Sensarea: an authoring tool to create accurate clickable videos

Pascal Bertolino

Sensarea: an Authoring Tool to Create Accurate Clickable Videos

Pascal Bertolino
Gipsa-lab
Grenoble Institute of Technology
France
pascal.bertolino@gipsa-lab.fr

Abstract

We present a user-friendly software application that can be used in a post-production environment and that allows to automatically or semi-automatically perform spatio-temporal segmentation of video objects. The central element in the application is a local pyramid-based segmentation algorithm. Several tools are provided to interactively guide or correct the automatic process when necessary. The masks of the extracted objects can be exported in a Flash or XML vectorized format and can be synchronized to the original video for many applications such as clickable videos.

1. Context

Offline extraction of object masks in videos is more and more a burning issue as it can be seen in the recent literature or in compositing software such as Adobe After Effects or Apple Motion. It allows to add information about an object inside the video or to modify its visual aspect without having to use complex and costly motion capture or blue screen techniques when shooting the video. Nevertheless, accurately delineating moving or deformable objects in hundreds of frame is a tedious task which is still mainly hand-made with more or less sophisticated drawing tools.

Indeed there is a gap between the quality reached by recent segmentation techniques and the lack of software that could meet the needs of a post-production environment: A couple of software (Motion and Final Cut by Apple, Video Deluxe by Magix, Adobe Premiere and Sony Vegas) allow to *track* a geometrical shape using a linear interpolated motion between two key frames. This functionality can be used for simple effects that do not need accuracy, like blurring a face or a plate number. To our knowledge, for advanced needs, the only software that proposes an accurate tracking tool is Mocha by Imagineer Systems, with its planar tracking. It is able to track a planar (or planar-like) surface initialized by the user. A parametric model guided by the user who indicates the type of the motion (translation, scale, rotation, shear, perspective) can describe any motion of the planar object. Then, this $2.5D$ tracking allows to insert extra content like virtual objects or to remove unwanted objects such as electric wires and cables. Since Mocha does not embed segmentation tools, it cannot cope will all the cases where the planar assumption is not anymore valid or when objects are non-rigid.

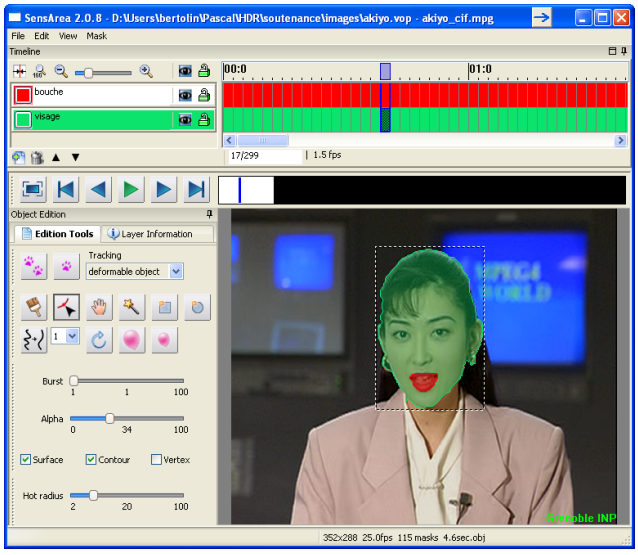


Figure 1. The Sensarea interface

As it is well known, none of the existing segmentation methods are able to accurately extract objects during many frames. In a post-production environment, user interaction is essential: it is necessary both to visually check frame by frame, on the fly, the quality of the masks and to stop the segmentation process whenever needed in order to correct, improve a mask or tune parameters before resuming the process. Furthermore, for the great majority of the recent methods, a user interaction is needed to provide an *a-priori* knowledge about the object of interest to be segmented:

In [13] the authors propose a semi-automatic segmentation technique where the user manually extracts masks in some representative frames and lets the matte-based process automatically segment the other frames. Video object segmentation can be seen as a semi-supervised learning problem as in [5] where the authors propose an incremental approach by iteratively labeling the least uncertain frame. Other approaches focus on object of interest initialized by the user: in [4], the authors model the object and the background (known from a manual initialization) with mixtures of Gaussians used in a level sets framework. The authors of [2] propose to use the graph-cut for videos. They initialize the hard constraints with a 3D interface by browsing through 2D slices. In [15], the user interaction combines marker drawing and region selection. A fast seeded region merging approach is proposed to extract the object from the regions of watershed segmentation. Interaction is the central point of [6] that combines classical morphological segmentation with motion estimation. More recently, in [9], the graph-cut based approach combined with an optical flow needs a classical initialization with scribbles. In [14], Tsai et al propose a convincing multi-label Markov Random Field model based on a manual initialization of the object in the first frame.

Concerning dedicated software applications to perform video object segmentation, only a few have been developed, most of the time prototypes with basic functionalities: many years ago, before democratization of digital videos, the VideoPrep application described in [1] was an authoring tool to segment, track and index the content of videos. The QIMERA project [12] aimed at developing a flexible modular software architecture for video object segmentation and tracking and supported user interaction when necessary. It mixed a system interface and a user interaction interface. In [11], Li et al extend the pixel-level 3D graph-cut proposed by [3] to the region-level 3D graph-cut to handle video objects. Their user-interface provides brush tools for the user to control the object boundary precisely wherever needed. In [8] the authors develop a special user interface to video markup, tracking and grouping that can be employed in a wide variety of applications such as authoring rich media.

Despite these works, it must be noted that there is still no off-the-shelf software to perform the video object segmentation task and to exploit the capacity of interactive methods. As an attempt to fill the gap, the main contribution of our work is the embedding of several segmentation tools in a user-friendly interface that allows a user in a post-production environment to use both automatic tools and editing tools whenever needed.

In the sequel, we first shortly present the main segmentation method that we implemented then we describe the software functionalities that mainly consist in video browsing, interactive correction, data export and the ability to easily embed other segmentation tools.

2. The spatio-temporal segmentation algorithm

We use a method developed during previous work [7] that we summarize shortly hereafter. It is applied between two consecutive frames I_1 (for which the partition P_1 is given by the user) and I_2. It is carried out in three steps (figure 2): First the projection of the known partition P_1 on the next frame I_2 gives an intermediate rough partition P_2'. Then the local spatial segmentation of I_2 to refine P_2' generates the temporary partition P_2'' in which some regions could not be labeled. At last, to label these regions, they are retro-projected on I_1. This provides the final partition P_2.

In order to deal with non rigid objects, the projection is performed with a regular block matching that corresponds to a local motion estimation of the object border. Its role is to find a good quality match for each block of $I2$ on $I1$. When a good match is found, the corresponding $P1$ block is projected to $P2$ which propagates the foreground and background labels to $P2$. Most of the time, the blocks are well located in the next frame but local changes such as rotation or small deformation may slightly change the contour location and make an irregular contour between neighboring blocks. Besides, some classical issues such as aperture problem, background occlusion or disocclusion may yield inaccurate matching. For these two reasons, a reassignment of the pixel labels in a thin area along of the foreground - background border is performed once all the blocks are projected. The reassignment is carried out with a local segmentation based on a previous work [10] that uses a graph pyramid to propagate inside unlabeled pixels the foreground or background label using a color similarity criterion.

Although the results provided by this technique cannot rival the most recent methods, it can deal with deformable objects (figure 4) and remains fast enough to be used in an interactive context.

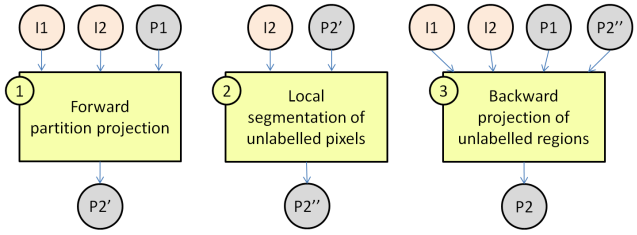
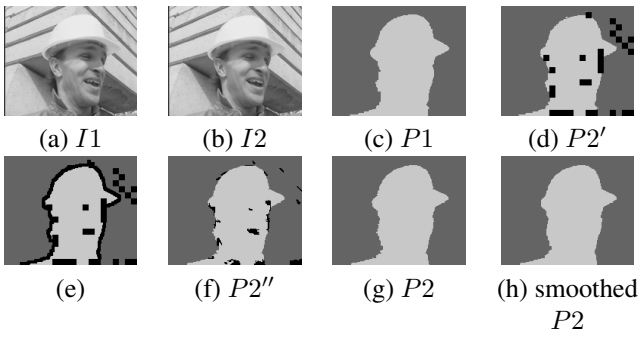


Figure 2. The three steps of the video object segmentation method



(a) $I1$ (b) $I2$ (c) $P1$ (d) $P2'$

(e) (f) $P2''$ (g) $P2$ (h) smoothed $P2$

Figure 3. Principle of the video object segmentation method

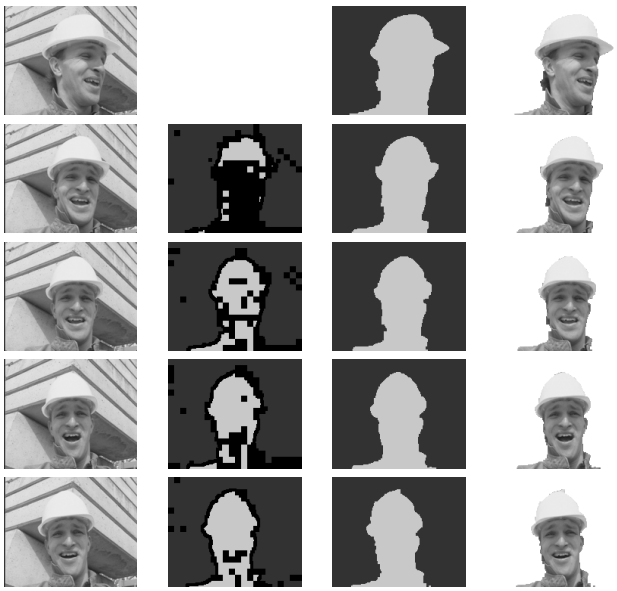


Figure 4. First column: the frames of a time under-sampled video. Second column: projected partitions. Third column: final partitions. Last column: segmented objects

3. Software functionalities

Although the algorithm described above can deal with any number of objects, one object at a time can be tracked in our implementation to facilitate the user interaction. One layer must be created for each object of interest. A layer will contain all the occurrences of the corresponding object. The main philosophy is that there is no constraint for extracting several objects inside a video. Initialization of the object of interest can be done either manually or with the magic wand or both. Then, spatio-temporal segmentation can be performed step by step (one frame at a time) or automatically on several frames. In the latter case, it can easily be stopped on the fly with a mouse click. Thus, manual refinement or correction of one ore several masks can be done at any time.

Our application has been developed in C/C++. All the image processing tools are written in C when all the graphic interface is coded in C++ using the cross-platform library wxWidgets. The database engine that stores all the results is SQLite, a cross-platform C library. At the moment, the application runs under Windows but it could be compiled under Linux or MacOS. The segmentation codes are independent of the graphic interface code. The sequel presents the main functionalities of the software.

3.1. Video browsing

When editing a video, it is important to be able to navigate quickly and accurately through the video. To that purpose a VCR-like interface and shortcut keys allow play/pause, random access and frame-by-frame playing. In order to process videos compressed with common codec formats and container formats, decoding is transparently performed with the help of the FFMPEG library. A part of the user interface called timeline (figure 5) displays one row per layer and one column per frame. With it, it is easy at a glimpse to know which object has already been extracted, where and when. A mouse click in the timeline gives direct access to the corresponding frame / layer and to the mask if any. Buttons allow to change the depth of the layers, to hide or lock them.

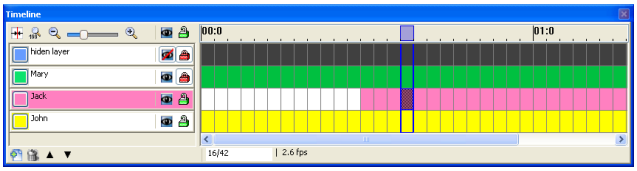


Figure 5. Video timeline

3.2. Interactive editing

Since the object masks are vectorized, they can be edited either as a set of pixels or as vectorized shapes depending on the used tool. Several tools are proposed to initialize, correct, modify the mask of one occurrence or of the occurrences within the selected range of frames in a layer: magic wand, brush, eraser (size can be tuned), move / resize, simplification, erosion and dilation, rotation. A vertex tool allows to add, move or remove a vertex. To help the user to edit or appreciate the quality of the masks, different renderings are available (figure 6). Color of the mask can be changed at any time, their transparency can be tuned as well as masks borders or vertices can be displayed or not.

Figure 6. Example of available renderings

3.3. Data export and openness towards other segmentation tools

The videos processed by Sensarea are not modified. The masks that are produced are independent and can be exported for any other application in three different formats. Note that our application can also be used to easily produce ground truth data. Our software architecture allows to enrich it with one or several other segmentation tools (either spatial or spatio-temporal) in a quite easy way. In order to test its openness and flexibility, we have added some extra spatio-temporal segmentation tools to track objects in different contexts.

4. Conclusion

Sensarea is a platform that already offers most of the facilities required for a multi-purpose authoring tool to make offline video object segmentation. It provides ergonomic tools to edit the results. We have already shown the software flexibility by embedding several segmentation tools. We aim at improving our solution by embedding more segmentation methods and we are open to the community working in this field. When several complementary segmentation tools are available, the automatic selection of the right tool according to the scene cues will really be a plus for the user. In the future, Sensarea could also include segmentation quality measures to provide a general framework for segmentation evaluation and comparison of methods.

Acknowledgments

This material is based upon work supported by the French OSIAM/RNRT program and the French Région Rhône Alpes. The support of GRAVIT is gratefully acknowledged.

References

[1] S. Benayoun, H. Bernard, P. Bertolino, P. Bouthemy, M. Gelgon, R. Mohr, C. Schmid, and F. Spindler. Structuring video documents for advanced interfaces. In *ACM Multimedia*, Bristol, UK, september 1998.

[2] Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[3] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *International Conference on Computer Vision*, 1:105–112, July 2001.

[4] P. Chockalingam, S. N. Pradeep, and S. Birchfield. Adaptive fragments-based tracking of non-rigid objects using level sets. In *IEEE International Conference on Computer Vision*, pages 1530–1537. IEEE, 2009.

[5] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg. Combining self training and active learning for video segmentation. In *Proc. BMVC*, pages 78.1–78.11, 2011. http://dx.doi.org/10.5244/C.25.78.

[6] F. C. Flores and R. d. A. Lotufo. Watershed from propagated markers: An interactive method to morphological object segmentation in image sequences. *Image Vision Comput.*, 28(11):1491–1514, Nov. 2010.

[7] G. Foret and P. Bertolino. Label prediction and local segmentation for accurate video object tracking. In *SPIE Visual Communications and Image Processing*, Lugano, Switzerland, 8-11 July 2003.

[8] D. B. Goldman, C. Gonterman, B. Curless, D. Salesin, and S. M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 3–12, New York, NY, USA, 2008. ACM.

[9] K. Housni, D. Mammass, and Y. Chahir. Article: Moving objects tracking in video by graph cuts and parameter motion model. *International Journal of Computer Applications*, 40(10):20–27, February 2012. Published by Foundation of Computer Science, New York, USA.

[10] J. Huart, G. Foret, and P. Bertolino. Moving object extraction with a localized pyramid. In *International Conference on Pattern Recognition*, Cambridge, UK, august 2004.

[11] Y. Li, J. Sun, and H. yeung Shum. Video object cut and paste. *ACM Transactions on Graphics*, 24:595–600, 2005.

[12] N. O'Connor, T. Adamek, S. Sav, N. Murphy, and S. Marlow. Qimera: A software platform for video object segmentation and tracking. In *In Proc. Workshop on Image Analysis For Multimedia Interactive Services*, pages 204–209, 2003.

[13] D. Ring and A. Kokaram. Feature-cut: Video object segmentation through local feature correspondences. In *IEEE International Conference on Computer Vision*, 2009.

[14] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *Proc. BMVC*, pages 56.1–11, 2010. doi:10.5244/C.24.56.

[15] L. Zhi and Y. Jie. Interactive video object segmentation: fast seeded region merging approach. *Electronics Letters*, 40(5):302–304, 2004.