Colour Segmentation-based Computation of Dense Optical Flow with Application to Video Object Segmentation

Michael Bleyer, Margrit Gelautz,
Christoph Rhemann

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

Abstract

We propose a method for the computation of dense optical flow fields between two images of a video sequence. In a preprocessing step, the algorithm segments the reference frame into regions of homogenous colour. The flow vectors inside such regions are supposed to vary smoothly, while motion boundaries are assumed to coincide with boundaries of those regions. Given an initial affine motion model for each segment, the algorithm extracts a set of affine layers, which represent the dominant image motion, in a clustering procedure. Each segment is then assigned to one of those layers in order to optimize a global cost function. The cost function aims at generating smooth flow fields and models occlusions in both images. Layer extraction and assignment are then iteratively applied until convergence. Since the algorithm assigns segments that undergo the same affine motion to the same layer, the proposed method can equivalently be regarded as motion segmentation algorithm. Strong experimental results are achieved, especially in regions of poor texture and close to motion boundaries, where conventional methods often show poor performance.

1. Introduction

The estimation of two-dimensional displacement vectors between two images of a video sequence plays a key role in a number of computer vision problems, including motion detection and segmentation, scene reconstruction, robot navigation, video shot detection, mosaicking and compression. Although being one of the oldest and still most active research topics in the vision community, 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. In this work, we propose a technique that tries to overcome those problems by the use of colour segmentation. The algorithm takes advantage of simultaneous computation of motion segmentation and motion estimates. Therefore, the method can potentially also be used for the task of extracting video objects. Robust video object extraction is of specific importance in the context of new video coding schemes (e.g. MPEG4).

For a review and comparison of optical flow methods we refer the reader to the works of Barron et al. [1] and McCane et al. [7]. We concentrate on the works that we see as closest related to our approach. The stereo algorithm described by Tao and Sawhney [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 by Bleyer and Gelautz [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 by Birchfield and Tomasi [2] is similar to our approach in the sense that motion estimation and motion segmentation are performed jointly. Finally, Ke and Kanade [6] apply the mean-shift algorithm for the extraction of motion layers.

2. Algorithm

The proposed algorithm starts by segmenting the reference image into regions of homogenous colour. Initial pixel correspondences are then computed and used to initialize each segment's motion model. In the layer extraction step of the algorithm, the motion segments are clustered to derive a set of robust layers. The motion model of a layer is computed using the spatial extent built by all segments belonging to this layer. Since the spatial assignment of each layer is known and serves to determine (or refine) the corresponding motion model, this can be interpreted as the motion estimation step. Knowing the layers' motion models, the spatial extent that is occupied by each layer is then optimized in the layer assignment step. This can be regarded as the motion segmentation component. The layer extraction and layer assignment steps are iterated until convergence.

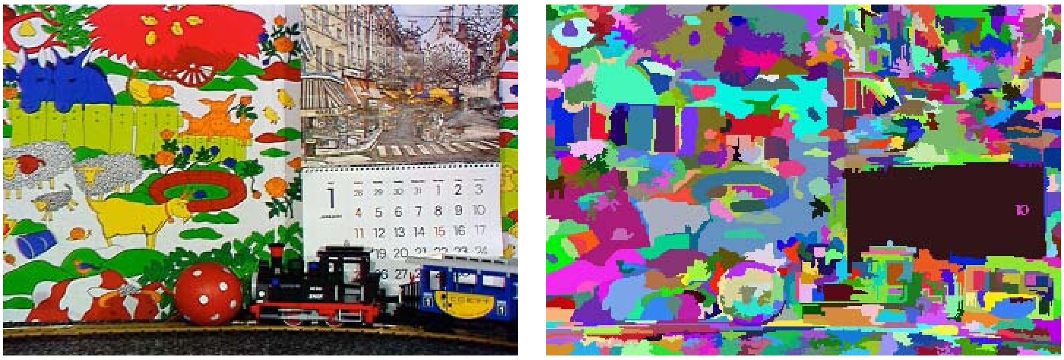
2.1. Colour Segmentation and Affine Motion Model

The proposed method applies colour segmentation to the reference image. We thereby embed two basic assumptions, which are based on the observation that motion discontinuities usually go along with discontinuities in the intensity image for most videos of natural scenes. 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 1. In our current implementation, we use an off-the-shelf segmentation algorithm described by Christoudias et al. [4].

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

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

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.



(a) (b)

Fig. 1. Colour segmentation. (a) Reference image. (b) Segmented image. Pixels of the same colour belong to the same segment.

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 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. In other words, the pixel dissimilarity between visible pixels of the warped and the real second view should be low. Reasoning about visibility has to be performed in the warping process. Let us assume that a pixel of the warped view gets contribution from more than one pixel of the reference view. A stereo algorithm can then declare the pixel of highest disparity as being visible [3], since this is the pixel closest to the camera. However, a similar reasoning is not obvious in the case of motion. We therefore decided to declare the pixel of lowest pixel dissimilarity as being visible, while the other pixels are marked as being occluded. Furthermore, there are pixels in the warped image that do not receive contribution from any pixel. 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 assigned to different layers. Putting this together, we formulate the cost function

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

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 np-complete. A greedy algorithm is therefore employed to find a local optimum. The basic idea behind the optimization algorithm is to propagate motion models across neighbouring segments. 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 the assignments after all segments are checked. 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 well-known MPEG test sequence Mobile & Calendar that are shown in figure 2a and 2b. 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 2c presents the final layer assignment. 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 2d and 2e. 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 2f. For the 352x240 pixel input images our current implementation needed 47 seconds on an Intel Pentium 4 2.0 GHz computer to generate the results.

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.

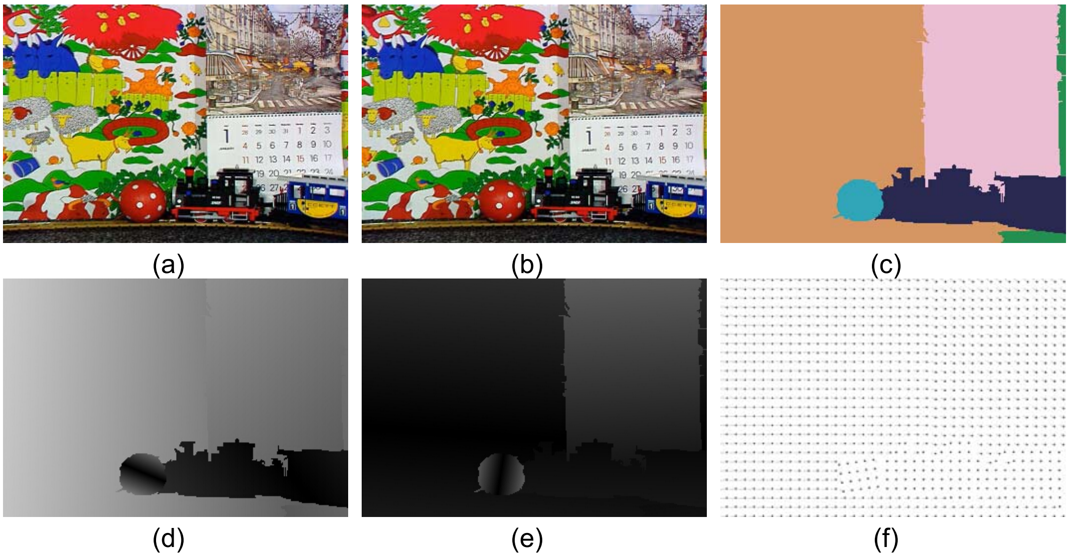


Fig. 2. Results for the Mobile & Calendar sequence. (a) Frame 50. (b) Frame 54. (c) Final layer assignments. (d) Absolute x-components. (e) Absolute y-components. (f) Flow vectors.

Acknowledgement

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

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 *IEEE 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 *International 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 *International 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.