Near Real-Time Motion Segmentation Using Graph Cuts

Thomas Schoenemann and Daniel Cremers

CVPR Group, University of Bonn
Römerstr. 164, 53117 Bonn, Germany
{schoenemann, dcremers}@iai.uni-bonn.de

Abstract. We present a new approach to integrated motion estimation and segmentation by combining methods from discrete and continuous optimization. The velocity of each of a set of regions is modeled as a Gaussian-distributed random variable and motion models and segmentation are obtained by alternated maximization of a Bayesian a-posteriori probability. We show that for fixed segmentation the model parameters are given by a closed-form solution. Given the velocities, the segmentation is in turn determined using graph cuts which allows a globally optimal solution in the case of two regions. Consequently, there is no contour evolution based on differential increments as for example in level set methods. Experimental results on synthetic and real data show that good segmentations are obtained at speeds close to real-time.

1 Introduction

Since the seminal works of Lucas and Kanade [17] and Horn and Schunck [13], motion estimation has become one of the major problems addressed in Computer Vision. Motion estimation techniques can be employed in numerous Computer Vision tasks such as the study of dynamical processes [14].

A closely related problem is motion *segmentation*, namely the grouping of image regions which are similar in their motion. Early approaches worked by first estimating the flow field, then segmenting it (cf. [19]). More recently, particularly since the work of [18] and [9], approaches to address the problems of motion estimation and segmentation by minimizing a single energy functional have become popular. Minimization is done by alternatingly updating the flow field and the segmentation boundary.

Most present approaches handle the case of piecewise affine motion [9,2,15] or allow non-parametric variation of the flow field [8,1]. In this work we present models with both piecewise constant and piecewise affine velocities.

Current approaches to motion segmentation are typically based on pde evolution such as the Level Set Method [9,8,1]. While there exist motion segmentation methods using graph cuts [2,15], they are based on non-linear flow-errors, resulting in run-times far from real-time. We use linearized flow errors, allowing accurate segmentations at speeds close to real-time at the costs of less accurate

K. Franke et al. (Eds.): DAGM 2006, LNCS 4174, pp. 455–464, 2006.

motion estimates. Given the velocity in each region, the globally optimal bilayer segmentation can be obtained in effectively linear time.

This paper is organized as follows: In Section 2 we derive an energy functional for motion segmentation with piecewise constant velocities by modeling the velocity of each region as a Gaussian-distributed random variable. Section 3 extends this framework to piecewise affine velocities. In Section 4 we propose an efficient optimization scheme, combining graph cuts and differential methods. Finally, we present in Section 5 experimental results on synthetic and real data which demonstrate that high-quality purely motion based segmentations can be obtained with run-times close to real time.

2 Statistical Formulation of Motion Segmentation

Given two frames I_1 and I_2 of a video sequence, we are interested in determining for each pixel $\mathbf{p} = (x_p, y_p)$ in the first image where it is to be found in the second, i.e. with what velocity it moved. In this work we deal with motion segmentation and for the sake of efficiency fix the number of regions to 2. Notice that an extension to multiple regions is easily possible using the expansion moves of [7]. In this case, however, we can no longer guarantee globally optimal segmentations as the multilabel problem is NP-hard. For the moment, each region i is associated a constant velocity $\mathbf{v}_i, i \in \{0, 1\}$. The next section will extend this to more elaborate parametric motion models. The problem then is to assign each pixel $\mathbf{p}$ a region $l(\mathbf{p}) \in \{0, 1\}$ and determine the optimal velocity for each region. We denote by R_i the set of all pixels labeled i. We address this problem by the Bayesian method of minimizing the negative logarithm of the posterior probability

$$\begin{aligned} &\arg\min_{l,\bar{\mathbf{v}}} \; -\log(pr(l, \bar{\mathbf{v}}|I_1, I_2)) \\ &= \arg\min_{l,\bar{\mathbf{v}}} \left[\underbrace{-\log(pr(l))}_{E_{smooth}(l)} \underbrace{-\log(pr(I_2|\bar{\mathbf{v}}, l, I_1))}_{E_{data}(l,\bar{\mathbf{v}})} - \log(\underbrace{pr(\bar{\mathbf{v}}, I_1|l)}_{uniform}) + \text{const} \right] \end{aligned} \quad (1)$$

where $\bar{\mathbf{v}}$ contains all velocities. In this work, we assume the third probability to be uniform within a reasonable range and assume that the first only depends on the length of the boundary. Based on the Cauchy-Crofton formula from integral geometry this length can be approximated by [6]

$$E_{smooth}(l) = \frac{\nu}{2} \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{N}} \frac{(1 - \delta(l(\mathbf{p}), l(\mathbf{q})))}{||\mathbf{p} - \mathbf{q}||} \quad (2)$$

with Kronecker-δ, the free parameter ν and a neighborhood system $\mathcal{N}$ we choose to be of size 8. Assuming the intensities of the moving objects to stay constant, $pr(\bar{\mathbf{v}}, I_2|l, I_1)$ might be restricted to be non-zero only if

$$0 = I_2(\mathbf{p} + \mathbf{v}_{l(\mathbf{p})}) - I_1(\mathbf{p}) \approx \nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_{l(\mathbf{p})} + I_t(\mathbf{p}) \quad \forall\, \mathbf{p} \quad (3)$$

The last quantity is known as *linearized flow-error*. However, there are several reasons for this assumption not to be true: For one, there is the camera noise. Then there are changes in lighting and reflections. Lastly, we assume that each region has a constant velocity, but desire to segment objects with partially different depths (think of a mirror on a car, or simply a sphere) as one. However, parts with different depths will move with different velocities in the image plane.

Nevertheless, we desire the constraint in (3) to be fulfilled as good as possible and assume the probability $pr(I_2|\bar{\mathbf{v}}, l, I_1)$ to be only dependent on its deviation from 0 for all pixels.

It can be shown that the likelihood proposed by [17] is equivalent to the assumption of Gaussian noise on the image data, where the noise is independent of the region. This results in a Gaussian-distributed flow-error

$$E_{data}(l, \bar{\mathbf{v}}) = \frac{1}{2} \sum_{i=0}^{1} \sum_{\mathbf{p} \in R_i} (\nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_i + I_t(\mathbf{p}))^2 \tag{4}$$

In this work, following [10] we assume that the velocity at each pixel $\mathbf{p}$ of region i is a Gaussian-distributed random variable, that is $\tilde{\mathbf{v}}_i(\mathbf{p}) = \mathbf{v}_i + \boldsymbol{\eta}(\mathbf{p})$ for $\mathbf{p} \in R_i$, where $\boldsymbol{\eta}(\mathbf{p}) \sim N(\mathbf{0}, \sigma_i^2 \mathbf{I}_2)$ with identity matrix I_2. We require that $\tilde{\mathbf{v}}_i(\mathbf{p})$ exactly fulfill the constraint in (3), resulting in

$$\nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_i + I_t(\mathbf{p}) = \nabla \mathbf{I}(\mathbf{p})^T \boldsymbol{\eta}(\mathbf{p})$$

The flow-error for $\mathbf{v}_i$ is now $N(0, \sigma_i^2 ||\nabla \mathbf{I}(\mathbf{p})||^2)$-distributed. This leads to

$$E_{data}(l, \bar{\mathbf{v}}, \boldsymbol{\sigma}) = \frac{1}{2} \sum_{i=0}^{1} \sum_{\mathbf{p} \in R_i} \left[\log(2\pi ||\nabla \mathbf{I}(\mathbf{p})||^2 \sigma_i^2) + \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_i + I_t(\mathbf{p}))^2}{||\nabla \mathbf{I}(\mathbf{p})||^2 \sigma_i^2} \right] \tag{5}$$

where $\boldsymbol{\sigma}$ contains all variances. In contrast to [10] we allow a separate variance for each region. Consequently the normalization term cannot be neglected. Lacking suitable estimates for the variances, we optimize it in each region. To avoid numerical instabilities, we replace $||\nabla \mathbf{I}(\mathbf{p})||^2$ by $\max\{||\nabla \mathbf{I}(\mathbf{p})||^2, 1\}$ in the denominator.

3 Extension to Parametric Motion Models

In this section we extend the data term in (5) to piecewise affine motion. A pixel $\mathbf{p} = (x_p, y_p)$ belonging to R_i is now no longer assigned the constant velocity $\mathbf{v}_i$, but an affine velocity $\mathbf{S}(\mathbf{p})\boldsymbol{\vartheta}_i$ where

$$\mathbf{S}(\mathbf{p}) = \begin{pmatrix} x_p & y_p & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_p & y_p & 1 \end{pmatrix}$$

and $\boldsymbol{\vartheta}_i$ is the vector containing all parameters of the affine model for region i:

$$\boldsymbol{\vartheta}_i = \begin{pmatrix} a_i & b_i & c_i & d_i & e_i & f_i \end{pmatrix}^T$$

We differ from the common notation $\mathbf{Ap} + \mathbf{b}$ as this simplifies the equations for the update of parameters greatly. The task is now to minimize the energy functional (1) with data term

$$E_{data}(l, \bar{\mathbf{v}}, \boldsymbol{\sigma}) = \frac{1}{2} \sum_{i=0}^{1} \sum_{\mathbf{p} \in R_i} \left[\log(2\pi ||\nabla \mathbf{I}(\mathbf{p})||^2 \sigma_i^2) + \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_i + I_t(\mathbf{p}))^2}{||\nabla \mathbf{I}(\mathbf{p})||^2 \sigma_i^2} \right] \tag{6}$$

with respect to the vectors $\boldsymbol{\vartheta}_i$, the variances $\boldsymbol{\sigma}$ and the segmentation l.

4 An Efficient Semi-discrete Optimization Scheme

Minimization of the energy functional in (1) is done by alternatingly updating the segmentation, the velocities and the variances. To this end we propose a semi-discrete optimization scheme, combining fast discrete global optimization methods for the segmentation step with continuous optimization for the motion parameters. In the following we first state how the segmentation is updated using graph cuts. We then give closed form global solutions for the velocities and variances. Each quantity is set to the globally optimal one given the others. In all cases we show the update for the affine motion model.

4.1 Fast Global Segmentation Via Graph Cuts

Given the velocities of each region the segmentation step requires the minimization of a cost functional with binary-valued variables. To solve this problem, we revert to the graph cut method, which will be detailed in the following.

Greig et al. [12] were the first to show how to exploit the graph cut technique for problems of Computer Vision. They were concerned with the problem of binary image restoration. In [16] the minimization of submodular functions of binary variables with at most ternary terms is discussed[1].

The complexity of the general problem is low-order polynomial, but using the fast algorithm of [5] for most Computer Vision problems (including ours) it is effectively linear. This algorithm makes use of the theorem of Ford and Fulkerson [11] by solving the related problem to compute the maximum flow in a graph.

To give the reader an intuition of how the method works, we explain it in the following. We state here the problem for undirected graphs as this suffices for our application. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. An edge $e = \{p, q\}$ links two nodes p and q. For the problem of the minimum 0/1-cut there are two distinguished nodes 0 and 1, i.e. $\mathcal{V} = \{0, 1\} \cup \mathcal{V}_0$. In our case $\mathcal{V}_0$ will correspond to the set of pixels. Furthermore, each edge $\{p, q\}$ is assigned a weight $w_{\{p,q\}}$.

[1] Code is available at `http://www.adastral.ucl.ac.uk/~vladkolm/software.html`

A cut on $\mathcal{G}$ is a labeling l of all nodes such that $l(v) \in \{'0', '1'\}$ for $v \in \mathcal{V}$ and $l(0) =' 0'$ and $l(1) =' 1'$. The costs $|l|$ of a cut is the sum of the weights of all edges between a node labeled '0' and one labeled '1':

$$|l| = \sum_{\{p,q\} \in \mathcal{E}: l(p) =' 0', l(q) =' 1'} w_{\{p,q\}}$$

The min-cut problem is to find a cut l with minimal costs. As it is inherently related to finding a binary labeling, many binary optimization problems of Computer Vision can be reduced to it, including ours.

In the following we show how the graph looks like for motion segmentation with two regions 0 and 1. In our case $\mathcal{V}_0$ is the set of all pixels. Each pixel p is linked to 0 by an edge $\{0, p\}$ and to 1 by an edge $\{p, 1\}$. These links are called *t-links*. Additionally there are *n-links* connecting pixels p and q for any $\{p, q\} \in \mathcal{N}$. Their weight is set to $w_{\{p,q\}} = \nu$. Setting

$$w_{\{0,p\}} = \log(2\pi\sigma_1^2 ||\nabla \mathbf{I}(\mathbf{p})||^2) \; + \; \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_1 + I_t(\mathbf{p}))^2}{\sigma_1^2 ||\nabla \mathbf{I}(\mathbf{p})||^2}$$

$$w_{\{p,1\}} = \log(2\pi\sigma_0^2 ||\nabla \mathbf{I}(\mathbf{p})||^2) \; + \; \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_0 + I_t(\mathbf{p}))^2}{\sigma_0^2 ||\nabla \mathbf{I}(\mathbf{p})||^2}$$

the reader may verify that the costs of any cut l correspond to the costs of a segmentation l' where $l'(p) = 0$ if $l(p) =' 0'$ and $l'(p) = 1$ if $l(p) =' 1'$ (see figure 1 for an example on a one-dimensional image). Hence, using graph cuts the globally optimal segmentation can be computed in one step. Our efficient implementation uses flow-recycling [4] where previously computed flows are reused and the t-links are updated in each iteration.

4.2 Update of Continuous Parameters

As suggested in [17,10] minimization with respect to the motion parameters $\mathbf{v}_i, \sigma_i$ can be done by setting the respective derivatives of (5) to zero. This leads to $\mathbf{v}_i = \mathbf{M}^{-1}\mathbf{b}$ with

$$\mathbf{M} = \sum_{\mathbf{p} \in R_i} \frac{\mathbf{S}(\mathbf{p})^T \nabla \mathbf{I}(\mathbf{p}) \nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p})}{||\nabla \mathbf{I}(\mathbf{p})||^2}$$

$$\mathbf{b} = -\sum_{\mathbf{p} \in R_i} \frac{\mathbf{S}(\mathbf{p})^T \nabla \mathbf{I}(\mathbf{p}) \cdot I_t(\mathbf{p})}{||\nabla \mathbf{I}(\mathbf{p})||^2}$$

The variance is given by

$$\sigma_i^2 = \frac{1}{|R_i|} \cdot \sum_{\mathbf{p} \in R_i} \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_i + I_t(\mathbf{p}))^2}{||\nabla \mathbf{I}(\mathbf{p})||^2}$$

So all quantities (segmentation, velocities and variances) are set to the globally optimal solution given the other quantities. Notice that convergence is guaranteed as the energy never increases and is always strictly positive.

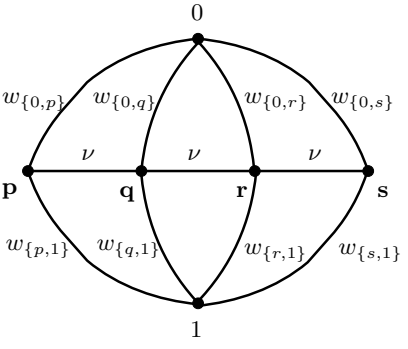


Fig. 1. An example of a graph for the segmentation problem on a 1-D image

5 Experimental Results

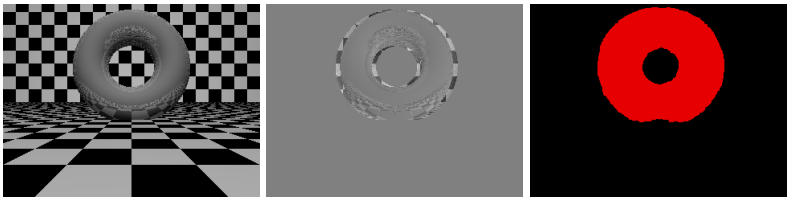
To show the abilities of the proposed model, we deal first with synthetic data, showing that surfaces with non-constant depth can be handled. Afterwards we present results on real-life data. All images were pre-smoothed. To initialize the segmentation process, we applied block-matching. Therefore the image was segmented into non-overlapping blocks, each block was tested against a set of integer-valued velocities and assigned the best one. Afterwards the two most frequent velocities were taken.

The only free parameter is the length penalty ν. For all experiments we chose the ν that gave the best performance. Notice that in contrast to existing pde-based methods, all minimization processes have been run until convergence.

5.1 Experiments with Synthetic Data

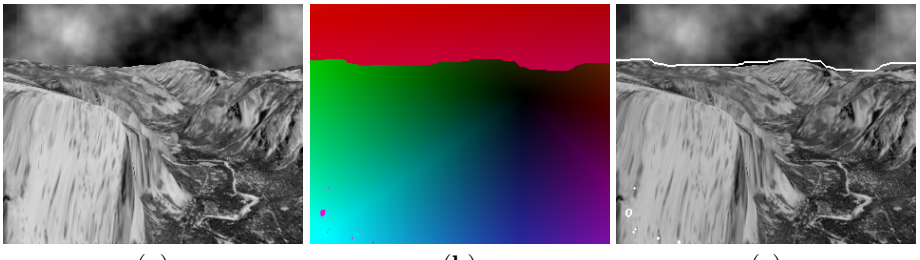
Figure 2 (**a**) shows an artificial scene with reflections (but without shadows). The difference image in part (**b**) shows that only the torus moved (to the right). The flow field depicted in part (**c**) shows that the proposed motion segmentation model is able to handle reflections and surfaces of non-constant depth. The affine model is not needed here. For all images used to visualize flow fields the color hue indicates the direction of movement whereas the intensity indicates the strength.

Additionally we tested the proposed model on the well-known Yosemite Sequence as shown in figure 3. As is common, we measure the motion between frame 8 (displayed in part **a**) and frame 9. The difference image gives little information here. Parts (**b**) and (**c**) display the segmentation result and the flow field, respectively. As desired, the clouds are separated from the mountains and the valley (except for some small regions in the lower left corner). The average angular error amounts to 11.27. While this error is larger than those reported for non-parametric motion models [8,1], it is still a good value considering the simplicity of the method.



(**a**) (**b**) (**c**)

Fig. 2. (**a**) first frame of an artificial scene (**b**) difference image to second frame (**c**) motion segmentation by piecewise constant velocity, $\nu = 9.5$



(**a**) (**b**) (**c**)

Fig. 3. Results for motion segmentation on the Yosemite Sequence between frame 8 (**a**) and frame 9 (not shown) of the sequence. (**b**) segmentation obtained with the affine model, $\nu = 9.5$. The white line indicates the boundary of the two regions (**c**) visualization of the flow field.

5.2 Experiments with Real Data

Here we present results on real data. We start with the well-known Flower Garden Sequence shown in figure 4 (**a**). The motion is estimated between frames 1 and 2. Parts (**c-f**) show the segmentations and flow fields obtained with constant and affine velocities. The affine model gives the better flow field as the background decreases smoothly in depth. For the segmentation this does not matter much. Figure 5 depicts how the flow field evolves at different stages of the minimization process. We only show this for constant velocities.

When applying the model with constant velocities to every pair of consecutive frames, we obtain an average run-time of 825 msec per frame pair on the full resolution of 360×240 on a 3.4 GHz machine. When reducing the resolution to 180×120, this reduces to 180 msec per frame[2]. If the number of iterations is reduced to 2 for each frame pair (excepting the first), 80 msec or 12.5 fps are achieved, which is good enough for real time. Actually, even 1 iteration per frame pair gives sufficient quality, yielding 16.5 fps. Admittedly, the quality reduces slightly compared to full resolution.

[2] The length penalty is then divided by 2 as that is the relative decrease of the length of a line compared to the number of pixels.

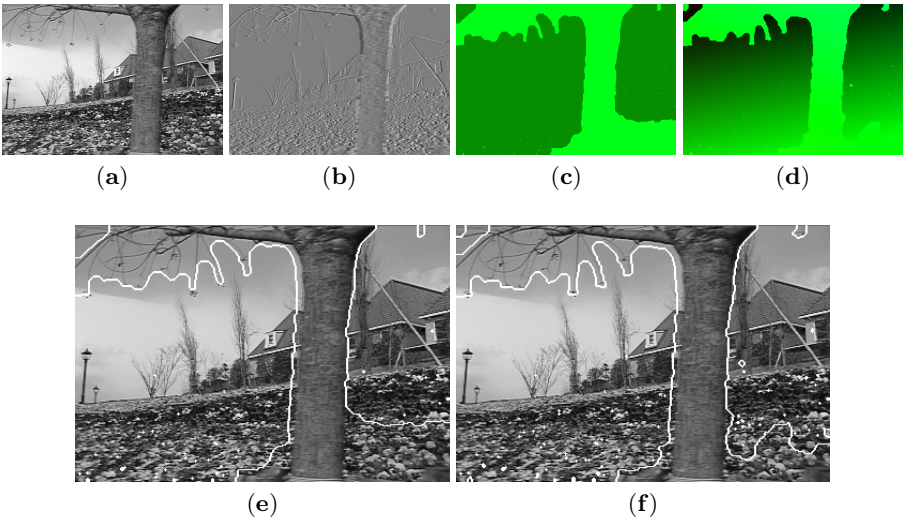


Fig. 4. Results on the Flower Garden Sequence (**a**) frame 1 of the sequence (**b**) difference image to second frame (**c**) flow field obtained with constant velocities, $\nu = 4$ (**d**) flow field obtained with affine velocities, $\nu = 4$. Note that the velocity of the background gradually decays toward the top of the image. (**e+f**) segmentations obtained in (c) and (d). The white lines indicate segmentation boundaries.

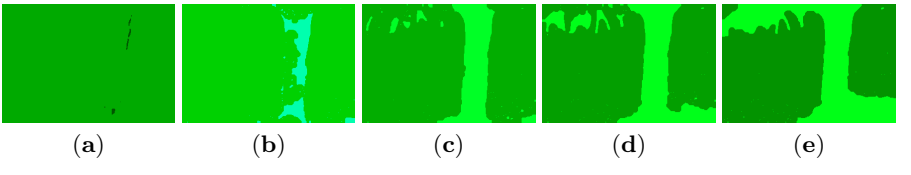


Fig. 5. Intermediate steps for minimizing the energy functional with constant velocities. 14 iterations were needed until convergence. Here the flow fields after iteration 1 (**a**), 3 (**b**), 5 (**c**), 7 (**d**) and 12 (**e**) are shown. Compare figure 4 (**c**) for the final result.

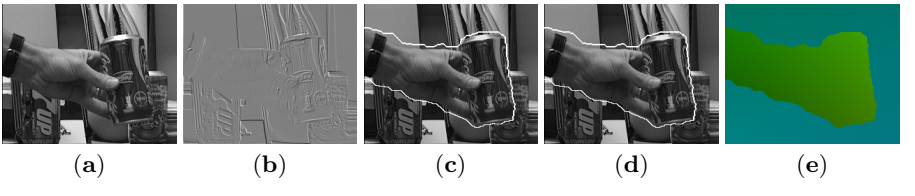


Fig. 6. Results for the Pickup Sequence (**a**) first frame (**b**) difference image to second frame (**c**) segmentation for the model with constant velocities, $\nu = 3$ (**d**) segmentation for the affine model, $\nu = 3$ (**e**) flow field for the affine model

About 40% of the run-time are used for the computation of the minimum cut. The other major sources are the computation of edge costs and the update of velocities. The affine model more than doubles the run-time.

Lastly, we present results on the Pickup Sequence from [3], which we modified by introducing artificial motion of the whole image (originally only the hand with the can moved). The first frame and the difference image are show in figure 6 (**a**) and (**b**). Part (**c**) shows that the model with constant velocities is able to separate the hand and the can from the background. As can be seen in part (**d**) the affine model produces a similar segmentation, but part (**e**) reveals that the lower parts of the arm are assigned lower velocities as their depth is greater.

6 Conclusion

We proposed an efficient semi-discrete optimization method for motion segmentation. Based on the assumption that the velocity in each region can be modeled as a Gaussian distributed random variable, we derived a cost functional for the joint estimation and segmentation of piecewise constant or piecewise affine motion. For the case of two motion models, we developed a fast minimization scheme which alternates a globally optimal segmentation via graph cuts with a globally optimal motion estimation. Experiments show that for moderate resolutions accurate purely motion-based segmentations can be obtained in real-time.

Acknowledgments. This work was supported by the German Research Foundation, grant #CR-250/1-1. We thank Thomas Brox for constant support and many helpful discussions and Kalin Kolev for helpful comments on the code.

References

1. T. Amiaz and N. Kiryati. Dense discontinuous optical flow via contour-based segmentation. In *Int. Conf. on Image Processing*, volume 3, pages 1264–1267, Genova, Italy, Sept. 2005.
2. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *IEEE Int. Conf. on Computer Vision*, pages 489–495, 1999.
3. M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Europ. Conf. on Computer Vision*, LNCS, pages 329–342, Cambridge, England, Apr. 1996. Springer.
4. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE Int. Conf. on Computer Vision*, volume 1, pages 105–112, 2001.
5. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In A. J. M. Figueiredo, J. Zerubia, editor, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of *LNCS*, pages 359–374. Springer, 2001.
6. Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *IEEE Int. Conf. on Computer Vision*, 2003.
7. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(11):1222–1239, 2001.

8. T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In *Europ. Conf. on Computer Vision*, Graz, Austria, May 2006. Springer.
9. D. Cremers and S. Soatto. Motion Competition: A Variational Framework for Piecewise Parametric Motion Segmentation. *Int. J. of Computer Vision*, 62(3):249–265, May 2005.
10. D. Cremers and A. L. Yuille. A Generative Model Based Approach to Motion Segmentation. In B. Michaelis and G. Krell, editors, *Pattern Recognition (Proc. DAGM)*, volume 2781 of *LNCS*, pages 313–320, Magdeburg, Sept. 2003. Springer.
11. L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, New Jersey, 1962.
12. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum *a posteriori* estimation for binary images. *J. Roy. Statist. Soc., Ser. B.*, 51(2):271–279, 1989.
13. B. Horn and B. Schunck. Determining optical flow. *A.I.*, 17:185–203, 1981.
14. B. Jähne, H. Haußecker, H. Scharr, H. Spies, D. Schmundt, and U. Schurr. Study of dynamical processes with tensor–based spatiotemporal image processing techniques. In H. Burkhardt and B. Neumann, editors, *Europ. Conf. on Computer Vision*, volume 1407 of *Lect. Not. Comp. Sci.*, pages 322–336. Springer, 1998.
15. O. Juan. *On Some Extensions of Level Sets and Graph Cuts & Their Applications to Image and Video Segmentation*. PhD thesis, École Nationale des Ponts et Chaussées, May 2006.
16. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24(5):657–673, 2004.
17. B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc.7th International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, 1981.
18. C. Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *Int. J. of Computer Vision*, 6(1):25–38, 1991.
19. J. Wang and E. Adelson. Representating moving images with layers. *IEEE Trans. on Image Processing*, 3(5):625–638, 1994.