YAV++: a software platform for medical image processing and visualization

H. Delingette[1], E. Bardinet[1], D. Rey[1], J-D. Lemarechal[1], J. Montagnat[2], S. Ourselin[1], A. Roche[1], D. Dormont[3], J. Yelnik[4], and N. Ayache[1]

[1] INRIA, Epidaure Project, Sophia Antipolis, France
[2] CREATIS... , France
[3] Neuroradiology Dept and LENA UPR 640-CNRS, Salpêtrière Hospital, France
[4] INSERM U289, Salpêtrière Hospital, France
`[hdeling,ebard,drey]@sophia.inria.fr`

Abstract. In this paper, we present a generic platform for the visualization and processing of volumetric medical images. This platform combines three characteristics that are required for developing large interactive applications : modularity, portability and scripting ability. We present three applications built upon this platform : deformable model based segmentation, multimodal registration and analysis of time series of images.

1 Introduction

This paper presents the status and some applications of the software platform **yav++** dedicated to medical image processing at the INRIA research group Epidaure in Sophia-Antipolis, France. The motivation of this platform is twofold. First, it aims at structuring the code being developed inside the Epidaure research group. Second, it is motivated by the belief that for many medical imaging applications, user interaction must be closely tighted with medical image processing. The primary targeted application of this platform was the segmentation of medical images based on 3D and 4D deformable models [3, 4] where user interaction is required to guide the surface deformation during the deformation process. We have then extended its scope to solve different problems related to medical image registration and analysis.

When designing such a platform, we have tried to reach three objectives :

1. **Modularity** At the heart of the platform is the notion of modules. A module is composed of a shared library together with a file containing some Tcl code. The modules are organised in a hierarchical manner thus allowing the sharing of code and ideas. A set of basics modules related to volumetric images and meshes is provided. Each developer can choose either to create its own module or improve the basic modules, depending on the genericity of its development. Finally, we have created specific modules as interfaces to other image processing libraries.
2. **Portability** We are currently supporting 5 different operating systems (Windows NT, Linux, Iris, Solaris and Digital Unix) with three different C++ compilers (Visual C++, g++ and Irix CC). In particular, the support of both Unix and Windows environments imposes strong constraints on the choice of the API for the 3D graphics and user interface API. Rather than relying on a high level 3D graphics API such as Open Inventor, we have chosen to directly use OpenGL in order to have a complete control of the platform portability.
3. **Scripting ability** Using scripts is useful in many ways for medical imaging applications. First of all, it allows to quickly build prototype applications especially when the scripting language can be used to produce a graphical user interface (as with Tcl/TK). Second, it is a convenient method for writing batch application for the automatic processing of large medical image databases. We have chosen to use Tcl/Tk for its portability and ability to produce user interface (with TK). The creation of Tcl

commands associated with each module is performed explicitly (not generated automatically by a parser as in the VTK package) which requires some additional development effort but also enables to have commands with a simple syntax together with strong parameter checking and explicit error messages.

This platform has an architecture which is somewhat similar to the Visualization Toolkit. Nevertheless, because it is based on the data-flow paradigm, VTK is not well suited for highly interactive application where computation and user interface are closely related. Furthermore, the **yav++** platform is not targeted to be a general visualization software but dedicated to the processing of volumetric images and to the deformation of contours, surfaces or volumes.

Instead of describing more formally the different components of the **yav++** platform, we describe in the next sections, three typical applications.

2 Application I : Deformable Models

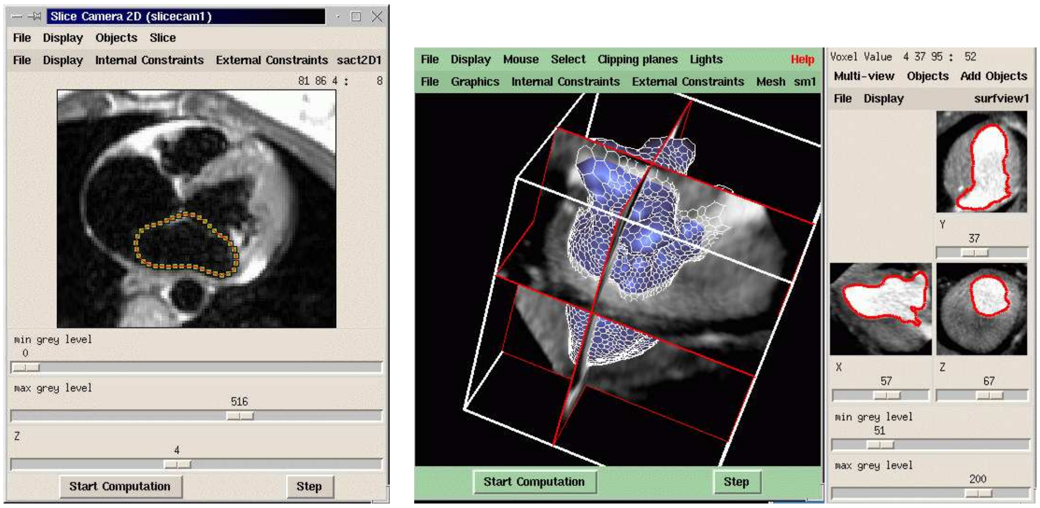
Among the different techniques for segmenting medical images, the methods based on deformable models have the advantage of providing a strong a priori knowledge about the structures to segment. Also these methods usually require some level of user supervision to guide the segmentation. In many cases, this medical supervision is not a drawback (and even very beneficial) as long as the computation of the deformable models is fast enough to perform the segmentation task in a few minutes.

In Figure 2 we present some snapshots of image segmentation on 2D, 3D and 4D medical images. For the 2D image segmentation, the user can drag the contour with the mouse and propagate each deformable contour to neighboring slices. Each active contour can change its topology when collisions are detected [2]. The 3D deformable surfaces are represented as simplex meshes [1] which are deformed inside volumetric images. The slice view of deformable surfaces can be seen in any orthogonal slicer and user interaction (dragging the surface towards or away an image region) can be performed in the 3D view or in each slice view. We found that the combination of 3D and slice views is very helpful for performing the difficult 3D segmentation of structures such as the kidney or liver. Indeed, the 3D view helps the user to understand globally the segmentation process whereas each orthogonal view allows to control the segmentation accuracy in specific planes. Deformable model based segmentation benefits a lot from the use of scripting language because we can easily set and modify the large number of parameters for a given application. For instance, we have created a set of scripted applications for segmenting the kidney with some user guidance such as definition of the ROI, image windowing, selection of a kidney template. However, the computation complexity is hidden by the interface so that the user do not have to enter any magic parameters.

The 4D (3D+T) deformable model interface is based on the interface for a 3D deformable model with the addition of a slider that select the time at which the scene (consisting of a volumetric image and a deformable surface) is viewed.

3 Application II : Segmentation of histological sections and fusion with MR images

In neurosurgery, localization of deep brain structures is a crucial issue. For instance, a new surgical treatment of Parkinson's disease consisting of deep brain stimulation has been recently developed. To determine with high accuracy the target localisation, the key issue appears to be the construction of a 3D atlas of the human basal ganglia, designed for further propagation onto the MR acquisition of a given patient. Our goal is thus to build such an atlas by fusing histological data with a 3D MR image of the same subject. This requires three steps: first an automatic 2D realignment of the histological sections in order to obtain a three-dimensional block, then a manual segmentation of the basal ganglia performed



(a) 2D DeformableModel

(b) 3D Deformable Model

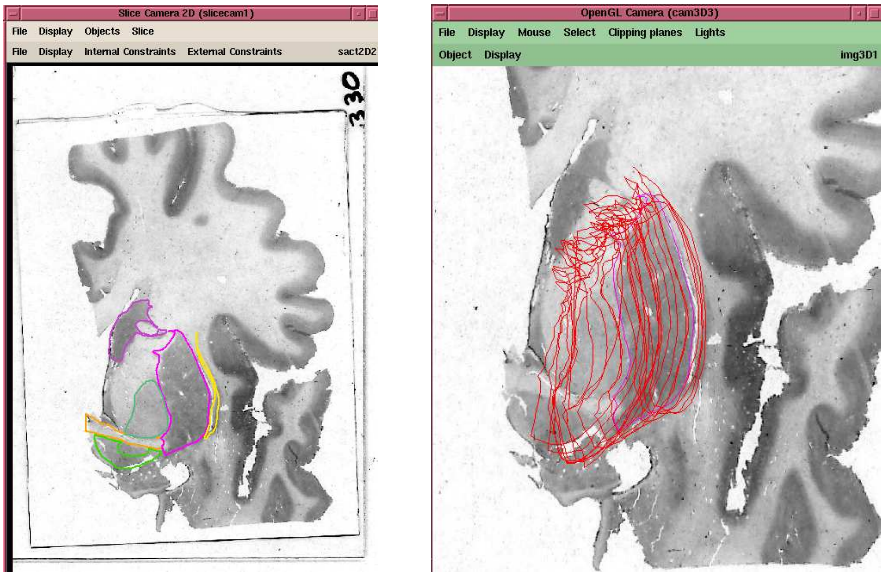


Fig. 1. *Supervised segmentation of the histological sections. Left: contours of the basal ganglia manually segmented section-by-section. Right: 3D view of the putamen contours superimposed with the associated histological section.*

by an anatomist, and finally a 3D registration between the reconstructed and segmented block and the MR image. The first and last steps are based on a robust registration algorithm (see [5]), whereas the second step, performed within the **yav++** platform, consists in a manual segmentation of the histological sections. A key feature of **yav++** that is used during this slice by slice segmentation is the seamless synchronisation of a 2D and a 3D view of both the images and the slice contours. The user manually traces each contour in the 2D view while checking its 3D consistency in the 3D view (see Figure 1).

4 Application III : Follow-up of lesions in time series of images

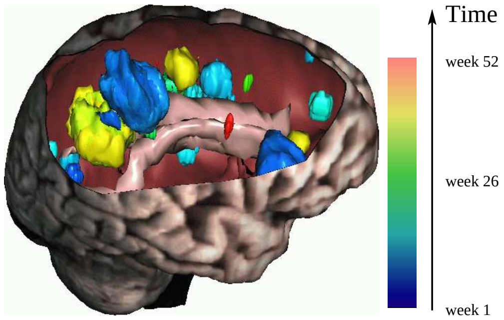


Fig. 2. *3D visualisation of evolving lesions. The color is related to the time of appearance of each lesion.*

The analysis of evolving processes over time, for instance multiple sclerosis lesions, tumors, or anatomical structures (i.e. cerebral ventricles) helps in diagnosis and allows the follow-up of a patient over time, especially to study the effects of a treatment. We developed two different methodologies to detect and quantify evolving multiple sclerosis lesions in series of 3D MR scans [6, 7]. **yav++** is very useful to look at 4D results, especially because it is possible to superimpose resulting segmentations of our methods on $3D + t$ images, as it allows to change all the parameters of the $3D + t$ images interactively and makes it possible to mix two $3D + t$ images with an opacity variable.

Moreover it is possible to visually combine these results in 3D (Figure 2) to help clinicians to localize evolving lesions with respect to typical brain structures.

5 Conclusion

With the applications described above, we have validated the architecture and concepts of the **yav++** platform. We are convinced that combining visualization and computation in the same platform is mandatory for complex medical applications both in the development and operational stage. In a research environment, scripting proved to be especially useful since it allows to quickly built and modify application prototypes.

Many features still need to be integrated in this platform from medical image processing to purely visualization tools. To extend the platform functionalities, we will partially rely on the creation of modules linked to existing software packages. Finally, we plan soon to make this platform available to the medical imaging community through anonymous FTP.

References

1. H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146, September 1999.
2. H. Delingette and J. Montagnat. Shape and topology constraints on parametric active contours. *Computer Vision and Image Understanding*, 2001. to appear (accepted for publication).
3. J. Montagnat and H. Delingette. A Hybrid Framework for Surface Registration and Deformable Models. In *Computer Vision and Pattern Recognition, CVPR'97*, pages 1041–1046, San Juan, Puerto Rico, June 1997.
4. J. Montagnat and H. Delingette. Space and Time Shape Constrained Deformable Surfaces for 4D Medical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI'2000)*, volume 1935 of *Lectures Notes in Computer Science*, pages 196–205, Pittsburgh, USA, October 2000. Springer.
5. S. Ourselin, E. Bardinet, D. Dormont, G. Malandain, A. Roche, N. Ayache, D. Tande, K. Parain, and J. Yelnik. Fusion of histological sections and MR images: towards the construction of an atlas of the human basal ganglia. In *Proc. MICCAI'01*, Utrecht, October 2001.
6. D. Rey, J. Stoeckel, G. Malandain, and N. Ayache. Using SPM to Detect Evolving MS Lesions. In *Proc. of MICCAI'01*, LNCS, Utrecht, Holand, October 2001. Accepted.
7. D. Rey, G. Subsol, H. Delingette, and N. Ayache. Automatic Detection and Segmentation of Evolving Processes in 3D Medical Images: Application to Multiple Sclerosis. In *Proc. of IPMI'99*, LNCS, pages 154–167, Visegrád, Hongrie, June 1999.