A Video Object Generation Tool Allowing Friendly User Interaction

B. Marcotegui, F. Zanoguera
Ecole des Mines - CMM
77305 Fontainebleau
France

P. Correia, R. Rosa
IT - IST
1049-001 Lisboa
Portugal

F. Marqués
Univ. Politècnica
08034 Barcelona
Spain

R. Mech, M. Wollborn
Univ. of Hannover
30167 Hannover
Germany

Abstract

In this paper we describe an interactive video object segmentation tool developed in the framework of the ACTS-AC098 MOMUSYS project. The Video Object Generator with User Environment (VOGUE) combines three different sets of automatic and semi-automatic tools (spatial segmentation, object tracking and temporal segmentation) with general purpose tools for user interaction. The result is an integrated environment allowing the user-assisted segmentation of any sort of video sequences in a friendly and efficient manner.

1. Introduction

The emerging MPEG-4 standard aims at providing a platform for audiovisual data communications and storage. With respect to previous standards, MPEG-4 adds content-based functionalities such as object manipulation that give support to a broad spectrum of applications. The future MPEG-7 standard will address the multimedia content description, and thus will stress the content-based approach. The success of these standards will strongly depend on the existence of tools allowing the required content accessibility. In this context new demands for content creation have appeared.

The automatic extraction of video objects from a sequence is in general a very difficult task, specially if no *a priori* information is available. This is the case in the MPEG-4 and MPEG-7 framework due to the large variety of treated images. On the other hand, manually segmenting a sequence is a very hard and time consuming task.

This paper presents a Video Object Generation Tool with User Environment (VOGUE), where the user has the possibility to interact with the algorithms in order to obtain the desired partition. Due to the good performance of the automatic algorithms, user interaction is minimised. The combination of automatic and interactive tools leads to an integrated process that:

- allows the segmentation of any sort of image sequences.
- is a good trade-off between efficiency and robustness. The process is more efficient than manual segmentation and more robust than automatic segmentation.

Three main approaches for introducing user interaction in segmentation algorithms can be found in the literature:

Feature-based: The user is asked to select a set of pixels belonging to the different texture-homogeneous parts of the objects to be segmented. Based on the features of the selected pixels, the remaining pixels are classified into one of the object-types defined by the user [2]. This sort of interaction does not require a precise selection of the objects. However, connectivity aspects have to be taken into account after classification in order to obtain the final objects and further user interaction may be necessary. Object tracking can be carried out using the same classification process or by introducing motion information [12].

Contour-based: The user must roughly mark the position of the object boundaries. An automatic algorithm accommodates these rough boundaries to the object real ones [6]. This approach leads to smooth contours accurately representing the real object shape. Further interaction is possible by adjusting the position of some control points defining the object shape. Object tracking is usually performed by exploiting the motion information in the sequence.

Region-based: The user can interact with an underlying partition of the image to create the object shape. Usually, the user is allowed to merge regions from this initial partition [1] until the object shape is obtained. The image partition can be used as well in the object tracking process.

In VOGUE, we have adopted and extended the region-based interaction approach. The main extension consists of working with a morphological hierarchical segmentation technique. The morphological approach presents the advantage of allowing the introduction of rough markers by the user, which is a type of interaction very close to the one proposed by feature-based approaches while directly using connectivity concepts. Furthermore, it allows the definition of rough contours to obtain the object shape, enabling a type of interaction very close to the contour-based scheme. However, in this case internal and external rough contours of the object are necessary. Also, since the object shape is created by merging small regions obtained by local decisions, object contours are not ensured to be as smooth as in the contour-based approach. The hierarchical approach allows the user to deal with regions at different levels of resolution. An initial definition of the object can be created by selecting regions from a low resolution partition and further refinements can be included by merging or removing regions from fine resolution partitions. Finally, the region-based tracking process has been exploited to propose possible improvements of the final result to the user. Such improvements are presented as regions that the user will likely want to remove or add to the object when further interaction is requested.

Section 2 describes the graphical user interface that has been implemented as a platform for efficient combination of the algorithms. Section 3, 4.1 and 5 respectively describe the spatial segmentation algorithm, the object tracking algorithm and the temporal segmentation algorithm. They are complementary to each other and are supported by different types of user interaction. Finally section 6 presents some conclusions.

2. Graphical User Interface

The graphical user interface (GUI) encapsulates the automatic segmentation algorithms and provides the means for the user to interact with the overall segmentation process [3, 4]. It has been implemented in a PC-Windows environment, using JAVA. The chosen software platform ensures maximum flexibility in the development and makes possible future porting to other environments, such as UNIX.

The GUI supports the general interfacing requests of an application dealing with video, such as sequence selection, viewing with resize possibility or saving. More specific actions supported by the GUI are:

- open, save or visualise partitions;
- launching the automatic segmentation algorithms, with their specific user interaction possibilities;
- calling general purpose user interaction algorithms (such as split, merge, or object isolate/define);
- undo/redo functionalities for the different modules.

According to the selected operation mode, a context sensitive toolbar is displayed as an alternative to the respective menu options.

The implemented GUI supports the intuitive specification by the user of initial selections and constraints, as well as corrections to the results being produced by the different automatic algorithms.

Figure 1 shows the aspect of the GUI. A partition is displayed with a different colour for each region. At the same time, this colour information is superimposed to the luminance of the original image for better evaluation of contour precision.

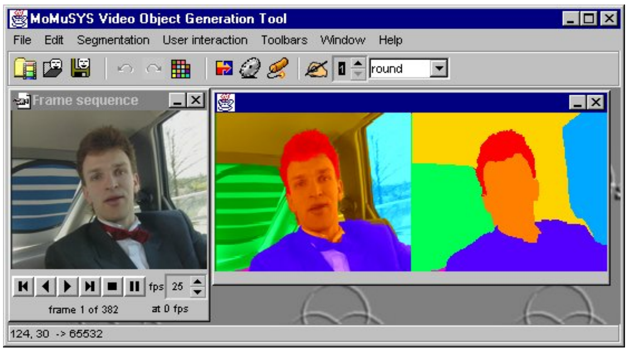

Figure 1: The Graphical User Interface.

3. Spatial segmentation

The interactive spatial segmentation algorithm permits to define the segmentation mask of the object of interest. It is based on a multi-scale segmentation scheme [11].

3.1. Multi-scale segmentation

A multi-scale segmentation consists of a family of partitions that represents an image at different resolution levels. The coarsest level considers the image as a whole (as only one region) and finer partitions are always included in coarser ones. This means that a finer level is obtained by re-segmentation of the regions of the previous level, or in other words, a contour present at a given level is also present in all the finer ones. The contours of the object should be present at a given level of resolution. Thus, the finest partition must be precise enough to fulfil this requirement.

We have implemented a morphological multi-scale segmentation whose finest resolution level contains all the regions of the watershed applied to the gradient image. Coarser partitions are obtained by merging neighbouring regions of the watershed. These fusions are ranked with a psycho-visual criterion: the volume extinction value [13]. This criterion is a combination of size and contrast and gives a good estimation of the visual significance of a region. The information of the whole multi-scale segmentation is stored as a minimum spanning tree which is calculated simultaneously to the flooding process [14]. Thus, the complexity of the algorithm is equivalent to a single watershed, but instead of having a single partition a whole family of partitions is obtained.

3.2. User interaction

The user can obtain the desired segmentation from the multi-scale segmentation presented in the previous paragraph. Several tools are offered for interaction.

Selection of a resolution level: Given a specified number of regions (N) the algorithm provides the N most significant regions. With a sliding bar the user can select a different number of regions and a new partition is displayed without perceptible delay. This type of interaction is equivalent to requesting an automatic segmentation into N regions.

This is a global action, as the same resolution level is chosen for the whole image. However, the user may be interested in a partition presenting different resolution in different zones of the image (according to a semantic criterion). In this case, two other tools are available: refining and coarsening of a region.

Refining a region: From a given level of resolution, the user can obtain a re-segmentation of a selected region just by clicking on it.

Coarsening a region: From a given resolution level, the user can obtain a coarser segmentation of a selected region just by clicking on it. The algorithm merges the selected region with its most similar neighbouring region(s).

Marker drawing: Segmenting with markers constitutes the classical morphological method for segmentation. By marker we mean a binary set included in the object of interest, its exact location or shape bearing little importance. The strategies for finding good markers are diverse and problem dependent. Nevertheless, in an interactive approach the user can roughly draw the markers for the objects of interest. Figure 2 shows an example of this type of interaction.

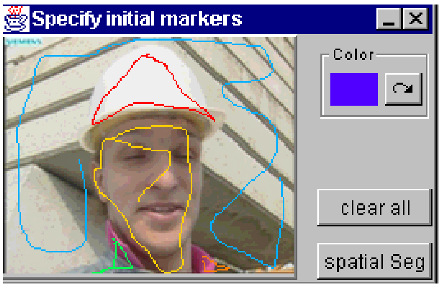

Figure 2: Interaction through marker drawing.

Contour adjustment: When the contours obtained through the previous types of interaction are not completely satisfactory, the user has the possibility to drag them to a better position. The new contours are chosen among the ones appearing in the highest resolution level of the multi-scale segmentation.

Further user interaction: In the cases when the interaction with the multi-scale segmentation does not allow to reach the desired object partition, other tools are provided to give the necessary flexibility. The *merge* tool allows to merge any number of regions of the image by simply clicking on it. The *refine* tool provides pixel level contour adjustment in a brush like manner. The *split* tool allows to divide a region into two by drawing a line. The *separate* tool separates a region into its connected components. Finally, the *define object* tool allows to select a region as being the object of interest, and the background is automatically merged into one region.

4. Object tracking

Once the object partition is available for an initial image, it can be automatically extended to the following images of the sequence. For this purpose a tracking algorithm has been implemented [8]. It is based on the projection of a partition allowing the introduction of new regions, followed by a decision on whether the new regions belong or not to the mask.

4.1. The automatic tracking algorithm

The object tracking approach that is used relies on the concept of partition projection [9], that consists of accommodating the previous partition to the information in the current frame. This algorithm requires spatial homogeneity of the regions to be projected. Since this requirement may not be fulfilled by the object partition, the latter is re-segmented using the spatial segmentation algorithm of section 3. The partition obtained is called *texture partition* (of the previous frame) and it guarantees the spatial homogeneity of each region.

The texture partition of the previous image is projected into the current frame to obtain the texture partition at the current image. The projection of the texture partition is carried out in two steps. Firstly, the motion between the previous and current images is estimated using a backward block matching algorithm. The previous texture partition is motion compensated obtaining a rough estimation of the regions in the current frame, called motion compensated markers. In a second step motion compensated markers are accommodated to the boundaries of the current image [8]. This is performed by means of a fitting process between the motion compensated regions and the regions obtained from a *fine partition*. This partition, that represents the colour boundaries of the current image, is said to be fine since it contains a large number of regions (typically, more than 1000 regions for a QCIF image).

New regions can be detected during the fitting process or by an additional step of new region extraction. Detected new regions are analysed and an automatic decision is made on whether they belong or not to the object being tracked.

4.2. User interaction

The result of the automatic tracking is displayed to the user, who has the possibility to stop the execution and ask for refinements of the object mask, adding or removing parts (regions). The user corrections will then be used by the automatic algorithm to improve its subsequent performance.

Object refinement based on proposed regions: When the user stops the tracking algorithm, the set of new regions that have been automatically detected and analysed in the current image are proposed for refinement of the current object definition. The user can click on any of these regions and the decision taken by the automatic algorithm is reversed (belonging or not belonging to the object). Figure 3 illustrates this type of interaction. On the left area of the working window the current mask is displayed, while the right part of the window shows the regions proposed for correction. The user can toggle the decision taken by the automatic algorithm by clicking on them.

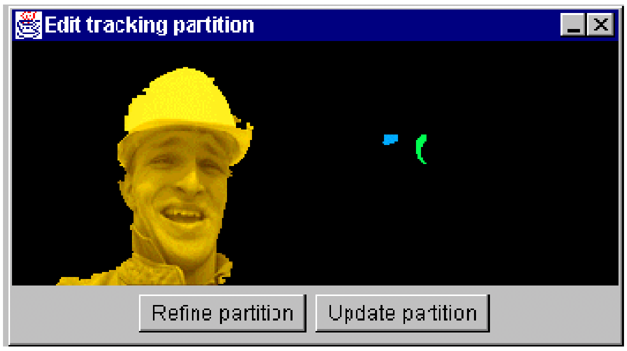

Figure 3: Region proposal during tracking interaction.

Direct object refinement: If the set of proposed regions does not allow the required corrections, direct mask re-definition is enabled. For this purpose, the interface offers the same type of user interaction as described in section 3.2.

5. Temporal segmentation

The temporal segmentation is useful when the user is interested in moving objects. The video objects are generated by evaluating the differences between consecutive frames, taking into account the motion of the objects. While in the completely automatic mode each moving object in the scene is segmented, the user can also interact with the segmentation process.

5.1. Automatic temporal segmentation

The temporal segmentation algorithm is based on a change detection followed by a motion analysis [10]. It consists of the following steps:

Camera motion estimation and compensation: assuming that the background of the scene is a rigid plane, the camera motion between two successive frames of the sequence is estimated using an eight parameter mapping model, called perspective model. Using this model, the camera motion is compensated for the complete frame.

Scene-cut detection: based on the evaluation of the mean absolute difference between the camera motion compensated previous frame and the current frame, a possible scene cut is detected. In the case of a scene cut, all the parameters of the segmentation algorithm (which in general adapt automatically to the properties of the image sequence) are reset to their initial values.

Estimation of change detection mask: after calculation of the frame difference between two successive frames, an initial change detection mask is calculated by thresholding the frame differences. Then, the final change detection mask is generated by a relaxation of the initial mask, evaluating the local neighbourhood of each pel. Furthermore, a memory for the change detection mask is used in order to improve the time coherence of the estimated object masks.

Uncovered background elimination: afterwards, uncovered background regions are detected and eliminated from the change detection mask. For this purpose, a displacement vector field is evaluated considering the change detection mask. The displacement vector field is estimated using hierarchical block matching algorithm.

Contour adaptation: finally, the contours of the estimated object masks are adapted to luminance edges of the current frame in order to get a more accurate object boundary.

This algorithm has been adopted as an informative annex of the ISO/MPEG-4 standard [7] and is part of the COST 211 Analysis Model [5].

5.2. User interaction

The temporal segmentation algorithm allows different types of user interaction, where the user can introduce external knowledge about the scene.

The user can provide a segmentation mask for the first frame. In this case, this mask is used to initialize the temporal segmentation algorithm, i.e. statistical parameters are measured within object and background regions, the memory is set to this initial mask, and the mask is considered for estimating the camera motion between the next two frames.

The user can roughly determine the image regions objects of interest are expected to appear. The temporal segmentation is then performed only within these image regions. If available, an image showing only the background can be provided. In this case all frames are temporally segmented against this background image, improving the quality of the segmented objects.

Also, all the parameters of the algorithm can be set, saved, and loaded. If nothing is specified, default values are used for the parameters. The results of the temporal segmentation may be interactively modified at any execution step using the tools described in section 3.2 The results of the temporal segmentation can as well be used as input masks for the tracking algorithm, described in section 4.1.

6. Conclusion

A Video Object Generator with User Environment has been presented, including three main segmentation algorithms: spatial segmentation, object tracking and temporal segmentation. The algorithms have been integrated into a common graphical user interface which also provides support for friendly user interaction. Interaction can take place in different forms at any step of the segmentation process, which makes the system highly flexible. This flexibility, combined with the good automatic performance of the algorithms, allows to obtain very good results with little effort for any type of video sequences. Finally, the whole process is very intuitive and does not require any expert knowledge from the user, although if this knowledge exists it can be used to obtain improved results.

References

[1] R. Castagno. “Interactive object extraction from video sequences for multimedia application based on multiple features”, In *Noblesse Workshop on Non-Linear Model Based Image Analysis*, Glasgow, July 1998.

[2] E. Chalom and M. Bove. “Segmentation of an image sequence using multidimensional image attributes”, In *Proceedings of the IEEE International Conference on Image Processing, ICIP-96*, Lausanne, September 1996.

[3] P. Correia and F. Pereira. “User interaction in content-based video coding and indexing”, In *EUSIPCO-98*, Rhodes, Greece, Sept 1998.

[4] P. Correia and F. Pereira. “The role of analysis in content-based video coding and indexing”, *Signal Processing-Special Issue on Video Sequence Segmentation for Content-Based Processing and Manipulation*, Vol 66, No. 2, pp.203-217, April 1998.

[5] M. Gabbouj, G. Morrison, F. Alaya-Cheikh and R. Mech. “Redundancy Reduction Techniques and Content Analysis for Multimedia Services - the European COST 211quat Action”, *Proc. Workshop on Image Analysis for Multimedia Interactive Services 1999 (WIAMIS’99)*, Berlin, May/June, 1999.

[6] Ch. Gu and M.Ch. Lee. “Semiautomatic segmentation and tracking of semantic video objects”, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 8, No. 5, 1998, pp.572-584.

[7] ISO/IEC JTC1/SC29/WG11 Doc. N2502. “Information Technology - Generic Coding of Audio-visual Objects, Part 2: Visual, ISO/IEC 14496-2”, *Final Draft of International Standard*, October 1998.

[8] F. Marqués and J. Llach. “Tracking of generic objects for video object generation”, In *IEEE International Conference on Image Processing*, Chicago, USA, Oct 1998.

[9] F. Marqués, M. Pardàs, and P. Salembier. *Video Coding: The second generation approach*, chapter Coding-oriented segmentation of video sequences, pages 79–124. L. Torres and M. Kunt (Eds). Kluwer Academic Publishers, 1996.

[10] R. Mech and M. Wollborn. “A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera”, *Signal Processing*, 66:203–217, 1998.

[11] F. Meyer. “Morphological multiscale and interactive segmentation”, In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Antalya. Turkey, June 1999.

[12] N.E. O’Connor and S. Marlow. “Supervised semantic object segmentation and tracking via EM-based estimation of mixture density parameters”, In *Noblesse Workshop on Non-Linear Model Based Image Analysis*, Glasgow, July 1998.

[13] C. Vachier and F. Meyer. “Extinction value: a new measure of persistence”, In *IEEE Workshop on Non-Linear Signal and Image Processing*, 1995.

[14] F. Zanoguera, B. Marcotegui, and F. Meyer. “A tool-box for interactive image segmentation based on nested partions”, In *IEEE International Conference on Image Processing (Submitted)*, Kobe, Japan, Oct 1999.