Shell Model for Reconstruction and Real-Time Simulation of Thin Anatomical Structures

No Author Given

No Institute Given

Abstract. This paper presents a new modelling technique for the deformation of thin anatomical structures like membranes and hollow organs. We show that the behaviour of this type of surface tissue can be abstracted with a modelling of their elastic resistance using shell theory. In order to apply the shell theory in the particular context of medical simulation, our method propose to base the geometrical reconstruction of the organ on the shape functions of the shell element. Moreover, we also use these continuous shape functions to handle the contacts and the interactions with other types of deformable tissues. The technique is illustrated using several examples including the simulation of an angioplasty procedure.

1 Introduction

The human body is composed of various deformable anatomical structures. Realistic modelling of organ deformations is a challenging research field that opens the door to new clinical applications including: medical training and rehearsal systems, patient-specific planning of surgical procedure and per-operative guidance based on simulation. In all these cases the clinician needs fast update of the deformation model to obtain a real-time display of the computed deformations. Another key challenge of soft-tissue modelling is the variousness of the mechanical behaviours. It seems unrealistic to target a unique model for all tissues. Yet most of previous works focus on volumetric models that are able to capture the behaviour of solid organs like the liver or the brain (see for instance [1,2]). In contrast, this paper seeks to propose a solution for simulating, in real-time, the deformation of thin anatomical structures whose volume is negligible compared to their surface area. Examples of such anatomical structures include hollow structures, such as the wall of blood vessels, or membranes, such as the Glisson's capsule surrounding the liver.

Shell theory allows the modelling of structure deformations when the thickness is small compared to its other dimensions [3]. The key idea is to model the physical shell as a surface but endowed with mechanical properties in the form of elastic resistance to stretching and bending forces. Rather than resorting to shell theory, previous works in medical simulation often rely on linear or angular mass-spring models as in [4,5]. Yet, such models are limited in their ability to describe certain behaviour, as they do not rely on continuum mechanics: it is difficult to derive spring stiffness (in particular for angular springs) from elastic properties (Young's modulus and Poisson's ratio). The use of fast shell-based modelling for interactive simulation was recently introduced in computer graphics with the work of Choi et al. [6]. Their model relies on simplified energy functions and precomputed modal analysis for fast and visually realistic results. We propose to rely on a similar approach but with more accuracy to be applicable to medical simulation. Our model is not based on modal analysis but uses a co-rotational formulation and polynomial shape functions presented in [7].

To model the deformation of complex anatomical structures using shell elements, the first step is to describe its surface with curved patches. This process is quite similar to the reconstruction of the surface of objects in computer vision. Indeed calculating curvature maps of 3D surfaces represented as digitised data (point clouds or triangulated meshes) has been extensively studied. One of the most common approach is to use continuous surface patches [8,9]. The surface is locally approximated by an analytic representation called surface patch, usually chosen to be an implicit surface. These works target non noise-sensitive approaches and coherent surface and curve extraction from 3D data [10]. However, our situation is substantially different as we want to model the deformation of the structure. In that regard the curvature of the surface has a physical meaning: it represents the mid-surface of the shell on which the plane stress hypothesis [3] applies. We propose to approximate the surface of anatomical structures with shell elements whose each surface is described by the shape function used in our shell formulation. These polynomial shape functions are used in three different ways in our computational model: (a) to approximate complex geometrical shapes, (b) to compute internal forces, (c) to compute contact forces onto a curved triangle. Section 2 presents our Finite Element Modelling (FEM) for shell elements and how we process contacts and interactions with other models. In section 3 we introduce an automatic process to obtain meshes from image based reconstruction. Finally the benefits of our approach (meshing of a curved surface, fast computation and possible interactions with solid models) are illustrated using various examples showed in section 4.

2 Co-rotational triangular shell model for thin structures

A complete description and validation of our co-rotational triangular shell finite element model is available in one of our previous publication [7]. Therefore we will only remind briefly the key points. We improved and extended a plate model first introduced by Przemieniecki [11] to a co-rotational formulation. Co-rotational approaches offer a good trade-off between computational efficiency and accuracy by allowing small deformations but large displacements. Once combined with an in-plane membrane formulation we obtain an accurate, yet computationally efficient, shell finite element method featuring both membrane and bending energies. In the following we detail the bending stiffness computation in order to present the polynomial shape functions that are used in the shell model.

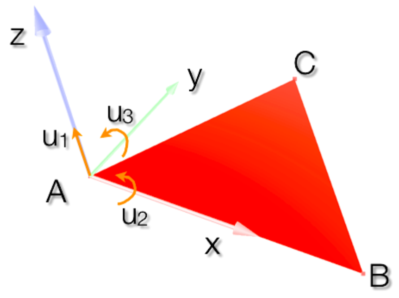

Fig. 1. The different degrees of freedom u of a triangular thin plate in bending.

Polynomial shape function To calculate the stiffness matrix for the transverse deflections and rotations shown on Fig. 1, the deflection u_z is computed using a polynomial interpolation:

$$u_z = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8xy^2 + c_9y^3 \tag{1}$$

where $c_1, \ldots, c_9$ are constants. Using a third-degree polynomial expression allows us to reach a greater precision for both the computation of the bending energy and the interpolation within the surface of the shell. Let us define the vector $\mathbf{u} = \{u_1u_2 \ldots u_9\}$ of the displacements and slopes at the three corners of the triangular plate using the following notations:

$$u_1 = (u_z)_{x_1,y_1} \qquad u_2 = \left(\frac{\partial u_z}{\partial y}\right)_{x_1,y_1} \qquad u_3 = -\left(\frac{\partial u_z}{\partial x}\right)_{x_1,y_1} \tag{2}$$

and so on for the two other vertices and we can derive a matrix $\mathbf{C}$ such as $\mathbf{u} = \mathbf{Cc}$ where $\mathbf{c} = \{c_1c_2 \ldots c_9\}$. We can then calculate the strains from the flat-plate theory using:

$$e_{xx} = -z\frac{\partial^2 u_z}{\partial x^2} \qquad e_{yy} = -z\frac{\partial^2 u_z}{\partial y^2} \qquad e_{xy} = -2z\frac{\partial^2 u_z}{\partial x \partial y} \tag{3}$$

Symbolically this may be expressed as $\mathbf{e} = \mathbf{Dc}$ where $\mathbf{D}$ derives from (1) and (3). Noting that $\mathbf{c} = \mathbf{C}^{-1}\mathbf{u}$, we have $\mathbf{e} = \mathbf{DC}^{-1}\mathbf{u} = \mathbf{bu}$ where the strain-displacement matrix $\mathbf{b} = \mathbf{DC}^{-1}$. The stiffness matrix $\mathbf{K}_e$ for an element is then obtained from:

$$\mathbf{K}_e = \int_v \mathbf{b}^T \boldsymbol{\chi} \mathbf{b} dV \quad \text{where } \boldsymbol{\chi} \text{ is the material matrix} \ . \tag{4}$$

Mechanical interactions with the curved surface of shells The practical interest of modelling complex behaviours such as bending and twisting would remain fairly low for medical simulation if contacts and constraints were not handled properly. In our case the difficulty comes from different sources. First the collision detection must be carried out with the curved surface of shell elements as opposed to the classic detection on plane triangles. Then forces applied to a given triangle need to be distributed between linear forces and torques onto its three vertices. As we will see, the same polynomial interpolation function chosen to compute the bending energy in our FEM formulation is also used to capture the interactions between the curved surface and other objects.

In order to detect the collision with the bent surface, we have chosen the subdivision approach. Therefore a finer mesh is created by adding vertices on the surface of each element. We first sample the flat surface of each element by recursively dividing each triangle into four smaller ones and the deflection of each new vertex is computed using (1) according to the displacements and slopes at the three vertices of the triangular element. This process of subdivision allows us to render each shell as a curved triangle (Fig. 2 (a) and (b)) and detect any collision with the curved surface of the shell using any of the classic collision detection algorithms working on flat triangles.

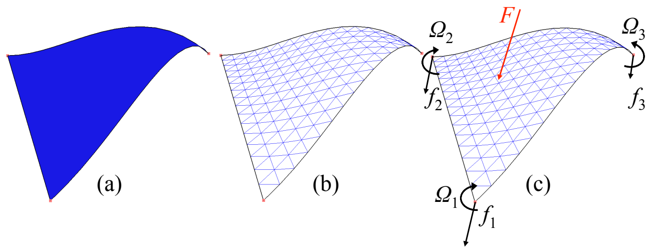

Fig. 2. (a) The triangle formed by the three vertices of the shell has been recursively subdivided 3 times and the deflection of each new vertex was computed according to the same deflection function used in our shell FEM. (b) Sampling the actual surface of the shell allows rendering and more accurate collision detection. (c) The shape function is used to distribute an external force F onto the triangle nodes.

Once a collision has been detected, it must be processed by distributing the linear force received on the bent surface between the three vertices of the triangle. First the linear part of the force is simply transmitted on each node using the barycentric coordinates of the contact point's projection onto the triangle.

The main difficulty is to convert the normal component of the force applied to the bent surface into a torque at each of the three nodes (Fig. 2 (c)). Our approach is the following: during force computation, we use the change in orientation measured at each node to compute the local deflection of each subvertex within the triangle. Differentiating the formulation twice yields a relation between the torque applied at each node and the generated force in bending. We therefore need to invert the latter formulation to convert a bending force into torques at each vertex. We start by retrieving the normal component of the applied force vector F_z. We project the application point of the force into the triangle's plane and compute its local coordinates (x, y). We create the polynom $P = F_z(1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2 \;\; x^3 \;\; xy^2 \;\; y^3)^T$. The moments at each vertex are then obtained with $\Omega = (\mathbf{C}^{-1})^T P$. Thus we are able to transmit any force coming from interactions with the curved surface of shells to the mechanical vertices used in our FEM formulation.

3 Physics-based reconstruction using shell elements

Because the surface of an anatomical structure has a physical meaning, we propose to patch the surface with triangular elements whose interpolation makes use of the same shape function designed for our shell FEM formulation. By suggesting a method to reconstruct the surface of any structures with curved shell, the meshing process is optimised. Indeed while many flat triangles are required to describe highly curved surfaces, fewer triangular shell elements are needed to describe the given geometry with the same precision since they can be curved. In the following we assume that we have a high resolution triangular mesh obtained from a binary segmented image of the organ we want to simulate (via a Marching Cube algorithm for instance). Our goal is to create a mesh featuring the optimal number of shell elements while staying as close as possible to our targeted geometry.

Therefore we need to insure that the distance between the surface of our shell-based mesh and the targeted high resolution mesh will be minimal. An efficient technique for measuring the error between two surfaces is the Hausdorff distance [12,13]. As a reminder the Hausdorff distance between two meshes is the maximum between the two so-called one-sided Hausdorff distances:

$$d_{\mathrm{H}}(X,Y) = \max\left\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\right\} \quad . \tag{5}$$

where $d()$ is the Euclidian distance between two points. The same technique of subdivision used for rendering allows us to sample the actual surface described by the shells to compute the Hausdorff distance with the targeted high resolution mesh.

The first step in the process of generating a shell-based mesh is a important decimation of the high resolution mesh, using quadric edge collapse technique implemented in Meshlab [14]. The algorithm tries as much as possible to preserve mesh boundaries and generates high quality triangular elements. We then apply a heuristic method derived from the work of Saupin et. al [15] with tetrahedral meshes based on simple geometrical rules. For each node of the coarse mesh, we find the three closest triangles on the high resolution mesh and we move the node to the barycenter of the three centres of mass of those triangles. This technique locally smoothes the surface of the mesh while converging towards the desired high resolution mesh. At each iteration of this algorithm we measure the error between the curved surface of shells and the target using the Hausdorff distance and the process is stopped when the required precision has been reached. A simple example is shown Fig. 3 to illustrate the method.

4 Results

Meshing of anatomical structures This approach has been applied to approximate more complex anatomical geometries with curved shell elements. In each case the error is expressed as a percentage of the diagonal of the object's bounding box.

Computation times We perform several tests on the aneurysm model at different resolutions. The shells are resisting to a uniform pressure load and are solved

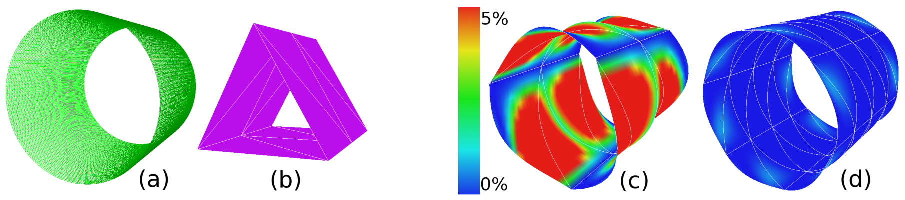

Fig. 3. The target (a) is a high resolution cylinder mesh of 16,384 triangles and we start from a very coarse mesh (12 triangles) approximating the shape of a cylinder, rendered with flat triangles here (b). In (c) the same coarse mesh is rendered with shells and a one-sided Hausdorff distance colour map is applied to show the initial error with the high resolution mesh. (d) One-sided Hausdorff distance colour map after only one iteration of our algorithm (48 shells).

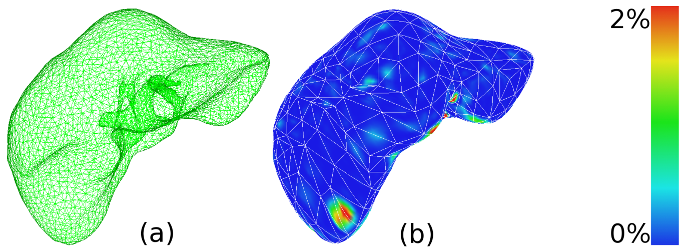

Fig. 4. (a) the targeteted high resolution Glisson's capsule mesh (8,000 triangles). (b) the one-sided Hausdorff distance error map after applying only one iteration of our algorithm to the coarse mesh (1,200 shells).

using a Conjugate Gradient (CG) iterative solver. The computation times are reported in Fig. 6. Implicit integration allows for large time steps (40ms) and the computation is real-time for 800 shell elements and a reasonable error criterion (5%). When the computation time must be bounded (critical real-time applications), one can fix the number of CG iterations to, for instance, 100 and remains real-time for 1000 shell elements. However, in that case the accuracy of the results is not checked.

Coupling between tetrahedra and shells for advanced modelling Structures in human body can be either solid (brain, liver, prostate etc.) or hollow (blood vessels, colon, stomach etc.). However knowing how to model the two kind of structures is not sufficient to reach a high degree of accuracy in medical simulation. Real life situations are more complex. As an example, the external surface of the liver is covered by a layer of cells called Glisson's capsule. Its interaction with the liver plays an important role into the overall structure's mechanical behaviour. Therefore considering the interaction between solid and hollow objects is as crucial as modelling the two structures separately.

An example of medical procedure to illustrate this point even further is angioplasty. Angioplasty is the technique of mechanically widening a narrowed or obstructed blood vessel, typically as a result of atherosclerosis. An empty and

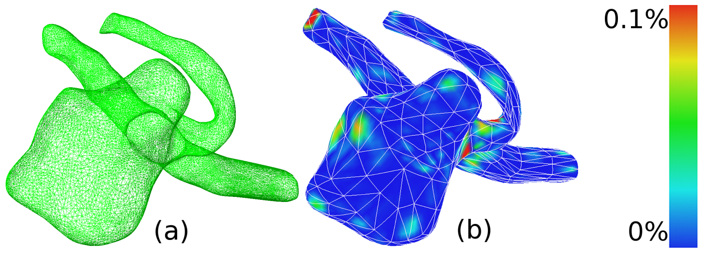

Fig. 5. (a) the targeteted high resolution aneurysm mesh (28,368 triangles). (b): the one-sided Hausdorff distance error map on a mesh of 772 shells generated with our method.

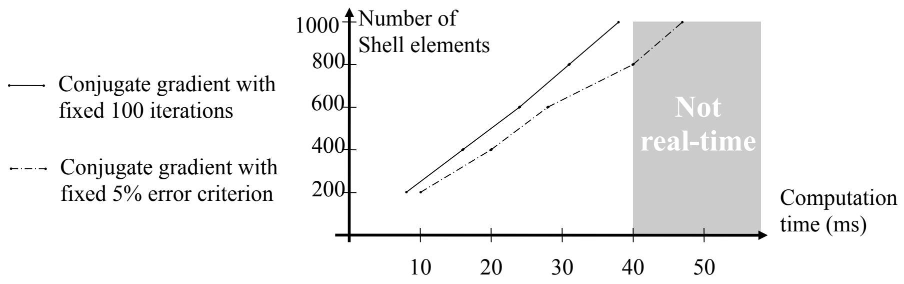

Fig. 6. Computation time on meshes of the aneurysm model with 200, 400, 600, 800 and 1000 elements.

collapsed balloon on a guide wire is passed into the narrowed locations and then inflated to a fixed size. The balloon crushes the fatty deposits, so opening up the blood vessel to improved flow. As a proof of concept we tried to simulate an angioplasty (Fig. 7). The blood vessel is modelled using the shell FEM formulation described in this paper and the fatty deposits are simulated with a tetrahedral FEM method and are fixed to the interior wall of the blood vessel. When the balloon inflates it crushes the deposits and they then apply a pressure onto the curved surfaces of shells modelling the interior wall. The forces are then distributed onto the mechanical nodes of the blood vessel mesh as detailled in section 2, which widens the blood vessel as expected.

5 Conclusion

We propose a framework for real-time modelling of thin anatomical structures. The novelty of our method relies on the combination of a shell finite element formulation and a geometric surface reconstruction both based on the same polynomial interpolation function used to describe the surface of shells. We also show how contacts and interactions with the curved surfaces of shells can be handled using the same function. The efficiency of the method is illustrated through shell-based reconstruction and real-time simulation of the deformations of various anatomical structures. We also present preliminary results on the simulation of an angioplasty procedure.

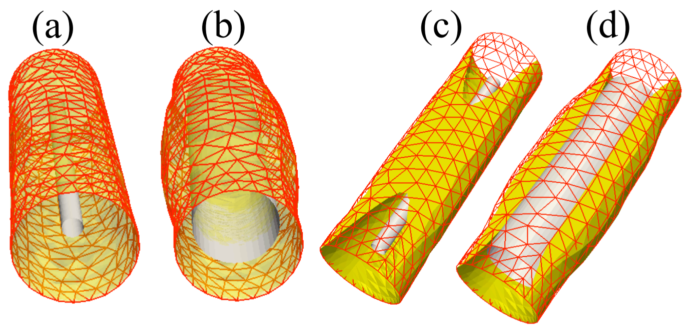

Fig. 7. Simulation of an angioplasty procedure. (a, c): A collapsed stent is inserted into the blood vessel simulated with our shell FEM formulation. The fatty deposits (in yellow) are modelled with tetrahedral FEM. (b, d): The stent is crushing the fatty deposits which creates a pressure onto the interior wall and widens the blood vessel.

References

1. Miller, K., Joldes, G., Lance, D., Wittek, A.: Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. Communications in Numerical Methods in Engineering **23**(2) (2007) 121–134
2. Delingette, H.: Biquadratic and quadratic springs for modeling st venant kirchhoff materials. In: Proceedings of ISBMS 2008. (2008) 40–48
3. Liu, G., Quek, S.: Finite Element Method: A Practical Course. Butterworth (2003)
4. Nedel, L.P., Thalmann, D.: Real time muscle deformations using mass-spring systems. In: Proceedings of CGI. (1998) 156
5. Hammer, P.E., Perrinb, D.P., del Nidob, P.J., Howe, R.D.: Image-based mass-spring model of mitral valve closure for surgical planning. In: Proceedings of SPIE Medical Imaging. Volume 6918. (2008)
6. Choi, M.G., Woo, S.Y., Ko, H.S.: Real-time simulation of thin shells. In: ACM SIGGRAPH/Eurographics symposium on Computer animation. (2007) 349–354
7.
8. Kolb, A., Pottmann, H., peter Seidel, H.: Metro: Measuring error on simplified surfaces. Fair Surface Reconstruction Using Quadratic Functionals **14**(3) (1995) 469–479
9. Douros, L., Buxton, B.: Three-dimensional surface curvature estimation using quadric surface patches. In: Proceedings of Scanning. (2002)
10. Tang, C.K., Medioni, G.: Robust estimation of curvature information from noisy 3d data for shape description. In: Proceedings of IEEE International Conference on on Computer Vision. (1999)
11. Przemieniecki, J.: Theory of matrix structural analysis. McGraw-Hill (1985)
12. Klein, R., Liebich, G., Straßer, W.: Mesh reduction with error control. In: Visualization 96. ACM. (1996) 311–318
13. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. Computer Graphics Forum **17**(2) (1998) 167–174
14. CNR, V.C.L.I.: Meshlab http://meshlab.sourceforge.net/.
15. Saupin, G., Duriez, C., Grisoni., L.: Embedded multigrid approach for real-time volumetric deformation. In: International Symposium on Visual Computing. (2007) 149–159