3D Haptic Needle Insertion Simulator Utilising An Agent Based Spherical Voxel Model

James Mullins, Hieu Trinh and Saeid Nahavandi

Intelligent Systems Research Laboratory,
Deakin University Geelong, Victoria Australia 3217
james.mullins@deakin.edu.au

Abstract This paper describes a technique for the real-time modeling of deformable tissue. Specifically geared towards needle insertion simulation, the low computational requirements of the model enable highly accurate haptic feedback to a user without introducing noticeable time delay or buzzing generally associated with haptic surgery simulation. Using a spherical voxel array combined with aspects of computational geometry and agent communication and interaction principals, the model is capable of providing haptic update rates of over 1000Hz with real-time visual feedback. Iterating through over 1000 voxels per millisecond to determine collision and haptic response while making use of Vieta's Theorem for extraneous force culling.

1. Introduction

As the performance of haptic force reflecting hardware increases and the cost of ownership decreases, the medical field is turning towards haptic technology as a surgical simulation and training aid. Concurrently, internal organ and tissue models are becoming increasingly complex. This places unprecedented strain on computational algorithms and the hardware on which they run. As no two individuals have identical internal organs and tissue, some ambiguity can be designed into the models and as a result, some model accuracy can be traded for haptic smoothness and computational speed. We are looking at the presentation of haptic data sets recorded from a robotic needle insertion operation to validate our model. Utilising the basic fundamentals of haptic research combined with several aspects of computational geometry and agent based research has enabled us to increase the resolution and effectiveness of our haptic needle insertion process and has ultimate benefits for a variety of haptic applications.

It should be noted that using computational geometry derivatives for developing models of haptic scene graphs of needle insertion is not new. Kataoka[1] and DiMaio[2] both describe the use of custom haptic hardware for plotting and resolving force vectors in needle-tissue interaction. These methods while effective, rely on custom hardware that is only suited for a small degree-of-freedom needle simulation. Gerovich[3] makes use of a 6 axis Phantom□□ haptic device from SensAble Technologies. This device can be used for multiple applications in the medical simulation field and is better value for money as a research and training aid. Gerovich's simulations however are limited to algorithms developed using a trial and error approach. Force vectors are set by an anesthesiologist or doctor and compared against real procedures. The model is fine-tuned over time and as a result is a slow and potentially inaccurate way of development.
There have been several studies on the collection of forces for developing models from needle insertions, Simone[4]. These studies, while thorough in their approach, are limited to data collection and analysis. Building upon this research, we have developed a model generation algorithm for creating models of soft tissue and internal organs in real time utilising a spherical voxel model based upon an agent based design philosophy. Utilising agents in soft tissue modeling is new to the literature and is our ultimate contribution to tissue simulation knowledge. To provide accurate force feedback once the needle has penetrated the surface of the tissue, we are using a spherical voxel collision detection algorithm. This system reduces the complexity of finite element analysis to a more computationally efficient level suitable for real time haptic feedback.

Development of an accurate three-dimensional model of a piece of human anatomy for needle insertion requires five steps:

1. Collection of generic force data from the area of interest.
2. Development of a 3D physical model to be imported into the haptic scene.
3. Effective detection of collisions between the model and the haptic probe.
4. Determination of correct response to forces applied once collision detection occurs.
5. Rendering of the haptic scene to screen for visual perception.

A human can feel kinesthetic changes below 1000Hz, it is important that a computational model run at least that rate[10]. Visually updating the world can occur at much slower rates of around 20Hz – 40Hz[11].

We are utilising two machines, a Dual 2.0GHz PowerMac G5 (64Bit) as a graphical processor and a dual processor 1.66GHz Core Duo as a link to the Phantom haptic device. These machines talk to one another by User Data-gram Protocol (UDP) sockets, as data transfer at such high data rates with Transmission Control Protocol (TCP) is inadequate. It was found that UDP maintains an adequate level of data integrity.

2. Haptic Recording

Haptic recording is the process by which force and torque data is recorded against a set of known events. It is important to know the physical properties of the tissue in which a model will be constructed. Viscosity, hydration and elasticity all play a part in the kinesthetic representation of haptic surgical simulation.
Forces required to perform a necessary interaction be it cutting, suturing or injecting are discovered and recorded across a broad area of a particular organ. In a structured environment, a robot can be used for precise measurement and positioning of the surgical instrument for force measurement. In the case of in-vivo measurements, surgical robots are still in their infancy. As a result in-vivo haptic surgical instruments are being produced. These instruments are fitted with a magnetic Polaris based tracking system to enable similar resolutions and ranges as those based on robotics arms and are much easier for a surgeon to adapt to in the short term.

For our simulation the process of haptic data is achieved using an Epson Pro-6 industrial robot coupled with an ATI 6-axis force torque sensor and a custom needle syringe end effector. This hardware enables the precise positioning of the needle to the desired test medium with high repeatability.

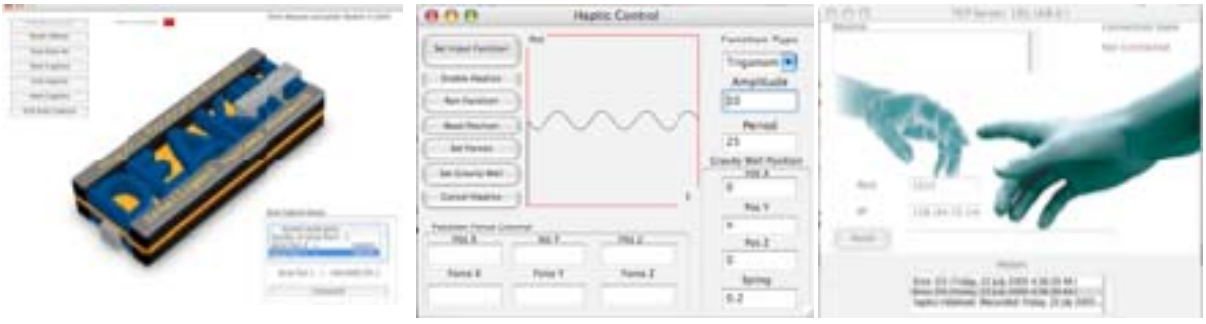


Figure 1: (A,B,C). Developed software for capturing haptic forces at high speed (~ 1200 hertz). This software also enables us to test haptic device latency and differing control schemes with instant graphical output of their effectiveness.

While we are using a robot to capture force reading via force sensor attached to the wrist of the robot, it should be noted that the technique could easily be applied to a hand controlled in-vivo surgical instrument. Because the tissue interactions are limited, algorithms have to be run to extrapolate these measurements to encompass the whole organ or tissue area of interest. Generally a surgeon would not have the capability to set up a structured grid of coordinates on an internal organ in order to gain haptic measurements in a precise well-defined manner. As a result, a form of interpolation is generally required.

Figure 2: Using an industrial robot attached to a six-axis force sensor to measure the haptic properties of neoprene. Neoprene was chosen for initial testing as it was found to have similar haptic properties to bovine liver. Differing gauges of needle were tested with a diverse array of tip geometries. These results have been stored in a material library for future reference and input into our needle model.

2. Agent Based Spherical Voxel Model

As a novel way of dealing with the problem of real-time computational efficiency, our spherical voxel model utilises over one thousand spherical nodes or voxels linked via a mass/spring damper system. These spherical voxels or nodes contain their own intelligence based on an intelligent agent schema.
The particles are constrained locally and can interact using a collision detection algorithm. This collision detection and resultant force reflection within the tissue allows the whole three-dimensional model to apply force back to a haptic user.

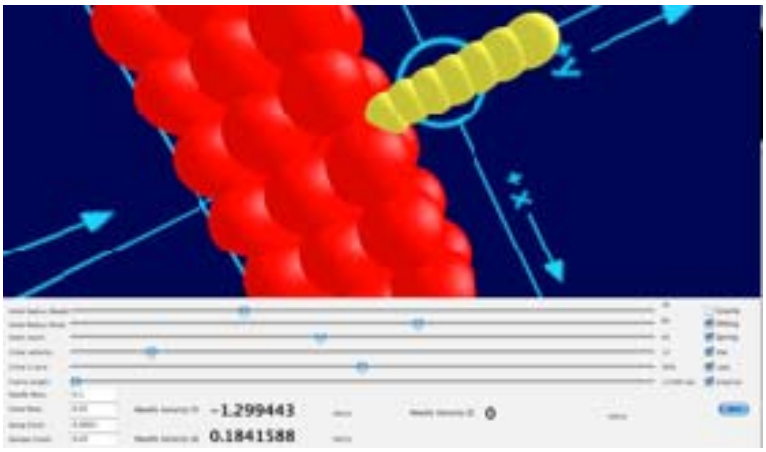


Figure 3. Spherical Voxel model as needle (yellow) penetrates modeled tissue surface.

Each element (voxel) within the modeled system has a mass, initial velocity, current velocity, elasticity (percentage deformation), radius, initial position, current position and human readable identification. This information is used to determine resultant force and voxel position once utilised by our model. External forces added to the voxels are gravity, spring and damper as well as the component forces of the haptic needle interacting with the tissue model. The spherical voxel model differs from other point force models in that it relies on the spherical surface of the voxels for force reflection.

3 Know Your Neighbour

Developed for this application, the Know Your Neigbour (KYN) algorithm is implemented by each voxel as part of the intelligent agent schema. Each voxel within the simulation keeps track of its surrounding voxels within its own memory map. The KYN thread can keep track of static and dynamic voxel movement.
Collision detection at its simplest level consists of an iterative loop of equation 1 comparing each voxel against its closest neighbours KYN. Where p_1 and p_2 are Voxel midpoints and r_1 and r_2 are voxel radii. This loop runs at approximately 2000Hz with 1000 voxels in a self contained thread.

$$|p_2 - p_1| = r_1 + r_2 \qquad (1)$$

Now that we can determine if a collision has occurred, the information is handed to the haptic frame buffer. The haptic frame buffer, which is locked at 1000Hz, derives each voxels relative position and velocity with respect to time using equation 2.

$$|p_2(x) - p_1(x) + (v_2 - v_1)t| = r_1 + r_2 \qquad (2)$$

The next important step is to determine exactly when a collision between voxels will occur. A standard quadratic module handles this function. Where $a = V^2$, $b = 2(pv)$ and $c = (p^2-r^2)$.

$$t = \frac{-2(pv) \pm \sqrt{2(pv)^2 - 4(v^2)(p^2 - r^2)}}{2v^2} \qquad (3)$$

This calculation is simplified even further by using Vieta's Theorem, which tells us that positive roots of equation 3 (we are only interested in positive roots as negative roots would have occurred in past timeframes) can be tested for by equation 4.

$$t_1 + t_2 = -2pv/v^2$$
$$t_1 \times t_2 = \frac{(p^2 - r^2)}{v^2} \qquad (4)$$

Where t_1 and t_2 are both roots of equation 3. If $t_1 = t_2$ then the voxels involved have had very minimal contact and this can be negated. If t_1 x t_2 is positive but $t_1 + t_2$ is negative, then both roots must be negative and there won't be a collision in this haptic frame. If t_1 x t_2 is negative, then either t_1 or t_2 is negative. Since we know that $t_1 <= t_2$, t_1 must be negative; again, no collision in this frame. The remaining case is that both t_1 x t_2 and $t_1 + t_2$ are positive. If both conditions are true a collision will occur within the haptic frame. Performing this simple test allows us to determine whether or not further expensive computation is required.
Now that we have determined a collision will occur, its response is relatively simple. Velocities for both interacting voxels v_1 and v_2 are calculated based on voxel mass and position as shown in equation 5.
As mentioned previously, all of these calculations are occurring at 1000Hz. Now that collision detection and resultant forces have been determined, it is time to add the mass spring/damper system to the model.

$$v_{1\,final} = v_{1initial} - k \times m_2 \times N$$
$$v_{2\,final} = v_{2initial} + k \times m_1 \times N$$
$$where:$$
$$N = normal(p2 - p1)$$
$$k = \frac{c(e+1)}{(m_1 + m_2)}$$
$$c = normal(v_1 \bullet v_2)$$
$$e = \text{elastic coefficient range}: (0\text{-}1) \qquad (5)$$

Because the voxels themselves are normally at rest, the mass spring system need only be applied to bodies that have experienced displacement within a preset time frame. The model at this stage utilises a standard parallel spring damper system as show in figure 4.

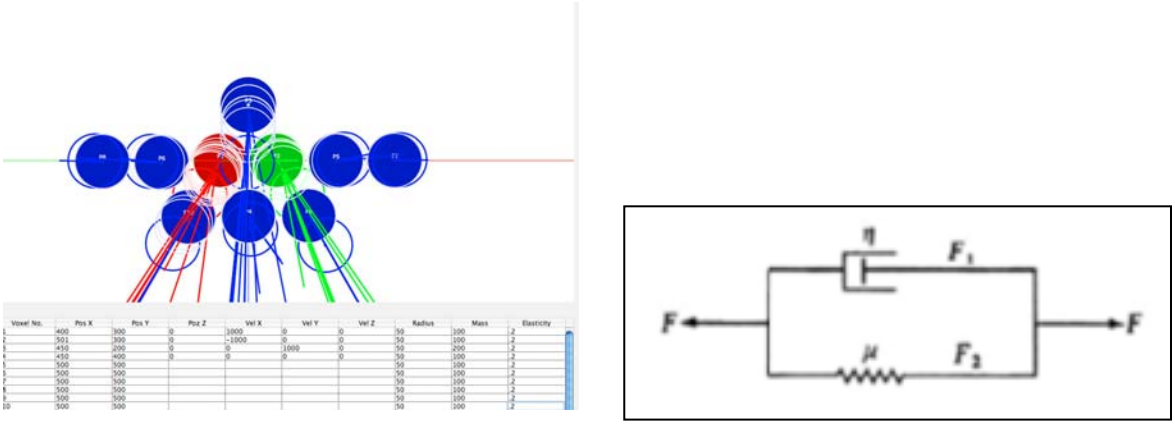


Figure 4a. Early 2D representation of agent based physics engine. 4b. Parallel spring damper system used in spherical voxel model

Standard equations (8,9) are used for spring and damper force calculation respectively.

$$\mathbf{f}_{spring} = -k_s\mathbf{x} \qquad (6)$$

$$\mathbf{f}_{damp} = -k_d\mathbf{v} \qquad (7)$$

As the damper requires instantaneous closing velocity, equation 8 is used. In Equation 6, x is used to denote displacement from the voxels initial resting position p_{init}.

$$\mathbf{e} = \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|}$$

$$v = \mathbf{v}_1 \cdot \mathbf{e} - \mathbf{v}_2 \cdot \mathbf{e} \qquad (8)$$

Values for k_s and k_d as well as voxel mass and radius are determined experimentally. Needle mass and velocity is a known quantity and is derived from the Phantom haptic device. Due to the nature of the collision detection algorithm, voxel radius has a significant impact on the forces acting against the virtual needle, most notably, the frictional forces acting upon the body of the needle shown in figure 5. Generally the equation for needle insertion can be described as in equation 9 (Simone et al). Cutting force is impacted by voxel mass, friction by radius and stiffness by a combination of the two combined with the voxel elasticity.

$$f_{needle}(x) = f_{stiffness}(x) + f_{friction}(x) + f_{cutting}(x) \qquad (9)$$

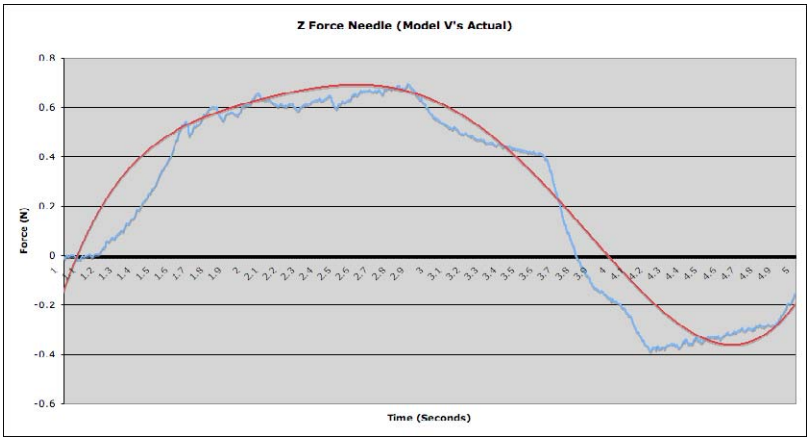
It was discovered early on that soft tissue is non-linear in that voids and granularity is represented in all

haptic recording data. This accounts for error in the modeled output. It should be noted however, that this granularity can be accounted for in the voxels themselves. Each agent based voxel has a damage property. When a virtual needle interacts with a tissue voxel.
The intelligent agent linked to that voxel determines the amount of contact and the likely damage that will occur. As a result, it our 3D simulations, the voxel radius is reduced to simulate damage to the tissue upon needle contact. This damage is reversible give a change in time (Δt).

3. Haptic Rendering

The last step from a kinesthetic point of view is the physical haptic rendering. Our algorithm interfaces directly with an OCX developed based upon Sensable's latest HDAPI and HLAPI's. Reporting to a standard haptic servo loop of approximately 1000Hz, the algorithm determines the position of the probe at a reduced rate of around 400Hz. Once the probes position is known, it is tested against the closest spherical voxel utilising the above mentioned collision detection techniques. The actual haptic hardware chosen for force reflection is the Phantom Omni from SensAble due to its relatively low cost.

4. Results



It is hard to represent the accuracy of a tactile insertion, needle the force model however, showing the graphs calculated forces with

model an actual needle respect to have shown to insertion extremely be similar.

Figure 5. Modeled haptic force plotted against a haptically recorded force from a needle insertion into neoprene rubber.

Because the haptic properties of the model rely on fast update rates, it is important to know when the algorithm is saturated. Saturation occurs when the spherical voxel count exceeds the computational capabilities of the host machine. Therefore, a selection of different processors has been tested as a basis for comparison. Note that the spherical voxel model will fail before the graphical representation meaning only the spherical voxel model failure rates will be shown.

Number of Spheres	Haptic Frequency (Hz) A	Haptic Frequency (Hz) B
250	833.3	3050.4
500	714.3	1993.4
750	666.7	1439.8
1000	66.7	1062.4

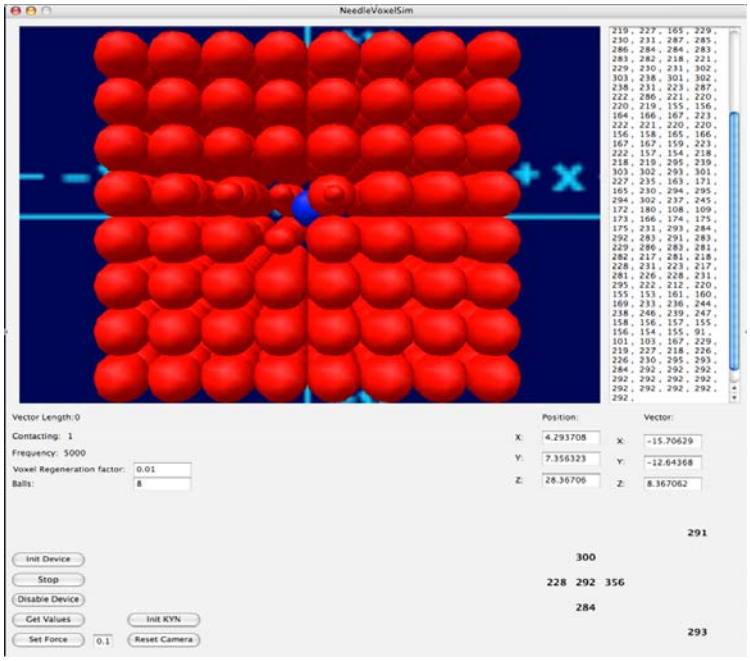
Table 1: Haptic servo loop speed versus the number of spheres on two different computers. (A) is a 1.4Ghz Pentium 3, (B) contains a dual 1.66Ghz Core Duo.

Table 1 shows the corresponding decrease in the speed of calculation for the haptic servo loop based on the number of calculated voxels. To achieve realistic results from the algorithm, a high speed computer is required (B). It shows that to maintain smooth and buzz free operation of the haptic device, a Core Duo at 1.66Ghz can render approximately 1000 spheres. The slower computer (A) can only successfully render around 150 spheres.

We chose neoprene foam rubber as a surface to inject to test our model because of its apparent similarity to that of human skin. Figure 5 (lighter - blue) shows a typical robotic needle insertion perpendicular to the surface. Figure 5 (darker - red) also shows a typical force plot of a needle insertion into our spherical voxel model. While still undergoing development, the model shows the general profile of a needle insertion operation. Being a 3D capable model, all three axis forces are resolved and presented to a user haptically.

5. Conclusions

This paper presents an overview of the algorithm we are currently testing and enhancing for the measurement and playback of haptic data sets to aid in medical needle insertion simulation and training. Utilising aspects of computational geometry, intelligent agents and haptic force reflection, allows the model to be developed in real time. As a surgeon or surgical robot adds data points, a graphic model continually evolves. Force data captured from the needle probing is iterated through our spherical voxel model to determine a correct force scale. Adding more data points enhances the resolution and accuracy of the model. Haptic scene servo loops are maintained up to around 1000 spheres using currently available high-end consumer workstations. It should be noted that due to the nature of the agent based voxel programming, the simulation supports distributed computing. While a current machine may only be able to model a particular organ, multiple machines collaborating are theoretically able to model entire structures.



The data collection and modeling procedure has been tested on varied rubber composites of differing shapes. This has allowed

the testing of a visual model creation algorithm [12], [13] as well as the spherical voxel haptic model generation capability.
By using readily available haptic devices and off the shelf components, we are hoping that this technology will be adopted into the medical community in the near future.

Figure 6. Window showing the NeedleVoxelSim program and a 5 x 5 x 5mm rendered tissue sample.
The needle tip is shown inserted and voxel damage has been depicted by a reduction in the damaged voxel radii.

6. Future Work

As our algorithm continually evolves, we hope to add more features to realistically simulate the physical organ or tissue. Our spherical voxel model will be evolving to take advantage of faster technology. An auto configuration script capable of setting core values such as voxel radius, mass and elasticity will simplify new tissue integration.
Our first step will be to develop a three-dimensional visual engine capable of rendering the organ or tissue under haptic evaluation. Adding a sense of depth perception to the model, which will greatly enhance users' understanding of the scene.
As computational hardware continues to evolve, we plan to combine the visual processing and haptic rendering on the one machine.

7. References

[1] H. Kataoka, T. Washio, K. Chinzei, K. Mizuhara, C. Simone, and A. Okamura, "Measurement of Tip and Friction Force Acting on a Needle During Penetration," Proceedings of the Fifth International Conference on Medical Image Computing and Computer Assisted Intervention -- MICCAI 2002, Lecture Notes in Computer Science (Vol. 2488), T. Dohi and R. Kikinis, Eds., 2002, pp. 216-223.

[2] S. P. DiMaio and S. E. Salcudean, "Needle Insertion Modelling for the Interactive Simulation of Percutaneous Procedures," Proceedings of Medical Image Computing and Computer-Assisted Intervention, Tokyo, September 2002

[3] O. Gerovich, P. Marayong, and A. M. Okamura, "The Effect of Visual and Haptic Feedback on Computer-Assisted Needle Insertion," Computer-Aided Surgery, Vol. 9, No. 3, 2004

[4] C. Simone and A. M. Okamura, "Haptic Modeling of Needle Insertion for Robot-Assisted Percutaneous Therapy," Proceedings of the IEEE International Conference on Robotics and Automation, 2002, pp. 2085-209

[5] Delaunay Refinement Mesh Generation, Ph.D. thesis, Technical Report CMU-CS-97-137, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, 18 May 1997

[6] Delaunay Refinement Algorithms for Triangular Mesh Generation, Computational Geometry: Theory and Applications 22(1-3):21-74, May 2002

[7] Fortune, S., "A Sweepline Algorithm for Voronoi Diagrams," Algorithmica, 2:153-174, 1987.

[8] Guibas, L. and Stolfi, J., "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams," ACT TOG, 4(2), April, 1985.

[9] Malcolm Sambridge, Jean Braun and Herbert McQueen
Geophysical Journal International v122, pp837-857, 1995.

[10] GHOST SDK Reference Manual GHOST ® SDK API Reference Version
4.0 SensAble Technologies, Inc. 2003.

[11] Parkhurst, D., Culurciello, E., & Niebur, E. (2000). Evaluating variable resolution displays with visual search: Task performance and eye movements. Proceedings of the ACM Eye Tracking Research and Applications Symposium, 1, 105-109

[12] Mullins, J., Trinh, H. and Nahavandi, S. (2004) A Real Time Haptic Model Generation Technique for Surgical Needle Insertion Simulation, *Mechatronics and Robotics '04*, pp. 350-354, Institute of Electrical and Electronics Engineers, United States

[13] Mullins, J., Trinh, H. and Nahavandi, S. (2004) Soft Tissue Modelling for Haptic Rendering in Virtual Environments, *Mechatronics 2004.Proceedings of the 9th Mechatronics Forum International Conference*, pp. 693-698, Atilim University Publications, Turkey