A Framework to Visualize and Interact with Multimodal Medical Images

Isabel Harb Manssour [1,2], Sérgio Shiguemi Furuie [3], Luciana Porcher Nedel [1,2] and Carla M. Dal Sasso Freitas [1]

[1] UFRGS - Universidade Federal do Rio Grande do Sul
[2] PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul
[3] InCor - Instituto do Coração

Abstract

The simultaneous use of images obtained from different sources is common in medical diagnosis. However, even though the quality of these images has been improving, the integration of multimodality data into a unique 3D representation is still non-trivial. To overcome this problem, multimodal visualization techniques provide better insight by finding suitable strategies to integrate different characteristics of multiple data sets into a single visual representation. This paper describes a framework for interactive multimodal visualization of 3D medical images, focusing on the multimodal visualization model and the requirements and open issues for the development of such systems. A short overview of multimodal visualization systems and techniques is also presented.

Keywords: Multimodal Visualization, Registration, Interaction Tools, and Medical Imaging.

1 INTRODUCTION

With the evolution of image acquisition technology in terms of resolution and tissue distinctiveness, the capacity and fidelity of image diagnosis were further extended. Several image acquisition modalities have been used for years to facilitate the medical diagnosis, e.g. Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET). While some modalities such as PET help to determine some functions, images from CT and MRI aid in the identification of anatomic structures. In summary, these modalities show different, complementary and/or partially overlapping aspects of the examined anatomy and function [1, 2, 3].

Although the increasing use of information acquired from multiple sources, the correlation of multivariate data into a 3D representation of the patient is extremely difficult, time-consuming and error-prone. Therefore, the integration of images from multiple modalities has rapidly evolved into an important area of research called multimodal visualization. The development of these new visualization techniques is concerned with the proper integration of images obtained from different modalities or from the same modality at different times. Systems that support this kind of application can combine functional and metabolic information with anatomical data, and increase the confidence of the observers in the location of a functional abnormality in relation to the anatomy [4].

Some good examples of applications that might benefit from multimodal visualization are: analysis of regional brain activity in patients suffering from schizophrenia; observation of tumor volume response to treatment; radiotherapy treatment planning; and surgical planning [2, 3, 4, 5]. This is a wide research area that allows a detailed and quantitative study of human body structures. The need for a framework for interactive multimodal visualization comes from the usefulness and clinical importance of the integrated display of functional and anatomical images in several medical applications. Also, despite of the evolution of volume rendering techniques, there are several open issues regarding the development of an integrated, flexible, extensible and portable system capable of solving a large range of visualization problems [1], since the majority of the available systems can not be extended and/or have a small set of tools that execute only in a specific platform.

The main goal of this paper is to present the design of a system architecture to integrate several tools, such as: segmentation, registration and interactive multimodal visualization. The description of a new technique to visualize inner structures of multimodal data sets, under development, is also included. This work is part of a large framework designed specifically for medical applications, which guarantees software reuse and integrates existing tools [6]. Section 2 presents a short survey describing the requirements and tools related to the implementation of such systems. The proposed framework is described in Section 3. Section 4 describes how visualization and interaction problems are solved within the framework. Algorithms and techniques that are being developed for registration and visualization are described in Section 5. Section 6 presents a comparison between some existing systems and the proposed framework. Finally, some conclusions and future work are presented in the last section.

Authors' Adresses: UFRGS - Universidade Federal do Rio Grande do Sul, Instituto de Informática/PGCC, Caixa Postal 15064, 91501-970 Porto Alegre, RS, Brazil, Tel: +55 (51) 316-6842, 316-7037 or 316-6811, Fax: +55 (51) 319-1576, {manssour, nedel, carla}@inf.ufrgs.br; PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática, Av. Ipiranga 6681, Prédio 30, Bloco 4, 90619-900 Porto Alegre, RS, Brazil, Tel: +55 (51) 320-3558, Fax: +55(51) 320-3758, manssour@inf.pucrs.br; InCor - Instituto do Coração, HC-FMUSP, Unidade de Pesquisa e Desenvolvimento, Serviço de Informática, Av. Dr. Enéas de Carvalho Aguiar, 44, 05403-000 São Paulo, SP, Brazil, Tel: +55 (11) 3069-5544, Fax: +55 (11) 3069-5311, Sergio.Furuie@incor.usp.br.

2 REQUIREMENTS

The first and fundamental step to generate images from multimodal volumes (as the ones shown on Figure 1) consists of bringing the involved modalities into spatial alignment, a procedure called *registration*. After registration, a *fusion* step is required for the simultaneous display of the two data sets. As reported by Maintz [1], it is important to emphasize that the terms registration and fusion, as well as matching, integration and others, appear with different meanings in the literature, either referring to a single step or to the whole integrated process. In this work we are considering the fusion as part of the multimodal visualization process. After visualization, *interaction* techniques and tools are used to provide volume exploration. In the following subsections, these three important requirements for interactive multimodal visualization are shortly described.

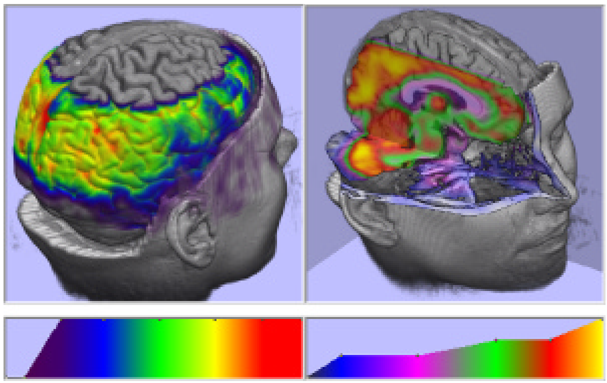


Figure 1: Examples of multimodal visualization generated from MRI and PET data, using RenderVox system [7].

2.1 Registration

Registration is a fundamental task in image processing used to match two or more images, or volumes, obtained at different times, from different sensors and scanners, or from different viewpoints. Simply, it consists of a process that maps pixels from one image, called *reference image*, to pixels in another image, called *test image*. The purpose of this operation is to represent information pertaining to the same object of interest to a common coordinate system. As mentioned by some authors [8, 9], registration is often necessary to integrate information taken from different sensors, or find changes in images taken at different times or under different conditions.

Several image registration techniques have been studied and developed in the last few years. Some differences among them can be: dimensionality (2D/2D, 2D/3D or 3D/3D); nature of registration basis (e.g. artificial objects introduced into the image, or voxel property); variety of modalities involved (e.g. mono or multimodal). The majority of them try to improve final results in performance, accuracy and reliability using different approaches [1, 8, 10, 11].

The analysis of current literature shows that there is a preference in the development of voxel property based registration methods that operates directly on the image gray values. In the specific case of multimodal image registration, researchers have focused on ***Mutual Information*** **(MI)** as the most suitable voxel property based image registration technique [4, 12, 13]. This technique is based on information theory and works directly with image data [12, 13, 14, 15, 16]. Registration is achieved by the adjustment of the relative position and orientation of the images until the mutual information between the images is maximized [17, 18, 19].

2.2 Interactive Multimodal Visualization

The integration of different volumes into a single visual representation (data intermixing or *fusion*) is necessary after the registration step. Since multimodal images are formed by merging complementary features of different volumes, possibly from different modalities, this fusion step is essential. Several approaches for this kind of visualization have been proposed and some of them are briefly described here.

The **Linked feature display** technique is used to obtain a 3D integrated display. In this case, there is a correlation between a 3D location and the equivalent position in a 2D image of each acquisition modality. The images are presented in multiple windows with separate controls for each one. This simple technique can be useful when the display is extended with a linked cursor indicating corresponding locations over the image slices of different modalities. For example, as a mouse-driven line-cursor is moved through a 3D model of the brain, one or more cross-sectional images of the corresponding section, e.g. MRI and PET, are updated on the right side of the screen (Figure 2) [4, 20].

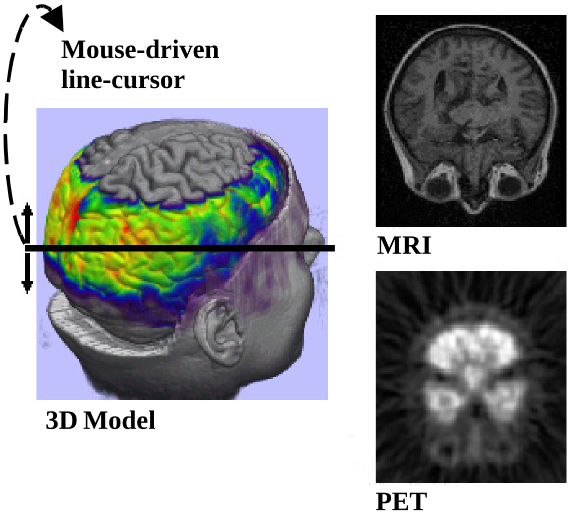


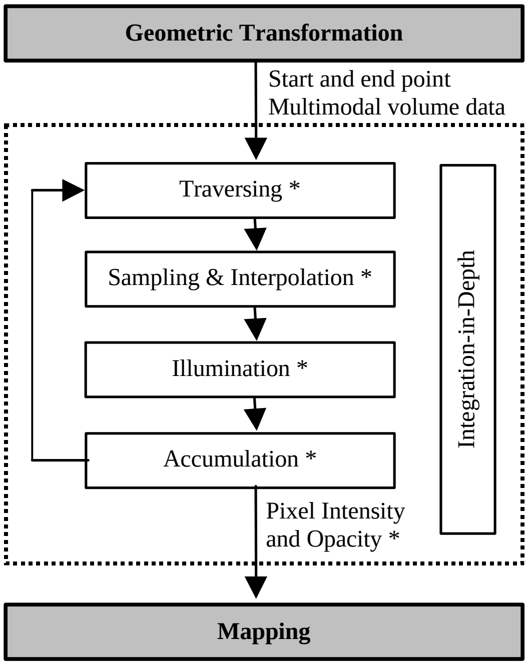
Figure 2: Example of linked feature display.

The **display of cut planes** is a common and powerful investigation technique in volume visualization of single modality. As an extension, the **multimodal cut plane display** has been also extensively used. One application example is a volume visualization of the brain from MRI with a cut plane representing functional information. Silva [7] implemented the display of cut planes, as illustrated in Figure 1 (right image). In this case, anatomical MRI data allows volume visualization, while functional PET data add colors to the cut plane that traverses the volume.

Surface texturing and mapping techniques integrate the information by mapping parts of the functional information from the volume onto a surface. Zuiderveld and Stokking [4, 21] implemented an algorithm using this technique for integrated visualization. In their first implementation, called Normal

Projection Technique [21], the functional values were mapped onto a surface extracted from anatomical volume data, e.g. MRI. A second implementation, called Normal Fusion Technique [4], included the use of the HSV color model, allowing the manipulation of color tables by the users in accordance to their perception and preferences.

In the **integrated data display** technique [22], volumetric structures derived from various modalities are integrated into one data set and, subsequently, displayed by standard rendering techniques. In this case, the data intermixing could be done in different levels of the rendering pipeline. Cai and Sakas [22] presented three levels of data intermixing in direct multimodal volume rendering. Their algorithm is based on the classical *ray casting* technique [23] and data flow in the rendering pipeline through three different stages: geometric transformation, integration-in-depth, and mapping, as presented in Figure 3. Data intermixing may be performed in different steps in the Integration-in-Depth stage: image level, accumulation level, or illumination model level.



** Steps involved in multimodal volume rendering*

Figure 3: Multimodal volume rendering pipeline [22].

The **Spectral Volume Rendering** technique [24] is based on direct volume rendering, and uses the full color spectra and physically realistic light/matter interaction models. The difference between this technique and traditional ray casting algorithms is that it simulates light interaction with the materials inside a voxel to calculate the final color. In this approach, a material density is assigned to each voxel value, instead of a RGB color and opacity. Several materials are used for cases of multimodal images.

2.3 Interaction Tools

Besides the integrated display techniques described in Section 2.2, it is important to provide ways to manipulate the data; extract measurements and functional information from the final image (quantification); and allow different types of visualization and navigation inside the structures. Some common interaction tools include:

- selection and manipulation of different regions and structures;
- cross-sections in any volume direction;
- cut planes and cut volumes;
- specification of different color and opacity tables;
- manipulation of the integrated volumes (e.g. zoom and rotation);
- parameters setting for inner structures visualization;
- quantification of selected structures, to obtain measurements information.

It is important to remember that the computational cost to perform all these tasks is very high, and often it is not possible to generate the results in real-time. Therefore, to allow interactive data manipulation, several acceleration techniques have been developed [7, 23, 24, 25, 26, 27]. Since the calculation of each ray in volume rendering algorithms is independent, another strategy to improve the performance is to take advantage from computers with parallel architecture, or execute the algorithm in several computers or workstations forcing a distributed parallel program implementation [10]. Recently, new rendering techniques are also being developed to take advantage of the imaging and texture mapping subsystem of graphics computers [28] and PC graphics accelerator boards [29].

Some types of interactions also require a pre-processing step. For example, a segmentation step becomes necessary to isolate a structure from the rest of the data set to make it transparent. It may also be necessary in multimodal visualization systems for visualization, registration and measurement extraction. Segmentation is a fundamental step in quantification and volume exploration techniques when the user points at some structure in the image and obtains several data about it, as dimension and functional information. This led us to conclude that building a multimodal visualization system involves the integration of tools such as registration, segmentation, and optimized interactive visualization techniques.

3 FRAMEWORK DESCRIPTION

One of the largest challenges in medical imaging is the development of strategies to integrate registration, segmentation, manipulation, and visualization of multimodal image sets. The pre-requisite step to match images acquired separately, as well as a reliable segmentation step that identifies and classifies interesting features in the data set, is essential to allow the visualization process. Optimized rendering algorithms, methods for inner structure visualization (Section 5.3), and interaction tools are also important for multimodal visualization systems. In this Section we specify an architecture for a multimodal, interactive visualization framework that integrates these tools, including the inner structure visualization algorithm that we are developing (Section 5).

We have considered the following requirements for the development of a new system: suitable features that an interactive multimodal visualization system should have (Section 2); possibility of using some of the optimized algorithms already presented in the literature; software reuse; and the fact that interactive applications should be easy to use and consequently, designed for a specific application or group of users. Based on these assumptions, we developed a conceptual model of a framework to allow easy and fast design and implementation of medical visualization and exploration systems.

The model is based on the *Model-View-Controller* (MVC) pattern [30] and is described here using the *Unified Modeling Language* (UML) [31]. MVC consists in a triad of classes extensively used in interactive systems to build user interfaces. Since the recent trend nowadays is to create systems with a high degree of user interaction, the majority of classes were modeled considering user interaction. For example, a manual segmentation is difficult and tedious, but without the know-how of the user it is very difficult to select the structures correctly. This is the reason because interactive segmentation is being considered very efficient nowadays [24].

The proposed architecture integrates registration, segmentation, and interactive visualization of multimodal data sets. Figure 4 shows a simplified UML description of this conceptual model. Object orientation allows easy integration of existing tools as well as extension to include new tools.

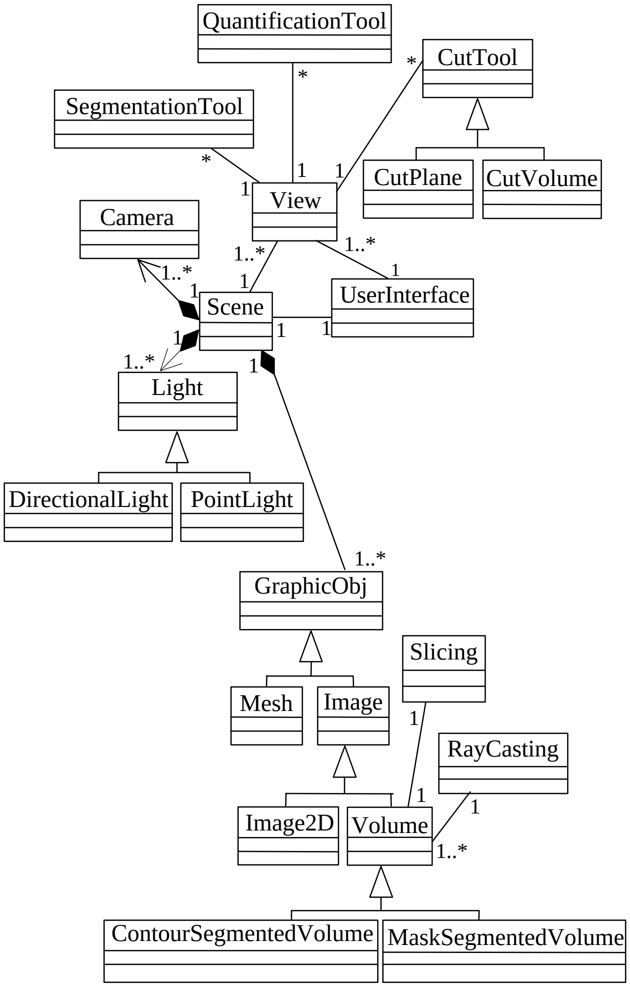


Figure 4: Simplified conceptual model.

In accordance to Figure 4, and taking into account the MVC pattern, the *UserInterface* class is the system controller, responsible for event management, as for example when the user click in a button or change a slider value. The *View* class is responsible for data presentation and could also manage some mouse events that are detected inside the viewport. In this way, the model, here represented by the *Scene* class composed by *Camera*, *Light* and *GraphicObject* lists, is totally independent of specific platforms. The detailed behavior of these relationships is discussed in Section 4.

Interaction tools, such as *CutTool*, *SegmentationTool* and *QuantizationTool*, are associated with the *View* class, since they need a graphical representation to be manipulated by the user, and invoke image generation and model changes. The *View* class is able, for example, to request the display of a cut plane and to provide the list of active graphic objects for an interaction tool. Since after the use of a segmentation tool the resultant segmented volume has to be saved and loaded, *Volume* class is being extended, to originate two new classes (Figure 4): *MaskSegmentedVolume* and *ContourSegmentedVolume*. In accordance with the applied segmentation technique, mask volume or marking contours, respectively, these classes could be instantiated or not.

The *RayCasting* class has a set of methods that implements different "getters" for volume visualization algorithms (e.g. intermixing, side by side, MIP[1] - Maximum Intensity Projection) and inner structure data. *Slicing* class is responsible for the slice generation, i.e., axial, coronal and sagittal planes. Registration is just a method of the *Volume* or *Image2D* class, which are also responsible for data loading. Color and opacity tables are attributes from the *Volume* class. *Mesh* class will be useful for hybrid visualization, which includes techniques for simultaneous display of volume data and geometric models. To simplify the diagram presented here, some basic implemented classes such as *Point*, *Vector*, *Line*, *Matrix* and *Color* are not shown.

It is also important to point out that, in order to develop flexible, extensible and as portable as possible systems (properties desired in multimodal visualization applications [4]), we chose to implement this framework using free software, standard C++ programming language and OpenGL. C++ has been shown to be the most suitable programming language, since it is possible to use just the ANSI libraries and compile the same code in different platforms, always guarantying a good performance. Zuiderveld et al. [10] presented an evaluation of its utilization in the development of a multimodal visualization system. OpenGL is used for the 3D interface taking advantage of the graphics hardware. For the GUI development we are using the FLTK user interface toolkit [32]. FLTK is portable, developed over the GNU Library General Public License and has an optimized code, regarding performance and code size.

4 INTERACTION AND VISUALIZATION MODEL

The MVC implementation separates the functional core from the user interface in interactive systems. The core usually remains stable, since is based on its functional requirements. The user interface, however, is often subject to change and adaptation. Then, it is essential to develop an architecture that supports the adaptation of user interface without causing major effects to application-specific functions or the data model underlying the software [30]. Hence, the class library was designed in such a way that the entire interface is concentrated in the *UserInterface* and *View* classes, while the other tools and functions remain

---

[1] MIP is a very simple composition operation accomplished by choosing the greatest scalar value along the ray [27].

completely independent from the interface. Using these classes, we provide the fundamental structural organization for an interactive multimodal visualization system.

Medical data visualization applications usually need more than one view. A classical example is the presentation of the axial, coronal and sagittal planes jointly with the reconstructed organ from CT or MRI volume data. Since in this case we have a one-to-many dependency between objects, when one object changes state, all its dependents have to be notified and updated automatically [33]. In our framework, the *UserInterface* object and all *View* instances are dependent from model modification. A *View* object, as well as a *UserInterface* object, acts as an observer: it is constantly watching the model, i.e. *Scene*. When the model changes itself, the *View* object updates the display. In fact, there is a change-propagation mechanism that maintains a registry of the dependent components within the model. Changes to the state of the model trigger the change-propagation mechanism, which is the only link between *Scene*/*GraphicObject*, *UserInterface*, and *View* classes [30].

To illustrate the interaction among objects, a collaboration diagram is presented in Figure 5. In this example the user changes the visualization type to MIP, e.g. clicking on a button. At first the event is detected (1); then one message is sent for the *GraphicObject*, more specifically to a *Volume* instance (2), notifying that it has to modify itself. After changes, a notification is sent to the *Scene* (3), which then notifies the changes to the *View* class (4) that is responsible to get the new object state (5) for image re-exhibition, and to the *UserInterface* class (6) that can also be modified if one specific feature can be available or not after object changes (7).

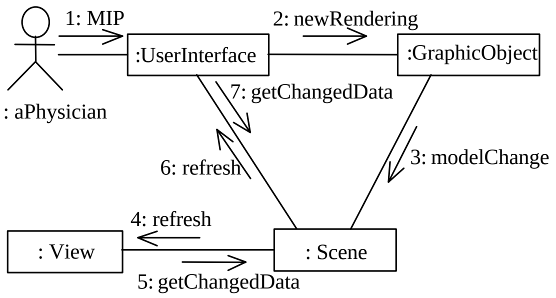


Figure 5: Collaboration diagram.

Beyond basic classes, such as *Point*, *Vector* and *Line*, another ones were also already implemented: *View*, *UserInterface*, *Scene*, *Camera*, *Light*, *GraphicObject*, *Volume* and *Slicing*. *CutTool* class is in development, as well as *RayCasting* class, which already has several methods implemented for the basic ray casting algorithm. In fact, we are now improving the visualization algorithms (see Section 5) and studying in detail how the interaction with *CutTool* objects will work. A visualization system prototype was developed to validate the interaction among objects, algorithm implementation and testing,.

Figure 6 shows the user interface designed for the prototype under development. The visualization system is composed of two main windows: the first one (shown in Figure 6a) contains just some system functions, represented by six icons, while the second window is dedicated to 2D and 3D visualization. Clicking with the mouse over one of these icons, a specific function will take place (from left to right): image segmentation; two-dimensional image visualization; mono and multimodal volume visualization; help window; and system exit.

For example, when the user selects the third icon (monomodal volume visualization) on the functions window, a dialog box opens, allowing the selection of a file name. This dialog is closed automatically after that, and a new visualization window is opened. This window, shown in Figure 6b, presents a default volume visualization and a set of icons allowing the selection of a new set of functionality. From the point of view of the application programmer, it is important to point out that, to allow this procedure, instances of the following classes should be created:

- *UserInterface*: responsible for buttons, sliders and the display of the views;
- *View*: three instances for the orthogonal planes images and one for volume visualization;
- *Scene*: composed by one Volume class instance, and four Camera class instances (one for each view).

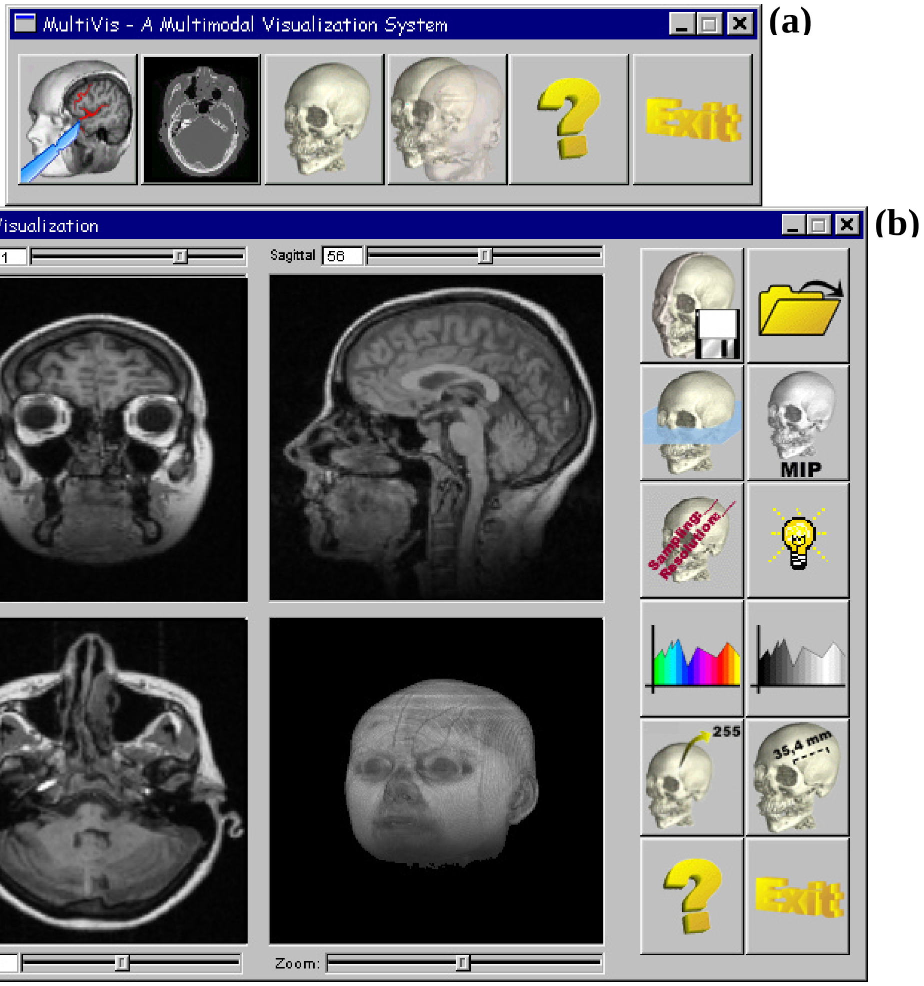
We are working on the design of a user-friendly interface, which intends to be quite simple and easy to use, with buttons and options distributed in several “levels”. As our goal is to provide useful tools for professionals that are not comfortable with this kind of application, we are trying to avoid the specification of numerical parameters in a single step to generate an image. Depending on the selected features, new windows can be opened, but with the results displayed always at the same view or set of views. As described before, for default visualization the user do not have to provide any parameter. However, after that first step the user can change the parameters selecting the appropriate icons or interacting directly into the view. The icons that appear in Figure 6b are (left to right, top to bottom): save volume visualization image; open another volume data file; generate cut plane view; allow MIP visualization; change rendering parameters; set light sources; manipulate the color table; manipulate the opacity table; extract functional information; extract the measurement information; help; and close the window.

5 IMPLEMENTATION

The framework easily supports the addition of other algorithms. Since the system architecture is already defined, with all basic classes implemented, we are now working on the development and adaptation of several algorithms necessary for system implementation. Due to the group experience in previous works, e.g. Figure 1 [7], we choose to develop the visualization algorithms based on direct volume rendering techniques. This choice avoids, for now, the implementation of a segmentation tool.

5.1 Registration

One indispensable pre-processing step for multimodal visualization is registration. As described in Section 2.1, the most used technique to allow registration is mutual information, and implementation of this algorithm to make automatic registration is under development. We are also studying some alternatives to improve its performance, e.g. ask for the user to interactively make an initial matching, in order to reduce the computation to maximize the mutual information [12, 14].



Coronal 91
Sagittal 56
Axial 58

Figure 6: Prototype interface.

5.2 Multimodal Visualization

Since volume visualization algorithms will be implemented as extensions to the classical ray casting algorithm, the multimodal visualization algorithms will also be based on the integrated data display technique (Section 2.2). Some improvements on the illumination model, parallel and perspective projection, among other optimization techniques are being studied. At present, we are integrating the algorithms and optimization techniques from RendexVox (a very efficient in-house system developed by Silva [7]) into the presented architecture. Results are shown in Figure 1, where the anatomical modality (MRI) provides a frame of reference for the correct interpretation of the functional information (PET) represented by colors in 3D-space. The images shown in Figure 1 were generated using data from different patients.

The multimodal visualization algorithm, including the visualization of inner structures we are developing will be presented now. As described by Cai and Sakas [22] (Section 2.2), data intermixing may involve different steps of the rendering pipeline: image, accumulation or illumination level. Silva [7] developed an algorithm where one volume is used for shape representation and the other one for coloring (left image of Figure 1). In this case, the rays are casted simultaneously in both volumes (e.g. PET and MRI). The color classified from the first volume (MRI) is combined with the color classified from the second volume (PET) at the same position, and this mixing is weighed using the opacity determined (by the user) for the second volume. The sampled opacity composition in the ray just use the values from the first volume, in such a way that one volume defines the shape and the other one the colors.

We decided to first implement image level intermixing, which is simpler and do not need modifications in the rendering pipeline. Later, we will implement the accumulation level opacity intermixing. With these two types of intermixing implementation, it will be possible to evaluate the performance and quality of the generated images to choose the best alternative.

5.3 Inner Structure Visualization

A current topic of research in multimodal visualization area is the development of a tool to allow the visualization of inner structures. Physicians are often interested in visualizing and quantifying isolated features in volume data, e.g. deep vessels and tumors. The ability to demonstrate a lesion in relation to the surrounding normal anatomy is also very useful, specially when the user can select a region of interest clearly defined along the neighboring structure [34]. Due to the importance of this kind of visualization, we are going to develop a new technique for the visualization of inner structures, not just based in transparency levels.

We are extending the *Confocal Volume Rendering* (CVR) technique to work with multimodal volumes and considering the possibility to define the required parameters interactively [34]. This technique enables the user to visualize inner structures in one data set just controlling physically defined parameters, without performing segmentation. Since image level intermixing does not cause changes in the rendering pipeline, and accumulation level intermixing changes are not so significant, it seems that this extension is feasible.

Without a pre-processing step like segmentation, CVR preserves surrounding structures and avoids the insertion of artifacts. Moreover, it reduces the time for visualization and allows to show referential landmarks outside the objects (e.g. surgical planning). Compared to volume of interest and oblique sectioning interior visualization methods, CVR does not sacrifice valuable information because clipping is not performed [34].

Finally, although the ray casting algorithm is still very computationally intensive, this is not being considered as a limiting factor. This problem has been reduced with the development of parallel architectures and graphics acceleration boards, as well as with the special purpose graphics hardware designed specially for fast manipulation of volume data [21, 28, 29]. In fact, since each ray is processed independently, we are modeling the inclusion of a new set of classes to allow system execution in a parallel processing system available at the Research Center in High Performance Computing at PUCRS [35]. We are going to use a *cluster* composed of sixteen computers working cooperatively [36], and we expect it will be possible to interact and generate the images in real-time. The only disadvantage in this case is that this set of classes will be totally dependent of an architecture that usually is not available at the physicians environment work.

5.4 Extending the View Class

Our framework was conceived, using object-oriented programming and the MVC pattern, which allows the design of a great number of different user interfaces. This is obtained by extending the *View* and *UserInterface* classes. In the *View* class (Figure 6), for example, we extended it to handle only ray casting techniques. Another extension of the *View* class, called *OpenInventorView*, is based on OpenInventor and OpenGL. Currently we only implemented the methods to visualize geometric data, as in Figure 7, where we show a reconstructed skeleton designed using the SoTriangleMesh OpenInventor class. We are now studying alternatives to display volumetric data using another OpenInventor-based extension of the *View* class, and a good possibility is the approach presented by Sommer et al. [37].

6 COMPARISON WITH OTHER SYSTEMS

Some multimodal visualization systems have already been developed and are described in the state-of-the-art literature. In this section, the functionality and features of such systems, as well as the algorithms developed, will be overviewed and compared to our approach.

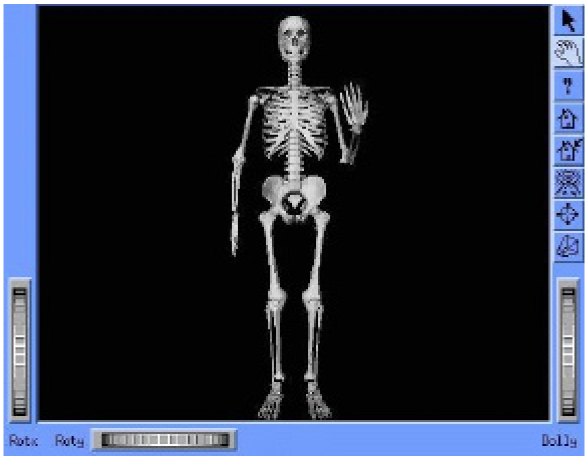


Figure 7: An example of using *OpenInventorView* class instance.

Confocal Volume Rendering (Section 5.3) seems to be a very interesting alternative for inner structure visualization, since it does not depend on a segmentation step and reduces the visualization time. We are proposing an extension of this algorithm to work with multimodal volumes, which is a new approach in the inner structure visualization field.

The software package called **ANALYZE™** [38], developed at Mayo Clinic, provides an integrated set of display, manipulation and measurement tools for detailed investigation and evaluation of three-dimensional biomedical images. It was designed and carefully programmed to be highly efficient, user-interactive and generic, but it runs on standard Unix workstations. This is the only limiting factor that is being considered in our framework.

VROOM (*Volume Rendering by Object-Oriented Methods*) is an object-oriented, flexible, extensible and portable software architecture, for the integrated visualization of multimodal volumetric data sets [4, 21]. However, segmentation and registration are not addressed, and an adequate preprocessing of the data is assumed. In our architecture we are providing ways to integrate these several tools. Moreover, this system also runs only in Unix platforms.

Another software package, **3Dbench**, offers general routines to manipulate and visualize volume graphics and provide an interactive slicer and volume renderer for the 2D and 3D visualization of unprocessed, segmented, multimodality and multi-channel volume images and any combination of them [24]. Measurement tools were also developed, but it does not provide tools for interactive segmentation and visualization of local features (e.g. inner structure visualization).

A new approach that allows versatile fusion operations was developed by **Hastreiter and Ertl** [28] for the simultaneous and interactive visualization of registered data sets. The registration algorithm implemented is based on mutual information. In order to integrate registration and visualization, the developed approach allows to render two registered data sets using hardware accelerated 3D texture mapping available in desktop graphics workstations. They based their work on OpenInventor and OpenGL. The differences from our framework are: the user can use OpenInventor, as well as other similar toolkit; instead of using hardware accelerated 3D texture mapping we are planning to work with a parallel processing system.

As described above, several tools and interaction techniques are available in current visualization systems, but their integration as a unique system is not available yet. Finally, several open issues in multimodal visualization area, e.g. the development of a user-friendly interface and new kinds of interactive visualization techniques show that there are still a number of research topics to be explored.

7 CONCLUSIONS AND FUTURE WORK

This paper presented a brief description of the multimodal visualization research area, which has the goal of finding suitable strategies to integrate important characteristics of multiple data sets into one image such that better insight can be provided [21]. We focused on the definition and development of an interactive multimodal visualization framework, which includes an algorithm for the visualization of inner structures.

Briefly, we have been implementing the visualization algorithms described in Section 5, and we would like to emphasize that the inner structure visualization technique we are going to develop has at least one advantage: it does not depend on a segmentation algorithm. Segmentation is not an easy task, specially for brain structures, and often add artifacts. Moreover, the development of such kind of multimodal visualization algorithm has not been thoroughly explored yet. Probably, after the algorithm implementation, many new directions will arise for multimodal visualization algorithms.

The development of new interaction techniques, such as linked feature display (Section 2.2), is being studied together with the design of user-friendly interfaces. The majority of medical visualization systems has a complex interface, which has lot of parameters to fulfil in just one window. Since the physicians do not have time to spend trying to understand their behaviour, we are developing a simple and easy to use prototype interface, with several "levels", each with new parameters that are provided accordingly to the user needs, as shown in Figure 6.

Analyzing current work, we conclude that some systems were already developed to allow interactive multimodal visualization, but a big challenge still remains: the development of techniques to integrate registration, quantification, interactive segmentation and visualization, including the visualization of inner structures in multimodal data sets. According to Johnson [39] interactive visualization systems need to be modular, easy to extend and portable to different hardware. As described in Section 3, our framework seems to fit these requirements. Moreover, since we are providing a class library, system extension became easy: the users just have to extend our classes to include new visualization and interaction techniques.

Previous work has also focused on the development of optimized registration and visualization algorithms, in order to allow user interaction in real-time. With the proposed architecture and visualization algorithm, it will be possible to use these optimized algorithms, in such a way that the user can integrate them with several visualization and interaction tools without having to rebuild entire modules.

Acknowledgements – We would like to thank all the staff of our lab involved on the development and implementation of the framework classes, Marcelo Rodrigo Silva for the RENDERVOX images, Anderson Maciel for the OpenInventorView class image, Marcelo Hideki Yamaguti for reviewing the UML diagrams, and João Luis Comba for helpful comments and text reviewing. This research was partially supported by CAPES, CNPq, FAPERGS and UFRGS.

8 REFERENCES

[1] J. B. A. Maintz and M. A. Viergever, Survey of Medical Image Registration. *Medical Image Analysis*, 2(1), 1998, 1-36.

[2] J. P. W. Pluim and J. B. A. Maintz, *Registration of Multimodal medical images*. Available at http://www.cv.ruu.nl/ Research/Registration/registration-frame.html(March 2000).

[3] G. Q. Maguire Jr., M. E. Noz, H. Rusinek et al. Graphics Applied to Medical Image Registration. *IEEE Computer Graphics & Applications*, March, 1991, 20-28.

[4] R. Stokking, *Integrated Visualization of Functional and Anatomical Brain Images* (Netherlands: Universiteit Utrecht, 1998, Ph.D. Thesis).

[5] R. A. Robb, Visualization Methods for Analysis of Multimodality Images. In: R. W. Thatcher, M. Hallet, T. Zeffiro et al. (eds). *Functional Neuroimaging: Technical Foundations*. Academic Press, 1994.

[6] Computer Graphics and Image Processing Group, *VPat - Visualization and interaction with Virtual Patients*. Available at http://www.inf.ufrgs.br/pesquisa/cg/projects.html#vpat (July 2000).

[7] M. R. M. Silva, *Ray Casting Based Volume Visualization Techniques* (Porto Alegre: Instituto de Informática/UFRGS, 2000, Master Thesis).

[8] L. G. Brown. A Survey of Image Registration Techniques. *ACM Computing Surveys*, 24(4), 1992, 325-376.

[9] J. K. Udupa and G. T. Herman, *3D Imaging in Medicine* (CRC Press LLC, 2000, 2nd ed.).

[10] K. J. Zuiderveld, A. H. J. Koning, R. Stokking, et al. Multimodality Visualization of Medical Volume Data. *Computers&Graphics*, 20(6), 1996, 775-791.

[11] J. West, J. M. Fitzpatrick, M. Y. Wang et al. Retrospective Intermodality Registration Techniques for Images of the Head: Surface-Based Versus Volume-Based. *IEEE Transactions on Medical Imaging*, 18(2), 1999, 144-150.

[12] A. Collignon, *Multi-Modality Medical Image Registration by Maximization of Mutual Information* (Belgium: Katholieke Universiteit Leuven, 1998, Ph.D. Thesis).

[13] J. B. A. Maintz, E. H. W. Meijering and M. A. Viergever, General Multimodal Elastic Registration Based on Mutual Information. In *Proceedings of SPIE'98 - Annual Meeting International Symposium on Optical Science and Technology* (San Diego, California, 1998), pages 144-154.

[14] F. Maes, *Segmentation and Registration of Multimodal Medical Images* (Belgium: Katholieke Universiteit Leuven, 1998, Ph.D. Thesis).

[15] J. B. A. Maintz, *Retrospective Registration of Tomographic Brain Images* (Netherlands: Universiteit Utrecht, 1996, Ph.D. Thesis).

[16] J. P. W. Pluim, J. B. A. Maintz and M. A. Viergever, Interpolation artefacts in mutual information based image registration. *Computer Vision and Image Understanding*, 77(2), 2000, 211-232.

[17] A. Collignon, F. Maes, D. Delaere, et al. Automated multi-modality image registration based on information theory. *Information Processing in Medical Imaging.* Kluwer Academic Publishers, 1995, p. 263-274.

[18] P. Viola and W. M. Wells III, Alignment by maximization of mutual information. In *Proceedings of International Conference On Computer Vision*, pages 16-23. IEEE Computer Society Press, 1995.

[19] W. M. Wells III, P. Viola, H. Atsumi et al. Multi-Modal Volume Registration by Maximization of Mutual Information. *Medical Image Analysis*, 1 (1), 1996, 35-51. Available by ftp.ai.mit.edu:/pub/sw/papers/mia-95-dist-color.ps.gz.

[20] D. N. Levin, X. Hu, K. K. Tan et al. The Brain: Integrated Three-Dimensional Display of MRI and PET Images. *Radiology*, 172(3), 1989, 783-789.

[21] K. J. Zuiderveld, *Visualization of Multimodality Medical Volume Data using Object-Oriented Methods* (Netherlands: Universiteit Utrecht, 1995. Ph.D. Thesis).

[22] W. Cai and G. Sakas. Data Intermixing and Multi-Volume Rendering. *Computer Graphics Forum*, Cambridge, 18(3), 1999, C359-C368. Proceedings EUROGRAPHICS, Milan, Italy.

[23] M. Levoy, Volume Rendering - Display of Surfaces from Volume Data. *IEEE Computer Graphics & Applications*, 8(3), 1988, 29-37.

[24] H. J. Noordmans, *Interactive Analysis of 3D Microscope Images* (Netherlands: Universiteit Utrecht, 1997, Ph.D. Thesis).

[25] M. Levoy, Efficient Ray Tracing of Volume Data, *ACM Transactions on Graphics*, 9(3),1990,245-261.

[26] M. Levoy, A Hybrid Ray Tracer for Rendering Polygon and Volume Data, *IEEE Computer Graphics and Applications*, March 1990, 245-261.

[27] B. Lichtenbelt, R. Crane and S. Naqvi, *Introduction to Volume Rendering* (Prentice Hall, 1998).

[28] P. Hastreiter and T. Ertl, Integrated Registration and Visualization of Medical Image Data. In *Proceeding of Computer Graphics International'98*, (Hannover, Germany, 1998), pages 78-85. IEEE Computer Society Press, 1998.

[29] C. Rezk-Salama, K. Engel, M. Bauer et al. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In *Proceedings of SIGGraph/Eurographics Workshop on Graphics Hardware 2000* (Interlaken, Switzerland, August 21-22, 2000).

[30] F. Buschmann, R. Meunier, H. Rohnert et al. *Pattern-Oriented Software Architecture, A System of Patterns* (John Wiley & Sons, 1996).

[31] C. Larman, *Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design* (Prentice-Hall, 1997).

[32] B. Spitzak, *FLTK - The Fast Light Tool Kit Home Page*. Available at http://fltk.easysw.com (March 2000).

[33] E. Gamma, R. Helm, R. Johnson et al. *Design patterns: elements of reusable object-oriented software* (Addison-Wesley, 1995).

[34] R. Mullick, N. Bryan and J. Butman. Confocal Volume Rendering: Fast Segmentation-Free Visualization of Internal Structures. In *Proceedings of SPIE'00 - Annual Meeting International Symposium on Optical Science and Technology* (San Diego, California, 2000), pages 144-154.

[35] CPAD - Research Center in High Performance Computing. Available at http://www.cpad.pucrs.br (April 2001).

[36] R. Buyya. *High Performance Cluster Computing: Architectures and System* (Prentice Hall, 1999).

[37] O. Sommer, A. Dietz, R. Westermann et al. An Interactive Visualization and Navigation Tool for Medical Volume Data. In *Proceedings of WSCG'98* (Plzen - Bory, Czech Republic, February 9 - 13, 1998).

[38] R. A. Robb, Visualization Methods for Analysis of Multimodality Images. In: R. W. Thatcher, M. Hallet, T. Zeffiro et al. (eds). *Functional Neuroimaging: Technical Foundations*. San Diego, CA: Academic Press, 1994. Available at http://www.mayo.edu/bir/reprints/VIS-METH.html (May 1999).

[39] C. Johnson, S. G. Parker, C. Hansen et al. Interactive Simulation and Visualization. *Computer*, 32(12), 1999, 59-65.