INTRODUCTION

According to World Health Organization estimates, 17 million people around the globe die of cardiovascular disease (CVD) each year. About 600 million people with high blood pressure are at risk of heart attack, stroke and cardiac failure. Low and middle-income countries contributed to 78 percent of CVD deaths. By 2010 CVD is estimated to be the major cause of death in developed countries. This huge impact has motivated the development of new non-invasively techniques in order to understand cardiovascular pathologies and interventions. The effect of blood flow on arterial wall remodeling the investigation of vascular hemodynamics within the vascular vessels is of great interest. It has the potential to greatly enhance the understanding of the pathogenesis and progression of vascular diseases and to aid in the decision of whether treatment is warranted. Non-invasive magnetic resonance imaging (MRI), with its intrinsic sensitivity to blood flow, offers the unique possibility to simultaneously acquire morphology and spatially coregistered hemodynamic information non-invasively.

METHODODOLOGY

2.1 Medical-GiD

Medical-GiD is a homemade software development in CIMNE, designed for qualitative evaluation of medical images, in particular for deformable registration of 4D MRI images. Medical-GiD is written in the Tcl interpreted language so that it is integrated as a module of the Personal Pre and Post Processor GiD [GiD 2008]. GiD is a universal, adaptive and user-friendly graphical user interface for geometrical modelling, data input and visualization of results for all types of...
numerical simulation programs. The object oriented structure has been implemented using Snit in order to be integrated Medical-GiD as a module of GiD or other programs. Medical-GiD is based on several open source libraries able to read and write different images formats using Insight Segmentation and Registration Toolkit library (ITK)[2]. For the image processing tasks, Medical-GiD uses ITK libraries and an inner development filters for noise filtering and eddy current correction in MR phase velocity data. The program GUI is based on VTK[3,4] (Visualization toolkit) which is an open source toolkit which can be integrated freely in commercial software as well ITK.

2.2 Architecture design.

In the figure (fig 1) the architecture structure of Medical-GiD is shown. The architecture is based on two different modules communicates by ITK-VTK filter. The whole visualization process is programmed using the VTK toolkit and the glue used to cement the code and create the GUI is Tcl/Tk[5]. Using the visualization the user can change some of the parameters used during the image processing. Those parameters are passed back to ITK and the image processing is updated for a new visualization. Although the user guide of ITK says that ITK has been wrapped using many scripting languages, such as Tcl, Java and Python, there are almost no documentation available online or off the shelves on how to use the wrapped functions. Therefore, we have not been able to use libraries using Tcl, and we have developed our own wrappers in C++. Although this seems (and was) a hard task, it allowed us to implement our own C++ classes to interact with the image and provide to implement easier our filters. A special care has been taken to use programming languages as well as libraries compatible with the actual form of GiD. The Tcl/Tk language is used for the GUI and as cement for the code. Snit has been used to
Snit has been used to develop an object oriented code in Tcl. And the two toolkits used (ITK and VTK) are open source code which are freely implementable into a commercial code.

2.3 Magnetic Resonance Imaging

All examinations were performed on a 3 T MR system (Magnetom TRIO; Siemens, Erlangen, Germany) using time-resolved, 3-dimensional MR velocity mapping based on an RF-spoiled, gradient-echo sequence with interleaved 3-directional velocity encoding (predefined fixed velocity sensitivity = 150 cm/s for all measurements). Data were acquired in a sagittal-oblique, 3-dimensional volume that included the entire thoracic aorta and the proximal parts of the supra-aortic branches. Each 3-dimensional volume was carefully planned and adapted to the individual anatomy (spatial resolution, 2.1 x 3.2-3.5 x 3.5-5 mm3). To resolve the temporal evolution of vascular geometry and blood flow, measurements were synchronized with the cardiac cycle. Prospective ECG gating in combination with k-space segmented data acquisition was performed resulting in a cine series of 3-dimensional magnitude and velocity data sets. Two-fold acquisition (k-space segmentation factor = 2) of reference and 3-directional velocity sensitive scans for each cine time frame resulted in a temporal resolution of 8 repetition time = 45 to 49 milliseconds. To minimize breathing artifacts and image blurring, respiration control was performed based on combined adaptive k-space reordering and navigator gating. Further imaging parameters were as follows: rectangular field of view = 400x (267-300) mm2, flip angle = 15 degrees, time to echo = 3.5 to 3.7 milliseconds, repetition time = 5.6 to 6.1 milliseconds, and bandwidth = 480 to 650 Hz per pixel.[6-7-8].

2.4 Segmentation and Meshing for computational simulations.

The development of computational simulations in medicine, molecular biology and engineering has increased the need for quality finite element meshes. For the segmentation procedure Medical-GiD includes a variety of ITK filters, which are use interactively by the clinicians to determinate the volume of interest of the problem. After segmenting the medical image we end with a file with the image data and the value of the isosurface value defining the boundary of the volume of interest. The imaging data V is given in the form of sampled function values on rectilinear grids,

\[ V = F(x_i; y_j; z_k) \text{ where } 0 \leq i \leq n_x; 0 \leq j \leq n_y; 0 \leq k \leq n_z. \]  

We assume a continuous function F is constructed through the trilinear interpolation of sampled values for each cubic cell in the volume. The format used to read the medical data is VTK structured point as it is agreed in [9]. The description of this format can be found in [9]. The image in this format can also be rendered as a volume and manipulated with ITK. Given an isosurface value defining the boundary of the volume of interest we can extract a geometric model of it. We are interested in creating a discretization of the volume suitable for finite element computation. The following methods to generate the finite element mesh to be used in the computational analysis has being integrated into Medical-GiD i) Dual contouring, ii) Marching cubes, iii) Advancing front and iii) Volume preserving Laplacian smooth.

In order to generate in Medical-GiD a tetrahedral mesh from voxels we combine the Marching Cubes method to generate first the boundary mesh first and then, after a smoothing, an Advancing Front [10] method to fill the interior with tetrahedral. The Marching Cubes [9] algorithm visits each cell in the volume and performs local triangulation based on the sign configuration of the eight vertices. If one or more vertex of a cube have values less than the user-specified isovalue, and one or more have values greater than this value, we know the voxel must contribute some component of the isosurface. By determining
which edges of the cube are intersected by the isosurface, we can create triangular patches which divide the cube between regions within the isosurface and regions outside. By connecting the patches from all cubes on the isosurface boundary, we get a surface representation. The Advancing Front [10] is an unstructured grid generation method. Grids are generated by marching from boundaries (front) towards the interior. Tetrahedral elements are generated based on the initial front. As tetrahedral elements are generated, the "initial front" is updated until the entire domain is covered with tetrahedral elements, and the front is emptied. Figure 2(b) shows a cut of the tetrahedral mesh generated by the Advancing Front method. Some of the triangles generated by the Marching Cubes method do not exhibit good quality to be used in finite element computation. In order to improve the quality of those elements we apply a Laplacian smooth which volume preserving. The smoothing algorithm implemented is simple: it tries to preserve the volume after each application of the laplace operator by doing an offset of the vertices along the normal. Figure 2(a) shows the boundary mesh generated by Marching Cubes and smoothed[11].

3 RESULTS

The obtained ECG-synchronized time series of 3D data sets were read in Medical-GiD and underwent fully automated noise filtering, segmented, meshing and visualization inside Medical-GiD, which offered different data visualization options to illustrate the dynamics of 3-dimensional blood flow encoded in the velocity data. Visualization of the patterns and dynamics of 3-dimensional blood flow was spatially registered with the anatomical information provided by the magnitude data and included vector graphs, 3-D streamlines and time-resolved.

We obtained really good segmentation of the aorta by applying a threshold filter to the velocity magnitude of blood in the vector data. The third iteration of the whole MRI data has been used for a good segmentation, because during this iteration, the blood is pulsing through the aorta, the velocities are high and therefore the threshold is working very satisfactorily. There are a lot of noise in the original images, especially in the zones where there are no tissues (the velocity seems to have a random value where there is air) (Fig 3).

In order to select only the region of interest and not only the zones of high velocity magnitude which would include the air surroundings, we used first a filter to compute the velocity magnitude, and then we used a connected threshold to eliminate points where the velocity is too low. But since we didn't want to capture the noise, we chose to use a neighborhood connected threshold image filter, this means that a voxel is accepted if itself and all its neighbors have a velocity magnitude above a defined threshold. Furthermore all voxels selected need to be connected with at least

Figure 3: Vertical velocity magnitude during heartbeat. The image obtained is really noisy in the zones without blood.
one seed point. The results almost exactly the aorta, but it looks like an eroded (thinner) version of the aorta. Indeed voxels located close to the edge of the aorta have their neighborhood crossing over outside of the aorta. Those points are really noisy and may have a value of velocity magnitude below the threshold and therefore the voxel is not selected as part of the aorta. It is then necessary to “dilate” the aorta by one or 2 pixels to capture exactly the aorta. The following image shows 3D blood velocity field in the descendent aorta obtained directly from the MRI. The segmentation is then used as a mask for the velocity image and is superimposed on a slice of the scalar data. (Fig 4). Figure 4(a) and Figure 4(b) show respectively the longitudinal section and detail of the aorta. The color lines represent the blood velocity field inside the aorta. Concerning the calibration of the magnetic resonance machine, a special procedure has been developed to obtain the blood velocity field: this special algorithm has to be properly adjusted to the velocity encoding parameters of the MRI.

Furthermore, vector graphs representing the pixel wise, 3-directional blood flow velocities could be superimposed on selected cut planes and animated over the cardiac cycle. For an overview over 3D blood flow patterns at a single time frame within the cardiac cycle, paths originating from the predefined emitter planes and tangent to all measured velocity vectors (3D streamline) were calculated. Temporal information could be added by generating traces within the 3D volume representing the path of imaginary massless particles within the measured time-resolved velocity vector fields starting at a specified time within the cardiac cycle (time-resolved, 3D particle traces).

4 CONCLUSION

Medical-GiD is a development environment of medical images treatment. Medical-GiD is based on ITK, VTK, TCL-TK libraries and C++ applications for visualize, segmentation and meshing of 3D and 4D images in Linux and Windows. One of the main applications of Medical-GiD is working directly from the data of 4D-MRI images, allowing in a unique user-interface: filtering, segmented, meshing and visualizes a real blood flood profiles. Medical-GiD is designed to be integrated easily in other programs, as GiD. Other visualization and segmentation tools will be implemented according to the clinician’s requirements. Future works will include advanced techniques in image processing, meshing and also improvements in the fitting techniques of the velocity encoding parameters.

5 ACKNOWLEDGMENT

The authors would like to acknowledge Dr. Francesc Carreras (from Hospital de la Santa Creu i Sant Pau) and Michael Markl (from the Departments of Diagnostic Radiology and Medical Physics, Freiburg, Germany) for their contributions and the support give us to do this work. The medical images used during this work are from the Departments of Diagnostic Radiology, Medical Physics; Neurology and Clinical Neu-
6 REFERENCES

[10] R. Löhner, J. R. Cebral: From medical images to cfd meshes. 8th International Meshing Round table, 1999