

Simulating Deformable Tools with Haptic Feedback in a Virtual Environment

S.D. Laycock, A.M. Day
School of Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK
S.Laycock@uea.ac.uk, AMD@sys.uea.ac.uk

ABSTRACT

Simulations for training and analysis incorporating haptic feedback are becoming more popular and the sophistication of these simulations is increasing. Many systems now allow a range of simulated rigid tools to be used in haptic feedback applications, such as scalpels in surgical simulations. However, very often the rigid property of the simulated tool does not correspond to the flexibility of the tool in the real world. This paper aims to show how deformable tools can be used in conjunction with haptic feedback to improve realistic interactive simulations. The work shows how a finite element model of a beam can be used to simulate a three dimensional deformable rod in a virtual environment. The ability to perform the collision detection, force responses and update the physical model at a suitable haptic rate is a crucial requirement. An approximation to the finite element method is illustrated which allows more time, per frame, to be allocated to the collision detection without noticeably degrading the physical model.

Keywords

Haptic Feedback, Finite Element Method, Deformable Objects, Interaction, Simulation.

1. INTRODUCTION

Haptic feedback is used extensively in training simulations where the user can practice a real world task by manipulating objects on the screen. The user is able to feel the objects via a device capable of exerting forces on the user. It is imperative that this simulation is as realistic as possible otherwise the training would be detrimental to the users' performance in the real world. Most current simulations for training involve probing the virtual environment with either a single point or a rigid object. By not considering the physical properties of tools in simulations the system is lacking an important factor which if present would increase the realism of the experience.

The aim of our research is to investigate ways of incorporating deformable tools in a virtual environment, with the main goal being to improve the

realism of tool use in simulations. The visual deformations of the tool will be combined with realistic haptic feedback. There are a variety of haptic feedback devices available, for a recent survey see [LAY02]. The PHANToM Desktop device [SEN] produced by SensAble Technologies has been chosen for this research. This device is particularly well suited to the desktop environment with a workspace area of 160mm by 130mm by 130mm and a suitable output force of 6.4N. The device is compact and the part the user holds is simple in design to represent the handle of many tools. In the virtual environment the simulated tool is required to be attached to the position of the virtual representation of the haptic device's endpoint, sometimes referred to as the haptic interface point.

Accurately modelling the tools is an essential part of this work. A method is required that will allow the simulations of the tools to be as realistic as possible. A flexible rod has been modelled using a finite element approach to simulate its physical behaviour and it is discussed in section 3. Collision detection is another crucial part of this work. The tools must be able to collide and interact accurately with the rest of the objects in the environment. For example a deformable tool could be required to collide with the cavity wall of a tooth for a dentistry application. Ensuring accurate collision detection is not the only requirement as the collisions must also be able to be

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG SHORT PAPERS proceedings.
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.
Copyright UNION Agency – Science Press

determined quickly in order to achieve the update rates of 1000Hz necessary for realistic haptic feedback. Section 4 describes how the rod will interact with the virtual objects in the environment.

2. PREVIOUS WORK

Haptic feedback has been used in conjunction with virtual rigid tools in a number of applications ranging from surgery [KUK00] to assembly tasks [STE98]. Developing simulations with rigid tools induces a large overhead when determining accurate reaction forces, particularly when there are multiple points of contact. A magnetic levitation haptic interface device has been discussed in [BER99] which was applied to simulating rigid tools. One test example allows the user to insert a virtual key into a virtual lock.

For certain applications the flexibility of tools should also be considered, otherwise the simulation will be too much of an approximation to be realistic. To achieve systems that match more closely to the real world combinations of hardware, representing tools and software have been developed. One application aims to train medical staff in performing laparoscopy and endoscopy procedures. It incorporates haptic feedback conveyed from holding the handles of the real tools as they interact with a plastic replica of a human body part. The ENT Surgical Simulator [ENT] was developed for endoscopic sinus surgery simulation where the operator controls a replica of an endoscope. A larger system has been developed by the partners Forschungszentrum Karlsruhe / Institut für Angewandte Informatik titled the Virtual Endoscopic Surgery Trainer (VEST)[KIS]. This system allows the user to manipulate an environment displayed on a screen whilst operating a variety of tools all mounted in a free standing case. These systems are useful for training applications and the VEST has produced some very promising results. The simulation of deformable tools in software has benefits over these integrated systems. An important benefit is the ability to tailor the software to a variety of applications without the need of new hardware. In conjunction with a smaller haptic device, such as the Phantom, a desktop environment can be used.

Koutek and Post [KOU01] have developed a method modelling flexible objects, using spring-damper models. Their approach allows flexible forks and probes to aid in object manipulation in a virtual environment. Reznik and Laugier [REZ96] have produced a more complex deformable object. Their work involves simulating the control and deformations of a virtual fingertip as it probes a rigid surface. The fingertip is modelled using a lattice of masses locally interconnected by damped springs. Both of the above procedures exhibit flexible

behaviour but are not linked with force feedback. The method described in our paper does convey force feedback and uses a finite element procedure which should provide an increased realism to the previous spring-damper models.

DAB [BAX01] is an example of a haptic paint program which allows the user to paint onto a virtual two dimensional canvas using a virtual three dimensional paintbrush. The Phantom Desktop is incorporated to provide the haptic feedback for this program. The user holds the stylus of the device like a paintbrush and the bristles of the brush are modelled such that realistic strokes can be produced under physical rules. Collision detection is required between the brush head and the canvas. When a collision occurs the brush head will deform. The deformation of the brush head is modelled using a subdivision surface encompassing a spring mass particle skeleton. A variety of brushes are simulated to allow several effects to be produced. Another example of simulating bristles with haptic feedback has been developed by Gockel et. al. [GOC02]. The aim of this research is to simulate the tooth cleaning process to aid in the design and manufacturing of new toothbrushes. A brush with more than one contact point cannot be modelled with realistic haptic feedback, instead keyboard entry is used for the control of the virtual brush. In the case of greater than thirty bristles a movie can be constructed simulating the operation of the brush when real time frame rates cannot be achieved. The main aim in our paper is to achieve realistic simulations of deformable tools whilst complying with an update rate suitable for stable interactions incorporating haptic feedback.

3. THE PHYSICAL MODEL OF THE ROD

The rod is required to deform realistically and so a physical model is employed to ensure the rod will behave appropriately as it comes into contact with the objects in the scene. A finite element method is used as the basis for the physical model. The next section discusses exactly how this model has been developed.

3.1 The Finite Element Method

The finite element method works by decomposing an object into smaller parts, known as elements. Physical formulae can then be applied to the elements individually before being assembled to describe the physical behaviour of the entire rod [COO95]. The theory of a beam has been incorporated to determine how much the rod should displace when under specified loads. The formulae for the beam elements can be obtained from standard deflection formulae describing the nodal displacements [FEN89]. Three

dimensional beam elements have been incorporated to approximate the rod. Each element comprises of two end points, called nodes, each with six degrees of freedom. The degrees of freedom correspond to translations and rotations about each of the axes. In this simulation the rotation about the x axis has not been considered and so each element has ten degrees of freedom. The equations for the nodal displacements can be written in matrix form. This matrix is the stiffness matrix for a single element [AGA00], shown in table 1. The stiffness matrix constitutes the equations that describe the displacements of each degree of freedom. The corresponding degrees of freedom have been labelled across the top of the matrix in table 1.

Δx_1	Δy_1	Δz_1	θ_{y_1}	θ_{z_1}	Δx_2	Δy_2	Δz_2	θ_{y_2}	θ_{z_2}
$\frac{AE}{L}$	0	0	0	0	$-\frac{AE}{L}$	0	0	0	0
0	$\frac{12EI}{L^3}$	0	0	$\frac{6EI}{L^2}$	0	$-\frac{12EI}{L^3}$	0	0	$\frac{6EI}{L^2}$
0	0	$\frac{12EI}{L^3}$	$-\frac{6EI}{L^2}$	0	0	0	$-\frac{12EI}{L^3}$	$-\frac{6EI}{L^2}$	0
0	0	$-\frac{6EI}{L^2}$	$\frac{4EI}{L}$	0	0	0	$\frac{6EI}{L^2}$	$\frac{2EI}{L}$	0
0	$\frac{6EI}{L^2}$	0	0	$\frac{4EI}{L}$	0	$-\frac{6EI}{L^2}$	0	0	$\frac{2EI}{L}$
$-\frac{AE}{L}$	0	0	0	0	$\frac{AE}{L}$	0	0	0	0
0	$-\frac{12EI}{L^3}$	0	0	$-\frac{6EI}{L^2}$	0	$\frac{12EI}{L^3}$	0	0	$-\frac{6EI}{L^2}$
0	0	$-\frac{12EI}{L^3}$	$\frac{6EI}{L^2}$	0	0	0	$\frac{12EI}{L^3}$	$\frac{6EI}{L^2}$	0
0	0	$-\frac{6EI}{L^2}$	$\frac{2EI}{L}$	0	0	0	$\frac{6EI}{L^2}$	$\frac{4EI}{L}$	0
0	$\frac{6EI}{L^2}$	0	0	$\frac{2EI}{L}$	0	$-\frac{6EI}{L^2}$	0	0	$\frac{4EI}{L}$

where E = Modulus of Elasticity, I = Moment of Inertia,
A = Cross-sectional Area, L = Element Length

Table 1. 3D Beam Element Stiffness Matrix, km

The next stage is to use the single element stiffness matrix for each element to describe the equations for the entire rod consequently the global stiffness matrix is formed. To show how this has been achieved an example of a rod partitioned into two elements will be described. Figure 1 illustrates a rod with the two elements labelled A and B and three nodes, denoted by black circles. The degrees of freedom of each node are labelled on the diagram.

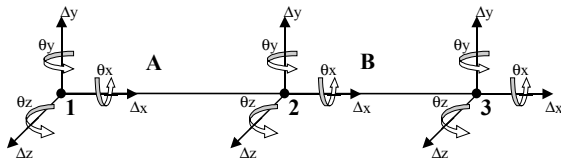


Figure 1. Beam partitioned into two elements.

The single element stiffness matrices for A and B need to be concatenated together. This procedure works by merging the equations which effect the nodes for both elements. Figure 2 illustrates the

concatenation procedure. It shows a rectangle, representing the global stiffness matrix divided into smaller boxes. Each box representing the locations of the formulae linked to each node. It is clear from Figure 1 that elements A and B share node 2. This is represented by the overlapping region, highlighted with a dark grey box in figure 2. Two ten by ten 3D beam element matrices can be inserted into the top left and bottom-right rectangular areas of the global matrix. In the overlapping section the terms concerned with node 2 in the matrix for element A are added to the terms concerned with node 2 in the matrix for element B. The white areas in the figure represent blocks of zeros.

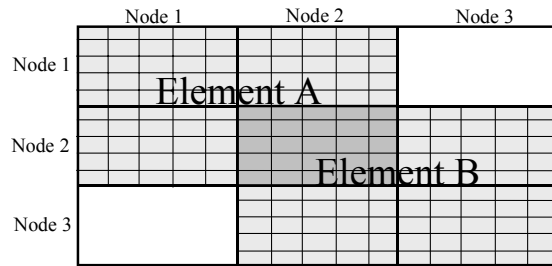


Figure 2. The concatenation procedure forming the global stiffness matrix, K.

The concatenation procedure can easily be extended for more elements. The global stiffness matrix will become large relatively quickly, as more elements are required, with the significant section running diagonally from top-left to bottom-right.

$$KD = R \quad (1)$$

where K = global stiffness matrix.

D = deflections

R = load vector

The load vector, R, is a column vector comprising of the force components acting on the rod. How these forces have been determined will be discussed in section 4.1. The result that is required is the column vector of deflections stored in D. These deflections describe the displacement for every degree of freedom of each element. The global stiffness matrix can be solved when the boundary conditions are taken into account. These conditions are defined by any known displacements. In this example it has been assumed that the first node of the rod is going to be located close to the handle and will not displace. This allows the first five rows and columns to be removed from the current global matrix to produce the actual global matrix, K used in equation (1).

The theory described so far assumes the rod to be aligned with the axes. This is not always the case as the user is able to orientate the rod to any position they desire as they move the arm of the haptic device.

The next section explains how the displacements of the beam can be determined when it is transformed to any orientation.

3.2 Orientating the Beam

In the previous section the theory assumes the rod to be axes aligned. To allow the beam to be arbitrarily orientated local axes can be assigned to each element. The procedure will then be to transform the set of equations from the local coordinates of the beam elements to the global coordinates of the scene in which the forces are represented. The outline for the procedure used in this section is described by Agarwal in [AGA00]. Figure 3 illustrates a two dimensional beam element rotated about an end point. The smaller local coordinate axes are shown on the diagram and labelled with underscored terms.

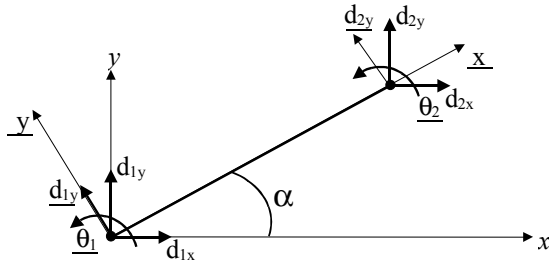


Figure 3: Arbitrarily Orientated Two dimensional Beam.

The local coordinate displacements, \underline{d}_{1y} and \underline{d}_{2y} , can be obtained from the global coordinates using trigonometry as shown in equations (2) and (3).

$$\underline{d}_{1y} = d_{1y} \cos \alpha - d_{1x} \sin \alpha \quad (2)$$

$$\underline{d}_{2y} = d_{2y} \cos \alpha - d_{2x} \sin \alpha \quad (3)$$

The transformation matrix can be generated by writing equations (2) and (3) in matrix form. Multiplying the transformation matrix by the global coordinate displacements will produce the local coordinate displacements. The relationship between the local and global coordinate stiffness matrices for each element can be formed using the element transformation matrix, T .

$$Kg = T^T km T \quad (4)$$

where Kg = global coordinate stiffness matrix.

T = element transformation matrix

km = local coordinate stiffness matrix

Once the global coordinate element stiffness matrix has been found it can be used in the concatenation and boundary condition procedure described in the previous section. This forms the set of equations for the entire beam. The complete algorithm is summarised in the finite element procedure in section 5. The theory shown in this section is illustrated with

the simpler two dimensional case but can be extended to three dimensional beam elements. The first step is to multiply together the transformation matrices for the three rotations about each of the axes. This will result in a three by three transformation matrix. The values in this transformation matrix can now be entered into the element transformation matrix as shown in table 2. The notation used in the table consists of two lettered pairs, firstly a c or s and secondly x, y or z. The c represents cosine of the angle and the s represents the sine of the angle. For example, cy = cosine of the angle about the y axis and $czcy$ represents the cosine of the angle about the z axis multiplied by the cosine of the angle about the y axis. Transforming the angles from local to global coordinates does not affect their values and so a '1' is inserted in the corresponding columns in the table. The degrees of freedom are labelled across the top of table 2. The element transformation matrix, T , can now be used in equation (4) to obtain the global coordinate stiffness matrix for a three dimensional beam element.

Δx_1	Δy_1	Δz_1	θy_1	θz_1	Δx_2	Δy_2	Δz_2	θy_2	θz_2
czcy	szcy	-sy	0	0	0	0	0	0	0
czsysx	sysxsx	sxcy	0	0	0	0	0	0	0
-cxsz	+cxcz								
cxsysz	cxsysz	cxcy	0	0	0	0	0	0	0
+sxsx	-sxcz								
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	czcy	szcy	-sy	0	0
0	0	0	0	0	czsysx	sysxsx	sxcy	0	0
					-cxsz	+cxcz			
0	0	0	0	0	cxsysz	cxsysz	cxcy	0	0
					sxsx	-sxcz			
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

where c=cosine, s=sine.

Table 2. Element Transformation Matrix, T

4. INTERACTION BETWEEN THE ROD AND THE VIRTUAL OBJECTS

The second part of our research is to incorporate the interactions between the rod and the other objects in the virtual environment. This section can be further split into two main tasks. Firstly to determine collisions between the rod and objects in the scene. Secondly to apply appropriate forces to the rod to prevent it from passing through the virtual objects in the environment.

The collision detection needs to be accurate and fast in order to achieve the required update rates for haptic feedback. Two levels of detection are incorporated. To speed up the collision detection the first level uses a bounding sphere encompassing each object in the scene to allow objects to be trivially rejected. Using this rejection method at least thirty

objects can be positioned in the scene, assuming each element only intersects one bounding sphere at once. When more than thirty objects are included then the performance begins to degrade¹. If the rod passes inside the bounding sphere of an object then the second level of collision detection is utilised. Test points on each of the rod's elements are used in a point relative to plane test [ROU99] providing a more accurate measure of intersection. The discrete nature of the test points makes it relatively easy to calculate collisions but has a disadvantage as four test points per edge of each element are required to provide the feeling of continuous collision. If scenes are created which have a large number of separated models then another level of detection could be included to remove blocks of objects. For instance, when a group of objects are located away from the rod then that whole group could be rejected in one test. This level of detection is currently not incorporated as only a small number of objects are being considered.

4.1 Calculating Forces on the Rod

The response to a collision needs to be dealt with by way of a force which will aim to prevent the rod from passing through the virtual objects. A relatively simple relationship based on the penetration depth of the point inside the object has been used to provide the force. One problem with this approach is determining which face the point entered first (contact face). In an attempt to alleviate this problem the distances from the test point to each triangular face of the colliding object are calculated. It is assumed that the test point will be closest to the contact face. There are notably some problems with this approach which can cause force discontinuities. These problems are discussed in [ZIL95] and aim to be resolved in future work.

Once the penetration depth and the contact information have been determined this can be used as the displacement of a spring attached to the surface of the polygon. A spring-damper model is then used for easy alteration of the behaviour of the objects' faces in the environment. The following equations are used for the spring-damper model.

$$F_s = -Kx \quad (5)$$

$$F_d = -Bv \quad (6)$$

$$F_{total} = F_s + F_d \quad (7)$$

where x = the distance of the test point from the contact face, B = attenuation constant, K = spring constant, v = velocity, F_{total} = total force, F_s = spring force, F_d = damping force.

Increasing the spring constant allows stiffer contacts to be perceived and introducing the damping factor aids in stabilising the haptic response.

Each element of the rod has a series of test points around its edges, as mentioned in the previous section. These are able to approximate the collision detection required. Each test point is used in the above method to determine the forces that act on the beam due to collisions with virtual objects. The forces determined for each element can be applied to the load vector in the finite element method discussed in the previous section. Each entry in the load vector corresponds to a degree of freedom of the elements. The resultant of each element force can finally be determined and sent to the haptic device. A simple linear force interpolation method described in [GRE00] is included to prevent any large force changes that may occur. This method works by adding a component of force to the resultant force if the change in force between consecutive loops is larger than a predetermined value.

5. IMPLEMENTATION

The main requirements to be considered during the implementation phase of this work are; to achieve realistic deformations of the rod and fast and accurate collision detection whilst complying with the update rate of 1000Hz.

The finite element approach requires a significant proportion of the computing time available per iteration. The reason is that as the user moves the haptic device the transformation matrix of the rod alters. This matrix is required to be used to transform the single element stiffness matrix before it can be concatenated to form the global stiffness matrix as described in section 3.1. The complete procedure from determining the local element stiffness matrix to obtaining the deflections of the rod is described below.

Finite Element Procedure

1. Create k_m , the ten by ten 3D beam element matrix as shown in table 1.
2. Obtain the three dimensional rotation matrix, m , of the rod.
3. Apply the relevant parts of m to the finite element transformation matrix, T , as illustrated in table 2.
4. Obtain the global element stiffness matrix using equation (4).
5. Use the concatenation procedure to obtain the global stiffness matrix, K , for the rod.
6. Apply the boundary conditions, i.e. remove the first five rows and columns of K .

¹ Computed on Pentium III 733MHz

7. Find the LU decomposition of K. $KD = R$ split to $LUD = R$.
8. Use forward substitution to find y ($Ly = R$) followed by backward substitution to obtain D ($UD=y$).

Steps 1 to 6 can be performed relatively quickly. The problem lies when the decomposition of the matrix is required to be found. If a rod comprised of ten nodes is used then the matrix to be decomposed will have forty-five columns and forty-five rows once the boundary conditions have been taken into account. Obtaining a fast method for determining the decomposition of a matrix is crucial to the running time of this method. An LU decomposition method has been applied. The usual LU decomposition works relatively well for this task but can be made more efficient if a special property of K is considered. K is a band diagonal matrix with the significant part of the matrix running close to the diagonal. An implementation of the LU decomposition utilising this property has been adopted which significantly reduces the running time of the standard LU decomposition as shown in table 3. The space requirement for the complete matrix is also reduced. Using the standard matrix a forty-five by forty-five matrix totaling 2025 locations is required. Equation 8 below generalizes the number of locations required for the standard LU decomposition based on the number of nodes and the degrees of freedom.

$$((N-1)*dof)^2 \quad (8)$$

where $N =$ Number of nodes in the rod
 $dof =$ number of degrees of freedom.

The matrix can be represented in a more compact form by not storing the zeros off the diagonal. The space requirement is reduced to 855 locations. Equation 9 can be used to calculate the space requirement.

$$((4*dof)-1)*((N-1)*dof) \quad (9)$$

Obtaining forces to be applied to the load vector is not a trivial procedure. Forces are required to be applied to the rod so that a smooth contact can be perceived as the edge of the rod slides over the surface of the other objects. To determine the forces four discrete test points are used on each edge of every element to simulate continuous collision detection and force calculations. Achieving these test point collisions in addition to the finite element procedure at real time frame rates can be accomplished for simple models. As the polygon count of individual objects in the scene increases the time required to compute each point relative to plane test increases. The time required to perform the collision detection and the finite element procedure will become higher than the time allowed for each

iteration. In this case an approximation to the finite element procedure can be incorporated. Steps 2 to 7 in the procedure can be computed less frequently allowing step 8 to be computed every update loop to obtain the deflections. Using this approximation allows real time frame rates to be achieved without noticeable deterioration of the physical model. As the collision problem becomes more time consuming it is necessary to reduce the frequency that steps 2 to 7 are performed. Consequently the incorrect orientated global stiffness matrix is used in the finite element method to obtain the deflections. At this point the physical model will become noticeably unrealistic. Table 3 below indicates the results obtained when varying the number of nodes. Row A, of the table illustrates the total time of the procedure when using the band diagonal method and Row B, indicates the times when the standard LU decomposition is utilized. The top row of the table indicates the number of nodes. The table shows that the time per iteration required increases significantly as the number of nodes increases for the LU decomposition. However, a rod consisting of ten nodes can be represented when using the band diagonal method.

	4	6	8	10	12	14	16	18
A	0.15	0.23	0.31	0.37	0.45	0.53	0.59	0.67
B	0.22	0.47	0.92	1.62	2.61	3.92	5.62	7.78

Table 3. Indicates times in seconds ($\times 10^{-3}$) to complete the finite element procedure, computed on Pentium III 733MHz.

The implementation has been undertaken using C++ with OpenGL for the rendering. The GHOST SDK supplied with the Phantom device is used to provide the haptic feedback. This SDK is object oriented and allows classes to be extended easily for more specific purposes. The `gstManipulator` class has been extended to allow the rod to be linked to the haptic device. Figure 4 shows a view of the simulation as the program runs. The rod can be seen to be deformed as it has been pressed against the box. The cone-like object represents the virtual endpoint of the haptic device.

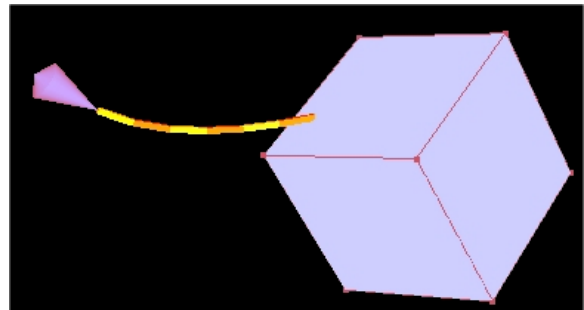


Figure 4. Deformation of the Rod when pressed against an Object

6. CONCLUSION

A method has been developed which can be used to simulate deformable tools with haptic feedback. This work has shown that a deformable tool modelled using a finite element approach can be implemented with reasonably accurate collision detection and at realistic frame rates. The efficiency, in terms of both space and time, of the band diagonal matrix decomposition has proved useful in achieving these frame rates. The approximation in obtaining the deflections has also been of benefit. It has allowed more time to be spent on the collision detection without too much of an approximation to the physical model.

This work has highlighted problems that can occur with collision responses in haptic rendering. The main problem is in obtaining continuous forces that act in an appropriate direction. A discrete test point approach has been developed with the direction of the force based on the normal of the face nearest to the test point. The test point approach is able to simulate continuous forces when four test points are used but this increases the computation required in performing the test. A method for partitioning the scene to allow objects to be trivially rejected has proven indispensable in achieving higher polygon counts. There are some limitations in determining the direction of forces and overcoming them will be the focus of future work.

The overall goal of this research is to develop a framework which will allow a variety of deformable tools to be simulated in the virtual environment. The interactions between these tools and the virtual objects will also be developed to allow material properties, such as friction to be perceived. This increase in realism should ensure better quality simulations with haptic feedback involving tool use.

7. REFERENCES

- [AGA00] R.B. Agarwal, ME273 Lecture Notes, Beam Element, http://www.engr.sjsu.edu/ragarwal/ME160/Chapter_4_Beam_Element.pdf
- [BAX01] B. Baxter, V. Scheib, V., M.C. Lin and D. Manocha. DAB: Interactive Haptic Painting with 3D Virtual Brushes, Proc. of ACM SIGGRAPH, pp. 461-468, 2001.
- [BER99] P. Berkelman, R. Hollis, and D. Baraff. Interaction with a Realtime Dynamic Environment Simulation using a Magnetic Levitation Haptic Interface Device, Proc. IEEE International Conference on Robotics and Automation, pp. 3261 - 3266, May, 1999, Michigan, United States.
- [COO95] R. D. Cook. Finite Element Modeling For Stress Analysis. John Wiley and Sons, Inc, 1995.
- [ENT] Ent Surgical Simulator Mid-Term Report, <http://www.lockheedmartin.com/akron/busdev/sim&rng/medsim/report/midterm.htm>
- [FEN89] R. T. Fenner, Mechanics of Solids, Blackwell Scientific Publications, 1989.
- [GOC02] T. Gockel, U. Laupp, T. Salb, O. Burgert and R. Dillmann. Interactive Simulation of the Teeth Cleaning Process using Volumetric Prototypes, Medicine Meets Virtual Reality (MMVR), January 2002, Newport Beach, USA.
- [GRE00] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin and D. Manocha. Six Degree-of-Freedom Haptic Display of Polygonal Models, Proc. Visualization 2000, pp139-146, October 2000, Utah, United States.
- [KOU01] M. Koutek and F.H Post. Spring-Based Manipulation Tools for Virtual Environments, Proc. of Immersive Projection Technology and Virtual Environments, 2001, Springer, Stuttgart, Germany, pp. 61-70.
- [KUK00] U. KuKhnafel, H.K. Cakmak and H. MaaS. Dynamic Medical Visualization Endoscopic surgery training using virtual reality and deformable tissue simulation, Computers and Graphics, vol.24, 2000, pp671-682.
- [LAY02] S.D. Laycock and A.M. Day. Recent Developments and Applications of Haptic Devices, Submitted to Computer Graphics Forum.
- [ROU99] J. O'Rourke, Computational Geometry in C, Second Edition, Cambridge University Press, 1999.
- [REZ96] D. Reznik and C. Laugier. Dynamic Simulation and Virtual Control of a Deformable Fingertip, Proc. IEEE Int. Conf. on Rob. & Autom. (ICRA), Minneapolis, MN, April 1996.
- [SEN] SensAble Technologies, <http://www.sensable.com>.
- [STE98] R. Steffan, T. Kuhlen, and A. Looock. A Virtual Workplace including a Multimodal Human Computer Interface for Interactive Assembly Planning, IEEE International Conference on Intelligent Engineering Systems (INES'98), Vienna.
- [KIS] Virtual Endoscopic Surgery Trainer, <http://www-kismet.iai.fzk.de/KISMET/VestSystem.html>.
- [ZIL95] C. B. Zilles, J.K. Salisbury. A constraint based God-Object method for haptic display. Proc. of the IEEE Conference on intelligent robots and systems, 1995, pp146-151.