HAVE 2005 - IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa, Ontario, Canada, 1–2 October, 2005

# Software Structure Design for A Haptic-Based Medical Examination System

Amir M. Tahmasebi[1], Purang Abolmaesumi[1], David Thompson[2] and Keyvan Hashtrudi-Zaad[3]
1. School of Computing, Queen's University, Kingston, ON, Canada
2. Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec
3. Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada
tahmaseb,purang@cs.queensu.ca

***Abstract –*** *This paper presents a novel approach to software structure design of a haptic-based medical examination simulator. The proposed software is implemented in a fully object-oriented structure that provides real-time interaction with a PHANToM$^{TM}$ haptic device, as a diagnostic tool for medical experts. The software allows radiology experts to visualize, display and interact with the graphical model of a patient's pre-scanned organ or tissue. The expert is also able to re-slice through pre-registered model data sets and view the 2D reslice images in different modalities simultaneously. The haptic interface guarantees position correspondence between the operator's hand and a virtual probe. Thus the simulated procedure emulates actual examinations condition in clinic. Our preliminary human factors study at Kingston General Hospital with radiology residents have demonstrated the significant potential of the developed software for scientific and commercial applications.*

***Keywords –*** *Medical Examination Simulator, Software Structure, Haptics, PHANToM, Graphical User Interface.*

## I. INTRODUCTION

Medical haptics is a new field of research that aims to augment or extend the capabilities of medical experts. As one of the applications of medical haptics, medical simulators have shown significant improvement in conventional medical examination procedures. Cavusoglu *et al.* [10] designed a haptic command station that provides a haptic virtual environment for the medical expert. The FeTouch system [13] allows pregnant mothers to interact with a model of their fetus. The system renders 3D visual-haptic models of fetus face from sets of 2D ultrasound slices. S. Gibson *et al.* [12] developed a real-time haptic-based system for simulating arthroscopic knee surgery that is based on volumetric object models derived from 3D MRI. Many other applications of haptic feedback in simulating neuroradiology procedures [15], interactive needle insertion into soft tissue [11] and endoscopic training [8] have also been proposed. Several medical simulators, for surgical [14, 16] or diagnostic [4, 7, 9, 18] purposes, have been designed that incorporate single-image data.

We propose a novel software structure that integrates the haptic technology with realistic simultaneous visualization of multi-modal image data for diagnosis purposes. As shown in Figure 1, using a PHANToM[1] haptic device, the operator is able to interact with the software via a dummy probe mounted on the device arm so as to move a virtual probe against a virtual patient's anatomical model. The proposed software is implemented in a fully object-oriented structure using multi-event handling of Qt [2] and visualization capabilities of an open source Visualization ToolKit (VTK) [6]. The developed software allows medical experts to: (1) load off-line captured 3D medical image data of a patient which are acquired by a technician and then, (2) explore the reconstructed volume data using a force feedback haptic device, without the patient being present at the time of re-examination. It also provides the opportunity for the expert to compare data from the same patient captured at different times to find abnormalities and to view the variation of these abnormalities during a period of time. The proposed setup has the potential to be used for off-line examination of elderly patients (for tele-homecare purposes) or patients in remote areas.

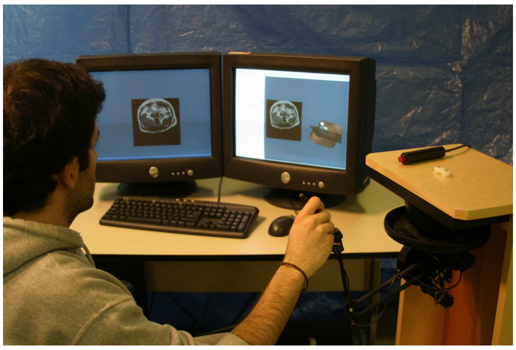


Fig. 1. The proposed setup; real-time MRI/US examination of abdomen.

The rest of the paper is organized as follows: Section II overviews the system architecture, whereas Section III details the software structure. Section IV describes the preliminary human factors study results as the validation of the proposed system. Conclusion and possible future improvements to the system are discussed in Section V.

[1] SensAble Technologies PHANToM Premium 1.5A

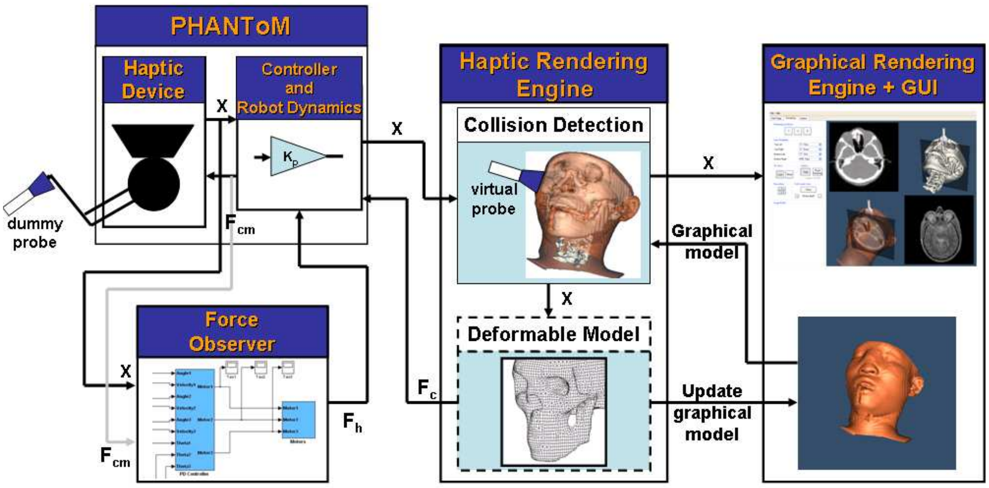


Fig. 2. Proposed system: overall system structure. $F_c$: Contact force, $F_{cm}$: Force command, $F_h$: Hand force, X: End-effector position.

## II. SYSTEM ARCHITECTURE OVERVIEW

Figure 2 displays an overall view of the developed simulator hardware and software structure, which consists of the following four major components: PHANToM, Force observer, Haptic rendering engine and GUI which incorporates the Graphical rendering engine. The user is physically interfered with PHANToM via a dummy probe mounted on the robot. Using GHOST SDK of SensAble Technologies Inc., a one-to-one mapping is obtained between the operator's hand motion and the location of the virtual probe on the virtual model of the patient. Thus, the simulated procedure tries to emulate actual examination conditions in clinic. PHANToM consists of the haptic device and its controller. The user hand (dummy probe) position $(X)$ is calculated from PHANToM motor angular position captured by encoders. The position is passed to the "Haptic-Rendering Engine", which is provided with graphical models of the patient anatomy and the virtual probe (in VRML[2] format). The engine then calculates the force of contact $(F_c)$ between the human anatomical model and the virtual probe. The contact force is conveyed to the user via backdrivable PHANToM. The haptic engine also provides the graphical engine with the current position of the dummy probe to update the position of the virtual probe on the screen.

Haptic force feedback is implemented within the software structure using the GHOST SDK framework. Virtual surface model of the examined anatomy is constructed using 2D images. Extracting surface point data of a stack of images, a surface model of the body is constructed in the form of a polygon mesh. The constructed mesh is saved in the form of a VRML file. Using GHOST SDK libraries, the position and surface normal information are translated to torque command vectors and passed to PHANToM motors to provide force feedback to operator's hand at the time of collision between the virtual probe and the virtual anatomy model.

A GUI is provided in the "Graphical-Rendering Engine" to visualize, display and interact with the graphical model of the examined tissue or organ, as well as to display the corresponding 2D slices of different modalities of the pre-registered data sets.

Current implementation provides constant force feedback to the operator's hand. However, deformable volume sets would more realistically simulate interactions with the patient. We have divided this task into two components: (a) new implementation of volume sets considering deformation using techniques such as the Finite-Element Method (FEM) which is considered as a future work, and (b) contact force estimation needed to calculate the deformation of the anatomy governed by the model and to control PHANToM, which has been previously reported in [17] and is shown as "Force Observer" block in Figure 2. The "Force Observer" block receives PHANToM control command $(F_{cm})$ and position of the dummy probe to estimate the operator's hand force $(F_h)$.

## III. SOFTWARE STRUCTURE DESIGN

The proposed software consists of four major components: (1) Image database, (2) Graphic rendering, (3) Haptic rendering and (4) GUI which are detailed next.

### A. Image Database

As input to the software, any sequence of 2D images with any modality and standard image format can be loaded into the software. It is also possible to simultaneously load several pre-registered data sets and anatomical model data.

Current implementation includes three types of image input that are differentiated according to the image modality and data set type:

1. In-house image acquisition: All ultrasound images are captured in-house using a high-end 3D/4D ultrasound machine, GE Voluson 730 Expert.

[2] Virtual Reality Modelling Language

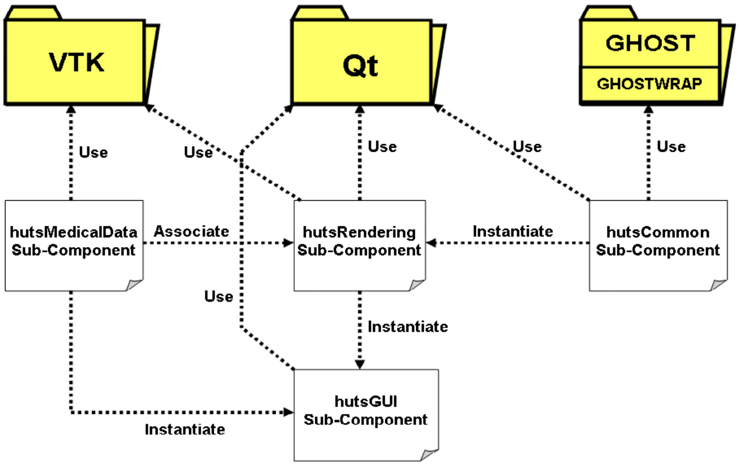


Fig. 3. A high-level overview of graphic rendering software structure and its sub-components and their relationships in the United Modelling Language (UML).

2. Hospital imaging: Images captured at Kingston General Hospital, such as MRI images captured from volunteer subjects.
3. Visible Human Project image data: CT and MRI data of head available from Visible Human Project website [5].

Depending on data set type, image acquisition stage has different routines. Among all three data acquisition methods, ultrasound imaging requires position data corresponding to each captured image since ultrasound images are captured based on freehand movement of the ultrasound probe. Other types of image data sets such as MRI and CT are provided as a stack of sequential images with constant and known spacing between the two consecutive images.

## *B. Graphic Rendering*

The graphic rendering component is a software package that is developed in VC++ .Net environment in a fully object-oriented programming structure. Graphic rendering engine consists of four sub-components: hutsGUI, hutsCommon, hutsMedicalData, and hutsRendering. These stand-alone sub-components can be compiled and built separately, and therefore could be reused in a new project quite easily.

Several open-source frameworks have been used in the development of the proposed software: (a) VTK: an open source, freely available software system for 3D computer graphics, image processing, and visualization, (b) Qt: a multi-platform C++ GUI toolkit created and maintained by Trolltech, (c) VTKQt: a cross-platform library to use multi-thread event handling features and advance GUI design features of Qt in conjunction with specific rendering and interaction features of VTK[3], (d) GHOST SDK: an object-oriented General Haptics Open Software Toolkit provided by SensAble Technologies, for interaction with the PHANToM device and (e) GHOSTWrap: a wrapper for the SensAble GHOST enabling programmers to use the vendor-provided Standard Template Library [1].

A high-level overview of the graphic rendering component including its sub-components and their relationships in the United Modelling Language (UML) is shown in Figure 3. There are three types of relationships defined between the proposed software sub-components. These terms are provided in UML language. All sub-components *use* VTK, Qt, GHOST and GHOSTWrap platforms. Some sub-components are *associated* with some other meaning that one sub-component is derived from the other (*e.g.* hutsMedicalData is *associated* with hutsRendering). A component may contain an *instance* of the other and therefore, it is possible to use any of the functions defined in that sub-component within the other sub-component. Each of the software sub-components is described below considering its functionality:

### B.1 hutsGUI

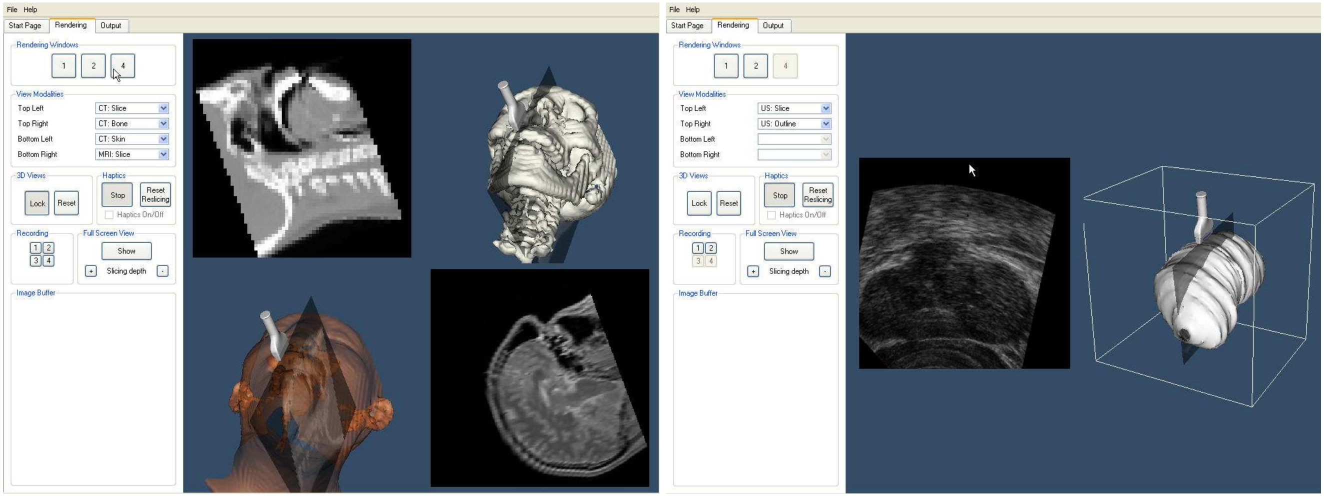
GUI's major functionality is included in this sub-component. A configuration file is included for each data set that is filled with primary information of the data set (such as file name, image size, data spacing, number of images and region of interest) belonging to that particular data set. Reading a stack of image files (in one of the forms RAW, BMP, PNG, and JPEG), or VTK file formats such as *vtkPolyData* for surface data or *vtkImageData* for volume data construction, entering/updating patient's medical information and image files are main functionalities of this sub-component.

### B.2 hutsCommon

The followings are the main procedures implemented in this sub-component:

- Read the selected data set path from XML-based data browsing page.

[3] The version we use for our project is the version distributed by M. Koenig [3]. We modified this version to allow render window interactors to be synchronized.



Top Left: CT: Slice
Top Right: CT: Bone
Bottom Left: CT: Skin
Bottom Right: MRI: Slice

(a)

Top Left: US: Slice
Top Right: US: Outline
Bottom Left:
Bottom Right:

(b)

Fig. 4. (a) Real-time MRI/CT re-examination of Visible Human data set and (b) Real-time ultrasound re-examination of prostate.

- Perform initial communication between the software and the PHANToM.
- Start/stop PHANToM servo loop.
- Load/add/delete objects in the haptic graph scene.
- Perform communication between the PHANToM and the software within force feedback loop.
- Receive dummy probe's position and orientation from the robot and pass it to the GUI rendering window loop.
- Allocate memory and handle memory leakage.

B.3 hutsMedicalData

This sub-component is implemented to load image data and construct required objects for rendering and re-slicing purposes. Three classes are considered for this sub-component. Each imaging modality (ultrasound, CT, and MRI) has a separate class. Some functionalities are general and are defined almost with the same pattern for all three modalities, such as loading stack of images (according to the specification provided in the config file).

B.4 hutsRendering

This sub-component generates the GUI for rendering and re-slicing purposes. Basically, visual interaction, control buttons for haptic loop, and displaying rendering windows build the main structure of this sub-component. Some major functions are camera initialization, start/stop the haptic interface and building rendering windows.

### C. *Graphical User Interface*

The GUI is developed in VC++ .Net environment in a fully object-oriented structure. Graphics and object rendering are implemented using VTK. The GUI allows the operator to move the 3D view point of the rendered volume data, as well as zooming in and out within each rendering window, independently or simultaneously. Capturing snapshots and saving them on disk is also another capability of the GUI. Figures 4(a) and 4(b) show snapshots of the GUI while loading head and prostate data sets, respectively. Using multi-thread handling capabilities of Qt, the GUI can render up to four volume data sets simultaneously.

The operator is able to add new data sets or open previously saved data sets. After loading stacks of 2D images, 3D volume sets, 3D anatomy models, haptic and VRML files for haptic interface are generated subsequently. Depending on how many different image modalities operator selects, there will be one, two or four rendering windows. By starting the interaction, a virtual probe will be displayed in each rendering window that contains a 3D volume and attached to the probe, a 2D plane is displayed where the re-slice image is texture mapped on top of it as shown in Figures 4(a) and 4(b). In cases where there are four viewing windows displayed on the screen, the size of each window might not be large enough for accurate observation of the details. To solve this problem, a second monitor is considered to have a full screen view of any of the available options as shown in Figure 1.

### D. *Haptic Rendering*

Two formats of data are transmitted between the software and the PHANToM robot. Position and orientation of the dummy probe attached to stylus, which are provided by the robot motor encoders, are transmitted to the GUI to update the position and orientation of the virtual probe. The second interaction between the robot and GUI is the transmission of the haptic model which is processed and provided by VTK and GHOST SDK classes. The GHOST SDK provides a high-level interface that allows developers to concentrate on the generation of haptic scenes and the manipulation of the scene properties. It also allows for creation of objects with different properties within these scenes without having to be concerned with the low-level interface device issues. A typical application using GHOST SDK starts by creating a *scene graph*. A scene graph is a hierarchical tree of nodes, representing scene objects. Figure 5 shows the resulting scene graph.

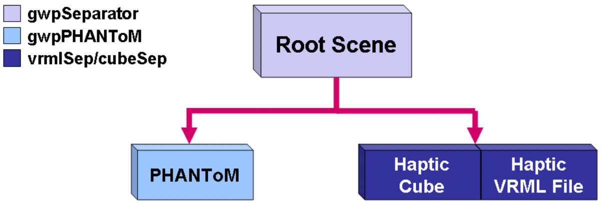


Fig. 5. GHOST SDK scene graph for haptic simulation.

To generate the scene graph, the following steps are considered respectively:

1. Create a haptic environment through the specification of our haptic scene graph.
2. Start the haptic simulation process (the servo control loop).
3. Perform application-specific functions that include the generation and use of computer graphics.
4. Communicate with the haptic simulation process as needed.
5. Perform clean-up operations when the application is stopped.

As it is shown in the above figure, there are two types of objects in our haptic scene graph which are distinguished according to their geometrical features.

*Complex-structure* scene objects that do not have any known regular shape, such as face skin or abdomen skin are created in the form of polygon mesh data using VTK data format, called "Polydata" models, that are converted and loaded as VRML files. As defined in VTK, a vtkPolyData represents a geometric structure consisting of vertices, lines, polygons, and triangle strips. Several filters are available in VTK that can be used to extract surface normals from image data. Cascading these filters one after the other as shown in Figure 6, a surface polygon mesh is generated in one of VTK known formats (*e.g.* vtkPolyData). There is another type of filter provided by VTK (vtkVRMLExporter) that exports the surface point data from VTK format to VRML format. The resulting VRML file is translated into a haptic model and passed to PHANToM robot motors in the form of force commands using GHOST SDK and GHOSTWrap interfaces.

There are also *Simple-shape* haptic objects such as cubes. Since ultrasound images do not provide a clear image to extract surface points from, skin surface model generation would not provide a smooth haptic surface model data. To provide haptic force feedback for such data sets, a simple cube-shape model is considered as the haptic interface. Figure 7 shows the haptic cube model generated for abdomen ultrasound data set.

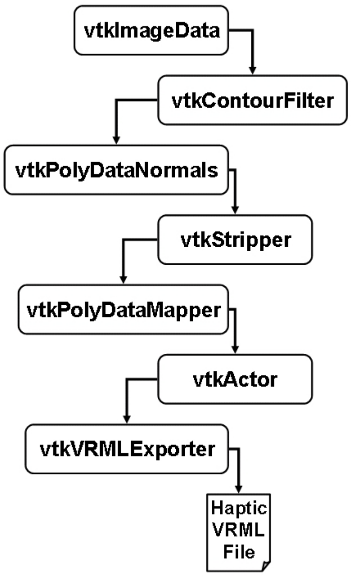


Fig. 6. Generating haptic VRML file from vtk "polydata".

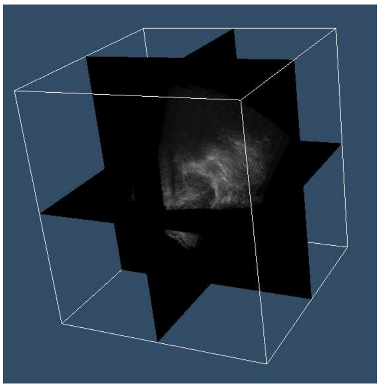


Fig. 7. Haptic cube model generated from abdomen ultrasound data set.

## IV. VALIDATION

A questionnaire was designed to evaluate the features of the proposed simulator software and system performance. This questionnaire was filled by four radiology residents from Kingston General Hospital. Three data sets were provided for the study: (1) the pre-registered MRI and CT data set of head, (2) pre-registered MRI and ultrasound data set of abdomen, and (3) ultrasound data set of prostate.

Regarding the software features, among three types of re-slice images (MRI, CT, and ultrasound) provided by re-slicing through reconstructed data sets, ultrasound re-slice images did not have acceptable image quality and MRI re-slice images were considered as high quality images according to residents' opinions. Other software features such as multi-modality view, and dual monitor display were found useful. The GUI was found user friendly and easy to interact with. The residents also believed that the proposed simulator could potentially be used as an examination tool at Kingston General Hospital.

According to subjects' responses, the force feedback was a useful feature but not adequate to provide the same feeling as scanning a real patient. As suggested, a mannequin would provide a more realistic feeling of scanning a real patient than the force feedback alone. Also, adding deformation to anatomy modelling in the GUI may improve the performance of haptic force feedback according to residents' opinion.

Overall, preliminary human factors study among radiology residents at Kingston General Hospital has demonstrated satisfactory performance in terms of haptic force feedback, system ergonomics, image clarity, and the usefulness of GUI features.

## V. DISCUSSION AND FUTURE WORK

This work presented a novel approach to software structure design of a haptic-based simulator for medical examination purposes. The proposed software is implemented in a fully object-oriented structure that provides real-time interaction for the operator using a PHANToM haptic device, as a diagnostic tool for medical experts. The PHANToM provides interaction in real-time with volumetric images of a virtual patient. The haptic interface guarantees position correspondence between the operator's hand and a virtual probe position that slices medical volume sets in the plane of the probe. A GUI is designed to visualize, display and interact with the graphical model of the examined tissue, as well as displaying the corresponding 2D multi-modality re-slices of pre-registered data sets. Although still at the preliminary stages of development, our pilot user study at Kingston General Hospital with radiology residents have demonstrated the significant potential of the developed system for scientific and commercial applications.

While static data sets seem to be sufficient for basic examination sessions, deformable volume sets would more realistically simulate an interaction with a patient. Currently, finite-element based deformation of volume sets under operator's hand force is being developed. The hand force is estimated through the identification of the PHANToM dynamic parameters [17]. The estimated hand force will be used to calibrate the deformable model and to control the haptic device.

## VI. ACKNOWLEDGEMENT

The authors would like to thank P. Foroughi and T. Chen for their contribution in this project. The authors would also like to thank the Human Mobility Research Centre (HMRC) at Queens University for providing imaging facilities as well as Natural Sciences and Engineering Research Council of Canada (NSERC), Institute of Robotics and Intelligent Systems (IRIS) and Canadian Space Agency (CSA) for funding this project.

## References

[1] "GhostWrap Library." [Online]. Available: http://caddlab.rad.unc.edu/software/index.html

[2] "Qt platform, Trolltech. Inc." [Online]. Available: http://www.trolltech.com/

[3] "VTKQt open-source platform." [Online]. Available: http://www.matthias-koenig.net/vtkqt/

[4] "UltraSim, a product by MEDSIM Inc." [Online]. Available: http://www.medsim.com/

[5] "Visible Human Project." [Online]. Available: http://www.nlm.nih.gov/

[6] "Visualization ToolKit." [Online]. Available: http://www.vtk.org/

[7] W. Arkhurst, A. Pommert, E. Richter, H. Frederking, S. I. Kim, R. Schubert, and K. H. Hohne, "A virtual reality training system for pediatric sonography," *Proc. Computer Assisted Radiology and Surgery (CARS)*, pp. 453–457, 2001.

[8] A. Benali, P. Richard, and P. Bidaud, "A six dof haptic interface for medical virtual reality applications: design, control and human factors," *Proc. IEEE Virtual Reality Conference*, p. 284, 2000.

[9] G. Burdea, G. patounakis, V. Popescu, and R. E. Weiss, "Virtual reality training for the diagnosis of prostate cancer," *Proc. IEEE International Conference on Information Technology Applications in Biomedicine*, pp. 6–13, 1998.

[10] D. d'Aulignac, M. C. Cavusoglu, and C. Laugier, "Modeling the dynamics of the human thigh for realistic echographic simulator with force feedback," *Proc. Intl. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 1191–1198, 1999.

[11] S. P. DiMaio and S. E. Salcudean, "Needle insertion modelling and simulation," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2098–2105, 2002.

[12] S. Gibson, J. Samosky, A. Mor, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, S. Nakajima, H. Ohkami, R. Osborne, and A. Sawada, "Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback," *Proc. Medical Image Analysis*, vol. 2, pp. 121–132, 1998.

[13] B. la Torre, D. Prattichizzo, F. Barbagli, and A. Vicio, "The fe-touch project," *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 1259–1263, 2003.

[14] A. Liu, F. Tendick, K. Cleary, and C. Kaufmann, "A survey of surgical simulation: applications,technology, and education," *Presence*, vol. 12, pp. 599 – 614, 2003.

[15] W. L. Nowinski and C. Chee-Kong, "Simulation of interventional neuroradiology procedures," *Proc. Intl. Workshop on Medical Imaging and Augmented Reality*, p. 8794, 2001.

[16] J. H. Shuhaiber, "Augmented reality in surgery," *Arch Surg*, vol. 139, no. 2, pp. 170–174, 2004.

[17] A. M. Tahmasebi, B. Taati, F. Mobasser, and K. Hashtrudi-Zaad, "Dynamic parameter identification and analysis of a phantom haptic device," *Proc. IEEE Intl. Conf. Control Applications*, pp. 1251 – 1256, 2005.

[18] UltraSim, "MedSim Advanced Medical Simulation Ltd." [Online]. Available: http://www.medsim.com/