Framework for Interactive Medical Imaging Applications

André Ferreira Bem Silva, Tiago H. C. Nobrega , Diego Dias Bispo Carvalho , Renan Teston Inácio , Aldo von Wangenheim

[1]LAPIX – Laboratory for Image Processing and Computer Graphics
UFSC – Federal University of Santa Catarina
Florianópolis, Santa Catarina, Brazil

{tigarmo,diegodbc,awangenh}@inf.ufsc.br

Abstract. *We propose a set of libraries with the goal of providing a framework to build medical applications. Instead of using the existing packages in libraries to perform the tasks of image and volume handling and visualization, all code has been written for scratch. We expose the reasoning for this decision while presenting a medical workstation and a few physical simulation prototypes as example applications.*

1. Introduction

Medical Computer-Aided Diagnostic tasks are critical in the sense that as the technology in the field advances, fewer are the possibly vital errors in medical judgement. We present the design and implementation of a radiological 3D workstation (abbreviated as "WS") together with a set of libraries for basic algorithm and rendering tasks. The set was developed in-house in its entirety instead of adopting existing solutions, and our reasons for this are presented and discussed.

1.1. Motivation

There are many established libraries and toolkits that aid in the creation of medical software, such as MITK, DCMTK, ITK and VTK and their extensions. In this sense, there should be good reasons for coding teams to implement (and maintain) everything from scratch.

In our case, we needed different, specific functionalities that the available APIs (at the time) didn't provide, such as GPU access for physical simulations (*GPGPU*) and multithreaded implementations of specific algorithms to take advantage of the multiple cores in today's desktops. We needed the understanding of every layer in the set of algorithms, from the low-level data-structure manipulation to the final rendering, in order to identify hot spots and bottlenecks (to obtain high performance). Such knowledge is vital when the time comes to port the application and the libraries to another architecture. Additionally, the comparison between CPU and GPU versions of an algorithm can be easier if the same environment and methodologies were used to implement both.

Finally, when there are different teams in the same group working on different tasks, having a common set of libraries developed in-house has the advantage of better feedback (framework and application developers working together). This way, problems are solved more easily and the learning curve is smaller (easy to share experience).

2. C3DE libraries architecture

In addition to the needs specified in the previous section, the Cyclops 3D Environment libraries are all written in standard C++ using cross-platforms APIs for widgets and 3D rendering. They're currently supported on Linux® and Microsoft Windows®, and are compatible with 32 and 64-bit x86 architectures.

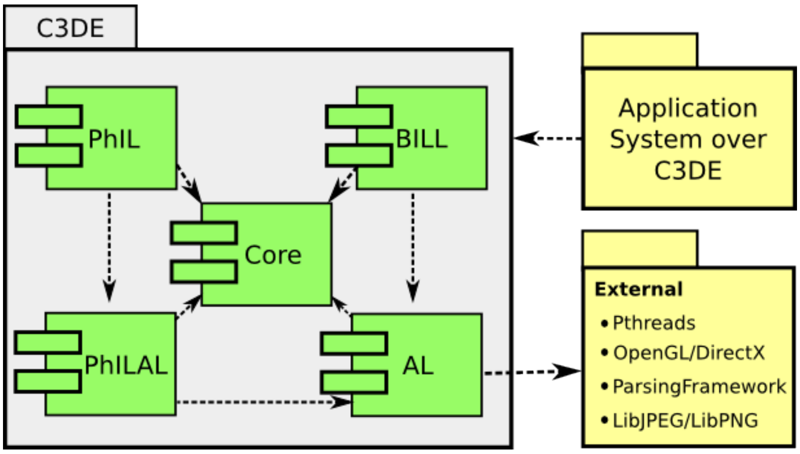


Figure 1. C3DE basic architecture

2.1. Core

The Core's tasked with base mathematical structures such as matrices and vectors, volume abstractions, image definitions, 3D spatial management such as the intersection of arbitrary meshes with planes and lines, and common utilities like smart pointers. For generical data and data structures treatment, the Core's most complex structures make heavy use of template metaprogramming.

2.2. AL

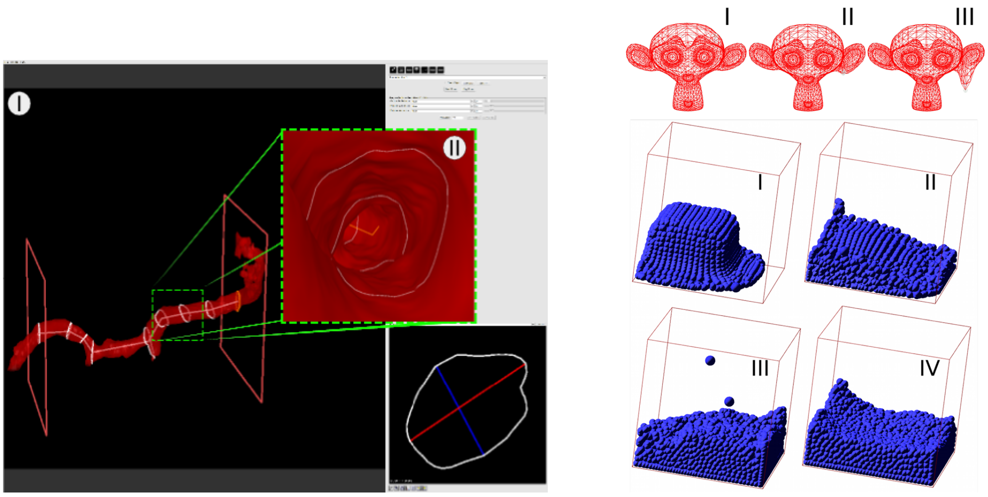
The Abstraction Layer, as the name implies, wraps the Core's complex metaprogramming with the standard Object-Oriented paradigm. It provides factories for images, volumes and three-dimensional reconstructions, drawable objects, shaders and texture handling through OpenGL, and standard multi-planar reconstruction.

2.3. PhIL

PhIL stands for Physical Interaction Layer and provides structures that extend the basic functionalities found in Core with physical properties. For instance, the standard Half-Edge mesh is extended to support deformations using the mass-spring model [Provot 1995]. Currently, the library also supports fluid simulations with the particle-based Smoothed Particle Hydrodynamics (SPH) method [Muller et al. 2003].

2.4. PhILAL

The Physical Interaction Abstraction Layer binds PhIL and AL, with multithreaded and GPU-enabled versions of PhIL algorithms. Like AL, there are also factory methods to quickly prototype and render PhIL structures, like deformable meshes (mass-spring) and liquid simulation (SPH), sawn in steps on 2(b).



(a) (b)

Figure 2. Centerline approximation (a) and physical simulation prototypes (b).

2.5. BILL

The Binding Input Layer Library (BILL) defines interfaces to connect input devices and the applications. Its core purpose is on the translation of events from devices such as a mouse into meaningful actions for the application, through event handlers.

BILL itself depends only on Core for metaprogramming facilities and AL for multithreading — this is because the library does not provide devices or actions, only their interfaces and the basic mechanisms of their integration.

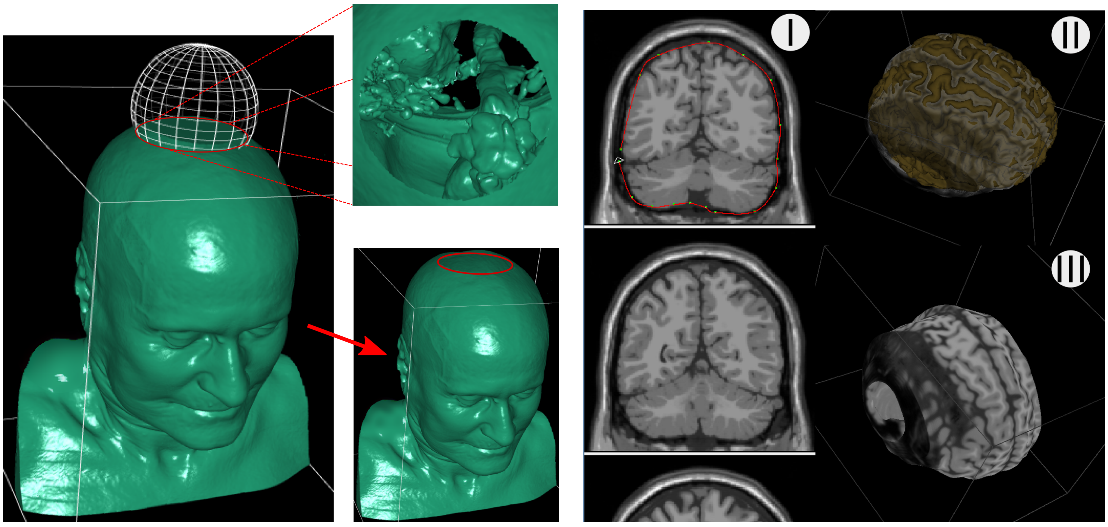
3. Workstation

The WS is built on top of the framework and uses extensively all Core and AL resources. Its features include reconstruction and segmentation, approximate centerline calculation, Volume-Of-Interest oriented reconstruction, Curvilinear reconstructions and fiber tracking.

Given two user-positioned planes of interest, the approximate centerline generation algorithm computes a set of line segments that pass roughly through the center of a structure by firing lines between the planes and computing the lines' intersection with the structure, recursively (Figure 2(a)).

The curvilinear reconstruction [Bastos et al. 1995] aims at reconstructing segments that are difficult to measure and distinguish with simple grayscale thresholding. By placing a set of points on two-dimensional slices (Figure 3(b)) representing regions of interest, the user obtains a curved surface that is interpolated from the points with Catmull-Rom splines.

The Volume-Of-Interest reconstruction works on objects to remove unwanted portions, or segment it manually. Its counterpart is the segmented reconstruction that differentiates distinct objects through thresholding on the CT or MRI volume. The inputs are the density limits (lower and upper) to be used, with optional fields for noise filters for



(a) (b)

Figure 3. VOI-oriented (left) and curvilinear (right) reconstructions.

relatively small objects and minimum and maximum absolute volumes. The VOI feature uses spheres and boxes to indicate the areas of interest (Figure 3(a)).

4. Conclusions and Future work

The libraries can be used for fast prototyping (like the physical simulation ones) while being robust enough to implement a medical workstation with standard features. We plan to extend the framework with new libraries such as a Network Interaction Layer over BILL for collaborative sessions over the internet. Additionally, we plan to build a sandbox prototype with PhIL and PhILAL to easy visualize and simulate interactions of solid and deformable objects and fluids. The WS is always maintained with the latest version of the framework to always maintain compability with it.

There are some interesting related work, like the ones that INRIA group made in medical imaging software. It would be interesting to exchange information and development experience with them and/or other groups related to the area.

References

Bastos, A. C., Korah, I. P., Cendes, F., Melanson, D., Tampieri, D., Peters, T., Dubeau, F., and Andermann, F. (1995). Curvilinear reconstruction of 3d magnetic resonance imaging in patients with partial epilepsy: A pilot study. *Magnetic Resonance Imaging*, 13(8):1107 – 1112. Workshop on Magnetic Resonance Techniques and Epilepsy Research.

Muller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Provot, X. (1995). Deformation constraints in a mass-spring model to describe rigid cloth behavior. In Davis, W. A. and Prusinkiewicz, P., editors, *Graphics Interface '95*, pages 147–154. Canadian Human-Computer Communications Society.