Proxy-based Haptic Feedback from Volumetric Density Data

Karljohan Lundin[*] Anders Ynnerman[†] Björn Gudmundsson[‡]
Department of Science and Technology
Linköping University, Sweden

Abstract

In this paper a new approach to volume haptics is presented. The developed method makes use of a proxy that is constrained by ‘virtual surfaces’, defined by the local gradient at the proxy position, and not by iso-values as in other approaches to volume haptics. By using a proxy, material properties like friction, stiffness and surface penetrability can be implemented. These material properties are controlled by user defined transfer functions. At the same time, using the gradient to define surfaces, rendering of infinitesimally close virtual surfaces that can be penetrated, is made possible.

The algorithm exhibits very high stability, is fast and represents fine details accurately. Compared to earlier techniques, less artifacts occur and higher configurability is provided.

1 Introduction

The ever-increasing speed of computers has now made it possible to perform real-time visualisation of volume data. A prime example of this is medical visualisation. Overwhelming amounts of data, in the form of arrays of slices, can be produced by modern medical equipment, such as computer tomography (CT) and magnetic resonance imaging (MRI). Using volume rendering, the physicians no longer have to look at the arrays of 2D images, but can be presented with a full 3D representation of scanned objects.

In medical visualisation, the enhanced feeling of presence and comprehension that haptics provide, can facilitate faster task execution[12, 8] and more accurate data analysis[6]. While the visual rendering gives global feedback from the volume data, the haptic feedback gives extended information of local neighbourhoods. Future medical visualisation systems will be able to allow the users to touch, hold, rotate and move authentic CT data, in the most intuitive way possible — the way in which they interact with the real word.

Present haptic methods for volumetric data do, however, not provide the natural haptic feedback needed in medical applications. They follow a general approach, which fits better for fluid content than for the solid content of medical CT data. For this more sophisticated techniques are needed, which can present material properties like friction, stiffness and viscosity. The algorithm presented here mimics surface haptics, while still acting directly on the volume data.

In section 2 related work is discussed, and the here presented algorithm is put in the context of other methods. In section 3 the algorithm is presented — how it implements surface feedback in section 3.1, how viscosity feedback is integrated in section 3.2 and how to give material specific feedback in section 3.3. In section 4 an implementation of the algorithm is presented. Results from the implementation are presented in section 5 and in section 6 we outline future developement of the algorithm.

2 Related Work

Some of the functionality of the proxy-based volume haptics presented here mimics proxy-based surface haptics techniques. Thus related work can be found in both the area of surface haptics and previous volume haptics applications.

2.1 Surface Haptics

The algorithms for surface haptics mostly act on an explicit surface representation, for example polygons. The tip of the haptic instrument is hindered from penetrating into the object by providing the user with haptic feedback, that pushes the instrument out of any virtual object.

The most recent in the evolution of surface haptics is the proxy-based method described in [9]. While earlier methods were forced to use explicit topology representations or suffered from force discontiueties, the proxy-based method can give smooth shaded feedback from unprocessed polygons. This allows for a fast and dynamic environment as

---

[*] karlu@itn.liu.se
[†] andyn@itn.liu.se
[‡] bjogu@itn.liu.se

well as low storage costs, since no pre-calculation is necessary. Since the motion of the proxy over the surface is controlled, the feeling of friction can also easily be implemented. By lagging the proxy relative to the haptic instrument, a force that pulls the instrument tip back is induced, which gives the impression of friction.

2.2 Volume Haptics

For volume data, the standard proxy-based method cannot be used, since the volumetric data has no surface representations. A common way to introduce haptics to volume data is therefore to extract an intermediate local or global surface[7, 3] from which proxy-based surface haptics can be calculated. This can be compared with the use of indirect volume rendering, which is common in volume visualisation applications for low-end computers. There are variations of this method, for example the extraction of inner and outer voxels of specific objects in the volume[2]. However, by defining a surface to calculate haptic feedback from, every data not a part of that surface is unrepresented in the haptic rendering. Furthermore haptic occlusion of potentially important areas is introduced by the definition of distinct impenetrable surfaces.

To achieve full haptic examination of volume data 'direct volume haptics' is required. Present systems for direct volume haptics, however, follow a very general approach[6, 4, 1, 5], where the haptic feedback is defined as a vector valued function, or can be translated to one. This function produces a force feedback solely from the data around the instrument tip and the velocity of the tip. A force based on the local gradient is common in this kind of haptics. The gradient vector depicts the orientation as well as the magnitude of changes in the scalar data, so by applying a negative multiple of it, a feeling of implicit surfaces can be achieved. One example of this is the following force function:

$$\vec{f} = -C_1\,\vec{\nabla}V(\vec{x}_{tip}) - C_2\,\vec{v}_{tip} \tag{1}$$

Disregarding the viscosity term, induced from the velocity input of the function, this can be considered a static force field. This general approach to volume haptics works well with fluid content, for example results from Computational Fluid Dynamics. Unfortunately when applying the general approach to other types of content, for example solid data, the haptic feedback lacks a natural connection to the data. The static force field gives haptic force response to volume data and not to user activity. Thus the haptic algorithm is adding energy to the system, instead of absorbing the energy added by the user. This gives a natural feeling with fluid content, since the added energy can be accepted as pressure or flow. With solid content, however, active haptic feedback makes little or no sense.

3 Proxy-Based Volume Haptics

The motivation for the development of this new volume haptics algorithm was to create a system that produces natural haptic feedback from density data with solid content, e.g. CT data. The haptic feedback should give a feeling of the object that becomes mentally coupled with the visual representation of the contact.

To bring about the feeling of soft and hard tissue one has to make the system push the instrument back the same way that, for example, skin pushes any instrument back — only to the surface, the origin of the push. For the algorithm to "remember" this position a proxy is needed, like in the approach common in surface haptics. In surfaces haptics the proxy is a spherical object, but in volume haptics the proxy is only a position, describing a 'centre of motion' — a position from which the real haptic instrument is displaced.

As in surface haptics the displacement of the proxy relative to the haptic instrument is used to calculate a spring simulating force, by using the spring equation $\left(\vec{f} = -k\,\vec{d}\right)$. This spring force is used as force feedback from the system. The same force is also considered to act on the proxy. The proxy is then moved by the system, a small distance every time-step, to simulate its reactions to the force.

The movement of the proxy during one time-step give rise to the haptic behaviour of the algorithm. It must implement surface feedback, friction and, to avoid haptic occlusion, penetrability of the surfaces.

To simplify the task of finding the surfaces, the assumption is made that the proxy is always moved a small distance in every time-step and is thus not moved past a surface that should be impenetrable. This is not always true, but tests have shown that the distance the proxy is moved in most situations is small enough to make an adequate simulation of surfaces and viscosity.

3.1 Surface Feedback

Primarily the movements of the proxy must be constrained by the virtual surfaces. These surfaces are defined by the local gradient as being the normal vector to a surface in every possible point of the volume. If the proxy is constrained to move only perpendicular to the gradient vector at the proxy position $\left(\vec{\nabla}V(\vec{x}_p)\right)$, the proxy is bound to follow the surface on which it lays.

To achieve this, the force vector from the spring simulation is split into a normal directed force component and a component perpendicular to the gradient vector at the proxy position:

$$\vec{d} \;=\; \vec{x}_{tip} - \vec{x}_p \tag{2}$$

$$\hat{N} = \frac{\vec{\nabla} V(\vec{x}_p)}{\left|\vec{\nabla} V(\vec{x}_p)\right|} \tag{3}$$

$$\hat{T} = \frac{\vec{d} - \hat{N}\left(\vec{d} \cdot \hat{N}\right)}{\left|\vec{d} - \hat{N}\left(\vec{d} \cdot \hat{N}\right)\right|} \tag{4}$$

$$f_N = \vec{f} \cdot \hat{N} \tag{5}$$

$$f_T = \vec{f} \cdot \hat{T} \tag{6}$$

This is also shown in figure 1. The perpendicular force component then lies in the plane defined by the gradient vector at the proxy position. It is thus tangential to the surface at that point. If disregarding the normal directed force component, the proxy can only be moved to a position on this plane. It is however still pulled towards the instrument tip, but only inside the plane:

$$\vec{x}_p' = \vec{x}_p + C_1 \hat{T} f_T \tag{7}$$

Since a new plane, to which the proxy is constrained, is used in every time step, the proxy can follow rounded surfaces, as shown in figure 2. If the distance, which the proxy is moved in every time step, is small enough the resulting surface from the integrated positions becomes smooth.

Unless the local gradient is undefined, this works well. However, when the gradient is undefined or very weak, the surface becomes undefined or very noise sensitive respectively. By using an arbitrary normal vector instead of the gradient vector, to replace an undefined gradient, the algorithm can work anyhow. This results in surfaces that have arbitrary orientation that are removed by applying full surface penetrability, which is discussed later on.

Since the motion of the proxy over the surface is fully controlled, implementing friction is straightforward. The normal directed force component is used to calculate a tangent directed friction force, using the friction formula $(f = \mu f_N)$. This friction force is then used as threshold for the motion over the surface, i.e. the tangential motion. If the magnitude of the tangent directed force component is less than this threshold, the proxy is not moved. If the force exceeds the threshold, the proxy is moved just as much as is needed to balance the threshold with the force induced by the new distance:

$$\vec{d} = \vec{x}_{tip} - \vec{x}_p \tag{8}$$

$$T_T = \mu f_N \tag{9}$$

$$\vec{x}_p' = \begin{cases} \vec{x}_p + \hat{T}\left(\vec{d} \cdot \hat{T} - T_T/k\right), & \text{if } T_T < k\left(\vec{d} \cdot \hat{T}\right) \\ \vec{x}_p, & \text{otherwise} \end{cases} \tag{10}$$

That way the proxy follows a moving instrument tip at an even distance, which induces a smooth retarding friction force. Equation 10 now replaces equation 7.

To produce a feeling of actual surfaces in the volume, the proxy is only constrained from moving inward through surfaces, i.e. towards higher density values. Thus the proxy can move out of high density areas towards low density areas without surface constraints. But when trying to push the instrument towards high density areas the proxy is stopped by the surface.

The motion through a surface is controlled in the same manner as the friction simulating motion of the proxy, in a surface. A threshold (T_N) describes the penetrability of the surface. When the normal directed force component exceeds this threshold, the proxy is moved as much as is needed to balance the threshold with the normal directed force component induced by the new distance:

$$\vec{d} = \vec{x}_{tip} - \vec{x}_p \tag{11}$$

$$\vec{x}_p' = \begin{cases} \vec{x}_p + \hat{N}\left(\vec{d} \cdot \hat{N} - T_N/k\right), & \text{if } T_N < k\left(\vec{d} \cdot \hat{N}\right) \\ \vec{x}_p, & \text{otherwise} \end{cases} \tag{12}$$

Thus, if moving the proxy outwards, a threshold value zero is used, to fulfil the rule mentioned in the previous paragraph.

Moreover, if the surface is not well defined, the surface should not give full force feedback. How well defined a surface is, is given by the magnitude of the local gradient. With a low magnitude of the gradient, the threshold is therefore lowered. In uni-valued areas, where the gradient is zero, no surface feedback should be presented. This calls for a new parameter, i.e. a transfer function $W_d(\left|\vec{\nabla} V(\vec{x}_p)\right|)$, that defines the distinctness of the surface from the local gradient magnitude. It should range between zero and one corresponding to a zero and a full gradient magnitude respectively. The parameter should however be one as soon as the surface can be considered to be well defined, i.e. at a gradient magnitude of a little more than zero. The surface threshold is multiplied with this parameter to produce a new threshold to be used instead, when making the normal directed proxy movements:

$$T_N' = T_N W_d(\left|\vec{\nabla} V(\vec{x}_p)\right|) \tag{13}$$

This is not a natural material property, but it is anyhow essential for the natural behaviour of the haptic feedback.

This movement in the normal direction is done first, using equation 12. Thereafter a new normal directed force is calculated, which is used in equation 9 through 10 when computing the friction simulation and tangent directed movements. Last, the resulting force after all proxy movements in the present time step is used as force feedback to the haptic instrument. This reduces the probability of unstable behaviour, since the resulting force is bounded by the thresholds that are specified by the user in the material properties of the volume.

3.2 Viscosity Feedback

To provide a force feedback also in uni-valued areas, where no surface feedback is present, some kind of viscosity feedback must be presented. As the proxy movements induce haptic feedback that mimic friction and surface feedback, the movements can also mimic a 3d friction — a resisting force only dependent on the direction of motion and the viscosity material property. How well the 3d friction works as a natural material property is yet to be established.

The current implementation of 3d friction makes the feedback dependent on the proxy motion from surface simulation. The proxy is first moved to simulate the surface and friction feedback, according to equations 10 and 12. The displacement of the proxy relative to the instrument tip now give rise to a surface and friction simulating force. By moving the proxy back towards the original position of the proxy before the final force of the current time-step is calculated, a 3d friction force is added, in the direction in which the proxy was moved. The distance to move the proxy is defined by the 3d friction force to be added, but the proxy must not be moved back further than the original position. Thus the resulting force can never be greater than the force induced by the user. If the original proxy position is called $\vec{x}_{pO}$, the vector pointing from the current proxy position to the original position is $\vec{x}_{pO} - \vec{x}_p$ and thus the proxy is moved according to:

$$\vec{x}'_p = \begin{cases} \vec{x}_p + \frac{R\,(\vec{x}_{pO} - \vec{x}_p)}{k\,|\vec{x}_{pO} - \vec{x}_p|}, & \text{if } R/k < |\vec{x}_{pO} - \vec{x}_p| \\ \vec{x}_{pO}, & \text{otherwise} \end{cases} \tag{14}$$

where R is the magnitude of the 3d friction force. The principle can be seen in figure 3, where the proxy is first moved to simulate penetrability (1), second to simulate friction (2) and third to simulate viscosity (3). The proxy is thus moved from $\vec{p}_{pO}$ to $\vec{p}'_p$ in the current time-step.

3.3 Material Properties

The material properties, i.e. friction, stiffness, viscosity and penetrability, are controlled by the scalar value of the area being examined. Transfer functions are used in the same manner as visual transfer function are used in volume rendering. The analogy between transfer functions in visual and haptic volume rendering is shown in table 1. The scalar value is used as input and the user specified transfer functions give the friction, stiffness, etc. The material parameters friction, stiffness, viscosity and surface strength then become $\mu = F_{tf}(V(\vec{x}_s))$, $k = K_{tf}(V(\vec{x}_s))$, $R = R_{tf}(V(\vec{x}_v))$ and $T_N = T_{tf}(V(\vec{x}_s))$ respectively. How a kind of tissue is haptically represented is thus defined from the scalar value of the tissue. The positions $\vec{x}_s$ and $\vec{x}_v$, which are used as the positions where to pick the scalar values, will be explained later in this section.

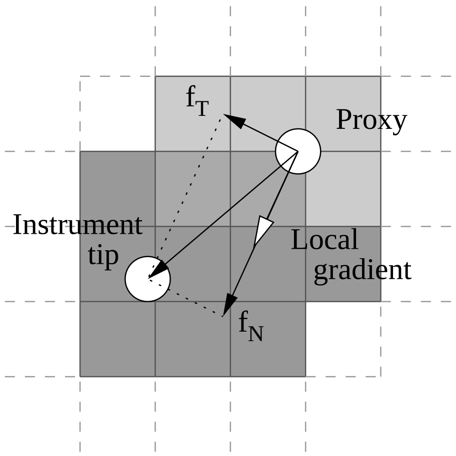

Figure 1. Force components from the proxy displacement

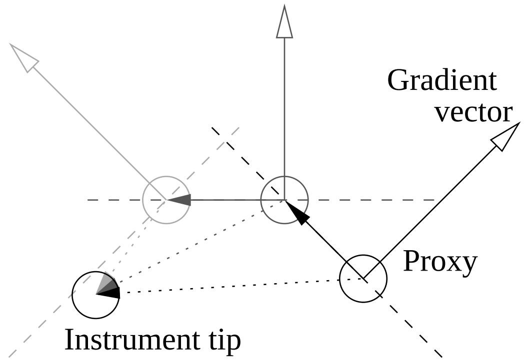

Figure 2. How the proxy follows an ‘virtual surface’

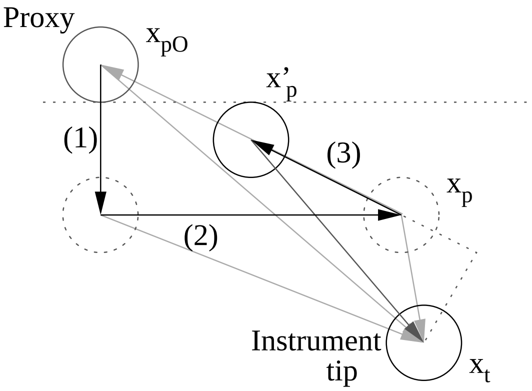

Figure 3. The proxy movement to simulate: 1) penetrability, 2) friction and 3) viscosity

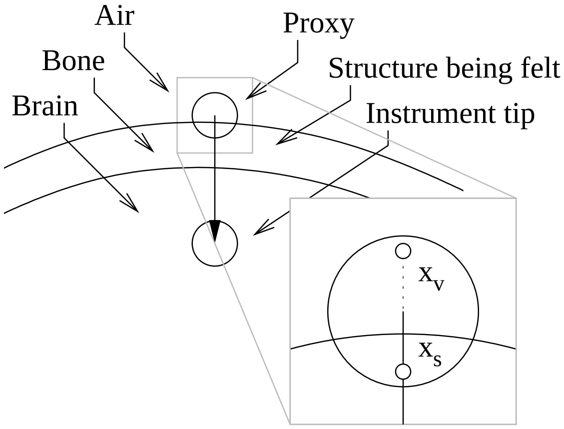

Figure 4. The scalar values used to control the material properties

By applying full penetrability and zero viscosity to certain scalar values, through the transfer functions T_{tf} and R_{tf} respectively, tissues can be set to have no haptic feedback. This can then be synchronised with the transfer functions for the visual rendering, so that only visible tissues give haptic feedback.

The scalar value to be mapped to the set of material properties for the surface, should be extracted from the structure that is being examined using the surface feedback. Thus, when examining surfaces of structures, the friction, stiffness and penetrability should be derived from a scalar value taken from inside that structure, while the viscosity property should be taken from the part of the volume in which the internal representation of the haptic instrument is located, i.e. outside the surface if such is being felt.

To be certain that the surface material is being picked from inside the structure and the viscosity material from the outside, the points from which to pick the material properties should be located on the boundary of the kernel used to estimate the gradient vector, with the rotation of the position controlled by the local gradient at that point, i.e. the surface normal. The positions where the surface properties and viscosity property are picked, $\vec{x}_s$ and $\vec{x}_v$ respectively, will then be positioned as shown in figure 4, relative to a distinct surface. When the gradient is undefined, i.e. the normal is in an arbitrary direction (as discussed in section 3.1), the neighbourhood is uni-valued or symmetric, and thus, since the points are located inside the kernel at an even distance, the values at the points are independent of their orientation.

4 Implementation

The haptic algorithm was implemented using `Reachin API`[11] from Reachin Technologies AB. The structure of the `Reachin API` is based on the scene-graph description file format `VRML`. Thus virtual environments, controlled by the `Reachin API`, are defined in `VRML`, but the nodes of the scene-graph are built in C++. The `Reachin API` handles a separate update loop for graphics and haptics at 60 and 1000 Hz respectively.

In the present work the `Reachin API` was extended to implement the proxy-based volume haptics algorithm by implementing additional `VRML` nodes. Two nodes, `VolumeRenderer` and `VolumeHaptics`, where created to provide visual and haptic representation of the volume data respectively. Moreover a `TransferFunction` node and a `Volume` node were implemented to provide control and data to the main nodes.

In the `VolumeHaptics` node, the haptization algorithm presented in section 3 was implemented. The parameters of the algorithm are defined as a set of transfer functions, like parameters of visual volume rendering. In addition to the material parameters the surface distinctness parameter, presented in section 3.1, is provided. The volume haptics node also provides the virtual environment with the position of the internal proxy representation, so that a visual proxy representation can be presented.

For comparison a variant of the general approach to direct volume haptics, described in section 2.2, was implemented as well.

5 Results

The implementation was tested on a Reachin Desktop Display[10] from Reachin, equipped with a stereo-scopic CRT monitor and a desktop PHANToM from Sensable Inc. The display was driven by a dual 800 MHz Pentium II with 256 MiB RAM and a Wildcat 4110 graphics card.

The gradient estimation is without comparison the most time consuming task in the algorithm. While the rest of the algorithm yields a constant delay of about 35μs, the time delay of the gradient estimation has cubic growth with respect to the radius of the gradient estimation kernel. With a kernel size of 2 voxels in radius, the haptic algorithm had no problems working at the 1kHz update rate, so with reasonable stiffness and friction settings no unstable behaviour has been experienced with the presently used datasets. The behaviour with high stiffness of surfaces is similar to the behaviour of systems using proxy-based surface haptics.

The friction feedback from the algorithm is effective even though the sensed difference between high and low friction is subtle. When following well defined surfaces it is virtually impossible to distinguish the volume haptics algorithm from the proxy-based surface haptics algorithm. The stiffness parameters, together with the parameter for threshold for letting the proxy pass through surfaces, can however be difficult to balance. The task to find a good set of

Visual Volume Rendering	Haptic Volume Rendering
Scalar value → color	Scalar value → material
Scalar value → opacity	Scalar value → surface penetrability
Gradient magnitude → opacity	Gradient magnitude → surface distinctness

Table 1. Analogy between transfer functions of visual and haptic volume rendering

parameters to control the normal directed maximum force, stiffness and threshold is yet to be solved.

The ability of the proxy-based method to follow and reconstruct fine details seems to be similar to that of general haptics. However, the number of parameters provided by this haptic algorithm makes it hard to configure to the desired behaviour. At the same time it is highly configurable and also flexible: The user can choose to remove the haptic feedback from certain tissues and define to which degree diffuse surfaces shall be rendered as surfaces.

6 Future Work

The algorithm shows promising results and will be used in several ongoing projects. With the presented haptic algorithm, the quality of force feedback from CT data can be dramatically increased. The algorithm can also be extended to facilitate complex interaction with CT datasets, but also aid in the comprehension of other datasets such as tensors and time varying data.

The introduction of transfer functions in the haptic rendering gives rise to new questions. How exact do the transfer functions have to be? What is needed to achieve a natural feeling? Since the haptic transfer function can be connected both to the nature of the data and to the visual rendering, other issues emerge. If the visual rendering does not have to be photorealistic, does the haptic rendering? The possibility of measuring material properties and scalar values of real world objects, to create automatic transfer functions, must also be further investigated.

References

[1] R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In IEEE Visualization, October 1996. pages 197–204.

[2] D. Bartz and Ö. Gürvit. Haptic navigation in volumetric datasets. Proc. of PHANToM User Research Symposium, 2000.

[3] K.-W. Chen, P.-A. Heng, and H. Sun. Direct haptic rendering of isosurface by intermediate representation. In ACM Symposium on Virtual Reality Software and Technology VRST '00, 2000.

[4] S. F. F. Gibson, J. Samosky, A. Mor, C. Fyock, W. E. L. Grimson, T. Kanade, R. Kikinis, H. C. Lauer, N. McKenzie, S. Nakajima, T. Ohkami, R. Osborne, and A. Sawada. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. In *CVRMed*, pages 369–378, 1997.

[5] W. Hashimoro and H. Iwata. A versatile software platform for visual/haptic environment. ICAT Paper, 1997. pages 106–114.

[6] H. Iwata and H. Noma. Volume haptization. IEEE 1993 Symposium on Research Frontiers in Virtual Reality, October 1993. pp. 16-23.

[7] O. Körner, M. Schill, C. Wagner, H.-J. Bender, and R. Männer. Haptic volume rendering with an intermediate local representation. Proceedings of the 1st International Workshop on the Haptic Devices in Medical Applications, June 1999. R. Dillmanns, T. Salb (eds.) Paris, 79-84.

[8] P. J. Passmore, C. F. Nielsen, W. J. Cosh, and A. Darzi. Effects of viewing and orientation on path following in a medical teleoperation environment. Proceedings IEEE Virtual Reality 2001, 2001.

[9] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. *Computer Graphics*, 31(Annual Conference Series):345–352, 1997.

[10] D. Stevenson, K. Smith, J. McLaughlin, C. Gunn, J. Veldkamp, and M. Dixon. Haptic workbench: A multisensory virtual environment. In Proc. SPIE Stereoscopic Displays and Virtual Reality Systems VI, volume 3639, pages 356–366, 1999.

[11] L. Thurfjell, A. Lundin, and J. McLaughlin. A medical platform for simulation of surgical procedures. In Westwood et al. (Eds), Medicine Meets Virtual Reality, IOS Press, 2001; pp. 509-514, 2001.

[12] S. Wall and W. Harwin. Quantification of the effects of haptic feedback during a motor skills task in a simulated environment. In Proceedings at Phantom User Research Symposium'00, 2000.