A Video Segmentation Toolbox with Support for User Interaction

João Ascenso, Paulo Correia, Fernando Pereira
Instituto Superior Técnico, Instituto de Telecomunicações, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Abstract

Video segmentation assumes an important role in many object-based video applications, such as those enabled by the recent MPEG-4 standard for coding audiovisual objects, or by the upcoming MPEG-7 standard for multimedia content description.

This paper presents a video segmentation toolbox, giving emphasis to the graphical user interface and to the support for user interaction. Two types of user interaction tools have been included: initial user interaction and segmentation refinement interaction. The first is mainly used to introduce basic semantic information to guide the segmentation, e.g. the number of objects, while the second allows correcting deviations of the automatically produced segmentation results regarding the desired ones.

In this paper, the toolbox architecture is presented and after the user interaction segmentation tools are described along with several segmentation examples illustrating the relevance of the user interaction role.

I. INTRODUCTION

Many applications exist where the identification of video objects, i.e. video segmentation, assumes a crucial importance. For some cases, the segmentation information is generated at the time of content production, such as when using chroma-keying techniques, as it is often done in producing news or weather programs for TV broadcasting. For other cases, the video segmentation has to be computed from natural video sequences directly captured by video cameras.

One of the objectives of the recent MPEG-4 standard is to specify an efficient representation for arbitrarily shaped 2D video objects. Video objects are here defined as semantic objects of interest to the end user, such as a person, a boat, some text, etc. In MPEG-4, a segmentation mask represents the shape of an object at each time instant of the audiovisual sequence. However, MPEG-4 does not standardize the way to extract the shape mask of an object for a natural video sequence. Efficient and accurate solutions are needed to segment objects from each image, thereby constructing the binary shapes masks required for MPEG-4 encoding purposes. The MPEG-7 standard is also a good candidate to make use of video objects. MPEG-7 descriptions can be very powerful and fine-grained as they can refer to specific entities of the scene, such as objects. An example of the usage of segmentation information for description purposes is the creation of a hierarchical structure describing relationships among objects in a scene, very useful for retrieval purposes.

The automatic segmentation of objects from a video sequence is a complex task, especially if only a limited amount of *a priori* information about the expected low-level object features (such as shape, color and motion) is available [Cor98a]. In fact, in many cases, fully automatic segmentation tools do not provide acceptable results, namely when dealing with complex video scenes, notably because these tools cannot express the semantic meaning associated to the objects.

The combination of automatic and interactive tools is a powerful way to achieve the desired segmentation results, independently of the complexity of the video sequence being segmented [Cor98b, Mar99]. User interaction, which should be limited to the minimum extend possible, allows the user to guide the segmentation process and to correct results that are not according to the targeted goals. Although this type of interaction is very limited for real-time applications, it can be much more sophisticated for off-line content creation, such as for movies, documentaries, games, etc. Not taking benefit of this type of tools, would bring a big and useless additional burden to these applications.

The video segmentation toolbox presented in this paper includes three automatic segmentation algorithms and the means for the user to interact with their execution. These automatic algorithms perform spatial segmentation, (temporal) object tracking, and temporal segmentation. The toolbox components are integrated in a graphical user interface (GUI) including specific viewing and user interaction modules, associated with each automatic algorithm.

This paper describes the toolbox architecture and the functionality of each of the three segmentation modules implemented, with special emphasis on the user interaction aspects.

II. ARCHITECTURE

The segmentation toolbox developed is based on the ideas presented in [Cor98c]. Its architecture is presented in Figure 1, showing the dependencies between the automatic segmentation algorithms and the user interaction modules (marked in gray).

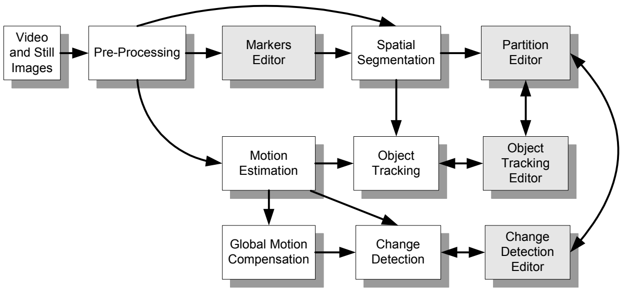


Figure 1 – Overview of the toolbox architecture

The toolbox can deal both with still image and video sequence segmentation. The segmentation process starts with a pre-processing module, responsible for transforming the input images into the most suitable format for subsequent processing. A typical example is image filtering to remove non-relevant details, thus simplifying the images to be processed.

A possible segmentation solution involves the spatial segmentation of a (set of) key image(s), usually the first of each shot, and the subsequent tracking of the objects identified in that initial spatial segmentation partition. Object tracking uses motion estimation information to keep the coherency of the partition labels along the temporal dimension. Alternatively, a temporal segmentation based on the detection of changes between consecutive images, eventually after global motion compensation, can be performed to directly identify the moving objects in a sequence. More sophisticated temporal segmentation, e.g. based on motion segmentation, may also be implemented.

User interaction can be used during the segmentation process, for instance to specify initial constraints to the spatial segmentation by marker drawing, or to refine the segmentation results as they are produced.

The initial user interaction is implemented using a *markers editor*, allowing to specify the number of objects of interest and a relevant subset of their pixels. The user interaction refinement of segmentation results supports such operations as merge, split, or refine by drawing. Referring to Figure 1, user interaction refinement can be used to correct the outputs of the spatial segmentation, using the *partition editor*, of the object tracking, using the *object tracking editor*, and of the temporal segmentation, using the *change detection editor*.

III. THE GRAPHICAL USER INTERFACE

All the toolbox components are integrated in a GUI, which has been developed in a PC-Windows environment, using Java to ensure maximum flexibility and portability for other environments.

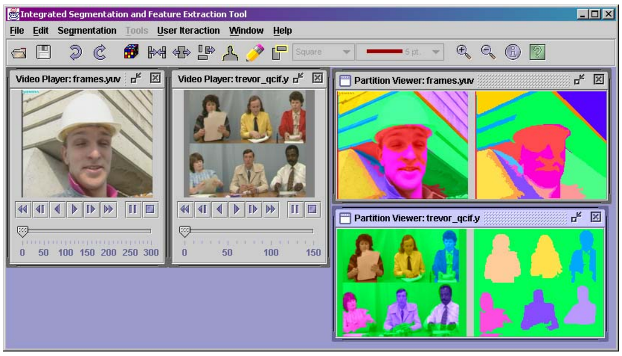

Integrated Segmentation and Feature Extraction Tool
File Edit Segmentation Tools User Iteraction Window Help
Video Player: frames.yuv
Video Player: trevor_qcif.y
Partition Viewer: frames.yuv
Partition Viewer: trevor_qcif.y
0 50 100 150 200 250 300
0 50 100 150

Figure 2 – Main window of the graphical user interface

The GUI is also responsible for other functionalities intended to facilitate the work of the user such as:

- Management of video sequence files (open and save), directories, and temporary files.
- Execution of the automatic algorithms with the corresponding settings and specific user input, if available.
- Undo/redo functionalities for the various interactive tools.
- Zoom of the segmentation results for detailed analysis.

The main window of the GUI is shown in Figure 2, where two segmentation partitions are displayed.

A. *The Menu Options*

The GUI functionalities are grouped into seven **main menus** – see Figure 3.

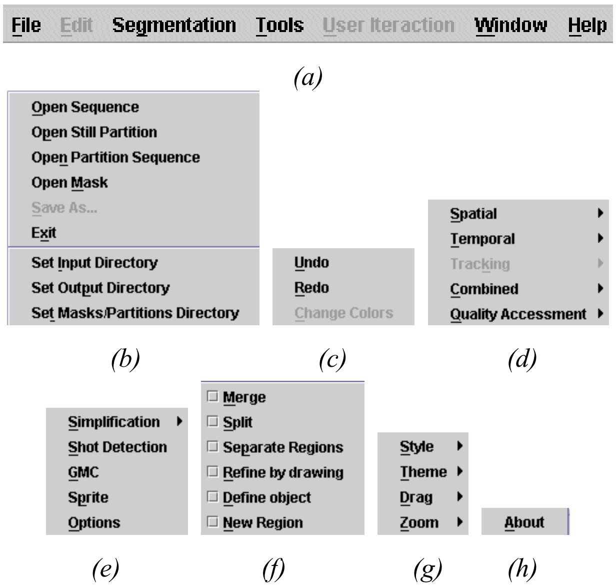


Figure 3 – Main menu of the segmentation toolbox (a), and the corresponding available options – (b) to (h)

The ***File* menu** groups the operations dealing with files and directories. This menu supports the options to open sequences, masks, and partitions (a mask is understood as a binary partition), and to save segmentation results. It is also possible to define the default directories from which sequences and partitions are read from disk, as well as the default directory to store segmentation results. Finally, there is an option to exit from the program.

The ***Edit* menu** includes the options to undo or redo operations (namely, user interaction operations), as well as the possibility to change the colors being used to display a partition.

The ***Segmentation* menu** includes the options allowing the selection of the type of automatic segmentation to perform, i.e. spatial, temporal or tracking, and the corresponding parameters configuration. Additionally for the spatial segmentation submenu the user can choose to specify the initial user interaction. Also combinations of the various segmentation algorithms may be selected, and finally, some segmentation quality assessment algorithms are included in this menu.

The ***Tools* menu** includes various options that can be used standalone or as part of the segmentation process. One of these options is the selection of the type of initial simplification to perform on the images of the sequence being analyzed. Other options are the detection of shot transitions, the estimation and compensation of global motion, and the creation of sprites. Finally, the adjustment of the configuration parameters for the various algorithms is also possible.

The ***User Interaction* menu** includes the options available for user interaction refinement. In particular, the available options allow the merging and splitting of regions, the separation of a component (separate region) of an object, the refinement of object contours by hand drawing, the definition of an object mask, and the definition of new regions.

The ***Window* menu** includes some window configuration options (style, theme and drag properties), as well as the zoom option.

The ***Help* menu** provides information about the software.

The options most frequently used can also be accessed through the toolbar – see Figure 4. The buttons correspond to the open sequence, save mask, undo/redo, change colors, operations for interactive spatial refinement and for zooming. The toolbar is context sensitive, this means the various buttons are enabled or disabled depending on the current stage of the processing.

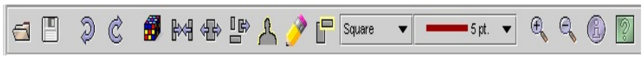


Figure 4 - Toolbar showing the available options

B. *Viewing Sequences and Partitions*

In order to carry out the segmentation of a video sequence, the first operation to perform is the selection of the desired video sequence. Once the selection is done, a video sequence viewer is opened and the first image of the sequence is displayed – see Figure 5. This viewer allows the user to play the sequence, either continuously, image by image, or using the slide bar. The direction in which the sequence is played, forward or reverse, can be controlled, and there are buttons to allow pausing the play, as well as to go to the first or last images of the sequence.

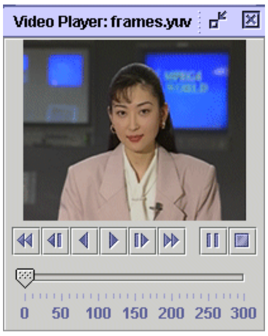


Figure 5 – Video sequence viewer

Making such a video sequence viewer available in a segmentation environment is very important, as it offers the user the possibility to become more familiar with the content of the sequence, thus allowing a correct decision on the objects to be segmented and the most appropriate segmentation strategy to be followed.

Another type of viewer is provided for viewing video partition sequences – see Figure 6. This partition sequence viewer is displayed once the user selects the *open partition sequence* option in the *File* menu, and it is used to display previously obtained segmentation results for a given sequence together with the corresponding original video. This viewer allows the continuously play or controlled browsing of the partition sequence, and supports the refinement of the partition for any set of time instances. The user can invoke the automatic tracking algorithm at any time instant.

It is also this viewer that allows the user to start and stop the execution of the automatic segmentation algorithms dealing with time (i.e. object tracking and temporal segmentation). Is also possible to visualize the segmentation results, and improve them with user input by selecting the play sequence or refine regions buttons.

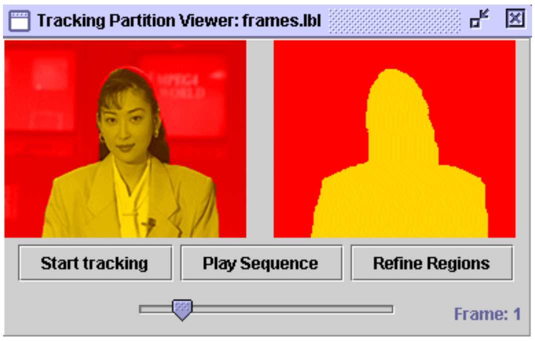
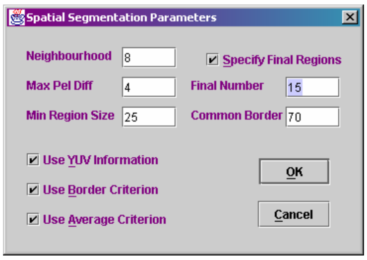


Figure 6 – Partition sequence viewer

IV. INTERACTIVE SPATIAL SEGMENTATION

Spatial segmentation processes each image isolated, whether or not it belongs to a video sequence. Therefore, no temporal consistency of the segmentation partitions produced can be ensured if only this type of technique is applied.

Two automatic algorithms are included for spatial segmentation purposes. The first is based on region merging and does not receive any initial user interaction input: it runs automatically once an input image is specified. To customize the results of the automatic spatial segmentation tool the user can specify a set of input parameters to the algorithm – see the corresponding dialog box in Figure 7.


Spatial Segmentation Parameters

Neighbourhood 8
☑ Specify Final Regions
Max Pel Diff 4
Final Number 15
Min Region Size 25
Common Border 70

☑ Use YUV Information
☑ Use Border Criterion
☑ Use Average Criterion

Figure 7 – Spatial segmentation parameters dialog box

For instance, the user can specify the number of final regions to be obtained or let that decision to the algorithm, he can decide whether or not to use color information, the minimum region size, the criteria to be used for merging regions and some other algorithm specific parameters.

The second automatic spatial segmentation algorithm available is a region-growing algorithm that creates an image partition constrained to a set of markers provided by the user.

In terms of user interaction support, the region-growing algorithm requires the existence of a markers editor, and both algorithms can benefit from the usage of a partition editor, to allow correcting the segmentation results. The functionality of the two user interaction modules developed for interactive spatial segmentation is described below.

A. *Markers Editor*

The markers editor implements *initial user interaction* with the spatial segmentation. It allows the user to input information to guide the segmentation process and consequently improve its performance.

Using the markers editor, the user can specify how many and which are the objects to be segmented – see example in Figure 8. A marker consists in a set of pixel locations that belong to an object of interest.

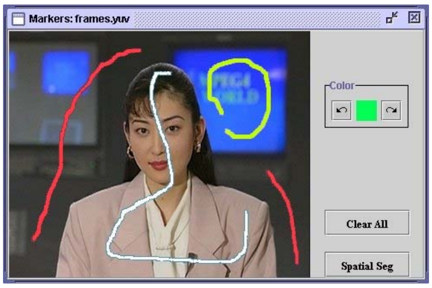


Figure 8 – Interaction using the markers editor

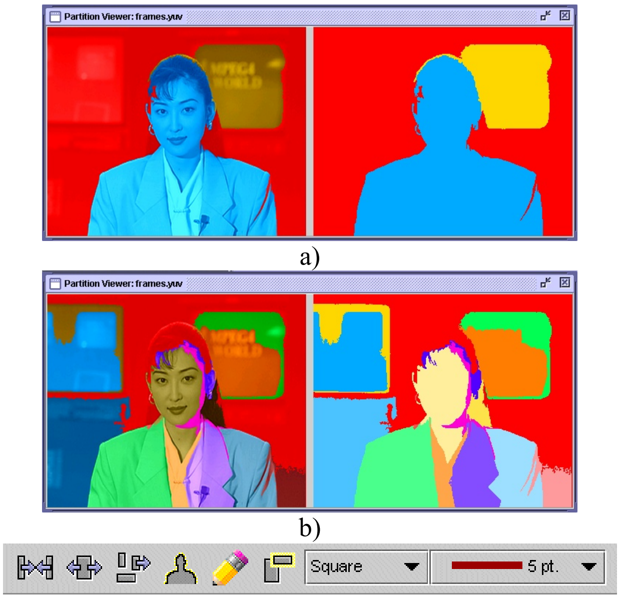
The editor supports the drawing of markers over the original image, each marker being attributed a different color. The number of objects corresponds to the number of markers drawn, and a single marker can have several connected components, eventually leading to an object composed of several spatially unconnected regions.

As an example of the region growing spatial segmentation result, the segmentation partition obtained from the markers specified in Figure 8 is included in Figure 9 a). For comparison purposes, results obtained without initial user interaction, using the region merging spatial segmentation algorithm, are included in Figure 9 b). This example shows that a simple input from the user allows the automatic algorithms to take some decisions (e.g. to merge some regions) significantly improving the segmentation results.

B. *Partition Editor*

The partition editor implements the *user interaction refinement* for the spatial and temporal segmentation and object tracking.

This editor displays a partition by assigning different colors to the various objects, side by side with the original luminance overlaid with the partition colors, to improve perception of how far is the current partition from the desired one – see Figure 9 a) and b). The partition editor also provides the means for the user to improve the partition – a toolbar with the available user refinement options is shown in Figure 9 c).



a)

b)

c)

Figure 9 – Interaction using the partition editor: a) Automatic spatial segmentation from markers; b) Automatic spatial segmentation without initial interaction. c) Toolbar showing the available user interaction possibilities, from left to right: *merge*, *split*, *separate*, *define object*, *refine by hand drawing*, *new region*, *pen shape*, and *pen size*

The six types of user interaction functionalities available in the partition editor are:

- **Merge** – Merges together the two regions in which the user clicks (giving them the same label).
- **Split** – Separates a region in two according to a dividing line drawn by the user.
- **Separate Regions** – When an object is composed by several connected components, a new label is attributed to the region selected by the user.
- **Refine by Drawing** – Contours can be refined using a drawing tool that allows extending a previously existing label.
- **Define Object** – The selected region is defined as the object of interest, and all the other regions are merged together (into one background region).
- **New Region** – Allows the user to create a new region, by drawing it. This corresponds to performing hand-segmentation, and its usage should be minimized (e.g. specifying initial markers, and using the other partition refinement functionalities).

These functionalities are also available through the user interaction menu, in addition to the toolbar.

V. INTERACTIVE OBJECT TRACKING

The automatic object tracking algorithm takes a partition corresponding to the previous time instant and computes a new partition for the present time instant, while maintaining the temporal coherency of the object labels.

For this, it estimates the motion of the objects and uses an affine motion model to project them into the current time instant. The contours of the projection objects are then refined using low-level spatial segmentation information computed for the present time instant.

Since results may deviate from the desired ones, an *object tracking editor* is available for the user to supervise the algorithm results and introduce the appropriate corrections.

A. *Object Tracking Editor*

The main purpose of the object tracking editor is to help the automatic tracking algorithm at more critical time instances, like scene cuts, occlusion of objects, changes in lighting. The main functionalities foreseen to overcome these problems are: definition of new objects and adjustment of objects contours.

The partition resulting from the automatic object tracking is displayed to the user, as the algorithm proceeds. At any moment, the user may stop the execution and ask to refine the object sequence partition. By using the slide bar control, the user may select the time instant for which the available partition is to be refined. The subsequent images can be reprocessed taking the user supplied information into account. Figure 10 illustrates this type of interaction.

When the user asks for refinements, a viewer similar to the one used for spatial segmentation partition refinement is launched, which supports the same functionalities described in the previous section.

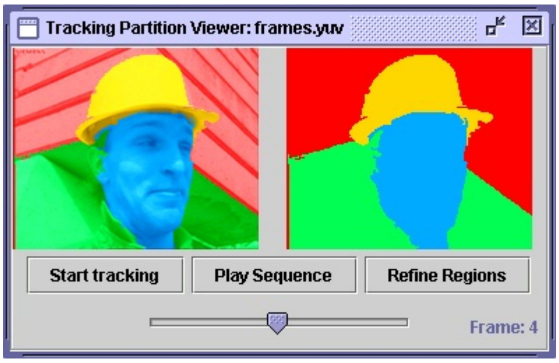


Figure 10 – Interface for tracking interaction

VI. INTERACTIVE TEMPORAL SEGMENTATION

The automatic temporal segmentation algorithm included in this version of the toolkit is based on a change detection algorithm, which classifies image pixels into two classes: moving or static.

The automatic algorithm performs camera noise estimation (using the previous knowledge about static areas) and adaptively thresholds the difference between the current image and the previous one to identify the changing regions. To be able to deal with moving cameras, global motion compensation is performed before the segmentation starts.

Again, the segmentation results obtained may deviate from the desired ones, and a change detection editor is included to let the user supervise the evolution of the results and correct them when appropriate.

The *change detection editor* has similar functionality as the editor implemented for the object tracking editor, allowing the supervision and correction of automatic results.

VII. CONCLUSION & FURTHER WORK

A video segmentation toolbox to perform video segmentation has been presented, including three automatic segmentation algorithms: spatial segmentation, object tracking and temporal segmentation, together with the corresponding user interaction editors.

The algorithms have been integrated in a friendly graphical user interface supporting several actions in order to facilitate the production of the desired segmentation partitions. The partition(s) generated by the automatic algorithms can be improved by means of user interaction. The whole process is very intuitive and no special knowledge is needed to operate the segmentation toolbox, and obtain good results for any type of sequence. This work is being continued in order to integrate other automatic algorithms together with their user interfacing requirements. Examples are motion segmentation, segmentation quality assessment, shot transition detection and sprite generation.

VII. REFERENCES

[Cor98a] P. Correia, 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, Issue 2, Elsevier, 1998, pp. 125-142

[Cor98b] P. Correia, F. Pereira, “User Interaction in Content-Based Video Coding and Indexing”, *Proceedings of EUSIPCO'98*, Rhodes, Greece, September 1998

[Cor98c] P. Correia, F. Pereira, “Proposal for an Integrated Video Analysis Framework”, *IEEE International Conference on Image Processing*, Chicago (IL), USA, October 4-7, 1998

[Mar99] B. Marcotegui, P. Correia, F. Marqués, R. Mech, R. Rosa, M. Wollborn, F. Zanoguera, "A Video Object Generation Tool Allowing Friendly User Interaction", *IEEE International Conference on Image Processing*, Kobe, Japan, October 25-28, 1999