Platform for tele-guidance applied in suture simulation

Feras Al Chama, Denis Chêne, Craig Cornelius

Abstract—**The combination of surgical simulation and haptic interfaces provides a powerful platform for reproducing the visual and motor sensations that a surgeon experiences during a surgical operation. With the wide availability of information technologies and new, robust telecommunication technologies, remote tele-guidance for learning surgical procedures has become feasible for more effectively employing expertise through remote instruction. In this paper we describe a system for tele-guidance in suture simulation, comprising haptic interfaces, communication infrastructure, the SPRING surgical simulator platform, and an instructional protocol. Furthermore, the effectiveness of the proposed platform will be validated by experimental results of surgeons professors, and students using the tele-guidance system.**

I. Introduction

THE advent of new computer technologies is considered a revolution in surgical education as well as in planning and executing surgical movements and gestures. Many manual activities may benefit from the use of haptic devices to assist and to replay fine movement. This provides great utility in such domains as surgical education and expert analysis, since the planning and execution of a learner's movements can be analysed, recorded, and replayed.

Traditionally, surgery lessons have been guided by the professor throughout beginning training classes, which consist of learning movements, developing a dissection plan, and making the choice for type of therapy.

The rate at which basic surgical movements and skills improve is often slow, due partly to the continuing development of novel materials, technologies, and surgical procedures, including methods for suturing and minimally-invasive surgery, which puts surgeons in the permanent condition of learning new skills and methodologies.

For this reason, the concept of surgical simulation is very useful especially for reproducing, with the help of an IT system, the visual and motor sensations that a surgeon will experience during the surgical operation.

These simulations allow the surgeon to practice surgical techniques with the same complexity of video-surgery (endoscope, laparoscopy...) but with more flexibility and lower cost compared with the actual surgical theatre experience that involves actual tools, operating rooms, and animal or human subjects.

Surgical simulation opens the possibility of accurate planning of delicate surgical operations and also the acquisition of skills in a new surgical procedure [1], both of which reduce the learning time of advanced surgical techniques. Additional important features of surgical simulation are reproducibility and analyzability that provide consistency of experience, objective evaluation of skills and technique, and metrics for individuals to track and improve their own performance over time, which are all crucial for effectively learning any new skill.

There is great benefit in disseminating new surgical techniques to students widely and quickly. However, in many specialties, expert surgeons are few in number and are often unavailable for teaching. Thus, e-learning and distant expert analyses may become important in teaching surgical skills..

We will first describe a haptic learning system that is used locally, and then we will describe system requirements and architecture for tele-guidance, provide an overview of the SPRING simulation platform, identify communication constraints in such a system, and describe user experiments in which we are engaged.

In general, a computerized platform for, single user local surgical simulation (i.e., the simulator software and the "the haptic interface" are installed on the same computer system) should include:

- haptic interfaces,
- 3-D virtual models of soft tissue (human organs) for the region of surgical interest,
- surgical tools models.

These are described below:

A. *The haptic interface*

The meaning of "haptic interface" is that the human touch sense is captured or /and simulated by a manipulation device [3]. In a surgical simulation, a haptic device controls simulated surgical tools (such as a grasper, probing tool, scalpel, needle, forceps, laparoscope, suction device, etc.) For any of these tools, effective simulation requires that the surgeon feel a realistic force applied on the contact surfaces between virtual tools and virtual tissues (e.g., force between the head of the needle and skin in the suturing operation).

Manuscript received September 15, 2006. This work was supported in part by Orange-France Telecom Research & Development and by French and Syrian Ministry of Research.

F. Al Chama is with the LISV 45, avenue des Etats-Unis, 78000, Versailles, France (e-mail: falchama.ext@orange-ft.com).

D. Chêne is with France Telecom R&D, 28 chemin du vieux Chêne BP 98, F-38243 Meylan cedex, France (e-mail: denis.chene@orange-ft.com).

C. Cornelius is with SUMMIT, Stanford University School of Medicine, Stanford, CA 94305, USA (e-mail: cwcornelius@stanford.edu).

A number of haptic tools and technologies are available for medical simulation (SensAble® PHANToM[2], immersion, …), but all implement the same general concept, which is that haptic interface provides the computer with the spatial Cartesian and angular coordinates of the interface arm or stick or handpiece (an object manipulated by the user). The computer then returns commands indicating what force should be produced (haptic feedback) from the interaction of the virtual objects being modelled (surgical tools, the human tissues or organs, and any additional forces such as gravity).

A. Soft tissues modelling and Suture simulation

Dynamic behavior of soft biological tissues is a particularly demanding application of deformable modeling [20], requiring detailed definition of several properties of small regions of the tissues represented. The computations required are complex and compute-intensive, and even defining the appropriate tissue properties and physical reactions to interactions with tools is difficult.

Suturing is an interesting and demanding simulation procedure, requiring the modeling of the virtual surgical instruments (needle, forcep, etc.), the behavior of the thread material, the dynamics of the soft tissues being sutured, and details of thread-needle, needle-tissue, and thread-tissue interaction. In addition, suture simulation includes a number of subtasks, including stitching, creating a suture loop, developing a knot, placing a knot, and securing the knot.

Pioneering work in this field was done by Berkley et al. They proposed a model of skin based on the finite element modeling method (FEM) ([4], [5]), which permits simulation of the needle-skin interaction while performing the suture operation. However, this work did not address the interaction between thread and soft tissues, nor the interaction of a thread with itself to make a knot, but instead focused on the construction of soft tissue models and their interactions with a hard object, i.e., the needle.

Recently, Brown et al. have developed a model of real-time simulation of knot tying ([6]), using the SPRING simulation platform [7]. With this model we can simulate a complete suture operation of two virtual blood vessels using virtual instruments ([8], [9]) (i.e., needle and forceps) driven by haptic devices. This work includes both thread-soft tissue interaction and thread self-interaction, which are very important for realistic suture training, but which have not been implemented in other research,

II. System requirements and architecture

The basic aim of a tele-guidance surgical system is to support knowledge transfer between expert surgeons and beginners, who may be located in remote locations anywhere in the world. This type of e-learning is especially important because experts in some surgical specialities are rare, are generally fully occupied with surgical practice, and access to their time and expertise is very expensive. In addition, physically brining students and such experts to the same location is expensive and time consuming.

In our work we have designed a multi-expert / single student tele-guidance surgical/suturing system that provides the possibility of real time student-expert collaboration through a surgical simulation platform. In this scheme there are two types of expert: one local to the student's location, and the other at a distant location. This second expert is usually a highly experienced professor of surgery, who specializes in delicate and difficult surgical movements.

In this system, collaboration is achieved by guiding the hand of the student (if needed) through haptic interfaces handpieces, in which forces are relayed from the expert's interfaces and duplicated on that of the student.

For efficient collaboration on tele-suturing or tele-surgical tasks (tele-guidance), a global real time synchronisation between experts and student is very important. A typical architecture of a virtual realty teleoperated surgical system will consist of several main modules: the platform for surgical and suturing simulation (for example, SPRING), a guidance protocol, a communication infrastructure, and the haptic interfaces. A detailed description of these modules is given in the following sections.

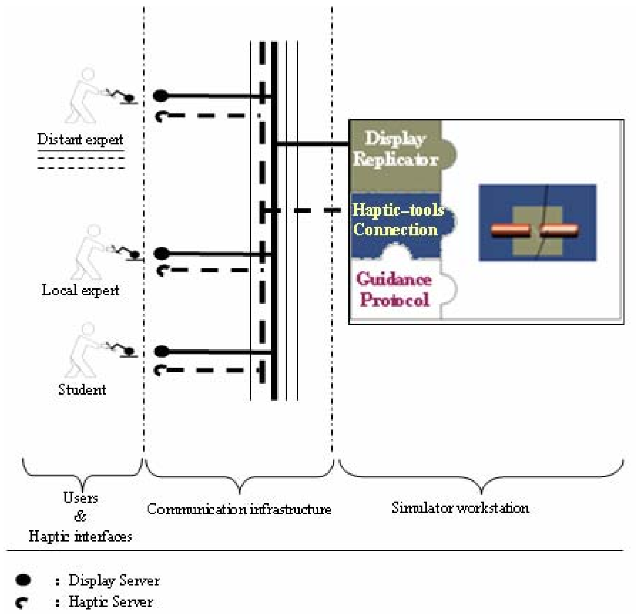

Figure 1: the architecture of the multi experts/one student tele-guidance surgical/suturing system

A. Platform of surgical and suturing simulation

Our platform includes the following principal components:

- SPRING Simulator ([6], [7], [8], [9]).
- Guidance protocol.
- Communication infrastructure.
- Haptic interfaces.

1) SPRING simulator

We have used the SPRING surgical simulator (an open source simulator developed jointly by the National

Bicomputation Center and SUMMIT, both at Stanford University) as a platform for our system. The internal architecture of this simulator is outlined in Figure 2

In SPRING, a world description information file describes objects that are created with specified geometric and interactive characteristics. Each object consists of arrays of

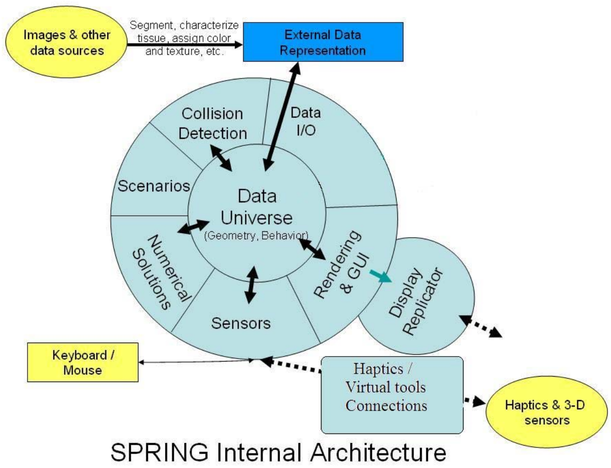

Figure 2: Simulator Architecture

Nodes, Edges, and Faces, although some objects may include only some of these parts. For example, the blood from tissue damage or a wound could be modelled as a particle system represented using only nodes. A suture thread could be modelled using only nodes and edges. In addition, 3-D objects that support cutting operations may also include arrays of tetrahedral (*tetra)* objects [7].

- **Simulation core:**

SPRING's core simulation loop processes the physical dynamics of each object (deformable, rigid-body, or suture dynamics) at every simulation time step. Suture dynamics are modelled by representing the thread as short, linear segments (edges) that form an articulating object. The thread object is constrained both by its contact with virtual instruments, tissues, and other , other objects, as well as by self-collision of constituent nodes. As one end of the thread is moved, the motion of each intermediate node is computed with a weighted, bi-directional *follow-the-leader* algorithm [7].

For deformable objects, the simulation system considers the nodes as point masses connected by spring/dampers on each edge, forming a 3D mesh for mass-spring simulation. Each edge is modelled as a 1D dampedspring employing linear, piece-wise linear, or non-linear force modeling. Edges may also be assigned different spring and damping coefficients, while nodes can be assigned mass distributions representing the inhomogeneity of actual anisotropic, heterogeneous tissues.

A number of numerical methods have been implemented for computing motions from the mass-spring forces, including traditional Euler and Runge-Kutta (2nd and 4th order). In addition, a quasi-static method, appropriate for heavily damped tissues and low interaction velocities, assumes that a tissue is always in static equilibrium, ignoring dynamic inertial and damping forces for a corresponding increase in simulation performance.

In addition, spring forces may propagate from the initial contact point with a tool to neighboring nodes, terminating the deformation when the deviation from a node's initial location falls below a minimum threshhold. This provides a significant (possibly an order of magnitude) speedup in many cases [7]. This is accomplished by breadth-first ordering of the nodes from of the tool's contact point using the number of edges from the contact point. The nodes are then processed in this level-based order, stopping computation when a level has no nodes that move more than the threshold amount. These regions may grow in size as a deformation continues, or decrease in size as the nodes return to rest.

This scheme allows SPRING to support more objects, each with a medium to large number of nodes and faces, since the simulation needs to process a much smaller set of nodes and edges on each simulation cycle. In addition, the limited extent of the deformed region may increase the performance of collision detection considering only nodes and face that have moved significantly.

- **Collision Detection**

SPRING first employed a node-node force-sphere collision model. Later, static partition methods, Axis-Aligned Bounding Boxes [11], and Oriented Bounding Boxes [12] were implemented. While these methods worked well in many cases, a more general scheme that better supported deformable objects was sought. For this reason, a Bounding Sphere algorithm [13] was adopted, incorporating enhancements for deformable objects [14]. The generality of this method, and its fast update capability, provided a reasonable trade-off for many cases. As with any hierarchical method, ultimately the detection method checks for collisions between primitive elements, such as face-face (surface collisions), edge-face (suture wrapping over vessel), and edge-edge (suture wrapping onto itself).

- **Display Replicator**

A mechanism for replicating the simulation's display is required for remote and collaborative viewing. SPRING includes a DisplayReplicator that copies the contents of the main display window at each screen refresh (in either stereo or monoscopic mode), compresses the image, and sends it to remote clients (expert, student) via standard network protocols. A simple DisplayServer program at the remote sites receives the image data, and decompress it to show the live video imagery of the simulation at real-time rates. The quality of the compression as well as the sending frequency can be adjusted to match network capabilities.

- **Sensors / Virtual Instruments :**

A *sensor* object in SPRING is an abstraction of a 6 degree of freedom (DOF) tracking and/or haptic device acting as the link between a physical input device and a graphical tool in the simulation. The real device sends the position and orientation information (i.e., a rotation matrix) to the sensor, and also one or more floating-point activation that represent the state of buttons, handles, or other controls associated with the device. A sensor is typically linked to a virtual instrument in SPRING's universe and the graphical object is transformed by updated sensor data. In addition, the activation values may affect the display and actions of the virtual tool, such as the hinge angle of a scissors or grasper, or the depth of the telescoping plunger of a syringe or resectoscope handle. SPRING supports the independent motion of each tool's subparts, with a subpart id denoted in each node.

The forces generated in the interaction of tissues, tools, thread, and other 3-D objects can be very useful to a learner. SPRING computes the forces of such contacts, and these are then sent via the sensor object to the actual haptic device across the network connection for timely force feedback.

Many virtual instruments types are provided in SPRING, including scalpel, needle, forceps, laparoscope, endoscopic, and others. Each may be linked to a haptic device (SensAble® phantom, Immersion® Bimanual Laparoscopic Device, and others) through a haptic Server program, which handles the device-specific interface, communicating with SPRING via network protocols. A haptic server program may be local to the simulation, or may be running on a remote client computer. In this way, interfacing details are decoupled from the simulation itself. In addition,, network connections may link a single simulation to several users almost anywhere on accessible computer networks. This method of network-based sensor- and haptic-servers inherently allows multi-user and multi-instrument interaction and supports collaborative procedures [7].

2) Guidance protocol

The guidance protocol is a method allowing an expert to guide a student's learning of surgical movements through the use of haptic, visual, audio and audio senses. We have developed a master-slave system employing the haptic interfaces that allow the expert to tele-link a student's haptic interface to the expert's device, allowing the student to feel the expert's touch in performing difficult surgical activities.

The tele-guidance protocol is divided into two cases, *show* and *monitor*, described here.

- *Show* (expert is the master): Here one of the experts controls the 3-D device, exerting force on the student haptic interface handpiece to demonstrate difficult surgical movements. Either the local or the distant expert may be designated as the driver, and the driver's movements are transmitted as forces to both the student's and the other expert's haptic device.
- *Monitor* (student is the master): In this case the student operates independently, using the 3-D interface to perform cutting, suturing, and other tasks. At the same time, these actions are transmitted to both the local and distant experts as forces via their haptic interfaces.

Note that 3-D position of the haptic device is not directly controlled by the computer, but instead the commands exert a force on the handpiece. In our experiments, this force value is based primarily on the vector between the controlling expert's haptic 3-D tool (the desired position) and the location of the student's handpiece. However, several effects, can cause oscillations in the force applied to the tool. This may be damped down by introducing a simulated increase of the inertia of the haptic device, including network latency and small motions of the users' hands. This inertial force value is dependent on both the network latency value, which can be measured, and the acceleration applied to the slave haptic device handpiece.

The control force equation here is applied to the motors of student (slave) haptic interface:

Force = a x (pos_diff) + inertia

Where: 'a' is an empirically obtained constant; 'Pos_diff' is the discrepancy between the master and slave positions, and inertia is a function of network latency, slave handpiece acceleration, and velocity.

3) Communication infrastructure

- **Network /telecom flows**

As can be seen in Figure 1, different flows of information are required in our system. Each requires a specific transmission quality that will be discussed below.

The audio–visual communication quality and bandwidth should be as high as possible to facilitate real-time collaboration between the experts and the student. The haptic control flow does not need high bandwidth, but it does require low network latency and also very low variation of the latency (jitter) for an acceptable haptic synchronization among the experts, the student, and the simulator.

Since the simulator itself needs high computational and network performance, a client-server protocol connection can be used, with the simulator on a high speed server, and smaller PCs used for haptic and display service for the student and experts. A peer-to-peer protocol is also possible if we install synchronized simulator applications on the PCs used by the student and experts, which will also run the haptic server programs. However this will require each PC to be a high performance machine, and also require synchronization of the simulation state between the peer simulator programs.

- **Display server (audio & graphical synchronisation)**

A remote user of the guidance protocol works through two programs running on the user's computer. These programs employ TCP/IP connections for two-way communications with the main SPRING simulation, each waiting for the SPRING to be the link.

DisplayServer shows a duplicate of SPRING's main display window, and also allows keyboard command of the simulation from the remote computer.

- **Haptics server (phantom server- haptic synchronisation)**

The second remote program is HapticServer, which provides a connection between haptic devices on a remote machine and the main SPRING program. The haptic server continually reads the output of one or more haptic devices, formats the geometric position and activation states of any buttons or controls of the device, and sends them to the connected SPRING program via TCP/IP.

HapticServer also accepts force commands from the SPRING program and converts them to appropriate calls for the particular haptic device. The force commands give the remote user the force feedback that SPRING has computed from collisions, weights of objects, and tool interactions with tissues in SPRING's simulated world.

SPRING accepts the geometric information to control the motions of tools used in the simulation. In collaboration mode, each user will control individual tools, simulating different roles in a surgical procedure. In guided mode, the force feedback sent to each of the haptic devices is the average of all the forces exerted by the users. This provides a sensation of hand-holding, helping a student to experience how the activity feels as well as how the procedure appears visually.

4) *Haptic interfaces*

The components "guidance protocol" and "haptic-tools connection" are designed in a flexible manner, which means that we can connect any haptic interface to the simulator. But for economic reasons and to make the platform available for large sector of medical students, we have used the Sensable PHANTOM Omni [2] in our experiments. It has a portable design and gives six degree-of-freedom positional sensing (horizontal, vertical and depth, pitch, yaw and roll) and three degree-of-freedom force sensing (Fx, Fy, Fz) calculated in Newtons.

III. Experiments

As is customary in our laboratory, we are working in a user-oriented approach ([18], [15], [19], [16], [17]). It means that we have designed several iterative user tests in order to validate (or not) our technical developments. This can be done through a partnership with Ecole Vétérinaire from Lyon (a veterinary surgeon school). We will interact with two user references: a teacher and a student, throughout the design process. They will follow each new software version and will help us design a well-suited ergonomics human-machine interface. Moreover, we plan to carry out five phases in this school, described here.

Observation phase: this first phase will enable us to define precisely the teaching situation we are dealing with. It will be our model for further development, but this model will need to be adapted as a human-machine interface. Indeed we will have to keep in mind to stay near this basis situation (natural situation), but, at the same time we will have to enrich this situation as our haptic interfaces are refined. The key is to keep in mind goals and constraints of the 'natural situation' and to verify during all the design process that our technical developments still enable users to achieve those goals without adding technical constraints.

Experimental phase1: When an initial software version is available, we will test it with several single users. Then we will validate the usability of the simulation and the quality of the force-feedback rendering, repeating until the interface is usable

Experimental phase2: We will test a two-user interaction in order to evaluate the ergonomic and technical quality of the handshake functionalities.

Experimental phase3: Here we will test a three-user interaction through a network. It will enable us to test feasibility of sharing fine control through haptic information.

Experimental phase4: Then, if all the other phases are successful we will test a situation where one teacher teaches to several students.

Each experimental phase will have the following protocol:

- basic training with the device
- task with a precise goal to achieve through the device
- questionnaire about interaction quality.

We will employ behavioural and verbalization feedback in additional to quantitative and qualitative data.

IV. Conclusion

The project's main purpose was to introduce medicine students or beginners surgeons to the field of telemedicine (tele-surgery) through applying haptic and force feedback and using total surgical environment simulation.

Experimental phase 1 is already reached and the iterative process has started. It will enable us to validate with user surgical environment simulation quality. Then it will be possible to investigate handshake functionalities.

Because rich multimodal interactivity between teachers and students is our main goal, and we will be able to evaluate interaction quality only if the first level of simulation is realistic enough.

References

[1] Delingette H. Réalité virtuelle et médecine. Rev *Électricité Électronique* 1997;8:43-5.

[2] http://www.sensable.com/ .

[3] Körner O, Rieger K and Männer R. Haptic Display for all Degrees of Freedom of a Simulator for Flexible Endoscopy. *Lecture notes in computer science,* 2004. vol 3078. pages 161-167.

[4] Berkley J, Weghorst S, Gladstone H, Raugi G, Berg D, Ganter M. Banded Matrix Approach to Finite Element Modeling for Soft Tissue Simulation. *Virtual Reality*, 1999. 4: p. 203-212.

[5] Berkley J, Turkiyyah G, Berg D, Ganter M, Weghorst S. Real-Time Finite Element Modeling for Surgery Simulation: An Application to Virtual Suturing *IEEE Transactions on visualization and computer graphics*, 2004 Vol. 10, N°3

[6] Brown J, Latombe J, Montgomery K. Real-Time Knot Tying Simulation. *The Visual Computer*, 2003. V20 (3). pp. 165-179.

[7] Montgomery K, Bruyns C, Brown J, Sorkin S, Mazzella F, Thonier G, Tellier A, Lerman B, Menon A. SPRING: A General Framework for Collaborative, Real-time Surgical Simulation. In: *Westwood J et al. (eds.): Medicine Meets Virtual Reality*. 2002. IOS Press. Amsterdam.

[8] Bruyns C, Montgomery K. (2002). Generalized Interactions Using Virtual Tools within the SPRING Framework: Cutting. In *J.D. Westwood. H.M. Ho_man. R.A. Robb.and D. Stredney, editors. Medicine Meets Virtual Reality 02/10, volume 85 of Studies in Health Technology and Informatics*. 2002. Pages 74-78. IOS Press.

[9] Bruyns, C., Senger, S., Wildermuth, S., Montgomery, K., Boyle, R.: Real-time Interactions Using Virtual Tools. In: *Niessen, W.J., et .al. (eds.) MICCAI* 2001, LNCS 2208, Springer, Berlin Heidelberg, (2001) pp. 1349-1351.

[10] Brown J, Sorkin S, Bruyns C, Latombe JC, Montgomery K, Stephanides M. "Real-Time Simulation of Deformable Objects: Tools and Application", Computer Animation 2001, Seoul, Korea, November 6-8, 2001.

[11] G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. Journal of Graphics Tools, 2(4):1–13, 1997.

[12] Gottschalk S, Lin M C, and Manocha D. OBB-tree: A hierarchical structure for rapid interference detection. In ACM SIGGRAPH 96 Conference Proceedings, pages 171– 180, 1996.

[13] Quinlan S. "Efficient Distance Computation Between Nonconvex Objects", Proc. IEEE Int Conf on Robotics and Automation, pp. 3324-3329, 1994.

[14] Sorkin S. "Distance Computation Between Deformable Objects", Honors Thesis, Computer Science Department, Stanford University, June 2000.

[15] Hennion, B. Gentaz, E., Gouagout P. & Bara F. (2005). Telemaque, a new visuo-haptic interface for remediation of dysgraphic children. *EuroHaptics* 2005.

[16] Klok, M., Uzan, G., Chêne, D., Zijp S. (2005). Blind gestural interaction: an experimental approach. *Gesture Workshop* 2005.

[17] Maillet, B., Guenand, A., Lenay, C., Chêne, D. (2005). Designing Mobile Phone. *DPPI* 2005.

[18] Rouzier, S., Hennion, B., Perez-Segovia T. & Chêne D. (2004). Teaching Geometry to Visually Impaired Pupils Using Haptics and Sound. *ICCHP* 2004.

[19] Tarrin, N., Petit G., Chêne D. (2006). Network Force-Feedback Applications for Hospitalized Children in Sterile Room. *Interaction Design with Children IDC06*, Tampere June 2006.

[20] Wu X, Tendick F. "Multigrid Integration for Interactive Deformable Body Simulation." In *S. Cotin and D. Metaxas. eds. Proc. Intl. Symp. Medical Simulation (ISMS 2004)*. Cambridge. MA. June 2004. SPRINGer-Verlag LNCS 3078. pp. 92-104.

[21] http://www.ijg.org/ .