UNIVERSITY OF MAURITIUS RESEARCH JOURNAL – Volume 15 – 2009
University of Mauritius, Réduit, Mauritius

Research Week 2007

Patient specific 3D visualisation of human brain

Sharvesh. Baichoo*
Email: sharvesh.baichoo@gmail.com

R. K. Subramanien
Faculty of Engineering,
University of Mauritius
Email: rks@uom.ac.mu

Paper Accepted on 05 May 2009

Abstract

Over the past thirty years, computer graphics and visualization have been used more and more increasingly to add value to a wide variety of medical applications. The significant increase in computing power in recent years has allowed the development of powerful new 3D image analysis and visualization algorithms that promise to change the way medicine is practiced. In general visualization algorithms used include direct volume rendering (DVR) algorithms such as ray casting and surface-fitting (SF) algorithms such as marching cubes. Image data obtained from 3D MRI or CT scans contain much information about internal anatomy of patients and SF algorithms tend to discard these information. Hence DVR is a more effective means of presenting image data obtained from a 3D MRI or CT scan to the attending physician or radiologist. Current imaging devices provide dramatically increased data complexity and resolution. A three dimensional view of any part of the human body will give doctors a valuable insight into the organ. This work aims at providing such a tool with enhanced capabilities. The tool is aimed to provide facility to reconstruct patient-specific 3D visualization of the organ. The tool will target human brains and enable visualization of the human brain using medical images such as MRI and/or CT scan of the patient. The doctors will be able to perform various operations, like looking at the sections of the brain and will be able to perform a preliminary diagnosis with out intrusion into the brain. In this paper we present a review of the work done in the field of surgical simulation

Keywords: Volume visualization, soft tissue modeling, segmentation

**For correspondences and reprints*

1. INTRODUCTION

Over the past thirty years, computer graphics and visualization have been used more and more increasingly to add value to a wide variety of medical applications. One of the earliest examples reported was the three-dimensional visualizations of Computerized Tomography (CT) data in the mid 1970's [VIDAL 06]. Today, the medical profession can make use of various imaging modalities for diagnostic purposes. These modalities provide a rich source of data for further processing using computer graphics and visualization techniques. Applications include medical diagnosis, procedures training, pre-operative planning, telemedicine and many more.

The standard two-dimensional information given to doctors can be difficult to interpret. In fact doctors are usually given a large sheet of transparent film containing a series of two-dimensional images of the site around the disease, for example tumour. The doctor must then mentally reconstruct the three-dimensional location and shape of the tumour. A three dimensional view will give the doctor valuable insight. The development of fast and effective ways of presenting three dimensional medical visualizations is a very active field of research among computer scientists. The significant increase in computing power in recent years has allowed the development of powerful new 3D image analysis and visualization algorithms that promise to change the way medicine is practiced. In the coming years, the computer based tools used in the medical field must evolve from being fundamentally two-dimensional to become fully three-dimensional.

2. BASIC TECHNIQUES AND ALGORITHMS USED IN VOLUME VISUALISATION

[ELVIN 92] defines volume visualisation as being "the process of projecting a multidimensional data set on a two dimensional image plane for the purpose of gaining an understanding of the structure (or lack of structure) contained within the volumetric data". Volume visualisation consists of several steps, namely: filtering and segmentation, 3D image registration, soft tissue modelling. The filtering and segmentation algorithm are used as a pre-processing prior to applying visualization techniques. The purpose of 3D image registration, also known as image fusion, is to combine two or more image modalities of the same patient. Soft tissue modelling allows the simulation of the physical behaviour of soft tissue. All the above mentioned steps are discussed below. This section ends with a discussion of the main approaches to visualization of volumetric data in medical applications.

2.1 Filtering and Segmentation

Image data acquired from scanners may contain noise. Filtering techniques such as mean filters have been proposed to reduce the noise. The mean filter averages the value at each pixel over a local neighbourhood. The main problem with such techniques is that edge information between anatomical features may be lost since blurring of the boundaries of the anatomical features occurs after the application of the filters. Due to this limitation, an edge-preserving approach to noise removal should be used instead. One such approach that is widely used according to

[VIDAL 06] is the Anisotropic Diffusion Filtering (ADF). According to [SCHMI 05], like mean filtering, ADF reduces noise by smoothing pixel values similar to the mean filter. However, unlike mean filtering, ADF makes use of image gradients to prevent smoothing across edge areas.

Segmentation usually follows noise removal. The objective of the segmentation step is to identify the different parts of interest of the anatomy. After the segmentation process, each pixel will be labelled using an identifier that gives an indication of the type of the material. Typically, segmentation processes are semi-automatic as user involvement is often required to obtain the required segmentation result. [VIDAL 06] reports that segmentation is a major bottle-neck in clinical applications as the process is time consuming and results are difficult to reproduce due to user involvement in guiding the process.

One of the simplest segmentation techniques is thresholding. With thresholding, an image is partitioned based on its pixels' intensities. If a single threshold is used, the image will be divided into two segments. This technique can be useful in situations where for example tumour cells and healthy cells need to be identified. Another fundamental segmentation technique is region growing where a seed point is usually identified by the user. An automatic algorithm uses the seed point as input and builds up the region around that point up to its boundary.

[VIDAL 06] identifies a class of segmentation algorithms as one that requires the user to guide the process. One such approach is "Livewire". In livewire, a user first identifies a point on the boundary using the cursor. As the user moves the cursor, a curve is drawn between the identified point and the current position of the cursor. This curve is supposed to approximate the boundary. When the user is satisfied with the curve, the current cursor position is made fixed and then the user continues moving the cursor so as to form curves that satisfactorily represents the boundary.

In [VIDAL 06], segmentation techniques that make use of deformable models have also been identified. Usually a shape model is acquired through a training set. The model is modified such that it matches the image data. Instead of only learning shape, a model can be created such that it includes appearance of an object as well. References for segmentation techniques for specific parts of the anatomy, such as tracheo-bronchial tree of the lungs, which are difficult to segment have been provided in [VIDAL 06].

2.2 Image Fusion

Many image modalities, for example MRI (Magnetic Resonance Imaging), CT (Computed Tomography) scans, fMRI (Functional MRI), have been used on a patient due to the fact that the different modalities provide complementary information. For a successful integration of the information provided by the different modalities, the images need to be aligned to each other thus establishing a spatial correspondence between the different features seen on the different imaging modalities. Image fusion involves the simultaneous rendering of multiple modalities in a form most useful to the clinician. References for image registration techniques can be found in [VIDAL 06].

2.3 Soft tissue modelling and haptics

Visual feedback, haptic feedback, and interactions between medical devices and anatomical structures form part of simulation. We end up in an iterative process where collisions between objects are detected, deformation and collision response are computed, and the resulting state can be visually and haptically rendered.

The purpose of soft-tissue modelling is to enable the simulation of tissue behaviour. Soft tissue modelling is essential for second generation surgical simulators and above. In fact, according to [DELIN 04], surgical simulators can be classified into three generations. First generation simulators are concerned mainly with the anatomy, in particular the geometry of structures involved in a surgical intervention. These simulators produce a virtual representation of the patient within which users can navigate. The simulators offer a limited set of possible interactions. Such simulators have already been developed, some of which are commercially available. First generation simulators are mainly being used as complementary diagnosis tools and also as an aid in surgery planning. However, they are not well adapted to the simulation of surgical gestures.

Second-generation simulators model not only the geometric structures of the body but also model the physical properties of living tissues. With the introduction of biomechanical properties, realistic interactions between surgical instruments and soft tissues are possible. Thus, second generation simulators are appropriate to be used in the simulation of surgical gestures. The interaction of a soft tissue with surrounding bodies, such as a surgical instrument, can be decomposed into two tasks: collision detection and computation of interaction forces. Third-generation simulators provide the modelling of the functions of some organic systems such as the cardiovascular, respiratory, or digestive systems. In [DELIN 04], it has been said that very few third-generation simulators exist because of the difficulty to realistically describe the coupling between physiology and physics.

According to [DELIN 98], computation time and deformation accuracy are the two main constraints for the modelling of soft tissue. Usually, the more accurate a modelling technique is, the more computation time it takes. Based on the application, the required deformation accuracy and acceptable computation time taken by a modelling technique varies. For example, consider surgery planning. Surgery planning may require several trials and also the result obtained may need to be evaluated. Therefore a considerably low computation time with realistic deformations is desirable.

[VIDAL 06] classifies soft tissue modelling algorithms as being either geometrically based or physically based. In a geometrically based algorithm, such as free form deformation (FFD), an object is embedded within a lattice of a simple shape. Whenever the lattice is deformed, the object as well is deformed. Such methods are fast, however the deformations obtained can be unrealistic. Physically based models attempt to model the material properties and characteristics of the object. As compared to geometrically based models, they can produce more realistic deformations. An example of such an approach is the spring-mass technique.

With the spring-mass technique, objects are constructed as mass points (particles) connected with damped springs. Particles move whenever a force is applied. Some of the advantages of this technique are that it is a fast modelling technique and it is easy to implement. However, it can result in unrealistic behaviour for large deformations. Also it may not work well for harder objects such as bones. Another physically based approach is Finite Element Modelling (FEM). As compared to spring-mass technique, FEM has the capacity to produce results of greater accuracy. FEM describes a shape as a set of basic elements such as triangles, quadrilaterals and tetrahedral. FEM is well suited for computing accurate and complex deformation. However it is difficult to obtain real time performance on a moderately powerful workstation using FEM. Details on both the spring-mass technique and FEM can be found in [DELIN 98, VIDAL 06].

The immersion during a surgical simulation session is improved if the user interface can provide an haptic feedback on the simulation. [RUSPINI 97] proposes a haptic rendering system that allows for the efficient tactile display of graphical information. The system models contact constraints, surface shading, friction and texture through the use of a common high level framework. [RUSPINI 97] states that the haptic system should avoid preprocessing steps that reduces the system's interactivity. Also the haptic system should make use of graphical information such as surface normal and texture maps. The haptic system should provide a high-level interface library that encapsulates details of the haptic rendering process. [RUSPINI 97] proposes "HL", a new haptic interface library which allows haptic environments to be quickly and efficiently incorporated into graphics applications.

One vital aspect of the proposed haptic system is the use of a virtual proxy. The virtual proxy is similar to the god-object mentioned in [SAUPIN 08]. The god-object basically helps in tracking the position of the haptic device during simulation. [RUSPINI 97] separates the haptic and application/graphics processes. Thus the system runs on two computers, one acts as the haptic server while the other computer contains the client application. The haptic server receives high level commands from the client, tracks the position of the haptic device, updates the position of the virtual proxy, and sends control commands to the haptic device. Thus bottle-neck is placed on the haptic server's processor rather than the I/O channel.

An important part of the system is collision detection of the proxy object to obstacles along its path. To efficiently detect collisions, a hierarchical bounding representation for objects, based on spheres, are constructed. Two heuristics are used to determine the bounding sphere. The first one finds the smallest bounding sphere that contains the sphere of its two children. The second method examines leaf spheres and the centre is taken to be the mid-point of a bounding box already computed earlier. The radius is taken to be large enough to contain all the descendant leaf nodes. The method generating the sphere with the smallest radius is used for a given node.

[SAUPIN 08] argues that though current surgical simulators propose realistic physical models of the organs, they often oversimplify the models of contact by considering a simple contact point instead of an area of contact and rarely account for friction. [SAUPIN 08] proposes a similar idea to [RUSPINI 97], as [SAUPIN 08] also, proposes the separation of simulation from the haptic rendering. A simulation loop and a haptic loop therefore are processed simultaneously and each are set to work at different frequencies with the simulation loop working at a lower frequency. [SAUPIN 08] makes use of a god-object, which is a virtual object subject to the physical laws of the simulation that helps in tracking the position of the haptic device during the simulation and also makes it easier to compute force feedback.

The main steps in the algorithm proposed by [SAUPIN 08] are as follows: First the integration of soft tissues models without considering contact forces and the update of the instrument position. Next comes the collision detection algorithm which determines which points of the tissue surface are in contact with the surgical instrument. Thirdly, a collision response is computed and lastly the displacement field induced by contact forces is applied.

As per [MOSEG 05], a tradeoff exists in surgical simulation between the costs of calculations, how realistic the tissue-deformation is reproduced and how detailed the morphology being simulated appears. Spring-mass deformable models are used in a lot of surgical simulators as it can achieve real-time visualisation for geometry of moderate size. [MOSEG 05] presents a surgical simulator based on spring-mass system and accelerated by an implementation on the graphics processing unit (GPU) so as to achieve a considerable speedup. The GPU spring-mass system was implemented using OpenGL, C++ and compiled with visual studio C++. A CPU spring-mass system was also implemented in C++ and compiled in Visual Studio C++. As per the results obtained, there was a significant speedup with the GPU implementation as compared to the CPU implementation.

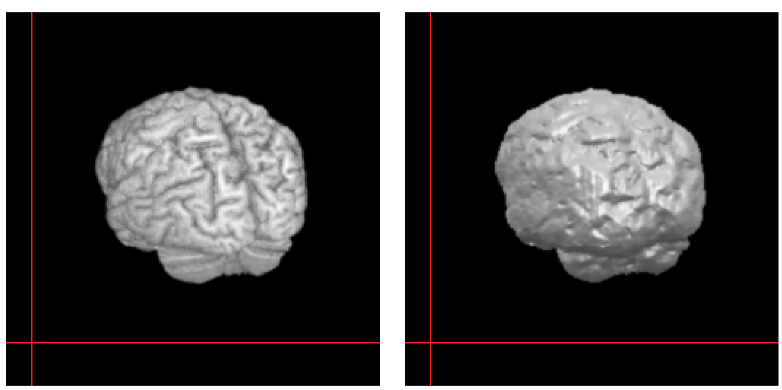
2.4 Volume Visualisation

One of the most popular volume visualisation technique used in clinical practice is known as Multiplanar Reformation (MPR). The MPR approach consists of simply visualising a series of slices either in a parallel direction to one of the faces of the volume or in an oblique direction. With experience, radiologists are able to build a 3D mental model of the anatomy by moving through the 2D slices. [VIDAL 06] reported that a 3D view may give valuable insights to the radiologists and thus much research is being done to develop fast and effective means for presenting 3D medical visualisations. Volume visualisation techniques should provide for an understandable data representation, quick data manipulation and a reasonably fast rendering so as to be useful.

Many steps in volume visualisation process are common to volume visualisation algorithms. First, data acquisition takes place. In the medical setting, the data is acquired using CT scanners or MRI scanners among others. After data acquisition, pre-processing is performed on the data acquired. Some examples of pre-processing operations performed on the image slices are noise removal and enhancing contrast within the slice. Then, the dataset is reconstructed such that the ratio of the dimensions is proportional to the ratio of the dimensions of the measured substance. A data classification is performed next. A mapping operation follows the classification where elements are mapped into a geometric or display primitive. This is the process that differs most in different volume visualisation algorithms. The primitives can be manipulated, shaded, transformed to screen space and displayed.

[ELVIN 92] categorises volume visualisation techniques as Direct Volume Rendering (DVR) Algorithms and Surface-Fitting (SF) algorithms. The main characteristic of DVR algorithms is that elements are mapped directly into the screen space without the use of geometric primitives as an intermediate representation. One disadvantage of DVR is that the entire dataset must be traversed each time an image is traversed. Ray casting, splatting and V-buffer are approaches to DVR. Maximum Intensity Projection (MIP) is a variation to volume rendering technique. Ray-casting technique can be modified so that it uses voxels with maximum intensity rather than considering contributions from all the voxels thus resulting in a fast and effective rendering technique. Figure 1(a) shows the image of a brain obtained using Direct Volume Rendering approach.

SF algorithms, less commonly called feature-extraction or iso-surfacing, fit surface primitives such as polygons to constant value contour surfaces. The required surface is identified by specifying a threshold. SF algorithms usually perform faster than DVR algorithms since in the case of SR the dataset needs to be traversed once only. Of course, changing the threshold value will require the dataset to be traversed again so as to identify the new surface that matches the new threshold. Contour-connecting, marching cubes and dividing cubes are approaches to SF. Figure 1(b) shows the image of a brain obtained using Direct Volume Rendering approach. [ELVIN 92] provides a description of some of the fundamental visualisation algorithms.



(a) (b)

Figure 1: (a) Volume Rendered Image (b) Surface Rendered Image
Images generated using MRIcro software obtained from [SPHSC 06]

3 EXISTING VISUALISATION SYSTEMS

In this section some visualisation systems that have been implemented mainly for the medical field are discussed. The first system discussed is a surgical simulator for hepatic surgery [DELIN 04, DELIN 05]. The system built falls under the category of second generation simulator. The main components of a second generation simulator are shown in figure 2.

To perform the geometric modelling, a combination of image processing techniques is used to extract the principal hepatic structures of interest from the clinical CT images taken before surgery (preoperative images). The structures of interest in this case are: the hepatic parenchyma, the main vessels, the Couinaud segments and the potential lesions. A geometric model based on simplex mesh that is deformed to fit the corresponding 3D edges of the CT image is used to extract the hepatic parenchyma. Mathematical morphology and digital topology techniques are used to extract the vessels. The Couinaud segments are identified by computing the region of influence of the first branches of the portal vein.

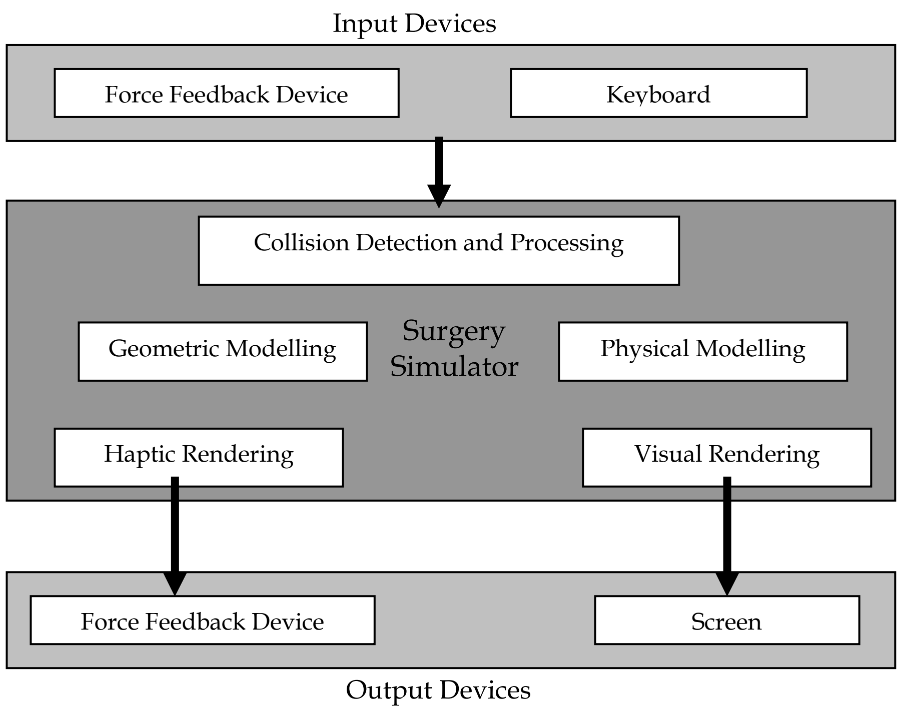


Figure 2: The different technological components of a second-generation simulator

In this work, two frameworks have been used for the physical modelling, namely the theory of continuum mechanics and the finite element method. These approaches were known to take a long computation time to solve a given problem. The researchers have proposed new algorithms for computing soft tissue deformation in a few milliseconds. First of all, they simplified the behaviour of biological soft tissues by assuming linear elastic materials. Next, the domain of computation was decomposed into a set of tetrahedrons. The number of tetrahedrons influenced both the accuracy of the computation and the time required to solve the problem.

In order to efficiently and effectively compute the deformations of linear elastic materials, two complementary algorithms, which were based on the finite element method, were proposed. In the first approach (precomputed models), deformations were computed prior the actual simulation had begun. The deformation that occurred when elementary forces were being applied to each of the nodes was stored. During simulation, a linear combination of these deformations was computed to obtain the actual deformation of the model. The problem with this method was that it assumed the topology of the liver to remain the same throughout the simulation. This implies that no cutting or suturing gestures were possible. A second problem of this method was that since the deformation was directly computed, the viscoelastic nature of the soft tissues was not accounted for.

To overcome the above problems, a second approach, called tensor-mass models, was proposed. This model is similar in a number of ways to the spring-mass. The major problem with the tensor-mass model was that it took more computational time than the first approach. To solve this issue, the first and second approaches were combined to form the hybrid models. In such a situation, a restriction was imposed on the regions of the liver where cutting was allowed. These regions were modelled using the tensor-mass model. The remaining regions were modelled using the precomputed models. Refer to figure 3 that shows some of the input images used and the results obtained [DELIN 04].

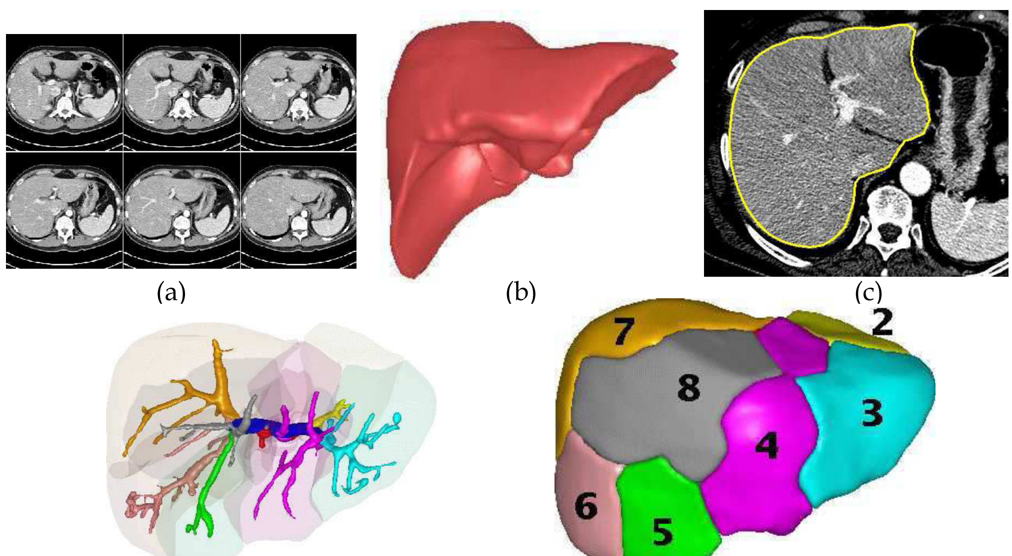


Figure 3: (a) Original CT-scan images of the liver; (b) reconstructed liver model; (c) outline of the liver surface model in a CT-scan image; (d) Segmentation of the portal vein; (e) reconstruction of the eight anatomical segments (Couinaud segmentation)

The system discussed next is used mainly for surgical planning and surgical support [LOREN 93]. According to [LOREN 93], the main steps used in producing a 3D medical image are:

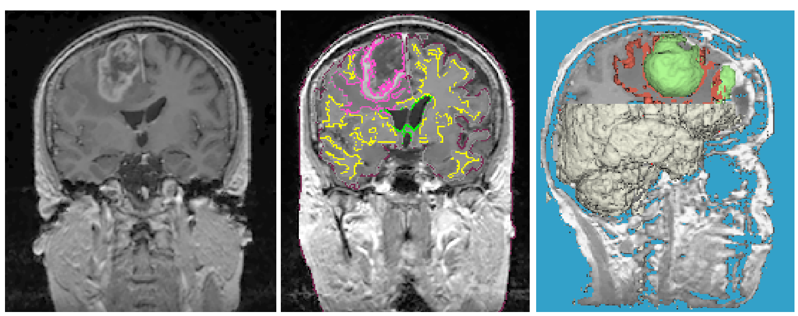
(i) **Data Acquisition**: For the work carried out in [LOREN 93], MRI images have been used.

(ii) **Image Processing**: The image was filtered to increase the signal to noise ratio using an edge preserving diffusion algorithm.

(iii) **Tissue Segmentation**: Segmentation classifies tissues within the 3D volume. Several approaches are possible and often combinations of these techniques are required to successfully segment the tissues in a volume. In [LOREN 93], the objective of the segmentation was to identify the brain surface, cerebral spinal fluid, edema (fluid), tumor and the skin.

(iv) **Model Construction**: Following segmentation, surfaces of each tissue are constructed. Two surface generation algorithms are mentioned in [LOREN 93], namely: marching cubes and dividing cubes. Four separate objects were created: the tumor, the edema surrounding the tumor (green), the surface of the brain and the surface of the face.

(v) **Model Display**: Viewing the 3D models requires a suite of display and manipulation tools including flexible virtual camera manipulation and independent object coordinate systems and attributes. See figure 4 showing inputs and\or outputs of some of the above stages. Once 3D models are created, the rendered images are combined with live video of the patient using a process called video registration.



(a) (b) (c)

Figure 4: (a) MRI Acquisition; (b) MRI Segmentation; (c) Model Generation

Like the previous system, the third system [NAKAJ 06] discussed here concerns surgical planning. The 3D model generated using the system was used to assist in the following: choosing the best intervention method, evaluate surgical risk, select a surgical approach and localise lesions. The input to the system was MRI data. Image pre-processing was applied on the slices for noise removal. An anisotropic diffusion filtering was used for this purpose. A segmentation process based on signal intensities and voxel connectivity was performed. The segmentation result was corrected manually if required. Marching cube algorithm was used for 3D reconstruction. The average time taken by the system from data acquisition to model generation was 6 hours. Though the segmentation approach used is fast, it imposes a limitation on the system. The generated 3D model loses detailed structure. The authors proposed the use of Expectation-Maximisation algorithm for performing the segmentation. A similar approach as in [NAKAJ 06] was used in [CHABR 06] to generate a 3D model of the human brain to be used for surgical planning and intraoperative navigation. In this case, data obtained from electrodes placed on the surface of the brain are combined with the 3D model. The 3D model obtained is shown in figure 5. In the remaining paragraphs of this section SOFA, SPRING, ITK (Insight Toolkit) and VTK (Visualisation Toolkit) are discussed.

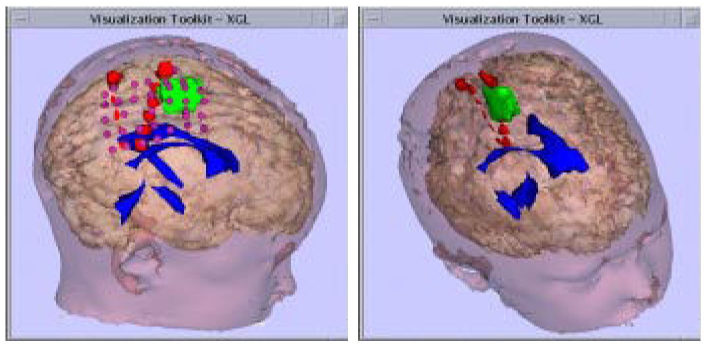


Figure 5: 3D reconstruction showing the position of the electrodes.

Medical simulation requires the integration within a single environment of leading-edge solutions in areas as diverse as visualisations, biomedical modeling, haptics or contact modeling. It becomes very challenging for reasearchers to make progress in specific areas. SOFA tries to solve this problem through its very modular and flexible open source software framework. The main objectives of SOFA framework are:

(1) Provide a common software framework for the medical simulation community.
(2) Enable component sharing/exchange and reduce development time.
(3) Promote collaboration among research groups.
(4) Enable validation and comparison of new algorithms
(5) Help standardise the description of anatomical and biomechanical datasets.

SOFA makes use of a multi-model representation whereby most simulation components can have several representations which are connected together through a mechanism called mapping. Each representation can be optimized for a particular task such as collision detection and visualisation. SOFA is highly customizable. Among its future work are support for multiprocessing, topological changes and haptic feedback.

SPRING is a real-time soft-tissue simulation platform maintained by Stanford University's SUMMIT group and the National Biocomputation Center at Standford University. Its purpose is for building and running surgical simulators to be used in medical education of surgeons. SPRING consists of a number of interacting processes which may be located on the same or different computers. SPRING has a main process which performs rendering, collision detection, accepts inputs from keyboard/mouse and haptic devices.

The Visualization ToolKit (VTK) is an open source, freely available software system for 3D computer graphics, image processing, and visualization. VTK includes a textbook published by Kitware (The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics, 3rd edition ISBN 1-930934-12-2), a C++ class library, and several interpreted interface layers including Tcl/Tk, Java, and Python. VTK has been implemented on nearly every Unix-based platform, PC's (Windows 95/98/NT/2000/XP) and Mac OSX Jaguar and later. The design and implementation of the library has been strongly influenced by object-oriented principles.

The graphics model in VTK is at a higher level of abstraction than rendering libraries like OpenGL or PEX. Hence it is much easier to create useful graphics and visualization applications. In VTK applications can be written directly in C++, Tcl, Java, or Python. Using the interpreted languages Tcl or Python with Tk, and even Java with its GUI class libraries, it is possible to build useful applications really fast.

Finally, the software is a true visualization system. VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques like implicit modelling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. Moreover, dozens of imaging algorithms have been directly integrated into the system. Thus you can mix 2D imaging / 3D graphics algorithms and data.

The Insight Toolkit (ITK) was developed by six principal organizations, three commercial (Kitware http://www.kitware.com, GE Corporate R&D http://www.crd.ge.com, and Insightful http://www.insightful.com) and three academic (UNC Chapel Hill http://www.unc.edu, University of Utah http://www.utah.edu, and University of Pennsylvania http://www.upenn.edu). Additional team members include Harvard Brigham & Women's Hospital, University of Pittsburgh, and Columbia University.

ITK is an open-source software toolkit for performing registration and segmentation. Segmentation is the process of identifying and classifying data found in a digitally sampled representation. Typically the sampled representation is an image acquired from such medical instrumentation as CT or MRI scanners. Registration is the task of aligning or developing correspondences between data. For example, in the medical environment, a CT scan may be aligned with a MRI scan in order to combine the information contained in both.

ITK is implemented in C++ and is cross-platform. An automated wrapping process generates interfaces between C++ and interpreted programming languages such as Tcl, Java, and Python. This enables developers to create software using a variety of programming languages. ITK's C++ implementation style is referred to as generic programming (i.e., using templated code). Such C++ templating means that the code is highly efficient, and that many software problems are discovered at compile-time, rather than at run-time during program execution.

Because ITK is an open-source project, developers from around the world can use, debug, maintain, and extend the software. ITK uses a model of software development referred to as extreme programming. Extreme programming collapses the usual software creation methodology into a simultaneous and iterative process of design-implement-test-release. ITK is currently under active development. ITK employs leading-edge segmentation and registration algorithms in two, three, and more dimensions. The goals for the project include the following:

- Support the Visible Human Project.
- Establish a foundation for future research.
- Create a repository of fundamental algorithms.
- Develop a platform for advanced product development.
- Support commercial application of the technology.
- Create conventions for future work.
- Grow a self-sustaining community of software users and developers.

4. CONCLUSION

From the above, it can be said that the main components of a surgical simulator are collision detection, soft-tissue modeling, visual rendering and haptic rendering. As such no best algorithms for each of the components exist till now that work for all situations. The algorithm to be used depends on the application for which the simulator is to be used and the degree of realism, interactivity, and accuracy required. SOFA tries to solve this problem through its multi-model representation. As mentioned previously there is a tradeoff between speed, accuracy and degree of realism. Use of GPU can help in speeding up the modeling of complex geometries as shown by [MOSEG 05]. Much research has already been undertaken in the field of Surgical Simulation and there is still a lot to be done.

5. REFERENCES

[**ALLARD 07**] J. Allard, S. Cotin, F. Faure, P.-J.Bensoussan, F. Poyer, C. Duriez, H.Delingette and L. Grisoni, "*SOFA – an Open Source Framework for Medical Simulation*", Medecine Meets Virtual Reality (MMVR), Volume 125, pp. 13-18, 2007

[CHABR 06] Alexandra Chabrerie, Fatma Ozlen, Shin Nakajima, Michael Leventon, Hideki Atsumi, Eric Grimson, Erwin Keeve, Sandra Helmers, James Riviello Jr., Gregory Holmes, Frank Du_y, Ferenc Jolesz, Ron Kikinis, and Peter Black, "*Three-Dimensional Reconstruction and Surgical Navigation In Pediatric Epilepsy Surgery*", Published in Proceedings MICCAI, LNCS volume 496, pp. 74-83, 1998

[DELIN 98] Herve Delinguette, "*Towards Realistic Soft Tissue Modeling in Medical Simulation*", Published in Proceedings of the IEEE: Special Issue on Surgery Simulation, pp. 512-523, April 1998

[DELIN 04] Herve Delinguette and Nicholas Ayache, "*Soft Tissue Modeling for Surgery Simulation*" In N. Ayache, Ed. Computational Models for the human Body: Handbook of Numerical Analysis, Elsevier, 2004

[DELIN 05] Herve Delinguette and Nicholas Ayache, "Creating a simulator for training physicians to perform minimally invasive surgical procedures", Communications of the ACM February 2005, Vol. 48. No. 2, pp. 31-35

[ELVIN 92] T. T. Elvins, "*A survey of algorithms for volume visualization*", Computer Graphics, 26(3), pp. 194-201, 1992

[ITK 06] Insight Toolkit: http://www.itk.org/

[LOREN 93] William Lorensen, H.Nafis, C.Kikinis, R., Altobelli, D. and Glesson L., "*Enhancing Reality in the operating room*", In proceedings of IEEE Visualization, pp. 410-415, 1993

[**MOSEG 05**] Jesper Mosegaard, Peder Herborg, Thomas Sangild Sorensen, "*A GPU Accelerated Spring Mass System for Surgical Simulation*", Studies in health technology and informatics, volume 111, pp. 342-348, 2005

[NAKAJ 06] Shin Nakajima, Hideki Atsumi, Abhir H. Bhalerao, Ferenc A. Jolesz, Ron Kikinis, Toshiki Yoshimine, Thomas M. Moriarty, Philip E. Stieg, "*Computer-assisted Surgical Planning for Cerebrovascular Neurosurgery*", Volume 41(2), pp. 403-9, 1997

[**RUSPINI 97]** Diego C. Ruspini, Krasimir Kolarov and Oussama Khatib "*The Haptic Display of Complex Graphical Environments*", Computer Graphics Proceedings, Annual Conference Series, pp. 345-352, 1997

[SAUPIN 08] Guillaume Saupin, Christian Duriez, Stephane Cotin,"*Contact model for haptic medical simulations*", Proceedings of the 4th international symposium on Biomedical Simulation, pp. 157-165, 2008

[SCHMI 05] Mark Schmidt, "*Automatic Brain Tumour Segmentation*", Thesis for Master of Science degree submitted at University of Alberta, 2005

[SPHSC 06] The free software MRIcro has been downloaded from http://www.sph.sc.edu/comd/rorden/mricro.html

[**SPRING 09**] SPRING: http://spring.stanford.edu/

[**VIDAL 06]** F.P. Vidal, F. Bello, K.W. Brodlie, N.W. John , D. Gould, R. Phillips and N.J. Avis, "Principles and Applications of Computer Graphics in Medicine", COMPUTER GRAPHICS forum, Volume 25, number 1 pp. 113–137, 2006

[VTK 06] Visualisation Toolkit: http://www.vtk.org/