Real-time Soft Tissue Modelling for Web-based Surgical Simulation: SurfaceChainMail

Ying Li[1] **, Ken Brodlie**[1]**, Nicholas Phillips**[2]
[1]*School of Computing, University of Leeds, Leeds LS2 9JT UK*
[2]*Department of Neurosurgery, Leeds General Infirmary, Leeds, UK*
E-mail: {ying,kwb}@comp.leeds.ac.uk
nickp@ulth.northy.nhs.uk

Abstract. The Web provides a useful environment for simple surgical training simulations. A combination of VRML for 3D rendering, and Java code for the simulation engine, has been used for a range of simple neurosurgical demonstrators. However the elements in these simulators are rigid, to avoid the computational complexity of deformable modelling. In this paper we describe a variation of the ChainMail technique that allows us to provide real-time deformable modelling, even in a Web browser environment on a PC. Our new algorithm, SurfaceChainMail, has been used to develop a simulator for the cutting of two layers of tissue, and separating the layers by pulling them apart.

1. Introduction

There is increasing interest in the use of the Web to deliver simple surgical training simulators. These use a combination of VRML for 3D rendering, and Java code for the simulation engine [5, 6]. A major advantage is their simplicity, available for use at any time, anywhere, with the only requirement being a PC and Web browser. This technology has proved sufficient to create simple neurosurgery demonstrators, where all the elements can be treated as rigid bodies and no deformations occur.

However there are many cases in which soft tissue is involved, and this requires the use of deformable modelling. The challenge we address in this paper is to provide an approach to soft tissue modelling that can be used in a Web-based simulation environment. The work is motivated again by a neurosurgical application; it involves the separation of two layers of tissue, by progressively cutting the material which links the layers, and gradually pulling on the layers to separate them. This requires real-time deformable modelling – which typically requires high performance computing facilities in order to provide an accurate solution. The challenge in our context is to find a feasible solution for simple PC-based computers, trading a degree of accuracy in return for greater speed.

The two principle approaches to soft-tissue modelling over the past decade have been the mass-spring approach, and the Finite Element Method (FEM). Mass-spring models [7] comprise a set of nodes connected by springs, with point masses attached at each node. Real time performance can be achieved with a limited number of nodes, but the behaviour is often unrealistic and can be unstable.

The FEM approach [1,8] produces more accurate results. However, a major problem for real-time modelling is the high computational cost, since the true elastic behaviour of soft tissue is nonlinear, and for large meshes, a large system of equations have to be solved. To counter this, Bro-Nielson *et al* [1] propose a simplified approach: the elastic deformation is assumed to be linear (so only small deformations are accurately modelled); a condensation procedure and pre-processing are included. This condensation step effectively focuses only on the surface nodes of the mesh, reducing the size of the linear system of equations. However, interactive topology changes, such as cutting of soft tissues, are not possible since the pre-processing assumes a certain topology.

A different approach to soft tissue modelling has been proposed by Gibson in a series of papers [2,3,4]. 3D ChainMail is a simple technique but is able to handle in real-time large datasets and large deformations. An object is modelled as a set of point elements, linked in a uniform rectilinear mesh. In contrast to FEM, where complex calculation is carried out on a (relatively) small number of mesh elements, ChainMail carries out simple calculations on a (potentially) large number of elements. There are two steps involved: the ChainMail process itself which imposes simple geometric constraints on movement, similar to chainmail armour; and a relaxation process where energy minimization is applied to refine the shape. An important advantage is that a change in position of one element is typically propagated only to a small number of neighbouring elements. In addition, changes in topology are easily accommodated by breaking links – so cutting is readily simulated.

For our application, speed is of the essence. The Gibson ChainMail is designed for volumetric objects, in which deformations, not only on the outer surface, but also in the interior, are modelled. We study a rather different approach, still using the ChainMail idea, but where we only consider surface elements and do not attempt to model interior behaviour. The result is a promising approach to soft tissue modelling that can be used for Web-based simulation on low-cost PCs.

2. 3D ChainMail on Uniform Meshes

In the ChainMail algorithm [2], the deformation of an object is determined by two processes. In the first process, a deformation step, one element is moved to a new position, potentially causing each of its neighbours, and their neighbours, and so on, to move to new positions, in a chain reaction. Neighbours only move if they fail to satisfy proximity constraints which determine the softness of an object. The raw positions of this deformation step are then adjusted by a second process, a relaxation step, which aims to minimize the energy of the configuration.

We illustrate the method in more detail in Figure 1 – which shows the technique in 2D for ease of understanding. Suppose element A has been moved to the new position shown. Then its right neighbour, E, is a candidate for movement in the chain reaction. E is constrained to be greater than *MinDist*, and less than *MaxDist*, in its horizontal separation from A; and to be less than *MaxShear* in its vertical separation. If any of these constraints are violated, then E moves the minimum distance to be within the marked region in Figure 1, and therefore within the constraints. Next the right neighbour of E has to be checked, and so on for all right neighbours. If an element is moved, its left, top and bottom neighbours also have to be checked, and the algorithm builds up lists of elements that need to be checked later. Once the right neighbours are processed, the list of left neighbours is dealt with; then the top neighbours; and finally the bottom neighbours. The top and bottom neighbours are constrained in a similar way to Figure 1, except that the movement is constrained relative to vertical position. The key point however is that any element is moved no more than once. Softness is controlled by the values of the constraints: clearly if *MinDist* is very close to *MaxDist*, then the object is nearly rigid, and all elements move in concert.

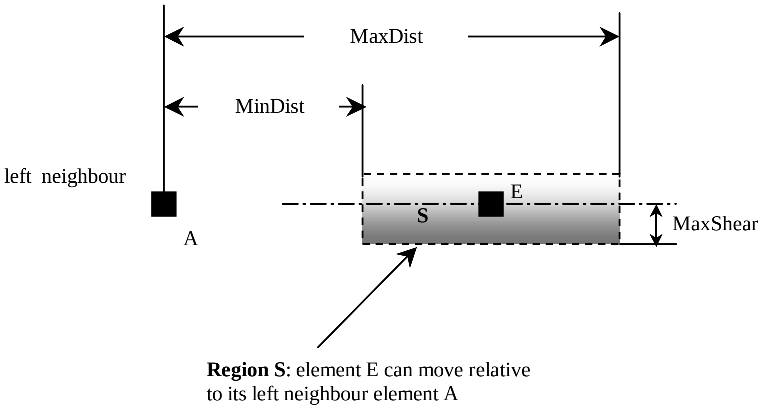


Figure 1 The moved element **E** is limited by the constraints of its left neighbour **A**.

The 3D ChainMail algorithm works in exactly the same way as the 2D version just described, except we now have front and back neighbours as well, and the constraints define a cuboid rather than the rectangle of Figure 1.

In the relaxation process, the system energy (defined in terms of the distance between elements) is iteratively reduced to a minimum. The positions of the elements are locally adjusted so that the distances between these elements are within an optimal range.

3. Extension to Non-uniform Grids

The Gibson ChainMail algorithm is defined only for uniform meshes. However it is straightforward to handle non-uniform meshes by replacing the absolute constraints of Figure 1 by constraints expressed relative to the separation. This generalisation to non-uniform meshes will enable us to use the ChainMail technique in a wider class of modelling applications. By non-uniform we mean any mesh topologically equivalent to a uniform rectilinear mesh – that is, a mesh with connections from each interior element to left, right, top, bottom, front and back elements. The mesh can be rectilinear or curvilinear.

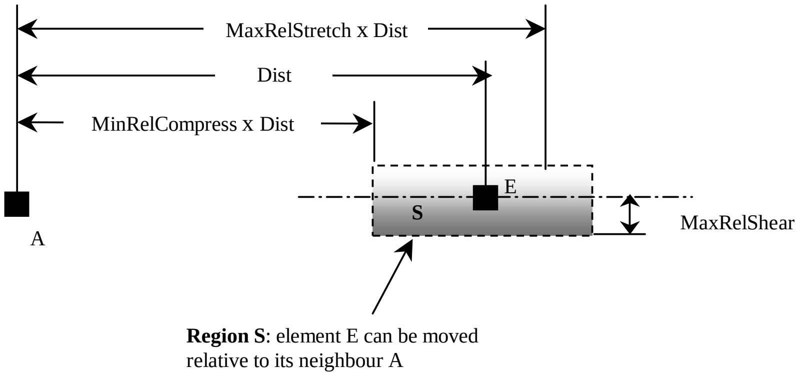


Figure 2. The constraints between element A and its neighbour E

The modification is to separate each geometric constraint into two parts: first, a material parameter which controls the softness, and second, the distance between the linked elements.

Again suppose A has been moved to a new position, and its right neighbour, E, has become a candidate for movement in the chain reaction. Suppose Dist is the distance between A and E. The valid region for E is shown in Figure 2, and is expressed by material parameters, *MaxRelStretch*, *MinRelCompress* and *MaxRelShear*, relative to the separation, *Dist*. With this modification, we can apply ChainMail to non-uniform grids. By tuning the values of the material parameters, objects with different elasticity can be modelled in an efficient way.

4. Extending to Surfaces in 3D : SurfaceChainMail

The Gibson ChainMail algorithm was designed for 3D volume modelling where the elements are on a regular, rectilinear mesh – the usual voxel-type structure. Many objects, however, are perfectly well modelled as surfaces. In this section we describe a new variation of ChainMail designed specifically for surfaces: we decompose the deformation proess into two steps – first, a deformation in the plane of the surface (using the extension to non-uniform grids just described), and second, a deformation in the direction of the surface normal.

In the first process, a 2D mesh is mapped onto the surface of the 3D object (as we do with texture mapping in graphics). Thus we first define a 2D non-uniform mesh, S, in the (u,v) – plane. The mesh is then mapped onto the target 3D surface object T(x, y, z), where x, y and z are the co-ordinates in the 3D surface domain. The transformation can be specified by three functions: x = X(u, v), y = Y(u, v) and z = Z(u, v). The 2D ChainMail technique is performed on the 2D mesh and the updated positions are then transformed onto the 3D surface.

In the second process, the surface normal at an element is obtained by the local average of the normals of surrounding facets. If the displacement of the moved element in the normal direction is less than *MaxRelShear*Dist*, then its neighbour is unchanged; but if the displacement is greater than this, the neighbour is moved in its normal direction so that the separation in that direction between the two elements is *MaxRelShear*Dist*.

At the end, the updated results from these two processes are combined together to produce a deformed new shape of the 3D surface. Figure 3 shows the examples of using these techniques. In the three pictures to the left, a cut cylinder is pulled and pushed; to the right, a cut sphere is deformed. We term the new technique: SurfaceChainMail.

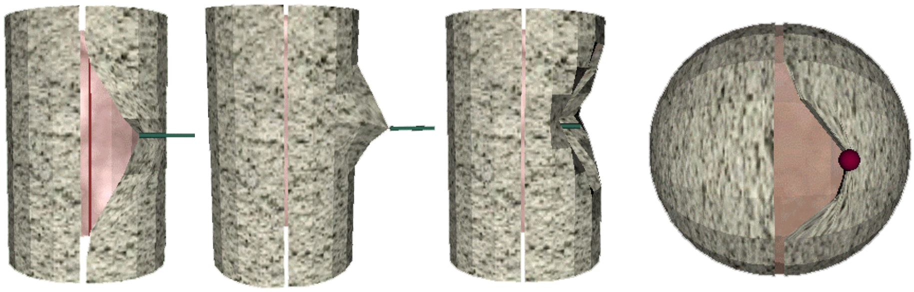


Figure 3. Examples of deformable surface using surface based ChainMail

5. *Implementation in Web-based Environment*

SurfaceChainMail has been implemented as a Web-based application so that it can be used as a surgical training simulator in the collection of such tools being developed [9]. It uses a combination of VRML and Java: VRML to provide the visual display and Java code to provide the real-time soft tissue modelling using SurfaceChainMail. The Java External Authoring Interface provides the link between the two. The system structure is shown in Figure 4.

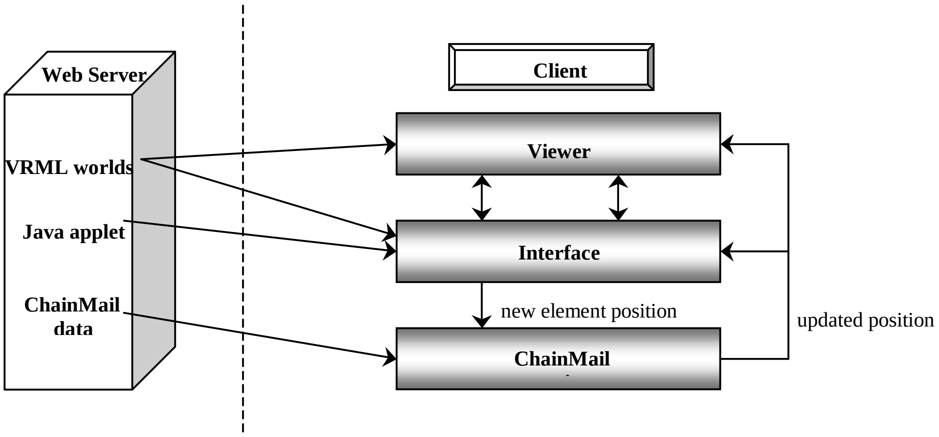


Figure 4 – System Structure

The system has three main components: the view, the interface and the ChainMail engine. When the simulator Web page is loaded from the server, the VRML world is downloaded into the view component and into the interface component, a Java applet is downloaded into the interface, and the mesh structure into the ChainMail engine. When the user alters a position of an element, this is transmitted from the interface to the ChainMail engine which then computes the resulting new mesh positions. These are transmitted back to the view and interface components. The Java EAI for VRML provides the 'glue' which allows the components to communicate with each other.

SurfaceChainMail has been applied to the cylindrical and spherical objects as shown in Figure 3, and also to a box-shaped object as described below (see Figure 5).

6. Surgical Training Application

One of the advantages of the ChainMail approach is that it supports topological changes in the interactive modelling : links between an element and its neighbours can be easily disconnected when needed. This applies in exactly the same way to SurfaceChainMail. This has allowed us to build a general simulator for the cutting and separation of layers of soft tissue. This models a very fundamental technique in surgery of all types for dissecting out structures as part of a more complex operation. In neurosurgery a common start to many operations involves the dissection of the covering layers of the brain (- the meninges) to allow access to deeper structures such as blood vessels.

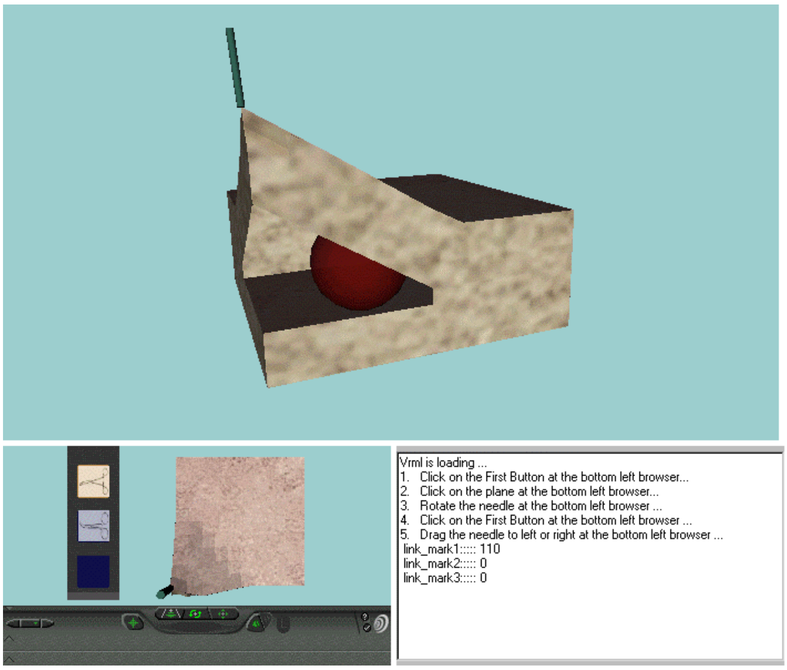
Figure 5 shows a screenshot of the simulator. There are two linked VRML worlds: the trainee manipulates the cutting instrument in the lower left browser window, while a view of the resulting deformation is provided for the trainee, and observers, in the upper window.

Cutting is simulated when the user defines a 'cut path' using a mouse as virtual instrument. Elements close to the cut path are identified, and any links with their neighbours which are crossed by the cut path are removed from the data structure. Deformation is simulated when the virtual instrument pulls or pushes the object: an element is moved to a new position by the instrument and SurfaceChainMail calculates the resulting effect. In Figure 5, we see two layers of tissue that have been cut, and teased apart using the instrument.

7. Conclusions and Future Work

We have described a variation of the 3D ChainMail algorithm introduced by Gibson for volumetric modelling. This variation uses non-uniform grids and can be used for modelling surfaces as well as volumes. Its simplicity makes it suitable for real-time deformable modelling in Web-based surgical simulators, and we have used it to develop a simple training simulator for tissue cutting.

The present approach works well for objects where there is a simple mapping from the (u,v) mesh to the 3D object – such as the box, cylinder and sphere described here. We are currently working on a further development in which the surface may be defined as a general unstructured mesh: this mesh is triangulated, and the sides of the triangles become the mesh links. This will extend to 3D tetrahedral meshes to give a chainmail-type method for volumetric modelling on unstructured meshes.



```
Vrml is loading ...
1.   Click on the First Button at the bottom left browser...
2.   Click on the plane at the bottom left browser...
3.   Rotate the needle at the bottom left browser ...
4.   Click on the First Button at the bottom left browser ...
5.   Drag the needle to left or right at the bottom left browser ...
 link_mark1::::: 110
 link_mark2::::: 0
 link_mark3::::: 0
```

Figure 5. Surface based deformable modelling and cutting with web-based environment. The browser window at the bottom left is the controller, where the user manipulates the instrument and performing deformation and cutting process. The instrument can be selected at the left bar of the window and rotated as well as inserted. The window at the top displays the results from the controller. The applet viewer at the bottom right is the monitor.

References:

[1] Morten Bro-Nielsen and Stephane Cotin, Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation. Computer Graphics Forum, 15(3) 57-66 (Eurographics'96), 1996.

[2] Sarah F. F. Gibson, 3D chainmail: a fast algorithm for deforming volumetric objects. In Michael Cohen and David Zeltzer, editors, 1997 Symposium on Interactive 3D Graphics, pages 149-154. ACM SIGGRAPH, April 1997. ISBN 0-89791-884-3.

[3] Markus A. Schill, Sarah F. F. Gibson, H. J. Bender, and R. Manner, Biomechanical Simulation of the Vitreous Humor in the Eye Using an Enhanced ChainMail Algorithm. Proceedings Medical Image Computation and Computer Assisted Interventions, MICCAI'98, October, 1998, pp. 679-687.

[4] Sarah F. Frisken-Gibson, Using Linked Volumes to Model Object Collisions, Deformation, Cutting, Carving, and Joining. IEEE Transactions On Visualization And Computer Graphic, Vol. 5. No. 4. 1999.

[5] Ying Li, Ken Brodlie and Nicholas Phillips, Web-based VR Training Simulator for Percutaneous Rhizotomy, in Medicine Meets Virtual Reality 2000, edited by JD Westwood, HM Hoffman, GT Mogel, RA Robb and D Stredney, IOS Press, pp175-181.

[6] Nigel W. John and Nicholas Phillips, Surgical Simulators Using the WWW, in Medicine Meets Virtual Reality 2000, edited by JD Westwood, HM Hoffman, GT Mogel, RA Robb and D Stredney, IOS Press, pp146-152.

[7] Kühnapfel U., Çakmak H. K. and Maass H, Endoscopic Surgery Training using Virtual Reality and deformable Tissue Simulation. Computers & Graphics 24(2000) 671-682, Elsevier (2000)
[8] Morten Bro-Nielson, Finite Element Modeling in Surgery Simulation. Proceedings of the IEEE. Vol. 86. No. 3. March 1998.
[9] Web-Based Surgical Simulators and Medical Education Tools. http://synaptic.mvc.mcc.ac.uk/simulators.html