VIDEO FRAME SEGMENTATION USING COMPETITIVE CONTOURS

Piotr Steć, Marek Domański

Institute of Control and Computation Engineering, University of Zielona Góra
ul. Podgórna 50, Zielona Góra, Poland
phone: + (48) 68 3282337, email: P.Stec@issi.uz.zgora.pl
web: http://www.issi.uz.zgora.pl/

Institute of Electronics and Telecommunications, Poznań University of Technology
Piotrowo 3A, 60-965, Poznań, Poland
phone: + (48) 61 6652762, email: domanski@et.put.poznan.pl
web: http://www.multimedia.edu.pl/

ABSTRACT

The paper presents a technique for the segmentation of colour images from video sequences. The technique is aimed at the extraction of multiple objects in the presence of background motion, and without global motion compensation. A modified version of the Multi-label Fast Marching algorithm is used in the segmentation process. The most important modification allows segment merging as well as pushing back borders of other segments. Thanks to this, the limitation of a one-way propagation for fast marching is removed. The experiments described in this paper are restricted to video sequences with translational motion of rigid objects only. The side effect of the algorithm is the regularized motion field.

1. INTRODUCTION

In their previous work [1], the authors have proposed an efficient technique to segment frames form a video sequence using the Fast Marching algorithm. The proposed technique provides extraction of a single moving object from background with different motion. The efficiency of the this technique is related with joint exploitation of color and motion information. Nevertheless, in many applications, it is desirable to extract several objects that differ in motion form the background and each from the other. An original method that would be presented in this paper is aimed at performing such a task. As the main tool, the multi-label fast marching method will be used. The idea of a simultaneous propagation of multiple contours using the fast marching algorithm was introduced by Sifakis and Tziritas [2, 3]. However, the method presented here shares with that approach merely the idea of multi-label fast marching. In the original method only two labels are used, and each of them has individual propagation speed. Such an approach is hard to extend to multiple object segmentation, especially when the number of objects is unknown. The approach presented here assues the same propagation speed for all labels, thus several labels flow may be easily accumulated. Moreover, both static and moving background may be treated by the algorithm proposed. An additional advantage of this approach is that it is easy to define the stop condition since contours are propagating toward each other. In the original method, the algorithm stops when the contours meet. In the method presented in this section some additional actions can be performed when two segments meet.

2. CONTOUR INITIALIZATION

In the technique proposed, the initialization procedure is based on the computation of the displaced frame difference (*dfd*) between two consecutive frames and requires the dense motion field to be computed prior to the initialization. Here it is assumed that the dense motion field was computed using one of the already known methods [4, 5].

It is assumed that regions with a zero-valued *dfd* are likely to be inside objects with the same motion properties. Therefore, such areas are good starting points for contour propagation. A similar procedure was successfully applied in the method presented in the papers [6, 7]. The displaced frame difference at the point (x, y) is computed as follows:

$$dfd = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1}|I_n(x+i, y+j) - I_{n-1}(mx+i, my+j)|, \tag{1}$$

where I_n is the sample value in the n-th frame, and mx and my denote the position of the point in the previous frame according to motion information. Such a computation procedure has two purposes: reducing image noise influence and rejecting single points as starting region candidates. All connected pixels with the dfd value equal to zero are labelled as one region with an additional constraint on motion uniformity. This prevents the regions on the border between the object and the background with a zero dfd but with different motion properties from merging into one region. An individual label is assigned to each region. Such regions will be seeds for contours propagated using the fast marching algorithm (Fig. 1). The number of seed regions is always larger than the number of final segments.

3. INITIAL SEGMENTS PROPAGATION

All initialized segments are propagated outwards using the modified fast marching algorithm. In this algorithm, segment labels for points visited by contours are positive integers. Trial points (boundary points sorted by contour arrival times, see [8]) for each contour are marked with negative numbers of segment labels. All trial points from all segments are included into the same sorted list. Thanks to this, no additional synchronization between the propagation of the segments is required. This situation is naturally handled by the fast marching algorithm since it can propagate contours of any topology. At this stage of propagation, there is in fact

Figure 1: Seed regions overlayed on a frame from the test sequence *Mobile and calendar* (indicated by arrows)

Figure 2: Segments during the initial stage of propagation

no difference between the standard and the multi-label implementation apart from the fact that the new label for the trial point is inherited from the segment that propagates at the current algorithm step (Fig. 2).

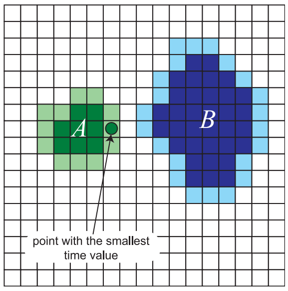
Propagation speed is based only on the current image properties. The propagation speed F is calculated from smoothed color components:

$$F = \frac{1}{\max(\nabla Y_\sigma, \nabla Cb_\sigma, \nabla Cr_\sigma) + 1}, \qquad (2)$$

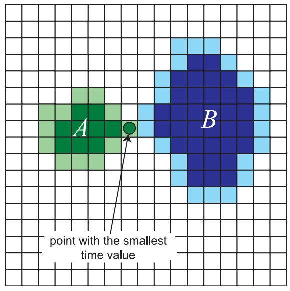
where σ denotes Gaussian blurring. Such a speed definition makes contour motion fast in smooth areas and slow as they approach edges. Thanks to this, contours are likely to meet on the object edge rather than inside the object.

4. DYNAMIC REGULARIZATION OF THE MOTION FIELD

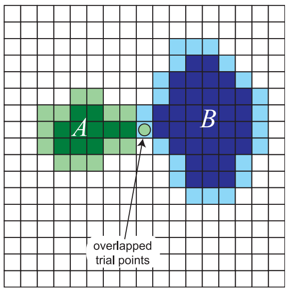
Here, in the proposed method, images are segmented using both color and motion information. Therefore, all motion vector inconsistencies may lead to erroneous segmentation. Therefore, some motion vector regularization may improve segmentation reliability. In the experiments a simple regularization technique has been used that is appropriate for sequences with translational motion of rigid objects. This regularization technique consists in propagating the correct motion vectors from the initialization regions along with the contour and replacing the original motion.



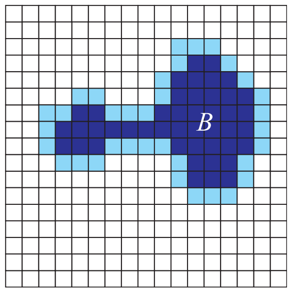
(a) Segments are one point away



(b) Segments are touching each other



(c) Trial point overlaps the existing one. Motion similarity is checked



(d) Segments merged together

Figure 3: Procedure of merging segments with high motion similarity. Trial points are marked in a brighter colour

Nevertheless, the application of a more sophisticated motion regularization method and using full motion information will extend the reliability of this method. This is possible since the only requirement of this segmentation method is motion consistency within the propagating segment.

5. SEGMENT MERGING AND PUSHING

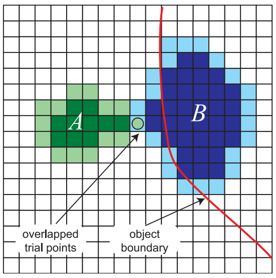
The expansion of the segment described in Section 3 is performed as long as new trial points can be set on the area not visited by any of the propagating curves. When a new trial point is going to be set in a place occupied by a trial point from another segment, two actions can be performed: the segments can be merged or one contour can be pushed back by another.

5.1 Segments merging

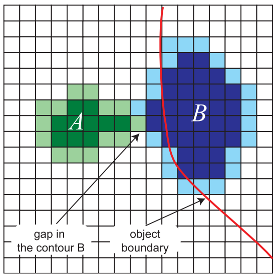
When two segments meet, the motion of these segments is compared. The meeting point is a trial point from one segment that must be placed over a trial point from another segment (Fig. 3(c)). Since motion within segments is the same for all points, it is sufficient to take one point from each segment for comparison. Motion from the segment A is compared with motion from the segment B according to the following expression:

$$|mx_A - mx_B| < \varepsilon \wedge |my_A - my_B| < \varepsilon, \qquad (3)$$

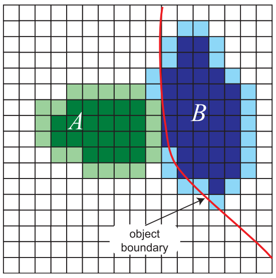
where mx and my are motion vector components and ε is an empirically chosen merging threshold. During the tests that were performed on a number of sequences, the best results gave $\varepsilon = 0.9$. This means that segments with motions different by less than one pixel per frame are connected. Motion



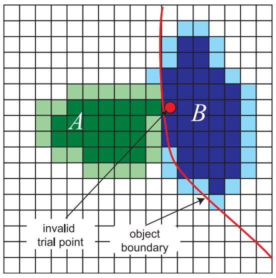
(a) dfd is computed for both overlapped trial points



(b) Trial point B replaced with the trial point A



(c) Segment A approaches the object boundary



(d) New trial point cannot be placed. dfd from the segment A is higher than that from the segment B

Figure 4: Segment A with lower dfd pushes the segment B with higher dfd back to the object boundary

vectors are estimated with sub-pixel accuracy. Additional research is needed to find a way of automatically adjusting ε. When the expression 3 is true, the segments A and B are merged.

To ensure maximum efficiency, labels from the smaller segment are changed to the value of those from the larger segment. Also, trial points from smaller segments are assigned the value from larger segments (Fig. 3(d)). Motion vectors from smaller regions are replaced with motion vectors from larger regions to ensure motion uniformity within segments.

5.2 Segment pushing

If two segments that meet are not classified to be merged, the propagating segment can push back another segment under certain circumstances.

When a trial point from the propagating segment A is going to be placed at the position (x,y) occupied by a trial point from another segment B (Fig. 4(a)) and motion similarity is not high enough, then the displaced frame difference is computed for both segments:

$$dfd_A = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} |I_n(x+1,y+j) - I_{n-1}(mx_A+i,my_A+j)|, \tag{4}$$

$$dfd_B = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1} |I_n(x+1,y+j) - I_{n-1}(mx_B+i,my_B+j)|, \tag{5}$$

where I_n is the n-th image from the sequence, mx, my are motion compensated positions of the pixels, and the indexes A and B denote the segment being the source of motion information. If $dfd_A < dfd_B$, then the trial point from the segment A replaces the trial point from the segment B. In the case when $dfd_A > dfd_B$, the trial point from the segment A is not placed and no further propagation is performed. The latter case means that the meeting point belongs to the segment B. At that point only the segment B has the possibility of propagating further, because the trial point from the segment A was not set. If the point considered lies on the object border, the segment B cannot propagate either because the trial point from B will have a higher dfd than the point from A set earlier. The segment B can propagate further if the segment A passed its object border and the meeting took place on the object that belongs to B. The segment B will push back the segment A to the nearest object border.

The replacement of the point from the trial list of the segment B creates a gap on the segment boundary (Fig. 4(b)). Nonetheless, it has no influence on the further propagation of neither the segment B nor the segment A. The replaced point has no chance of propagating anyway because its dfd was higher than that of the segment A. The remaining portion of the segment B is propagated normally. The fast marching algorithm does not require a closed contour for propagation.

The segment A stops pushing back the segment B on the boundary of the object which has motion properties similar to those represented by the segment B. In such a case, the segment A cannot propagate further, because its dfd for the trial point that is going to be set inside the object occupied by the segment B will be higher than that for the segment B (Fig. 4(d)).

When a contour has no possibility of propagating further, no new trial points are set. This implies the reduction of the total length of the sorted list used by the fast marching algorithm and the same performance improvement.

6. STOP CONDITION

The presented algorithm stops propagation when all image points are assigned to segments and there is no segment that could push back another segment. The algorithm cannot run infinitely because oscillations between segments are impossible. No segment can visit twice the same area. Namely, when a segment was pushed back by another segment, it cannot get the lost pixels back.

7. EXPERIMENTS

The proposed method of segmentation using multi-label fast marching was evaluated experimentally. The algorithm was able to segment complex scenes with multiple overlapping objects and with objects partially visible in the scene. An example of such a scene is presented in Fig. 5.

The algorithm requires a *partially reliable* motion field for correct performance. This means that the motion estimation algorithm must be able to produce at least some parts of the motion field, with motion vectors that point precisely onto the corresponding pixels from the previous frame (the dfd for these points is zero). When motion vectors are mostly erroneous, the consequence is a wrongly segmented image. For the testing purposes, only simple classical motion estimation algorithms were implemented. The implementation of a faster and more precise motion estimation method will improve the performance of the segmentation algorithm. Despite the simple definition of the propagation speed, the

(a) Original image (b) Segmented image

Figure 5: Frame 112 from the 'Bus' sequence segmented using multi-label fast marching

segmentation of one frame takes about 3 to 4 seconds using a PC with AthlonXP 1400 MHz processor.

The current speed definition allows calculating speed for the whole frame before propagation begins using fast convolution filters. During the propagation, speed is only read from the table. However, the total length of the propagated contours is quite big and the biggest impact on the performance comes from the implementation of the sorting algorithm used by the FMM. The performance of the algorithm can be improved by the parallelization of the propagation process. Because timing between contours is not important, the propagation of the segments can be divided between an arbitrary number of threads. This is possible because multiple contours can be propagated in a single thread like in the implementation presented here. Another way is to use parallel implementation of the FMM like the one proposed by Dejnozkova and Dokladal in [9].

8. CONCLUSION

The algorithm presented in this paper allows for fast and fully automatic (unsupervised) segmentation of colour video sequences in the presence of a moving background without the necessity for global motion compensation. The algorithm is designed to segment individual frames without object tracking. The motivation for such an approach is the fact that there is a large number of object tracking algorithms [10, 11, 12, 13, 14] that require manual initialization of the object boundary. However, there exists the problem of automatic search of objects at the beginning of the sequence.

The presented algorithm was designed to segment video frames into multiple disjoint objects. Segmentation is proposed for natural sequences, i.e., sequences that represent the natural world as perceived through a camera, and not created by computer graphic tools.

Here, the main concern was algorithm speed and stability rather than segmentation quality. The algorithm is suitable for real-time processing of video with the use of fast processors. The current version of the algorithm cannot deal with complex motion and sometimes may produce oversegmented frames. Nevertheless, the authors have found that further extensions and improvements are possible.

REFERENCES

[1] P. Steć and M. Domański, "Efficient unassisted video segmentation using enhanced fast marching," in *IEEE International Conference on Image Processing – ICIP 2003*, (Barcelona, Spain), pp. 246–253, September 14-17 2003.

[2] E. Sifakis and G. Tziritas, "Moving object localisation using a multi-label fast marching algorithm," *Signal Processing: Image Communication*, vol. 16, no. 10, pp. 963–976, 2001.

[3] E. Sifakis and G. Tziritas, "Video segmentation using fast marching and region growing algorithms," *EURASIP Journal on Applied Signal Processing*, pp. 379–388, April 2002.

[4] B. K. Horn and B. G. Rhunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[5] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of DARPA Image Understanding*, pp. 121–130, 1981.

[6] P. Steć and M. Domański, "Two-step unassisted video segmentation using fast marching method," in *10th International Conference – CAIP 2003*, Lecture Notes in Computer Science, (Groningen, Holland), pp. 246–253, Springer-Verlag, 2003.

[7] P. Steć and M. Domański, "Fast two-step unassisted video segmentation technique evaluated by tolerant ground truth," in *5th International Workshop on Image Analysis for Multimedia Interactive Services*, (Lisboa, Portugal), April 21-23 2004.

[8] J. Sethian, "Fast Marching Methods and Level Set Methods for propagating interfaces," in *29th Computational Fluid Dynamics*, vol. 1 of *VKI Lectures series*, von Karman Institute, 1998.

[9] E. Dejnozkova and P. Dokladal, "A parallel algorithm for solving the eikonal equation," in *International Conference on Image Processing*, (Barcelona, Spain), September 14-17 2003.

[10] M. Irani, B. Rousso, and S. Peleg, "Detecting and tracking multiple moving objects using temporal integration," in *European Conference on Computer Vision*, vol. 588 of *LNCS*, pp. 282–287, Springer-Verlag, 1992.

[11] M. Dubuission Jolly, S. Lakshmanan, and A. Jain, "Vehicle segmentation and classification using deformable templates," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 18, pp. 293–308, March 1996.

[12] A.-R. Mansouri and J. Konrad, "Motion segmentation with level sets," in *IEEE International Conference on Image Processing*, (Kobe, Japan), pp. 126–130, October 1999.

[13] J. Guo, J. Kim, and C. Kuo, "An interactive object segmentation system for mpeg video," in *International Conference on Image Processing*, (Kobe, Japan), pp. 140–144, 1999.

[14] G. Iannizzotto and L. Vita, "Real-time object tracking with movels and affine transformations," in *IEEE International Conference on Image Processing*, vol. I, (Vancouver, Canada), pp. 316–322, 2000.