

Animating cuts with on-the-fly re-meshing

F. Ganovelli and C. O'Sullivan

Image Synthesis Group, Computer Science Department, Trinity College Dublin, Ireland

Abstract

The problem of defining a model for deformable objects which allows the user to perform cuts is still open. Generally speaking, the reason is that such a task affects the connectivity and the topology of the mesh, while the assumption that they never change is the basis of most algorithms for both computation of deformation and collision detection. The drawback of approaching this problem as one of cutting based on re-meshing, is that the mesh exhibits a higher density where it has been cut than elsewhere. This paper proposes an on-the-fly tetrahedral simplification scheme to cope with such a fragmentation problem.

1. Introduction

To model deformable objects is still a challenge which provides many opportunities for research, thanks to the increasing demands to apply theoretical models to real situations. Especially in application domains like virtual surgery, the model should enable the user to perform cuts onto the object, while such a feature is not considered in most of the previous approaches. This paper is mainly an extension of the approach for cutting tetrahedral meshes proposed in ⁵. We show experimentally that the badly shaped tetrahedra created by the referred approach can be deleted from the mesh without any notable effect on the simulation, hence we introduce an on-the-fly edge collapse simplification that prevents the creation of many degenerate tetrahedra. The rest of the paper is organized as follows. In Section 2 we review the previous work on the subject, outlining advantages and drawbacks of every proposed approach and Section 3 briefly reviews our approach to the problem (for the extended version see ⁵). Section 4 introduces the extension we developed to cope with the problem of fragmentation and finally Sections 5 and 6 present the results of the proposed technique and future work, respectively.

2. Previous work

In the common cutting paradigm the user holds a scalpel and moves it towards the surface and then through the volume of the object itself, producing a cut on the object. This means that the representation of the object, always built in a pre-processing phase, has to be updated coherently with the cut performed. In this paper we focus attention on how the rep-

resentation of the object could be changed according to a cut action.

So far three kinds of solutions have been proposed to manage the representation of cuts and to update the mesh representing the object:

1. simply delete from the mesh the cells that are touched by the scalpel (see Figure 1.a). This approach is proposed in ² for tetrahedral meshes: using a discrete model for physical simulation, the elimination of a tetrahedron requires few updating operations. These are performed on the parameters of the system related to the tetrahedra sharing one of the four vertices of the deleted tetrahedron. Unfortunately, both the visual effect and the simulation accuracy strictly depend on the density of the tetrahedra. Basically the problem is that the object is not really cut but partially erased from the scene.
2. duplicate the nodes which are near to the scalpel and remove adjacency relations (see Figure 1.b): firstly, the mesh nodes closest to the trajectory of the scalpel are moved so that the set of internal faces sharing such vertices are on the trajectory; secondly the moved vertices are duplicated (as well as the faces they define) and the resulting mesh is conformed to the cut. This approach does not lead to loss of volume, as was the case in the previous technique. Furthermore the number of node cells remain constant after the cut. In ⁶ the authors use this scheme but do not move the nodes, therefore obtaining a jagged look. An advantage of this approach seems to be that finite element analysis could be used, since the duplication of nodes only requires a local modification to

the rows of the stiffness matrix related to such nodes. Actually, to move the nodes to match the trajectory of the scalpel means changing the resting position of the mesh, hence invalidating the computations for the cells affected by such displacements. Furthermore, as for the previous technique, the results depend on the density of the cells, since the bigger the cells are the more the displacement of their nodes affect the shape of the mesh.

- replace the cells that are cut with a new set of cells (see Figure 1.c). This approach is similar to the first one, except than the deleted cells are replaced with a set of cells occupying the same volume but with no proper intersection with the trajectory of the scalpel. This solution, first proposed in ¹ for tetrahedral cells, offers the most pleasant visual results, even if the strategy used to find the new set of tetrahedra causes a high fragmentation of the mesh and degenerate tetrahedra.

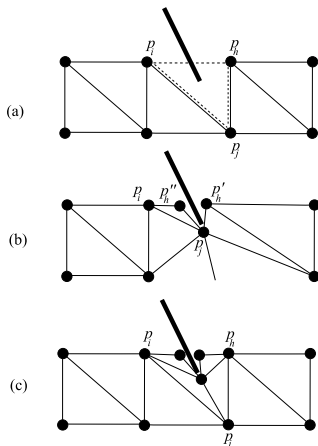


Figure 1: Three strategies for implementing cuts: (a) deleting the cells intersected by the scalpel; (b) adapting the shape of the mesh; (c) splitting the cells intersected into more cells

We chose this last approach to manage cutting actions and we proposed a solution that limits the increase of the mesh size w.r.t. the previous approach, briefly described in the next section.

3. Review of our approach

Our approach, presented in ⁵, consists of replacing the tetrahedra intersected by the surface defined by the movement of the scalpel with a set of tetrahedra occupying the same volume and with no proper intersections with such a surface. We compute an integer value by determining which edges are intersected by the scalpel and use this value to access a Look Up Table storing the set of tetrahedra to use for replacement (details can be found on the paper previously referred to in ⁴). Although this technique modifies the mesh with neither loss of volume nor vertex displacement and it

gives an accurate representation of the cut, it also has two drawbacks connected with the generation of new tetrahedra:

- Fragmentation Problem. Every cut tetrahedron is replaced by a number of tetrahedra ranging from 4 to 9, depending on how it has been traversed by the scalpel. This gives rise to a fragmentation of the mesh around the walls created by the cut, resulting in a mesh with non-uniform complexity.
- Degenerate Tetrahedra. The shape of the new tetrahedra depends solely on the intersection points between the trajectory of the scalpel and the edges of the cut tetrahedra (see Figure 2, so a every kind of degenerate tetrahedron is likely to be created.

4. On the fly edge-collapse to eliminate mesh fragmentation

We propose a simple and effective strategy to solve both the problems above simultaneously: an on-the-fly edge-collapse simplification of the mesh. Edge collapsing provides a simple way of simplifying simplicial complexes in d dimension. It consists of unifying the extremes of an edge and then updating the mesh, deleting from the data structure all the simplexes that share the collapsed edge. This technique has the nice property of avoiding an explicit re-triangulation phase, which marks an important difference from the approaches based on vertex elimination and enables more efficient processing.

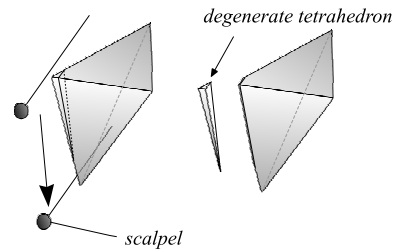


Figure 2: Example of how a degenerate tetrahedra is created. The scalpel intersects the edges very close to the vertices.

We are in general not free to select any edge to be collapsed because we are interested in preserving the topology and the quality of the mesh. For example, the collapse of an edge which is on the border of a hole can cause the removal of the hole from the mesh and a change of the mesh topology (see Figure 3). Furthermore, not all of the edges can be collapsed while still maintaining the properties of the input simplicial complex or the quality of the decomposition. A 2D example is shown in Figure 4: the collapse of edge (a,b) introduces two degenerate cells (a,c,d) and (c,b,d) each having zero area.

The topological correctness of an edge collapse can be ensured using the *link conditions* ³.

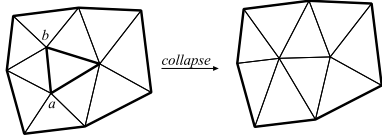


Figure 3: The collapse of the edge (a, b) causes the removal of the hole and a change in the mesh topology.

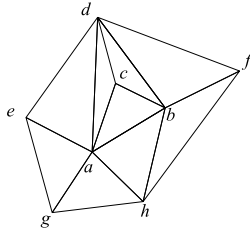


Figure 4: The collapse of edge (a, b) does not respect the topological conditions and is therefore rejected; this prevents a mesh update that would have introduced degenerate cells.

4.1. Driving simplification

We use edge-collapse as a tool for re-meshing the part of the solid involved in the cut. The domain of that re-meshing consists of all the tetrahedra created during the cut, so the candidate collapses i.e. the set of edges that possibly will be collapsed, corresponds to the edges of such tetrahedra. The approach is straightforward: we only have to find out which are the badly shaped tetrahedra and which of their edges to collapse in order to remove them from the mesh. In order to proceed as fast as possible, the collapse is done by move one of the extremes of the edge one the other. Generally speaking, the strategy of collapsing the edge in the point along the edge that optimizes the resulting mesh would be preferable, but it requires more time. Furthermore, as can be seen below, the collapsed edges are always very short and so the location of the point where the edge is collapsed is negligible.

A measure of the quality of a tetrahedron is the *aspect ratio*, i.e. the ratio of the radius of the inscribed sphere to the radius of the circumscribed sphere. This has a maximum value of $(\frac{1}{3})$ for a regular tetrahedron. We will refer to the quality as the value $3 \times \text{aspectratio}$ (i.e. between 0 and 1).

Every time a new tetrahedron σ is created we apply the following steps:

1. compute its quality $Q(\sigma)$
2. if $Q(\sigma) < \tau$ then add the pair $\langle e(\sigma), Q(\sigma) \rangle$ to H

where $e(\sigma)$ is the shortest edge of the tetrahedron and H is a heap of pairs with the ordering relation on the second element. So the root of the heap contains the shortest edge

of the tetrahedron with the minimum quality.

The reason to collapse the shortest edge first is that we want to delete the badly-shaped tetrahedron with the minimum modification to the rest of the mesh.

We store the edge to be collapsed because it is not possible to actually collapse an edge when the related tetrahedron is created. The reason is that during the cut the mesh is not a simplicial complex, as can be seen in Figure 5. This condition must hold, at least in the region of influence of the edge to be collapsed.

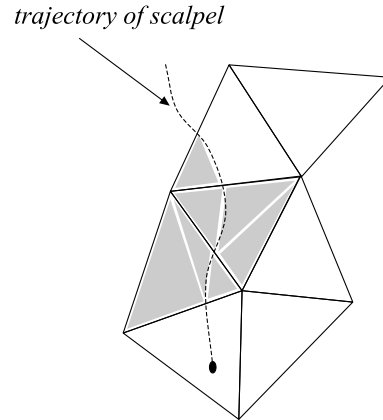


Figure 5: As long the scalpel does not leave the tetrahedron, it cannot be replaced because not all intersections between the trajectory of the scalpel and the edges are known.

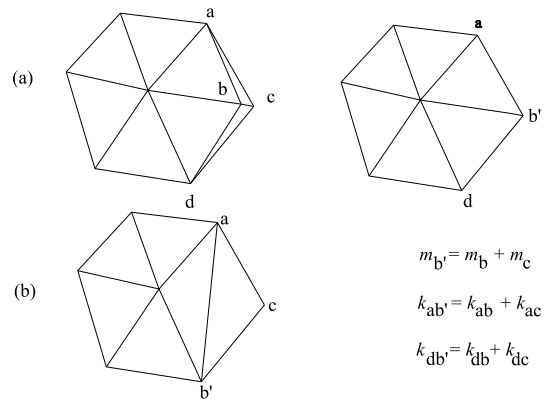


Figure 6: Elimination of the simplex (b, c, d) . To collapse the edge (b, c) corresponds to a smaller change in the system than to collapse any other edge of the same simplex.

Changes on the Mass Spring System

In the best case the average volume of the tetrahedra created by the cut (before edge collapse) is $1/4$ of the volume of the original tetrahedra. We studied experimentally the relationship between the quality of tetrahedra and their volume

in this context, iterating several cutting operations and computing the density distribution of the volume over the badly shaped tetrahedra (see Figure 7 for details). As expected, the badly shaped tetrahedra have almost 0 volume (note that this is not true in general). Therefore their contribution to the mass values of the vertices and to the stiffness of their edges is almost 0 as well and could be neglected without any notable difference.

Figure 6.a shows how the values of the mass spring system are modified after an edge collapse. The mass of the moving node is simply added to the other extreme of the edge and the same is done with the springs that collapse together. Note that this can be done because we choose the shortest edge of the degenerate tetrahedron, therefore the rest of the mesh is almost unchanged. It would be very different if, as Figure 6.b shows, we did not choose the shortest edge.

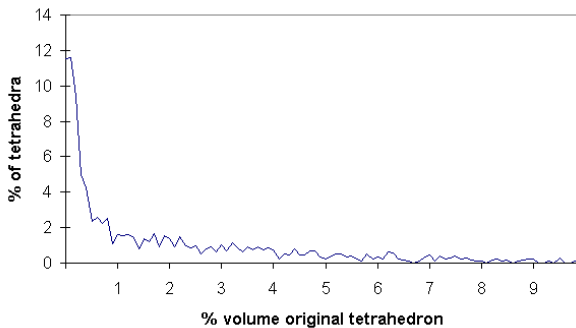


Figure 7: The density distribution of the volume of the created tetrahedra whose quality is under 0.3

5. Results

We evaluate the results of our approach in terms of performance and also quality and size of the resulting mesh. In particular we compare the mesh resulting from a cut operation with or without applying on-the-fly edge collapse. Table 1 reports the time required for updating the mesh as the sum of the time required for replacing tetrahedra plus the time required by re-meshing by means of edge collapse. It should be noted that the algorithm is naturally suited to a time-critical implementation: the only thing required at the time of cut is to store the edges to be collapsed on the heap, while the actual collapses can be done later on.

Table 2 shows the improvement of the mesh due to the edge-collapse. We measured quality with two parameters (the visual effect can be seen in Figure 9): the number of tetrahedra and the density distribution of the quality (see graph 8). The table has two rows for each cut: one with the quality values if no re-meshing is applied and one when edge collapse is applied. The data in this table relate to the part of the mesh involved in the cut (neither the tetrahedra replacement nor the re-meshing affect the rest of the mesh).

#CuT	#CrT	#RmT	t.Repl.	t.EC
85	455	335	254	651
103	536	258	273	431

Table 1: Times are expressed in milliseconds and refer to a Pentium II 1Ghz 256MB RAM. CuT = cut tetrahedra, CrT = created tetrahedra, RmT = removed tetrahedra, t.Repl = time for replacing tetrahedra, t.EC = time for removing tetrahedra. Please note that these time relate to the interval of time where the cut is performed (8 – 10 sec.)

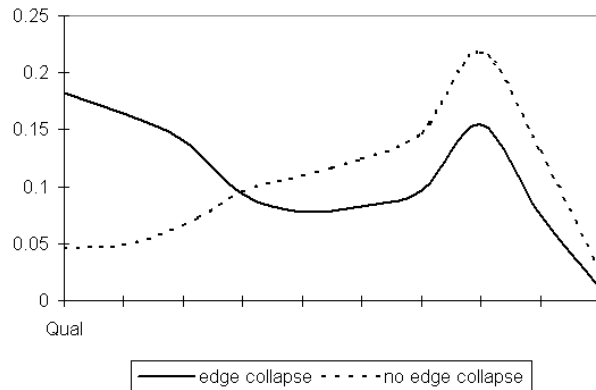


Figure 8: The density distribution of the quality of tetrahedra produced with and without applying edge collapse (dashed and continuous line, respectively). With the edge collapse technique a higher number of tetrahedra have better quality.

6. Future work

In this paper we presented a technique to limit fragmentation and avoid degenerate tetrahedra during a cut operation on a tetrahedral mesh. Currently our approach does not apply to physical modelling through Finite Element Analysis, because it is not clear if and how the re-meshing we use (both tetrahedra replacement and edge collapse) can be mapped on an on-the-fly modification of the stiffness matrix. We are currently orienting our research towards this problem.

Another aspect concerns level of detail: splitting the tetrahedra intersected by the scalpel always produces an accurate result, but when the mesh is fine enough, simply duplicating faces would also achieve this (recall that to duplicate faces does not create new tetrahedra). Therefore, a hybrid approach that uses both techniques depending on the fineness of the representation could be considered.

#cut tetra	#crTet	ave. qual.
85	455	0.28
	240	0.46
137	712	0.35
	324	0.38

Table 2: Examples of cut: the upper row shows the number and the average quality of the tetrahedra if the edge collapse technique is not applied and the lower row shows the same data when it is applied.

References

1. D. Bielser, V.A. Maiwald, and M.H. Gross. Interactive cuts through 3-dimensional soft tissue. *Computer Graphics Forum (Eurographics'99 Proc.)*, 18(3):C31–C38, September 1999. 2
2. H. Delingette, S. Cotin, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *CAS99 Proceedings*, pages 70–81, May 1999. 1
3. T.K. Dey, H. Edelsbrunner, S. Guha, and D.V. Nekhayev. Topology preserving edge contraction. Technical Report RGI-Tech-99, RainDrop Geomagic Inc. Champaign IL., 1999. 2
4. F. Ganovelli. *Multiresolution and Cuts in Deformable Objects Modelling*. PhD thesis, Universita' degli Studi di Pisa, April 2001. 2
5. F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. A multiresolution model for soft objects supporting interactive cuts and lacerations. *Computer Graphics Forum*, 19(3):271–281, aug 2000. 1, 2
6. H.W. Nienhuys and A.F. van der Stappen. Combining finite element deformation with cutting for surgery simulations. In (*Proceedings of Eurographics 2000, short presentations*), pages 43–51, 2000. 1

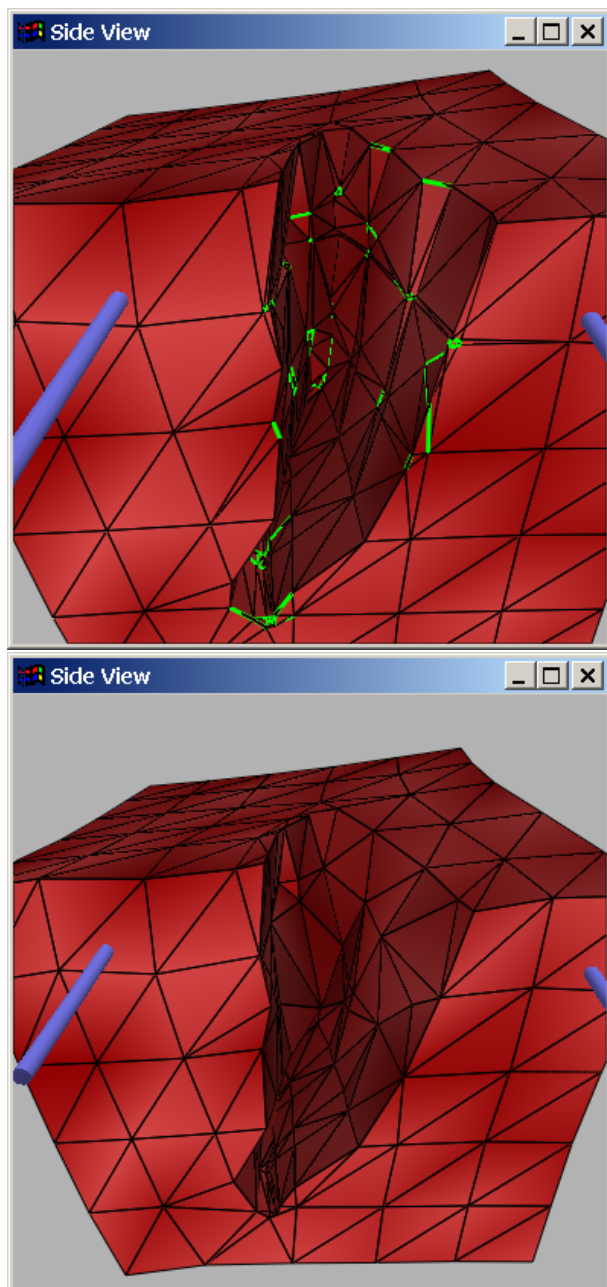


Figure 9: Upper snapshot. A example of cut without apply edge collapse. The edge which would be collapsed are marked green. Note how they correspond to very tiny tetrahedra. Lower snapshot. The same cut applying edge collapse. Note that the mesh is more regular, although it can happen that some edge is topologically not collapsable.