# Segmentation-Based Motion with Occlusions Using Graph-Cut Optimization

Michael Bleyer*, Christoph Rhemann, and Margrit Gelautz

Institute for Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria
{bleyer, gelautz}@ims.tuwien.ac.at

**Abstract.** We propose to tackle the optical flow problem by a combination of two recent advances in the computation of dense correspondences, namely the incorporation of image segmentation and robust global optimization via graph-cuts. In the first step, each segment (extracted by colour segmentation) is assigned to an affine motion model from a set of sparse correspondences. Using a layered model, we then identify those motion models that represent the dominant image motion. This layer extraction task is accomplished by optimizing a simple energy function that operates in the domain of segments via graph-cuts. We then estimate the spatial extent that is covered by each layer and identify occlusions. Since treatment of occlusions is hardly possible when using entire segments as matching primitives, we propose to use the pixel level in addition. We therefore define an energy function that measures the quality of an assignment of segments *and* pixels to layers. This energy function is then extended to work on multiple input frames and minimized via graph-cuts. In the experimental results, we show that our method produces good-quality results, especially in regions of low texture and close to motion boundaries, which are challenging tasks in optical flow computation.

## 1 Introduction

The estimation of optical flow is one of the oldest, but still most active research topics in computer vision. Major challenges are twofold. Firstly, matching often fails in the absence of discriminative image features that can be uniquely matched in the other view. This is the case in *untextured regions* and in the presence of texture with only a single orientation (*aperture problem*). Secondly, a pixel's matching point can be *occluded* in the other view. Those occlusions often occur at motion discontinuities, which makes it specifically challenging to precisely outline object boundaries. In spite of its obvious importance, standard optical flow approaches still tend to ignore the occlusion problem (e.g., [1,2,3]).

This paper proposes an algorithm that explicitly addresses these problems by taking advantage of *colour segmentation* and robust optimization via *graph-cuts*. Our contribution lies in that we show how to set up an energy function

* This work was funded by the Austrian Science Fund (FWF) under project P15663.

K. Franke et al. (Eds.): DAGM 2006, LNCS 4174, pp. 465–474, 2006.

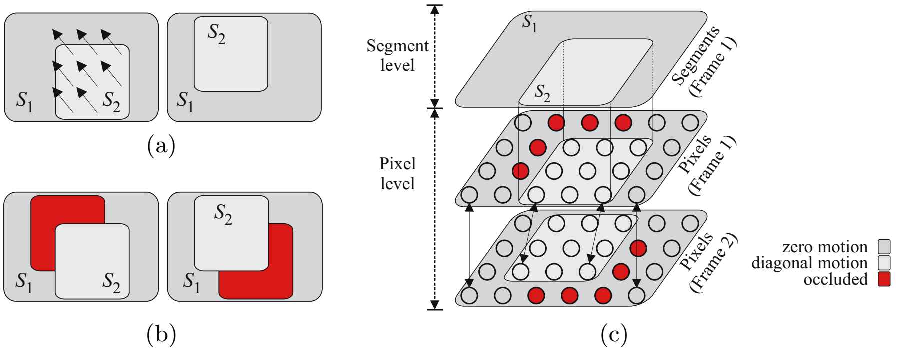

**Fig. 1.** The occlusion problem in segmentation-based matching and our proposed solution. Explanation is given in the text.

that formulates segmentation-based motion with treatment of occlusions. The advantage of this energy function is that it can be optimized via robust graph-cut-based optimization. The motivation for using colour segmentation is that energy minimization approaches often bias towards the reconstruction of simple object shapes and consequently fail in the presence of complex motion discontinuities. To explain the idea behind our energy function, let us consider the two views illustrated in Fig. 1a. The images show two segments $S_1$ and $S_2$ at different instances of time with segment $S_2$ undergoing motion as indicated by the arrows. As a consequence of the moving foreground object, occlusions occur in both frames (coloured red in Fig. 1b). $S_1$ is partially affected by occlusions, which is problematic in the following sense. When using segments as matching primitives, we can only state that the *complete* segment $S_1$ has zero motion. However, we cannot express the fact that some pixels of $S_1$ are affected by occlusion. In other words, *occlusions cannot be dealt with in the domain of segments.*

In order to correctly model occlusions, we propose an energy function that operates on two levels, one representing the extracted segments and the other representing pixels. In addition to all segments (top layer of Fig. 1c), we as well assign every pixel of the reference image to a motion model (middle layer of Fig. 1c). The basic idea is to enforce that every (visible) pixel is assigned to the same motion model as the segment to which it belongs. However, and this is the important point, a pixel is also allowed to be occluded. Finally, we as well include every pixel of the second image into our problem formulation (bottom layer of Fig. 1c). We enforce that a (visible) pixel and its matching point in the other image must both have identical motion model assignments. This constraint serves to implement the uniqueness assumption [4]. This assumption is used to identify occlusions symmetrically in both images.

In relation to prior work, using colour segmentation for the dense correspondence problem does not represent a novel idea. Black and Jepson [5] propose a colour segmentation-based motion algorithm that fits a variable order parametric model to each individual segment using a precomputed flow field. Analogous to our approach, the basic idea behind this procedure is that the flow field is likely to vary smoothly inside such a segment. However, the authors do not account for the occlusion problem and miss to model smoothness across segments. Recently, segmentation-based techniques have also gained attention in the stereo community (e.g., [6,7]). Although quite different from each other, segmentation-based stereo methods take benefit from increased robustness in untextured regions and in areas close to disparity discontinuities. This is well reflected by the good experimental results of those algorithms. For the motion layer extraction problem, segmentation-based techniques using clustering methods are proposed in [8,9].

In the context of energy minimization approaches, our technique is most closely related to various motion segmentation algorithms. Ayer and Sawhney [10] employ the minimum description length (MDL) encoding principle in order to derive the smallest set of layers necessary to describe the image motion. They formulate statistical cost functions that are optimized by an expectation maximization algorithm. Willis et al. [11] present a graph-cut-based approach to achieve a dense and piecewise smooth assignment of pixels to layers. They do, however, not explicitly model the occlusion problem. In contrast to this, Xiao and Shah [12] embed occlusion detection into a graph-cut-based method in a very recent work. They claim to be the first ones to deal with the explicit identification of occluded pixels for the motion segmentation task. The most obvious difference to those approaches is that none of them uses image segmentation.

Among prior work, the closest related one originates from literature on the simpler stereo correspondence problem. Hong and Chen [7] combine colour segmentation-based matching with graph-cut optimization. They heuristically identify occlusions in a preprocessing step, which then allows them to model the correspondence problem on the segment level only. However, the results of this method depend on the success of this preprocessing step, and it is not clear how well an a-priori identification of occlusions can work, especially in the presence of large motion. In contrast to this, our energy function knows about the existence of occlusions. Flow vectors and occlusions are computed simultaneously, which we believe results in a more accurate reconstruction of both.

## 2 Our Approach

### 2.1 Colour Segmentation and Initial Models

In the first step, we apply colour segmentation to the reference image. Since our basic assumption states that the flow values inside a colour segment vary smoothly, it is important that a segment does not overlap a motion discontinuity. It is therefore safer to use oversegmention (Fig. 2b). In the current implementation, we apply the mean-shift-based segmentation algorithm described in [13].

The optical flow inside each segment is modelled by affine motion, which is

$$\begin{aligned} V_x(x,y) &= a_{x0} + a_{xx}x + a_{xy}y \\ V_y(x,y) &= a_{y0} + a_{yx}x + a_{yy}y \end{aligned} \tag{1}$$

with $V_x$ and $V_y$ being the x- and y-components of the flow vector at image coordinates $x$ and $y$ and the $a$'s denoting the six parameters of the model. However, our approach could easily be extended to a more sophisticated model. To initialize the motion of each segment, a set of sparse correspondences is computed using the KLT-tracker [14]. A segment's affine parameters are then derived by least squared error fitting to all correspondences found inside this segment. We apply the iterative plane fitting algorithm described by Tao et al. [6] to reduce the sensitivity of the least squared error solution to outliers.

### 2.2 Layer Extraction

When using a layered representation [15], the first questions one has to answer are: How many layers are present in the sequence and what are their motion parameters? Initially, the set of our layers $\mathcal{L}$ is built by all motion models found in the previous step. In order to extract a small set of layers out of $\mathcal{L}$, we minimize a simple energy function $E(f)$, which measures the optimality of an assignment $f$ of segments to layers, in the form of

$$E(f) = E_{data}(f) + E_{smooth}(f). \tag{2}$$

The data term $E_{data}$ calculates how well $f$ agrees with the input images and is defined by

$$E_{data}(f) = \sum_{S \in \mathcal{R}} \sum_{p \in S} d(p, m[f(S)](p)) \tag{3}$$

with $\mathcal{R}$ being the set of all segments of the reference view and $f(S)$ being the index of the layer to which segment $S$ is assigned. We write $m[k](p)$ to denote the matching point of a pixel $p$ in the other view according to the $k$th layer. More precisely, $m[k](p)$ is derived by computing the displacement vector at $p$ using the affine parameters of the layer at index $k$ (equation (1)) and adding it to the coordinates of $p$. The function $d(\cdot, \cdot)$ measures the dissimilarity of two pixels, which is the sum-of-absolute-differences of RGB values in our implementation. The second term $E_{smooth}$ of the energy function measures to which extent the current assignment $f$ is spatially smooth. $E_{smooth}$ is defined by

$$E_{smooth}(f) = \sum_{(S,S') \in \mathcal{N}} \begin{cases} \lambda_{smooth} \cdot b(S, S') & : \quad f(S) \neq f(S') \\ 0 & : \quad \text{otherwise} \end{cases} \tag{4}$$

with $\mathcal{N}$ being all pairs of neighbouring segments, $b(\cdot, \cdot)$ computing the border length between such and $\lambda_{smooth}$ being a constant user-defined penalty.

We approximate the minimum of the energy function in equation (2) using the $\alpha$-expansion algorithm of Boykov et al. [16]. Starting from an arbitrary configuration, we iteratively change this configuration by computing the optimal $\alpha$-expansion move for each layer until convergence. The graph built for calculating the optimal $\alpha$-expansion consists of nodes that correspond to segments. Since the number of segments is significantly lower than the number of pixels, minimization of equation (2) via graph-cuts is quite efficient.

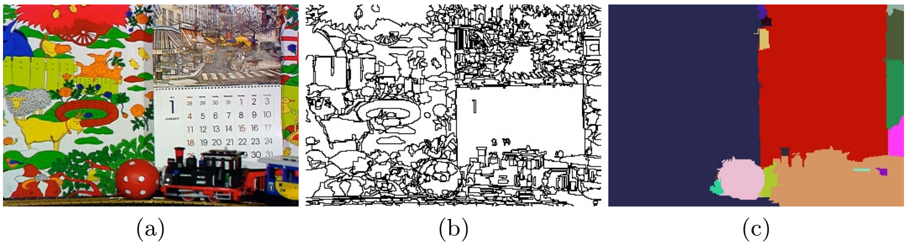

**Fig. 2.** Colour segmentation and layer extraction. (a) Original image. (b) Result of the colour segmentation step. Segment borders are shown. (c) Result of the layer extraction step. Pixels of the same colour belong to the same layer.

Those layers that are not present in the computed configuration $f^*$ are removed from the set of layers $\mathcal{L}$, which drastically decreases the number of layers. However, it is quite likely that the correct layer was not contained in our initial set, due to the small spatial extent over which the motion parameters were initially computed. We therefore refit the layers over their new spatial extents according to the assignment of segments to layers in $f^*$ to derive a set of refined layers $\mathcal{L}'$. We then update $\mathcal{L}$ by $\mathcal{L} := \mathcal{L} \cup \mathcal{L}'$. Starting from the configuration $f^*$, we apply the $\alpha$-expansion algorithm using our refined layer set $\mathcal{L}$ to obtain the new configuration $f^{**}$. We again remove those layers from $\mathcal{L}$ that do not occur in $f^{**}$. If the costs of $f^{**}$ are not lower than those of $f^*$, $\mathcal{L}$ represents our final set of layers. Otherwise, this procedure is iterated.

We show results of the layer extraction step in Fig. 2c. Since the proposed algorithm operates on the segment level only, it is not capable of handling occlusions. It therefore produces artefacts in regions close to motion boundaries. Although there are only small occluded areas in the sequence shown in Fig. 2 such artefacts are visible in the proximity of the rotating ball.[1] However, this strategy works well enough to deliver the dominant image motion and it is computationally efficient.

### 2.3 Layer Assignment

Knowing the set of layers, the task of the assignment step is to estimate which parts of the images are covered by which layers as well as to identify occlusions. As stated in the introduction, the segment level alone is not sufficient for treatment of occlusions. In the following, we therefore design an energy function involving both, the segment and the pixel level. Minimization of the derived objective function via the $\alpha$-expansion algorithm is not discussed in this paper for space limitations, but is thoroughly described in [17].

**Energy Function.** In contrast to the previous section, a configuration $f$ is no longer an assignment of segments to layers, but an assignment of segments

[1] We will present an example where this effect is more severe in the experimental results.

*and* pixels to layers. Moreover, a pixel can be assigned to a dedicated label 0 expressing the fact that the pixel's matching point is occluded in the other view. We define the energy function $E'(f)$ measuring the quality of a configuration $f$, which assigns segments and pixels to layers, by

$$E'(f) = E'_{data}(f) + E'_{segment}(f) + E'_{mismatch}(f) + E'_{smooth}(f). \tag{5}$$

The individual terms of $E'(f)$ are described one after the other in the following.

The first term $E'_{data}$ measures the agreement of $f$ with the input data and is defined by

$$E'_{data}(f) = \sum_{p \in \mathcal{I}} \begin{cases} d(p, m[f(p)](p)) & : \quad f(p) \neq 0 \\ \lambda_{occ} & : \quad \text{otherwise} \end{cases} \tag{6}$$

with $\mathcal{I}$ being the set of all pixels of the reference image $\mathcal{I}_R$ as well as of the second view $\mathcal{I}_S$ and $\lambda_{occ}$ denoting a constant predefined penalty. While $E'_{data}$ measures the pixel dissimilarity for visible pixels, it imposes a penalty on occluded ones. This penalty is necessary, since otherwise declaring all pixels as occluded would result in a trivial minimum of $E'(f)$. To allow for a symmetrical identification of occlusions, $E'_{data}$ operates on both images. The matching point $m[k](p) \in \mathcal{I}_R$ of a pixel $p \in \mathcal{I}_S$ is thereby computed using the inverse motion model of the $k$th layer. The second term $E'_{segment}(f)$ of the energy function enforces the segmentation information on the pixel level and is defined by

$$E'_{segment}(f) = \sum_{p \in \mathcal{I}_R} \begin{cases} \infty & : \quad f(p) \neq 0 \wedge f(p) \neq f(seg(p)) \\ 0 & : \quad \text{otherwise} \end{cases} \tag{7}$$

with $seg(p)$ being a function that returns the segment to which pixel $p$ belongs. The basic idea is that a pixel is either occluded or assigned to the same layer as all other visible pixels of the same segment. Solutions that violate this constraint generate infinite costs. The third term $E'_{mismatch}$ accounts for a consistent layer assignment across the reference and the second images. It is defined by

$$E'_{mismatch}(f) = \sum_{p \in \mathcal{I}} \begin{cases} \lambda_{mismatch} & : \quad f(p) \neq 0 \;\wedge\; f(p) \neq f(m[f(p)](p)) \\ 0 & : \quad \text{otherwise} \end{cases} \tag{8}$$

with $\lambda_{mismatch}$ being a user-set penalty. This penalty is imposed for each pixel $p$ whose matching point is assigned to a different layer than that of $p$. Finally, we apply the smoothness assumption on the segment level. $E'_{smooth}$ is identical to the smoothness term of the previous section. For completeness, we write:

$$E'_{smooth}(f) = E_{smooth}(f). \tag{9}$$

**Extension to Multiple Input Frames.** The energy function of equation (5) is designed to be used with only two input images. However, oftentimes frames in between these two images are available as well and can be used to improve the matching results. Let $I_1$ and $I_n$ be the first and last views of a short video

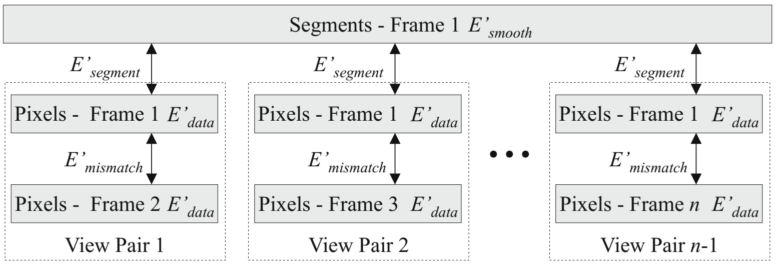
Segments - Frame 1 $E'_{smooth}$
$E'_{segment}$
$E'_{segment}$
$E'_{segment}$
Pixels - Frame 1 $E'_{data}$
$E'_{mismatch}$
Pixels - Frame 2 $E'_{data}$
View Pair 1
Pixels - Frame 1 $E'_{data}$
$E'_{mismatch}$
Pixels - Frame 3 $E'_{data}$
View Pair 2
Pixels - Frame 1 $E'_{data}$
$E'_{mismatch}$
Pixels - Frame $n$ $E'_{data}$
View Pair $n$-1

**Fig. 3.** Conceptual view of the energy function $E'(f)$

clip of $n$ frames. For computing the optical flow between $I_1$ and $I_n$, we do not only match $I_1$ against $I_n$, but also match $I_1$ against any intermediate view $I_k$ with $1 < k < n$. The basic idea behind this is that a pixel of the reference frame $I_1$, which is occluded when matching $I_1$ and $I_n$, might be visible (and therefore matchable) when computing the correspondences between $I_1$ and $I_k$. This concept was originally used by Xiao and Shah [12,18].

To implement this idea, we split up a sequence of $n$ images into $n-1$ view pairs. Each view pair consists of the reference frame $I_1$, on which we apply colour segmentation, and a second image $I_k \neq I_1$, i.e. we derive the view pairs $I_1 - I_2$, $I_1 - I_3$,$\cdots$, $I_1 - I_n$. From the layer extraction step, we have the dominant motion models of the view pair $I_1 - I_n$. For simplicity, we assume that within a very short image sequence the motion is linear, so that the motion models for the other view pairs can be linearly interpolated from those. To propagate the layer assignments of the individual view pairs between each other, we connect the reference frame $I_1$ of each view pair to the segment level using the term $E'_{segment}$ (Fig. 3). From its definition in equation (7), $E'_{segment}$ enforces a pixel of the reference view to have the same layer assignment as its corresponding segment, unless the pixel is occluded. Since the reference frames of all view pairs are now connected to the segment level, a pixel $p$ of $I_1$ in view pair $VP$ that is assigned to layer $l$ has to be assigned to $l$ in any other view pair $VP'$ or carry the occlusion label. This is what Xiao and Shah refer to as the *General Occlusion Constraint* [18], which is integrated into our energy function without additional effort.

## 3 Experimental Results

We have tested our algorithm on a standard test set (Fig. 4) as well as on a self-recorded one (Fig. 5). Throughout our test runs, we set $\lambda_{occ} := \lambda_{mismatch} - 1$. The effect of this is that every view inconsistent pixel is labelled as occluded on the pixel level. More precisely, if two pixels assigned to different layers project to the same pixel of the other view, one of them is view inconsistent and has to be declared as occluded. Therefore, the uniqueness constraint is enforced.

As a first test sequence, we have picked five frames from the *Mobile & Calendar* sequence (Fig. 4a). Within this sequence, there is translational motion

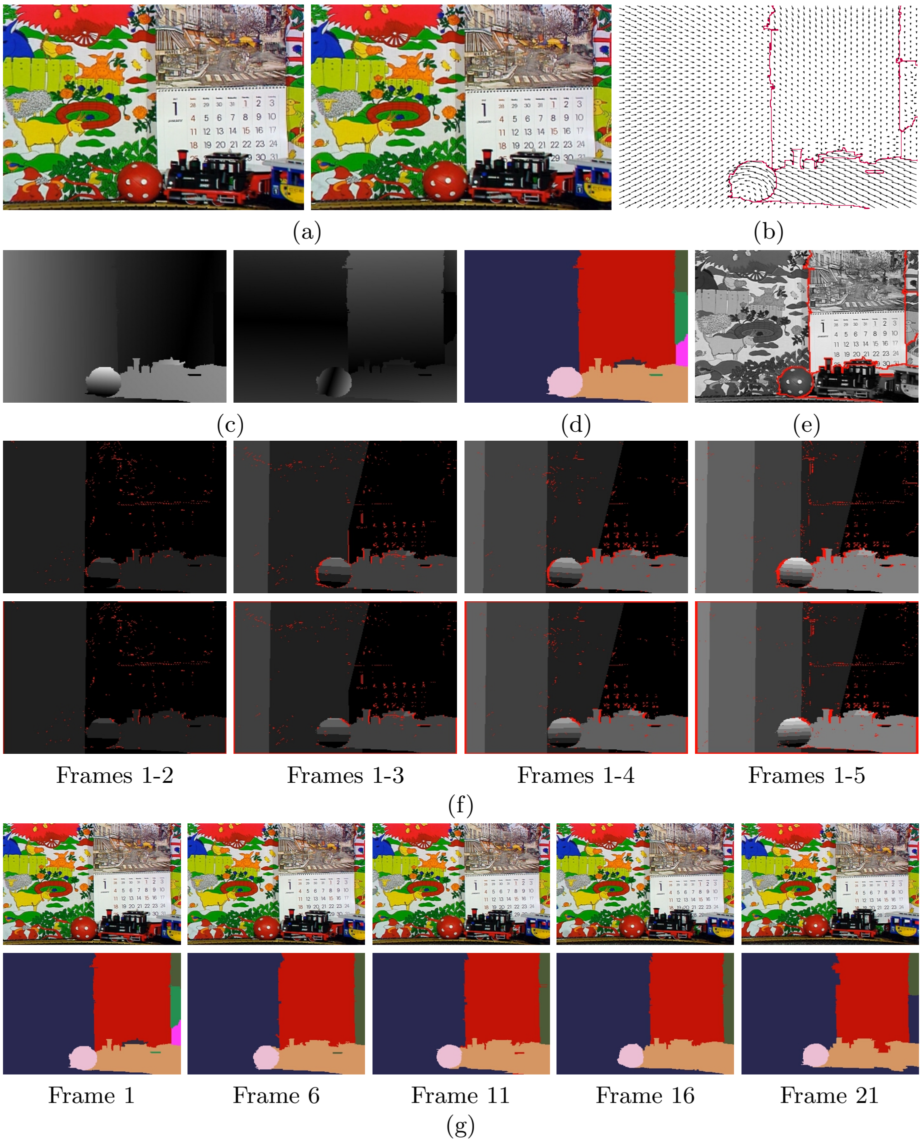

**Fig. 4.** Results for the *Mobile & Calendar* sequence. (a) Frames 1 and 5 of five input frames. (b) Flow vectors with layer boundaries outlined. (c) Absolute x- and y-components of the computed flow vectors. (d) Assignment of segments to layers. (e) Layer boundaries coloured in red superimposed on input frame 1. (f) Absolute x-components of the flow vectors on the pixel level. The top row shows the reference view (frame 1), while the match images (frames 2 – 5) are presented at the bottom. Pixels carrying the occlusion label are coloured in red. (g) Motion segmentation for each fifth frame of the complete sequence.

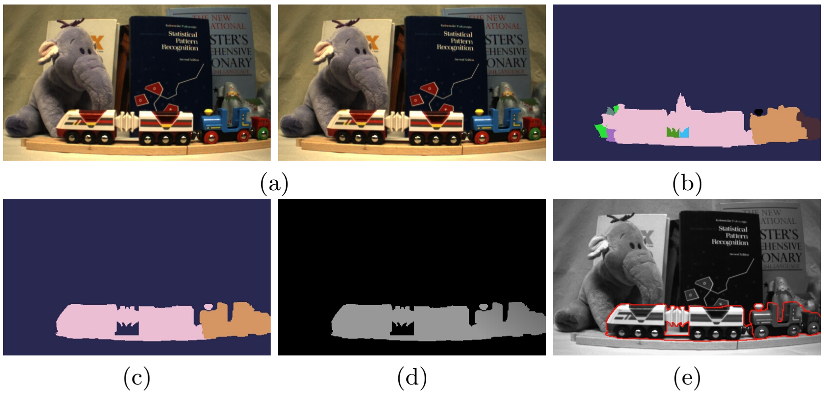

**Fig. 5.** Results for a self-recorded sequence. (a) Frames 1 and 3 of three input frames. (b) Results of the layer extraction step. (c) Assignments of segments to layers. (d) Absolute x-components. (e) Layer borders superimposed on view 1.

on the train and the poster, while rotational motion originates from the ball. Furthermore, the camera zooms out. Results computed by our method (Figs. 4b–g) indicate that the algorithm is well suited to precisely delineate motion discontinuities. Moreover, our technique can equivalently be regarded as a motion segmentation method, since the layer assignment result (Fig. 4d) divides the image into homogeneously moving regions. In Fig. 4g, we apply our algorithm to segment the complete sequence into video objects that undergo homogeneous motion. A more detailed explanation of this process is, however, found in [17].

In addition to the standard test set, we tested the proposed method on a self-recorded sequence (Fig. 5a). The sequence shows a train moving from right to left in front of a static background. Although the motion is relatively simple, the scene contains complex motion boundaries (e.g., the link connecting the wagons) and large occluded areas. These occlusions are the reason why the layer extraction step delivers poor results in the proximity of the motion discontinuities (Fig. 5b). In contrast to this, the assignment step that explicitly models occlusions seems to be able to outline the motion boundaries correctly (Fig. 5c).

## 4 Discussion

We have presented a layered segmentation-based algorithm for the estimation of dense motion correspondences. In the layer extraction step, we optimize a simple energy function on the segment level. Since the segment level alone is not sufficient to handle occlusions, we define an energy function that operates on the segment and on the pixel level in the assignment step. This energy function is extended to allow for the computation of the motion between multiple images. Our method determines correct flow information in traditionally challenging regions such as areas of low texture and close to motion discontinuities.

Further research will concentrate on overcoming two limitations of our approach. The algorithm currently describes the image motion using the affine model. This may result in an oversimplification of the real motion, especially in the presence of large motion. However, the affine model could easily be replaced by a more sophisticated one without major changes in our implementation. A more severe problem is that the segmentation assumption is not guaranteed to hold true. Our current remedy to this is to apply a strong oversegmentation. However, since this does not completely overcome this problem, our algorithm could take benefit from an operation that allows splitting segments.

## References

1. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo. IJCAI (1981) 121–130
3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: ECCV. Volume 4. (2004) 25–36
4. Zitnick, C., Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. TPAMI **22**(7) (2000) 675–684
5. Black, M., Jepson, A.: Estimating optical flow in segmented images using variable-order parametric models with local deformations. TPAMI **18**(10) (1996) 972–986
6. Tao, H., Sawhney, H., Kumar, R.: A global matching framework for stereo computation. In: ICCV. (2001) 532–539
7. Hong, L., Chen, G.: Segment-based stereo matching using graph cuts. In: CVPR. Volume 1. (2004) 74–81
8. Ke, Q., Kanade, T.: A subspace approach to layer extraction. In: CVPR. (2001) 255–262
9. Altunbasak, Y., Eren, P., Tekalp, A.: Region-based parametric motion segmentation using color information. GMIP **60**(1) (1998) 13–23
10. Ayer, S., Sawhney, H.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In: ICCV. (1995) 777–784
11. Willis, J., Agarwal, S., Belongie, S.: What went where. In: CVPR. (2003) 37–44
12. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. TPAMI **27**(10) (2005) 1644–1659
13. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low-level vision. In: ICPR. Volume 4. (2002) 150–155
14. Shi, J., Tomasi, C.: Good features to track. In: CVPR. (1994) 593–600
15. Wang, J., Adelson, E.: Representing moving images with layers. Transactions on Image Processing **3**(5) (1994) 625–638
16. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI **23**(11) (2001) 1222–1239
17. Bleyer, M.: Segmentation-based Stereo and Motion with Occlusions. PhD thesis, Vienna University of Technology (2006)
18. Xiao, J., Shah, M.: Accurate motion layer segmentation and matting. In: CVPR. Volume 2. (2005) 698–703