Using Registration, Calibration, and Robotics to Build a More Accurate Virtual Reality Simulation for Astronaut Training and Telemedicine

Anil S Menon[1], Bobby Barnes[1], Rose Mills[1], Cynthia D Bruyns[1], Alexander Twombly[1], Jeff Smith[1], Kevin Montgomery[2], Richard Boyle[1]

[1]BioVis Laboratory, NASA Ames Research Center
Mail Stop 239-11
(011) Moffett, CA, 94035
USA

[2]National Biocomputation Center
Suit 1128
(011) Palo Alto, CA, 94304
USA

{asmenon, bbarnes, rmills}@stanford.edu; {bruyns, xtwombly,jdsmith}@mail.arc.nasa.gov; Kevin@biocomp.stanford.edu; rboyle@mail.arc.nasa.gov

ABSTRACT

Computer simulation of surgery and scientific experiments help in preparation, training, and assessment. These benefits can be further extended with the integration of robotics for teleoperation and assistance. We describe our efforts to build a realistic and useable simulation for astronaut training and experiment planning. Most of our development focused on user interaction with hand sensors, a necessary component for realism. For the hands, we developed a simulation and focused on aspects of fine tuning the registration and calibration to increase realism and functionality. This proved to be a necessary basis to integrate robotics and further the simulation's range of applications. Accurate registration, calibration, and robotic integration helped build a foundation for a useable simulation for astronaut training on ground and avenues of robotic assistance during flight.

Keywords: Registration, robotics, virtual reality, astronaut training.

1. INTRODUCTION

Surgical simulation offers a unique tool to improve physician performance and training [Satav92]. Its usefulness is highlighted in situations where practice is difficult as in the operating room, at remote locations, and in novel environments such as microgravity. In a similar capacity training for scientific experiments and protocols can be simulated for researchers and astronauts. Such a simulation allows for practice and accelerates learning and acquisition of new surgical skills [Otool99].

WSCG Short Papers Proceedings
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

Also, experiment and surgical simulation provide an easy method to record data and an ability to duplicate an experience, thereby, providing an objective means of quantifying surgical skill [Gorma99]. Most importantly, the practice and learning are done in the simulation and not on a patient or on an expensive experiment—reducing risk and overall cost.

These potential benefits have driven many groups to develop simulators [Baraf92; Keeve96; Kuhna00; Picin01; Terzo90; Ayach98; Berke99; Bosdo98; BroNi96; Szeke99]. In the arena of simulating experiments, astronaut training has been targeted because costs are high and opportunity for practice is minimal [Savag00]. In fact, a mock-up of experiments for the International Space Station is being designed [Smith02]. Both surgical simulation and astronaut training have the same goal of increasing proficiency and skill. In addition, astronaut training simulations can greatly benefit from telemedicine applications such as teleoperation where personnel is limited and telerobotic assistance can save valuable time.

At the forefront of simulation design is the simulation "Spring", developed by the Stanford/NASA National Biocomputation Center and recently open sourced [Montg02]. Spring is written in C++ to run on Windows, Unix, Irix or Linux. It can read in and render most 3D file formats and can display these objects in stereo display for virtual reality immersion. Additional functionality includes soft-tissue modeling, limited rigid body dynamics, suture modeling, and collision detection. The distributed nature of Spring allows it to be distributed over the Internet so that remote devices and users can interface with the simulation through different types of sensors (Figure 1).

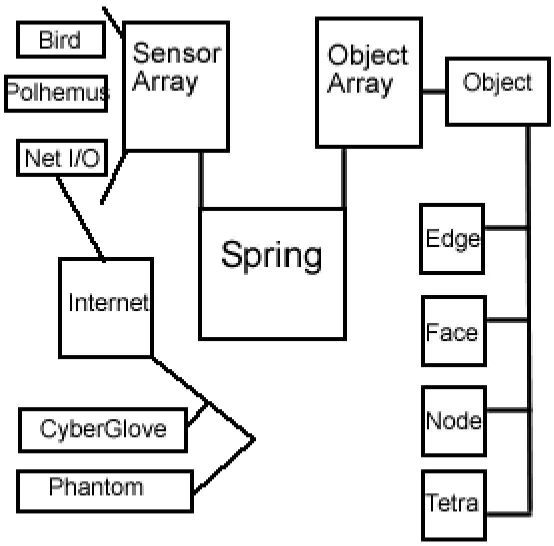

Figure 1. Spring simulation engine interacts with virtual objects through node, edge, face, and tetra arrays and interfaces with a variety of devices over the Internet through the same sensor framework.

Object registration—accurately mapping object position in real world space to virtual space—is an important aspect contributing to the simulations usability. An accurately registered simulation makes the interaction more believable and it can make the difference between negatively training motor skills or improving performance. Poor registration detracts from the simulation and can be disturbing enough to cause motion sickness [Pauch92]. Over time the user will adapt to the registration differences and be able to use the simulation accurately [Welch78]. Still, these adaptations may lead to negative training of motor skills and counterproductive training, underlying the importance of accurate registration [Yokok99]. Also, registration plays a key role in extending a virtual environment beyond just simulation to actual real world interaction through robotics. A robot must relate to specific positions in a virtual world if it is going to usefully interact with the two. With this functionality, Spring could monitor the users hand movements and allow the user to pick up a vial or use a scalpel in the real world and virtual environment simultaneously. As the user sweeps a hand across an environment or a body the distance and position will feel like it would in the real world. Augmented reality systems currently use cameras to monitor position and display real world objects in video [Bajur95; Kutul98; Yokok96].

Spring offers the ability to improve registration at a lower computational cost, in real-time, as well as integrating many different objects and sensors easily. Registration delay can be separated into static delay associated with the hardware and setup and dynamic delay associated with the running simulation [Azuma97]. The dynamic delay related to latency of response contributes the most to this error and is greatly reduced by a real-time simulation like Spring with little delay [Hollo95].

Robotics goes hand in hand with experiment simulation and improvements through registration. Robotics and teleoperation are needed for NASA applications and include manipulation in space and planning (NASA89; Kim94). These applications can be forwarded by a well-integrated virtual reality teleoperation system. A robotic arm can assist in experiments, can provide access to remote operators, or can be used in teaching, assessment, or training. The virtual interface facilitates these goals by integrating and simulating a real and interactive world. Such an interface can put users in a shared environment or allow them to interact with the environment [Karlg95; Benfo94). Improved accuracy of Spring's simulation directly increases functionality when robotics is added to the simulation for teleoperation purposes. An accurate simulation leads to accurate manipulation and together make a useable virtual and real world interface.

2. REGISTRATION CALIBRATION AND TELEOPERATION FOR HANDS

The hand is a natural mode of human interface. It is easily understandable and provides universal functionality. With a working hand model users can interface and use any device in a simulation be it a laproscope, scalpel, or vial. The hand is also more challenging than most tools because of its diverse functionality and user familiarity with its operation. We chose to use the hand as a major tool to focus on for user immersion. Registration is the first step in making it functional, by allowing the user to realistically see their hands in the simulation and move them as desired.

The realism and functionality provided by good registration is further extended by equally well represented movement of articulating objects such as a hand and fingers. Finely tuned manipulation increases functionality in tasks such as opening vials and it furthers realism and immersion. A quicker and more accurate calibration routine is needed to allow finer hand manipulation and adapt to a number of different users. Different users have different hand sizes, and the sensors sit differently on each user's hand. This creates the need for a calibration routine that each user can perform prior to using the glove. Thus the underlying details are hidden such as gain and offset movements of each sensor. A faster calibration routine is desired to allow users to easily begin the simulation.

Together, accurate registration and calibration will allow the user to not only interact with the virtual world but also the real world through telerobotics. Advancements in tracking the hand and representing it in the virtual world directly translate into our efforts to better position the robotic extension of this simulation and control a robot with the users hand movements.

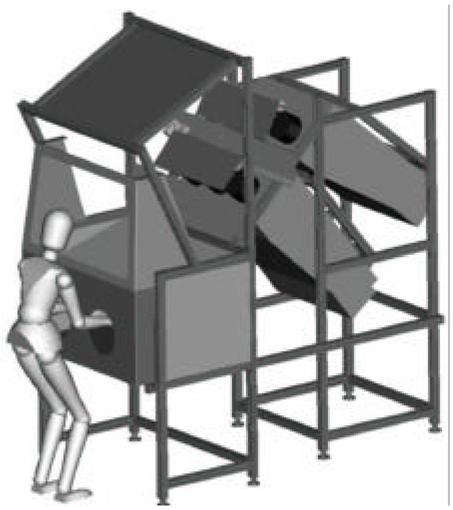

Figure 2. In the glovebox model, a stereoscopic image is projected on a screen in front of the user. The user reaches under the screen to interact with virtual environment and the real world.

3. BUILDING AN ACCURATE AND FUNCTIONAL SIMULATION

The prototype described here and pictured in figure 2 allows a user to interact with a virtual environment with two hands and view the virtual representation exactly where the interaction is occurring. The display is created with twin LCD projectors with circular polarization options to provide high-resolution images (1280 x 1024 pixel resolution). The user looks down into the display with circularly polarized stereo glasses at a 3D virtual environment 40x30x24 (width x length x height inches).

The virtual world represented also models a sample glovebox and experiment for the International Space Station Life Science Glovebox (Figure 3). An example model contains the boundaries of the glovebox and represents the working space beneath the screen. It contains tools such as forceps, scalpel, syringe, or any object to be used such as the virtual rat for dissection [Bruyns01]. Objects represented in such an image can be attached to sensors over the Internet through Spring.

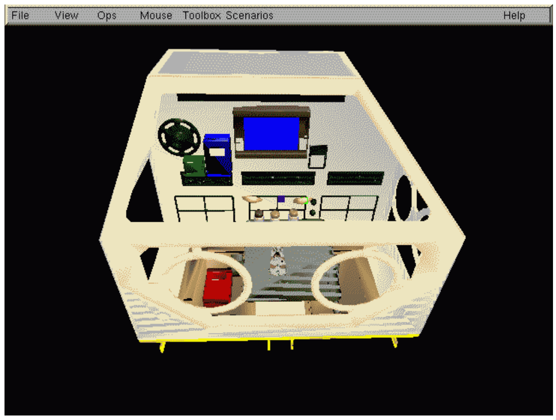

Figure 3. The projected 3D image of the glovebox contains functional tools, hands, and components necessary for an experiment such as manipulating the soft tissue of a rat.

3.1 Registering Hand Position

Registration of object position in this prototype will focus on the hand models since they are a major mode of user interaction. Setup in the real world requires an adequate operational area in front of the user for a full range of arm and hand motion. Any operational area is directly below the screen where the virtual image is displayed. This is vital for the virtual world to parallel the real world and allow the user to gain a realistic experience from a simulated interaction. Displaying the user's interaction in an area other than the workspace might lead to incorrect motor training during experiment or surgical simulation.

For similar reasons, the dimensions of the operational area should also mirror the virtual world, and having similar object sizing is important for registration. To do this physical units such as millimeters were kept consistent between devices. A mock glove box, in this prototype, which the user physically reaches into measures 1000 mm X 400 mm X 400 mm. With the use of CAD software, a virtual model was created with comparable sizing and appearance, including the external holes that allow the user to enter the mock-up and simulation. Equal dimensions facilitate registration by allowing a one to one correspondence between real and virtual position. The same rule must be applied to all objects in the virtual world. In this case, objects in the scene and tools are proportional, and the virtual hands are also proportional to a reasonable hand size approximately 170 mm X 80 mm X 20 mm.

Hand position and motion of the user is measured with magnetic trackers and sensor gloves. CyberGlove® (Immersion Corp.) is used here, with the ability to measure three flexion sensors per finger, four abduction sensors, a palm-arch sensor, and sensors to measure flexion and abduction. Thus, specific hand conformations can be translated into a virtual representation. Positions of the two hands in space are registered using magnetic transmitters and receivers (Flock of Birds®, Ascension Corporation). Such a tracker introduces error into the simulation which increases if it is nearby other magnetic fields and metal objects [Nixon98]. Error will impede any system for accurate registration no matter how well designed. Optical trackers can reduce this problem but they require a direct line of sight to function properly which may be difficult to maintain in the operational area unimpeded [Bisho84; Ward92]. To reduce error, sensor cables are run axially up the users arm to avoid interference and large pieces of metal surrounding the mock-up were replaced with non-conductive synthetic materials. Algorithms can be applied to reduce error but may decrease computational speed and may not be portable or general [Livin97]. Also, the operational space must be kept within the 3 foot range of the tracker. Together our standard deviation from a given point was 2.7mm.

CyberGloves® interact with the main simulation engine by way of a networked server (Figure 4). Position and angles from the CyberGloves® are interpreted by the device drivers for each hand and the information is sent out in packets using TCP. Each cycle of the main simulation loop updates information from the CyberGloves® and then updates the display to register the new position, rotation, hand angles and additional functionality such as behaviors like grabbing. Since the CyberGloves® can send information at 144Hz and the main simulation engine can run >1000Hz the user can interact with the virtual environment in real time.

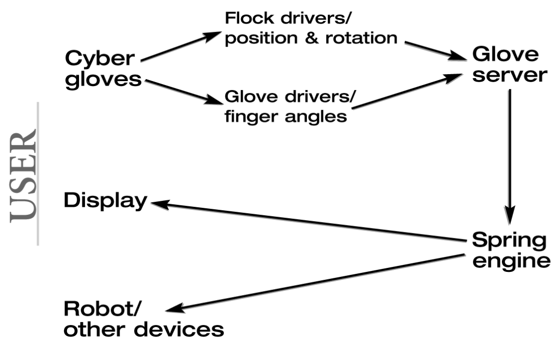

Figure 4. CyberGlove integration into simulation loop.

An important element in registering position accurately in this design is to initialize the position of the gloves by subtracting the actual values and starting the relative motion from zero. With corresponding measurement scales, the hands can be positioned in the desired location in the virtual world and real world. Then, after they are initialized they will begin in the correct location.

For motion in one world to be consistent with the other the coordinate axes must be aligned and the relative motion must also be in scale. Aligning the axes is easily achieved on the server side so that motion in any direction is converted into the same for the simulation engine. Drivers from the magnetic trackers report in centimeters, so scaling the reported data by a factor of 10 allows for corresponding motion, since the main simulation engine operates in millimeters. Further signal specification can be achieved by building a virtual scale and moving the tracker from known real distances and scaling the reported motion accordingly. This works in situations where interference or distance requires a different scale or non-linear scale such as areas with metal causing distortion variation.

These methods were chosen for simplicity and speed in a real time simulation. Other alternatives include using forces to move objects in space, which would aid in rigid body dynamics of the simulation but is difficult to maintain accurate registration. Also, visual methods such as monitoring object position with a camera were avoided because of the computational overhead associated with such an algorithm. Speed is an essential factor for this simulation as it provides better and more realistic training if it is run in real time. This is all achieved by using the Spring engine and these registration techniques. Reduced latency, as compared to augmented reality simulations, plays a large role in more accurate registration. Also, cameras provide another alternative but slow the simulation engine and require sacrifices in device input. In the current paradigm, multiple objects and sensors can be added and accurately registered, so a user can use a hand to grab a haptic device and receive the correct visual and tactile sensation from the virtual display.

3.2 Calibrating Hands for Believable Performance

The objectives of this project were to develop an improved calibration routine to better track the movement of different sized hands and allow many different people to use the glove with ease by being fast and user friendly. In addition to improving the calibration routine, improvements to the Spring hand model were also needed: identifying the joint structures and movements, adding abduction into the Spring model, and improving overall thumb motion.

The calibration routines that come packaged with the CyberGlove Device Manager result in a rough approximation of finger movement. The default calibration routine involves placing the hand in two different conformations (Figure 5). There is also the option for a more advanced, finer calibration, which allows the user to adjust each individual sensor gain and offset.

Figure 5. The two default hand positions used by the Immersion calibration routine.

One calibration approach has been to use devices such as wooden blocks to position the hand at known positions and relate the sensor values to those known positions [Kessl95]. Another approach has been to use a kinematics model with the thumb and index finger [Griff00]. Both approaches had limited resulting motion, and the kinematics model did not account for thumb roll.

The general calibration approach for this project was to collect data from several different hand positions (Figure 6) and use linear regression to relate the sensor values with hand position. At each hand position, the actual angle of the joint (based on the hand conformation) is saved along with the sensor value. The actual joint angles are the independent variable, while the sensor values are the dependent variable.

Figure 6. Examples of hand positions that may be used for calibration to capture different motions.

$$\text{Slope} = \Sigma[(x_i - x_{avg})(y_i - y_{avg})] / \Sigma(x_i - x_{avg})^2$$
$$\text{Intercept} – y_{avg} – \text{slope} * x_{avg}$$

These calculations signify the end of the calibration process. During the simulation all that is needed are the slope, the intercept, and the current sensor value. This information is then used to calculate joint angles in the hand.

Joint angle = sensorValue*slope + intercept

These equations may also be rearranged slightly into the terms of sensor gain and offset, and then saved to a file so that the calibration application can be separated from the simulation software.

Collecting data from several different hand conformations allows for a more accurate mapping of sensor values to hand position. In addition to providing a more accurate mapping, the routine is fast and simple.

3.3 Using an Accurate Simulation to Interact with the World

The development and implementation of a “robot server” program incorporates the real-time usage and functionality of a mechanical robot with our Spring simulation engine. The robot that is currently used is a variable configuration Robix™ Rascal Beta 0.2.17 robot with six motor servos. The robot server program’s main functions include, (1) Establishing a TCP/IP communication link between the Spring simulation or another device and the PC hosting the mechanical robot, (2) Computing the kinematics necessary to translate positions sent from a controlling device into servo motor rotations, and (3) Providing an interface for data passed to and from the robot over the host PC’s parallel port.

3.3.1 Passive TCP/IP Link: Force-based

As mentioned in the list of functions above, the first step to the development of a robot server was to establish a TCP/IP communications link. Two modes of operation were developed for this TCP/IP link for the robot server. The first mode, referred to as "passive," simply has the robot server software open a port on the host PC and listen for a connection from the Spring simulation engine. The benefits of doing this are twofold. First, this minimizes the amount of knowledge that the robot server software needs to know about what and where its information is derived, making it the responsibility of Spring to connect to it. Secondly, computational gains are achieved since this programming design allows the Spring simulation engine and the robot server to operate on different machines.

After the communications link between Spring and the robot host PC is established, the robot server program makes a connection to the robot's electronic interface over the parallel port. Each of the six motor servos on the robot are initialized to the initial positions of various parts (arms, joints, motors, etc.) that correspond to a particular robot design configuration created by the Robix™ software.

Communication links are created between Spring, the robot server, and the robot, the server program enters a loop where it continuously reads its open socket for packets from Spring and drives the servos of the robot where necessary. Several steps actually take place, however, between the reception of a data packet from Spring and the activation of the servos on the robot.

Spring sends force vectors and activation values to haptic devices that are networked to it. Thus, the packets received by the robot server contain a force in the x, y, and z directions and an activation value for the gripper on the robot. These forces must be translated into a new position to place the endeffector of the robot (Equations 1-4). This is done by taking the current position of the tip of the robotic arm and adding to it a scaled quantity of the original force vector received from Spring (Equation 4).

Eq1. Input Force Vector = (f_x, f_y, f_z)
Eq2. $a_{(x,y,z)} = f_{(x,y,z)} / m$
Eq3. $p_1 = 0.5 * t^2 * a_{(x,y,z)} + p_0$
Eq4. $p_1 = a_{(x,y,z)}/(\text{scale}) + p_0$

m = mass	p_0 = old position
a = acceleration	p_1 = new position
f = force	

After this is done, this new position in space is passed to a software library that does the inverse kinematics to determine what the rotations of each of the robot's servos should be to put its endeffector at this new position in space.

This is a very nontrivial task because the complexity (number of joints) of the current robot design provides an infinite number of solutions for any given point in space for the endeffector. (See Figure 7) However, each of the motor servos has physical limitations of +-90 degrees of motion. These limitations on each of the joints allow for a deterministic kinematics solution to be found if certain assumptions are made.

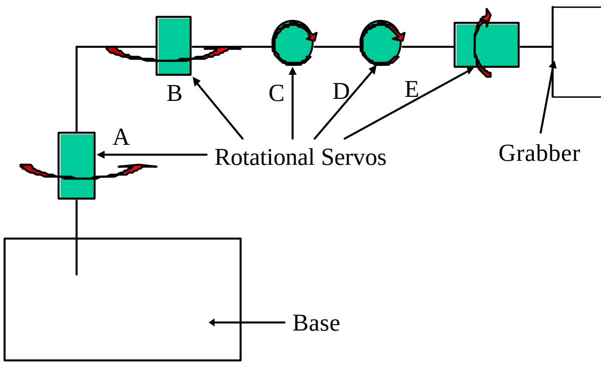

Figure 7. Robot Configuration.

3.3.2 Active TCP/IP Link: Position-based

The other mode of operation of the robot server program creates a direct TCP/IP link with a controlling device such as a Phantom® or a CyberGlove®. This mode is referred to as "active" since the robot server program is in control of establishing the necessary connections between itself and other devices and the Spring simulation is not involved in the communications loop.

Since the robot server is connected directly to the phantom or CyberGloves®, the packets received from these devices do not contain forces. Instead, they contain a 3-D position, a rotation matrix, and activation values (the number of these depends on the particular device). Building the communications protocol in this way allows us to use the same server programs for each respective device, which are usually used to connect to Spring, to connect directly to the robot server. 3-D positions do not need to be translated as forces do; thus, they are passed directly to the inverse kinematics routine to calculate the joint rotations needed to place the robot's endeffector at that position in space.

4. PERFORMANCE

While the robot sever program is running the internal update rate is monitored and displayed on the console window. In addition, the update rates of other systems connected to the robot sever via TCP/IP are estimated and displayed by computing the number of packets received from the device each second. This feature allows us to monitor the overall performance of the system. Through testing it has been observed that the phantom has an update rate of approximately 1kHz, the CyberGloves® have a rate of approximately 144Hz, and the robot has a maximum update rate of 4Hz. The update rate of the Spring simulation engine is not in this list because Spring is setup to only send one packet for each one it receives from a particular device. Thus, from the device's view point Spring always has an update rate equal to it's own. The robot's low update rate is limited by the speed in which the robot's drivers will accept commands. The disparity between the higher update rates of the controlling devices and the low rate of the robot causes the robot to exhibit intermittent motion while these other devices sweep smoothly in any given direction. Furthermore, a small lag time can be observed between actions made at the controlling device and corresponding actions at the robot since the robot server software drops all packets in its receive buffer while keeping only the last (most recently received).

5. CONCLUSION

We have developed a virtual reality simulation for astronaut training that can be immediately extended to any experimental protocol or surgical simulation. After working on integrating hardware and software we focused on aspects most important to making the simulation believable and useable so that it can effectively train and simulate experiments that would otherwise only be experienced in microgravity. Most important in increasing the functionality and realism of the simulation proved to be registration and accuracy as well as extensions into real world interaction through robotics. Further evolution of these elements would provide maximum benefit in improving the simulation. With registration and calibration our techniques allowed for real-time operation and confined the major source of error to static limitations of the sensing devices (i.e. accuracy was limited by the devices). In robotic interaction we were device limited by the robot driver hardware's slow update rate.

Together, these elements, after improvement, provided a virtual environment that can be tested by astronauts, used for training, and, more importantly, provides an avenue for remote operation and administration of simple experimental tasks.

6. ACKNOWLEDGEMENTS

We would like to acknowledge Kevin Montgomery and the National Biocomputation Center at Stanford for help with further development of Spring for Astronaut training applications. CJ Slyfield for 3D tool development. Also, Stanford Medical School and Electrical Engineering Department for support.

7. REFERENCES

[Ayach98] Ayache, N., Cotin, S., Delingette, H.: Surgery Simulation with Visual and Haptic Feedback. In *Robotics Research,* pp. 311-316, 1998.

[Azuma97] Azuma, R.: A Survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, Vol. 6, No. 4, pp. 355-385, 1997.

[Baraf92] Baraff, D., Witkin, A.: Dynamic simulation of nonpenetrating flexible bodies. *Computer Graphics*, Vol. 26, No. 2, pp. 303– 308, 1992.

[Benfo94] Benford, S., Bowers, J., Fahlen, L, Grenhalg, C.: Managing mutual awareness in collaborative virtual environments. *Proceedings of VRST'94,* 1994..

[Berke98] Berkley, J., Weghorst, S., Gladstone, H., Raugi, G., Berg, D., Ganter, M.: Fast Finite Element Modeling for Surgical Simulation. *Proc. Medicine Meets Virtual Reality (MMVR'99),* pp. 55- 61, 1998.

[Bisho84] Bishop, G.: Self-tracker: A smart optical sensor on silicon. *Ph.D. Thesis, University North Carolina, Chapel Hill,* 1984.

[Bosdo98] Bosdogan, C., Ho, C., Srinivasan, M., Small, S., Dawson, S.: Force Interaction in Laparoscopic Simulation: Haptics Rendering of Soft Tissues. *Proc. Medicine Meets Virtual reality (MMVR'98)*, pp. 28-3, 1998.

[BroNi96] Bro-Nielsen, M., Cotin, S.; Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation. *Proc. Eurographics'96,* Vol. 15, pp. 57-66, 1996.

[Bruyns01] Bruyns, C., Montgomery, K., Wildermuth, S.: Advanced astronaut training/simulation system for rat dissection. *Medicine Meets Virtual Reality (MMVR'01).*, 2001.

[Gorma99] Gorman, P., Lieser, J., Murray, W., Haluck, R., & Krummel, T.: Evaluation of Skill Acquisition Using a Force-Feedback, Virtual Reality-based Surgical Trainer. *Medicine Meets Virtual Reality,* pp. 121-123, 1999.

[Griff00] Griffin, W., Findley, R., Turner, M., Cutkosky, M. Calibration and mapping of a human hand for dexterous telemanipulation. *ASME IMECE 2000 Symposium on haptic Interfaces for Virtual Environments and Teleoperator Systems,* 2000.

[Hollo95] Holloway, R.. Registration errors in augmented reality systems. *Ph. D. Dissertation TR95-016, Department of Computer Science, The University of North Carolina,* 1995.

[Karlg95] Karlgren, J., Bretan, I., Frost, N., Jonsson, L.: Interaction models, reference, and interactivity for speech interfaces to virtual environments. *Proceedings of second Eurographics Workshop on Virtual Environments – Realism and Real Time.*

[Kim94] Kim, J., Weidner, R., Sacks, A. Using virtual reality for science mission planning: A mars pathfinder case. *ISMCR 1994: Topical Workshop on Virtual Reality,* pp. 37-42, 1994.

[Keeve00] Keeve, E., Girod, S., Girod, B. : Craniofacial surgery simulation. In *Proceedings of the 4th International Conference on Visualization in Biomedical Computing (VBC '96)*, pp. 541–546, 2000.

[Kessl95] Kessler, G., Hodges, L., Walker, N.: Evaluation of the CyberGlove as a whole-hand input device. *ACM Transactions on Computer-Human Interaction,* Vol. 2, No. 4, pp. 263-283, 1995.

[Kuhna00] K¨uhnapfel, U., Akmak, H., Maaß, H. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers & Graphics*, Vol. 24, pp. 671–682, 2000.

[Livin97] Livingston, M., Slate, A. (1997). Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators and Virtual Environments,* Vol. 6, No. 5, pp. 532-536, 1997.

[Montg02] Montgomery, K., Bruyns, C., Brown, J., Thonier, G., Tellier, A., Latombe, J., Lerman, B., Menon, A. Spring: A General Framework for Collaborative, Real-Time Surgical Simulation. Medicine Meets Virtual Reality (MMVR'02). Pp. 23-26, 2002.

[NASA89] NASA.: *Proceedings of the NASA Conference on Space Telerobotics, JPL Publication 89-7,* pp. 1-5, 1989.

[Nixon98] Nixon, M., McCallum, B., Fright, R., Price, B.: The effects of metals and interfering fields on electromagnetic trackers. *Presence: Teleoperators and Virtual Environments,* Vol. 7, No. 2, pp. 204-218, 1998.

[Otool99] O'Toole, R., Playter, R., Krummel, T., Blank, W., Cornelius, H., Roberts, W., Bell, W., Raibert, M.: Measuring and developing suturing technique with a virtual reality surgical simulator. *J. Amer Coll Surgeons*, Vol. 189, No.1, pp. 114-127, 1999.

[Pausch92] Pausch, R., Crea, T., Conway, M.: A literature survey for virtual environments: Military flight simulator visual systems and simulator sickness. *Presence: Teleoperators and Virtual Environments*. Vol. 1, No. 3, pp. 344-363, 1992.

[Picin01] Picinbono, G., Delingette, H., Ayache, N.: Non-linear and anisotropic elastic soft tissue models for medical simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[Satav97] Satava, R. Robotics, telepresence, and virtual reality: a critical analysis of the future of surgery. *Minimally Invasive Therapy,* Vol. 1, pp. 357-363, 1997.

[Savag00] Savage, P., Smith, J. Utilization of virtual environments for astronaut crew training. *SAE Technical Paper Series*, 2000.

[Smith02] Smith, J., Gore, B., Dalal, M., Boyle, R.: Optimizing Biology Research Tasks in Space Using Human Performance Modeling and Virtual Reality Simulation Systems Here on Earth. *ICES 32nd Annual Conference*, 2002.

[Szekely99] Szekely, G., Bajka, M., Brechbuhler, C., Dual, J., Enzler, R., Haller, U., Hug, J., Hutter, R., Ironmonger, N., Kauer, M., Meier, V., Niederer, P., Rhomberg, A., Schmid, P., Schweitzer, G., Thaler, M., Vuskovic, V., Troster, G.: Virtual Reality-Based Surgery Simulation for Endoscopic Gynecology. *Proc. Medicine Meets Virtual reality (MMVR'99),* pp. 351-357, 1999.

[Terzo90] Terzopoulos, G., Waters, K. (1990). Physically-based facial modelling, analysis, and animation. *The Journal of Visualization and Computer Animation,* Vol. 1, pp. 73–80, 1990.

[Ward92] Ward, M., Azuma, R., Bennett, R., Gottschalk, S., Fuchs, H.: A demonstrated optical tracker with scalable work area for head-mounted display system. *Proceedings of 1992 Symposium on Interactive 3D Graphics,* pp, 43-52.

[Welch78] Welch, R.: Perceptual Modification: Adapting to Altered Sensory Environments. *Academic Press*, ISBN 0-12-741850-4, 1978.

[Yokok99] Yokokohji, Y., Hollis, R., Kanade, T.: WYSIWYF Display: A visual/haptic interface to virtual environment. *Presence: Teleoperators and Virtual Environments,* Vol. 8, No. 4, pp. 412-434.