A Bottom Up Algebraic Approach to Motion Segmentation

Dheeraj Singaraju and René Vidal

Center for Imaging Science, Johns Hopkins University,
301 Clark Hall, 3400 N. Charles St., Baltimore, MD, 21218, USA
{dheeraj, rvidal}@cis.jhu.edu
http://www.vision.jhu.edu

Abstract. We present a bottom up algebraic approach for segmenting multiple 2D motion models directly from the partial derivatives of an image sequence. Our method fits a polynomial called the multibody brightness constancy constraint (MBCC) to a window around each pixel of the scene and obtains a local motion model from the derivatives of the MBCC. These local models are then clustered to obtain the parameters of the motion models for the entire scene. Motion segmentation is obtained by assigning to each pixel the dominant motion model in a window around it. Our approach requires no initialization, can handle multiple motions in a window (thus dealing with the aperture problem) and automatically incorporates spatial regularization. Therefore, it naturally combines the advantages of both local and global approaches to motion segmentation. Experiments on real data compare our method with previous local and global approaches.

1 Introduction

Motion segmentation is a fundamental problem in many applications in computer vision, such as traffic surveillance, recognition of human gaits, etc. This has motivated the recent development of various local and global approaches to motion segmentation.

Local methods such as Wang and Adelson [1] divide the image in small patches and estimate an affine motion model for each patch. The parameters of the affine models are then clustered using the K-means algorithm and the regions of support of each motion model are computed by comparing the optical flow at each pixel with that generated by the "clustered" affine motion models. The drawback of such local 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. [2] fits a mixture of parametric models by minimizing a Mumford-Shah-like cost functional. [3, 4, 5, 6, 7, 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 [9, 10, 11, 12] proposes to solve the problem by globally fitting a polynomial to all the image measurements and then factorizing this polynomial to obtain the parameters of each 2-D motion model. These algebraic approaches do not require initialization, can deal with multiple motion models across motion boundaries and do not suffer from the aperture problem. However,

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3851, pp. 286–296, 2006.

these algebraic techniques are sensitive to outliers in the data and fail to incorporate spatial regularization, hence one needs to resort to some ad-hoc smoothing scheme for improving the segmentation results.

1.1 Paper Contributions

In this paper, we present a bottom up approach to direct motion segmentation, that integrates the advantages of the algebraic method of [12], and the non-algebraic method of [1], and at the same time reduces the effect of their individual drawbacks.

Our approach proceeds as follows. We first consider a window around each pixel of the scene and fit a polynomial called the multibody brightness constancy constraint (MBCC) [12] to the image measurements of that window. By exploiting the properties of the MBCC, we can find the parameters of the multiple motion models describing the motion of that window. After choosing a dominant local motion model for each window in the scene, we cluster these models using K-means to obtain the parameters describing the motion of the entire scene [1]. Given such global models, we segment the scene by allotting to every pixel the dominant global motion model in a window around it.

This new approach to motion segmentation offers various important advantages.

1. With respect to local methods, our approach can handle more than one motion model per window, hence it is less subject to the aperture problem or to the estimation of a single motion model across motion boundaries.
2. With respect to global iterative methods, our approach has the advantage of not requiring initialization.
3. With respect to global algebraic methods, our approach implicitly incorporates spatial regularization by assigning to a pixel the dominant motion model in a window around it. This also allows our method to deal with a moderate level of outliers.

2 Problem Statement

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 $\{\boldsymbol{u}_i \in \mathbb{P}^2\}_{i=1}^{n_t}$ or by n_a 2-D affine motion models $\{A_i \in \mathbb{R}^{3\times 3}\}_{i=1}^{n_a}$

$$\boldsymbol{u} = \boldsymbol{u}_i \quad i = 1, \ldots n_t \qquad \text{or} \qquad \boldsymbol{u} = A_i \boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_{i1}^\top \\ \boldsymbol{a}_{i2}^\top \\ 0, 0, 1 \end{bmatrix} \boldsymbol{x} \quad i = 1, \ldots, n_a, \tag{2}$$

respectively. Under these models, the BCC (1) reduces to

$$\boldsymbol{y}^\top \boldsymbol{u}_i = 0 \qquad \text{and} \qquad \boldsymbol{y}^\top A_i \boldsymbol{x} = 0 \tag{3}$$

for the 2-D translational and 2-D affine motion models, respectively.

In this paper, we consider the following problem.

Problem 1 (Direct 2-D motion segmentation). Given the spatial-temporal derivatives $\{(I_{xj}, I_{yj}, I_{tj}))\}_{j=1}^{N}$ *of a motion sequence generated by a known number of* $n = n_t$ *translational or* $n = n_a$ *affine motion models, estimate the optical flow* $\boldsymbol{u}(\boldsymbol{x})$*, the motion model at each pixel* $\{\boldsymbol{x}_j\}_{j=1}^{N}$ *and the segmentation of the image measurements, without knowing which measurements correspond to which motion model.*

3 Global Algebraic Motion Segmentation from the Multibody Brightness Constancy Constraint

In this section, we review the global algebraic approach to direct motion segmentation introduced in [12], which is based on a generalization of the BCC to multiple motions.

Let $(\boldsymbol{x}, \boldsymbol{y})$ be an image measurement associated with any of the motion models. According to the BCC (1) there exists a 2-D motion model, say the kth model, 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

$$\text{MBCC}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{n} (\boldsymbol{y}^\top \boldsymbol{u}_i(\boldsymbol{x})) = 0. \tag{4}$$

From equation (4) we can see that in the purely translational case, the MBCC is a homogeneous polynomial of degree n_t which can be written as a linear combination of the monomials $y_1^{l_1} y_2^{l_2} y_3^{l_3}$ with coefficients $\mathcal{U}_{l_1,l_2,l_3}$. 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

$$MBCC(\boldsymbol{x}, \boldsymbol{y}) = \nu_{n_t}(\boldsymbol{y})^\top \mathcal{U} = \prod_{i=1}^{n_t} (\boldsymbol{y}^\top \boldsymbol{u}_i). \tag{5}$$

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

Similarly, if the entire scene can be modeled by affine motion models only, 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

$$MBCC(\boldsymbol{x}, \boldsymbol{y}) = \nu_{n_a}(\boldsymbol{y})^\top \mathcal{A} \nu_{n_a}(\boldsymbol{x}) = \prod_{j=1}^{n_a} (\boldsymbol{y}^\top A_j \boldsymbol{x}). \tag{6}$$

3.1 Computing the Multibody Motion Model

As the MBCC holds at every image measurement $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{N}$, we can compute the multibody motion model $\mathcal{M} = \mathcal{U}$ or $\mathcal{A}$ by solving the linear system

$$L_n \boldsymbol{m} = 0, \tag{7}$$

where $\boldsymbol{m}$ is the stack of the columns of $\mathcal{M}$. In the case of translational models, the jth row of $L_{n_t} \in \mathbb{R}^{N \times M_{n_t}}$ is given by $\nu_{n_t}(\boldsymbol{y}_j)^\top$. In the case of affine models, the jth row of $L_{n_a} \in \mathbb{R}^{N \times (M_{n_a}^2 - Z_{n_a})}$ is given by a subset of the entries of $(\nu_{n_a}(\boldsymbol{y}_j) \otimes \nu_{n_a}(\boldsymbol{x}_j))^\top$. The dimension of L_{n_a} is $N \times (M_{n_a}^2 - Z_{n_a})$ rather than $N \times M_{n_a}^2$, because Z_{n_a} elements of $\mathcal{A}$ are zero, as the $(3,1)$ and $(3,2)$ elements of every affine motion model $\{A_i\}_{i=1}^{n_a}$ are zero. The enforcement of this constraint leads to a more robust calculation of $\mathcal{A}$.

With noisy data the equation $\text{MBCC}(\boldsymbol{x}, \boldsymbol{y}) = 0$, becomes $\text{MBCC}(\boldsymbol{x}, \boldsymbol{y}) \approx 0$. Nevertheless, since the MBCC is linear in the multibody motion parameters $\mathcal{U}$ or $\mathcal{A}$, we can solve a linear inhomogeneous system by enforcing the last entry of $\boldsymbol{m}$ to be 1. It is easy to prove, that when $n_t = 1$ or $n_a = 1$, this method of solving the linear system, reduces to the standard local approaches of fitting a single motion model to a given window.

3.2 Motion Segmentation Using the MBCC

In this subsection, we demonstrate how one can calculate the parameters of the multiple motion models associated with the entire scene from its MBCC.

A very important and powerful property of the MBCC is 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. Since each pixel $\boldsymbol{x}$ is associated with one of the n motion models, there is a $k = 1, \ldots, n$ such that $\boldsymbol{y}^\top \boldsymbol{u}_k(\boldsymbol{x}) = 0$, so $\prod_{\ell \neq i} (\boldsymbol{y}^\top \boldsymbol{u}_\ell(\boldsymbol{x})) = 0$ for all $i \neq k$. Therefore, the optical flow at a pixel obeying model k can be obtained as

$$\frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y}} = \sum_{i=1}^{n} \boldsymbol{u}_i(\boldsymbol{x}) \prod_{\ell \neq i} (\boldsymbol{y}^\top \boldsymbol{u}_\ell(\boldsymbol{x})) \sim \boldsymbol{u}_k(\boldsymbol{x}). \tag{8}$$

For 2-D translational motions, the motion model is the optical flow at each pixel. Hence, we can take the optical flow at all the pixels in the scene and obtain the n_t different values $\{\boldsymbol{u}_i\}_{i=1}^{n_t}$ using any clustering algorithm in $\mathbb{R}^2$. Alternatively, one can choose n_t pixels $\{\boldsymbol{x}_i\}_{i=1}^{n_t}$ with reliable optical flow and then obtain $\boldsymbol{u}_i = \boldsymbol{u}(\boldsymbol{x}_i)$. As shown in [12], under the assumption of zero-mean Gaussian noise in $\boldsymbol{y}$ with covariance $\Lambda \in \mathbb{R}^{3 \times 3}$, one can choose a measurement $(\boldsymbol{x}_{n_t}, \boldsymbol{y}_{n_t})$ that minimizes

$$d_{n_t}^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{|\text{MBCC}(\boldsymbol{x}, \boldsymbol{y})|^2}{\| \Lambda \frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y}} \|^2}. \tag{9}$$

The remaining measurements $(\boldsymbol{x}_{i-1}, \boldsymbol{y}_{i-1})$ for $i = n_t, n_t - 1, \ldots, 2$ are chosen as the ones that minimize

$$d_{i-1}^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{d_i^2(\boldsymbol{x}, \boldsymbol{y})}{\frac{|\boldsymbol{y}^\top \boldsymbol{u}_i|^2}{\|\Lambda \boldsymbol{u}_i\|^2}}. \tag{10}$$

Notice that in choosing the points there is no optimization involved. We just evaluate the distance functions d_i at each point and choose the one giving the minimum distance.

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. 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

$$\frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \sim \boldsymbol{y}^\top A_i. \tag{11}$$

Thus, 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

$$\boldsymbol{g}_{i1} \sim \boldsymbol{a}_{i1} - \text{u}e_3 \quad \text{and} \quad \boldsymbol{g}_{i2} \sim \boldsymbol{a}_{i2} - \text{v}e_3, \tag{12}$$

where e_i is given by the ith column of the 3×3 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 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

$$\tilde{\boldsymbol{a}}_{i1}^\top = \lambda_{i1}^{-1} \boldsymbol{a}_{i1}^\top = \left.\frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{b}_{i1}, e_1)}, \tilde{\boldsymbol{a}}_{i2}^\top = \lambda_{i2}^{-1} \boldsymbol{a}_{i2}^\top = \left.\frac{\partial \text{MBCC}(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{b}_{i2}, e_2)}. \tag{13}$$

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

$$\lambda_{i1} = \frac{\text{u}}{\tilde{\boldsymbol{a}}_{i1}^\top \boldsymbol{x}} \qquad \text{and} \qquad \lambda_{i2} = \frac{\text{v}}{\tilde{\boldsymbol{a}}_{i2}^\top \boldsymbol{x}}. \tag{14}$$

In order to obtain the 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 (9) and a modification of (10). For the 2-D affine models, (10) is modified as

$$d_{i-1}^2(\boldsymbol{x}, \boldsymbol{y}) = \frac{d_i^2(\boldsymbol{x}, \boldsymbol{y})}{\frac{|\boldsymbol{y}^\top A_i \boldsymbol{x}|^2}{\|\Lambda(A_i \boldsymbol{x})\|^2}}. \tag{15}$$

Once the n models have been computed, the scene is segmented using the following scheme: assign $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ to group i if

$$i = \arg \min_{\ell=1,\ldots,n} \frac{|\boldsymbol{y}_j^\top \boldsymbol{u}_\ell|^2}{\|\Lambda \boldsymbol{u}_\ell\|^2} \qquad \text{for the translational case,} \tag{16}$$

$$i = \arg \min_{\ell=1,\ldots,n} \frac{|\boldsymbol{y}_j^\top A_\ell \boldsymbol{x}_j|^2}{\|\Lambda(A_\ell \boldsymbol{x}_j)\|^2} \qquad \text{for the affine case.} \tag{17}$$

4 A Bottom Up Approach to Direct 2-D Motion Segmentation

The local method of [1] considers a window around every pixel, fits a single motion model to each window, and then clusters the locally estimated motion models. As earlier pointed out, this method can suffer from the aperture problem and hence, one would

be required to take a large window to avoid it. However, using a large window can lead to the estimation of a single motion model across motion boundaries. The global method of [12] helps in overcoming these two problems by globally fitting a MBCC to the entire scene. However, this purely algebraic method does not incorporate spatial regularization, because the segmentation is point-based rather than patch-based, as suggested by equations (16) and (17). This results in noisy segmentations, which need to be post-processed using ad-hoc techniques for spatial smoothing. In addition, the method does not deal with outliers in the image measurements.

In this section, we propose a bottom up approach to motion segmentation which integrates the local and algebraic approaches by exploiting their individual advantages. We propose to fit multiple motion models to a possibly large window around each pixel using the algebraic method, to then cluster these locally estimated models. The details of our approach are given in the following subsections.

4.1 Local Computation of Multiple Motion Models

We consider a window $\mathcal{W}(\boldsymbol{x})$ around a pixel $\boldsymbol{x}$ and fit a MBCC to the measurements in that window. In doing so, we use a variable number of models $n = 1, \ldots, n_{\max}$, where $n_{\max}$ is the maximum number of motion models in the scene. For every n, we use the method described in Section 3 to calculate n motion models $M_n^1 \ldots M_n^n$ for that window. As n varies, this gives a total of $\frac{n_{\max}(n_{\max}+1)}{2}$ motion models for every window. From these candidate local models, we choose the *dominant local motion model* for that window as the one that minimizes the sum of the squares of the brightness constancy constraint evaluated at every pixel in the window. That is, we assign to $\boldsymbol{x}_j$ the model

$$M(\boldsymbol{x}_j) = \min_{\substack{n=1\ldots n_{\max} \\ l=1\ldots n}} \{M_n^l : \sum_{\boldsymbol{x}_k \in \mathcal{W}(\boldsymbol{x}_j)} (\boldsymbol{y}_k^\top \boldsymbol{u}_n^l(\boldsymbol{x}_k))^2\}, \tag{18}$$

where $\boldsymbol{u}_n^l(\boldsymbol{x}_k)$ is the optical flow evaluated at $\boldsymbol{x}_k$ according to M_n^l, i.e. the lth motion model estimated assuming n motion models in the window. 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 estimate a collection of N local motion models for the entire scene.

Note that, in comparison with the local approach of [1], our method can account for more than one motion model in a window. In addition, the structure of the MBCC lets us choose the size of the window as large as necessary without having to worry about the motion boundary problem. In fact [12] deals with the case where the window size is the size of the entire image and hence fits the motion models to the entire scene.

An additional feature of our method is that equation (18) can also be used to estimate the number of motion models in a window. However, an accurate estimation of the number of models is not critical, as long as $n_{\max}$ is larger than or equal to the true number of models in the window. This is because if the true number of motion models is over estimated, then the estimated MBCC in (4) has additional factors apart from the true factors. Once can show that these additional factors do not affect the calculations described in equations (8) - (15). We omit the details of the proof due to space limitations.

4.2 Clustering the Model Parameters

Ideally, pixels corresponding to the same motion should have the same motion parameters. However, due to noise and outliers, the locally estimated motion model parameters may not be the same for pixels corresponding to the same motion.

In order to obtain a set of reliable motion model parameters that define the motion of the entire scene, we apply the K-means algorithm in the space of model parameters. Note that if we were to apply [12] to the entire scene followed by the K-means algorithm, we would have had problems due to outliers. However, in our approach, even for windows centered at outliers, we choose the pixels with most reliable motion model parameters in the window, thus providing better estimates of the local motion model at a pixel than [12]. We also provide better estimates than [1] that evaluates just one motion model per pixel, because we can evaluate multiple motion models at motion boundaries. Though we finally consider only one motion model per pixel on motion boundaries also, we claim that this motion model is more accurate than the motion model given by [1], because we choose the best among multiple local models.

4.3 Segmentation of the Motion Models

Once the motion model parameters describing the motion of the entire scene are calculated, it remains to be decided as to how one should segment the scene. While [12] performs well to a great extent, it does not incorporate spatial regularization. As a result the segmentation has a lot of holes and one has to use some ad-hoc method for smoothing the results.

We would like to design a segmentation scheme which incorporates spatial regularization, because it is expected that points that are spatially near by will obey the same motion model. Hence, we consider a window $\mathcal{W}(\boldsymbol{x}_j)$ around every pixel $\{\boldsymbol{x}_j\}_{j=1}^N$ and assign to it the *dominant global motion model* for that window, that is, the global model that minimizes the residual with respect to the BCC for the entire window. In the case of translational models, this can be expressed mathematically as follows

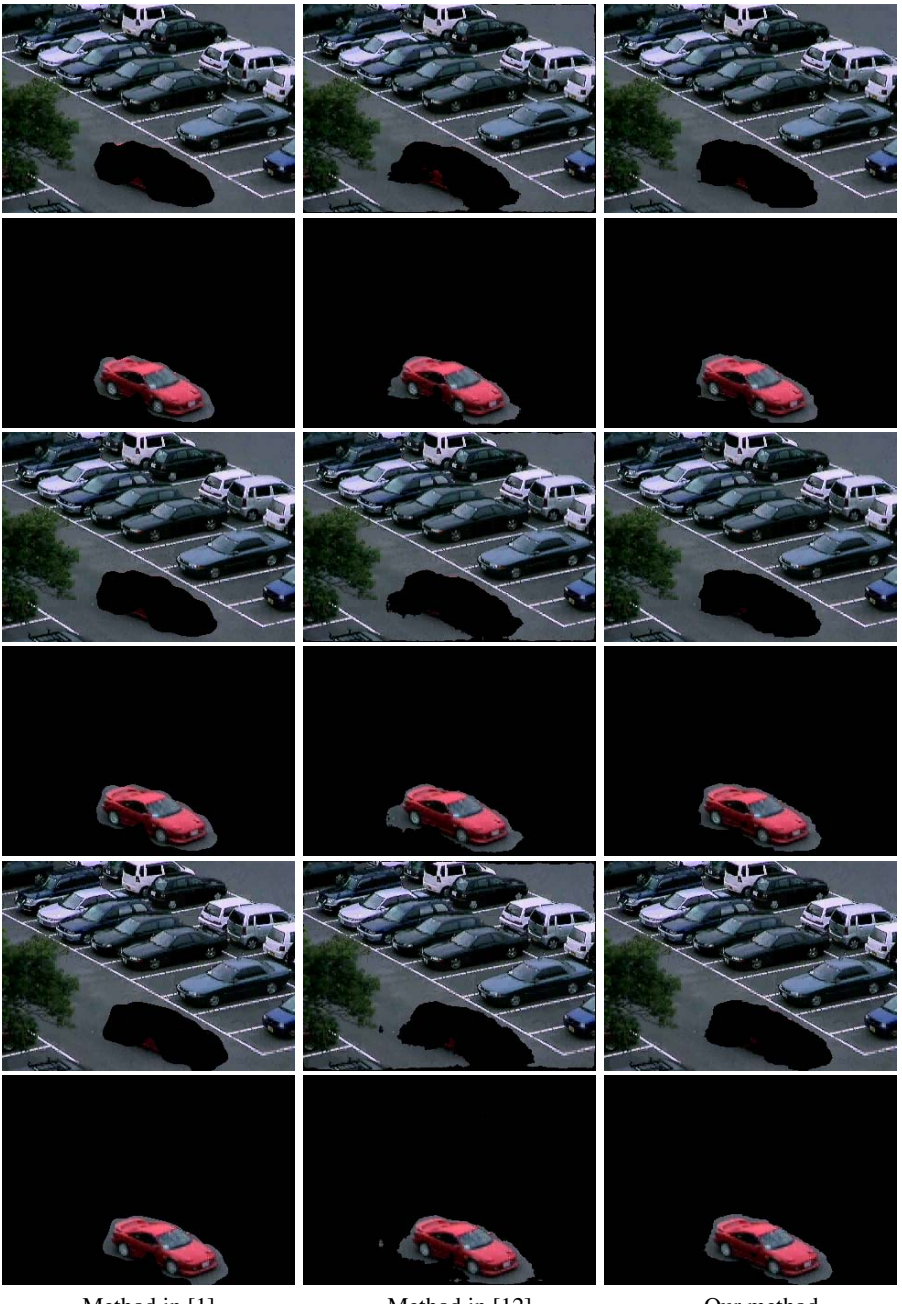
$$\boldsymbol{u}(\boldsymbol{x}_j) = \min_{i=1\ldots n_t} \{\boldsymbol{u}_i : \sum_{\boldsymbol{x}_k \in \mathcal{W}(\boldsymbol{x}_j)} (\boldsymbol{y}_k^\top \boldsymbol{u}_i)^2\}. \tag{19}$$

In the case of affine motion models, the segmentation of the scene is obtained as follows

$$A(\boldsymbol{x}_j, \boldsymbol{y}_j) = \min_{i=1\ldots n_a} \{A_i : \sum_{\boldsymbol{x}_k \in \mathcal{W}(\boldsymbol{x}_j)} (\boldsymbol{y}_k^\top A_i \boldsymbol{x}_k)^2\}. \tag{20}$$

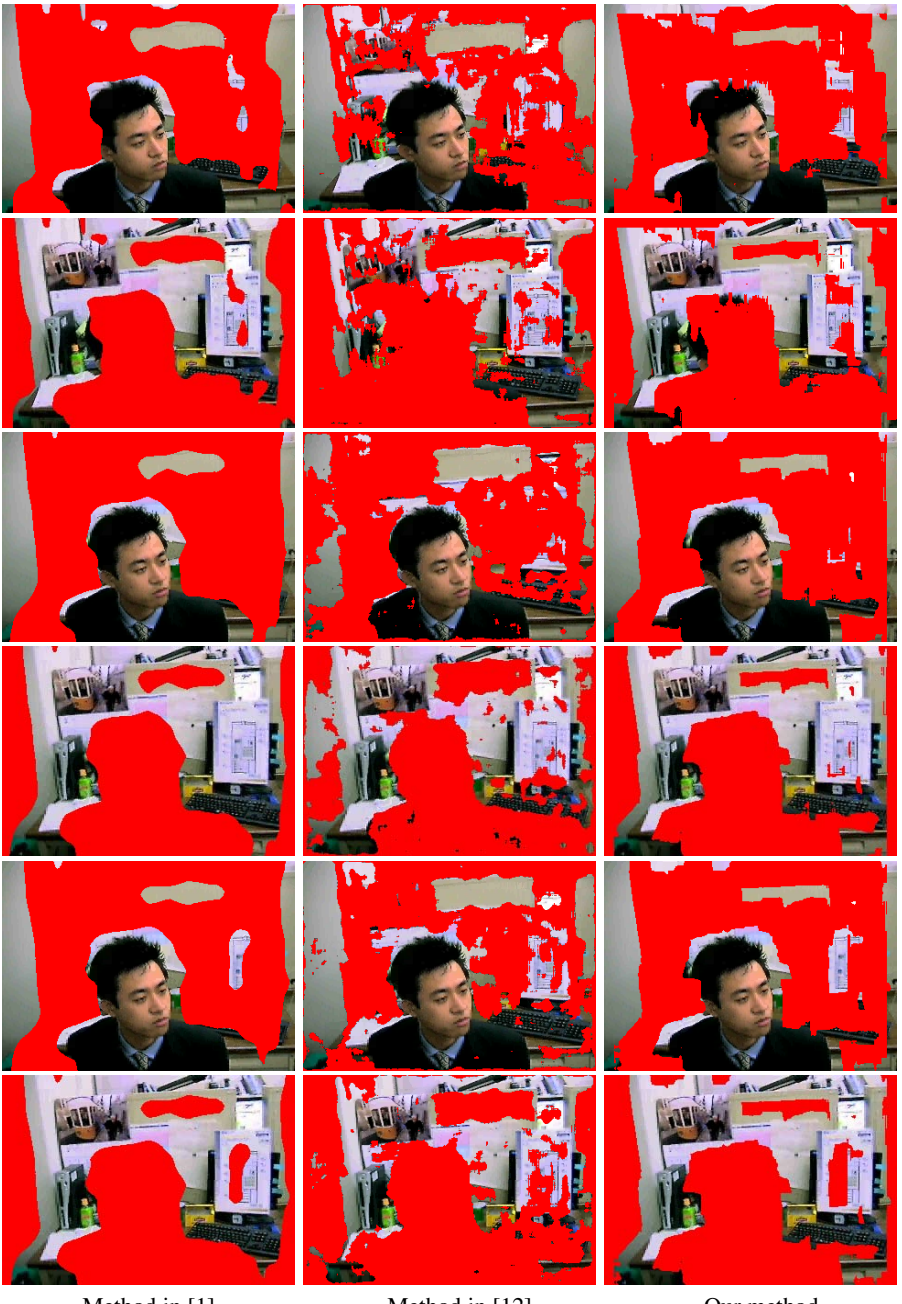
5 Results

In this section, we test our algorithm on real world data and compare its performance with that of the algorithms in [1] and [12]. For all methods, we model the scene as a mixture of 2-D translational motion models. For the method in [12], we post process the segmentation results by spatially smoothing them with a median filter in a window of size 10×10.



Method in [1] Method in [12] Our method

Fig. 1. Segmenting 3 frames from the car-parking lot sequence



Method in [1] Method in [12] Our method

Fig. 2. Segmenting 3 frames from the head-lab sequence

Figure 1 shows an example of segmentation of a 240×320 sequence of a car leaving a parking lot. The scene has 2 motions, the camera's downward motion and the car's right-downward motion. We use a window size of 10×10 to define the local neighborhoods for the method in [1] and for our method. The first and second columns of Figure 1 show the segmentation obtained using the methods in [1] and [12], respectively. The final column shows the results obtained using our method. In each image, the pixels that do not correspond to the group are colored black. Note that the best segmentation results are obtained using our approach. Although the improvement with respect to the method in [1] is not significant, the segmentation of the car is very good as compared to the method in [12] in the sense that very less amount of the parking lot is segmented along with the car.

Figure 2 shows an example of segmentation of a 240×320 sequence of a person's head rotating from right to left in front of a lab background. The scene has 2 motions, the camera's fronto-parallel motion and the head's motion. We use a window size of 20×20 to define the local neighborhoods for the method in [1] and for our method. The first and second columns of Figure 2 show the segmentation obtained using the methods in [1] and [12], respectively. The final column shows the results obtained using our method. In each image, pixels that do not correspond to the group are colored red. Notice that we cannot draw any conclusion for this sequence as to which algorithm performs better, because essentially all the methods misclassify the regions that have low texture. However, our method does perform better than [12] in terms of spatial regularization of the segmentation.

6 Conclusions and Future Work

We have presented a bottom up approach to 2-D motion segmentation that integrates the advantages of both local as well as global approaches to motion segmentation. An important advantage of our method over previous local approaches is that we can account for more than one motion model in every window. This helps us choose a big window without worrying about any aperture problem or motion boundary issues, and also reduces the need for iteratively refining the motion parameters across motion boundaries. An important advantage of our method over global algebraic approaches is that we incorporate spatial regularization into our segmentation scheme and hence we do not need to apply any ad-hoc smoothing to the segmentation results. Future work entails developing a robust algorithm for determining the number of motions in a window.

References

1. Wang, J., Adelson, E.: Layered representation for motion analysis. In: IEEE Conference on Computer Vision and Pattern Recognition. (1993) 361–366
2. Cremers, D., Soatto, S.: Motion competition: A variational framework for piecewise parametric motion segmentation. International Journal of Computer Vision **62** (2005) 249–265
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** (2001) 297–303
9. 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
10. Vidal, R., Sastry, S.: Segmentation of dynamic scenes from image intensities. In: IEEE Workshop on Motion and Video Computing. (2002) 44–49
11. 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
12. 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