Hierarchical Multi-Resolution Finite Element Model for Soft Body Simulation

Matthieu Nesme[1&2], François Faure[1], and Yohan Payan[2]

[1]GRAVIR/IMAG-INRIA [2]TIMC/IMAG Grenoble, France

Abstract. The complexity of most surgical models has not allowed interactive simulations on standard computers. We propose a new framework to finely control the resolution of the models. This allows us to dynamically concentrate the computational force where it is most needed.
Given the segmented scan of an object to simulate, we first compute a bounding box and then recursively subdivide it where needed. The cells of this octree structure are labelled with mechanical properties based on material parameters and fill rate. An efficient physical simulation is then performed using hierarchical hexaedral finite elements. The object surface can be used for rendering and to apply boundary conditions.
Compared with traditional finite element approaches, our method dramatically simplifies the task of volume meshing in order to facilitate the using of patient specific models, and increases the propagation of the deformations.

1 Introduction

1.1 Context

Soft body simulation is a growing research domain, for communities such as Computer Aided Surgery, Virtual Reality or Computer Graphics. Computer aided surgery (CAS) aims at assisting surgeons for the realization of diagnostic and therapeutic gestures in a rational and quantitative way in order to increase safety and accuracy [1]. While the first designed systems focused on orthopaedics, researchers addressed more recently anatomical structures that cannot be considered as ”rigid” as they are mainly composed of biological soft tissues. The corresponding CAS systems therefore need to take into account the displacements of the structures as well as their deformations. In most cases, authors propose to build biomechanical models of the anatomical structures and use these models to predict, in the most accurate way, the tissue deformations induced by the surgical gesture.
Virtual Reality (VR), in its interactions with the Medical community, has recently provided surgical simulation systems (Cotin et al., 1996). As for the flight simulators used to train pilots, the idea is that these surgical VR systems could be a great help in the learning and training processes, allowing the surgeon to acquire, for example, some difficult hand-eye coordinations, to repeat several times the most difficult gestures or to choose the best surgical procedure for a given pathological case. As for the CAS systems, deformable models have been included into the simulators, with constraints in term of robustness and computation times.
Computer Graphics (CG) has developped methods for the visually plausible animation

of complex physical objects such as clothes and hairs. Considerable speedups have been obtained for stiff flexible bodies using implicit time integration, which allows arbitrary time steps to be performed. Significant advances have also been made using hierarchical modeling and the control of levels of detail.
These three communities now converge towards the same needs, in terms of soft body modelling: accuracy, robustness and interactivity (*i.e.* fast computation times). Indeed, from one side the CAS community is now looking for models that could be per-operatively used, with possible real-time re-planning of the surgical gesture. From the other side, the VR and CG communities now focus on the accuracy of the deformations, in order to be as realistic as possible, in comparison with real data.
In this framework, some recent works, coming from these communities, try to provide mechanical models with innovative implementations that preserve a continuous modelling context (with, for most of the works, a numerical resolution through the Finite Element Method) while proposing improvements, in terms of robustness and computation times. In addition, models built in the CAS or VR contexts need to be adapted to each patient anatomy. This point is particularly challenging (and time consuming) when a patient-specific Finite Element mesh needs to be defined.

Next part tries to summarize all of these recent works (part 1.2), while a new modelling approach is introduced in part 2. An example of implementation is presented in part 3 before providing some results (part 4).

1.2 Related Work

In order to improve the computational efficiency of continuous biomechanical models, researchers have proposed new approaches concerning (1) the Finite Element discretization, (2) the dynamical integration and (3) the numerical resolution methods.
Because of the need for speed, the first interactive methods were based on precomputed matrix inversion [2]. To extend these methods to large deformations frameworks, a non-linear computation of the strains is used in [3, 4]. Recently proposed methods favor a new approach based on the decomposition of the displacement of each element into a rigid motion and a pure deformation tractable linearly in the local frame [5–8]. These methods allow a large displacements and rotations framework.
In the animation community, implicit integration methods have become popular, thanks to the iterative solution based on conjugate gradient presented by [9]. Although these methods permit large time step, they become ”expensive” when a fast propagation of the deformation is suited, since they require a lot of iterations to solve the system accurately. On the contrary, explicit integrations do not use iterations but require small time steps to maintain stability. Therefore, if a fast propagation of the deformations is needed, both implicit and explicit integration schemes remain computationally expensive. To face this problem, hierarchical methods have been proposed, providing an improvement of the propagation of the deformations (see for example the hierarchical solvers proposed by [10, 11]).
In order to adapt the numerical solution schemes to the adequate level of details, authors have proposed to adapt the Finite Element (FE) mesh according to the actual state of the model (in terms of displacement, strain or stress). They propose therefore a

multi-resolution FE approach ([12, 3, 13]). The idea is for example to define, for a given anatomical structure, different FE meshes, from a very coarse one to a full refined one. If boundary conditions induce small deformations inside the structure, the coarse mesh is sufficient for providing accurate FE discretization. On the contrary, a dense mesh is used where the deformation is high. [14] shows how to link hexahedral elements of different sizes but the octree hierarchy is not used in the dynamics computations.

2 Contribution

Our approach proposes to merge a multi-resolution description with a Hierarchical FE integration. It is supposed to provide a numerical scheme that can be used for any type of mechanical description, from a small deformation framework to hyperelasticity. The objective is only to gather some methods already proposed in the literature in order to improve the propagation of the deformations as well as the efficiency of the computation according to the mechanical and geometrical state of the soft body. The specificity of the method is that a global 3D mesh is defined from a classical octree division of a bounding box including the soft body. Therefore, no FE mesh is needed to specificaly model the 2D or 3D geometry of the body. Indeed, the FE computations are applied to the 3D mesh defined by the octree. This octree can be directly built from the volumic data resulting of the segmented patient scan, so it is very efficient when we have few time to built a patient specific mesh, for exemple when the patient comes in emergency. To improve the propagation of the deformations, a hierarchical basis is defined to interpolate the FE computations, from the global parent cell defined by the bounding box to each child cell of the octree. A real difficulty is to convincingly animate stiff materials in an interactive context, when only an approximate solution can be computed. Our hierarchical approach gives more realistic results in this case. Moreover, the multi-resolution scheme is used to decide, for a given state of the body, which levels of the octree should serve as basis for (1) FE computation and (2) 3D rendering: for example, only regions with high strains level should use a dense octree level for FE computation, while regions that are not displayed on the screen because they are not seen by the camera should use a coarse octree mesh level for the 3D rendering.

2.1 Octree Mesh and Multi-resolution

The first step consists in defining the complete 3D octree mesh as presented in figure 1-a. Starting from a cubic bounding box of the body, an iterative algorithm is used to divide each ”parent” cube in order to generate eight ”child” cubes. The cubes that do not contain any part of the body are removed from the octree mesh. The remaining cubes are again divided, so that each of them will generate 8 new cubes. A maximal level of division N_{max} is defined once, leading to the ”maximal density” octree mesh (figure 1-a). Using this octree mesh architecture as a baseline, two intermediate resolutions will be defined at each time step of the global computation of the system, namely the octree resolution N_{FE} used for the FE mesh interpolation (figure 1-b) and the octree resolution N_{Rend} used for the rendering display (figure 1-c). These two resolutions can change during the solving of the system, according to the changes in the boundary conditions as well as the location of the camera that looks at the scene. For example, one condition to define whether a given level of the octree mesh is suitable for a given point of the body consists in looking at the strain rate. If it is sufficiently low, this means that the current resolution is sufficient. On the contrary, if a high strain variation is observed, a denser mesh is preferable around this region of the body leading to the use of the child cells of the actual octree element. Once the N_{FE} level is reached, the corresponding octree 3D mesh is used for the FE computation. In order to limit the influence of cells that would contain a small amount of the body (cells located at the surface of the body), it is proposed to ponderate the rheology of these cells by their filling ratio as illustrated in figure 4.

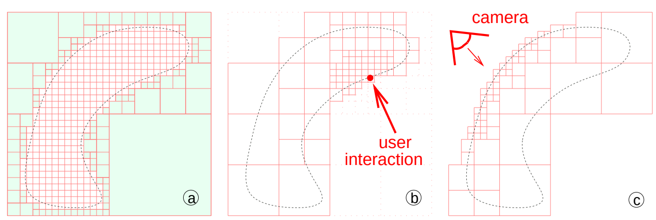

Figure 1. An example of adaptativity. (a) leaves of the octree mesh = the finest level of details. (b) mechanical leaves = the finest mechanical level. (c) geometric leaves = the finest geometric level.

2.2 Hierarchical FE Bases

A function decomposed in a hierarchical basis is modeled using a rough approximation based on a few broad-range sample points, along with a number of recursively narrower-range sample points encoding local detail added to the approximation. Each value of the function is thus the sum of shape functions with various radius of influence. This approach allows one to easily control the level of detail by simply inserting or dropping control values where desired [15]. Another nice feature of this approach is to considerably speed up the convergence of shape optimization, as shown in geometric modeling [16]. It has been successfully applied to finite element methods [17, 18].
In the case of the octree mesh introduced above, the position stored for each vertex is relative from its parents position. Therefore, only vertices from the root cell (*i.e.* the cubic bounding box that includes the body) have real position in the 3D space. At start, all others child cells have a null relative position which only depends on their parents. Figure 2 illustrates the FE interpolation that will be provided with hierarchical linear functions.

3 An Implementation

This section proposes an example of the implementation of the previously presented hierarchical approach. This implementation uses the Cauchy's deformation tensor with a co-rotational handeling of large displacements [5–8] and a viscoelastic material.

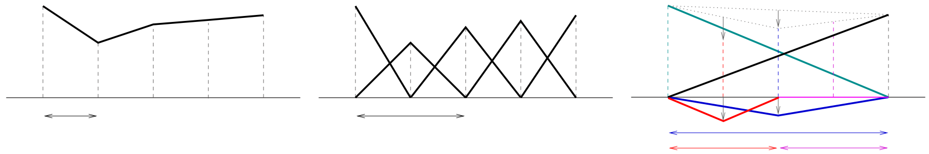

Figure 2. Illustrations of the finite element (top-left), basis function (top-right) and hierarchical basis function (bottom) points of view with linear interpolation. In the basis function point of view, the solution corresponds to a combinaison of functions associated with the nodes. In the hierarchical point of view, the influence support of these functions varies.

3.1 Large displacements

To handle rotational displacements, each detail (*i.e.* hierarchical value) has to be represented in a local frame that follows the rotation as showed in figure 3. In this way, a local frame is attached at each cell, and each node is defined in the local frame of a cell. The corresponding cell of a node is determined in taking the finnest cell that contains all the incident cells to the node as represented in the figure 3-right. To compute cell rotations, the eigenvectors are used as explain in [5] by decomposing the matrix formed with the egdes averages in the three directions.

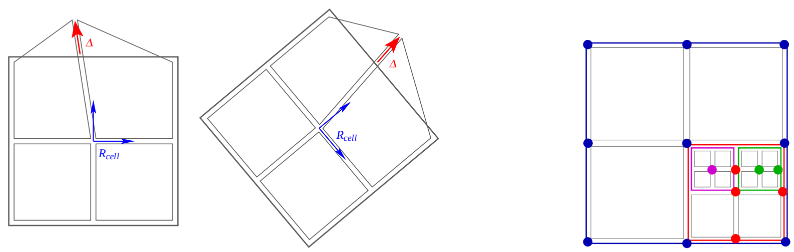

Figure 3. By taking into account the local frame of the cell, the hierarchical values are invariant in rotation.

3.2 Mechanics

The standard method used to simulate viscoelastic solids is considered, equations and notations can be found in [19].

Our approach induces differences with classical formulation concerning the displacement u which is not defined in global space coordinates, but is defined hierarchically. Only displacements of the vertices of the root cell are in space coordinates. The displacement of others vertices is relative from their parents. To build the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{K}$, not only finest elements are considered. Indeed, for each elements along the hierarchy we take into account all nodal functions that influence the considered element as exprimed in algorithms 1 and 2. $\mathbf{C}$ is the stress-strain matrix relating the material properties and the strain-displacement matrix $\mathbf{B}_i$ is obtained by differentiation of interpolation functions $\mathbf{h}_i$ with respect to natural coordinate and premultiplying the result by the inverse of the Jacobian operator. $\mathbf{H}_i$ is a matrix of the interpolation functions $\mathbf{h}_i$.

Algorithm 1 BUILD MATRICES **K** AND **M**

```
for each cell do
  for each vertex i defined at level of cell do
    INTEGRATE(B_i, B_i, H_i, H_i, C_cell, J_cell)
    for each vertex j ≠ i defined at level of cell do
      INTEGRATE(B_i, B_j, H_i, H_j, C_cell, J_cell)
    end for
    for each ancestor of cell do
      for each vertex j defined at level of ancestor do
        take function h_j between range of cell in ancestor // detail in section ??
        INTEGRATE(B_i, B_j, H_i, H_j, C_cell, J_cell)
        INTEGRATE(B_j, B_i, H_j, H_i, C_cell, J_cell)
      end for
    end for
  end for
end for
// Note that some computations can be omitted in considering the symetric aspect of matrices
K and M: K_{i,j} = K^T_{j,i} and M_{i,j} = M^T_{j,i}
```

Algorithm 2 INTEGRATE($\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}, \mathbf{J}$)

$$\mathbf{K}_{i,j} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{B}_i^T \mathbf{C} \mathbf{B}_j \; det\mathbf{J} \; dr \; ds \; dt$$

$$\mathbf{M}_{i,j} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{H}_i^T \mathbf{H}_j \; det\mathbf{J} \; dr \; ds \; dt$$

4 Results

Figure 4 shows an octree mesh for a liver model.

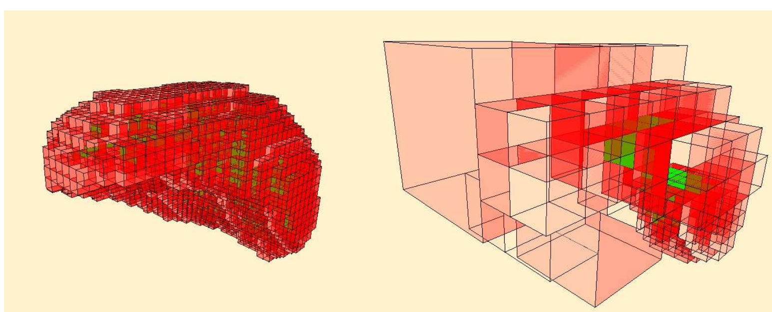

Figure 4. An octree-mesh for a liver: densities of mechanical leaves for the finnest level of details and for a multiresolution mesh.

Table 1 compares the number of iterations necessary to converge to the equilibrium with a static solver using the nodal approach (*i.e.* the classical non-hierarchical one) against the hierarchical approach. Two examples are considered for several numbers of elements, the first one consists in a cubic fixed beam subject to gravity, while in the second a force is applied to a corner of the beam. As expected, the convergence is faster using the hierarchical approach. When the corner is pulled, the other end moves directly, whereas in the nodal model it is necessary to propagate the deformation along all elements.

Number of elements		1	8	64	512
example 1	nodal	1	13	55	146
(gravity)	hierarchical	1	11	27	47
example 2	nodal	8	50	87	198
(boundary force)	hierarchical	6	24	37	52

Table 1. Number of CG iterations of the static solver until convergence, on two examples on a cubical fixed beam.

This faster propagation is useful in case of real-time simulation when only few iterations can be performed on the implicit integration at each step. Using hierarchy, a small number of iterations (approximately ten) provides a much more accurate result, as illustrated in figure 5 that plots the convergence speed of the second example of table 1. In case of very soft materials, fast propagation can be unrealistic. In this case a classical nodal approach can be better suitable.

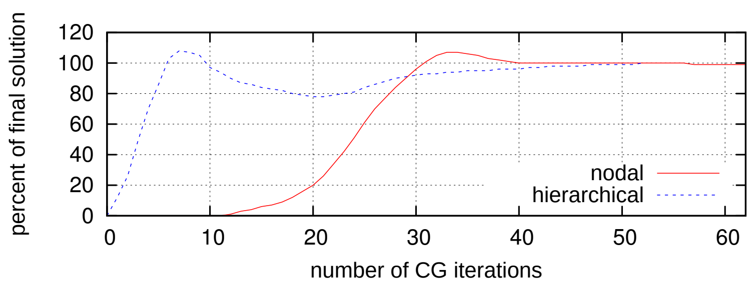

Figure 5. Convergence speed in a static solver.

5 Conclusion

We proposed in this paper a hierarchical multiresolution technique to animate soft bobies. This new approach based on a octree mesh permits to work on various geometrical representations of an object without needing to provide a volumetric mesh of this object. Using the hierarchical approach improves the propagation and permits to simulate more rigid materials. Despite it, we do not yet obtain better results in terms of computation time because an optimized structure is difficult to set up. To make this work usable, it will be necessary to integrate criteria of adaptivity (automatic definition of N_{FE} and N_{Rend} values), and to take into account effective boundary conditions.

References

1. Taylor, R., Lavallée, S., Burdea, G., Mosges, R.: Computer integrated surgery: Technology and clinical applications, Cambridge, MA: MIT Press (1996)
2. Cotin, S., Delingette, H., Clement, J.M., Tassetti, V., Marescaux, J., Ayache, N.: Volumetric deformable models for simulation of laparoscopic surgery. In: Computer Assisted Radiology. (1996)
3. Debunne, G., Desbrun, M., Cani, M.P., Barr, A.H.: Dynamic real-time deformations using space and time adaptive sampling. In: SIGGRAPH '01. (2001)
4. Picinbono, G., Delingette, H., Ayache, N.: Non-linear anisotropic elasticity for real-time surgery simulation. Graph. Models (2003)
5. Etzmuß, O., Keckeisen, M., Straßer, W.: A Fast Finite Element Solution for Cloth Modelling. Proc Pacific Graphics (2003)
6. Hauth, M., Straßer, W.: Corotational simulation of deformable solids. In: Proc WSCG. (2004)
7. Müller, M., Gross, M.: Interactive virtual materials. In: Proc Graphics Interface. (2004)
8. Nesme, M., Payan, Y., Faure, F.: Efficient, physically plausible finite elements. In: Eurographics (short papers). (2005) 77–80
9. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: SIGGRAPH '98. (1998)
10. Terzopoulos, D., Fleischer, K.: Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In: SIGGRAPH '88. (1988)
11. Wu, X., Tendick, F.: Multigrid integration for interactive deformable body simulation. In: ISMS. (2004) 92–104
12. Debunne, G., Desbrun, M., Barr, A.H., Cani, M.P.: Interactive multiresolution animation of deformable models. In: Eurographics Workshop on Computer Animation and Simulation. (1999)
13. Wu, X., Downes, M.S., Goktekin, T., Tendick, F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshe. In: EG 2001 Proceedings. (2001)
14. G.P.Nikishkov: Finite element algorithm with adaptive quadtree-octree mesh refinement. In: ANZIAM J. Volume 46(E). (2005) C15–C28
15. Stollnitz, E.J., DeRose, T.D., Salesin, D.H.: Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann Publishers, Inc. (1996)
16. Gortler, S.J., Cohen, M.F.: Hierarchical and variational geometric modeling with wavelets. In: SI3D '95. (1995)
17. Grinspun, E., Krysl, P., Schröder, P.: Charms: a simple framework for adaptive simulation. In: SIGGRAPH '02. (2002)
18. Capell, S., Green, S., Curless, B., Duchamp, T., Popović, Z.: A multiresolution framework for dynamic deformations. In: SCA '02. (2002)
19. Bathe, K.J.: Finite Element Procedures in Engeneering Analysis. Prentice-Hall, Inc. (1982)