VOGUE: The MoMuSys Video Object Generator with User Environment

Marcotegui B[1], Correia P[2], Marqués F[3], Mech R[4], Rosa R[2], Wollborn M[4] and Zanoguera F[1]

[1] Ecole des Mines-CMM, 77300 Fontainebleau, France
[2] Instituto Superior Técnico/IT, 1049-001 Lisboa, Portugal
[3] Universitat Politècnica de Catalunya, Dpto TSC, 08034 Barcelona, Spain
[4] University of Hannover, Appelstr. 9A, 30167 Hannover, Germany

Beatriz.Marcotegui@cmm.ensmp.fr

Abstract

The video object manipulation context set by the MPEG-4 standard has raised a need for respective content creation tools. This demand is expected to grow with the availability of MPEG-4 applications and devices in the marketplace as well as with the development of future content-based standards like MPEG-7. In this paper a Video Object Generator with User Environment (VOGUE) is described, comprising spatial and temporal segmentation as well as object tracking algorithms. The individual algorithms have been integrated into a common graphical user interface that allows to combine them and also to perform the necessary tasks for user-assisted video object generation. The work described has been carried out in the framework of the European ACTS-MoMuSys project.

I. Introduction

The success of the content-based audiovisual multimedia coding and representation standards MPEG-4 [1] and MPEG-7 [2] will strongly depend on the availability of multimedia content. This requires, in the video area, the development of efficient algorithms for generating video objects of high spatial quality and temporal coherency with as little manual work as possible.

This paper presents a tool for this task: a Video Object Generator with User Environment (VOGUE) which is currently being developed in the framework of the European ACTS-MoMuSys project [3]. It includes a set of algorithms for spatial and temporal segmentation and for object tracking. They are integrated in a common graphical user interface that allows the user to easily interact with the algorithms. This approach is a good trade-off between efficiency and robustness – the process is more efficient than pure manual and more robust than fully automatic segmentation. Furthermore, the user interaction mechanisms are well adapted to the segmentation algorithms so that the proposed correction/refinement actions have high efficiency at a low cost for the user.

Section II describes the graphical user interface that has been implemented as a platform for efficient combination of the algorithms, and also as the support to user interaction. Sections III, IV and V respectively describe the spatial segmentation algorithm, the object tracking algorithm and the temporal segmentation algorithm. They are complementary to each other and allow different types of user interaction. Finally section VI gives some conclusions.

II. Graphical User Interface

The graphical user interface (GUI) encapsulates the segmentation algorithms, and provides the means for the user to interact with the overall segmentation process [4][5]. It has been implemented on 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. Also, a minimal effort is needed to implement an applet version for running in an intranet/internet browser.

The GUI supports the general interfacing requests of an application dealing with video, such as sequence selection, viewing with resize possibility, or saving, open or visualize partitions, launching the automatic segmentation algorithms, performing user interaction, etc.

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

Figure 1 shows the aspect of the interface. A partition is displayed with a different color for each region. At the same time, this color information is superimposed to the luminance of the original image for better evaluation of contour precision. Typical results of user interaction are shown in figure 2.

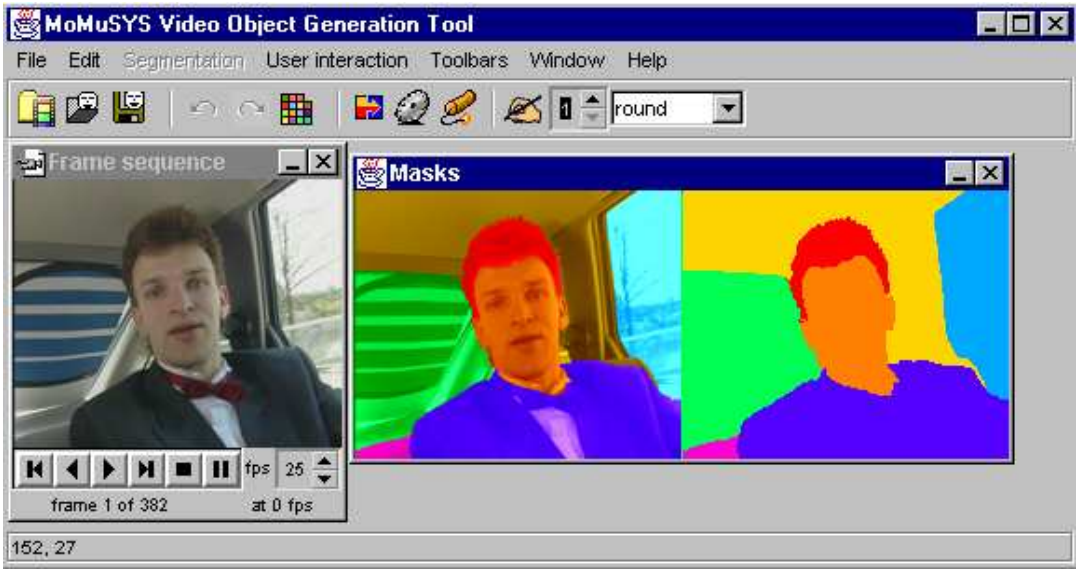

Figure 1: The Graphical User Interface

(a) (b)

Figure 2: Typical results of the user interaction where a partition with a foreground and a background object has been defined by merging regions of the initial partition in Figure 1. In (a) the creation of the foreground object is presented, whereas (b) shows the merging of all the regions defining the background.

Figure 3 shows a diagram representing the interactions between the three algorithms supported by the user interface. The spatial segmentation module provides the tracking with the initial object partition and with a fine partition. The user can stop the tracking at any time and call the spatial segmentation module in order to interactively correct the results. The spatial segmentation can as well optionally provide the temporal segmentation module with an initial object partition, which will serve to skip the initialisation phase. In the same way as for the tracking module, there is the possibility to stop the segmentation and call the spatial module to correct the results. Finally, it is possible to produce the initial object partition needed by the tracking module through the detection of moving objects performed by the temporal segmentation module.

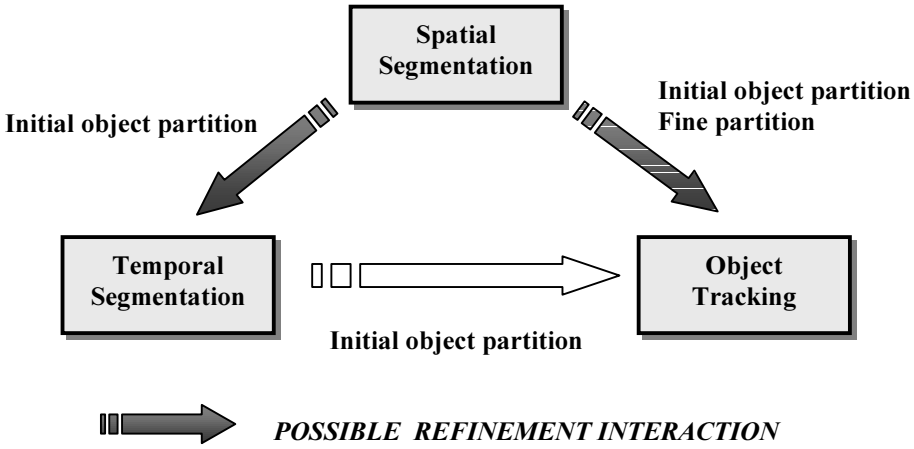

Figure 3: Interactions between the algorithms supported by the Graphical User Interface

III. Spatial segmentation

The interactive spatial segmentation algorithm permits to define the initial partition of the object of interest. It is based on a multi-scale segmentation scheme [6] as described in the following.

A multi-scale segmentation yields 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-segmenting the regions of the previous level.

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 neighboring regions of the watershed. These fusions are ranked with a psycho-visual criterion: the volume extinction value. 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 in a tree which is obtained during the flooding process [7]. Thus, the complexity of the algorithm is equivalent to a single watershed, but instead of having a simple partition we obtain a whole family of partitions.

The user can obtain the desired partition from the multi-scale segmentation presented in the previous paragraph. Several tools are offered for interaction: selection of a number of regions, re-segmenting a selected region, coarsening a selected region (merging it with its most similar neighbors) and drawing a marker for each region.

IV. Object tracking

Once the object partition is available for an initial image, it can be automatically tracked through the following images of the sequence. The object tracking approach [8] 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 III. 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: motion compensation and a fitting process that accommodates the motion compensated contours to the color boundaries.

New regions in the image can be detected during the fitting process or by an additional step of new region extraction. Detected new regions are analyzed and an automatic decision is made on whether they belong or not to the object being tracked. 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 partition, adding or removing parts (regions).

V. Temporal segmentation

The temporal segmentation is useful when the user is interested in moving objects. While in the completely automatic mode each moving object in the scene is segmented, the user can also interact with the segmentation by defining an initial object partition, specifying the area where the object should be detected or by calibrating the process with a certain background image.

The temporal segmentation algorithm [10] is based on a change detection followed by a motion analysis. It consists of the following steps: (1) camera motion estimation and compensation, assuming that the background of the scene is a rigid plane, (2) scene cut detection based on the evaluation of the mean square difference between the camera motion compensated previous frame and the current frame, (3) estimation of a change detection mask by thresholding the difference image followed by a relaxation, (4) elimination of uncovered background by evaluating an estimated displacement vector field, and finally, (5) adaptation of the mask to luminance edges of the current frame.

This algorithm has been adopted as an informative annex of the ISO/MPEG-4 standard [1] and is part of the COST 211 Analysis Model [11].

VI. Conclusions

A user-assisted tool for video object generation has been presented, which is currently being developed within the European ACTS-MoMuSys project. This tool comprises algorithms for spatial and temporal segmentation and for object tracking, which are integrated by means of a common graphical user interface. Their combination allows the efficient and robust generation of high quality MPEG-4 content in the form of video objects. Moreover, the user has the possibility to interact with the whole segmentation process. This will make possible to segment nearly any sort of video material, and respective MPEG-4 content can be generated for a wide variety of applications.

References:

[1] 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.

[2] MPEG Requirements Group, „*MPEG-7 Context and Objectives*“, ISO/IEC JTC1/SC29/WG11 Doc. N1920, October 1997.

[3] Project overview available on www.tnt.uni-hannover.de/project/eu/momusys/overview.html.

[4] P. Correia and F. Pereira, „*User interaction in content-based video coding and indexing*“, EUSIPCO-98, Rhodes, Greece, September 1998.

[5] 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, 66(2):125-142, April 1998.

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

[7] F. Zanoguera, B. Marcotegui, and F. Meyer., „*An interactive colour image segmentation system*“, Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'99, Berlin, Germany, May/ June 1999.

[8] F. Marqués and J. Llach, „*Tracking of generic objects for video object generation*“, IEEE International Conference on Image Processing, Chicago, USA, October 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(2):203-217, 1998.

[11] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, „*Image sequence analysis for emerging interactive multimedia services* - The European COST 211 framework“, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8(No. 7):802-813, November 1998.