Region-based Optical Flow Estimation with Treatment of Occlusions[1)]

Michael Bleyer, Margrit Gelautz, Christoph Rhemann
Interactive Media Systems Group
Institute for Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria
e-mail: bleyer@ims.tuwien.ac.at, gelautz@ims.tuwien.ac.at

Abstract:
This paper describes an algorithm for computing the optical flow field between two consecutive frames. The algorithm takes advantage of image segmentation to overcome inherent problems of conventional optical flow algorithms, which are the handling of untextured regions and the estimation of correct flow vectors near motion discontinuities. Each segment's motion is described by the affine motion model. Initial motion segments are clustered to derive a set of robust layers. The assignment of segments to layers is then improved by optimization of a global cost function that measures the quality of a solution via image warping. Occlusions in both views are detected and handled in the warping procedure. Furthermore, the cost function aims at generating smooth optical flow fields. Since finding the assignment of minimum costs is $\mathcal{NP}$-complete, an efficient greedy algorithm searches a local optimum. Good quality results are achieved at moderate computational expenses.

1 Introduction

The estimation of two-dimensional displacement vectors between two images represents one of the oldest and most active research topics in computer vision. However, computation of accurate optical flow fields remains challenging for several reasons. Conventional correspondence techniques often fail to produce correct flow vectors in homogeneous coloured regions and regions of texture with only a single orientation due to the well-known aperture problem, which is especially true for local methods. Furthermore, to simplify the search, the fact that there are occlusions, i.e. pixels that are visible in only one view, is often ignored. Consequently, the performance in regions close to motion boundaries, where occlusions occur, is generally poor.

[1)] This work was supported by the Austrian Science Fund (FWF) under project P15663.

In this work, we propose a technique that tries to overcome those problems by the use of image segmentation. Based on the assumption that motion discontinuities go along with discontinuities in the intensity image, we take benefit from the segmentation information in three ways. First, the motion values inside each segment are constrained to follow the same motion model, which allows the assignment of smooth flow values in regions of poor texture. Second, we believe that motion boundaries can be accurately identified by the use of monocular cues, such as the partition of the reference image into regions of homogeneous colour. Third, occluded regions can be assigned to meaningful flow values that are propagated using the segmentation information.

For a review and comparison of optical flow methods we refer the reader to [1, 7] and concentrate on the works that we see as closest related to our approach. The stereo algorithm described in [9] models the disparity of segments by a planar equation and propagates disparity across neighbouring segments in a hypothesis testing framework. The stereo method presented in [3], which builds the basis for the proposed technique, clusters disparity segments to form a set of layers. Assignments of segments to layers are then improved by optimization of a cost function. The motion algorithm described in [2] is similar to our approach in the sense that motion estimation and motion segmentation are performed jointly. Finally, in [6] the mean-shift algorithm is used for the extraction of motion layers.

The organization of the remainder of this paper is as follows. The proposed optical flow algorithm is presented in section 2. We start with an overview of the algorithmic framework and then focus on a more detailed description of the individual steps in the corresponding subsections 2.1, 2.2 and 2.3. Section 3 shows and discusses experimental results that were achieved using the proposed method. Finally, we give our conclusions in section 4.

2 Algorithm

The overall algorithm consists of several modules that are illustrated in figure 1. In a first step, colour segmentation is applied to the reference image. The affine model of each segment is then initialized from a set of initial correspondences. Motion segments are clustered in the layer extraction step of the algorithm to derive a set of layers that represent the dominant image motion. The affine model of each layer is refined based on its spatial extent. In the layer assignment step, a global cost function is optimized in order to improve the assignment of segments to layers. The algorithm then iterates the layer extraction and assignment steps until the costs could not be improved for a fixed number of iterations and returns the solution of lowest costs.

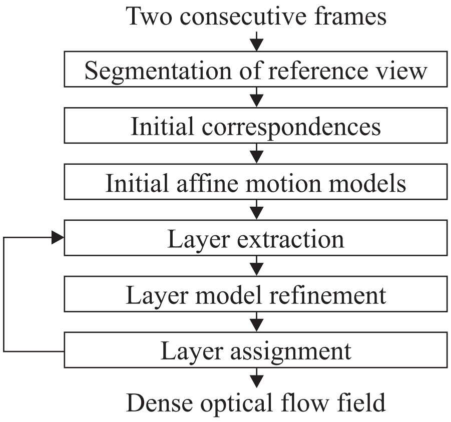


Figure 1: Algorithmic outline.

2.1 Colour Segmentation and affine Motion Model

The proposed method applies colour segmentation to the reference image. We thereby embed two basic assumptions. It is assumed that all pixels inside a region of homogeneous colour follow the same motion model and motion discontinuities coincide with the boundaries of those regions. To ensure that our assumptions are met, we apply a strong oversegmentation as shown in figure 2. In our current implementation, we use an off-the-shelf segmentation algorithm described in [4].

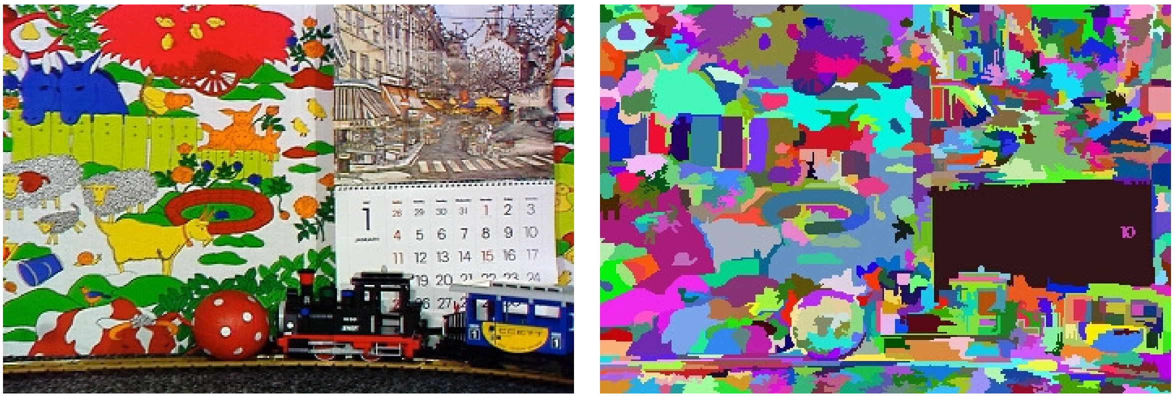
The optical flow inside each segment is modelled by affine motion, which is

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

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. We compute a set of correspondences using the KLT feature tracker [8] and derive each segment's affine parameters by least squared error fitting to all correspondences found inside this segment. A robust version of the method of least squared errors is employed to reduce the sensitivity to outliers.

2.2 Layer Extraction

Unfortunately, the segments' motion models are not robust, which is due to the small spatial extent over which their affine parameters were estimated. To overcome this problem, we identify groups of segments that can be well described by the same affine motion model. Each segment is therefore projected into an eight-dimensional feature space, which consists of the six parameters of the affine motion model and two parameters for the coordinates of the segment's center of gravity. A modified version of the mean-shift algorithm [5] is then employed to extract clusters in this feature space. Segments of the same cluster are combined



(a) (b)

Figure 2: Colour segmentation. (a) Reference image. (b) Segmented image. Pixels of the same colour belong to the same segment.

to form a *layer*. The affine motion parameters of a layer are computed by fitting the model over the larger spatial extent, which is built by all segments belonging to this layer. Each segment is then assigned to the motion model of its corresponding layer.

2.3 Layer Assignment

We try to improve the assignment of segments to layers by optimizing a global cost function. The quality of a solution is thereby measured by image warping. The basic idea behind this procedure is that if the reference image is warped to the second view according to the correct flow field, the resulting warped image should be very similar to the real second view. Speaking more technically, the pixel dissimilarity between visible pixels of the warped and the real second view should be low. Detection of occlusions and reasoning about visibility has to be performed in the warping process. We illustrate this in figure 3. Let us assume that a pixel of the warped view gets contribution from more than one pixel of the reference view, which is the case for vertical hatched areas of figure 3b. Since we assume surfaces to be opaque, only one of those pixels can be visible. Consequently, the other pixels are occluded in the second view. Unfortunately, for a motion algorithm the reasoning about the pixels' visibility is not obvious. We decided to declare the pixel of lowest pixel dissimilarity as being visible, while the other pixels are marked as being occluded. However, it is interesting to note that a stereo algorithm can naturally do this decision by declaring the pixel of highest disparity as being visible [3], since this is the pixel closest to the camera. There are also pixels in the warped image that do not receive contribution from any pixel, which occurs at the horizontal hatched area of figure 3b. This case corresponds to an occlusion in the reference view. Our cost function has to penalize occlusions, since otherwise declaring all pixels as being occluded would form a trivial optimum. The last term of our cost function aims at generating smooth optical flow fields. We therefore penalize neighbouring pixels of the reference image that are

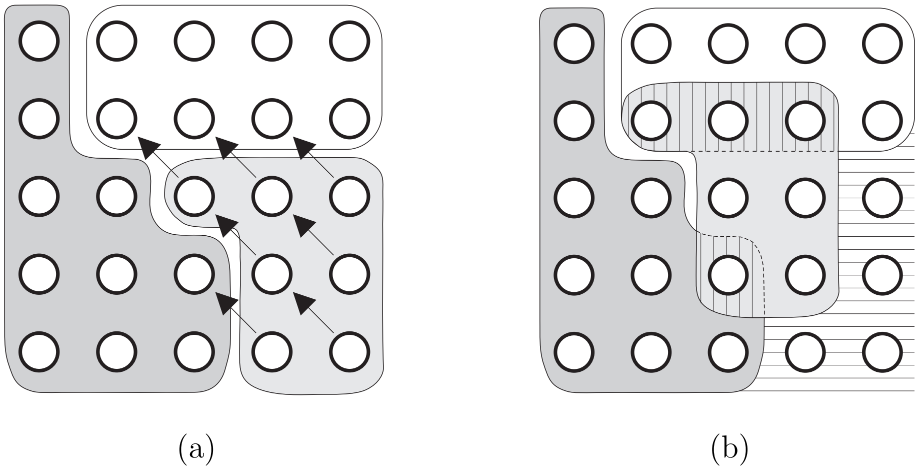


Figure 3: The warping operation. (a) Reference view. The image is divided into three segments. The estimated motion for two segments is zero, while the third segment undergoes a translational motion as indicated by the arrows. (b) Warped view. The reference image is warped according to the estimated motion field. Hatched areas represent regions that are affected by occlusion.

assigned to different layers. Putting this together, we formulate the cost function

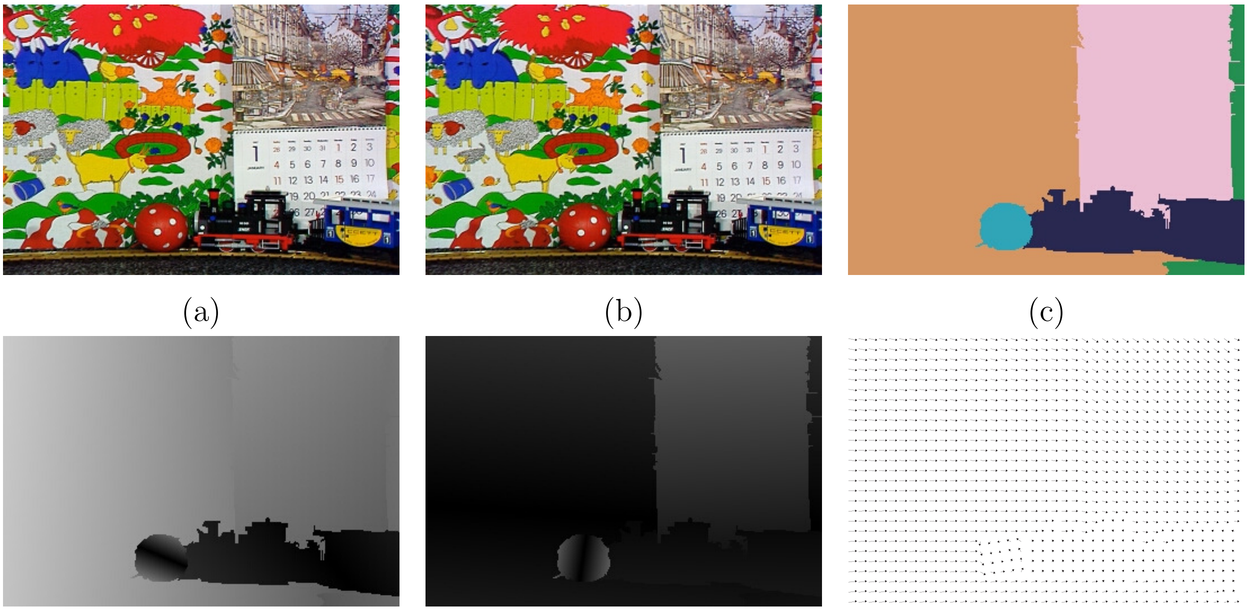
$$C = \sum_{p \in Vis} d(W(p), R(p)) + N_{occ}\lambda_{occ} + N_{disc}\lambda_{disc} \quad (2)$$

with Vis being the set of visible pixels, $d(W(p), R(p))$ being the dissimilarity function of the pixel p in the warped image $W(p)$ and in the real second view $R(p)$, which is implemented as the summed up absolute differences of RGB values, N_{occ} and N_{disc} being the number of detected occlusions and discontinuities and λ_{occ} and λ_{disc} are constant penalties for occlusion and discontinuity, respectively.

Unfortunately, finding the assignment of lowest costs is $\mathcal{NP}$-complete. A greedy algorithm is therefore employed to find a local optimum. For each segment we check whether changing its layer assignment to the assignment of a neighbouring segment reduces the costs. If this is the case, we record the corresponding layer and update assignments after all segments are checked. This procedure is iterated until the costs could not be improved for a certain number of iterations. An incremental warping scheme thereby significantly reduces the computational costs.

3 Experimental Results

We demonstrate the performance of the proposed algorithm using the frames 50 and 54 of the Mobile & Calendar sequence that are shown in figure 4a and 4b. In this sequence, the camera pans to the left, while there are moving objects (calendar, train and ball) in the scene. Since no ground truth is available, we have to focus on a qualitative discussion of the results. Figure 4c presents the final layer assignment. Although motion segmentation is not the primary goal



(a) (b) (c)

(d) (e) (f)

Figure 4: Results for the Mobile & Calendar sequence. (a) Frame 50. (b) Frame 54. (c) Final layer assignments. (d) Absolute x-components. (e) Absolute y-components. (f) Flow vectors.

of this work, the computed layers seem to correspond well to scene objects. To visualize the flow field, we plot the absolute x- and y-components of the flow vectors scaled by a factor of 32 in figures 4d and 4e. Motion boundaries appear to be correctly captured, while also the image motion in untextured regions seems to be accurately identified (e.g. lower part of calendar). Finally, we show the two-dimensional flow vectors for some pixels of the reference frame in figure 4f. For the 352 × 240 pixel input images our current C++ implementation needed 47 seconds on an Intel Pentium 4 2.0 GHz computer to generate the results.

As a second test pair we used the frames 11 and 14 of the Tennis sequence that are shown in figure 5a and 5b. There are two moving objects in the scene, which are the arm and the ball. While the arm undergoes a relatively small motion, there is large motion on the ball. Since the x-components of the flow vectors are almost zero, we decided to show the warped view in figure 5c instead. This image is generated by warping the reference view according to the computed flow vectors and should be compared against the real second view presented in figure 5b. Regions that were identified as being occluded in the reference view are coloured black. We then present the final layer assignment in figure 5d. The arm is thereby represented by five different layers, which is most likely for the reason that only a single affine motion model can hardly capture the real motion of the arm. The y-components of the flow vectors scaled by a factor of 16 are then shown in figure 5e. The motion boundaries seem to be correctly identified and also the large motion of the ball seems to be captured. Finally, we present the corresponding flow vectors in figure 5f. 99 seconds were needed to generate the results for the 352 × 240 pixel images.

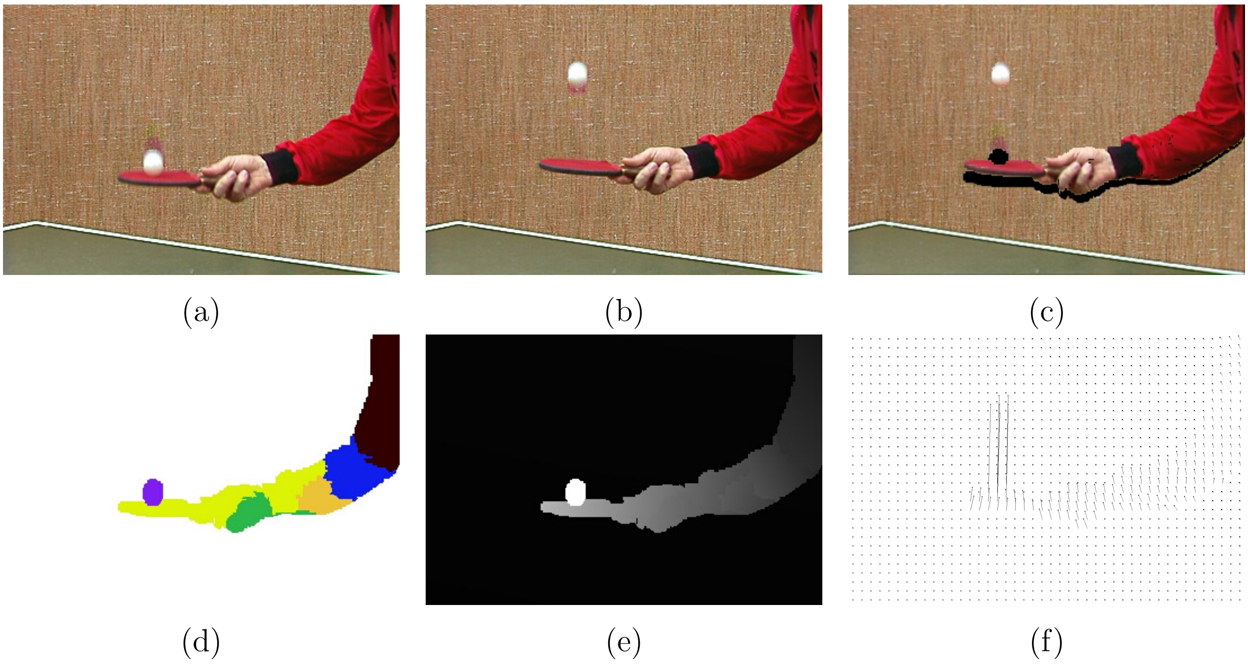


Figure 5: Results for the Tennis sequence. (a) Frame 11. (b) Frame 14. (c) Warped image. (d) Final layer assignments. (e) Absolute y-components. (f) Flow vectors.

4 Conclusions

We have presented an optical flow algorithm that uses image segmentation to improve the quality of flow estimates in untextured regions and to allow a precise extraction of motion boundaries. The proposed method uses a layered representation and employs the affine motion model to describe image motion. The assignment of segments to layers is refined by an efficient greedy algorithm that optimizes a global cost function. Experimental results demonstrate the good performance of the algorithm, especially in regions of poor texture as well as in regions close to motion boundaries. Further work will concentrate on applying a more global optimization method to the layer assignment problem (e.g. graph cuts) and using a more sophisticated motion model. The robustness of the algorithm could also be improved by taking more than two frames into account.

References

[1] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, pages 489–495, 1999.

[3] M. Bleyer and M. Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. In *International Conference on Image Processing*, pages 2997–3000, 2004.

[4] C. Christoudias, B. Georgescu, and P. Meer. Synergism in low-level vision. In *International Conference on Pattern Recognition*, volume 4, pages 150–155, 2002.

[5] D. Comaniciu and P. Meer. Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 1(2):22–30, 1999.

[6] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *Conference on Computer Vision and Pattern Recognition*, pages 255–262, 2001.

[7] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):216–143, 2001.

[8] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[9] H. Tao and H. Sawhney. Global matching criterion and color segmentation based stereo. In *Workshop on Applications of Computer Vision*, pages 246–253, 2000.