LEM - An approach for real time physically based soft tissue simulation *

Ivan F. COSTA
SHARP/GRAVIR †
INRIA Rhône-Alpes
Ivan-Ferreira.Costa@inrialpes.fr

Remis BALANIUK
CATSS ‡
Stanford University
remis@robotics.stanford.edu

Abstract

This paper presents LEM - Long Elements Method, a new method for physically based simulation of deformable objects, suitable for real time animation and virtual environment interaction. The approach implements a static solution for elastic global deformations of objects filled with fluid based on the Pascal’s principle and volume conservation. The volumes are discretized in long elements, defining meshes one order of magnitude smaller than tetrahedral or cubic meshes. The physics of the objects are modeled using bulk variables: pressure, density, volume and stress. No pre-calculations or condensations are needed. The approach is particularly interesting for soft tissue real time simulation and for graphic and haptic rendering.

1 Introduction

Physically based simulation of deformable objects is a key challenge in Virtual Reality (VR). Deformable object modeling has been studied in computer graphics and animation for three decades. In engineering, computer-aided design, and entertainment, deformable models are used to create and edit complex surfaces and solids. In image analysis, deformable models have been used to segment images and to fit curved surfaces to noisy image data. As computational power increases deformable models start due starting to be used in VR as well. Researchers from new fields such as human tissue modeling, interactive character animation and surgical simulation, are working to extend the simulation of deformable models to a virtual interactive reality. They essentially aim at physically based simulations of complex deformable objects, and enhanced multi-modal interactivity: graphic and haptic interfaces to manipulate and to change the topology in real time.

Figure 1: Soft-tissue touched by a rigid probe

Modeling of deformable objects is indeed a nontrivial task. A number of methods have been proposed ranging from non-physical methods, where individual or groups of control points or shape parameters are manually adjusted to shape editing and design, to methods based on continuum mechanics, which account for material properties and internal and external forces on object deformation [1]. These methods can also be classified by the degree of interaction they allow and their accuracy. The usefulness of a simulation method in VR is defined by these two conflicting demands. Interactive models must have low latency and are based in some internal structure suitable for topological changes in real time. Accurate models have the precision of their results limited only by some scale factor and the computational power. Unfortunately these classifications go frequently together. Interactive methods, as the mass-spring [2], are mainly non-physical and inaccurate [6]. Accurate physically based methods, as the Finite Elements (FEM), are typically simulated off-line, and the changes on these methods to achieve real time performance normally compromise their accuracy or their interactivity. In FEM for
instance, pre-calculations [3] and matrix condensation techniques [4] enhance performance but forbid further topological changes on the models. Simulation methods are also classified regarding additional aspects. Deformations can be dynamic or static, global or local. Models can be elastic or visco-elastic, linear or non-linear, surface or volumetric. Collision detection and handling can be implemented using different approaches. All these aspects define the way a simulation will behave, its accuracy and performance, and the kind of applications for which it will be well suited.

The method proposed in this paper was conceived for soft tissue real time simulation, particularly for surgical simulation. The priorities in this kind of application are: unrestricted multi-modal interactivity, including interactive topological changes (cutting, suturing, removing material, etc.), physically based behavior, volumetric modeling (homogeneous and non-homogeneous materials) and scalability (high accuracy when needed). The choices made to define the method were driven by these priorities.

The approach is based on a static solution for elastic deformations of objects filled with incompressible fluid, which is a good approximation for biological tissues. The volumes are discretised in a set of Long Elements (LE), and an equilibrium equation is defined for each element using bulk variables. The set of static equations, plus Pascal’s principle and the volume conservation, are used to define a system that is solved to find the object deformations and forces. Global and physically consistent deformations are obtained (Fig. 1).

For a survey of deformable modeling in computer graphics the reader is referred to [1].

Others recent methods proposed are the ”Geometric Nonlinear finite element method” [6], the ”Boundary Element Method” [3] and some medical simulators [7], [8], [9], [10], [11].

2 Method Formulation

In this section we introduce the physical principles used to derive our approach.

2.1 Pressure and Stress

Consider the long elastic element illustrated in figure 2. The force $F$ per unit of area $A$ is defined as pressure:

$$ P = F/A. $$  \hspace{1cm} (1) $$

However the force per area unit producing the deformation is also the stress. For small applied forces, the stress $s$ in a material is usually linearly related to its deformation (its change in length in our long elastic object). Defining elasticity $E$ as the variable relating stress and the fractional change in length: $\Delta L/L$, it is possible to write:

$$ s = E \Delta L/L. $$  \hspace{1cm} (2) $$

Since the stress is related to the fractional change in length, the force can be related to the elongation $\Delta L$ in the well known form:

$$ F = K \Delta L $$ \hspace{1cm} (3) $$

where

$$ K = AE/L. $$  \hspace{1cm} (4) $$

Note that $K$ is not constant, but it depends on the length $L$.

2.2 Static Solution

The static condition states that the forces, or pressures, in one sense have a correspondent of the same magnitude in the contrary sense on each point of the surface of the object, or:

$$ P_{int} = P_{ext}. \hspace{1cm} (5) $$

The external pressure $P_{ext}$ on the surface is affected by the atmospheric pressure and by the stress when an elongation exists, so:

$$ P_{ext} = P_{atm} + E\Delta L/L. \hspace{1cm} (6) $$

The surface tension also affects the external pressure, as described further in section 2.4.

Considering that the object is filled by fluid, the internal pressure ($P_{int}$) is formed by the pressure of the fluid (without gravity) and the effect of the gravity acceleration ($g$), so:

$$ P_{int} = P_{fluid} + dg/h. \hspace{1cm} (7) $$

where $h$ is the distance between the upper part of the fluid and the point where the pressure is calculated.
From the last three equations, a continuous equation can be obtained as:

\[ E \Delta L/L - \Delta P = dgh \]  

(8)

where \( \Delta P = P_{\text{fluid}} - P_{\text{atm}} \).

Another external pressure to be considered comes from contacts between the object and its environment. At the points on the object surface, where are some external contacts, a term is added to the right side of equation 6. To obey the action-reaction law, the force applied to the external contact and to the object must to have the same magnitude. It means that the external pressure applied by the contact must be equal to \( \Delta P \). The elongation \( \Delta L \) is defined by the penetration of the contact in order to make the surface follow the contact position \( (y) \). With these considerations, the equation 8 can be rewritten for the elements where there is external contact as:

\[ \Delta L = y. \]  

(9)

### 2.3 Long Elements

To simulate a deformable object we propose a discretisation of its volume in a set of long elements (Fig. 2). The idea is to fill the volume with long elements, to define equilibrium equations for each element based on the stated principles and to add global constraints in order to obtain a global physical behavior. Different meshing strategies can be conceived to fill the objects.

A long element can be compared to a spring fixed in one extremity and having the other extremity attached to a point in the movable object surface. These springs are relaxed when the solid is not touched (not deformed). The spring constant \( K_i \) depends on its length (eq. 4). The force due to each spring has magnitude given by equation 3 in the direction of the spring. A long element does not occupy real space and has no mass. The real space inside the solid is occupied by some incompressible fluid of density \( d \).

Applying the continuous equations (eqs. 8 and 9) for each of these elements we obtain:

\[ E_i \Delta L_i/L_i - \Delta P_i = d_i g_i \, h_i \]  

(10)

for the untouched elements.

\[ \Delta L_i = y_i \]  

(11)

for the touched elements.

To make the connection between the elements two border conditions are applied:

1. Pascal’s principle says that an external pressure applied to a fluid confined within a closed container is transmitted undiminished throughout the entire fluid. Mathematically:

\[ \Delta P_i = \Delta P_j \text{ for any } i \text{ and } j. \]  

(12)

The first equation of this section (eq. 10) can then be written without the index \( i \) in the term \( \Delta P_i \).

2. The fluid is considered incompressible. It means that the volume conservation must be guaranteed when there is some external contact to the object. The volume dislocated by the contact will cause the dislocation of the entire surface, or in other words, the variation of volume due to the elements touched by the contact have to be equal to the sum of the volume created by the dislocation of all untouched elements to ensure the volume conservation:

\[ \sum_{i=1}^{N} A_i \Delta L_i = 0 \]  

(13)

where \( N \) is the total number of elements. Note that this equation ensures the conservation of volume defined by the set of elements, implying the conservation of the object volume. It does not mean that the volume of the elements correspond to the object volume. The set of elements can fill only part of the object volume and the same part of the volume can be occupied by more than one element.

Note the difference at this point between LEM and FEM. To re-establish the continuum, or object, the FEM discrete equations are subject to constraints at specific points on the surface of the elements, called nodes. In general, the continuity between the elements is achieved making the position of the nodes of adjacent elements the same. In our method there is no need to define nodes and the object is re-established by the border conditions (Pascal principle and volume conservation).

### 2.4 Surface Tension

Deformations cause a change in the object surface area, even if its volume is kept constant. This change creates forces on the surface generating a surface tension. One of the effects of these forces in a deformable object is to make the contours of the surface smoother.
To reproduce these forces due to the change in the surface area we include linear elastic connections between neighbor elements coupling their changes in length. A number of terms will be added to the right side of the equation 6 corresponding to the neighborhood considered around element. These terms are of the form $P = F'A = kxA$, where $x$ is the difference between the deformations of a element and its neighbor and $k$ is a local spring constant. For a given element $i$ the term relating its deformation to the deformation of its neighbor $j$ is:

$$k_j(\Delta L_i - \Delta L_j)A_i \quad (14)$$

3 Mathematical Solution

Equations 10, 12 and 14 define the final equation for the untouched elements (considering 4 neighbors):

$$(E_i/L_i + 4kA)\Delta L_i - kA(\Delta L_{i-1} + \Delta L_{i+1}) = \Delta L_i + \Delta L_{i+1} - \Delta L_{i-1} - \Delta P = d_i g_i h_i \quad (15)$$

where $k$ and $A$ were done constant for all elements to make easier the notation.

The untouched elements (equation 15) plus the elements in contact with the environment (equation 9) define a set of $N$ equations, where $N$ is the number of elements used to fill the object. Adding the equation of volume conservation (eq. 13) we have $N + 1$ equations and $N + 1$ unknowns: the pressure ($\Delta P$) and the deformation of each element ($\Delta L_i$ for $i = 1$ to $N$).

These $N + 1$ equations can be written as a problem of the type $Ax = B$.

Since $A$ depends on the environment and its contacts with the object, it is not possible to invert $A$ ahead in time. Nor it is realistic to invert and store all possible matrices ahead in time. However, in our approach $A$ is a sparse matrix and the system can be solved using fast standard numerical methods [12].

4 Method Implementation

The described method was used to implement a generic soft tissue VR simulator. The simulator was implemented in C++ in a Windows NT platform. The system main modules are:

1. **model definition**: the geometry and the physics of the objects to be simulated are defined using available information. This information can be derived from segmented real data or from intermediate models. The volumes are meshed to define the long elements.

2. **simulation loop**: solves the system equations to obtain the deformed shape of the objects.

3. **rendering loops**: graphic and haptic loops enable the interaction with the objects.

This first prototype simulates deformations of a compliant object contacted by a rigid probe.

4.1 System organization

It has been clearly shown that is necessary to decouple simulation, graphic and haptic loops in order to maintain reasonable display update rates [3]. The system is organized around three decoupled main loops, executed concurrently in different processing units (threads, process and/or machines). The first loop simulates the deformations, the second renders the graphics and the third renders the haptics. The main loops share the data structure containing the long elements.

The simulation and rendering modules are organized in a client-server architecture. A simulation server solves the static deformation problem and the rendering clients interact with the simulation informing contacts and receiving the updated surface. This organization permits multi rendering and multi haptic interaction in a shared virtual environment.

Figure 3: Soft cylinder modeled using one horizontal LE mesh

4.2 Model definition

An appropriate model definition is necessary to obtain a desired simulated behavior. The following model features need to be defined:

1. **geometry**:
   (a) number of axis and faces being meshed
   (b) number of elements per axis and per face
2. physics:

(a) for each element, or set of elements: density and elasticity
(b) for the surface, locally or globally: friction and superficial tension

A reference point and a Cartesian local reference frame are defined. The object is then discretized in a number of Cartesian meshes, each mesh containing long elements parallel to one axis of the reference frame. A Cartesian mesh defines a grid of parallel elements crossing the object. Each element starts in a point of the surface and crosses the volume until the end of the material, defining a line segment parallel to one of the reference frame axis. Each model will use up to 6 meshes (3 directions and 2 senses). The total number of elements is about $O(mn^2)$ where $m$ is the number of meshes and $n$ is the average number of elements in each side of the grid. The long element deforms only in one direction. Its extremity attached to the surface moves back or forth. The choice on the number of meshes depends on the type of the object being modeled, and its situation. The figures 3 and 3 illustrate some meshes. In figure 3 a cylinder is filled by a horizontal LE mesh. The first two images show a cross section of the mesh at the contact point. The first image shows the mesh at rest and the second shows the deformed mesh during a contact. The last

1 A similar cubic mesh in a FEM model would have $O(n^3)$ elements, which leads to a much larger system of equations.

image shows the deformed cylinder. In figure 4 the same cylinder is filled by two orthogonal meshes. The first two cross sections show each mesh separately and the last one shows the superposed meshes.

The physics of the object can be defined locally, element-by-element, or once for the entire object. The parameters that define the deformations are the density, elasticity and superficial tension. To define these values one can use measured data and/or material properties [13].

Non-homogeneous materials can be modeled. Rigid and different deformable materials can be mixed in the same volume as illustrated in the figure 5.

Figure 4: Soft cylinder modeled using two orthogonal LE meshes

Figure 5: Non homogeneous materials: soft object with a rigid nucleus

4.3 Main loops

simulation loop The iterative biconjugate gradient method [12] is used to solve the system of equations defined in section 3. The static equations system does not demand any particular concern about time steps, stiffness or stability. The matrix is dynamically defined and the system $Ax = B$ can be rapidly solved.

The $A$ matrix is defined by two sets of equations: the touched elements for which there is a restriction imposed on their lengths (equation 9) and the untouched elements (equation 15). A collision detection algorithm must to inform the touched elements and the penetration. The vector $B$ contains the penetration for the touched elements and the pressure for the untouched ones.

The solution ($x$) is the surface deformation, defined by a set of length differences in each element, and the difference in pressure.

Graphic loop OpenGL and GLUT are used to render the 3D volumes. There is no explicit geometric model of the object surface. In order to draw the object we use vertices directly derived from the long elements extremities and polygons defined between neighbor elements.
**Haptic loop** The LE representation of a volume is excellent for haptic rendering. The one point collision detection between the haptic probe position and the volume can be easily done using directly the LE cartesian meshes. Each mesh defines a grid, or a space filling map, and the collision detection in one mesh consists in checking the grid position corresponding to the probe position to see if the probe is penetrating a LE. Each mesh being parallel to one axis of the reference frame, the force feedback estimation is naturally decomposed. Each component of the force vector is independently estimated using the corresponding LE mesh.

During a collision between the haptic probe and the volume the probe position is used to define which elements are been touched and the penetration. To estimate the deformation caused by the collision on the object equation (11) is used for the touched elements and the penetration determines $y$.

During the collision two forces are been applied by the object to the haptic probe: a force applied by the touched elements (eqs. 3 and 4) on the direction of the element and a force applied by the fluid inside the object (eq. 16). Multiplying both sides of equation 8 by the contact area $A_c$ and comparing to equations 3 and 4 we obtain:

$$F = \Delta P A_c + d g h A_c \quad (16)$$

This force is perpendicular to the object surface and depends on the internal pressure, not on the penetration.

### 4.4 Results

In a standard dual 700 MHz PC one iteration of the simulation loop takes about 0.05 seconds for a 600 elements mesh. Note that a given discretisation using long elements gives far more precise deformations than the same number of tetrahedra or cubes in a standard mesh. A full 3D soft cube meshed in 600 long elements corresponds to 1000 elements in a cubic mesh. Restricting the number of deformable faces to 4 (two rigid faces) the same quality of deformation can be obtained with 400 long elements. For instance, the soft cube in fig. 1 was defined with 4 meshes (4 deformable faces) and 400 elements ($4 \times 10^2$). The haptic interface was implemented using a PHANTOM haptic device (http://www.sensable.com). See figs. 1 and 6 for some examples of deformation.

The global deformations are physically consistent and important phenomena such as the movement of all parts of the solid due to the preservation of volume are automatically produced.

### 5 Future directions

The method presented in this paper is intended for being part of a more general VR simulator. The method solves the deformation problem in an elegant and simple way, independently of other simulation aspects as movements (translation, rotation, dynamics integration, etc), topology changes and collision handling. To obtain a generic simulator the LE method must be integrated to methods suitable to handle these aspects.

We intend to implement a surgical interface based on medical procedures (cutting, suturing, removing material, etc). The method is particularly well suited to topological changes such as cutting, because it intrinsically preserves volume.

The method can also be useful to validate and to adjust estimated biomechanical parameters in biologic tissues. Experimental data, as in [8], can be used to validate simulation results.

### 6 Conclusions

Utilizing three meshes of long elastic elements (each aligned with a Cartesian axis) we have been able to physically model elastic deformations in a way that preserves volume, permits realtime topology changes and is rapidly computable. The discretisation adopted by the method has two main advantages: the number of elements used to fill an object is one order of magnitude less than in a discretisation based on tetrahedral or cubic elements; the graphic and the haptic feedback can be directly derived from the elements, and no intermediate geometric representation is needed. The use of static instead of PDE equations avoids all the problems concerning numerical integration, ensuring stability for the simulation. No pre-calculations or condensations are used, in order to enable real time topology changes.
7 Acknowledgments

The authors would like to acknowledge the Cisco Corporation and SunMicrosystems for their support of this research. We would also like to thank Dr. Kenneth Salisbury, Dr. Thomas Krummel and the CATSS Laboratory at the Stanford University Medical School Department of Surgery; Christian Laugier and the INRIA; and the Catholic University of Brasilia for providing support, encouragement, and facilities. Finally, we would like to acknowledge Diego d’Aulignac for his advices on the implementation.

References


