

Shell Model for Reconstruction and 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 tissue can be abstracted with a 2D modelling of their surface 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

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. To realise this kind of interactive systems, one faces the trade-off problem between accuracy of the model and the need of fast computations. 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 even if there are obvious advantages to make maximum use of optimised generic models. Yet most of previous work focus on volumetric models that are able to capture the behaviour of solid organs like liver [1] or brain tissues [2] **CD: TODO: biblio**. In contrast, this paper seeks to propose a solution for simulating, close to real-time, deformation of thin anatomical structures whose volume is negligible compared to their surface area. Examples of such anatomical structures include hollow structures, like the wall of blood vessels or the colon but also the envelope of solid organs, like the capsule of Glisson.

Shell theory allows the modelling of structure deformations when the thickness is small compared to its other dimensions [1]. The key idea is to model the physical shell as if it were a surface of zero thickness, but endowed with mechanical properties in the form of elastic resistance to stretching and bending forces.

OC: Pas de shearing il me semble. Et il y a bien une thickness dans la formulation, ta phrase laisse sous-entendre qu'on la prend pas en compte. Rather than resorting to shell theory, previous work on medical simulation often relies on linear or angular mass-spring models as in [2,3]. 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 model for computer graphics and interactive simulation was recently introduced with the work of Choi et al. [4]. The model relies on simplified energy functions and precomputed modal analysis. It allows fast and visually realistic results. We propose to rely on a similar approach but with more accuracy in order to be applicable to medical simulation. Our model is not based on modal analysis but uses a co-rotational formulation and polynomial shape functions. In [5], we present and apply this shell model to a flat deformable implant. **OC: ptet un peu trop specifique pour preserver l'anonymat non ?**

In this paper, we propose to apply this technique for modelling the deformations of some anatomical structures. To be able to do this, one of the main problem is to geometrically model the surface of the structures using curved triangular shell elements. This problem 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 in this field. One of the most common approach is to use continuous surface patches [6,7]. The surface is locally approximated by an analytic representation called surface patch, usually chosen to be an implicit surface. In [8] an elegant and non noise-sensitive approach that infers sign and direction of principal curvatures directly from the input is presented and the authors use this information for coherent surface and curve extraction from 3D data. However, our situation is substantially different as extracting the surface is not our ultimate goal. We want to model the deformation of the structure and in that regard the curvature of the surface has a physical meaning: it represents the rest shape of our anatomical structure, that is the configuration with zero energy. Therefore we need to match the shape (polynomial) functions used in our shell FEM formulation to describe the surface of the shells with the surface of the anatomical structures that we wish to simulate.

Our contribution is to unify these two approaches (shell based modelling and geometric surface reconstruction) separately used in mechanical engineering and computer vision in a framework for realistic simulation of thin organ deformations. In addition, we demonstrate the ability of the technique to simulate thin anatomical membrane which enfolds a solid organ without any requirement on mesh correspondence. **OC: j'enleverais bien cette derniere phrase moi, on en fait pas la demonstration au final, on evoque juste que ca serait possible** Our method, presented in section 2, is based on shell theory and computes the internal forces using Finite Element Modelling (FEM) on curved triangles. The design of the elements relies on a polynomial interpolation function that is

also used to capture and process the interactions with other models, including solid models. In section 3 we also present an automatic process to obtain meshes from image based reconstruction that relies on the exact same polynomial function. Finally the benefits of our approach (meshing of a curved surface, fast computation and possible interactions with solid models) are illustrated using various examples presented 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 [5]. Therefore we will only remind briefly the key points. We improved and extended a plate model first introduced by Przemieniecki [9] 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.

2.1 Triangular elastic membrane

The element stiffness matrix \mathbf{K}_e can be computed as follows:

$$\mathbf{K}_e = \int_v \mathbf{J} \boldsymbol{\chi} \mathbf{J}^T dV \quad (1)$$

where \mathbf{J} is the strain-displacement matrix and $\boldsymbol{\chi}$ embodies the material's behaviour. The stiffness matrix in the global frame is eventually obtained using the rotation matrix of the element: $\mathbf{K} = \mathbf{R}^T \mathbf{K}_e \mathbf{R}$ where \mathbf{R} describes the rotation of the (triangular) element with respect to its initial configuration.

2.2 Triangular plate bending

To calculate the stiffness matrix for the transverse deflections and rotations shown on figure 1, we start with the assumed deflection u_z of the form

$$u_z = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8xy^2 + c_9y^3 \quad (2)$$

where c_1, \dots, 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 $\mathbf{u} = \{u_1 u_2 \dots u_9\}$ of the displacements and slopes at the three corners of the triangular plate using the following notations:

$$u_1 = (u_z)_{x_1, y_1} \quad u_2 = \left(\frac{\partial u_z}{\partial y} \right)_{x_1, y_1} \quad u_3 = - \left(\frac{\partial u_z}{\partial x} \right)_{x_1, y_1} \quad (3)$$

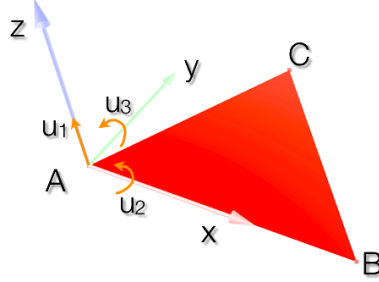


Fig. 1. The different degrees of freedom u of a triangular thin plate in bending are illustrated above.

and so on for the two other vertices. We then can derive a matrix \mathbf{C} such as $\mathbf{u} = \mathbf{C}\mathbf{c}$ where $\mathbf{c} = \{c_1 c_2 \dots c_9\}$.

We can calculate the strains from the flat-plate theory using:

$$e_{xx} = -z \frac{\partial^2 u_z}{\partial x^2} \quad e_{yy} = -z \frac{\partial^2 u_z}{\partial y^2} \quad e_{xy} = -2z \frac{\partial^2 u_z}{\partial x \partial y} \quad (4)$$

Symbolically this may be expressed as $\mathbf{e} = \mathbf{D}\mathbf{c}$ where \mathbf{D} derives from equations 2 and 4. Noting that $\mathbf{c} = \mathbf{C}^{-1}\mathbf{u}$, we have:

$$\mathbf{e} = \mathbf{D}\mathbf{C}^{-1}\mathbf{u} = \mathbf{b}\mathbf{u} \quad (5)$$

where the strain-displacement matrix $\mathbf{b} = \mathbf{D}\mathbf{C}^{-1}$. The stiffness matrix \mathbf{K}_e for an element is then obtained from:

$$\mathbf{K}_e = \int_v \mathbf{b}^T \boldsymbol{\chi} \mathbf{b} dV \quad (6)$$

where $\boldsymbol{\chi}$ is the material matrix. The stiffness matrix in the global frame is eventually obtained using the rotation matrix of the element as in the membrane case.

2.3 Mechanical interactions with the curved surface of shells

But 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 (2) 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 (2) and 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) and detect any collision with the curved surface of the shell using any of the classic collision detection algorithms working on flat triangles.

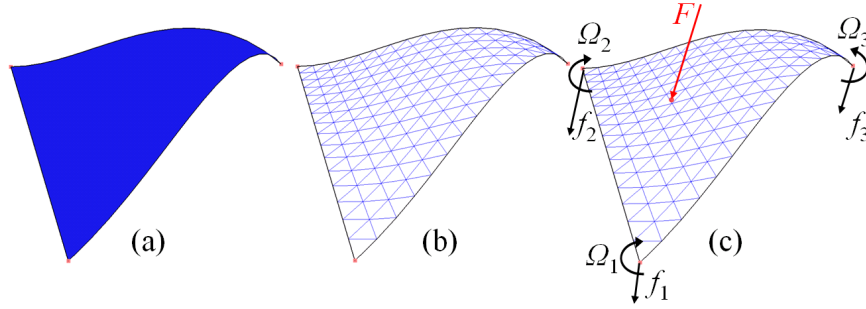


Fig. 2. The triangle formed by the three vertices of the shell has been recursively subdivided 3 times and the deflection of each new vertex was computed according to the same deflection function used in our shell finite element formulation. Sampling the actual surface of the shell allows rendering and more accurate collision detection.

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. Our approach is the following: during force computation, we use the angle 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 polynomial $P = F_z(1 \ x \ y \ x^2 \ xy \ y^2 \ x^3 \ xy^2 \ y^3)^T$. The moments at each vertex are then obtained with $\Omega = (C^{-1})^T P$. Thus we are able to transmit any force coming from elaborate interactions with the curved surface of shells to the mechanical vertices used in our FEM formulation.

3 A mechanically-driven method to mesh curved surfaces with shells

Because the surface of an anatomical structure has a physical meaning, we cannot use any type of function to reconstruct this surface. The surface must be patched 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 often 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 naturally 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 [10,11]. As a reminder the Hausdorff distance between two meshes is the maximum between the two so-called one-sided Hausdorff distances:

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\} . \quad (7)$$

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 an aggressive simplification of the targeted high resolution mesh by using the quadric edge collapse decimation technique implemented in Meshlab [12]. The algorithm tries as much as possible to preserve mesh boundaries and generate high quality triangular elements. We then apply a heuristic method derived from the work of Saupin et. al [13] with tetrahedral meshes based on very 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 smooths 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. We can notice that differences with the target mesh have been substantially reduced.

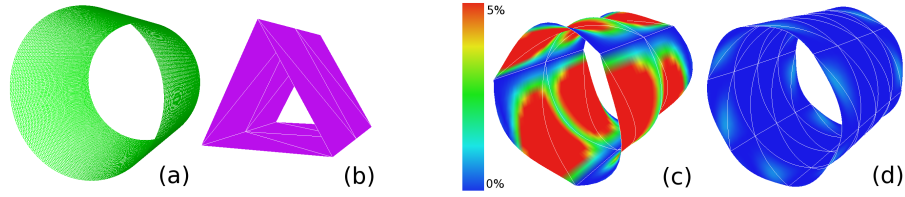


Fig. 3. Illustration of our method on a simple example. The target is a high resolution cylinder mesh of 16,384 triangles (a) and we start from a very coarse mesh (12 triangles) approximating the shape of a cylinder, rendered with flat triangles here (b). In (c) the same coarse mesh is rendered with 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).

4 Results

4.1 Meshing of anatomical structures

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

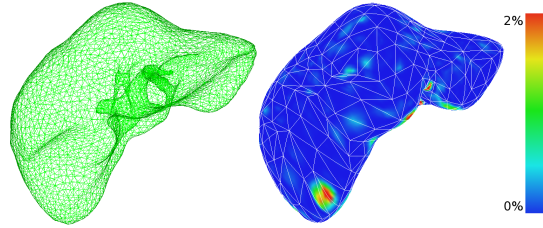


Fig. 4. Left: the targeted high resolution liver mesh (8,000 triangles). Right: the one-sided Hausdorff distance error map after applying only one iteration of our algorithm to the coarse mesh (1,200 shells)

4.2 Computation times

Aneurysm: 7.5 FPS with a timestep of 0.02s so we would need to be about 7 times faster to be real-time (easily doable on GPU)

4.3 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

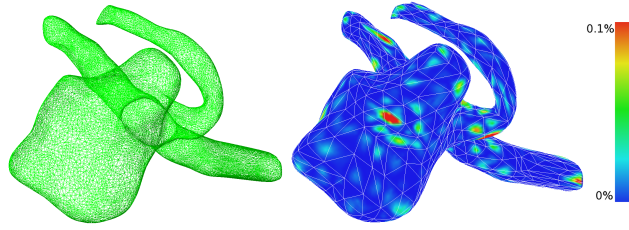


Fig. 5. Left: the targeted high resolution aneurysm mesh (28,368 triangles). Right: the one-sided Hausdorff distance error map after applying only one iteration of our algorithm to the coarse mesh (1,160 shells)

two kind of structures is not sufficient to reach a high degree of accuracy in medical simulation. Indeed real life situations are more complex. As an example, the external surface of the liver is covered by a single 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 collapsed balloon on a guide wire, known as a balloon catheter, is passed into the narrowed locations and then inflated to a fixed size using water pressures. 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. 6). 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 the deposits 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 detailed in section 2.3, which widens the blood vessel as expected.

5 Conclusion

Acknowledgments

References

1. Liu, G., Quek, S.: Finite Element Method: A Practical Course. Butterworth (2003)
2. Nedel, L.P., Thalmann, D.: Real time muscle deformations using mass-spring systems. In: Proceedings of Computer Graphics International. (1998) 156
3. 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: Proc. of SPIE Medical Imaging. Volume 6918. (2008)

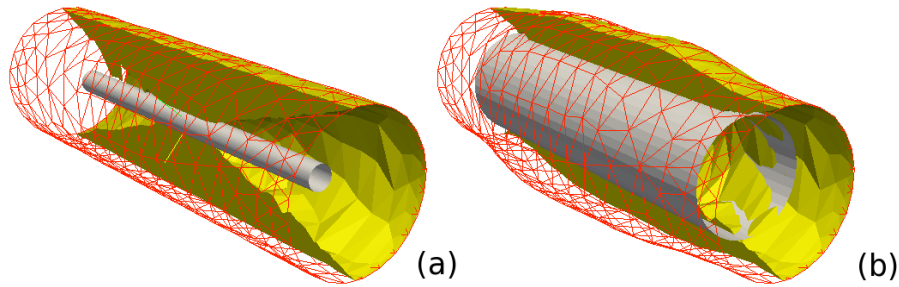


Fig. 6. Simulation of an angioplasty procedure. (a) The collapsed balloon was inserted into the blood vessel simulated with our shell FEM formulation. The fatty deposits (in yellow) are modelled with a tetrahedral FEM. (b) Upon inflation the balloon is crushing the fatty deposits, which is applying a pressure onto the interior wall and widening the blood vessel.

4. 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
- 5.
6. Kolb, A., Pottmann, H., Peter Seidel, H.: Metro: Measuring error on simplified surfaces. *Fair Surface Reconstruction Using Quadratic Functionals* **14**(3) (1995) 469–479
7. Douros, L., Buxton, B.: Three-dimensional surface curvature estimation using quadric surface patches. In: *Scanning 2002 Proceedings*. (2002)
8. Tang, C.K., Medioni, G.: Robust estimation of curvature information from noisy 3d data for shape description. In: *Proceedings of IEEE International Conference on Computer Vision*. (1999)
9. Przemieniecki, J.: *Theory of matrix structural analysis*. McGraw-Hill (1985)
10. Klein, R., Liebich, G., Straßer, W.: Mesh reduction with error control. In: *Visualization 96*. ACM. (1996) 311–318
11. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* **17**(2) (1998) 167–174
12. CNR, V.C.L.I.: Meshlab <http://meshlab.sourceforge.net/>.
13. Saupin, G., Duriez, C., Grisoni, L.: Embedded multigrid approach for real-time volumetric deformation. In: *International Symposium on Visual Computing*. (2007) 149–159