ in $\boldsymbol{y}$. This assumption is analogous to the standard assumption in structure from motion that image measurements do not live in a critical surface. Under this assumption, we have the following:

Theorem 1 (Number of translational and affine motion models). *Let $\tilde{L}_{(n'_a,n'_t)} \in \mathbb{R}^{N \times M_{n'_t+n'_a} M_{n'_a} - Z_{(n'_a,n'_t)}}$ be the matrix in (9), but computed with the Veronese map of degree n'_a in $\boldsymbol{x}$ and $n'_a + n'_t \geq 1$ in $\boldsymbol{y}$. If rank$(A_i) \geq 2$ for all $i = 1, \ldots, n_a$, and a large enough set of image measurements in general configuration is given, then the number of affine and translational motions is, respectively, given by*

$$\begin{aligned} n_a &= \arg\min_{n'_a} \{n'_a : \exists n'_t \geq 0 : \tilde{L}_{(n'_a,n'_t)} \text{ drops rank by 1}\} \\ n_t &= \arg\min_{n'_t} \{n'_t : \tilde{L}_{(n_a,n'_t)} \text{ drops rank by 1}\}. \end{aligned} \tag{10}$$

Proof. From the non-degeneracy assumption we have that

1. If $n'_a < n_a$ or $n'_t + n'_a < n_t + n_a$, there is no polynomial of degree n'_a in $\boldsymbol{x}$ or of degree $n'_a + n'_t$ in $\boldsymbol{y}$ fitting the data, hence $\tilde{L}_{(n'_a,n'_t)}$ is of full column rank.
2. If $n'_t + n'_a = n_t + n_a$ and $n'_t \leq n_t$, there is exactly one polynomial fitting the data, namely $\nu_{n'_t+n'_a}(\boldsymbol{y})^\top \mathcal{M} \nu_{n'_a}(\boldsymbol{x})$, thus $\tilde{L}_{(n'_a,n'_t)}$ drops rank by 1. This is true for all $n'_t \leq n_t$, given $n'_t + n'_a = n_t + n_a$, because each translational motion model can also be interpreted as an affine motion model.
3. If $n'_t + n'_a > n_t + n_a$ and $n'_a \geq n_a$, there are two or more polynomials of degree n'_a in $\boldsymbol{x}$ and $n'_a + n'_t$ in $\boldsymbol{y}$ that fit the data, namely any multiple of the MBCC. Therefore, the null space of $\tilde{L}_{(n'_a,n'_t)}$ is at least two-dimensional and $\tilde{L}_{(n'_a,n'_t)}$ drops rank by more than 1.

We conclude that there can be multiple values of (n'_a, n'_t) for which the matrix $\tilde{L}_{(n'_a,n'_t)}$ drops rank exactly by 1, i.e. whenever $n'_t + n'_a = n_t + n_a$ and $n'_t \leq n_t$. Thus, the correct number of motions (n_a, n_t) can be obtained as in (10).

As a consequence of the theorem, we can immediately devise a strategy to search for the correct number of motions. Since we know that the correct number of motions occurs for the minimum value of n'_a such that $n'_t + n'_a = n_t + n_a$ and $\tilde{L}_{(n'_a,n'_t)}$ drops rank by 1, we can initially set $(n'_a, n'_t) = (0, 1)$, and then increase

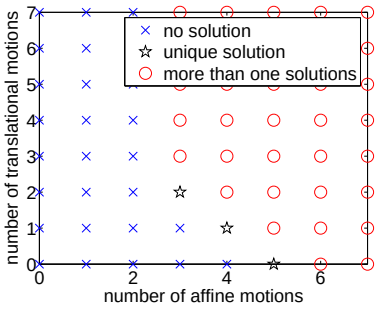


Fig. 2. Plot of the possible pairs of (n'_a, n'_t) that give a unique solution for the MBCC. The correct number of motions is $(n_a, n_t) = (3, 2)$.

n'_a while keeping $n'_a + n'_t$ constant and check if $\tilde{L}_{(n'_a,n_t)}$ drops rank. If $\tilde{L}_{(n'_a,n'_t)}$ does not drop rank, we increase $n'_a + n'_t$ by one, reset $n'_a = 0$ and repeat the process until $\tilde{L}_{(n'_a,n'_t)}$ drops rank by 1 for the first time. This process will stop at the correct (n_a, n_t).

Figure 2 shows the possible solutions for which the data matrix would have a rank deficiency of 1 and illustrates our method for searching for the number of motions (n_a, n_t) in the particular case of $n_a = 3$ affine motions and $n_t = 2$ translational motions. In this case, we search for the correct (n_a, n_t) in the following order $(0, 1)$, $(1, 0)$, $(0, 2)$, $(1, 1)$, $(2, 0)$, $(0, 3), \cdots (0, 5)$, $(1, 4)$, $(2, 3)$,$(3, 2)$.

Notice that the proposed search strategy will give the correct number of motions with perfect data, but will fail with noisy data, because $\tilde{L}_{(n'_a,n'_t)}$ will be full rank for all (n'_a, n'_t). In this case, we can find (n_a, n_t) as the pair that minimizes the cost function

$$\left[\frac{\sigma^2_{M_{n'_t+n'_a}M_{n'_a}-Z_{(n'_a,n'_t)}}(\tilde{L}_{(n'_a,n'_t)})}{\sum_{j=1}^{M_{n'_t+n'_a}M_{n'_a}-Z_{(n'_a,n'_t)}-1}\sigma^2_j(\tilde{L}_{(n'_a,n'_t)})}\right]^{\frac{1}{2}} + \kappa(n'_a + n'_t) + \mu n'_a, \qquad (11)$$

where $\sigma_j(L)$ is the jth singular value of L, and κ and μ are parameters that penalize increasing the complexity of the multibody motion model $\mathcal{M}$. As before, this two-dimensional optimization problem is reduced to a one-dimensional search by evaluating the cost function for values of (n'_a, n'_t) chosen in the order $(0, 1)$, $(1, 0)$, $(0, 2)$, $(1, 1)$, $(2, 0)$, $(0, 3), \cdots$.

2.4 Computing the Motion Type at Each Pixel

Given the number of motion models (n_a, n_t) and the multibody motion model $\mathcal{M}$, we now show how to determine the type of motion model associated with each pixel: 2-D translational or 2-D affine. As it turns out, this can be done in a remarkably simple way by looking at the rank of the matrix

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y} \partial \boldsymbol{x}} \in \mathbb{R}^{3\times 3}. \qquad (12)$$

For the sake of simplicity, consider a scene whose optical flow at every pixel can be modeled by one translational and one affine motion model, $\boldsymbol{u}$ and A, respectively. In this case, the MBCC for the scene can be written as $\text{MBCC}(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{y}^\top \boldsymbol{u})(\boldsymbol{y}^\top A \boldsymbol{x})$, hence

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{u}\boldsymbol{y}^\top A + (\boldsymbol{y}^\top \boldsymbol{u})A. \tag{13}$$

Therefore, if an image measurement comes from the translational motion model only, i.e. if $\boldsymbol{y}_j^\top \boldsymbol{u} = 0$, then

$$\mathcal{H}(\boldsymbol{x}_j, \boldsymbol{y}_j) = \boldsymbol{u}(\boldsymbol{y}_j^\top A) \implies \text{rank}(\mathcal{H}(\boldsymbol{x}_j, \boldsymbol{y}_j)) = 1. \tag{14}$$

Similarly, if the image measurement comes from the affine motion model, i.e. if $\boldsymbol{y}_j^\top A \boldsymbol{x}_j = 0$, then

$$\mathcal{H}(\boldsymbol{x}_j, \boldsymbol{y}_j) = \boldsymbol{u}(\boldsymbol{y}_j^\top A) + (\boldsymbol{y}_j^\top \boldsymbol{u})A \implies \text{rank}(\mathcal{H}(\boldsymbol{x}_j, \boldsymbol{y}_j)) = 3. \tag{15}$$

This simple observation for the case $n_a = n_t = 1$ generalizes to any value of n_a and n_t as stated in the following theorem.

Theorem 2 (Identification of the type of motion model). *Given the multibody motion model $\mathcal{M}$ of the scene, the type of motion model associated with an image measurement $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ can be found as follows*

1. *2-D translational if rank$(\mathcal{H}(\boldsymbol{x}_j, \boldsymbol{y}_j)) = 1$.*
2. *2-D affine if rank$(\mathcal{H}(\boldsymbol{x}_j, \boldsymbol{y}_j)) = 3$.*

Thanks to Theorem 2, we can automatically determine the type of motion model associated with each image measurements. In the case of noisy image data, we declare a model to be 2-D affine if

$$\sum_{i=1}^{9} \frac{|\det(\tilde{H}_i(\boldsymbol{x}_j, \boldsymbol{y}_j))|}{\|\tilde{H}_i(\boldsymbol{x}_j, \boldsymbol{y}_j))\|^2 + \delta} > \epsilon, \tag{16}$$

where $\tilde{H}_i(\boldsymbol{x}_j, \boldsymbol{y}_j), i = 1, \ldots, 9$ are all the distinct 2×2 sub-minors of $H(\boldsymbol{x}_j, \boldsymbol{y}_j)$. δ is added in equation (16) to prevent the term on the left from blowing up when any of the $\tilde{H}_i(\boldsymbol{x}_j, \boldsymbol{y}_j), i = 1, \ldots, 9$ has a value close to 0.

2.5 Computing the Motion Model at Each Pixel

Given the number and types of motion models, and the multibody motion model $\mathcal{M}$, we now show how to compute the individual 2-D translational $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ and 2-D affine $\{A_i\}_{i=1}^{n_a}$ motion models. One possibility is to simply separate the data into two groups, 2-D translational data and 2-D affine data, and then solve separately for the 2-D translational and 2-D affine motion models by using the algorithms in [13] for motion models of the same type. This amounts to solving for the multibody optical flow $\mathcal{U}$ in (6) and the multibody affine matrix $\mathcal{A}$ in

(7), and then applying polynomial differentiation to obtain $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ from $\mathcal{U}$ and $\{A_i\}_{i=1}^{n_a}$ from $\mathcal{A}$. However, at this point we already have the multibody motion $\mathcal{M}$ which is a matrix representation for $\mathcal{U} \otimes \mathcal{A} + \mathcal{A} \otimes \mathcal{U}$. Therefore, having to recompute $\mathcal{U}$ and $\mathcal{A}$ would be extra unnecessary computation.

In this section, we show that the last steps of the method in [13] for motions of the same type can also be applied to motions of different type by showing that one can directly compute $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ and $\{A_i\}_{i=1}^{n_a}$ from the derivatives of the MBCC defined by $\mathcal{M}$. We first notice that one can 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, since each pixel $\boldsymbol{x}$ is associated with one of the $n = n_t + n_a$ motion models, there is a $k = 1, \ldots, n$ such that $\boldsymbol{y}^\top \boldsymbol{u}_k(\boldsymbol{x}) = 0$, where $\boldsymbol{u}_k(\boldsymbol{x})$ is the optical flow evaluated as per the kth motion model. Note that the product $\prod_{\ell \neq i}(\boldsymbol{y}^\top \boldsymbol{u}_\ell(\boldsymbol{x})) = 0$ for all $i \neq k$. Therefore, the optical flow at a pixel can be obtained as

$$\frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y}} = \sum_{i=1}^{n_t+n_a} \boldsymbol{u}_i(\boldsymbol{x}) \prod_{\ell \neq i} (\boldsymbol{y}^\top \boldsymbol{u}_\ell(\boldsymbol{x})) \sim \boldsymbol{u}_k(\boldsymbol{x}). \tag{17}$$

Since the last entry of $\boldsymbol{u}_k(\boldsymbol{x})$ is 1, we can scale the derivative accordingly and this immediately gives us the optical flow at all pixels belonging to only one motion model at a time. If a pixel happens to belong to two motion models, e.g., in regions of low texture for which $\boldsymbol{y} = 0$, then the MBCC has a repeated factor, hence its derivative is zero, and we cannot compute $\boldsymbol{u}(\boldsymbol{x})$ as before.

In the case of 2-D translational motions, the motion model is precisely the optical flow at each pixel. Since we already know which pixels obey a 2-D translation model, we can take the optical flow at those pixels only and obtain the n_t different values $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ using any clustering algorithm in $\mathbb{R}^2$, e.g., K-means. Alternatively, one can choose n_t pixels $\{\boldsymbol{x}_i\}_{i=1}^{n_t}$ with reliable optical flow and then obtain $\boldsymbol{u}_i = \boldsymbol{u}(\boldsymbol{x}_i)$. Since we know that the image derivative $\boldsymbol{y}$ at a pixel $\boldsymbol{x}$ must be orthogonal to the optical flow $\boldsymbol{u}(\boldsymbol{x})$, one can choose a measurement $(\boldsymbol{x}_{n_t}, \boldsymbol{y}_{n_t})$ that minimizes

$$d_{n_t}^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{|\text{MBCC}(\boldsymbol{x}, \boldsymbol{y})|^2}{\|\Lambda \frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y}}\|^2 \|\boldsymbol{y}\|^2}. \tag{18}$$

The remaining measurements for $(\boldsymbol{x}_{i-1}, \boldsymbol{y}_{i-1})$ for $i = n_t, n_t - 1, \ldots, 2$ are chosen by minimizing

$$d_{i-1}^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{d_i^2(\boldsymbol{x}, \boldsymbol{y})}{\frac{|\boldsymbol{y}^\top \boldsymbol{u}(\boldsymbol{x}_i)|^2}{\|\Lambda \boldsymbol{u}(\boldsymbol{x}_i)\|^2}}. \tag{19}$$

Notice that in choosing the points there is no optimization involved. We just need to evaluate the distance functions at each point and choose the one giving the minimum distance. Once the $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ are calculated we can cluster the data by assigning $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ to the model i that minimizes $\frac{(\boldsymbol{y}_j^\top \boldsymbol{u}_i)^2}{\|\boldsymbol{u}_i\|^2}$.

In the case of 2-D affine motion models, one can obtain the affine motion model associated with an image measurement $(\boldsymbol{x}, \boldsymbol{y})$ from the cross products of

the derivatives of the MBCC constraint. More specifically, note that if $(\boldsymbol{x}, \boldsymbol{y})$ comes from the ith motion model, i.e. if $\boldsymbol{y}^\top A_i \boldsymbol{x} = 0$, then

$$\frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \sim \boldsymbol{y}^\top A_i. \tag{20}$$

That is, the partials 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 $\boldsymbol{u} = [\text{u}, \text{v}, 1]^\top$ at pixel $\boldsymbol{x}$ is known, we can evaluate the partials of the MBCC at $(\boldsymbol{x}, \boldsymbol{y}_1)$, with $\boldsymbol{y}_1 = [1, 0, -\text{u}]^\top$, and $(\boldsymbol{x}, \boldsymbol{y}_2)$, with $\boldsymbol{y}_2 = [0, 1, -\text{v}]^\top$, to obtain the following linear combination of the rows of A_i

$$\boldsymbol{g}_{i1} \sim \boldsymbol{a}_{i1} - \text{u}e_3 \quad \text{and} \quad \boldsymbol{g}_{i2} \sim \boldsymbol{a}_{i2} - \text{v}e_3, \tag{21}$$

where e_i is the ith row of the identity matrix. Let $\boldsymbol{b}_{i1} = \boldsymbol{g}_{i1} \times e_3 \sim \boldsymbol{a}_{i1} \times e_3$ and $\boldsymbol{b}_{i2} = \boldsymbol{g}_{i2} \times e_3 \sim \boldsymbol{a}_{i2} \times e_3$. Although the pairs $(\boldsymbol{b}_{i1}, e_1)$ and $(\boldsymbol{b}_{i2}, e_2)$ are not actual image measurements, they do satisfy $e_1^\top A_i \boldsymbol{b}_{i1} = \boldsymbol{a}_{i1}^\top \boldsymbol{b}_{i1} = 0$ and $e_2^\top A_i \boldsymbol{b}_{i2} = \boldsymbol{a}_{i2}^\top \boldsymbol{b}_{i2} = 0$. Therefore we can immediately compute the rows of A_i up to scale factors λ_{i1} and λ_{i2} as

$$\tilde{\boldsymbol{a}}_{i1}^\top = \lambda_{i1}^{-1} \boldsymbol{a}_{i1}^\top = \left. \frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \right|_{(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{b}_{i1}, e_1)}, \tag{22}$$

$$\tilde{\boldsymbol{a}}_{i2}^\top = \lambda_{i2}^{-1} \boldsymbol{a}_{i2}^\top = \left. \frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \right|_{(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{b}_{i2}, e_2)}. \tag{23}$$

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

$$\lambda_{i1} = \frac{\text{u}}{\tilde{\boldsymbol{a}}_{i1}^\top \boldsymbol{x}} \text{ and } \lambda_{i2} = \frac{\text{v}}{\tilde{\boldsymbol{a}}_{i2}^\top \boldsymbol{x}}. \tag{24}$$

By applying this method to all pixels in the image obeying a 2-D affine motion model, we can effectively compute one affine matrix A for each pixel, without yet knowing the segmentation of the image measurements according to the n_a affine models. In order to obtain n_a different affine matrices, we only need to apply the method to n_a pixels corresponding to each one of the n_a models. We can automatically choose the n_a pixels at which to perform the computation using the same methodology proposed for 2-D translational motions, i.e. by choosing points that minimize (18) and a modification of (19). For the 2-D affine models, (19) is modified as

$$d_{i-1}^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{d_i^2(\boldsymbol{x}, \boldsymbol{y})}{\frac{|\boldsymbol{y}^\top A_i \boldsymbol{x}|^2}{\|A(A_i \boldsymbol{x})\|^2}}. \tag{25}$$

Once the $\{A_i\}_{i=1}^{n_a}$ are calculated we cluster the data by assigning $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ to the model i that minimizes $\frac{(\boldsymbol{y}_j^\top A_i \boldsymbol{x}_j)^2}{\|A_i \boldsymbol{x}_j\|^2}$.

Note that if we were to account for a 2-D translational motion as a 2-D affine motion, we would have $\boldsymbol{a}_{i1} \sim \boldsymbol{a}_{i2} \sim e_3$.Then (21) would give us that

$\boldsymbol{g}_{i1} \sim \boldsymbol{g}_{i2} \sim e_3$ and hence imply that $\boldsymbol{b}_{i1} = \boldsymbol{g}_{i1} \times e_3 = 0$ and $\boldsymbol{b}_{i2} = \boldsymbol{g}_{i2} \times e_3 = 0$, and the entire framework for estimating the 2-D affine motion model breaks down. Consequently, it is not possible to estimate a 2-D translational model as a 2-D affine model using the framework of [13]. Note that if $n_a = 0$ or $n_t = 0$, our algorithm is the same as [13].

2.6 Segmentation Scheme

We have demonstrated that given the spatial-temporal image derivatives $\boldsymbol{y}$ at each pixel $\boldsymbol{x}$ in the image, one can obtain a 2-D translational or a 2-D affine motion model describing the optical flow of that pixel using a linear technique. The fundamental reason for this to be possible is that, even though different regions in the image obey different motion models, by taking the product of the equations defining each model in the MBCC, one obtains a multibody motion model that is satisfied by every pixel in the image. This multibody model does not take the region of support of each motion model into account and treats each pixel independently. As such, whether two pixels are far or close to each other has no effect on whether they belong to the same group or not, because the segmentation given by the MBCC is based purely on motion information.

In most real sequences, however, nearby pixels usually move according to the same motion model. Therefore, even though the MBCC leads to an elegant closed form solution to segmentation, the segmentation results of the scheme discussed in section 2.5 will have a lot of holes, as is evident in the results of [13]. One would then have to use some ad-hoc method for smoothing the results.

We would like to design a segmentation scheme that incorporates spatial regularization, because it is expected that, in general, the points that are spatially near by will obey the same motion model. Hence, we adopt the following segmentation scheme. We assign to every pixel $\{\boldsymbol{x}_j\}_{j=1}^N$, one of the $n_t + n_a$ motion models that are evaluated as per the discussion in Section 2.5. We consider a window $\mathcal{W}(\boldsymbol{x}_j)$ around every pixel $\boldsymbol{x}_j$ and choose the model that minimizes the sum of the squares of the BCC evaluated at every pixel in the window. That is, we assign to $\boldsymbol{x}_j$ a motion model M as follows.

$$M(\boldsymbol{x}_j) = \min_{k=1\ldots n_t+n_a} \{M_k : \sum_{\boldsymbol{x}_m \in \mathcal{W}(\boldsymbol{x}_j)} (\boldsymbol{y}_m^\top \boldsymbol{u}_k(\boldsymbol{x}_m))^2\}, \tag{26}$$

where $\boldsymbol{u}_k(\boldsymbol{x}_j)$ is the optical flow evaluated at $\boldsymbol{x}_j$ according to the motion model M_k. This is equivalent to assigning to a window the motion model that gives the least residual with respect to the BCC for that window. By applying this procedure to all pixels in the image, $\{\boldsymbol{x}_j\}_{j=1}^N$, we can segment the entire scene. One can then refine the motion model parameters by re-calculating the motion parameters for each segment.

3 Experimental Results

In this section, we analyze the performance of our proposed algorithm for segmenting image measurements arising from multiple motion models.

3.1 Synthetic Data: Estimation of Number of Motion Models

We first demonstrate the performance of the algorithm in the estimation of the number of 2-D motion models in the presence of noise. For a particular (n_a, n_t) we first randomly generate n_a 2-D affine and n_t 2-D translational motion models and then randomly choose 500 points for each model. The optical flow $\boldsymbol{u}$ at each point is generated according to its corresponding motion model and this is then used to generate a random vector $\boldsymbol{y}$ of spatial and temporal image derivatives satisfying the brightness constancy constraint (1). 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 $\sigma \in [0, 0.01]$ is added to the partial derivatives $\boldsymbol{y}$. We run 1000 trials for each noise level and for every trial we estimate the number of translational and affine motion models as per (11). We used $\kappa = 1.5 \times 10^{-4}$ and $\mu = 2 \times 10^{-5}$ in our experiments.

The results are displayed in the form of histograms in Figure 3. Each histogram helps us analyze the number of trials in which our algorithm predicts a particular number of translational or affine models at different noise levels. It can be seen that in most cases the estimation of the number of models is very good. In fact, we see that when the number of models is not correctly estimated, it usually is the case that a translational model is estimated as an affine model. In such cases, the number of affine models is overestimated and the number of translational models is underestimated. This can be easily verified from the histograms. Note that the estimation of number of models is quite good for $(n_a, n_t) = (2, 1)$ and $(n_a, n_t) = (1, 2)$ but the estimation of number of translational models is not good for $(n_a, n_t) = (1, 1)$. This leads us to believe that the estimation process is sensitive with respect to κ and μ.

3.2 Real Data

We now demonstrate the performance of our algorithm on real world sequences. The pixels co-ordinates are normalized to be 0 mean and lie between -1 and 1. We use the combinations $(n_a, n_t) = (0, 2)$ and $(n_a, n_t) = (2, 0)$ for the methods of [13]. We use $(n_a, n_t) = (1, 1)$ for our method and use windows of size 3×3 to describe local neighborhoods for our segmentation scheme. In each frame, points that do not correspond to a particular group are colored black.

Figure 4 shows the segmentation of the sequence shown in Figure 1, obtained using the methods in [13] and our method. As mentioned earlier, the rotating patch in the foreground obeys a 2-D affine motion model, while the background obeys a 2-D translational motion model. Note that the group of points obeying the 2-D affine motion (shown in white color) is estimated quite accurately using our method. The segmentation obtained assuming two 2-D translational motions is bad. This is expected, because it is not possible to represent a 2-D affine motion as a 2-D translational motion. The segmentation obtained assuming 2-D affine motions is also bad. This is in conjunction with our argument that we cannot use the method of [13] to estimate a 2-D translational motion as a special case of a 2-D affine motion. Our method on the other hand gives good segmentation results

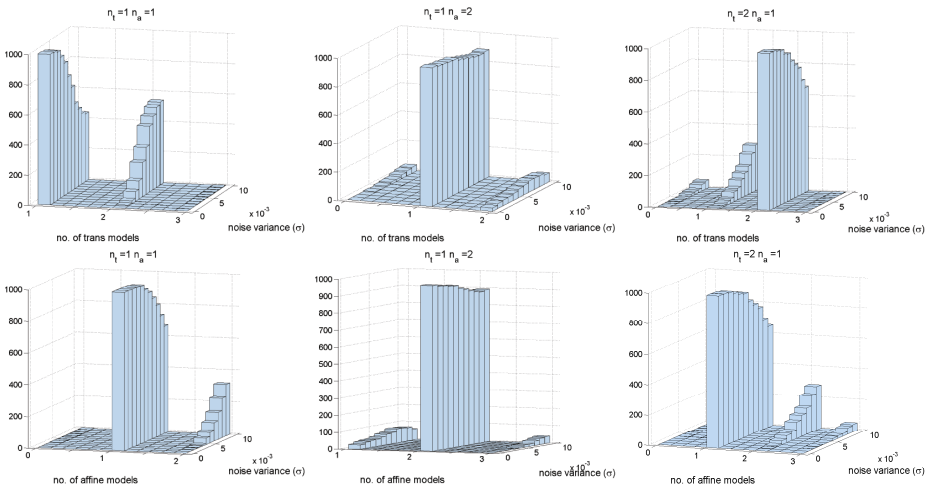


Fig. 3. Results of estimation of number of motion models for 1000 trials and noise levels with $\sigma \in [0, 0.01]$ for the cases $(n_a, n_t) = (1, 1), (2, 1)$ and $(1, 2)$. Every histogram has the number of true motion models listed above it in the format $n_t = n_1, n_a = n_2$.

as we account for the correct types of motion models in the scene. Although there are a few areas that are segmented incorrectly by our method, note that these patches are textureless. Since they can obey any motion model, they are expected to be segmented arbitrarily.

Figure 5 shows the segmentation of the sequence shown in the first row, obtained using the methods of [13] and our method. The sequence in Figure 5 has a rotating image frame of a parking lot in the background that is obeying a 2-D affine motion. The patch in the foreground is undergoing a left-upward translational motion. Note that our method gives much better segmentation results than [13]. In fact, the areas that are incorrectly segmented mostly correspond to textureless patches in the scene.

4 Summary and Conclusions

We have presented a closed form solution for segmenting the motion of a scene consisting of a mixture of 2-D translational and 2-D affine motion models, directly from the image intensities. We have shown that if one were to adopt the algebraic approach of [13], it is imperative that we do not estimate a 2-D translational model as a degenerate case of a 2-D affine model. The highlight of our algorithm is that it provides an algebraic framework that lets us deal with a mixture of motion models of different types.

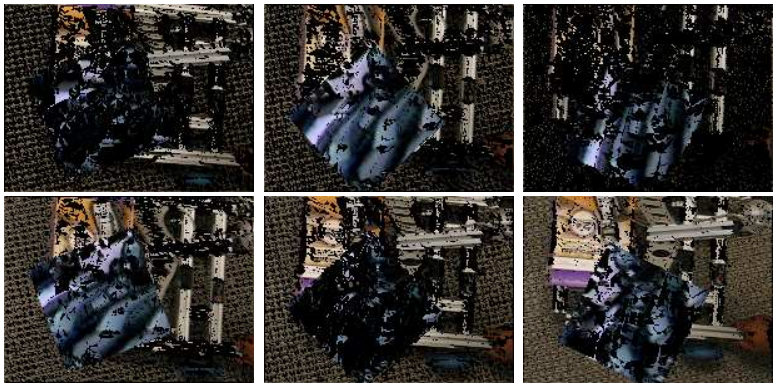
A major bottleneck in the performance of the method is the evaluation of the rank of $\mathcal{H}(\boldsymbol{x}, \boldsymbol{y})$. If the rank is not estimated properly then this will result in an incorrect identification of type of motion. This could obviously result in a bad estimation of the motion model parameters and hence poor segmentation. Future work entails finding a robust way of estimating the rank of $\mathcal{H}(\boldsymbol{x}, \boldsymbol{y})$.

Acknowledgements

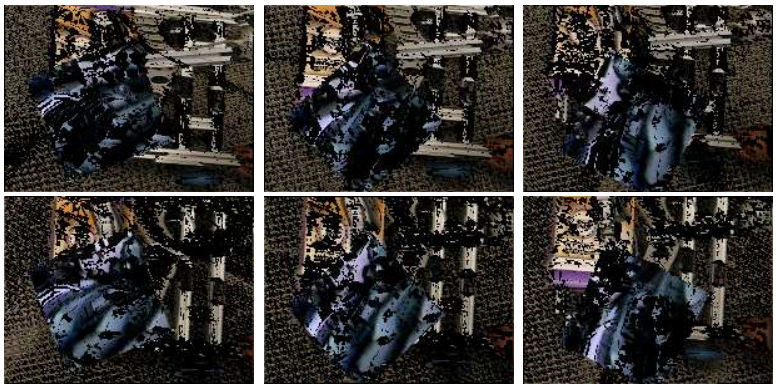
This work was supported by Hopkins WSE startup funds, and by grants NSF CAREER IIS-0447739, NSF CRS-EHS-0509101, and ONR N00014-05-1-0836.

References

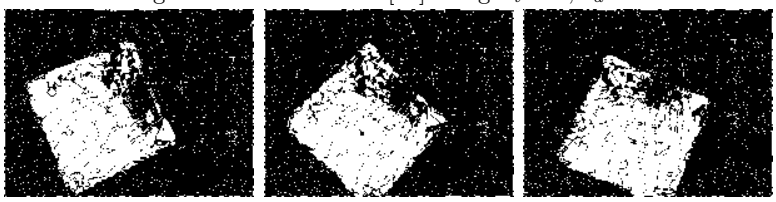
1. Irani, M., Rousso, B., Peleg, S.: Detecting and tracking multiple moving objects using temporal integration. In: European Conference on Computer Vision. (1992) 282–287
2. Wang, J., Adelson, E.: Layered representation for motion analysis. In: IEEE Conference on Computer Vision and Pattern Recognition. (1993) 361–366
3. Darrel, T., Pentland, A.: Robust estimation of a multi-layered motion representation. In: IEEE Workshop on Visual Motion. (1991) 173–178
4. Jepson, A., Black, M.: Mixture models for optical flow computation. In: IEEE Conference on Computer Vision and Pattern Recognition. (1993) 760–761
5. Ayer, S., Sawhney, H.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In: IEEE International Conference on Computer Vision. (1995) 777–785
6. Weiss, Y.: 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) 321–326
7. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: IEEE Conference on Computer Vision and Pattern Recognition. (1997) 520–526
8. Torr, P., Szeliski, R., Anandan, P.: An integrated Bayesian approach to layer extraction from image sequences. IEEE Trans. on Pattern Analysis and Machine Intelligence **23**(3) (2001) 297–303
9. Cremers, D., Soatto, S.: Motion competition: A variational framework for piecewise parametric motion segmentation. International Journal of Computer Vision **62**(3) (2005) 249–265
10. Shizawa, M., Mase, K.: A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In: IEEE Conference on Computer Vision and Pattern Recognition. (1991) 289–295
11. Vidal, R., Sastry, S.: Segmentation of dynamic scenes from image intensities. In: IEEE Workshop on Motion and Video Computing. (2002) 44–49
12. Vidal, R., Ma, Y.: A unified algebraic approach to 2-D and 3-D motion segmentation. In: European Conference on Computer Vision. (2004) 1–15
13. Vidal, R., Singaraju, D.: A closed-form solution to direct motion segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume II. (2005) 510–515
14. Singaraju, D., Vidal, R.: A bottom up algebraic approach to motion segmentation. In: ACCV (1). (2006) 286–296
15. Fischler, M.A., Bolles, R.C.: RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **26** (1981) 381–395
16. Rao, S., Yang, A.Y., Wagner, A., Ma, Y.: Segmentation of hybrid motions via hybrid quadratic surface analysis. In: ICCV. (2005) 2–9



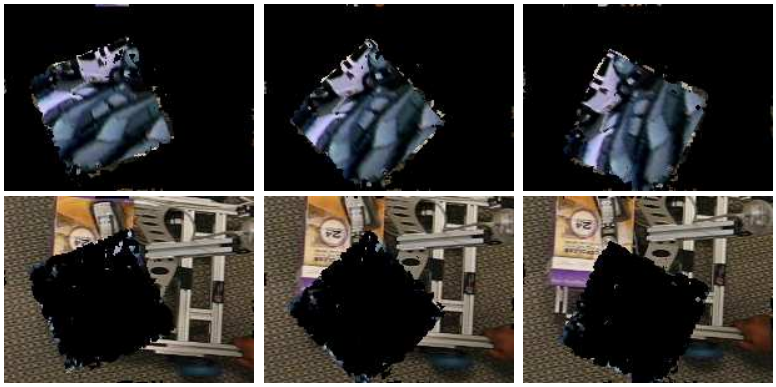
Segmentation results of [13] using $n_t = 2, n_a = 0$



Segmentation results of [13] using $n_t = 0, n_a = 2$



Points satisfying 2-D affine motion as predicted by our method



Segmentation results of our method using $n_t = 1, n_a = 1$

Fig. 4. Comparison of segmentation results of our method with the methods of [13].

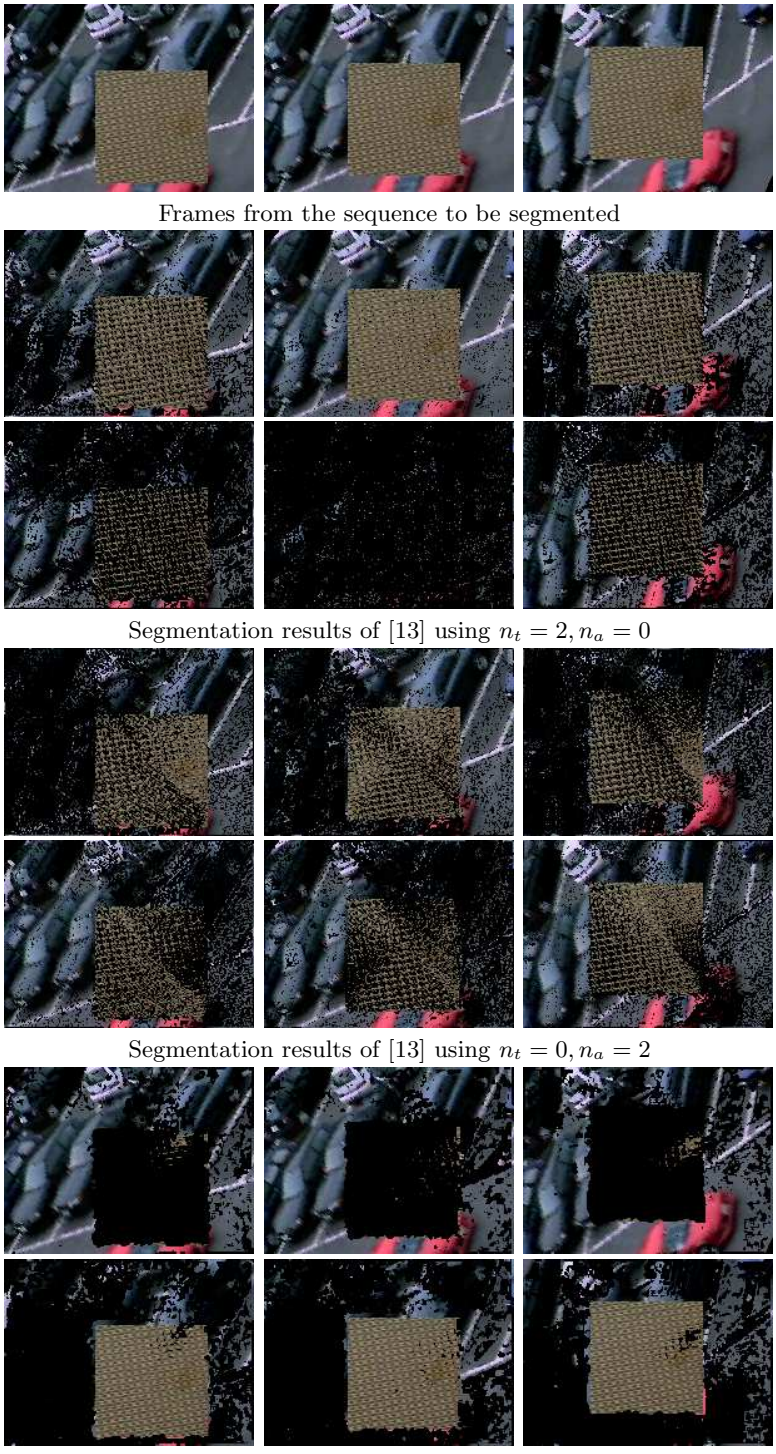


Fig. 5. Comparison of segmentation results of our method with the methods of [13].