Computer Graphics International 2011

Pitting a New Hybrid Approach for Maintaining Simulation Stability after Mesh Cutting Against Standard Remeshing Strategies

Luis F. Gutiérrez · Iker Aguinaga · Basil Fierz · Félix Ramos · Matthias Harders

Abstract In previous work we have developed a new hybrid approach for handling ill-shaped elements possibly appearing after cutting simulation meshes. In this work we compare the approach with existing standard techniques commonly employed to treat topological changes in deformation meshes. Our results indicate that the hybrid approach might be better suited for real-time simulations than using standard mesh improvement methods, since the latter require an order of magnitude larger processing times.

Keywords cutting · mesh improvement · explicit integration · physically-based simulations

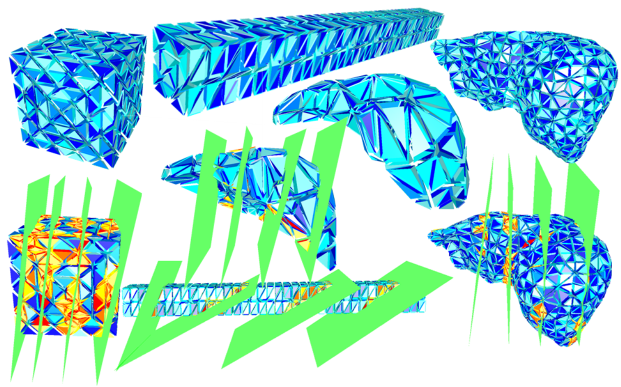

Fig. 1 Meshes used in all tests, from left to right at the top: *cube*, *bar*, $liver_1$ and $liver_2$. For the tests each mesh is dissected using four cutting planes.

1 Introduction

Physically-based simulations of deformable objects have been an important research field in computer graphics since several years. Interactive applications, especially surgery training simulators, require computation of object deformations in real-time. Most current techniques rely on mesh-based representations of objects (e.g. using tetrahedral meshes) [7]. Cutting soft tissue is a common procedure in surgery simulations. This usually involves modification of mesh topology, e.g. by subdividing elements touched by a cutting tool. Unfortunately, this potentially decreases the mesh quality. This especially becomes a problem when explicit time integration schemes are employed to compute deformations. These solvers are only conditionally stable, i.e. the simulation is only stable for time steps below a critical limit. As discussed in [10], this limit is strongly dependent on the shape and size of the mesh elements. A few ill-shaped elements can dramatically reduce the critical time step and thus slow the simulation.

In [3] we have proposed a new *hybrid approach* that relies on using shape matching [6] for all elements whose time step is below a specific threshold (so-called *critical elements*); while all remaining non-critical elements are treated with a standard co-rotational finite element method (FEM). The hybrid approach allows using larger time steps with only a small reduction of accuracy for meshes undergoing cutting. In contrast to this, the more common approach to treating ill-shaped elements in real-time cutting is the optimization of the simulation mesh (see e.g. [5,8,11]).

In this brief study we focus on a performance analysis of the new hybrid approach in the context of mesh optimization. We compare the effect in the scope of interactive simulations when using either a standard deformation computation with mesh improvement techniques or the new hybrid approach. In the following we

CINVESTAV Guadalajara
E-mail: lgutierr@gdl.cinvestav.mx

ETH Zurich, Computer Vision Laboratory
E-mail: mharders@vision.ee.ethz.ch

will first provide further details on the hybrid method, as well as on the used mesh improvement techniques. Note that we employ for the deformation computation a co-rotational FEM approach that is solved with an explicit integrator. Various alternatives to this exist; however, a discussion of these is out of the scope of this work.

2 Hybrid Approach

A key element of the hybrid technique is the reliable identification of mesh elements that would become unstable for a given simulation time step. The maximum stable time step or *critical time step* Δt_{crit} of a complete mesh can be computed for a given explicit integration approach using:

$$\Delta t_{crit} = \frac{c}{\sqrt{\lambda_{max}(\mathbf{M}^{-1}\mathbf{K})}}, \tag{1}$$

where $\lambda_{max}()$ returns the largest eigenvalue of a given matrix, $\mathbf{M}$ is the mass matrix and $\mathbf{K}$ the system stiffness matrix of the finite element model. The constant c is specific for a numerical integration method, for instance $c = 2$ for semi-implicit Euler [6]. While this formula does provide the overall critical time step, it does not identify which ill-shaped elements mainly govern a possibly low threshold. Therefore, we have proposed in [3] a number of approaches to determine these elements. The key idea is to approximate the global critical time step in a local neighborhood. In the *reduced 1-ring* approach only the direct neighbors of an examined element are considered. A 12×12 sub matrix based on the mass and stiffness matrices of the local neighborhood is extracted. Using this matrix in Eqn. 1 yields an estimate of the critical time step for the considered element. This for instance allows determining which elements would become unstable for a desired simulation time step. In addition, using this heuristic an approximation of Δt_{crit} can be made by testing all mesh elements.

In the second step of the hybrid technique, the elements identified as critical for a given time step are treated by a geometric deformation model. The resulting displacements are blended with the restitution forces computed for all stable elements via the standard FEM. This hybrid approach allows taking much larger time steps than when using only a standard explicit FEM. Moreover, the total computational costs per second are significantly lowered. More details on this approach can be found in [3].

3 Mesh Optimization Techniques

For the comparison of the hybrid technique with standard mesh improvements we have implemented and combined a number of techniques published in the literature. Note that for our experiments we only consider non-progressive cutting. Also, a cut through a test mesh is specified by an arbitrary plane.

After a cut surface is defined, elements are first processed following the strategy outlined in Steinemann et al. [11]. This step includes *subdivision* of elements as well as *node snapping* for mesh vertices close to the cut plane. This is the initial basic mesh update that results in a new valid tetrahedral mesh.

Following this, a number of mesh improvement steps are carried out. First, edges on the contour of the newly created cutting surface are processed. Vertices of small edges are displaced on the contour in order to increase the volume of incident elements. During these changes special care is taken to not produce inverted elements. In addition to that, various topological mesh improvements are applied, such as *multi-face removal*, *edge removal*, and *basic flips* (see [2,4] for further detail). These operations are based on a characteristic length L_e that is defined for a tetrahedron as $L_e = 3V_e/\max A_e$. Here, V_e is the element volume and $\max A_e$ the maximal area of the tetrahedron faces [1]. Using this quality metric produced better results than other possible alternatives (e.g. minimum sine, volume length, radius ratio) proposed in [4]. Finally, we also apply a *smoothing* algorithm in the local neighborhood of a cut. Mesh vertices are displaced by a combined mass-spring and particle simulation in order to regularize element shape [9]. By using a highly damped system only a small number of simulation steps have to be carried out. All parameters in the mesh improvement pipeline have been optimized for best performance on our test cases.

4 Results

For our comparisons four tetrahedral example meshes have been examined (see Fig. 1 and Tab. 1). The corotational FEM was solved with a semi-implicit Euler. For all meshes a Young's modulus of 30 kPa, density of 1000 kg/m^3, and Poisson ratio of 0.3 was used. All tests were performed on a PC running under Windows 7 with an Intel Core i7-720 1.6 GHz processor.

The initial critical time step has been determined for each mesh by exact computation via the global stiffness and mass matrices. In order to investigate the effect of newly created ill-shaped elements, four consecutive cuts were applied to each mesh, as indicated in Fig. 1. After

Table 1 Meshes used for our analysis and their critical time steps. The mesh components are nodes/triangles/tetrahedrons. Δt_{crit} is obtained through the exact computation.

Mesh	Components	Δt_{crit}(ms)
cube	365/3264/1536	0.69764
bar	493/3504/1600	0.56572
$liver_1$	181/1330/596	0.45096
$liver_2$	1307/6701/3052	0.44801

Table 3 Δt_{crit} after the fourth cut for the hybrid approach (case NI) and the FEM (case I). $\Delta t_{sim} = \Delta t_{crit} * 0.95$ based on the improved meshes (case I). Number and percentage of elements detected as critical in hybrid method.

	Δt_{crit} NI	Δt_{crit} I	Δt_{sim}	critical elem./%
cube	0.1453	0.1815	0.1725	183/3.66
bar	0.1190	0.1584	0.1505	55/1.91
$liver_1$	0.0788	0.2476	0.2352	21/1.23
$liver_2$	0.2041	0.2115	0.2009	898/10.93

each cut, mesh subdivision and node snapping was carried out for all affected elements. This was required to obtain new submeshes again consisting of tetrahedral elements. In order to examine the effect of mesh improvements we then considered two cases – in the first the mesh was further processed using the mesh improvement pipeline (case I), while in the second no further processing was carried out (case NI). In order to illustrate the creation of new elements and the effect on the critical time step, Tab. 2 provides more details for cuts into one example mesh for case I. As can be seen the number of elements increases considerably, while the critical time step is reduced. Note that sometimes the time step does not change after a cut since the effect of a particularly badly shaped element that appeared in the previous cut can dominate the critical time step. In the following, we will focus the discussion on differences after the fourth cut. Nevertheless, similar effects are in general encountered for each prior incision.

Table 2 Increase of elements in $liver_1$ after each cut (case I) and change of critical time step.

	cut_1	cut_2	cut_3	cut_4
Elements	687	919	1258	1604
Δt_{crit}	0.3669	0.3668	0.2476	0.2476

In our experiment, we compared the performance of a pure FEM simulation using improved meshes with the hybrid approach using meshes with no optimization. The first item for the further analysis is to determine the critical time step via Eqn. 1. As can be seen in Tab. 3 mesh processing does result in an improvement in the time step.

However, the more important point to consider is the computation times of both methods. In order to allow a performance comparison, the same simulation time step should be selected for both approaches. We set the time step to $\Delta t_{sim} = \Delta t_{crit} * 0.95$. Note that Δt_{sim} has to be below Δt_{crit} to ensure stability of the FEM simulation; for the hybrid approach this would not be required, since any possible instable elements would be treated with the shape matching approach. For the hybrid method, we further have to set the threshold time step which is used for identifying critical elements in the hybrid method via the reduced 1-ring metric. For this usually the critical time step of the non-improved meshes is extended by a safety margin. Nevertheless, since we use a common simulation time step for both cases, we set the threshold to $\Delta t_{thr} = \Delta t_{sim} * 1.1$. Note that this actually penalizes the hybrid approach, since potentially more elements are flagged as being critical than necessary. The simulation time step, as well as the number and percentage of identified critical elements are also specified in Tab. 3.

The resulting computation time per step is presented in Fig. 2. As can be seen, using the hybrid method does not result in a significant increase in computation time. While the FEM does perform slightly better, there are two points that have to be considered for interpreting this result. First, the simulation time step cannot be enlarged further for the FEM since it is already close to the critical limit. In contrast, this is not true for the hybrid method for which larger time steps could still be selected. Doing this would result in a lower overall computation time for one simulated second. Secondly, the processing time of carrying out the mesh improvements has to be taken into account. The computational requirement for the different steps after cutting are compiled in Tab. 4. The first row shows the processing time for mesh subdivision and node snapping, which is required in all cases. The second line reports the times

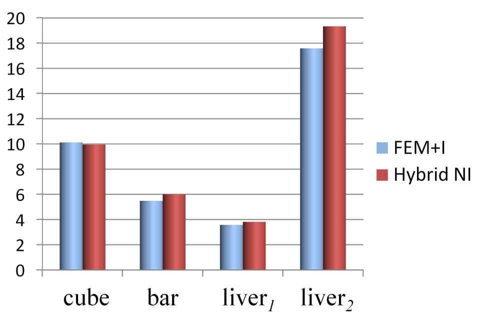

Fig. 2 Computation time per step (ms) after the fourth cut.

Table 4 Processing times (in seconds): element subdivision, mesh improvement (MI), and identification of critical elements (Hybrid). Timings obtained for the fourth cut.

	cube	*bar*	$liver_1$	$liver_2$
Cutting	0.170	0.050	0.102	0.613
MI	1.046	0.499	0.215	2.591
Hybrid	0.089	0.049	0.022	0.132

for the additional mesh improvements. The final line provides the timings for identification and processing of critical elements in the hybrid approach. As can be seen, the processing time of the mesh improvement is an order of magnitude larger than that required for the hybrid technique. Considering all points together, applying mesh improvements for a pure FEM approach appears to be less optimal than using the hybrid method on non-improved meshes; this is especially true in the context of real-time simulations, where a mesh has to be cut as quickly as possible.

We examined the results further to investigate the low performance of the mesh improvement strategy. Fig. 3 shows that the number of critical elements can be significantly reduced using mesh improvements. However, the right plot reveals that the effect on the critical time step is less strong. In fact, there is often only a minor difference as compared to not applying any improvements. Thus, the application of costly mesh optimization schemes might not be worthwhile. Usually, the critical time step is affected by a small number of particularly badly-shaped elements. These are often located close to the cut surfaces and cannot easily be modified by mesh improvements. Our results seem to indicate that even if a considerable effort is invested to improve a mesh overall, the few ill-shaped elements that have the strongest influence on the critical time step might not be sufficiently changed. Specifically identifying such elements and treating them with an alternative deformation technique could therefore be a better approach.

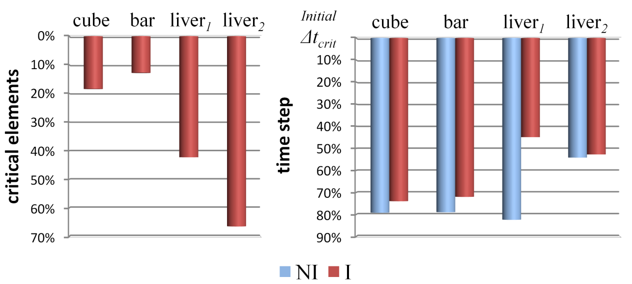

Fig. 3 Left: Reduction of number of critical elements after fourth cut due to mesh improvements (compared to non-optimized mesh). Right: Reduction of critical time step for both cases.

5 Conclusion

We have presented a brief performance analysis of a new hybrid approach in the context of mesh improvement techniques. Our results indicate that the hybrid approach might be better suited for real-time simulations than mesh improvements; especially, since the latter require larger processing times and often fail to improve the most critical elements significantly.

As future work, we will examine testing the hybrid approach together with other techniques, for instance adaptive strategies.

Acknowledgements This research was supported by the EU project PASSPORT FP7 ICT-2007-223894 and CONACYT grant no. 213030.

References

1. Aguinaga, I., Fierz, B., Spillmann, J., Harders, M.: Filtering of high modal frequencies for stable real-time explicit integration of deformable objects using the Finite Element Method. Progress in biophysics and molecular biology pp. 1–12 (2010)
2. Burkhart, D., Hamann, B., Umlauf, G.: Adaptive and Feature-Preserving Subdivision for High-Quality Tetrahedral Meshes. Computer Graphics Forum **29**(1), 117–127 (2010)
3. Fierz, B., Spillmann, J., Aguinaga, I., Harders, M.: Maintaining Large Time Steps in Explicit Finite Element Simulations using Shape Matching. TVCG (2011). Preprint
4. Klingner, B., Shewchuk, J.: Aggressive tetrahedral mesh improvement. In: Proc. of the 16th international meshing roundtable, pp. 3–23 (2007)
5. Mor, A., Kanade, T.: Modifying soft tissue models: Progressive cutting with minimal new element creation. In: MICCAI 2000, pp. 598–607. Springer-Verlag (2000)
6. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless Deformations Based on Shape Matching. ACM Transactions on Graphics (TOG) **24**(3), 471–478 (2005)
7. Nealen, A., Muller, M., Keiser, R., Boxerman, E., Carlson, M.: Physically Based Deformable Models in Computer Graphics. Computer Graphics Forum **25**(4), 809–836 (2006)
8. Nienhuys, H.W., van der Stappen, A.: Combining finite element deformation with cutting for surgery simulations. In: Proc. EuroGraphics, pp. 274–277 (2000)
9. Serby, D., Harders, M., Szekely, G.: A new approach to cutting into finite element models. In: MICCAI 2001, pp. 425–433. Springer (2001)
10. Shewchuk, J.R.: What is a good linear element? interpolation, conditioning, and quality measures. In: 11th international Meshing Roundtable, pp. 115–126. Springer (2002)
11. Steinemann, D., Harders, M., Gross, M., Szekely, G.: Hybrid Cutting of Deformable Solids. In: IEEE Virtual Reality, pp. 35–42 (2006)