Project Proposal
Dynamic Local Remeshing for Elastoplastic Simulation

Project presentation

The goal of this project is to extend on the deformation algorithm presented in the paper *Dynamic local remeshing for elastoplastic simulation* [1]. To keep a locally injective transformation between the material space, in which the remeshing is performed and no deformation occur, and the world space where external forces are applied to the objects. Code, videos as well as a slideshow presentation are available at the following *link* [1]

To perform the deformations, they use a set of mesh operations on tetrahedra of bad quality to dynamically improve the geometry in the regions containing them. These operations are the following:

- **Smoothing**: This consist of moving vertices to improve the quality of the neighbouring elements. This is the only operation that does not change the connectivity of the mesh.
- **Edge removal**: This consist of removing an edge from the mesh along with the tetrahedra containing it. Different types of *flips* of tetrahedra allow this. The general formula for the tradeoff between the new and old number of tetrahedra is to replace m tetrahedra with $2m - 4$ new ones.
- **Multi-face removal**: This is the inverse of the *edge removal* presented above. With the help of *flips*, we aim to remove m faces by replacing $2m$ tetrahedra into $m + 2$.
- **Vertex insertion**: This is used in two cases, to increase the resolution of some part of the mesh or to eliminate tetrahedra of poor quality. First, it hollows out a polyhedral cavity and then replaces the deleted tetrahedra by new ones joining the new vertex to a face of the cavity. Vertex insertion is effective as part of a compound operation, edge and multi-face removal are applied after the vertex insertion to evaluate the quality of the mesh and accept or rollback the operation.
- **Edge contraction**: This has two uses, coarsen the mesh where its tetrahedra are unnecessarily small and remove tetrahedra that have poor quality because an edge is too short. It removes an edge by by replacing its endpoints by a single vertex. Every tetrahedron that shares the deleted edge is then also removed. This operation can be rejected if it worsen the mesh quality.

To note is that these operations are constrained by the two spaces used. It could be that an operation is not valid in both spaces simultaneously.

Objectives

The goal of this project is to understand and adapt the remeshing process presented in the paper as well as to see how the material space can limit the improvement in the world space.

To do so, I will first implement similar operations as the ones presented in the previous section in the 2D domain. This allows for a better understanding of the different challenges or difficulties that could arise in 3D before they happen as well as to provide a simpler experimenting setting.

The second step will be to transcribe the 2D implementation into a 3D setting, working with tetrahedral meshes similarly to the topology used in the paper.

Lastly and depending on the progress of the project, I could use my implementation in some sort of physics simulation to test its capabilities and resistance to challenging scenarios or settings.

[1] http://graphics.berkeley.edu/papers/Wicke-DLR-2010-07/

Plan

We can split the work in the following manner:

Task	Duration
2D implementation and evaluation	
Setup	1 week
OpenFlipper plugin setup	
OpenFlipper interface	
2D implementation	1 month
2D version of operations design	
2D version of operations implementation	
Evaluation of quality with high deformations	
Evaluation of consequences for material space without any restriction	
Going further	1-2 months
Implementation of restriction criteria based on material space	
Re-evaluation of quality of world space mesh with restrictions	
Allow modifications of material space mesh to loosen restrictions?	
3D implementation and evaluation	
Similar to 2D	3-4 months
Total	5-7 months

References

[1] Martin Wicke et al. "Dynamic Local Remeshing for Elastoplastic Simulation". In: *ACM Trans. Graph.* 29.4 (July 2010). ISSN: 0730-0301. DOI: 10.1145/1778765.1778786. URL: https://doi.org/10.1145/1778765.1778786.