Tissue Modeling for Surgery Simulation

Jeremy Kuzub
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada
jkuzub@sce.carleton.ca

Abstract – Surgery Simulation techniques are emerging as a viable training and rehearsal tool. This paper describes the current state of tissue modeling workflow and implementations and describes a tissue model implementation using agent-based framework for parallel computation.

Index Terms – surgery simulation, tissue modeling, agent-based computation, tissue characterization.

I. SURGERY SIMULATION

Role of Simulation
Simulation systems have been used throughout history whenever the expense or consequences of mistakes on the trainee’s part were prohibitive. An especially successful application of simulation has been in pilot training, where both expense and consequences of mistakes is extreme. Electromechanical and later computerized simulators were able to meet an increasing breadth and depth of simulated tasks and scenarios, with a notable large jump in capability associated with computer-generated visuals. While some computational requirements of flight simulation were within the contemporary state-of-the-art, new techniques were also developed to the requirements of the simulations. The success of this co-development of simulator capability and the task breadth has ultimately led to pilots’ flight simulator hours being interchangeable with actual flight hours.

Potential Role of Simulation in Surgical Training
Surgical simulation is in the early stages of a similar process. The need for simulation is driven by several factors. First, there is a lack of surgical analogues to a live human. Animal and cadaver alternatives are subject to ethical considerations and supply limitations. Neither of these accurately simulates both the tissue behavior and the anatomical aspects necessary for high-fidelity training. Second, increasingly complex surgical procedures, techniques, and tools have increased required the breadth and depth of the surgeon’s skill set. The scope of surgery changed and surgical procedures have developed to include minimally invasive procedures and microsurgery which are non-intuitive because the surgeon is decoupled from direct haptic and visual feedback. Third, population demographic changes have increased demand for surgeons and procedures which has in turn created a need for faster more effective training and surgeons who can perform a given procedure more efficiently. Simulations can address the demand for this training.

II. CRITERIA FOR AN EFFECTIVE SIMULATOR

The effectiveness of a surgery simulator is ultimately defined by its effect on a surgeon’s performance in real surgery. The phenomena of negative training, in which incorrect experience is developed from inaccurate simulation, is a potential pitfall for any interactive system, and must be guarded against with verification and a careful design process.

Figure 1 shows the interaction between components of a surgery simulator. Components include geometrically accurate anatomical models, mechanically validated tissue models, a real-time simulation engine, a visual interface, and a surgical instrument interface.

Geometrically Accurate Models
These can be derived from MRI, CT, Ultrasound and other sources. It has been shown that anatomical structures can be reconstructed in 3-dimensions from this 2-dimensional sliced data and subsequently transformed into geometry that allows dynamic interaction. [2][5]. The final geometry and the constituent atomic components, such as voxels [11] and tetrahedra [1] affect the dynamics model that can be used to create realistic tissue behavior.

Realistic Tissue Dynamics
The major computational and modeling challenge in surgical simulation is the dynamics of human tissue. Human soft tissues exhibit nonlinear characteristics when deformed
which follow an approximately visco-elastic behavior [9]. Physical characteristics that would require modeling including elasticity, stiffness, tensile strength, and plastic deformation, which in turn vary with the state of the tissue (affected by temperature extremes, disease or loss of blood-flow). In addition, the behavior of tissue when cut or torn must be taken into account for a practical simulation, where tissue manipulation is the central activity for most surgeries. Finally, homogenous tissues do not exist in isolation, they are interconnected and the heterogeneous structure and connection dynamics must be modeled.

**Haptic Feedback**
Haptic feedback provides support and confidence information to the surgeon. It has been shown that without haptic feedback, subjects tend to apply significantly more force to tissues during surgical task [22]. A haptic update rate of between 300Hz and 1000Hz has been shown to be necessary to provide an immersive experience [6] [7]. This feedback must be mapped through the specific instrument used by the surgeon factor in loads presented by passing the instrument into the work area [9].

**Visual Feedback**
Visual feedback provides the single highest bandwidth information channel for surgeons. Surgery can be performed without haptic feedback, but cannot be efficiently performed without visual feedback. A higher resolution display image is preferable to clearly show texture cues, though this can become computationally expensive.

For real-time applications visual feedback must be refreshed at a rate of 20Hz to 40Hz [6]. If a stereoscopic visualization with depth information is desired, this rate effectively doubles as a virtual camera view for each eye must be rendered.

Resolution of visual feedback can vary depending on application. Since many non-invasive or microsurgery techniques currently use video monitors, it is necessary for the simulation to match the resolution and refresh characteristic of these displays in order to simulate the entire surgical workflow.

**Feedback Coupling**
Disorientation can occur when visual and haptic cues are unsynchronized and it is preferable to have no haptic feedback rather than out of synch feedback. In addition, the actions of the surgeon must be seen to be effecting the simulation with minimal delay to assure real-time performance. This roundtrip delay should not be confused with the actual update rate for each feedback channel.

**Computational Limitations**
A key enabling technology for surgical simulations has been the geometric improvement in computational power that is approximated by Moore’s Law. The calculations required to simulate tissue dynamics in real-time are significant and as of this writing specialized hardware beyond a standard ‘desktop’ computer would be required to simulate non-linear visco-elastic models of sufficient spatial resolution[4] [6] [18] [23].

**Empirical Limitations**
Understanding of human tissue behavior for simulation applications has recently become and area of significant research activity, however there are challenges to this research. The most readily apparent issue is the inability to test dynamics on live subjects. Surgical tools are generally not equipped with force sensors and other data gathering systems that would allow a database of tissue behavior to be built up during routine procedures and mined for information.

Although measurement of homogenous tissue mechanical characteristic is well understood, the practical and ethical considerations in obtaining suitable samples would require a methodology to be developed that effectively logs data from routine surgical forces and correlates then with the type of tissue being manipulated.

Human cadaver sources suffer from similar inconsistencies with in vivo tissue measurement, as post mortem changes, in tissue consistency are pronounced. Animal and non-living analogues suffer from fidelity issues. Although animal and specifically mammal tissues are made from the same basic building blocks as human tissue, specific tissues may not have the same mechanical characteristics. [9]

Finally, tissues are typically heterogeneous in structure and these composite structures have different mechanical properties from the homogenous structure. The ability to build a database with enough information to allow differentiation of tissue contributions to overall measurements would be beneficial. For example, measurements of enough cuts through fatty tissue, would average out influences from connective tissue. Combining video record of a surgery with force instrument data could potentially allow manual categorization of data based on specific tissue types.

**IV. Overview of Computational Methods**
Modelling tissue for real-time simulation is a balance between computational efficiency and fidelity. Methods to date typically subdivide the tissue geometry in interconnected units through which dynamic interactions are numerically calculated in discrete time steps. This approach allows the application of a generalized dynamics model many geometries and tissue model coefficients.

III. Current Limitations
Spring-Mass Model

This model uses Newtonian dynamics of a mesh composed of masses ("cells") interconnected by linear springs. The equations of motion for any given nodes are given by:

\[
A_{cell} = \sum k_{connection} \times D_{neighbour} / M_{cell}
\]

(1)

Where the acceleration of a given cell is the sum of vector forces from neighbouring cells divided by the given cell’s mass. The force vector from a given neighbour is linearly dependent on the distance vector \(D_{neighbour}\) multiplied by a linear spring constant \(k_{connection}\). This was one of the first dynamic models used to for tissue applications.

This spring-mass model does not reflect visco-elastic behaviour, and does not model nonlinear elastic coefficients, which are viewed as a realistic model of tissue dynamics [8]. However, it is straightforward to implement as a numeric integration algorithm. A lack of damping can create unstable behaviour especially when numeric integration errors accumulate during a simulation [18]. This model also has extensive computational requirements for large cell arrays, especially in highly interconnected models.

Instabilities can result from solving the spring-mass models using explicit integration methods such as the Euler method. This numeric integration method is straightforward in its implementation since acceleration, velocity and position values are recalculated each simulated time step based on their previous values. However, this can effectively lead to accumulated errors and ‘model blow-up’, where an error in acceleration will continuously increase the velocity of a cell without bound.

It is shown in Section V of this paper that a similar model with damping a parameter and an intercellular collision model can produce tissue-like behaviours.

A more computationally complex but stable method for determining integration results (i.e. solving ordinary differential equations) is implicit integration. This is also referred to as the Finite Element Method in its specific quasi-static implementation. This is generally applied to tetrahedron-based geometries.

\[ [K][X]=[F] \]

(2)

In Equation (2), the matrix of force vectors \([F]\) is the external forces acting on the tissue. The matrix \([K]\) is the stiffness matrix of the tissue model, effectively the spring coefficients between all cells in the tissue geometry. The displacement matrix \([X]\) is the unknown positional deformation of the tissue geometry needed to solve this equation. Because each cell’s position is being solved directly from force and tissue geometry relations, the integration process is implicit.

Methods of solving this system have been explored in the interest of achieving high computational efficiency and real-time performance [1] [8]. A common method in tissue simulation involves using linear tissue dynamics so that a superposition of precomputed “basic” deformations can be applied during the simulation. A disadvantage of this optimization method is that these precomputations become invalid if the tissue is cut, i.e. the geometry is changed in ways other than deformation. This model is also a simplification of the full equation of motion, and is termed “quasi-static” because cell mass and velocity terms are not used in (2).

Methods of higher fidelity dynamics have been explored using different numerical integration techniques such as Verlet Integration and preservation of element volumes and surface tensions to better model fluidic behaviour of cells [17].

Other approaches have replaced tetrahedral primitives and their strict interconnections with volumetric pixel models, or voxel models. These models approach the geometry of the tissue as a collection of volumes whose interactions are defined by an inter-voxel collision and intra-voxel deformation models. These models can be generated directly from MRI and CT scan data using a marching-cubes algorithm [11]. Since the method of solving voxel-based system interaction is based on collision checking rather than iterative implicit integrator methods like FEM, real-time performance can be accurately predicted [7]. Voxel-based systems can also provide a framework for agent-based programming techniques, in which each voxel independently runs a simple dynamics algorithm and interacts only with nearest neighbours; the collective dynamics of the whole tissue are an ‘emergent’ effect of voxel interactions [6]. A disadvantage of Voxel modelling is the inherently low resolution of voxel models derived from MRI and CT data. Generally haptic feedback must be smoothed to remove any stair-step effect from running a tool over a comparatively low-resolution surface with surface discontinuities (i.e. many small cubes or spheres). Simulation of cutting operations also presents challenges in voxel-based models since voxels must either be removed, resulting in a loss of model mass, or subdivided, which negates many of the computational advantages in the dynamics of a of a uniform 3D voxel grid.

V. A TISSUE SIMULATION IMPLEMENTATION

Tissues Simulation Implementation

As part of this project, an approach to tissue simulation was implemented by the author as a web-based interactive application. An interactive demonstration can be found at http://www.jufaintermedia.com/samples/surgery/
This simulation uses an agent-based approach to tissue modelling in which each ‘cell’ element of the tissue runs its own independent dynamics simulation code. By taking a cell-level approach to dynamics, tissue-level tissue visco-elastic characteristics emerge when enough cells are in interconnections. This approach is potentially scalable to massively parallel processing hardware because calculations are effectively independent.

**Key Features of the Tissue Simulation**

This simulation is a proof of concept and an exploratory computational model. Real-time performance to allow visualization of several hundred cells in a 2D matrix was desirable. To ascertain the algorithm’s behaviour in arbitrary cell configurations, the need for geometric constraints (i.e. a grid or strict tetrahedron interconnections) was circumvented by a fast ‘neighbour-checking’ algorithm at start-up. This essentially allows drag-and-drop design of a tissue.

Cell dynamics are modelled in two parts: an intercellular visco-elastic simulation with spring-mass damper modelling and numerical integration, and an intercellular elastic collision model with energy loss and rebound vector calculation.

The ability to create heterogenous tissue structures using cells with varying characteristics is desirable and tissues are created by a neighbour-checking algorithm eliminate any need for neighbouring cells to have similar dynamics.

Algorithms for both pulling and cutting function interactions are implemented.

Behavioural code is contained within each cell object allowing for implementation on parallel computer architectures – this is an agent-based approach to emergent tissue dynamics which allows for parallel processing and scalability.

**Agent-Cell State Model**

Each individual cell tracks the following internal state variables:

1) Home position (i.e. a connection to an underlying plane)
2) Constant: spring constant on return force to home position
3) Instantaneous acceleration, velocity, and position.
4) Instantaneous external force vector
5) Pending external force vector (prevents race conditions)
6) List of four nearest neighbour cells
7) Constant: cell mass
8) Constant: spring constant for neighbour-cell connections
9) Constant: velocity damping for neighbour-cell connections
10) Constant: elastic collision efficiency

During a single processing cycle, each cell processes the following information in parallel:

1) Acceleration from external forces and cell mass
2) velocity from numeric integration of acceleration and damping factor
3) position from numeric integration of velocity
4) collision detection and collision force transfer
5) force transfer to immediate neighbours via spring connections

Determining Neighbours
A fast configuration stage during the first few cycles of simulation establishes neighbour connections so that tissue architectures made using a 'drag and drop' editor using heterogenous cells will almost immediately begin interacting. This stage does not require recalculation of the entire tissue connection structure as new cells are dynamically added, allowing for 'sewing' and 'gluing' simulation as well as cutting.

Because a pending force state variable separates the forces acting in the current cycle from those imparted for the next cycle contributes to the parallelism of this algorithm, thereby providing simple scalability with hardware parallelism.

Cell Dynamics
Equations (3) through (7) are calculated sequentially each time step of the simulation for each cell. This is a numerical integration procedure and can be implemented with integer or floating point arithmetic.

\[ F_{\text{pending}} = \sum k_{\text{connection}} \times D_{\text{neighbour}} \]  \hspace{1cm} (3)
\[ F = F_{\text{pending}} + k \times D_{\text{origin}} \]  \hspace{1cm} (4)
\[ A_{\text{cell}} = A_{\text{cell}} + f / m_{\text{cell}} \]  \hspace{1cm} (5)
\[ V_{\text{cell}} = V_{\text{cell}} + A_{\text{cell}} - b \times V_{\text{cell}} \]  \hspace{1cm} (6)
\[ P_{\text{cell}} = P_{\text{cell}} + V_{\text{cell}} \]  \hspace{1cm} (7)

Although this model uses an explicit numeric integration scheme, out-of-sync cell position errors are addressed by the use of a asynchronous variable \( F_{\text{pending}} \) that stores summed force vectors that are imparted by pulling or collision actions of neighbouring cells. This effectively allows all cells to move synchronously in a single time step, even though calculations are carried out sequentially on a single processor.

The implementation of these dynamics are in two dimensions for computational efficiency and proof-of-concept, however, there are no new concepts required to move to three dimensions beyond increasing vector sizes from two to three elements.

Collisions
Equations (8) and (9) determine the dynamics of a cell collision.

\[ F_{\text{collision}} = \frac{1}{2} k_{\text{collision}} \times m_{\text{cell}} V_{\text{cell}}^2 \]  \hspace{1cm} (8)
\[ F_{\text{pending}} = F_{\text{pending}} + F_{\text{collision}} \]  \hspace{1cm} (9)

This is an asynchronous event that takes place in zero time in the simulation. The collision function does not need to be calculated for cutting and pulling functions, but assists in intrusion operations such as needle puncture. In order to communicate the asynchronous collision force, the new force vector is added to the \( F_{\text{pending}} \) variable for each cell, which is factored into cell dynamic at the next time step by Equation (4).

Simulated Tool Interactions
Cutting and push-pull functions are implemented to demonstrate the emergent properties of this architecture. Push-pull allows the user to grab and cell and translate it. Any cell being manipulated in this way propagates force to neighbouring cells but does not update its own acceleration and force values except for collision. Resulting effects on the tissue are calculated in real-time. Cutting is implemented by removing intercellular connections at the cut point. Any two neighbouring cells between which the scalpel cuts remove each others’ entries in their “neighbours” array. No recalculation at a tissue level is needed, and in fact tissue complexity is reduced since total intercellular connections are reduced.

Sample Source Code
The following is the main loop activity for a given cell agent within the tissue mesh. Each cell runs this code simultaneously, though not necessarily synchronously:

```java
For (each timeStep in simulation) {
    if(this.dragging == false) {
        updateForce();
        updateAccel();
        updateVel();
        updatePos();
    }
    pullOnNbrs(); //Pull on neighbor cells
    collision(collCheck()); //check collisions
}
```

Implementation Scalability
The numerical simulation can be scaled from two dimensions to three by adding a third orthogonal axis to force and motion vectors. In addition, the initial neighbour search algorithm would add two or four neighbour entries to each cell’s neighbour list, depending on desired connection density. Scaling to three dimensions would not involve a departure from the existing dynamics algorithm except for these additions.

Results
The behaviour of the resulting tissues has visco-elastic properties and responded collectively as damped spring-mass system. Rates of 24 frames per second are achievable with upwards of 400 cells, despite the fact that a relatively inefficient coding platform is use (Flash Player 9, using...
visco-elastic properties requires specialized hardware, but is
within reach. The push-pull and collision
management algorithms work convincingly when running
simultaneously, though processing per time step was
increased. Cutting of the tissue using the ‘scalpel’ tool is
implemented by removing ‘neighbour’ associations in cells on
either side of the cut, allowing arbitrary cutting paths through
the tissue. The dynamics of tissue that could be simulated are
broad, from highly damped to very stiff. It was found that the
tissue instabilities were a potential problem if too low a cell
mass or damping coefficient were used. Creation of tissues
with arbitrary layout and heterogenous cell characteristics is
greatly aided by the ‘drag-and-drop’ capability of the design
environment and the automatic neighbour discovery
algorithm. Tissues composed of different cells can be created
without any placement constraint, providing freedom in
design and implementation.

**Computation Example with Existing Hardware**

Because this cellular architecture is inherently parallel in
nature, there is the potential to use massively parallel
hardware architectures to generate the dynamics. A sample
case is developed to give a sense of current hardware
capabilities.

There has been a recent trend towards developing multi-core
RISC architecture processors for video compression and
wireless communication. **Ambric** currently offers a design
(Ambric2045) with 336 RISC cores on a single die, running
at 300MHz. Each chip is specified by the manufacturer to
perform 1.2 trillion instructions per second. With the estimate
that each cell requires approximately 100 RISC instructions
to update its dynamics, and an update rate of 333Hz (suitable
for tactile feedback), each processor can manage 333,000,000
cells. To fully simulate the dynamics of a solid cubic volume
1000 cells per side would require a minimum of 16 processors
working in parallel, as well as a supervisory processor to
retrieve data for a video and haptic rendering pipeline. In
reality, it is likely that more than 100 RISC instructions may
be required per cell. Ambric is expected to release higher
performance version of this processor with four times the
processing capability which would reduce the number of
processors required to four in this scenario. This represents a
haptic-rate surgery simulation with one billion cells which are
not limited to full-packing density meaning an arbitrary
geometry of this number of cells is possible.

This development in parallel processing hardware
demonstrates that high resolution tissue simulation with
visco-elastic properties requires specialized hardware, but is
within reach.

**Future Development of this Simulation**

Developing this simulation architecture into a 3D model on a
faster computational platform would allow further testing and
verification. This could potentially include proof-of-concept
architecture with parallel processing hardware such as a
single Ambric2045 and a hardware reference design.

Adding haptic feedback test using hardware such as the
SenseAble Phantom would allow for further testing of this
model’s suitability to haptic feedback. Additional tests to
develop accurate cutting, pulling, and insertion dynamics
could then be made.

It will be of benefit to tune the parameters of several types of
simulated cell so that the resulting tissue closely matches data
gathered from real tissue or similar analogues. A method
could be devised to duplicate an actual measurement setup
within the simulation (such as a force sensor) to assist in
validation of the entire verification procedure.

Additional dynamics can be added to the simulation. A
tearing function can be added in which deformation of
intercellular connection beyond a certain maximum length
breaks that connection. Intermediate to this, plastic
deformation could be implemented by permanently changing
default intracellular connection length if a certain stretch
threshold is exceeded. A nonlinear spring constant could also
be added to intracellular connections as a look-up table to
maintain high performance which more closely matching real
tissue in behaviour.

Haptic feedback is not implemented in this model due to
hardware limitations; however, two types of forces can be
impacted on a surgical tool by this model. In the first
scenario, collision detection between the tool and cells can be
converted to a sum of vector Forces acting on the tool.
Second, when pulling of a cell or group of cells by the tool
(i.e. a forceps), the net force vector that would be applied to
the captured cells by neighbours would be transferred to the
tool instead. The fidelity of these haptic sensations would be
dependent on a high volumetric resolution (i.e. a small cell
size with respect to the tool), otherwise any change in number
of contributing cells to any given force effect would be
granular enough to be noticed by the user. The tools
themselves could be rendered using the same cellular
structure so that dedicated algorithms for interaction would
not be necessary.

There is also the potential to simulate non-connected tissues
such as blood. Surface tension and viscosity would be
modelled by rapidly creating connections to neighbour cells,
impacting forces until the connection length exceed a
constant value. For example, this connection length would be
small for low viscosity fluids. The force coupling in these
connections would be tuned to simulate a specific fluid’s
characteristics.
VI. SIMULATION DEVELOPMENT WORKFLOW

Overview

Figure 4 represents a workflow proposed by the author that could potentially produce computationally practical surgical simulations with suitable fidelity for a given application. This is a technology-independent workflow that attempts to capture the entire process from empirical measurement of tissue behaviour through delivery of a final simulation.

<table>
<thead>
<tr>
<th>TIER 1: Concurrent measurements during surgeries</th>
<th>TIER 2: Standardized Measurement Protocol and Data Ontology</th>
<th>TIER 3: Non-tissue analogues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Tissue Characteristic Library</td>
<td>High-Fidelity Human Tissue Mathematical Models (Numerical or analytical)</td>
<td>Constraints Filter</td>
</tr>
<tr>
<td>Verification</td>
<td>Constraints: Procedure Types, Required Fidelity, Hardware Capability, Tissue Geometry, Modeling Type, Real-time Requirements, Haptic Requirements, Visual Requirements</td>
<td>Constraints Filter</td>
</tr>
<tr>
<td></td>
<td>Real-Time Simulation: Tissue Geometry, Appropriate Tissue Model Complexity for application, Appropriate Interface Response</td>
<td>Verification</td>
</tr>
</tbody>
</table>

Fig. 4: Proposed workflow for development of High-fidelity, computationally practical surgery simulation.

Human Tissue Characteristic Library

Implementation of this workflow would require detailed specification of tissue measurement protocols and the associated ontology used to categorize and store these measurements. This would then be stored as a Human Tissue Characteristic Library (HTCL). For example, tissue data could be stored as coefficients in a generalized continuous analytical model, or as raw time-domain samples in response to standardized disturbances such as shears, impulses, and incisions. In each case, the source of the measurement would determine its fidelity as a human analogue, with three tiers shown as an example, Tier 1 being the most reliable source of measurement.

High-Fidelity Human Tissue Model

To interface this Human Tissue Characteristic Library (HTCL) to simulation implementations a highly accurate ‘reference’ simulation model is specified. This model, referred to here as “High-Fidelity Human Tissue Model” (HHTM) would be able to accurately reproduce tissue behaviours using information in the HTCL, although not necessarily at real-time interactive rates. Its purpose would be a validation point against empirical tissue measurement, as well as a globally available reference model for other simulations. As a benchmark, all validation would be against this model.

Constraints Filter

Simulations usable at real-time rates would be developed from this High-Fidelity Human Tissue Model through a process of constraints filtering. This could be implemented as an expert system which recommends a suitable simulation framework or engine based on inputs such as geometric complexity of the simulation, interactivity modes, hardware capability and other parameters. It is possible that the constraints provided would not allow for a real-time simulation, in which case the constraints filter would generate an exception.

Initially it would not be expected that the output of this workflow would be a functioning simulation for the desired constraints, but rather a ‘recipe’ for generating this simulation. For example, minimum hardware requirements, suitable numeric algorithms from a sample code library, and possibly optimized tissue geometries in a suitable voxel or tetrahedral format. This library could be built up and contributed to as more simulations are developed, creating an open contribution framework.

VII. DISCUSSION AND CONCLUSION

This paper has discussed some of the challenges of surgery simulation, focusing on computational limitations. A simulation was developed which allowed design of a 2D tissue cross section with visco-elastic properties and allows pulling and cutting operations on heterogeneous tissue structures. Finally, a workflow was proposed to allow development of validated surgical simulations using a reference simulation model, empirical measurements of tissue mechanics, and a constraints filter.

REFERENCES
