Workshop on Dynamics Simulation: Methods and Applications. Grenoble France.
Proc. IROS'97 1997 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, pages 52–57

Real-time Finite-elements Simulation of General Visco-elastic Materials for Haptic Presentation

Oliver R. Astley and Vincent Hayward

Department of Electrical Engineering and Center for Intelligent Machines
McGill University
3480 University Street, Montréal, Qc, H3A 2A7, Canada
http://www.cim.mcgill.ca/~{oliver,hayward}/Home.html

Abstract

This paper introduces a methodology to simulate the dynamics of visco-elastic 3-dimensional bodies in real-time using a finite element approach. The method is currently being applied to a haptic simulation of a deformable body incorporating mass, damping, and stiffness. The central idea in this scheme is to reduce the computation required in regions which are to the periphery of area of interaction between the virtual haptic device and the virtual body. This is accomplished by implementing a multi-layer mesh; the top layer, or parent mesh, consisting of a coarse mesh, while child meshes represent sub-regions of the coarse mesh but have a much finer resolution. The effective decoupling of regions enables the system not only to have different resolutions in different regions, but also allows the regions of the mesh to be updated at different frequencies. The key to achieving this is by utilizing equivalent impedances.

1 Introduction

The goal of this research is to provide an environment which not only enables simulation of deformable bodies with respect to mass, damping, and stiffness properties, but also has the potential, or has the ability to achieve the following objectives.

- Simulation of complex three dimensional bodies real-time.
- Implement not only stiffness, but damping and inertia as well, and possibly other constitutive properties.
- Incorporation of the realities of the simulation with respect to redundancies in calculations and also absolve calculations which have a minimal effect on the system.
- Scalability in order to accommodate other tasks, for example virtual cutting of tissue in surgical simulation and other applications.

The advance in computer power with respect to cost has made the notion of the interactive simulation of three dimensional deformable bodies feasible. This is reflected in current research focussed on performing these tasks in real time. However, the numerical complexity of these simulations require that some form of "numerical compression" is applied to achieve reasonable simulation update rates.

Examples of using numerical compression are numerous. Cover *et al.* [7] elect not to use a finite element approach due to its data density and adopts an energy surface instead. Many of the modeling issues in deforming solids have been addressed by Terzopoulos [15, 17] and then applied to finite element systems in [14, 16]. Cotin *et al.* [5, 6] have integrated a finite element system with haptic feedback in real-time. They have used linearity principles and thresholding to reduce on-line computations. As yet their system has not implemented damping and inertia.

These approaches have not considered the need for accuracy enhancement close to the region of interaction between the virtual device and the deformable body. Similarly, in areas to the periphery of the interaction region, less detail and accuracy is required in the simulation, and hence numerical compression opportunities exist.

This paper presents a finite element based method which enables the mesh to be decoupled into high node density regions and low node density regions in a multi-layered mesh. By decoupling the mesh into separate finite element systems, different regions of the mesh can be computed at different rates while simultaneously facilitating the goal of reducing computation in areas to the periphery of direct interaction. As the region of interaction moves with the input device the higher density meshes are activated and deactivated as required. Importantly, most of the overhead for the multi-layered mesh can takes place in pre-processing.

Workshop on Dynamics Simulation: Methods and Applications. Grenoble France.
Proc. IROS'97 1997 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, pages 52–57

2 Constraints and Assumptions

The goal of this research is to emulate deformable bodies and facilitate interaction using a haptic device. It is therefore necessary to comprehend the constraints imposed by the human user in design of the system as well as the available hardware.

2.1 Human Factors

Understanding human bandwidth constraints is essential when designing interactive systems.

- Humans can sense force vibrations well in excess of 300 Hz.[8, 10]
- An update rate in the range of 20-40 Hz is preferred for the visual feedback rendered using computer graphics.
- The maximum human force of motion output bandwidth is 10 Hz although in general the output is in the region of 1 or 2 Hz [2].

Clearly designing a system which solely runs at the highest target frequency of 500Hz would be over designed. (If it was indeed computationally possible.) This indicates that to *efficiently* distribute the computational load in order to service the various human response requirements, a system will have components running at different rates. Clearly at the haptic output, high rates are needed compared to visual update rates.

2.2 Deformable Body Properties

It is clear that most accuracy is required in the region of interaction between the virtual tool and the deformable body. The contact point defines the location of a neighborhood of interest and should be as accurate as it is computationally possible, *i.e.* within the haptic time constraints; yet, to the periphery of this region of interest, neither the eye nor the dynamic simulation calculations require a high accuracy or a fast update rate. Obviously this is an accuracy for computation reduction trade-off. This tradeoff has the potential of reducing the required CPU load tremendously—a necessary factor for real-time interaction.

2.3 General Hardware Constraints

From the outset it was decided not to simply "throw" multiple processors at the real-time dynamic simulation problem. It can be easily shown that any naive implementation will be much too slow. Rather, a reduction of the per cycle processing is sought. The hardware which the following work is implemented on is as follows:

- A single processor R10000 Silicon Graphics IMPACT computer. This is used for the dynamic simulation and graphics display.
- A Pentium processor used for low level control of the haptic device.
- A 7 D.O.F. force feedback hand-controller [11].

Somewhat ironically the algorithm to be presented is also conducive to a multi-processor environment, but as yet, we are using just the one CPU.

3 Adaptive Meshing

Given the notion of representing different regions of the deformable body with varying mesh densities, how can this be achieved using traditional methods? Looking at the field of finite adaptive meshing, the problem is two-fold.

1. The error of the FE approximation must be estimated in order to decide where the mesh must be made finer or, where it can be coarser, based on an a given error bound.
2. The mesh must be regenerated in order to reduce error in the identified regions.

Fortunately step one is not explicitly necessary in the virtual interaction case. This is because the region in which we desire to reduce the error will always be in the region surrounding the point in which device is in contact with the deformable body.

Unfortunately, despite being alleviated of the error estimation, the mesh re-generation procedure requires a powerful mesh generation engine to achieve the desired densities [20]. Further investigation makes it clear that on-line re-meshing is problematic not only with respect to the fact that these procedures are very time consuming, but also with respect to real-time usage, re-meshing times are hard generally indeterminate. A real-time system, by its very nature, **must** be able to guarantee that processing will have an upper-bound on the time it takes to complete. It is apparent, especially in the three dimensional case, that the re-meshing times are difficult to establish due to [1, 3, 9, 12, 13, 19, 18]:

- A search of surrounding nodes must be performed in order to verify density.
- When a new element is added, the new element, as well as the surrounding elements, must be checked for numerical conditioning.
- The mesh must be conforming.
- In the event that ill-conditioned elements are formed, the system must must backtrack to correct for the numerical instability.

Workshop on Dynamics Simulation: Methods and Applications. Grenoble France.
Proc. IROS'97 1997 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, pages 52–57

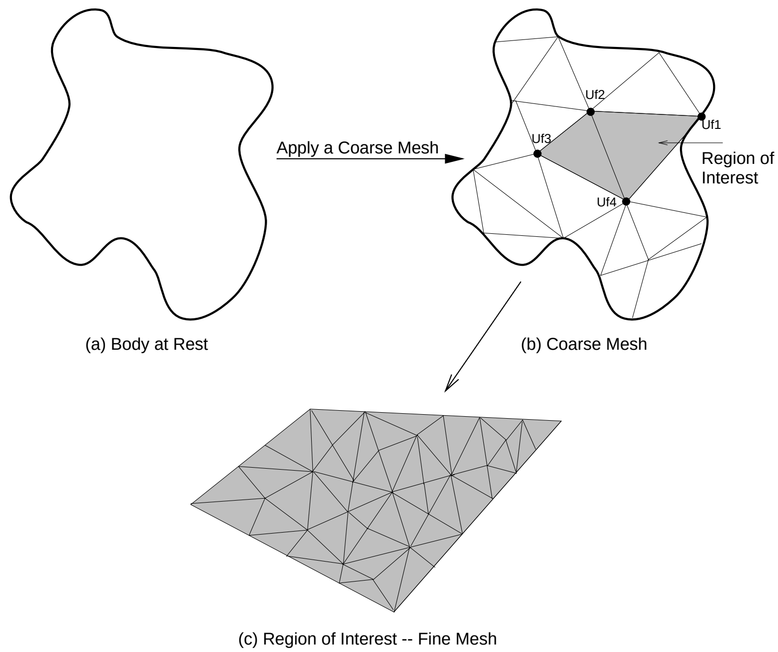

Figure 1: A Deformable 2D Body – Segmenting into Macro and Fine Grids

It is clear that even if a local adaptive re-meshing scheme could be executed at the speeds required, it would not be satisfactory for real-time use due to these often in-determinant factors.

In two dimensions, this approach might seem quite promising, however the jump in complexity in three dimensions surpasses its useful properties. In summary, adaptive meshing techniques do not provide the guarantees necessary for real-time operation. It is clear that if a variable density mesh is desired which will follow the user input, a different approach is required.

Therefore a more "reflex" approach was investigated in which the structure of the original mesh is so defined that given any point from which the mesh is to be refined, a pre-determined refinement will be carried out without calculations. The goal being to eliminate the need for checks on element singularities and other conditions.

4 Introducing the Multi-Phase Mesh

The following meshing approximation is proposed which circumvents the problems found in traditional adaptive meshing systems, and fulfills the goals delineated in the introduction.

In brief, the mesh is made of more than one level. For example a two layer mesh consists of one top layer coarse mesh, and several finer meshes which are subsets of the coarse mesh. At anyone time, one mesh is active from each level. A lower level mesh is physically linked to its higher level mesh parent through equivalent impedances.

By explicitly splitting the mesh into several layers the following objectives are achieved:

- The region of direct interest has a finer mesh, while regions to the periphery are considered only by the coarser mesh. This means that regions of the mesh undergoing less change will be resolved at lower resolutions.
- The two systems, although physically coupled, can be solved independently and at different rates. This allows the fine mesh which is the region of principle interest to be solved at the fastest rate, while the coarser mesh is solved at a slower rate.
- The majority of the workload in the two level adaptation can be done in pre-processing, rather than on-line, this frees up valuable CPU time.
- By creating a mechanism in which a mesh can be plugged into another, the possibility of plugging other functional units into the system is also possible. For example, rather than using an FE system in the finer mesh position, a cutting functional unit could be implemented.

The mechanisms involved in the system are now described.

4.1 Mesh Description

In this section the multi-layer mesh is described using a two layer mesh in two dimensions. The concept is

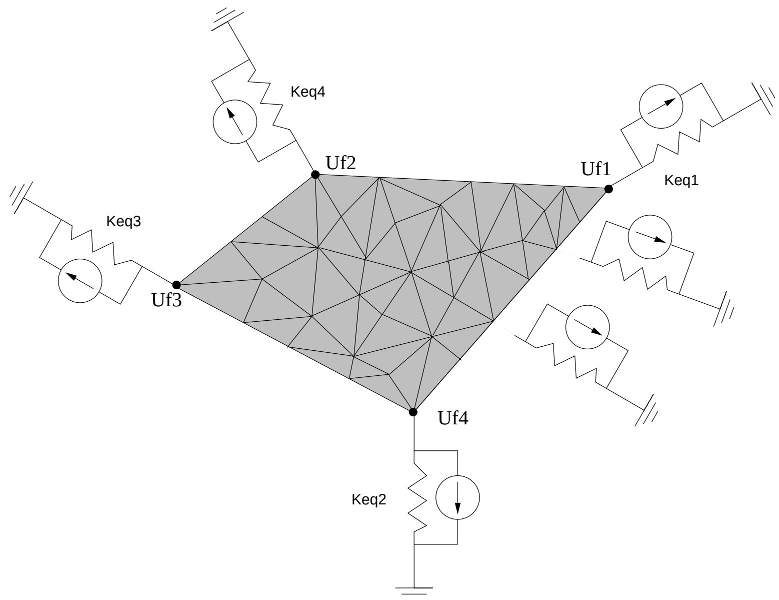

Figure 2: Application of Norton Equivalent to the fine mesh.

scalable to an arbitrary number of layers and to three dimensions.

Figure 1.a shows an arbitrary body at rest. It is segmented into a coarse mesh in the usual manner in Figure 1.b. Finer meshes are then defined based on the general structure of the coarse mesh. For example two coarse elements in Figure 1.b have been chosen to form a fine mesh region. An exploded view of this region is shown in Figure 1.c with the fine meshing complete.

The next issue is to physically join these two regions such that the coarse and fine mesh are physically coupled.

4.2 Coupling the Two Meshes

Using an electrical analogy, the concept for coupling the two meshes can be found in any basic electronic circuit text—that of the Thevenin or Norton equivalent [4].

In network analysis it is frequently desired to find the voltages of just part of a large circuit, the part which is not of direct interest can be replaced by a simpler equivalent. This equivalent consists of a voltage or current source and an impedance as shown in Figure 2.

Using displacement and force as mechanical analogs for voltage and current, the equivalent impedance concept is applied to the two layer mesh as shown in Figure 3.

The fine and coarse mesh share some nodes. These are labeled Uf1, Uf2, Uf3, and Uf4 in the Figures. The task is to find the impedance as seen from these common nodes. The impedance is taken from the mesh surrounding the region of interest at the coarse mesh

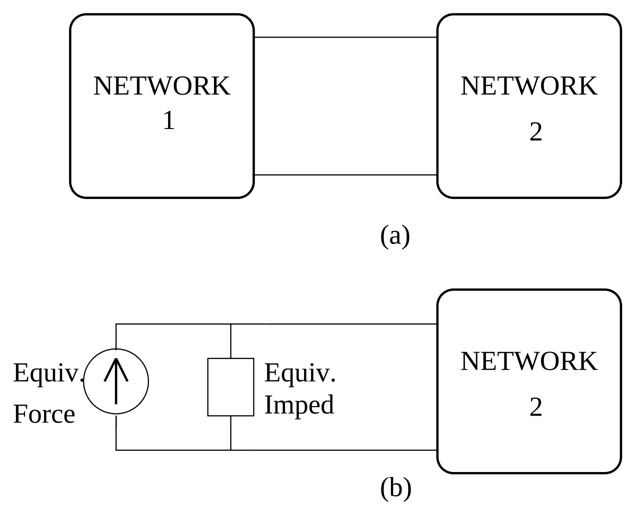

Figure 3: The Norton Equivalent Network Theory

level. It is important hat the equivalent impedance calculation at the coarse level does not include the region of interest (the shaded region in Figure 1) because this region is explicitly modeled in the fine mesh using regular meshing techniques. Once the equivalent impedances are found for each common node, their effects are augmented to the fine (second level) mesh.

The calculation of the equivalent impedance for a given common node i is carried out as follows. Given the finite element system at the coarse (top level) mesh as:

$$\mathbf{K}u = F \tag{1}$$

The contribution of the coarse elements in the region of interest (the region which the fine mesh represents) are subtracted from the system. Giving $\mathbf{K}^*$.

The equivalent impedance at that node can be found by applying an arbitrary force to the node F_i. The system $\mathbf{K}^* u = F_i$ is then solved. The equivalent impedance is simply.

$$K_{equiv} = \frac{F_i}{u_i} \tag{2}$$

This impedance is then augmented to the fine mesh system $\mathbf{K_f} u_f = F_f$.

The multi-layer mesh creation, the equivalent impedances, and the generation of all stiffness, damping, and inertia matrices can be done off-line in preprocessing.

It can be seen from Figure 3 that along the fine-coarse mesh boundary the order of approximation of the fine mesh will be much greater than the coarse mesh as is desired. This causes problems in conformity of elements along the boundary. To eliminate this problem the elements in the fine mesh are constrained to the same order as the coarse elements of which they abut.

4.3 Process System Flow

By decoupling the mesh into a fine and coarse regions it is possible to solve each mesh separately and at different frequencies. At present, the system is configured such that the fine mesh is solved ten times for each time the coarser mesh is solved. Each time the slower mesh has completed its calculations the system checks to see which child fine mesh should be activated. This is based on the position of the haptic device in the virtual world.

The coarse mesh experiences the effects of the interaction between the fine mesh and the virtual device through the displacement of the common nodes shared between two layers.

5 Implementation Results

Currently the system is being implemented using cubic, quadratic, and linear brick elements. Using a 40 element quadratic coarse mesh with a 32 element cubic fine mesh update rates of over 10Hz have been achieved for the fine mesh, while the coarse mesh is running at 1Hz. The simulation implements not only stiffness, but damping and inertia.

Currently we are negotiating several problems with regard to the overall haptic interaction, these are:

- Collision detection between the device and the deformable body.
- Numerical integration stability.
- Generating a simple model to executed at rates of $\approx$ 500Hz for the a smoother force rendering on the haptic device.

6 Conclusions and Future Work

This paper has introduced a method to simulate a deformable body with mass, damping and stiffness. The design methodology is based around three central concepts.

- Regions to the periphery of the point of integration between the device and the deformable body require less detail and numerical accuracy than regions close to the point of interaction.
- The ability to decouple the mesh into different layers based on node density allows different regions of the mesh to be executed at different rates, allowing priority to rates of greater interest.
- In the future, algorithms for cutting virtual tissue may require methods which do not involve finite elements; yet, in regions away from the cut, the mass, damping, and stiffness affects are still pertinent. Using the concept of equivalent stiffnesses, damping, and inertia, affects can be coupled between two very different operational processes. For example, a cutting process and a finite elements process.

What is more, although not an original goal of this research, this method can be seamlessly implemented on several processors for additional speed-up.

We are currently in the process of adding a more complete interface for the haptic device. It is intended to utilize and implement current haptics research within this study. For example perceptual phenomena sometimes known as "haptic illusions" will prove useful in making simulations both more realistic while also affording the opportunity to reduce computational load as well.

7 Acknowledgements

Funding for this research is provided by the project "Haptic Devices for Teleoperation and Virtual Environments" (HMI-6) supported by IRIS (Phase 2), the Institute for Robotics and Intelligent Systems part of Canada's National Centers of Excellence program (NCE). Additional funding is provided by NSERC, the National Science and Engineering Council of Canada, in the form of a postgraduate scholarship for the first author and an operating grant "High performance robotic devices" for the second. The first author also receives funding from a Raychem Cooperation Paul M. Cook International Scholarship.

Workshop on Dynamics Simulation: Methods and Applications. Grenoble France.
Proc. IROS'97 1997 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, pages 52–57

References

[1] M.J. Baines. *Moving Finite Elements.* Oxford Science Publications, 1994.

[2] T. L. Brooks. Telerobotic response requirements. *STX Publication ROB 90-03*, 1990.

[3] J.C. Cavendish, D.A. Field, and W.H. Frey. An approach to automatic three-dimensional finite mesh generation. *International Journal for Numerical Methods in Engineering*, 21:329–347, 1985.

[4] W.K. Chen. *Linear Networks and Systems: Algorithms and Computer Implementations.* World Scientific, 1990.

[5] S. Cotin, H. Delingette, and N. Ayache. Real time volumetric deformable models for surgery simulation. *Visualization in Biomedical Computing*, September 1996.

[6] S. Cotin, H. Delingette, J.M. Clement, V. Tassetti, J. Marescaux, and N. Ayache. Volumetric deformable models for simulation of laparoscopic surgery. In *Computer Assisted Radiology CAR'96*, Paris, France, 1996.

[7] S.A. Cover, N.F. Ezquerra, J.F. O'Brien, R. Rowe, T. Gadaxz, and E. Palm. Interactively deformable models for surgery simulation. *IEEE Computer Graphics & Applications*, 13(6):68–75, November 1993.

[8] P. Fischer, R. Daniel, and K.V. Siva. Specification and design of input devices for teleoperation. *IEEE International Conference on Robotics and Automation*, 1:540–545, 1990.

[9] P. Frey, B. Sarter, and M. Gautherie. Fully automatic mesh generation for 3-d domains based upon voxel sets. *International Journal for Numerical Methods in Engineering*, 37(16):2735–2753, August 1994.

[10] V. Hayward. Toward a seven-axis haptic device. *IROS '95, IEEE/RJS Int. Conference on Intelligent Robots and Systems*, 1995.

[11] V. Hayward, P. Gregorio, O. Astley, S. Greenish, M. Doyon, L. Lessard, J. McDougall, I. Sinclair, S. Boelen, X. Chen, J.-P. Demers, J. Poulin, Benguigui I., N. Almey, B. Makuc, and X. Zhang. Freedom-7: A high fidelity seven axis haptic device with application to surgical training. *DRAFT PAPER: PREPRINTS OF ISER'97.* June 15–18 1997, Barcelona Spain.

[12] Moller O. and P. Hansbro. On advancing front mesh generation in three dimensions. *International Journal for Numerical Methods in Engineering*, 38(21):3551–3569, November 1995.

[13] R. Schneiders and J. Debye. Refining quadrilateral and brick element meshes. In I. Babuska, J.E. Flaherty, W.D. Henshaw, J.E. Hopcroft, J.E. Oliger, and T. Tezduyer, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, volume 75 of *The IMA Volumes in Mathematics and Applications*, pages 53–65. Springer-Verlag, 1995.

[14] McInerney T. and D. Terzopoulos. A finite element model for 3d shape reconstruction and nonrigid motion tracking. *IEEE 4th International Conference on Computer Vision*, pages 518–523, 1993.

[15] D. Terzopoulos and Fleischer K. Modelling inelastic deformations: Viscoelasticity, plasticity, fracture. *Computer Graphics*, 22(4):269–278, August 1988.

[16] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[17] D. Terzopoulos, J. Platt, A. Barr, and Fleischer K. Elastically deformable models. *Computer Graphics*, 21(4):205–214, July 1987.

[18] F.G. Uler and O.A. Mohammed. A 3-d finite element mesh generator for complex volumes. *IEEE Transactions on Magnetics*, 30(5), September 1994.

[19] N.P. Weatherill and O. Hassan. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering*, 37(12):2005–2039, June 1994.

[20] O.C. Zienkiewicz and J.Z. Zhu. Adaptivity and mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):783–810, September 1991.