Distributed Training System with High-Resolution Deformable Virtual Models

Felix G. Hamza-Lup[1,2] Anand Santhanam[1] Cali Fidopiastis[3] Jannick P. Rolland[1,2,3]

(1) School of Computer Science
(2) College of Optics and Photonics: CREOL/FPCE
(3) Institute for Simulation and Training
University of Central Florida
{ fhamza , anand , cali , jannick }@odalab.ucf.edu

ABSTRACT

Virtual environments (VEs) allow the development of promising tools in several application domains. In medical training, the learning potential of VE is significantly amplified by the capability of the tools to present 3D deformable models in real-time. This paper presents a distributed software architecture that allows visualization of a 3D deformable lungs model superimposed on a human patient simulator at several remote trainee locations. The paper presents the integration of deformable 3D anatomical models in a distributed software architecture targeted towards medical prognostics and training, as well as the assessment of the shared state consistency across multiple users. The results of the assessment prove that with delay compensation, the distributed interactive VE prototype achieves high levels of shared state consistency.

Keywords

Medical simulation, real-time deformable models, distributed interactive systems, virtual environments, augmented reality.

1. INTRODUCTION

Virtual Environments (VE) allow the development of promising tools in several application domains from design and manufacturing [33] to medical applications [12, 28]. This paper focuses on distributed medical applications. Specifically, we present the design and integration of deformable 3D anatomical models in a distributed, interactive VE targeted towards medical training and simulation.

The distributed simulation capability can substantially facilitate experts' interactions, especially during quick-response conditions such as medical emergencies [30], and has the potential to become an efficient medical training tool. The deformable 3D models combined with remote interactivity have the potential to enhance training effectiveness, allowing participants to perform various complex medical procedures with frequent exposure to the techniques and without the cost of traveling. We anticipate the diffusion of the simulation techniques presented in this work to many other application domains.

The paper is structured as follows. Section two presents related work regarding deformable 3D models as well as several related research efforts in distributed interactive VEs. Section three summarizes the approach used for 3D model deformation and provides an application domain analysis. It then introduces the deformation distribution and the consistency maintenance approach. Section four illustrates the prototype implementation and the experimental setup. Experimental results and preliminary assessments are presented in section five, followed by discussions, conclusions, and future work in section six.

2. RELATED WORK

We first acknowledge the most significant physically-based deformation techniques followed by a brief review of shared state consistency management in distributed interactive systems.

2.1 Deformable Models

Physically-based deformation has been applied in areas ranging from animation to surgical simulation. Initial models for such deformations were pioneered by [29]. These models represent the mechanics of deformation as a second-order differential equation of the position change for a vertex in a 3D model. The approach was further modified to simulate plasticity and fractures by [29], and the differential equations were replaced by a first-order linear equation to simulate complex deformations [9] with high computational complexity. Some of the improvements that lowered the computational complexity were focused on either decimating the 3D model or simplifying the equations' complexity. Graph simplification algorithms [7] and matrix simplification methods [4] were combined using multi-resolution wavelets [17].

Of particular importance to real-time simulation is the linearization of deformations. Early work used iterative solutions for the first-order differential equation of deformation [32]. Iterative methods were further investigated in [2]. Once the deformation is computed for a given force, it is linearly scaled for any force that deforms the object. This fact, combined with the previously mentioned approaches, suggested a possible simplification of deformation computation for real-time simulations. The previous methods were still computationally expensive for complex deformations. The deformations' non-linearity occurred especially for large and complex deformations. The methods proposed for compensation [34] did not allow any pre-computation.

Another technique to simulate deformations is to represent the change in shape in terms of a set of basis functions. Early work in this direction can be found in [4, 6]. A Green’s function is used to represent the required basis functions. Interactivity was enhanced by smooth interpolation among a set of pre-computed deformations.

A clear trade-off exists between the computational complexity and the deformation accuracy representation. For distributed VEs, the problem is further aggravated by the communication latency that leads to an inconsistent shared state among remote participants, as further discussed.

2.2 Distributed Interactive Simulations

In distributed interactive simulations the interactions and information exchanges generate a state referred to as the dynamic shared state that has to remain consistent for all participants at all sites. The interactive and dynamic nature of a distributed simulation is constrained by the communication latency.

One of the first and most intensive efforts in building a distributed interactive simulation was the SIMNET project started in 1983 [20] followed by the Naval Postgraduate School's NPSNet [35]. Since then a number of consistency maintenance techniques have been employed in distributed VEs and the research efforts can be grouped in four categories. The first category consists of approaches to optimize the communication protocol through packet compression and packet aggregation. The second category includes data visibility management, which makes use of area of interest management [11] and multicasting [18] and is focused on reducing the bandwidth throughout the system. Taking advantage of the human perceptual limitations, like visual and temporal perception, [27] constitutes the third category. The fourth category investigates systems' architecture [14].

While the visualization of 3D deformable models for medical training may be done in fully immersive environments, we belive that augmented reality (AR) systems offer advantages. AR systems were proposed in the mid '90s as tools to assist different fields: medicine [10], complex assembly labeling [5], and construction labeling [31]. With advances in computer graphics, tracking systems, and 3D displays, the research community has shifted attention to distributed collaborative environments that use extensively the AR paradigm [3, 26]. To our knowledge, none of these AR distributed environments use high-resolution real-time deformable 3D models.

3. A DISTRIBUTED AR TRAINING TOOL

We present a distributed interactive tool aimed at medical training that allows visualization of deformable 3D medical models. Within the system, the AR component takes place at each distributed site, however, in different ways. At the trainer's site, the trainer and other potential trainees can see each other as well as the 3D lungs model augmenting their office space. At the trainee site, the participants can see the 3D lungs model superimposed on a Human Patient Simulator (HPS). We have presented an early integration of such a unique software/hardware system in [21].

The distributed interactive AR medical training system was developed around the following simple scenario; A HPS is used as an accident victim with several facial, neck and shoulder abrasions and multiple broken ribs. A student (the trainee) in a practice intensive care unit is responsible for stabilizing the patient, while the instructor (the trainer) located in his office observes the procedure and changes the simulations parameters (e.g. breathing rate, tidal volume). The instructor may simultaneously train several students performing live saving procedures on different HPSs.

All users wear lightweight head-mounted displays (HMDs) [19] (Figure 1) optimized for near field visualization (i.e. 1 meter) as required while working with the HPS.

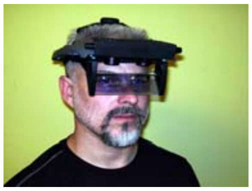

Figure 1. Lightweight (i.e. <600g) HMD.

The instructor(s) and other trainee(s) wear similar HMDs in the instructor's office and are able to visualize the 3D lungs model while seeing and interacting with each other in a natural way as illustrated in Figure 2a. Moreover, the instructor(s) and local trainee(s) are able to change the simulation parameters of the deformable lungs such that one or more remote trainees shown in Figure 2b are exposed to different medical emergency scenarios.

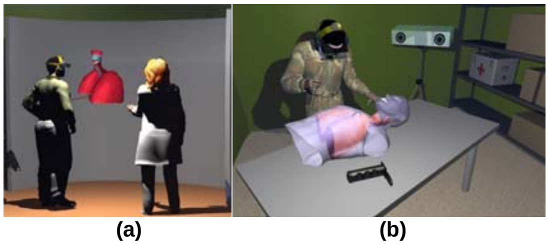

Figure 2. (a) Trainer and a trainee interacting with 3D models and (b) Remote trainee performing a medical procedure where the dynamic parameters of the models are created by the participants in (a).

The remote students interacting with the HPS can see the changes in the lungs behavior and will adopt different procedures on the simulator in order to stabilize the patient. As the trainee interacts with the HPS s/he triggers changes in the lungs behavior. These changes are visualized immediately in the deformable lungs model that augments the trainer's office. Such a training tool has the potential to:

1. Simultaneously train local and remotely located students.
2. Allow students to actually "see" and therefore better understand their actions on the HPS which affect the behavior of the deformable lungs model.
3. Allow an instructor to change the simulation parameters of the deformable lungs and practice different emergency scenarios (e.g. lungs collapse, different ventilatory patterns).

3.1. Deformable Lungs Model

It is beyond the scope of this paper to provide a detailed physiologically-based working of the lungs and detail the algorithms for deformation; the details of the mathematical methods and algorithms have been reported elsewhere [25] [22-24] however the main principles, assumptions, and methods are now summarized.

The steps to model the deformation involve computing the total volume of air that is inside the lungs during breathing as a function of the change in pressure inside the lungs, followed by the computation of the local deformation of the external surface of the lungs caused by the regional expansion of the alveoli. We may regard the 3D model as a polygonal or a wired mesh model. The main objective of the deformation computation in this scenario is to model elastic deformations caused by an increase or a decrease in the lungs volume and represent them as a deformable model. Air "flowing" into the model, generates a force upon each vertex.

The forces applied at each vertex of the mesh get transferred, in part, to their neighbors based on the local elasticity of the lung tissue. Each vertex has an associated attribute that represents the local elasticity and the connectivity to other nodes. This attribute sets the amount of force that is transferred in part to each vertex's neighbors. The detailed algorithm for forces distribution is presented in [25].

The framework to deform the lungs consists of two software components: a *driver* and a *deformer*. The driver controls the volume of the lungs for a particular atmospheric, intra-alveoli, elastic recoil (i.e. associated with the lung tissue), and intra-pleural (i.e. pressure between the thoracic cage and the outer layer of lungs) pressure, while the deformer controls the shape of the lungs for a given volume and the local properties of the alveoli.

3.2 Application Domain Analysis

A well designed distributed interactive system accounts for the transmission delays of participant actions and provides real-time response. Participants make their judgments and react based on the situation presented to them by the human-computer interfaces. The graphical user interface (GUI) available for the instructor allows changes in the lungs dynamics simulation parameters such as breathing rate and tidal volume. The interactive and dynamic nature of the distributed environment is constrained by latency.

We perform a brief application domain analysis. The corresponding upshot frequency (ν_0), defined in Eq.1 allows us to estimate the capability of the distributed infrastructure where the application will be deployed [16]. In Eq.1, t_{xy} represents the average communication delay between two participants (i.e. distributed system nodes).

$$v_0 = \frac{1\ action}{t_{xy}} \qquad (1)$$

The average action frequency (ν_k) defined in equation (2) is dependent on the instructor's interaction on the virtual model. In this equation, *m* represents the number of virtual models affected by the interaction. In the proposed training tool there is one virtual model, the deformable lungs model (i.e. *m=1*). b_{jk} represents the number of actions (i.e. change in the simulation parameters) applied by participant *k* (i.e. the instructor) on the object *j* (i.e. the virtual lungs model) over a time interval *Δt*.

$$v_k = \frac{\sum_{j=1}^{m} b_{jk}}{m \cdot \Delta t} \qquad (2)$$

Based on the infrastructure communication latency, the distributed interactive application falls in the category where the average action frequency is less than the upshot frequency, ($\nu_k < \nu_0$) as discussed in the implementation Section 4.

3.3 The Deformation Distribution

To drive the deformation simulation remotely, we have investigated three alternatives. The first approach is a frequent state regeneration, in which we send the value of the lungs volume for each step of the deformation. In the second approach, we aim to reduce the number of packets sent between nodes, since the input parameters for the driver can be merged into a packet. The components of the packet are the input parameters for the lungs deformation driver, which relate the intra-pleural pressure and the lung volume during breathing. The input parameters are the boundary values of the pressure and volume, and a set of floating point values referred to as control constants. The boundary values for volume are the Functional Residual Capacity (FRC) (i.e. the volume of air that remains in the lungs after expiration) and Tidal Volume (TV) (i.e. the maximum volume of air that enters the lung upon inspiration). PR is the maximum pressure value. During one breadth, the pressure varies from a minimum constant value to PR. The control constants (CP_0...CP_N) are multipliers for the parametric basis function that models the neural control relating the increase in the lungs volume with the pressure. The total number of control constants (N) varies from 3 to 9. The format of the packet is given in Figure 3.

FRC	TV	PR	N	$CP_0 \ldots CP_N$

Figure 3. Components of a Control Packet Object

In the third approach, to compensate for the communication latency, we combine the distributed application with an adaptive delay measurement algorithm [15] that computes the delay between each pair of nodes that interact. To reduce the transmission delays and to improve consistency, the dimension of the data packets exchanged as well as the number of messages exchanged among participants is kept to a minimum using a dead-reckoning approach.

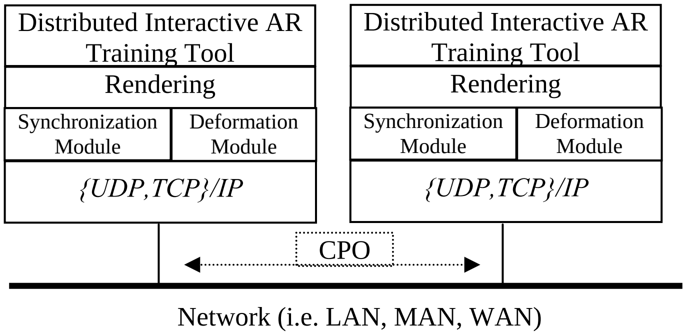

Figure 4. Software layers

Figure 4 describes the overall architecture of the distributed tool. The Control Package Object (CPO) encapsulates the data that has to be distributed among different participants.

The network delay compensation uses an adaptive approach to compute the delay between two participants. The latency measurement frequency increases when the delay varies abruptly following the network jitter pattern as illustrated in Figure 5. We adaptively trigger the round trip measurements for each node, based on the delay history, which better characterizes the network jitter and the application behavior. A fixed threshold is initially used at each participant node to build the delay history denoted H_p. The delay history is a sequence of *p* delay measurements h_i where *i=1,p*. In the implementation we have chosen *p* equal to 100, although higher values can be used based on the infrastructure setup.

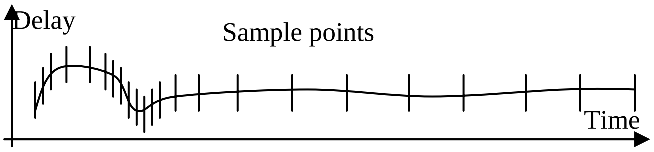

Figure 5. Adaptive delay measurements

4. PROTOTYPE IMPLEMENTATION

We have implemented a prototype distributed AR training tool using Linux based desktops with NVidia GeForce4 based GPUs, first in a two node setup. To superimpose the virtual model of the lungs on the human patient simulator we have used a least-squares pose estimation algorithm [1] and a predefined set of markers [13].

At the client side we have superimposed the deformable model on the HPS using a Polaris^{TM} infrared optical system from Northern Digital, Inc. The user's head pose is updated at a rate of 60 Hz. The update cycle is combined with the deformation rendering cycle to obtain an average display rate of 25 Frames per Second.

The deformable 3D lungs model shown in Figure 6 was implemented using OpenGL with Cg2.0 graphics hardware programming. For data distribution we have used the DARE (Distributed Artificial/Augmented Reality Environment) communication library (see *http://odalab.ucf.edu/dare*).

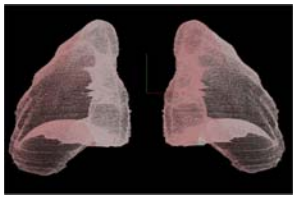

Figure 6. High resolution 3D deformable lungs model

Figure 7b shows a camera view of the superimposition of the deformable 3D lungs model on the HPS. The augmentation was filmed with a digital camera through the HMD.

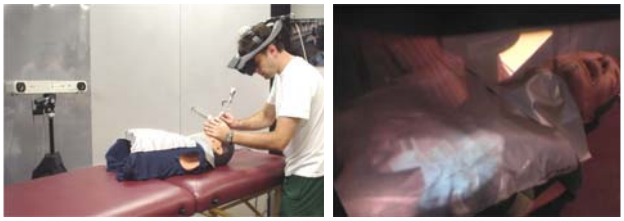

Figure 7. (a) Trainee and (b) trainee's augmented view filmed with a camera placed in place of one eye behind the HMD.

We performed three sets of experiments to compare the deformation distribution approaches. In the first set, we used the frequent state regeneration approach. The value of the lungs volume was frequently broadcasted to each participant and drove the deformation of the lungs at each site. A normal breathing cycle was divided into 300 steps and took around 9 seconds. The average action frequency induced by the server was ν_k = 300/9 = 33.33 (actions/second).

In the second set of experiments, we optimized the data distribution by reducing the number of packets, i.e. for each change in the simulation parameters, a CPO packet was sent that contains the simulation parameters (see Section 3.3). We refer to this method as *the CPO based updates* method. Once the packet was received, each node knew how to drive its own lungs deformation simulation up to the point the simulation parameters were changed. When the change occurred (triggered by the participants) another CPO was broadcasted to the other nodes. The action frequency was limited by the fact that the human-computer response time included perceptual, cognitive, and motor cycle times which can add up to an average of 240ms [8]. In other words the average action frequency induced by the participant was $\nu_k = 1/0.24 \approx 4$ (actions/second).

In the third approach we combined the CPO update method with the delay compensation algorithm to improve the consistency of the shared state, allowing all participants (local and remote) to see the same lungs deformation state at the same time. All the experiments were executed on a 100Mbps local area network having an average round trip time of 1ms. The upshot frequency in this case was ν_0 = 1/0.001 = 1000 (actions/second). We have simulated additional delays of 5, 10, 20, 30, 40 and 50 ms to study the behavior of the AR system under various delay conditions.

5. RESULTS, PRELIMINARY ASSESSMENT

To assess the consistency level of the deformation between a client node (i.e. trainee's AR view) and the server node (i.e. trainer's AR view), the normalized breathing volume of the deformable lungs and a time stamp were recorded at each node. Before the experiments, we synchronized the internal clocks on the distributed system nodes in within 0.8ms accuracy using the network time protocol (NTP). Such accurate clock synchronization is possible since the nodes are located in the same sub-network and they use the same time server for synchronization. Moreover we considered the small clock drifts between the nodes up to 0.1ms accuracy when we interpreted the results.

5.1 Volume Consistency, Breathing Cycle

We ran the prototype for fifteen minutes and recorded the normalized breathing volume for each breathing cycle. The frequent update method deployed on the 1ms network infrastructure delayed the deformation of the lungs at the client side with respect to the deformation seen at the server side for every breathing cycle as illustrated in Figure 8.

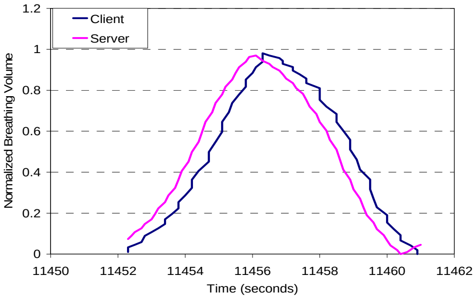

Figure 8. Frequent update

The CPO updates method reduced the number of packets flowing in the network by two orders of magnitude and produced a smoother rendering as compared to the frequent updates method. The simulation was improved because after the parameters were received, rendering proceeded independently at each site and was not affected by the network latency. Still without latency compensation, the deformation of the lungs at the client side fell behind the deformation seen at the server side at every breathing cycle, as illustrated in Figure 9.

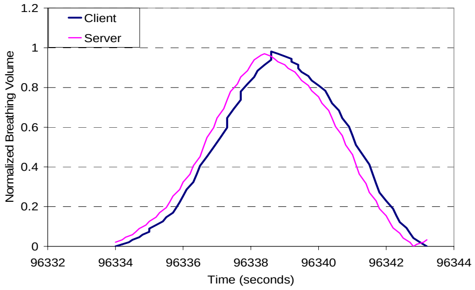

Figure 9. CPO updates no Synchronization

Finally, combining CPO updates with the delay compensation method, we obtained a smooth and synchronized view between participants, as shown by the normalized breathing volume in Figure 10. The normalized breathing volume reached the same values over time simultaneously at each node proving that high levels of shared state consistency can be achieved.

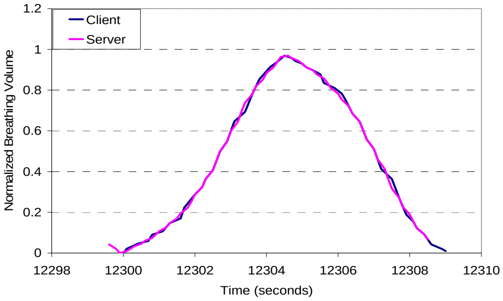

Figure 10. CPO updates with Synchronization

5.2 Latency Effects

To investigate the effect of latency, we have simulated an increase in delay from 1ms to 50ms and we computed the average difference (drift) in volume between the server and the client for one breathing cycle for each approach.

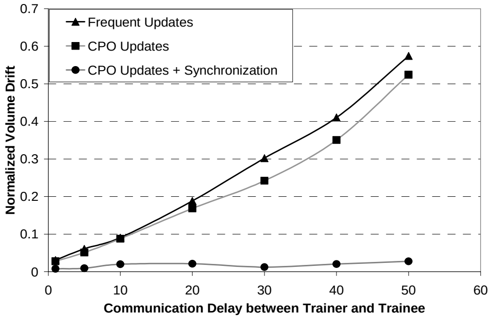

Figure 11. Normalized volume drift variation with delay

As expected, the results in Figure 11 show that the delay compensation algorithm has a positive effect on the consistency. The normalized breathing volume drift is maintained at low levels and remains fairly constant. Without delay compensation the normalized breathing volume drift increases almost linearly with the increase in the communication delay.

5.3 Preliminary Scalability Assessment

Figure 12 illustrates the values of the normalized breathing volume when three participants use the system. The third approach for data distribution was employed (i.e. CPO updates combined with synchronization).

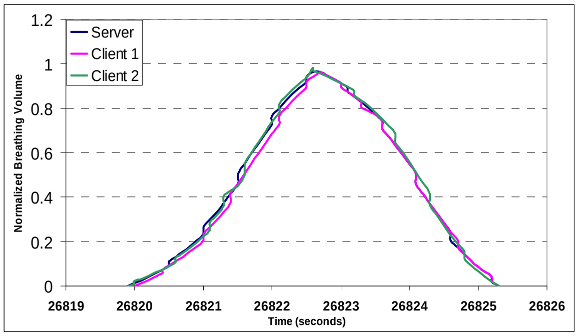

Figure 12. CPO updates with Synchronization, 3 nodes

The average normalized breathing volume drift per breathing cycle was in this case 0.015 as compared with 0.008 in the two nodes setup. Such variations are dependent not only of the network infrastructure but also on the hardware systems delay. The hardware attributes are described in Table 1.

Table 1. Hardware system attributes

Node no.	CPU (GHz)	RAM (MB)	GPU (GeForce)
1(server)	2.8 AMD	512	4 Ti4200
2(client)	1.5 AMD	1024	4 Ti4600
3(client)	1.7 AMD	1024	4 Ti4600

Results confirm that the drift was almost doubled when we added a third node. This increase may be attributed to the difference in hardware attributes. Further experiments will need to be conducted to establish whether more nodes will add drift in a similar way and how the heterogeneity of the system's nodes affects the simulation. Since each node independently renders the deformable 3D lungs model, we anticipate a relatively good scalability property.

6. CONCLUSIONS AND FUTURE WORK

We have presented the integration of a distributed interactive training prototype that uses deformable medical models combined with the capability for augmented reality. The distribution of deformable three-dimensional models at dispersed locations allows efficient communication of concepts and generates a vast potential for training.

The main assumption is that the distributed system is composed of homogeneous nodes i.e. each node has similar rendering and communication capabilities. The experiments were performed on a low latency network with up to three nodes, one server and two clients. We plan to increase the number of users to further test the scalability of the training tool and to assess the efficiency of the system from the human factors perspective.

7. ACKNOWLEDGEMENTS

We wish to thank our sponsors: the Link Foundation, the Office of Naval Research grant N00014-03-10677, The Florida Photonics Center of Excellence, and the US Army STRICOM for their support for this research.

8. REFERENCES

[1] Argotti, Y., Davis, L., Outters, V. and Rolland, J.P. Dynamic Superimposition Of Synthetic Objects On Rigid And Simple-Deformable Objects. *Computers and Graphics*, *26* (6). 919-930.

[2] Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C. and Van der Vorst, H. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia,PA, 1994.

[3] Billinghurst, M. and Kato, H. Collaborative augmented reality. *Communications of the ACM*, *45* (7). 64-70.

[4] Bro-nielson, M. and Cotin, S., Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. in *6th Eurographics workshop of Computer Animation and Simulation 96*, (1996), Springer-Verlag NY, 57-66.

[5] Caudell, T.P. and Mizell, D.W., Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. in *IEEE International Conference on Systems Sciences*, (Hawaii, 1992), 659-669.

[6] Cotin, S. and Delingette, H. Real-time Elastic Deformations of Soft Tissues for Surgery Simulation. *IEEE transactions on Visualization and Computer Graphics*, *5* (1). 62-71.

[7] Desbrun, M. Smoothed Particles for deformation modeling. *ACM Siggraph 1997*. 123-134.

[8] Eberts, R.E. and Eberts, C.G. Four Approaches to Human Computer Interaction. in Hancock, P.A. and Chignell, M.H. eds. *Intelligent interfaces: theory, research, and design*, North-Holland, 1989, 69-127.

[9] Faure, F., Interactive Solid Animation using Linearized displacement constraints. in *9th Eurographics Worshop on Computer Animation and Simulation*, (Lisbon, 1997).

[10] Fuchs, H., State, A., Livingston, M., Garrett, W., Hirota, G., Whitton, M. and E., P., Virtual Environments Technology to Aid Needle Biopsies of the Breast: An Example of Real-Time Data Fusion. in *Medicine Meets Virtual Reality*, (Amsterdam, NL, 1996).

[11] Greenhalgh, C., Purbrick, J. and Snowdon, D., Inside MASSIVE3: Flexible Support for Data Consistency and World Structuring. in *ACM Collaborative Virtual Environments (CVE)*, (2000).

[12] Hamza-Lup, F.G. A Distributed Augmented Reality System for Medical Training and Simulation. *Energy, Simulation-Training, Ocean Engineering and Instrumentation: Research Papers of the Link Foundation Fellows*, *4* (Rochester Press).

[13] Hamza-Lup, F.G., Davis, L., Hughes, C. and Rolland, J.P., Marker Mapping Techniques for Augmented Reality Visualisation. in *XVIIth Intl. Symposium on Computer and Information Sciences*, (Orlando, FL, 2002), CRC Press, 152-156.

[14] Hamza-Lup, F.G., Hughes, C.E. and Rolland, J.P., Distributed Consistency Maintenance Scheme for Interactive Mixed Reality Environments. in *International Conference on Cybernetics and Information Technologies, Systems and Applications*, (Orlando, FL, 2004), 7-12.

[15] Hamza-Lup, F.G. and Rolland, J.P., Adaptive Scene Synchronization for Virtual and Mixed Reality Environments. in *IEEE Virtual Reality*, (Chicago, MI, 2004), IEEE, 99-106.

[16] Hamza-Lup, F.G. and Rolland, J.P. Scene Synchronization for Real-Time Interaction in Distributed Mixed Reality and Virtual Reality Environments. *PRESENCE: Teleoperators and Virtual Environments*, *13* (3). 315-327.

[17] James, D.L. and Pai, G.K. Multiresolution Green's Function Methods for interactive simulation of elastostatic models. *ACM Transactions of Graphics*, *22* (1). 47-82.

[18] Macedonia, M., Zyda, M., Pratt, D., Donald, P., Brutzman, P. and Barham, P. Exploiting Reality with Multicast Groups. *IEEE Computer Graphics and Applications*, *15* (5).

[19] Martins, R. and Rolland, J.P., Diffraction properties of phase conjugate material. in *Proceedings of the SPIE Aerosense: Helmet- and Head-Mounted Displays VIII: Technologies and Applications*, (Orlando, FL, 2003), 277-283.

[20] Miller, D. and Thorpe, J.A. "SIMNET: The advent of simulator networking". *Proceedings of IEEE*, *83* (8). 1114-1123.

[21] Rolland, J.P., Davis, L. and Hamza-Lup, F., Development of a training tool for endotracheal intubation: Distributed Augmented Reality. in *Medicine Meets Virtual Reality (MMVR)*, (2003), 288-294.

[22] Santhanam, A., Fidopiastis, C. and Rolland, J.P., An adaptive driver and real-time deformation algorithm for visualization of high-density lung models. in *MMVR*, (Newport, CA, 2004), OS Press, 333-339.

[23] Santhanam, A., Fidopiastis, C. and Rolland, J.P., A Biomathematical Model for Pre-Operative Visualization of COPD and Associated Dyspnea. in *NIH Symposium on Biocomputation and Bioinformation*, (Washington D.C, 2003), I-31.

[24] Santhanam, A., Fidopiastis, C. and Rolland, J.P., PRASAD: An Augmented Reality based Non-intrusive Pre-operative Visualization Framework for Lungs. (poster). in *IEEE VR*, (Chicago,MI, 2004), IEEE Computer Society, 253-255.

[25] Santhanam, A., Pattanaik, S., Fidopiastis, C., Rolland, J.P. and Davenport, P., Physically-Based Deformation of High-Resolution 3D Lung Models for Augmented Reality Based Medical Visualization. in *MICCAI -AMI-ARCS International Workshop on Augmented environments for Medical Imaging and Computer-aided Surgery 2004.*, (Rennes, St.Malo France, 2004), Springer Verlag.

[26] Schmalstieg, D. and Hesina, G., Distributed Applications for Collaborative Augmented Reality. in *IEEE Virtual Reality*, (Orlando, FL, 2002), 59-66.

[27] Sharkey, P.M., Ryan, M.D. and Roberts, D.J., A local perception filter for distributed virtual environments. in *IEEE Virtual Reality*, (Atlanta, GA, 1998), 242-249.

[28] State, A., Livingston, M.A., Hirota, G., Garrett, W.F., Whitton, M.C., Fuchs, H. and Pisano, E.D., Technologies for augmented-reality systems: realizing ultrasound-guided needle biopsies. in *SIGGRAPH*, (1996), 439-446.

[29] Terzopolous, D. and Fleischer, K. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *Siggraph: ACM Special Interest Group on Computer Graphics and Interactive Techniques*, *1*. 269-278.

[30] Thomas, B.H., Quirchmayr, G. and Piekarski, W. Through-Walls Communication for Medical Emergency Services. *International Journal of Human-Computer Interaction*, *16* (3). 477-496.

[31] Webster, A., Feiner, S., MacIntyre, B., Massie, W. and Krueger, T., Augmented Reality in Architectural Construction, Inspection and Renovation. in *ASCE Third Congress on Computing in Civil Engineering*, (Anaheim, CA, 1996), 913-919.

[32] White, R.E. *An Introduction to Finite Element Methods*. John Wiley and Sons, Raleigh,NC, 1991.

[33] Wolfgang, F. *Augmented Reality for Development, Production and Servicing - Augmented Reality in Entwicklung, Produktion und Service*, 2003.

[34] Zhuang, C. Real-Time Non-linear Deformation. *IEEE Transactions on Visualization*.

[35] Zyda, M., Pratt, D.R., Monahan, J.G. and Wilson, K.P., NPSNET: Constructing a 3D Virtual World. in *ACM Symposium on Interactive 3D Graphics*, (Cambridge, Massachusetts, 1992), 147-156.