

# Motion-based segmentation of image sequences using orientation tensors

Gunnar Farneback

Computer Vision Laboratory  
Linköping University  
S-581 83 Linköping, Sweden  
E-mail: gff@isy.liu.se

## Abstract

*This paper addresses the problem of motion-based segmentation of image sequences. One motion estimation algorithm and two segmentation algorithms are presented. The motion estimation is based on 3D orientation tensors and the algorithm can be used to estimate a large class of motion models, including the affine model that is used in the segmentation. The segmentation algorithms are based on a competitive region growing approach.*

## 1. Introduction

Segmentation is an important step in many image processing applications. Obvious examples can be found in object recognition and second generation video coding techniques [5], among others. Depending on the application, different criteria can be chosen as basis for the segmentation, such as texture or motion. In the latter case one is interested in finding regions characterized by having a coherent motion, with respect to some motion model. A difficulty with this approach is the fact that precise estimation of the motion in the different regions requires a good segmentation, while on the other hand, a good segmentation cannot be obtained without accurate motion estimates. Hence the two subproblems of segmentation and motion estimation are strongly interlinked.

In [6, 7, 8], which served as an initial inspiration for this work, Wang and Adelson use motion-based segmentation to obtain a layered representation of image sequences. They start from an arbitrary partition of the first frame and reach the final segmentation iteratively. Each iteration consists of estimation of affine motion model parameters, clustering of these models in parameter space, and resegmentation with respect to the new affine motion models. Later frames are instantiated with the previous segmentation instead of an arbitrary partition. The motion model estimation is based on an optic flow field, computed by a gradient based algorithm.

Here the same motion model is used, but about everything else is done differently. The motion estimation is based on 3D orientation tensors [3]. The segmentation uses

a unsupervised region growing approach, based on competition between seeds to become regions and between regions to grow. For subsequent frames a simpler algorithm has also been developed. It can be noted that while the motion estimation works well with the segmentation algorithms, they are largely independent. The motion estimation can be used in other applications and the segmentation algorithms can be adapted to other homogeneity criteria.

Since this paper is a condensation of a Master's thesis, many details have been omitted. These details, as well as some additional material, can be found in [2].

## 2. Motion model estimation

The velocity estimation is based on 3D orientation tensors, as described in [3]. An orientation tensor  $\mathbf{T}$  can be considered as a quadratic form, describing the directional distribution of the signal energy in a neighborhood of a point. By the spectral theorem the tensor can be decomposed as

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T, \quad (1)$$

where  $\lambda_i$  are the eigenvalues,  $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ , and  $\{\hat{\mathbf{e}}_i\}$  is the corresponding set of orthogonal eigenvectors. From the eigenvalues it is possible to determine whether the signal is locally isotropic, two-dimensional or one-dimensional. In the case of an image sequence the two latter cases correspond to the moving point and moving line cases, respectively. Estimates of the velocity or the normal velocity component can be obtained immediately from the eigenvectors. In the context of estimating a motion model it would, however, be a mistake to give up the additional information contained in the tensor at this early stage.

A 2D velocity  $(v_x, v_y)$ , measured in pixels/frame, can be extended to a 3D directional vector  $\mathbf{v}$  by setting

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ 1 \end{pmatrix}. \quad (2)$$

In the direction corresponding to the true velocity, the signal energy is ideally zero. Therefore a distance measure between a velocity hypothesis and an orientation tensor is given by

$$d(\mathbf{v}, \mathbf{T}) = \mathbf{v}^T \tilde{\mathbf{T}} \mathbf{v} = \mathbf{v}^T \frac{\mathbf{T} - \lambda_3 \mathbf{I}}{\text{tr}(\mathbf{T})} \mathbf{v}, \quad (3)$$

where the isotropic part of the tensor, giving no information about the velocity, has been removed. The tensor has also been normalized with respect to the total energy in the neighborhood. Minimization of this distance measure gives an estimation of the velocity. In the moving line case, all velocities with correct normal component will yield the minimum value.

The chosen motion model is the commonly used affine motion model [1], where the motion at each point  $(x, y)$  is given by

$$v_x(x, y) = ax + by + c, \quad (4)$$

$$v_y(x, y) = dx + ey + f, \quad (5)$$

where  $v_x$  and  $v_y$  are the  $x$  and  $y$  components of the velocity and  $a$  through  $f$  are the coefficients of the model. To derive a distance measure between the motion model and an orientation tensor, note that equations (2), (4), and (5) can be rewritten as

$$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{S} \mathbf{p}, \quad (6)$$

where

$$\mathbf{S} = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (7)$$

$$\mathbf{p} = (a \ b \ c \ d \ e \ f \ 1)^T. \quad (8)$$

Hence

$$d(\mathbf{v}, \mathbf{T}) = \mathbf{v}^T \tilde{\mathbf{T}} \mathbf{v} = \mathbf{p}^T \mathbf{S}^T \tilde{\mathbf{T}} \mathbf{S} \mathbf{p} = \mathbf{p}^T \mathbf{Q} \mathbf{p} \triangleq d(\mathbf{p}, \mathbf{T}), \quad (9)$$

where  $\mathbf{Q} = \mathbf{S}^T \tilde{\mathbf{T}} \mathbf{S}$  is a positive semidefinite quadratic form.

Motion model parameters for a region are determined by minimization of the sum of the distances from the parameters to the orientation tensors in the region:

$$d_{tot}(\mathbf{p}) = \sum_i d(\mathbf{p}, \mathbf{T}_i) = \mathbf{p}^T \left( \sum_i \mathbf{Q}_i \right) \mathbf{p} = \mathbf{p}^T \mathbf{Q}_{tot} \mathbf{p}, \quad (10)$$

where the sum is taken over the pixels in the region. Now the problem is to find the vector  $\mathbf{p}$  that minimizes the

quadratic form  $\mathbf{p}^T \mathbf{Q}_{tot} \mathbf{p}$  with the restriction that the last element of  $\mathbf{p}$  has to be 1. Make the partitions

$$\mathbf{p} = \begin{pmatrix} \bar{\mathbf{p}} \\ 1 \end{pmatrix}, \quad \mathbf{Q}_{tot} = \begin{pmatrix} \bar{\mathbf{Q}} & \mathbf{q} \\ \mathbf{q}^T & a \end{pmatrix}, \quad (11)$$

where  $\bar{\mathbf{p}} = (a \ b \ c \ d \ e \ f)^T$ ,  $\bar{\mathbf{Q}}$  is a symmetric matrix,  $\mathbf{q}$  a vector, and  $a$  a scalar. Then

$$d_{tot}(\mathbf{p}) = \bar{\mathbf{p}}^T \bar{\mathbf{Q}} \bar{\mathbf{p}} + \bar{\mathbf{p}}^T \mathbf{q} + \mathbf{q}^T \bar{\mathbf{p}} + a, \quad (12)$$

which is minimized by

$$\hat{\mathbf{p}} = -\bar{\mathbf{Q}}^{-1} \mathbf{q}. \quad (13)$$

In the case that  $\bar{\mathbf{Q}}$  should happen to be singular or close to singular, the inverse can be replaced by the pseudo inverse.

This method of estimating motion model parameters is by no means restricted to the affine motion model used here. In fact, as can be deduced from equation (9), the only requirement is that the model be linear in its *parameters*.

### 3. Segmentation

The goal of the segmentation is to partition the image into a set of disjoint regions, so that each region is characterized by a coherent motion, with respect to the chosen affine motion model. Here a region is defined to be a nonempty, *connected* set of pixels. The segmentation algorithms are based on competitive region growing. The basic algorithm is first presented in abstract form.

#### 3.1. The competitive algorithm

To each region  $R$  is associated a cost function  $C_R(\mathbf{x})$ , which is defined for all pixels in the image and may vary as the region grows. Regions are extended by adding one pixel at a time. To preserve connectivity the new pixel must be adjacent to the region, and to preserve disjointedness it must not already be assigned to some other region. The new pixel is also chosen as cheap as possible. The details are as follows.

Let the border  $\Delta R$  of region  $R$  be the set of nonassigned pixels in the image which are adjacent to some pixel in  $R$ . For each region  $R$ , the possible candidate,  $N(R)$ , to be added to the region is the cheapest pixel bordering to  $R$ , i.e.

$$N(R) = \arg \min_{\mathbf{x} \in \Delta R} C_R(\mathbf{x}). \quad (14)$$

The corresponding minimum cost for adding the candidate to the region is denoted  $C_{min}(R)$ . In the case of an empty border,  $N(R)$  is undefined and  $C_{min}(R)$  is infinite.

Assuming that a number of regions  $\{R_n\}$  in some way have been obtained, the rest of the image is partitioned as follows.

1. Find the region  $R_i$  for which the cost to add a new pixel is the least, i.e.  $i = \arg \min_n C_{min}(R_n)$ .
2. Add the cheapest pixel  $N(R_i)$  to  $R_i$ .
3. Repeat the first two steps until no pixels remain.

Note that it does not matter what the actual values of the cost functions are. It is only relevant which of them is lowest. Hence the algorithm is called competitive.

### 3.2. Segmentation algorithm 1

The first segmentation algorithm segments each frame in the sequence independently. The only input is an orientation tensor field for the image, containing one tensor for each pixel. Since no previous knowledge of how the image should be segmented is assumed, there is no way of knowing how many regions there should be, or where they should be located. To handle this problem it is necessary to introduce some kind of seeds into the competitive algorithm. These have the form of preliminary regions.

The cost function should of course be related to how well the points in the image fit to the motion in the region. Affine parameters for a given region are computed by the method described in section 2. The cost function is then given essentially by  $d(\mathbf{p}, \mathbf{T})$ , but with some modifications.

It should be noted that robust estimation of the motion parameters requires that the regions are not too small, less than about 200 pixels is not recommended. Therefore the seeds has to be at least that size. At the beginning of the algorithm there is a large number of seeds spread all over the image. In contrast to the regions, the seeds are allowed to overlap. The quality of a seed is determined by how well it can be described by a single motion model. Motion parameters for the seeds are computed in the same ways as for the regions and the maximum cost among a seed's own points is used to rank the seeds.

The first thing that happens is that the seed with the least maximum cost is converted into a region. From now on there is a competition between the remaining seeds, trying to be converted into regions, and between the regions, trying to grow. This competition can be formalized into the following steps.

1. Among the seeds, choose the one with the least maximum cost as aspirant for conversion into a region.
2. As in the competitive algorithm, find the cheapest pixel that may be added to one of the real regions.
3. Compare the least maximum cost from step 1 to the cost of the cheapest pixel in step 2.
  - (a) If the least maximum cost is best, convert the corresponding seed to a region.

- (b) Otherwise, add the cheapest pixel to the corresponding region.

To avoid excessive fragmentation of the image into small regions, the comparison cannot be made directly between the least maximum cost and the cost of the cheapest pixel. Instead the first value is multiplied by a penalty factor  $\lambda$  before the comparison is made.

4. The seeds may overlap each other but not the regions. Therefore some seeds may have to be rebuilt.

This process is repeated until all pixels have been claimed. An example of the algorithm at work is given in figure 2, where frame 1 from figure 1 is being segmented. The darker the region, the earlier it has been included in the segmentation.

### 3.3. Segmentation algorithm 2

The second segmentation algorithm takes advantage of the segmentation of the previous frame in the sequence. One reason for this is that by using old information, the new segmentation can be done faster and more robustly. Another reason, that is important in some applications, is to obtain a segmentation of the sequence that is consistently labeled from frame to frame.

The inputs to this algorithm are an orientation tensor field for the current frame and a segmentation of the previous frame. Initial regions are constructed from the previous segmentation. It is assumed that the current frame contains the same regions as the previous one and that the movements of all regions are small. Thereby good initial regions can efficiently be obtained by shrinking the old regions in a connectivity preserving manner.

For the initial regions, new motion parameters are computed. Then the rest of the pixels are distributed strictly according to the competitive algorithm. The cost function from algorithm 1 is still used. The algorithm is illustrated in figure 3.

## 4. Discussion

The presented algorithms have a number of interesting properties and, admittedly, some weaknesses. There is still a large potential for improvements and further development.

To start with the motion model estimation, it is interesting to note that it copes well with the aperture problem, by combining information over a region and because the orientation tensors can represent normal velocities as well as point velocities. A weakness is that it is hard to estimate tensors correctly for large velocities. A solution would likely require a multiple scale approach [4].

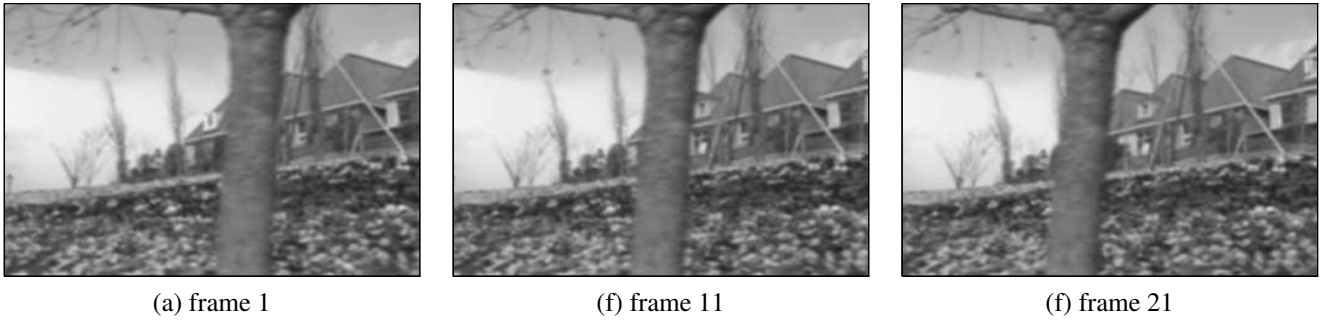
The segmentation algorithms are characterized by their competitive nature and the connectivity requirement.

Neighboring regions grow towards each other until they meet, and even though the orientation tensors close to the border may be somewhat off, far away regions cannot grab these points. Occasional points with very noisy tensors may fit badly to the motion models of all regions, but they will get surrounded by some region and sooner or later be included themselves. The same is true for nontextured parts of an object. Although these parts give no information about the motion, they will be incorporated with the surrounding region. A weakness with the connectivity requirement is that an object may artificially be split into multiple regions if it is occluded. But then a comparison of the motion models is likely to reveal this fact.

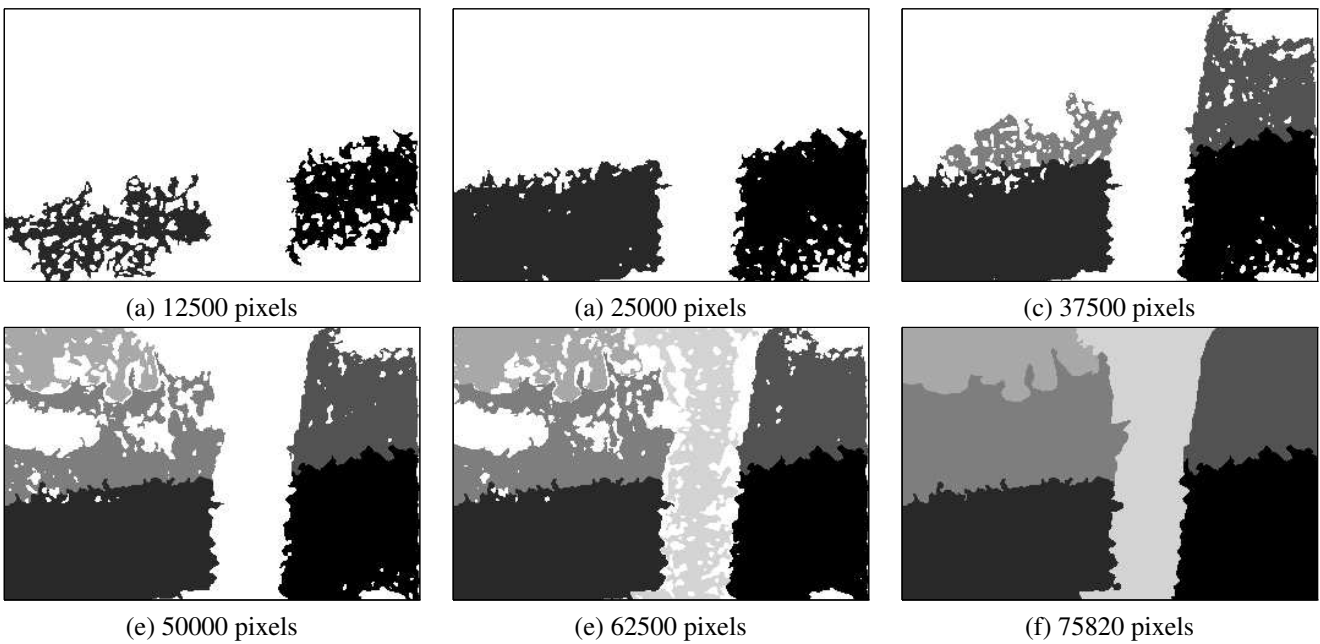
Additional weaknesses are that the algorithms have problems finding very small objects and fast moving objects, due to limitations in the motion model estimation. Algorithm 2 suffers from the assumption that no regions appear or disappear between frames. An algorithm that both takes advantage of previous segmentations and allows changes to the set of regions would be a useful addition. Finally both algorithms, but algorithm 1 in particular, have high computational complexity. They do, however, have good potential for parallelization.

## References

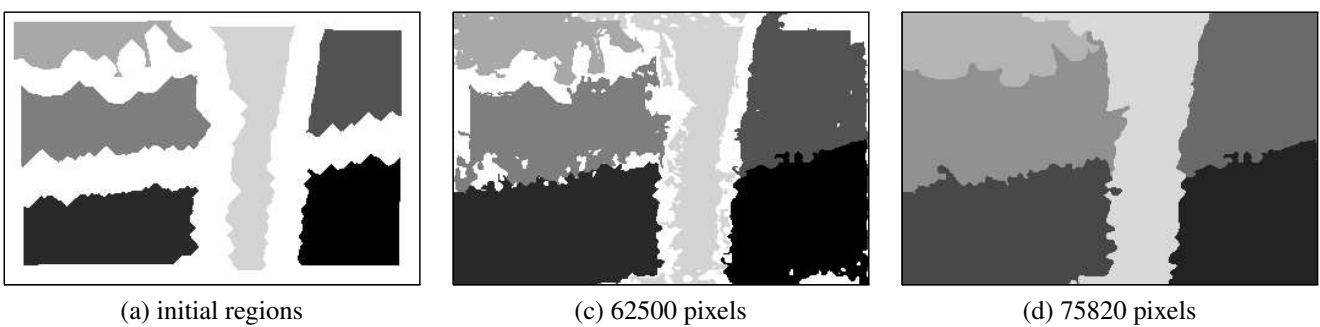
- [1] F. Dufaux and F. Moscheni. Segmentation-based motion estimation for second generation video coding techniques. In L. Torres and M. Kunt, editors, *Video Coding: The Second Generation Approach*, chapter 6, pages 219–263. Kluwer Academic Publishers, 1996.
- [2] G. Farnebäck. Motion-based Segmentation of Image Sequences. Master's Thesis LiTH-ISY-EX-1596, Computer Vision Laboratory, S-581 83 Linköping, Sweden, May 1996.
- [3] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
- [4] J. Karlholm. Efficient spatiotemporal filtering and modelling, June 1996. Thesis No. 562, ISBN 91-7871-741-8.
- [5] L. Torres and M. Kunt, editors. *Video Coding: The Second Generation Approach*. Kluwer Academic Publishers, 1996.
- [6] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.
- [7] J. Y. A. Wang and E. H. Adelson. Spatio-temporal segmentation of video data. In *Proceedings of the SPIE Conference: Image and Video Processing II*, February 1994.
- [8] J. Y. A. Wang, E. H. Adelson, and U. Desai. Applying mid-level vision techniques for video data compression and manipulation. In *Proceedings of the SPIE: Digital Video Compression on Personal Computers: Algorithms and Technologies*, February 1994.



*Figure 1. Selected frames from the flower garden sequence.*



*Figure 2. Development of the regions in segmentation algorithm 1. The number of classified pixels is indicated.*



*Figure 3. Development of the regions in segmentation algorithm 2.*