MOTION SEGMENTATION AND TRACKING USING A SEEDED REGION GROWING METHOD

I. Grinias and G. Tziritas
Institute of Computer Science - FORTH,
P.O.Box 1385, Heraklion, 711 10 Crete, Greece
and
Department of Computer Science, University of Crete
P.O. Box 1470, Heraklion, Greece
E-mails: {grinias,tziritas}@csd.uch.gr

ABSTRACT

We describe a user-guided system for the segmentation of an image sequence. A first segmentation which involves two images of the sequence is presented, as well as the tracking of its result during a number of frames. The segmentation algorithm is a Region Growing algorithm. The main segmentation feature is the motion of the objects presented in the image, which is combined with the information obtained by their intensity or colour –if neccessary. Two "post-processing" techniques are proposed to stabilize the object boundaries over time. A sense of layering is embedded in the whole process.

1 INTRODUCTION

2-D motion segmentation is an important step in image sequence analysis. Its results could be used for the determination of the 3-D motion characteristics of the scene objects, as well as for feature-oriented coding [1], object-based video manipulation and video retrieval by content. A recent review of 2-D motion segmentation methodology is given in [2].

In the user guided, semi-automatic motion segmentation method described below, only translational object motion is admitted, but it could be extended to any other motion parametric model. Motion parameters are estimated by a *region matching* (RM) technique, which is an extension of the block-matching for regions of any shape and provides the required motion estimation robustness.

A Seeded Region Growing (SRG) segmentation algorithm, initially proposed in [3], is modified adequately and used here. It is used in two ways: for segmenting the motion field between two frames, when the regions seeds are user provided, and for tracking previously extracted objects. In this last case, a simple user-given *layered representation* of the objects is introduced, in order to implement the automatic extraction of the SRG's initial sets. Clearly, this representation is supplied by the user and not by any information extracted by the segmentation algorithm.

Furthermore, two "post-processing" procedures have been implemented to stabilize the object boundaries over time. The first one "smooths" the boundary while the latter computes an "average" shape for each object.

This paper is organized as follows. In Section 2, we present how SRG segmentation algorithm is modified for the needs of the motion-based segmentation. Section 3 describes the way that the SRG algorithm can also be used for the tracking of the initial segmentation in time. Next, the operations, that are performed on each segmentation result in order to improve it, are explained in detail (Section 4). Finally, in Section 5, we present the results of applying our segmentation system on the sequences *Foreman* and *CoastGuard*.

2 MOTION SEGMENTATION USING SRG

In this section we describe the initial segmentation, when the user provides an approximative pattern for each image object. Thus the number of regions is known, and a subset of points of each region is provided. A velocity vector is computed for each region based on the provided subset of points using a region matching technique, as an extension of block-matching algorithm. Since at this first segmentation there is no previous information about objects motion, the region matching method is often performed in a large search area and may be computationaly expensive. It must be noticed however that this segmentation is applied only one or two times over the whole sequence.

Let Ω be the set of points of the whole image, R_i $(1 \leq i \leq n)$ be the sets corresponding to the image regions, and B be the set of boundary points $(\cup_{i=1}^{n} R_i \cup B = \Omega)$. Initially, the image regions are incomplete. Let $A_i (1 \leq i \leq n)$ be the initial sets of region points. For each A_i a motion vector is computed. This motion vector is only once computed, and considered as the feature characterizing region R_i.

For the needs of motion segmentation the criterion that measures the dissimilarity of an unlabeled point $p = (x, y)$ to its neighboring set A with velocity vector $(\hat{u}, \hat{v})$ becomes:

$$\delta_{IM}(p) = |I_k(x, y) - I_{k-1}(x - \hat{u}, y - \hat{v})| \qquad (1)$$

where I_k, I_{k-1} are the corresponding intensity frames at times k and $k-1$. However, when motion information is not sufficient, the criterion used is:

$$\delta_I(p) = \delta_{IM}(p) + \lambda|I_k(x,y) - \mu_{W(p)\cap A}| \quad (2)$$

where $\mu_{W(p)\cap A}$ is the average intensity of points of region A in a small window centered at p. By substituting intensity with the colour images $\vec{I}_{k-1}$ and $\vec{I}_k$ in equations (1), (2) we obtain two similar criteria denoted as δ_{CM} and δ_C.

A list of all unlabeled points is created. This list is ordered according to the criterion value δ and has been implemented using AVL binary trees [4] in order to reduce the computational cost. Pixels are labeled according to this list and the list is updated after a new label assignment.

The criterion value $\delta(p)$ of each unlabeled pixel p is computed only once, when the pixel is inserted in the list, since the characteristic feature of each set is initially computed and remains unchanged during the algorithm' s execution. This fact, further reduces the computational cost.

At the end of this procedure every pixel p is assigned a label $i(p) \in \{1, 2, \ldots, n, b\}$ where b is a special index of the boundary set B. The output of the algorithm also includes the velocity vector of each set, as its characteristic feature.

3 OBJECT TRACKING

We now describe how the result of the first segmentation (set map i_0) is tracked during a number of sequence frames.We assume that this result has been traced up to frame $k-1$ (set map i_{k-1}) and we now wish to obtain the set map i_k that corresponds to frame k (k segmentation). The initial sets for the k segmentation are provided by the set map i_{k-1}.

Furthermore, for the needs of tracking, we assume that sets are ordered by the user according to their depth from the camera:

$$\begin{aligned} &\forall i, j \in \{1, 2, \ldots, n\}, \quad R_i \text{ moves behind } R_j \\ &\text{if, and only if,} \quad i < j \end{aligned} \quad (3)$$

Therefore the sets, as considered here, correspond to depth layers, with approximately an homogeneous translational motion for each layer. Then, for each set $R_i (i > 1)$ that is included in the set map i_{k-1} (except the background, R_1), the following operations (from the deepest layer to the less deep one, in decreasing index order) are performed in order to provide the initial sets of the set map i_k:

1. Estimation of the displacement $(\hat{u}_k, \hat{v}_k)_i$ of R_i from image $k-1$ to image k by applying RM (possibly with subpixel accuracy) in a small search area around the velocity vector of R_i, $(\hat{u}_{k-1}, \hat{v}_{k-1})_i$ (assumption of small acceleration).
2. Dilation of the boundary set B, which provides the unlabeled points needed by SRG. The dilation degree is specified by the user. Although this approach is simple and rapid, it cannot retain important "thin" elongated parts of points that may be contained in the object's morphology. For images that include such objects we have implemented a more complex dilation operation: points of object R_i, whose deletion by the dilation operation locally disconnects it, are not deleted; candidate points for deletion that do not belong to set R_i are deleted as described previously.
3. Translation of the "shrinked" object R_i from image $k-1$ to image k according to the displacement $(\hat{u}_k, \hat{v}_k)_i$.

The last step, before the execution of SRG, is the estimation of the background's velocity vector. Finally, SRG is applied giving the segmentation map i_k.

4 POST PROCESSING

In order to stabilize the boundary of objects over time, the shape of the obtained regions is post-processed, as described below.

4.1 Boundary Smoothing

This operation involves only the set B of boundary pixels of SRG's set map. First, the boundary is traced (for example, in clockwise direction) in order to obtain an ordered list of the boundary pixels:

$$B_{ord} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)\}$$

where L is the number of these boundary pixels. Given a symmetric 1-D low-pass filter $h = [h_n]$ $(-N \leq n \leq N)$ the new position $(\hat{x}_i, \hat{y}_i)$ of each $(x_i, y_i) \in B_{ord}$ is computed as

$$\begin{bmatrix} \hat{x}_i \\ \hat{y}_i \end{bmatrix} = \begin{bmatrix} \sum_{k=-N}^{N} h_k x_{i-k} \\ \sum_{k=-N}^{N} h_k y_{i-k} \end{bmatrix}$$

Boundary pixels that become unlabeled, are assigned to the set they adjoin.

This procedure may be repeated more than once in order to achieve the required boundary smoothing. On the other hand, this operation may alter the shape of some objects by oversmoothing the angles that it contains. The method described below overcomes this difficulty.

4.2 Shape Averaging

This method improves the set map i_k of the k-segmentation using the information that is provided by m previous segmentation maps, noted as $i_{k-m}, i_{k-m+1}, \ldots, i_{k-1}$. Again, we consider the layered representation of sets-objects (Eq. 3). Then, for each set $R_i (i > 1)$ (from the less deep to the deepest one) we proceed as follows.

1. First, each set map i_{k-l} $(1 \leq l \leq m)$ is warped to its respective position in the map i_k according to the total displacement from frame $k-l$ to frame k. In this way, we obtain m set maps for frame k.

2. Given these compensated maps, the the majority law is applied for obtaining the final segmentation map.

The process ends with the extraction of the boundary points of the new map that is computed.

5 EXPERIMENTAL RESULTS

We have implemented the segmentation system in C on an ULTRA Sparc workstation. We present hereafter the results that were obtained by the segmentation of two sequences.

Figure 1: Tracking result for *Foreman*'s frame 51.

Figure 2: The layered representation of the segmentation result of *Foreman*'s frame 51.

We first applied the system on the 250 frames of the *Foreman* sequence. Image dimensions are 352×288. In this sequence, the camera tries to fixate the face of the foreman, while he moves, performing a complex motion which the most time is translational. Hence, the background appears to move too. The tracking algorithm is performed in the parts of the sequence, where the motion of objects is described by the translational model. In order to extract SRG's initial uncompleted sets, we use the pure dilation of the boundary set, since the illustrated foreman's face does not contain "thin" elongated parts. For the needs of SRG, the segmentation criterion used is the metric δ_C, because the intensity of the frames isn't sufficient to distinguish foreman from his background objects. The value of λ was set to 0.4. The tracking result for frame 51 is illustrated in Fig. 1. In this figure the white curve represents the boundary between the layer of foreman and the background layer (see Fig. 2). On the other hand, when foreman's motion is not translational, we use the first segmentation between two frames with the same similarity criterion, but with large λ value, which shows that the segmentation is mainly based on the colour information. After each segmentation the boundary smoothing operation is performed twice, on the segmentation result in order to keep the boundary stable over time. The average computation time per image is about 25 sec including the boundary smoothing operation.

Figure 3: Segmentation result for *CoastGuard*'s frame 66.

Figure 4: Improved result for *CoastGuard* 's frame 66.

We also applied our system on the 300 frames of the

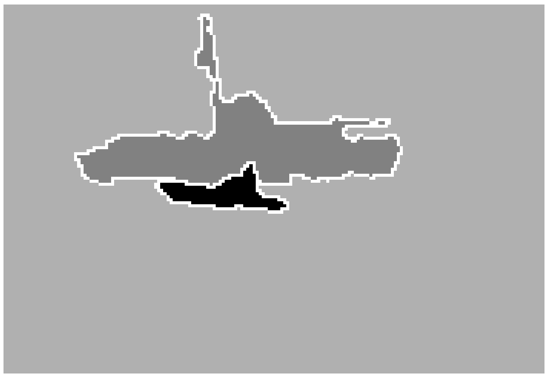

Figure 5: *CoastGuard* 's Layers.

CoastGuard sequence, with image dimensions 352×240. In the first 120 frames of the sequence, a ship and a boat move, one against the other. Furthermore, the boat moves in front of the ship. The overall scheme can be represented by three layers, one for the background, one for the ship and one for the boat, in that order (according to their depth from the camera). In the remaining frames, the boat slides out of the image, the ship appears to be stationary and the background undergoes a translational motion due to the camera pan. The motion of the objects remains translational for the entire sequence. This fact, allows the tracking of the first segmentation result over a large number of images and therefore decreases the number of the needed applications of the initial segmentation. The initial sets for SRG's execution are obtained by the "conditional" dilation operation. In this manner, the mast and the main body of the ship retain their connectivity. The segmentation criterion we use for the segmentation is δ_{IM}. This criterion is used for unlabeled pixels that adjoin the background set and the ship. For the pixels that are neighbors of the boat, we use the criterion δ_I (with $\lambda = 0.2$), because the boat's intensity is uniform. This selection of criteria improves the segmentation result of the frames in which the boat moves in front of the ship. After each segmentation, we apply the boundary smoothing method only once, because when this process is repeated once again the shape of the objects changes. Furthermore, in the part of the sequence in which the ship appears to be stationary, this operation does not give the required boundary stability. For these reasons, the post-processing also involves the shape-averaging procedure (Subsection 4.2). This procedure not only stabilizes the boundary of the objects, but also improves the segmentation result of the frames which contain occlusions providing a very good layered representation. The effect of this method is shown in Figures 3 and 4. Fig. 3 illustrates the segmentation result for the frame 66 and Fig. 4 shows the effect of shape averaging filter on that result, using the information of the previous $m = 12$ segmentation results. In Fig. 5, we see the three layers that are obtained by the improved segmentation result of Fig. 4. The total average computational time per image is about 40 sec for $m = 12$.

A complete demonstration of the segmentation and tracking algorithm is given in: *http://www.ics.forth.gr/proj/cvrl/demos/NEMESIS/*.

6 CONCLUSION

We have described a segmentation system based on the motion of the objects of a sequence. It is assumed that the objects move translationaly in the plane. The estimation of motion parameters is obtained by a region matching technique. Other features such as the intensity or the colour of the image objects are used in order to obtain a better segmentation result. The segmentation algorithm is called Seeded Region Growing and has been used for the segmentation of static images. For the needs of tracking, we introduce a layered representation of the image objects in order to be able to cope with sequences which include multiple motions, moving background and occlusions. The order of layers is supplied by the user. Two procedures have been developed in order to improve the segmentation result that is obtain by SRG. The first one involves only the boundary pixels while the former is applied on every point of the objects. The segmentation system can be extented to cope with motions that are not described by the simple translational model –which we have assumed. In addition, we could let the user provide unique parameters for each individual sequence object –since, for example, the number of previous set maps that are needed in order to obtain a good "average" shape may be different for each object.

Acknowledgement: This work is funded in part under the NEMESIS ESPRIT project.

References

[1] G. Tziritas and C. Labit, *Motion Analysis and Image Sequence Coding*, Elsevier, 1994.

[2] A. Mitiche and P. Bouthemy, "Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods", *International Journal of Computer Vision*, Vol. 19, No. 1, pp. 29-55, July 1996.

[3] R. Adams and L. Bischof, "Seeded Region Growing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 6, pp. 641-647, June 1994.

[4] H. Lewis and L. Denenberg, "Data Structures and their Algorithms".