ELSEVIER

Signal Processing: *Image Communication* 16 (2001) 977–986

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.com/locate/image

A semi-automatic seeded region growing algorithm for video object localization and tracking

I. Grinias, G. Tziritas*

Computer Science Department, University of Crete, P.O. Box 2208, Heraklion, Greece

Received 16 February 2000; received in revised form 5 April 2001; accepted 23 April 2001

Abstract

This paper describes a semi-automatic method for moving object segmentation and tracking. This method is suitable when a few objects have to be tracked, while the camera moves and fixates on them. The user delineates approximately the initial locations in a selected frame and specifies the depth ordering of the objects to be tracked. First, motion-based segmentation is obtained through an initial application of a region growing algorithm. The partition map is sequentially tracked from frame to frame using motion compensation and location prediction. The segmentation map is obtained by the region growing algorithm. Translational motion is assumed for the moving objects, and local intensity or color average may be used as additional features. A post-processing procedure regularizes the object boundaries over time.

1. Introduction

Motion segmentation is a key step in image sequence analysis and its results are extensively used for determining motion features of scene objects, as well as for coding purposes to reduce storage requirements [13]. The development and wide-spread use of the international coding standard MPEG-4 [12,5], which relies on the concept of image/video objects as transmission elements, has raised the importance of these methods. The semi-automatic segmentation technique presented here is suitable for video object extraction for post-production purposes and object-scalable coding such as that introduced in the MPEG-4 standard. It could be also useful in video content description for video indexing and retrieval, and in user-assisted surveillance applications.

Various approaches have been proposed for motion or spatio-temporal segmentation. A recent survey of these techniques can be found in [7]. In these approaches a 2-D motion or optical flow field is taken as input and a segmentation map is produced, where each region undergoes a movement described by a small number of parameters. There are top–down techniques which rely on the outlier rejection starting from the dominant motion, usually that of the background. Other techniques are bottom–up starting from an initial segmentation and merging regions until the final partition emerges. Direct methods are reported too. All these techniques could be considered automatic, since only some tuning parameters are fixed by the user.

*Corresponding author. Fax: +30-81-393501.

E-mail addresses: grinias@csd.uoc.gr (I. Grinias), tziritas@csd.uoc.gr (G. Tziritas).

0923-5965/01/$ - see front matter
PII: S0923-5965(01)00014-5

In the pioneering work of Adiv [2] a planar surface is assumed, and affine or 8-parameter models are used for describing and segmenting the optical flow field. Wang and Adelson [14] obtained a segmentation of the optical flow field by fitting an affine parametric model on the basis of robust estimation and k-means clustering. Kruse [6] proposed a motion segmentation method combining three techniques: (a) randomized Hough transform for an initial segmentation, (b) merging of regions with similar motion, and (c) refinement based on a Markov random field model. Odobez and Bouthemy [10] proposed a direct segmentation algorithm consisting of four steps: (a) prediction of the partition map from the segmentation map of the previous frame using motion compensation, (b) robust estimation of a parametric motion model for the different regions, (c) updating the predicted partition map using a Markov random field model for the labels, and (d) detection of new regions to handle the appearance of new objects. Moscheni and Bhattacharjee [8,9] proposed a bottom–up approach starting from an initial set of regions and then merging them using a similarity criterion based on both brightness and motion information. A graph-based hierarchical clustering algorithm is used to merge regions. Cheong and Aizawa [4] proposed an algorithm consisting of two main steps: in the first step a pre-segmentation of the optical flow field is carried out based on a probabilistic clustering method; then a parametric motion model is estimated and regions with similar motion parameters are merged to obtain the final partition map. Altunbasak et al. [3] proposed an iterative technique with three steps: (a) pixel labeling and parametric motion estimation using the 2-D motion field, (b) pixel labeling and motion parameter updating minimizing the sum of squares of the displaced frame difference, and (c) region clustering by color region-based intensity matching. Another method is proposed by Salembier et al. [11] for video coding applications, which have specific requirements and subsequent implications for the motion segmentation.

The algorithm proposed in this paper exploits, adapts and extends known techniques in a scheme where the user is present in the processing loop. The human operator provides high-level information indicating the initial position of the objects. Thus, the operator defines these objects and gives their relative depth for handling occlusions. In the user-guided and semi-automatic motion segmentation method described in this paper, only translational motion is handled; it could, however, be extended to other motion parametric models, for example the affine model, if they are globally valid for the corresponding object. The motion parameters of each object are estimated by a region matching (RM) technique, which is an extension of block matching to regions of any shape and provides the required computational robustness.

Our approach relies on a seeded region growing (SRG) segmentation algorithm, initially proposed in [1] and modified to suit our purposes. We use this algorithm for the "initial segmentation", involving two consecutive images of the sequence, as well as for tracking the resulting regions over the whole sequence. In the first case, we make no assumption about the shape or the velocity vector of objects. An initial segmentation map is provided by the user-only in a selected frame, often the first frame in the sequence. During tracking, the determination of initial sets for each segmented region is based on the extracted image objects of the previous segmentation. Furthermore, a simple user-given layered representation of objects is introduced, in order to implement the automatic extraction of the initial sets. Indeed, the depth layer ordering is supplied by the user and not deduced from information extracted by the segmentation algorithm.

The remainder of this paper is organized as follows. In Section 2, the motivation for the proposed method is presented, the initial segmentation is described, as well as the SRG algorithm, as modified for the needs of motion-based segmentation. Section 3 presents how the SRG algorithm may be used for the temporal tracking of the initial segmentation. Next, in Section 4, the post-processing operations that are performed for the enhancement of the segmentation result are explained in detail. Finally, in Section 5, we present the results of applying the proposed segmentation algorithm to the MPEG-4 test sequences *Foreman* and *CoastGuard*.

2. User-guided region growing-based motion segmentation

2.1. Overall structure of video segmentation algorithms

A common requirement in image sequence analysis is the extraction of a small number of moving objects from the background. The key feature for obtaining the segmented image can be motion, but the segmentation of a dense 2-D motion field often leads to over-segmentation. A fully automatic motion segmentation method could give results which cannot be directly interpreted and exploited for editing and post-production, or for image content description. The presence of a human operator, called here the *user*, can greatly facilitate the segmentation work, for obtaining a semantically interpretable result.

The more demanding stages of the whole process of object localization and tracking are the localization in the first frame, the possible topology changes due to motion and the tracking of objects when they become occluded/unoccluded. The proposed algorithm incorporates an active user for segmenting the first frame, and for subsequently dealing with occlusions during the moving object tracking.

For each object, including the background, the user draws a closed contour entirely contained within the corresponding object. For each object thus specified by the user, a 2-D motion vector is estimated by region matching, considering essentially rigid translational motions. Finally, a region growing algorithm expands the initial objects to their actual boundaries. The region growing is based on a dissimilarity criterion which includes two terms, the displaced frame difference and a local difference from the labeled objects.

Having obtained the segmentation of the first frame, the tracking of any moving object can be done automatically. Only the layered representation of the scene is needed. For this the user must specify the depth ordering for all the objects. In the simplest case of two objects the foreground should be discriminated from the background. This is needed for correctly handling overlaps.

In the first step the motion vector of each object is re-estimated, since its movement may not be temporally constant: motion amplitude and/or direction might change from frame to frame. As some errors may occur in the moving object localization stage, this motion estimation is performed after shrinking the objects, in order to ensure that object contours lie within the objects. The contracted objects are projected onto the next frame using motion compensation. Thus, they are projected onto their predicted position. From that predicted position the region growing algorithm is applied.

As the motion estimation module is common to both the first segmentation and tracking, we give a concise description now, fixing the notation used herein and the exact formulation of the technique applied. We assume that each moving region undergoes a simple translational planar motion, represented by a 2-D velocity vector (u, v). We need to estimate this vector using the intensity functions of two consecutive frames I_k and I_{k-1}, without making any assumptions about the shape or size of the region. Since the region matching technique used is a standard one, we give here only the matching or "distance" criterion. The estimation of the velocity vector $(\hat{u}, \hat{v})$ is based on the sum of absolute displaced frame differences:

$$(\hat{u}, \hat{v}) = \arg\min_{(u, v)\in S} \sum_{(m, n)\in R} |I_k(m, n) - I_{k-1}(m-u, n-v)|, \tag{1}$$

where S is a set of possible displacement vectors defining the search area, and R the set of region points considered for region matching and motion estimation. The search area is defined with respect to a predicted displacement vector, in which case only the vector update is searched, limiting significantly the computational cost, even if sub-pixel accuracy is required.

2.2. Object initialization

The initial regions required by the region growing algorithm must be provided, or at least

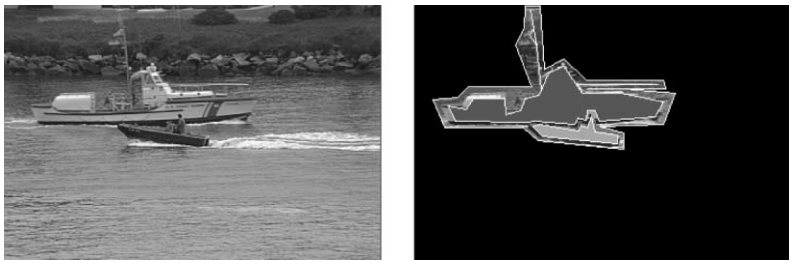


Fig. 1. User provided input of initial sets for *CoastGuard*'s frame 58.

approved, by the user. The region growing algorithm requires the input of some initial uncompleted sets of points. A tool has been built for drawing a polygon as desired inside any object. Then points which are included within these boundaries define the initial sets of object points. This concept is illustrated in Fig. 1, where the input of initial sets for the frame 58 of the sequence *CoastGuard* is shown. The user provides an approximate pattern for each object in the image that is to be extracted and tracked.

Once the initial sets have been specified, the velocity vector of each set can be computed. Since there is no previous information about the objects' motion, the region matching in Eq. (1) is often performed in a large search area and may be computationally expensive. Fortunately, this segmentation is rarely applied, normally once or twice over the whole sequence. The next and final step, described in the following subsection, involves the application of the region growing algorithm to the unlabeled points. The region growing method is called seeded because all the initial regions are supplied by the user.

2.3. Motion segmentation

2.3.1. Dissimilarity measure

As mentioned above, the choice of a particular dissimilarity measure $\delta(\cdot,\cdot)$ depends on the segmentation feature used. In our implementation, the main feature is the velocity vector of initial set A_i, which also characterizes the final segmented region R_i. Hence, the distance of pixel $p=(x,\ y)$ from set A, characterized by the displacement vector $(\hat{u},\ \hat{v})$, is defined as

$$\delta_{1I}(p,\ A)=|I_k(x,\ y)-I_{k-1}(x-\hat{u},\ y-\hat{v})|, \tag{2}$$

where I_k, I_{k-1} are the corresponding intensity functions for frames k and $k-1$. In other words, $\delta_{1I}(p,\ A)$ measures how well the displacement of pixel p is described by the velocity vector that has been computed for set A. However, if the information provided by the motion of sets is not sufficient for an acceptable segmentation result, we may use a criterion that takes into account the difference between the intensity value of p and the average intensity $\mu_{W(p)\cap A}$ of points of region A in a small window centered at p. Then, the dissimilarity criterion becomes a weighted sum of the two dissimilarity measures:

$$\begin{aligned}\delta_{2I}(p,\ A)=&(1-\lambda_A)\delta_{1I}(p,\ A)\\&+\lambda_A|I_k(x,\ y)-\mu_{W(p)\cap A}|.\end{aligned} \tag{3}$$

The parameter λ_A expresses the relative significance of the second term in the composite measure $\delta_{2I}(\cdot,\ A)$ for the set A, and it should be related to the variance of the two terms of the weighted sum. The parameter λ_A could be set to the ratio of the mean absolute deviation of the displaced frame difference to the mean absolute local intensity deviation.

Similarly, in cases where the intensity information is not sufficient, we may use the information provided by the color of images $k-1$ and k, denoted by $\vec{I}_{k-1}$ and $\vec{I}_k$,

$$\delta_{1C}(p,\ A)=\|\vec{I}_k(x,\ y)-\vec{I}_{k-1}(x-\hat{u},\ y-\hat{v})\| \tag{4}$$

and

$$\delta_{2C}(p, A) = (1 - \lambda_A)\delta_{1C}(p, A) + \lambda_A \|\vec{I}_k(x, y) - \vec{\mu}_{W(p)\cap A}\|. \quad (5)$$

The parameter λ_A has the same role as for the criterion based on intensity alone. Window dimensions depend on the objects' intensity/color distribution near their border. Hence, window W should be kept small enough to represent the local intensity around the initially unlabeled pixels of each object.

2.3.2. The seeded region growing algorithm

Motion segmentation is carried out by a seeded region growing algorithm which was initially proposed for static image segmentation using a homogeneity measure on the intensity function [1]. It is a sequential labeling technique, in which each step of the algorithm labels exactly one pixel, that with the lowest dissimilarity. Letting n be the number of objects (classes), an initial set of connected components $A_1^0, A_2^0, \ldots, A_n^0$ is required. These are sets of image points, such as the manually drawn segments of Section 2.2, referred to as *seeds*. At each step m of the algorithm, let B^{m-1} be the set of all yet unlabeled points which have at least one immediate neighbor already labeled, i.e., belonging to one of the partially completed connected components $\{A_1^{m-1}, A_2^{m-1}, \ldots, A_n^{m-1}\}$. In this work 8-connection neighborhoods are considered. Then one pixel of the border set B^{m-1}, that with the lowest dissimilarity, is selected and labeled. Thus the number of the connected components remains n, preset by the initial seeds, and we have the following sets of points $\{A_1^m, A_2^m, \ldots, A_n^m\}$. Finally, when the border set becomes empty after a number of steps equal to the number of initially unlabeled pixels, a segmentation map $(R_1, R_2, \ldots, R_n)$ is obtained with $A_i^m \subseteq R_i$ (for all i, m) and $R_i \cap R_j = \emptyset (i \neq j)$, where $\bigcup_{i=1}^n R_i = \Omega$ is the whole image.

When the seeded region growing algorithm is applied to motion segmentation, first, the velocity vector $(\hat{u}_i, \hat{v}_i)$ of each set A_i^0 is estimated by the region matching technique. This vector is the primary segmentation feature and remains unchanged for all the steps of the sequential algorithm until all pixels are labeled. The process continues, as described above, by successively labeling all unlabeled pixels. For each pixel $p \in B^{m-1}$, let us denote by $i(p) \in \{1, 2, \ldots, n\}$ the index of the set A_i^{m-1} that p adjoins. If the characterization of the sets is not updated during the sequential labeling process, the dissimilarity will be $\delta(p, A_{i(p)}^0)$ according to Eq. (2) or (4). Otherwise, the dissimilarity is measured from the features describing the set $A_{i(p)}^{m-1}$. If p adjoins two or more of the sets A_i^{m-1}, we define $i(p)$ to be the index of the set that minimizes the criterion $\delta(p, A_j^{m-1})$ over all neighboring sets A_j^{m-1}. In addition, we can distinguish a set F of boundary pixels and add p to F when p borders more than one set. In our implementation boundary pixels p are flagged as belonging to F and at the same time, they are associated with the set that minimizes the dissimilarity criterion over all sets on whose boundary they lie. The set of boundary points F is useful for boundary operations, as we shall see in Section 4. Then we choose among the points in B^{m-1} one satisfying the relation

$$z = \arg\min_{p \in B^{m-1}} \{\delta(p, A_{i(p)}^{m-1})\} \quad (6)$$

and append z to $A_{i(z)}^{m-1}$, resulting in $A_{i(z)}^m$. This completes one step of the algorithm and the process terminates when all pixels have been labeled.

For the implementation of the SRG algorithm, a list that keeps its members (pixels) ordered according to the criterion value $\delta(\cdot, \cdot)$ is used, traditionally referred to as sequentially sorted list (SSL). With this data structure available, the complete SRG algorithm is as follows:

S1 Label the points of the initial sets.
S2 Estimate the motion vector $(\hat{u}_i, \hat{v}_i)$ of each initial region (for all i).
S3 Insert all neighbors of the initial sets into the SSL (B^0).
S4 While the SSL is not empty:
  S4.1 Remove the first point y from the SSL and label it.
  S4.2 Test the neighbors of y and update the SSL:
    S4.2.1 Add neighbors of y which are neither already labeled nor

already in the SSL, according to their value of $\delta(\cdot,\cdot)$.

S4.2.2 Test for neighbors which are already in the SSL and now border on an additional set because of y's classification. These are flagged as boundary points. Furthermore, if their $\delta(\cdot,\cdot)$ is reduced, they are promoted accordingly in the SSL.

Note that motion parameters of each set are computed only once at the beginning of SRG (Step 2). In other words, we assume that the choice of initial sets is such that it leads to a sufficiently accurate estimation of the objects' motion using region matching on the initial regions. Consequently, we consider that the insertion of points into a set during SRG execution does not change the estimation of its parameters. These assumptions allow us to use RM only once for each set. Furthermore, there is no need to reorder the SSL elements, if their $\delta(\cdot,\cdot)$ value does not change during SRG execution. Hence, the execution time of SRG is greatly reduced. When SRG is completed, every pixel p is assigned a label $i(p)\in\{1, 2, \ldots, n\}$, while boundary information is maintained in set F. The output of the algorithm also includes the velocity vector for each set.

3. Tracking

We now describe how the result of the initial segmentation (set map i_0) is tracked over a number of consecutive frames. We assume the result has been tracked up to frame $k-1$ (set map i_{k-1}) and we now wish to obtain the set map i_k corresponding to frame k (partition of frame k). The initial sets for the segmentation of frame k are provided by the set map i_{k-1}. The description of the tracking algorithm follows, while the motivations and the steps of the algorithm have already been presented in Section 2.1.

However, for the purpose of tracking, a layered representation of the sets, rather than the planar one implied by SRG, is introduced in order to be able to cope with real world sequences which contain multiple motions, occlusions (caused, for example, by the motion of one object in front of another), or a moving background. Thus, we assume that sets are ordered according to their distance from the camera:

$$\forall i, j\in\{1, 2, \ldots, n\},\quad R_i \text{ moves behind } R_j \text{ if and only if } \quad i<j. \qquad (7)$$

In this way, set R_1 refers to the background, set R_2 moves in front of set R_1 and behind the other sets, etc. We further assume that this set ordering is supplied by the user.

Having this set ordering available, for each set $R\in\{R_2, R_3, \ldots, R_n\}$ of set map i_{k-1}, we perform the following operations in order of proximity, beginning with the most distant:

- The first step involves dilating the border between R and its neighboring sets-objects, denoted as ∂R. In this way, set R as well as the sets that adjoin it are shrunk along ∂R, providing the set of seeds A for R. Thus, this operation provides the labeled points that are needed by SRG. The degree of shrinkage is specified by the user. Although this approach is simple and rapid, it cannot retain important "thin" elongated parts of objects that may be contained in the object's morphology. For images that include such objects we have implemented a more complex dilation operation, which preserves connected components.
- The next step is to determine A's new position in image k. This localization requires the estimation of the displacement $(\hat{u}_k, \hat{v}_k)$ describing the translation of A from image $k-1$ to image k. It is assumed that the velocity of every object remains almost constant over time. For this reason, we assume that set R moves with the velocity $(\hat{u}_{k-1}, \hat{v}_{k-1})$, which was extracted for it by the segmentation $k-1$ and we further compute a vector $(\hat{du}, \hat{dv})$, which represents the estimated difference between the vector $(\hat{u}_{k-1}, \hat{v}_{k-1})$ and the true displacement (u_k, v_k) for the segmentation k. Thus, RM (with sub-pixel accuracy) can be limited to a small search area around the vector $(\hat{u}_{k-1}, \hat{v}_{k-1})$ which in turn leads to low computational cost.

- Once the displacement $(\hat{u}_k, \hat{v}_k)$ of A has been estimated, the "shrunken" version A of region R is moved from image $k-1$ to image k according to this displacement.

The last step, before applying SRG, is the estimation of the background's velocity vector.

Finally, SRG is applied to points that remain unlabeled after the above operations. In the case of tracking, SRG's similarity criterion has been modified to take into account information about occluded pixels provided by the set map i_{k-1} and the velocity vectors of objects $(\hat{u}_k, \hat{v}_k)$. Let us suppose that the unlabeled pixel $p = (x, y)$ must be inserted into the SSL according to its distance $\delta_1(p, A_i)$ from its neighboring set A_i with motion vector $(\hat{u}_i, \hat{v}_i)$. Pixel p is considered to be "occluded", if

$$i_{k-1}(x - \hat{u}_i, y - \hat{v}_i) \neq i_k(x, y). \tag{8}$$

In that case, p is inserted into the SSL using the criterion $\delta_{2I}(\cdot,\cdot)$ ($\delta_{2C}(\cdot,\cdot)$, if color information is used) with $\lambda_{A_i} = 1$. Otherwise, the criterion that has been specified at the beginning by the user for set A_i is used. In other words, since for occluded pixels there is no motion information at all, the criterion used is based only on the local intensity/color information of set A_i around pixel p.

4. Post-processing

The operations described below are applied to the segmentation result in order to regularize the object boundaries over time. They could be used in two ways:

C1. Independently, for the enhancement of the final segmentation result for all frames, or
C2. as part of the processing loop, in which case it is expected that they contribute to an improvement of the segmentation result by reducing boundary noise, as the result is temporally propagated.

Boundary smoothing involves only the set F of boundary pixels as defined in Section 2.3. First, the boundary is extracted and traced in order to obtain an ordered list of the boundary points. Then, a low-pass filter is applied to both pixel coordinates. Boundary points that become unlabeled are assigned to the set onto which they adjoin. The boundary smoothing may alter the shape of some objects by over-smoothing the angles that they possibly contain. The method described below overcomes this difficulty.

Shape averaging improves the set map i_k using information provided by previous segmentation maps. The number of such maps L_S is specified by the user and we denote the maps by $i_{k-L_S}, i_{k-L_S+1}, \ldots, i_{k-1}$. The improvement is obtained by computing an "average" shape for each image object, assuming that the shape of objects does not change over time. The output of this process is also a set map similar to that of SRG's output, denoted as S_k. Again, we consider the layered representation of objects as described in Section 3 and by Eq. (7), to cope with multiple motions and occlusions.

The nearest object can only occlude the others and will never itself be occluded. If a majority rule is used for the "average shape" operation, this object is definitely allocated to a point, if in the majority of the set maps this point is given the greatest label. We continue in the same way with the next object, having acquired the decision for the nearest object, and so on, progressing from the nearest object to the background. We describe in some technical detail these rules.

The procedure begins by setting $S_k(p) = 1$ for each pixel p. Then, for each set R_i $(i > 1)$ in order of proximity, beginning with the least distant, we perform the following operations:

- First, each set map i_{k-l} $(1 < l \leqslant L_S)$ is warped to its respective position in the map i_k. In this way, for each set map i_{k-l}, we construct a new set map M_{k-l,R_i}, with the property that the set R_i has been moved to the position that it is placed by the map i_k.
- Once the motion-compensated set maps M_{k-l,R_i} have been extracted, for each pixel p, we compute the number of maps $n_{\{R_j; j<i\}}(p)$, in which the pixel p is included in a set which moves behind the set R_i. Then, we compute the number of maps $n_{R_i}(p)$ in which $M_{k-l,R_i}(p) = i$.

- In the last step, the result of a simplified temporal filter that is applied on each pixel p, is allocated to the set map S_k:

$$S_k(p) = i, \quad \text{if } n_{\{R_j; j<i\}}(p) < n_{R_i}(p). \tag{9}$$

5. Experimental results

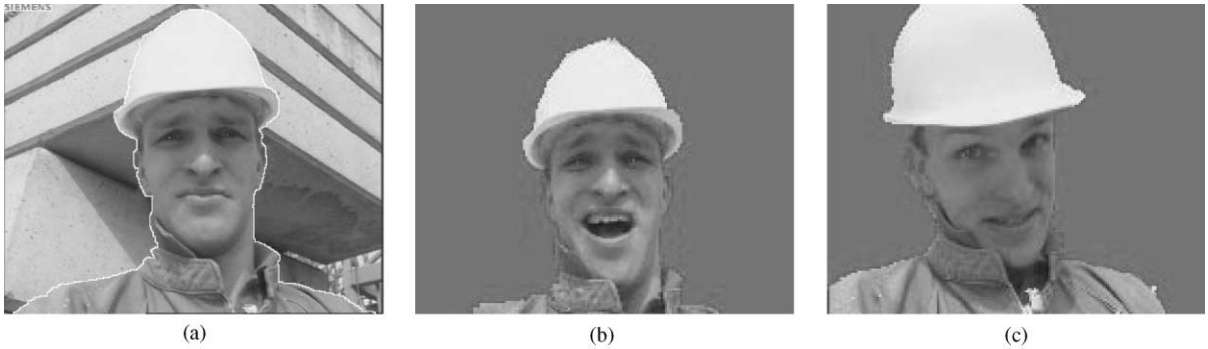
We now present the segmentation results obtained for real image sequences. We first show results on the MPEG-4 *Foreman* sequence. In this sequence the camera tries to fixate on the Foreman's head while he moves, performing a complex motion which, at the beginning, is nearly translational, and later involves an additional important rotational component. Hence, the background appears to move also.

The first user-guided segmentation is applied to the first pair of frames. Fig. 2 shows the partition result on the first frame (a). The tracking algorithm is performed in those parts of the sequence where the motion of objects is described by the translational model, without any user intervention. For the needs of SRG, we use the similarity criterion $\delta_{2C}(\cdot,\cdot)$, which involves colour information, because the intensity of the frames is not sufficient to distinguish the foreman from his background objects; λ_A has been set to 0.3. In Fig. 2 two other frames of the sequence are given. The whole sequence of segmented images is accessible at `http://www.csd.uoc.gr/~tziritas/animations/man_lays.gif`.

On the other hand, when Foreman's motion is not translational, the motion feature will be unreliable as a segmentation criterion. We therefore reduce the role of the motion feature, and use the first segmentation algorithm between two frames with the same similarity criterion, but with a large value for λ_A, implying that the segmentation is based primarily on the spatial color information. After each segmentation the boundary smoothing operation is performed in order to keep the boundary stable over time.

We also applied our method to the MPEG-4 *CoastGuard* sequence. 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, as shown in Fig. 1. The overall scheme can be represented by three layers, one for the background, one for the ship and one for the boat, in that depth order. 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 track. Object motions remain translational over the entire sequence. This allows the tracking of the first segmentation result over a large number of images and therefore decreases the number of needed applications of the first segmentation. The user-guided initialization was given, as shown in Fig. 1, on frame 58, all three layers being present in that scene.

The initial sets required for SRG's execution are obtained by the "conditional" dilation operation. In this manner, the mast and the hull of the ship retain their connectivity. The similarity criterion



(a) (b) (c)

Fig. 2. Results for the *Foreman* sequence.

Fig. 3. Results for the *CoastGuard* sequence.

we use for the segmentation is $\delta_{1I}(\cdot,\cdot)$ for the background set and the ship, which takes into account only motion information. For the neighboring pixels of the boat, we use the criterion $\delta_{2I}(\cdot,\cdot)$ with $\lambda_A = 0.3$, since the boat's intensity is uniform. This selection of criteria improves the segmentation result of frames in which the boat moves in front of the ship.

The post-processing of this sequence does not include the boundary smoothing method, for it may alter the shape of the objects, while the movement is approximately translational. Furthermore, when there is only background motion, this operation does not give the required boundary stability. Shape averaging, however, not only regularizes the object boundaries, but also improves the segmentation result of frames containing occlusions, providing a very good layered representation. Fig. 3 shows the tracking results on three frames of the sequence, where the shape averaging filter was used with $L_S = 12$. The whole sequence of segmented images is accessible at `http://www.csd.uoc.gr/~tziritas/animations/coast_lays.gif`.

6. Conclusions

We have described a segmentation method based on object motion throughout a sequence. Object motion was assumed to be translational in the plane. The estimation of motion parameters was obtained by a region matching technique. Other features such as the intensity or the color of image objects are used in order to obtain a better segmentation result. The segmentation algorithm is a seeded region growing algorithm, which was originally introduced for the segmentation of static images and extended here for the extraction and tracking of moving objects. For the needs of tracking, we introduce a layered representation of the image objects in order to be able to deal with sequences including multiple motions, moving background and occlusions. The order of layers is supplied by the user. Two post-processing procedures have been developed in order to improve the segmentation result.

The user guides the whole process, providing the initial object patterns, their layer ordering and parameter values for the segmentation criteria. Furthermore, the user decides how many times the boundary smoothing operation will be applied and how many previously derived segmentation results should be used by the shape averaging procedure.

The segmentation method can be extended to deal with motions that are not described by the simple translational model assumed herein. 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.

Acknowledgements

This work has been funded in part by the European ESPRIT NEMESIS (New Multimedia Services using Analysis Synthesis) and the Greek "MPEG-4 Authoring Tools" projects.

References

[1] R. Adams, L. Bischof, Seeded Region Growing, IEEE Trans. Pattern Anal. Mach. Intell. 16 (6) (June 1994) 641–647.

[2] G. Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-7 (July 1985) 384–401.

[3] Y. Altunbasak, E. Eren, M. Tekalp, Region-based parametric motion segmentation using color information, Graphical Models Image Process. 60 (January 1998) 13–23.

[4] C.K. Cheong, K. Aizawa, Structural motion segmentation based on probabilistic clustering, in: Proceedings of the IEEE International Conference on Image Processing, Vol. I, 1996, pp. 505–508.

[5] R. Koenen, MPEG-4 – Multimedia for our time, IEEE Spectrum 36 (2) (February 1999) 26–33.

[6] S.-M. Kruse, Scene segmentation from dense displacement vector fields using randomized Hough transform, Signal Processing: Image Communication 9 (1996) 29–41.

[7] A. Mitiche, P. Bouthemy, Computation and analysis of image motion: a synopsis of current problems and methods, Int. J. Comput. Vision 19 (1) (July 1996) 29–55.

[8] F. Moscheni, S. Bhattacharjee, Robust region merging for spatio-temporal segmentation, in: Proceedings of the IEEE International Conference on Image Processing, Vol. I, 1996, pp. 501–504.

[9] F. Moscheni, S. Bhattacharjee, M. Kunt, Spatiotemporal segmentation based on region merging, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-20 (September 1998) 897–915.

[10] J.-M. Odobez, P. Bouthemy, Direct incremental model-based image motion segmentation for video analysis, Signal Processing 66 (1998) 143–155.

[11] P. Salembier, L. Torres, F. Meyer, C. Gu, Region-based video coding using mathematical morphology, Proc. IEEE 83 (June 1995) 843–857.

[12] T. Sikora, The MPEG-4 video standard verification model, IEEE Trans. Circuits Systems Video Technol. 7 (February 1997) 19–31.

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

[14] J. Wang, E. Adelson, Representing moving images with layers, IEEE Trans. Image Process. IP-3 (5) (1994) 625–638.