Deformable 3D Objects for a VR medical application

M. Mero
Dpto. de Matemáticas - FACYT,
Universidad de Carabobo (Venezuela)
mmero@lsi.upc.es

A. Susin
Dpt. Matemàtica Aplicada 1
Universitat Politècnica de Catalunya (Spain)
toni.susin@upc.es

Abstract

In this paper , we enhance existing techniques for simulating flexible volumetric objects. The idea is to use a mixed model of Finite Element and Mesh Free Methods. From this approach we will be able to build a 3D deformable model which can be included in a general application of virtual reality for a medical surgery simulator. The final model will be the heart of a patient builded from their actual SPECT data. The mixed approach allow us to construct a multiresolution model that can be used to obtain real time response when an extern user interacts with the model.

1 Introduction

Simulating and animating 3D deformable objects in real time is essential to many interactive applications such as surgery simulators. One of the main characteristics of these simulations is the dynamic interaction between the deformable model and the possible external forces acting on it. The dynamic behavior of our volumetric 3D deformable model is based on linear elastic mechanics. It is essentially based on techniques presented in computer graphics and mechanical engineering literature [RD89] , [Y.F65] and [D.T86].

From the work developed by J. O'Brien and J. Hodgins [JJ99] and G. Debunne [GMMA01] we can state the formulation of the problem in terms of Finite Elements (FEM). Although [JJ99] is focused on the study of fractures in almost rigid materials, the elastic model can be used for simulating more deformable objects. In particular, [GMMA01] build a model for the human liver with the same finite element formulation.

When the final goal of one application is a Virtual Reality environment, topics like precision and speed need to be properly balanced. To combine both characteristics the usual solution is to build a multiresolution model [GMMA01]. The structure of such a model is organized in different layers from a coarse to a finer mesh. The computational accuracy is related to the number and size of the elements. When an external force is applied to the model in a delimited zone, the finer mesh is activated. The other resolutions are used to animate the model according to the distance from the force location. The different multiresolution models differs in how are related two consecutive mesh levels. They can be obtained from a refinement of the previous level mesh or completely independent (just meshing the same volume). This relation is critical in the transition zone where the two different meshes have to be activated.

In any case, we are forced to work with many meshes with the resulting increase in the difficulty of the data structure of the problem. One possibility to avoid this problem is to use only one mesh (i.e. a coarser mesh) and apply another multiresolution approach to the deformable model. The idea is to refine the zone where the forces are applied with particles and to use a Mesh Free Method (MFM) for computing the elastic reaction of the model ([L.L77],[JCCW96],[WYS$^+$96], [AS00],[S.F01]). In this way, our idea is to build a mixed model with FME and MFM that simplify the data structure and uses the multiresoltion approach. A transition zone is defined in order to achieve this goal.

2 Finite Element Formulation

The usual modelization is based on continuum mechanics [Fung]. Regarding our simulation, the first assumption in the continuum approach defines the scale effects is significantly greater than the scale of the material's composition. Therefore, the behavior of the molecules or particles that compose of the material can be modeled as continuous media.

We begin the description of the continuous model by defining material coordinates that parameterize the volume of space occupied by the object being modeled.

In the rest of this section, we will follow [JJ99] and [GMMA01] to formulate our FEM approach.

Let $\mathbf{u} = (u_1, u_2, u_3)^T \in \mathbb{R}^3$ be a vector in $\mathbb{R}^3$ that denotes a location in the material coordinate frame as figure 1 shows. The deformation of the material is defined by the function $\mathbf{x}(\mathbf{u}) = (x, y, z)$ that maps locations in the material coordinate frame to locations in world coordinates. In areas where material exists, $\mathbf{x}$ is continuous. In areas where there is not material, $\mathbf{x}$ is undefined.

We make use of Green's strain tensor ϵ, to measure the local deformation of the material [Y.F65], as is pointed out in [GMMA01]. This tensor is the most appropriated for multiresolution models. The strain tensor is formulated as a symmetric matrix

$$\epsilon_{ij} = \left(\frac{\partial \mathbf{x}}{\partial u_i} \cdot \frac{\partial \mathbf{x}}{\partial u_j} \right) - \delta_{ij}. \tag{1}$$

Here δ_{ij} is the Kronecker's delta. This strain metric only measures deformations, it is invariant with respect to rigid body transformations applied to $\mathbf{x}$ and vanished when the material is not deformed. Both systems, material coordinate and world coordinate, match when the object is not deformed, resulting in a null strain tensor. The material coordinate of a points of mass are fixed, but his position in space will vary over time.

The theory of linear elasticity assumes that stress and strain are linearly linked. In this way we assume that the material is isotropic and with symmetric considerations state that only two independent coefficients describe the material behavior. In this case, the elastic stress is

$$\sigma_{ij}^{(\epsilon)} = \sum_{k=1}^{3} \lambda \epsilon_{kk} + 2\mu\epsilon_{ij}, \tag{2}$$

μ and λ are the Lamé coefficients, μ represents the rigidity of the material while λ measures its ability to preserve volume.

In this work we also use damping forces which are very important if we want to improve realism. The strain rate tensor ν ,measures the rate at which the strain is changing and is the time derivative of ϵ

$$\nu_{ij} = \left(\frac{\partial \mathbf{x}}{\partial u_i} \cdot \frac{\partial \dot{\mathbf{x}}}{\partial u_j} \right) + \left(\frac{\partial \dot{\mathbf{x}}}{\partial u_i} \cdot \frac{\partial \mathbf{x}}{\partial u_j} \right), \tag{3}$$

where $\dot{\mathbf{x}} = \dfrac{\partial \mathbf{x}}{\partial t}$ is the velocity of a point. The viscous stress is

$$\sigma_{ij}^{(\nu)} = \sum_{k=1}^{3} \phi \nu_{kk} + 2\psi\nu_{ij}, \tag{4}$$

where ϕ and ψ control how fast the material looses or dissipate kinetic energy.

The total stress tensor is obtained adding both the elastic stress and the viscous stress $\sigma = \sigma^{(\epsilon)} + \sigma^{(\nu)}$

For our FEM implementation we will use tetrahedral linear elements. We denote by N_i the linear interpolation function associated to each vertex $\mathbf{x}_i$,

$$N_i(x, y, z) = \beta_{i1}x + \beta_{i2}y + \beta_{i3}z + \beta_{i4}. \tag{5}$$

The matrix $\beta = (\beta_{ij})$ is defined from the material coordinates of the vertex. It is always a nonsingular matrix unless the associated tetrahedron is degenerated. Numerical instabilities are associated to near degenerated configurations.

Position and velocity of each point can be interpolated from the respective vertex values

$$\mathbf{x}(\mathbf{u}) = \mathbf{U}\beta \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix}, \qquad \dot{\mathbf{x}}(\mathbf{u}) = \mathbf{V}\beta \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix}, \tag{6}$$

where the matrices $\mathbf{U}$ and $\mathbf{V}$ are defined as $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4]$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4]$, being $\mathbf{u}_i$,$\mathbf{v}_i$ the position and velocity vectors respectively of each vertex.

From (6) we can also compute the derivatives that we will need to compute the strain and strain rate tensor from (1), (3) respectively,

$$\frac{\partial \mathbf{x}}{\partial u_i} = \mathbf{U}\beta\delta_i, \quad \frac{\partial \dot{\mathbf{x}}}{\partial u_i} = \mathbf{V}\beta\delta_i \tag{7}$$

where $\delta_i = [\delta_{i1}, \delta_{i2}, \delta_{i3}, 0]^T$. These partial derivatives are constant because we are using linear basis functions at each element.

The internal force is computed for each element and is applied at their vertices

$$f_i^{(el)} = \frac{vol^{(el)}}{2} \sum_{j=1}^{4} \mathbf{x}_j \sum_{k=1}^{3} \sum_{l=1}^{3} \beta_{jl}\beta_{ik}\sigma_{kl}, \tag{8}$$

where $vol^{(el)}$ is the element volume.

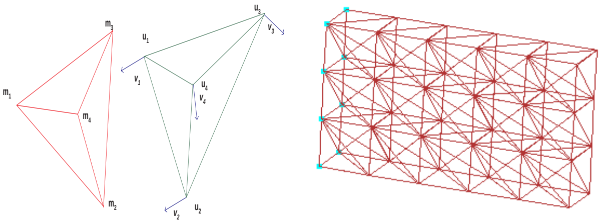

Figure 1. Material coordinate system. One example of a wall FME decomposition.

3 Dynamical Time evolution

Once internal forces of the model are calculated, we can applied other possible external forces and compute the time evolution of our system. Essentially a Newtonian classical dynamics are considered

$$\dot{\mathbf{x}} = \mathbf{v}, \qquad \dot{\mathbf{v}} = \frac{\mathbf{f}}{m}. \tag{9}$$

Here, $\mathbf{x}$, $\mathbf{v}$ and $\mathbf{f}$ are considered for each of the nodes of the FME model.

The Courant condition states that time increments have and upper maximum value to ensure stability,

$$\triangle t < h\sqrt{\frac{\rho}{2\lambda + \mu}}, \tag{10}$$

where h is the minimum of the distances between a node and its neighbors and ρ is the material's rest density.

A simple Euler's explicit scheme can be used to integrate (9) giving acceptable results in terms of speed and accuracy. To increase time steps, we follow [GMMA01] establishing a *semi-implicit* scheme that improves stability.

A first approximation of the force applied at a node when it moves is given by $\mathbf{J} = \frac{\partial f}{\partial x}$, the force Jacobian matrix associated at each node. We can precompute this matrix in the object's rest position, assuming that its neighbors are fixed, and consider that it is constant during the simulation. The *semi-implicit* scheme can be stated as

$$\begin{aligned} \tilde{\mathbf{v}}_{n+1} &= \mathbf{v}_n + \tfrac{\mathbf{f}}{m}\triangle t, \\ \mathbf{v}_{n+1} &= \left(\mathbf{I}_3 - \tfrac{(\triangle t)^2}{m}\mathbf{J}\right)^{-1} \tilde{\mathbf{v}}_{n+1}, \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{v}_{n+1}\triangle t. \end{aligned} \tag{11}$$

The update step for the velocities acts as a filter at the only cost of a product by a constant matrix. An implicit method would update the Jacobian matrix $\mathbf{J}$ at each time step. The semi-implicit approach is only a simplification, but because of the increase in the stability, the time step can be almost doubled.

4 Results

We have implemented the above FME method for different types of geometries. Although this can be a general modelization, our final goal is to obtain a deformable model for the human heart. One of the main problems will be to get the tetrahedra volume decomposition for this model. Moreover, the total amount of elements we need has to be as low as possible to get real-time simulation rates. The problem is that if only a coarse approximation of the model is used, then the accuracy on the computed deformation will be low. This will force us, as a future work, to improve our model with a multiresolution strategy.

With figure 2, we show a first model corresponding to a deformable column and a deformable wall. We can build different number of columns and use these models to test the different parameters of the model.

Figure 3 shows an example of two concentric ellipsoids. The inside ellipsoid is empty and only the volume between them is decomposed in tetrahedra elements. This is a rough approximation of our final model but we used it as a test model both for elasticity parameters and time rates.

Finally, the table 1 shows the elasticity parameter values, time step and geometry values used in each of the examples.

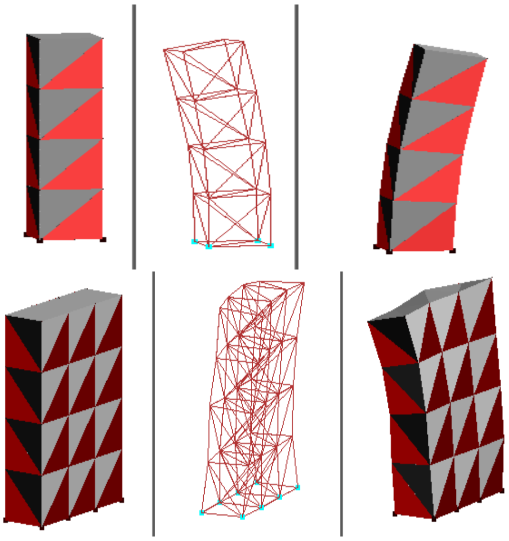

Figure 2. Example of wall FME simulation. A wire and a polygonal representation is shown

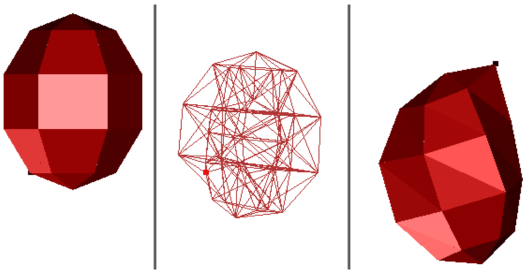

Figure 3. Example of two concentric ellipsoids

	column	wall	ellipsoids	heart
λ	5.0	250.0	5.0	0.04
μ	2.0	100.0	2.0	1.6e-3
Φ	9.0e-5	0.0	9.0e-5	1.0e-4
Ψ	9.0e-5	0.1	9.0e-5	1.0e-4
$\triangle t$	7.5e-3	0.01	0.04	9.0e-3
# nodes	43	20	40	353
# elements	118	24	72	1352

Table 1. Model parameters and geometry values

5 Future work: Multiresolution with Mesh Free Method

As we explain in the previous sections, a multiresolution model is preferred to our final implementation. The approach we have chosen is to add different particle points on our FEM model. This is still a work in progress but the idea is taken from [L.L77],[JCCW96],[WYS$^+$96] and the recent works of [AS00] and [S.F01].

The first *Mesh Free Method* (MFM) is the *Smooth Particle Hydrodynamic* (SPH) [L.L77]. It is based in a simple property of Dirac's delta function, $\delta(x)$, which is

$$u(x) = \int \delta(x-y)u(y)dy.$$

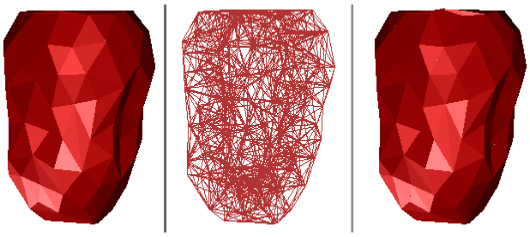

Figure 4. An actual heart recovered geometry

Here, $u(x)$ is the function to be interpolated and the basic idea is to approximate this equation as

$$u(x) \approx \tilde{u}^{\rho}(x) = \int C_{\rho}\phi(\frac{x-y}{\rho})u(y)dy, \qquad (12)$$

where ϕ is known as *weight function* which is an even positive function with compact support.ρ is the expansion parameter related with the function's support of ϕ and C_ρ is the normalization constant chosen to satisfy

$$\int C_{\rho}\phi(\frac{x}{\rho}) = 1.$$

A discrete interpolation is computed by integrating (12) numerically,

$$u(x) \approx \tilde{u}^{\rho}(x) \approx u^{\rho} = \sum_i C_{\rho}\phi(\frac{x-x_i}{\rho})u(x_i)\omega_i. \qquad (13)$$

Here x_i and ω_i are points and weights used in the numerical integration scheme. Usually these points are called *particles*. The interpolation for the SPH method can be expressed as

$$u(x) \approx u^{\rho}(x) = \sum_i N_i u(x_i),$$

with the interpolation function basis

$$N_i(x) = C_{\rho}\phi(\frac{x-x_i}{\rho})\omega_i.$$

This definition cause that $u^{\rho}(x_i) \neq u(x_i) = u_i$, it means that, in general, the basis functions do not satisfy the Kronecker's delta condition $N_j(x_i) \neq \delta_{ij}$.

In the context of continuous interpolation for the equation (12), a basis function ϕ can be modified for exactly reproduce the real polynomial space $\mathcal{P}$, that is

$$p(x) = \int C_{\rho}\phi(\frac{x-y}{\rho})p(y)dy, \quad \forall p \in \mathcal{P}.$$

From this polynomial approximation context, we can stated our idea of multiresolution approach. It is based on a mixed method of FEM and MFM that uses FEM with a coarse mesh and uses particles with a MFM to enrich the number of points used for computation in the parts of the model where an external force is applied.

For that approach we will use a variant of the MFM, the *Free Galerkin Method* [S.F01] with two different approaches shown in figures 5 and 6 respectively. In the first case, we will study a mixed model with two different zones computed with only FEM or MFM and with a transition zone between the two methods. The second approach will consist on using mainly FEM and a mixed zone with both FEM and MFM just to improve precision in the external zones where occasionally an external force will be applied.

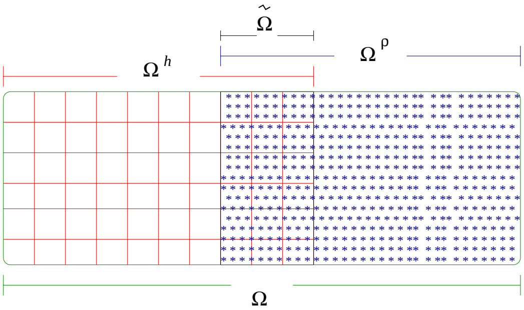

Figure 5. Three different areas with FEM, MFM and a central mixed transition zone

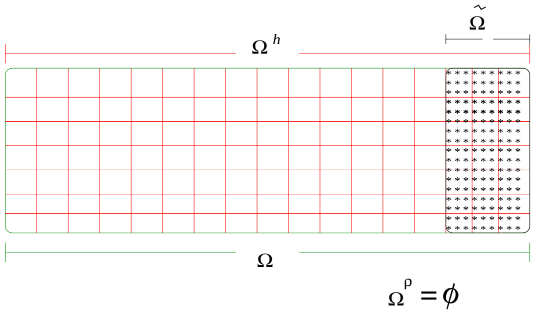

Figure 6. Two areas in the model with FEM and a mixed zone

6 Conclusions

We have presented an in progress work where a multiresolution deformable 3D model is aimed. By the moment, a Finite Element Method has been implemented and it works properly. This work is included in a more general frame where a complete medical tool is being constructed. The final goal is to be able to reconstruct the shape of the left ventricle (LV) of an actual patient from its SPECT data capture. A surface reconstruction will be the first step its final output will be two meshed surfaces corresponding to the external and internal walls of the LV. From these two meshes a tetrahedralization of the inside volume will be obtained with an advance front method relying on the external and internal meshes. This tetrahedral mesh will be the input of our model, which is planed to work in an Virtual Reality environment as a training tool for medical surgery. In this paper we have introduced the formulation of the model and the idea to implement a mixed model with FEM and MFM methods.

7 Acknowledgments

The authors are thankful to the computer graphics medical group of the IRI (UPC) for their suggestions and collaborations during this research, which is partially supported by a TIC2000-1009 CYCIT project. The first author is also supported by a Carabobo University (Venezuela) grant.

References

[AS00] A.Huerta and S.Fernández. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48:1615 – 1636, 2000.

[D.T86] D.Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(4):413 – 424, 1986.

[GMMA01] G.Debunne, M.Dsbrun, M.Cani, and A.Barr. Dynamic real-time deformations using space and time adaptive sampling. *In SIGGRAPH 2001 Conference Proceedings*, 2001.

[JCCW96] J.Chen, C.Pan, C.Wu, and W.Liu. Reproducing kernel particle methods for large deformation analysis of no-linar. *Computer Methods in Applied Mechanics and Engineering*, 139:195 – 227, 1996.

[JJ99] J.O'Brien and J.Hodgins. Graphical models and animation of brittle fracture. *In SIGGRAPH 99 Conference Proceedings*, pages 137 – 146, 1999.

[L.L77] L.Lucy. A numerical approach to the testing of the fissionhypothesis. *The Astronomical Journal*, 82:1013 – 1024, 1977.

[RD89] R.Cool and D.Malkus. Concepts and applications of finite element analysis. In John Wiley and Sons., editors, *Concepts and applications of finite element analysis.* 1989.

[S.F01] S.Fernández. *Mesh-Free Methods and Finite Elements, friend or foe?* PhD thesis, Universitat Politècnica de Catalunya, UPC, Spain, 2001.

[WYS$^+$96] W.Liu, Y.Chen, S.Jun, J.Chen, T.Belytschko, C.Pan, R.Uras, and C.Chang. Overview and applications of reproducing kernel particle methods. *Archives of Computational methods in Engineering, State of the art Reviewa*, 3:3 – 80, 1996.

[Y.F65] Y.Fung. Linear elasticity. In Prentice Hall Inc., editor, *Foundations of Solid Mechanics.* Englewood Cliffs, N.J, 1965.