Tetrahedral Mesh Improvement Using Multi-face Retriangulation

Marek Krzysztof Misztal[1], Jakob Andreas Bærentzen[1], François Anton[1], and Kenny Erleben[2]

[1] Informatics and Mathematical Modelling, Technical University of Denmark
{mkm,jab,fa}@imm.dtu.dk
[2] Department of Computer Science, University of Copenhagen
kenny@diku.dk

Abstract. In this paper we propose a simple technique for tetrahedral mesh improvement without inserting Steiner vertices, concentrating mainly on boundary conforming meshes. The algorithm makes local changes to the mesh to remove tetrahedra which are poor according to some quality criterion. While the algorithm is completely general with regard to quality criterion, we target improvement of the dihedral angle. The central idea in our algorithm is the introduction of a new local operation called multi-face retriangulation (MFRT) which supplements other known local operations. Like in many previous papers on tetrahedral mesh improvement, our algorithm makes local changes to the mesh to reduce an energy measure which reflects the quality criterion. The addition of our new local operation allows us to advance the mesh to a lower energy state in cases where no other local change would lead to a reduction. We also make use of the edge collapse operation in order to reduce the size of the mesh while improving its quality. With these operations, we demonstrate that it is possible to obtain a significantly greater improvement to the worst dihedral angles than using the operations from the previous works, while keeping the mesh complexity as low as possible.

1 Introduction and Motivation

For many types of physical simulation, the tetrahedral mesh representation is the natural choice. For instance, finite element computations in 3D usually employ tetrahedral meshes which are far better at adapting to boundaries and changes in scale than e.g. regular voxel grids.

For 2D triangulations, Delaunay triangulation is often a natural choice since it leads to a mesh which is optimal in the sense that the minimal angles are maximized which is a reasonable quality criterion in 2D. In 3D however, it is less clear what quality criterion we should strive for and a 3D Delaunay tetrahedralization may contain very flat sliver tetrahedra with extreme dihedral angles, and extreme dihedral angles are often precisely what we wish to avoid since they may lead to problems, such as great interpolation errors or ill-conditioned stiffness matrices in some finite element computations

(although in the anisotropic case they might be desirable) or problems with interpolation accuracy [21].

Consequently, in this paper and in other recent work [12], the goal is to optimize a tetrahedralization obtained through either Delaunay or other methods in order to improve some criterion – particularly dihedral angles. However, little is known about globally optimal meshes in the sense that the smallest dihedral angle is maximal or that the largest dihedral angle is minimal. Consequently, one strives instead for a set of simple, local transformations which improve the mesh by removing poorly shaped tetrahedra. The best one can hope for in this case is a good local minimum, and whether one attains such a minimum is highly dependent on one's vocabulary of local transformations. It is this vocabulary which we extend by the addition of two local transformations which are highly beneficial to the mesh quality yet have not previously been used in tetrahedral mesh optimization.

The most powerful way of improving triangle or tetrahedral meshes is through the insertion of more vertices (as shown in [12]). Indeed this is sometimes the only way to improve quality. Unfortunately, one pays the price of adding (sometimes significantly) more tetrahedra, and finding the optimal place to put a vertex can be hard. Besides, many applications (such as dynamic meshes) require their own Steiner vertex insertion routines. For these reasons, we opine that it is very worthwhile to explore to what extent our mesh improvement vocabulary can be augmented without adding vertices.

Our main contribution is the *multi-face retriangulation* operation. Assume a set of tetrahedra which we can divide into *upper* and *lower* tetrahedra. Any upper tetrahedron shares a face with precisely one lower tetrahedron (and vice versa) and the upper tetrahedra all share a vertex (the upper vertex) as do the lower tetrahedra (the lower vertex). We can say that the set of tetrahedra is *sandwiched* between the upper and the lower vertex (as illustrated in Figure 1). The union of the triangular faces shared between upper and lower tetrahedra can be seen as a triangulation of a polygon. Our proposed operation simply retriangulates this polygon to obtain better sets of upper and lower tetrahedra. Multi-face retriangulation can also be seen as a composition of the known *multi-face removal* and *edge removal* operations (as shown in Figure 1) [9, 12]. However, multi-face retriangulation is more powerful than the concatenation of these two operations: in the case of some configurations multi-face removal followed by edge removal would never be selected because very poor or inverted tetrahedra would result from the multi-face removal operation (as illustrated in Figure 2). Additionally, multi-face retriangulation works on boundaries whereas concatenation of multi-face removal and edge removal does not.

The other contribution is the use of the well known *edge collapse* operation. Curiously, to the best of our knowledge, this operation has not been incorporated into any tetramesh improvement algorithm previously. It significantly reduces the complexity of the mesh and it also might improve the worst quality within the set of affected tetrahedra.

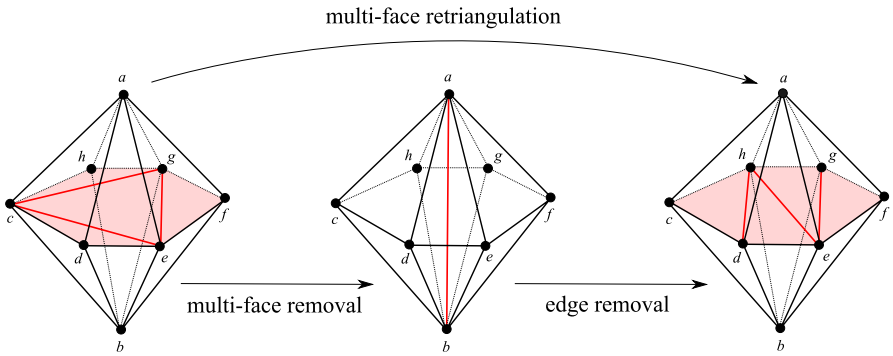


Fig. 1. Multi-face removal, edge removal and their superposition – multi-face retriangulation.

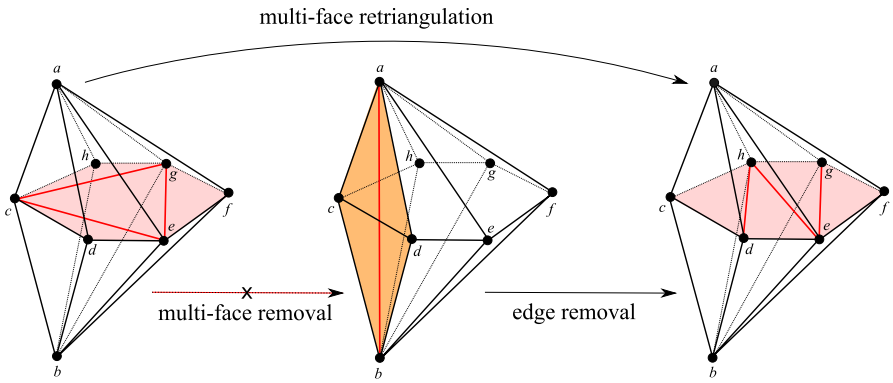


Fig. 2. Configuration in which multi-face removal would not be performed. Vertices a, b, c and d are nearly coplanar. Performing multi-face removal in such a configuration would lead to creating a very poorly shaped tetrahedron $abcd$ (highlighted in orange) of extremely low quality. Also, by perturbing vertex a or b we can easily create a situation in which tetrahedron $abcd$ would be inverted. In both situations multi-face removal would not be performed by a greedy algorithm – hence the arrow is crossed out. The strength of multi-face retriangulation is in tunneling through these kinds of hills in the energy landscape.

2 Related Work

Clearly, whether mesh improvement is needed depends on how the mesh was generated. Broadly speaking, there are three ways of producing tetrahedral meshes from a boundary representaion of an object. First, if the boundary is a piecewise linear complex (in particular – triangulated manifold), we could use constrained Delaunay tetrahedralization to produce a conforming mesh [19, 22]. Alternatively, we could use an advancing fronts method which would build the tetrahedralization out from the boundary. As mentioned,

the former approach will often have problems with sliver tetrahedra even after Delaunay refinement, unless the boundary satisfies a set of strict conditions [3,19] limiting the practical applications of this approach, and the latter tends to produce some bad tetrahedra around areas where the front collides on itself [16]. These problems are compounded if the boundary mesh has poorly shaped triangles. An alternative approach is the centroidal Voronoi tessellation based Delaunay tetrahedralization [6] which can, however, still leave some poorly shaped tetrahedra. We conclude that the mesh optimization is likely to be useful as a step following both Delaunay based methods and also advancing fronts based methods.

A third and alternative strategy is to force the boundary to conform to an isosurface of an implicit function rather than a mesh. The spatial domain is first divided into tetrahedra, and a subset which approximates the shape well is selected. In a subsequent compression step, the boundary vertices of this subset are forced to lie precisely on the isosurface [15]. However, we note that the compression step is an optimization procedure because, generally, not only the boundary vertices are moved but also the interior vertices in order to improve the quality of the mesh. In recent work, Labelle and Shewchuk were able to demonstrate good provable bounds on the dihedral angles using such a method [13]. However, methods which fit meshes to isosurfaces [15,13] cannot be expected to capture sharp edges and corners because the vertices are not constrained to lie in particular positions. Consequently, in some cases they simply do not apply.

Most of the existing work for tetrahedral mesh improvement uses the following three types of mesh operations:

1. *Mesh smoothing* – relocation of the mesh points in order to improve mesh quality without changing mesh topology.
2. *Topological operations* – reconnection of the vertices in the mesh (without displacing them).
3. *Vertex insertion* – adding extra vertices into the mesh (through eg. splitting of the edges, faces or tetrahedra) and reconnecting affected regions of the mesh.

2.1 Mesh Smoothing

One of the best known smoothing methods, Laplacian smoothing, in which a vertex is moved to the centroid of the vertices to which it is connected, is a popular and quite effective choice for triangular meshes. In tetrahedral meshes, however, it often produces poor tetrahedra [7]. Optimal (with regard to linear interpolation error) Delaunay vertex placement has been investigated by Chen and Xu [2]. More general mesh smoothing algorithms are based on numerical optimization. One of the most popular local algorithms for mesh smoothing was suggested by Freitag et al. [8]. This method relocates one vertex at a time. Given one vertex, its new position is found, so

that the minimum quality of all the tetrahedra adjacent to this vertex is maximized (this requires non-smooth optimization). This procedure is performed for each vertex in the mesh and can be iterated until a stable configuration is attained. It can also be performed on the boundary of the mesh, given extra constraints for the position of the vertex. Another optimization based approach, using generalized linear programming, was presented by Amenta et al. [1], but this one is not as general as Freitag's and is not well suited for dihedral angles optimization. Mesh smoothing can also be performed by continuous optimization in the space of coordinates of all vertices of the mesh (as in [10]), but Freitag's method has advantages over this approach – it is easier to use with non-smooth quality measures, and its characterised by stable behavior even if the initial quality of the mesh is very low.

2.2 Topological Operations

Reconnection of the mesh can be pictured as picking a set of adjacent tetrahedra and replacing them with another set of tetrahedra, of higher minimum quality, filling in the same volume. This can be performed in a more or less arbitrary manner (as small polyhedron re-tetrahedralization in [14]), or can be organized into a set of *topological operations*, such as:

- *2-3 flip* and its inverse, *3-2 flip*, as shown in Figure 3.
- *4-4 flip* and its version for boundary configuration, *2-2 flip*, illustrated in in Figure 3 – ambiguous, requires specifying which edge pair of vertices is going to be connected after the operation.
- *Edge removal* is illustrated in Figure 1 – generalizes 3-2 flip, 4-4 flip and 2-2 flip; ambiguous, requires specifying the final triangulation of the link of the removed edge, which can be performed by using triangulation templates, as in [9] or by using Klincsek's algorithm [11] in order to maximize the minimal quality of the created set of tetrahedra, as in [12]. Edge removal can be performed for boundary edges.
- *Multi-face removal* of de Cougny and Shephard [5] is the inverse to the edge removal, as shown Figure 1 – generalizes 2-3 flip and 4-4 flip; requires dynamic programming in order to select the subset of faces sandwiched between two vertices, which gives the best improvement.

The original paper of Freitag and Ollivier-Gooch [9] uses the first three operations. It can easily be noticed, that multi-face removal can actually be decomposed into a sequence of a single 2-3 flip followed by a certain number of 3-2 flips. However, this can not always be performed in the *hill-climbing* approach (which is usually the choice for the tetrahedral mesh-improvement algorithms) if one of the operations in the sequence decreases quality locally. Klingner and Shewchuk [12] use all the operations from the list above.

2.3 Vertex Insertion

Klingner and Shewchuk showed in [12] that mesh improvement is far more effective with the inclusion of transformations that introduce Steiner vertices.

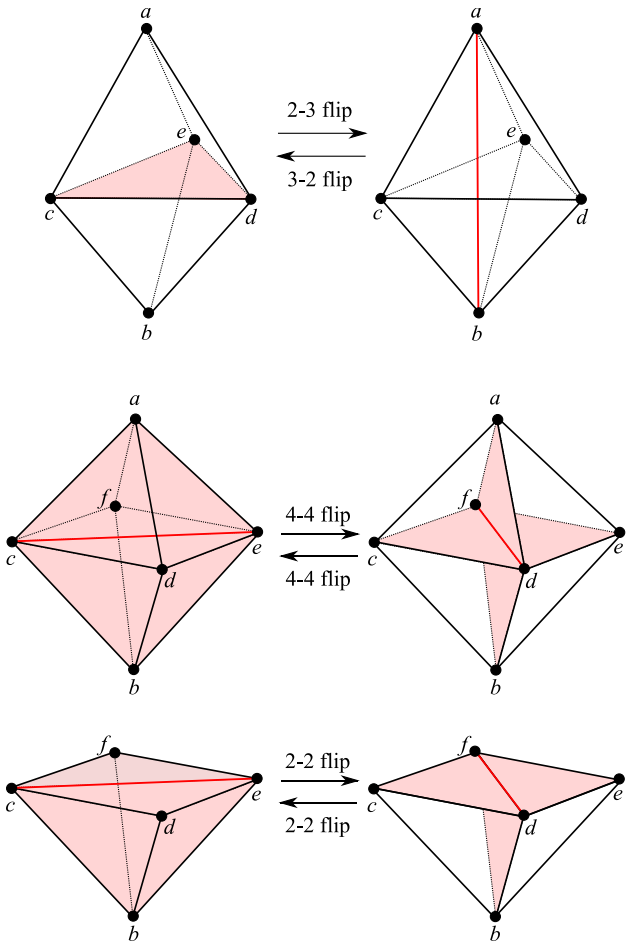


Fig. 3. Simple topological operations.

Proper placement of Steiner vertices is a hard problem. Klingner and Shewchuk describe a sophisticated and rather complex algorithm for vertex insertion which mimics Delaunay vertex insertion and, together with optimization based smoothing and topological operations, allows them to improve the meshes so that all dihedral angles are between 31° and 149°, or, using a different objective function, between 23° and 136°.

3 Tetrahedral Mesh Quality Improvement

Our mesh improvement algorithm is based on the algorithm proposed by Klingner and Shewchuk [12] (which in turn extends one by Freitag and Ollivier-Gooch [9]) which uses vertex smoothing by Freitag et al. [8], edge

removal, multi-face removal and vertex insertion (most of the operations they use can be performed on the boundary of the mesh). In turn, our algorithm uses the following set of operations:

- Vertex-smoothing as in Freitag et. al [8],
- Topological operations:
  - edge removal,
  - multi-face removal,
  - multi-face retriangulation.
- Edge collapse.

Vertex smoothing and edge removal can be performed for the boundary vertices and edges, if the boundary is sufficiently flat around them. Additionally, vertex smoothing can be performed along *straight* ridges on the boundary of the mesh, if the surface patches separated by the ridge are sufficiently flat. Multi-face removal and edge removal are implemented essentially the same way as in [20].

3.1 Multi-face Retriangulation

Multi-face retriangulation can be seen as a composition of multi-face removal and edge removal, however, it can be also performed on the boundary of the mesh. It includes the 4-4 and 2-2 flips. Multi-face retriangulation does not change the number of tetrahedra in the mesh. So far as we know, it has never appeared in the literature.

The reasons in favor of using MFRT alongside multi-face removal and edge removal are:

- In some cases, the configuration produced by multi-face removal is of lower quality, as illustrated in Figure 2. Thus a greedy approach would not select that configuration even if the subsequent edge removal led to a state of lower energy than the initial configuration.
- In some cases the configuration produced by multi-face removal includes inverted tetrahedra, and no approach would select that (also shown in Figure 2). However, MFRT cannot produce inverted tetrahedra, as the best triangulation of the multi-face cannot be worse than the initial one and we assume we run our algorithm on valid tetrahedral meshes.
- MFRT can be applied to boundary configurations. To see this, let us only consider a set of lower tetrahedra in Figure 1. In such a configuration, multi-face consists of boundary faces and it cannot be removed using multi-face removal, but it can easily be retriangulated. However, if the multi-face is not sufficiently flat, which is the case when the angles between the normals to the faces are greater than $0°$, MFRT can change the geometry of the boundary of the mesh, which is usually not desirable.
- MFRT does not change the number of tetrahedra. This property is a direct consequence of a well known fact that every triangulation of a polygon has the same number of triangles.

In our implementation, the input is a single face f we wish to remove. We find the apices a and b of the two tetrahedra adjoining f. Among the set of all faces sandwiched between a and b we find the connected component that contains f. For the multi-face defined like that, we find the optimal triangulation of its bounding polygon using Klincsek's algorithm. The routine is similar for a boundary face f, although in this case we have to make sure that the retriangulated multi-face is sufficiently flat (otherwise geometry of the boundary might change).

3.2 Edge Collapse

Edge collapse (also known as *edge contraction* or *half-edge contraction*) is a well known mesh operation which has been used as a primary tool for simplifying 2D and 3D meshes in numerous works, such as [4, 17]. It identifies one of the vertices of an edge e with the other, removes e and all faces and tetrahedra which contain it. This can, however, lead to invalid configurations (violating the simplicial complex criterion) or alter the surface geometry of the mesh, unless certain conditions are fulfilled, described in detail by Natarajan and Edelsbrunner in [17]. If edge collapse is not performed for the boundary edges, which is the case in our implementation, those conditions simplify to the following:

$$\mathrm{Lk}(e) = \mathrm{Lk}(a) \cap \mathrm{Lk}(b),$$

where a and b are the vertices of the edge e, and $\mathrm{Lk}(\sigma)$ denotes the link of a simplex σ which, in tetrahedral meshes, can be defined as a set of those simplices (vertices, edges and faces) in the mesh, that do not intersect with σ, but are contained by the one of the tetrahedra containing σ. In our implementation this is performed if the minimum quality of the set of tetrahedra affected by the operation increases, or if it does not decrease below a certain quality threshold q_{min}, which is a global parameter of our algorithm.

3.3 Quality Measures

Both the smoothing algorithm and the topological operations which we are using are indifferent to the tetrahedron quality measure. In order to be able to compare our results to those provided in [12] and [9], we are using:

- The *minimum sine measure* – the minimum sine of a tetrahedron's six dihedral angles, penalizes both small and large dihedral angles.
- The *minimum biased sine measure*, which is like the minimum sine measure, but if a dihedral angle is obtuse, its sine is multiplied by 0.7 (before choosing the minimum). This quality measure penalizes large angles more aggressively than the small angles.

Many quality measures have been proposed for tetrahedral meshes reviewed by [21], [10]. Our two choices are well behaved and very intuitive, although non-smooth.

4 Implementation

Our mesh improvement schedule follows that of Klingner and Shewchuk [12] (pseudo code is shown in Algorithm 1). Same as in their work, we use a short list of *quality indicators* in order to measure progress in lowest quality tetrahedra improvement. Those are: the quality of the worst tetrahedron in the entire mesh and seven *thresholded means* of the qualities of all the tetrahedra in the mesh. A mean $\bar{q_\theta}$ with threshold θ is computed the following way:

$$\bar{q_\theta} = \frac{1}{\#\{\text{tetrahedra in } M\}} \sum_{t \in M} \min(q(t), \theta),$$

where M is the mesh and q is the tetrahedron quality measure we use. For our quality measures we use thresholded means with thresholds $\sin(1°)$, $\sin(5°)$, $\sin(10°)$, $\sin(15°)$, $\sin(25°)$, $\sin(35°)$ and $\sin(45°)$. A quality indicator designed like that is a good measure of how narrow the distribution of the tetrahedron qualities is and allows us to detect the quality improvement even if the minimum quality does not change. The minimum quality alone is much less efficient as a mesh quality indicator – it leads to premature termination of the mesh improvement algorithm and significantly worse final results. We consider the improvement in the mesh quality *sufficient* if either the quality of the worst tetrahedron improves, or if one of the thresholded means increases by at least 0.0001.

We begin mesh improvement with a vertex smoothing pass, followed by a topological pass. In the topological pass, pseudo code of which is shown in Algorithm 2, we first obtain the list of all the tetrahedra in the mesh and then try to remove every tetrahedron t on the list by first trying to remove its edges using the edge remove operation and then, if we have not succeeded, by trying to remove its faces using multi-face retriangulation followed by multi-face removal. Such an ordering of the operations is justified by the fact that first performing multi-face retriangulation still leaves room for extra improvement through multi-face removal, while it does not work the other way round. The optimal multi-face for multi-face removal is chose using dynamic programming, accordingly to an algorithm described in [20]. Any of those operations are performed only if they locally improve the quality. If that happens, we proceed to the next tetrahedron on the list. Every topological operation that we use can destroy more tetrahedra than the one for which it was called, so before attempting to remove any tetrahedron we have to make sure that it still exists in the mesh.

After two initial passes we begin the main loop, in which we first smooth all the vertices until there is no more improvement detected by our mesh quality indicators. Then we start the topological pass again. If it improves the quality of the mesh sufficiently, we start the loop again, otherwise we start the thinning pass (pseudo code is shown in Algorithm 3). In the thinning pass we attempt to collapse every edge which is not a boundary one, does not connect two boundary vertices and fulfills the edge collapse feasibility

Algorithm 1. IMPROVE(M)
{M is a mesh}

```
Smooth each vertex of M.
TopologicalPass(M)
failed ⇐ 0
while failed < 3 do
    Smooth each vertex of M.
    if M improved sufficiently then
        failed ⇐ 0
    else
        TopologicalPass(M)
        if M improved sufficiently then
            failed ⇐ 0
        else
            ThinningPass(M)
            if M improved sufficiently then
                failed ⇐ 0
            else
                failed ⇐ failed + 1
            end if
        end if
    end if
end while
```

Algorithm 2. TOPOLOGICALPASS(M)

```
Create the list of all tetrahedra in M.
for each tetrahedron t on the list, that still exists in M do
    for each edge e of t (if t still exists) do
        Attempt to remove edge e.
    end for
    for each face f of t (if t still exists) do
        Attempt to remove face f by first using multi-face retriangulation
        followed by multi-face removal.
    end for
end for
```

condition. We perform the collapse only if it improves the quality locally or if the quality of the affected tetrahedra after the operation is not smaller than a threshold value $q_0 = 0.5$. If the thinning pass improved the quality of the mesh sufficiently, we start the loop again, otherwise we record that the sequence of smoothing, topological and thinning passes did not manage to improve the quality of the mesh. If that happens three times in a row, the algorithm stops.

Algorithm 3. ThinningPass(M)

```
for each edge e ∈ M that is not on the boundary do
    if e still exists then
        Find the vertices a and b of e.
        if b is not a boundary vertex then
            Attempt to collapse edge e: a ← b.
            if success then
                Smooth a.
            end if
        end if
        if a is not a boundary vertex and e still exists then
            Attempt to collapse edge e: b ← a.
            if success then
                Smooth b.
            end if
        end if
    end if
end for
```

5 Tests and Results

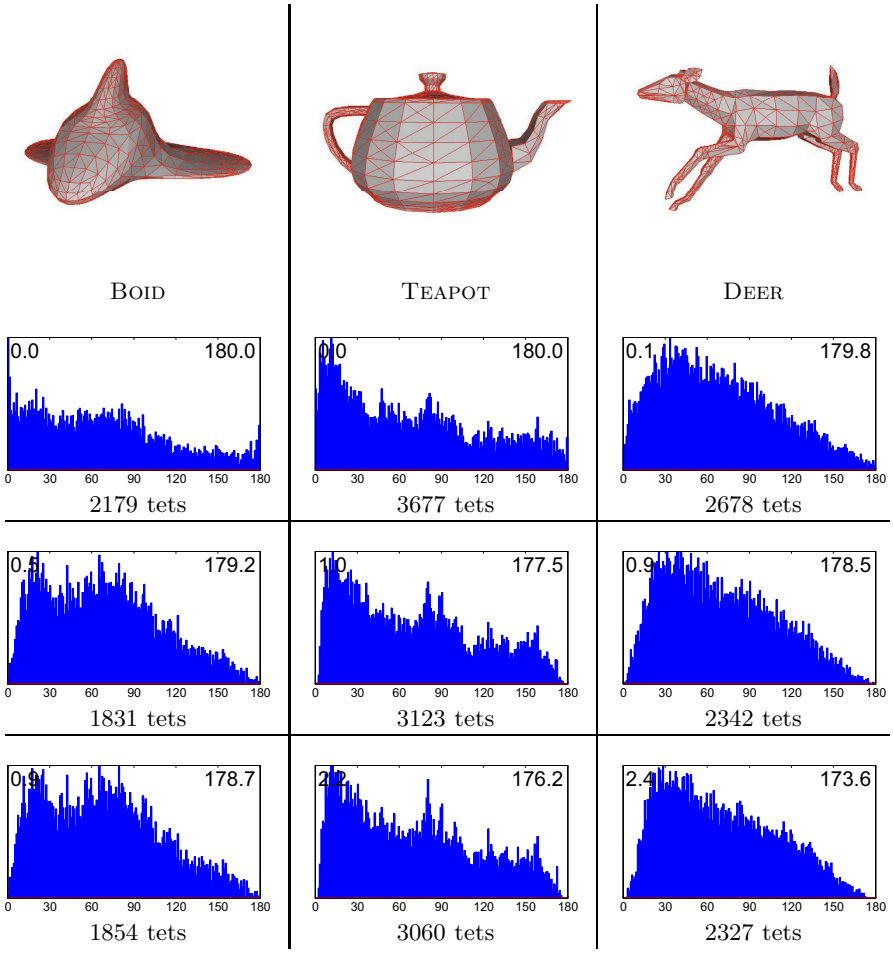
We tested our schedule on the following meshes:

- Boid, Teapot and Deer are Delaunay tetrahedralizations generated by TetGen [22] with extremely bad dihedral angles due to the lack of interior vertices.
- Rand1 – used by Freitag and Ollivier-Gooch and also by Klingner and Shewchuk to evaluate their mesh improvement algorithms.
- P and TFire – used by Klingner and Shewchuk to evaluate their mesh improvement algorithm.
- Glass – medium size mesh generated using TetGen [22] with few interior vertices and low quality boundary triangles.

Unfortunatelly, Klingner and Shewchuk published the results of mesh improvement without vertex insertion only for a very few meshes, so the possibility of comparing our results to theirs was limited.

The results of mesh improvement for those meshes are presented in the Tables 1, 2 and 3. For the Boid mesh we tried to maximize the minimum biased sine quality measure for this mesh. The boundary of the mesh is nowhere flat so smoothing and topological operations are not allowed on the boundary. There are no interior vertices, so in fact smoothing and thinning cannot take place at all, as they would alter the surface geometry. Not much improvement can be achieved without vertex insertion in this case, but still we can see that the topological pass with MFRT is significantly more effective at fighting the worst dihedral angles than the topological pass without MFRT. The situation and the results are similar in the case of the Teapot mesh.

Table 1. Mesh quality improvement results for meshes: Boid, Teapot and Deer. Minimum biased sine measure was used for the first two and minimum sine quality measure was used for the last one. Pictures in the first row show the initial surface geometry of our meshes. Surface geometry remains the same after the mesh improvement, although the tesselation might change in flat regions. Histograms show, from the top to the bottom, the distribution of all dihedral angles in the original mesh, mesh improved without MFRT and mesh improved using MFRT.



We also obtain a significant extra improvement (6.4°) by the use of MFRT for the Deer mesh, while in this case we tried to maximize the minimum sine quality measure.

For Rand1 the use of MFRT allows us to narrow the dihedral angles range by as much as 8° for both sine and biased sine quality measures. Additionally, edge collapse allows us to decrease the complexity of the meshes by almost 35% and to narrow the dihedral angles range by almost 3° for sine quality

Table 2. Mesh quality improvement results for meshes: P (using minimum biased sine quality measure) and RAND1 (left column – using minimum biased sine quality measure, right column - using minimum sine quality measure). Histograms show, from the top to the bottom, the dihedral angle distribution in the original mesh, mesh improved without MFRT, mesh improved using MFRT, meshed improved using MFRT and thinning. Red bars indicate particularily abundant dihedral angles and were scaled down to increase the readability of the histograms.

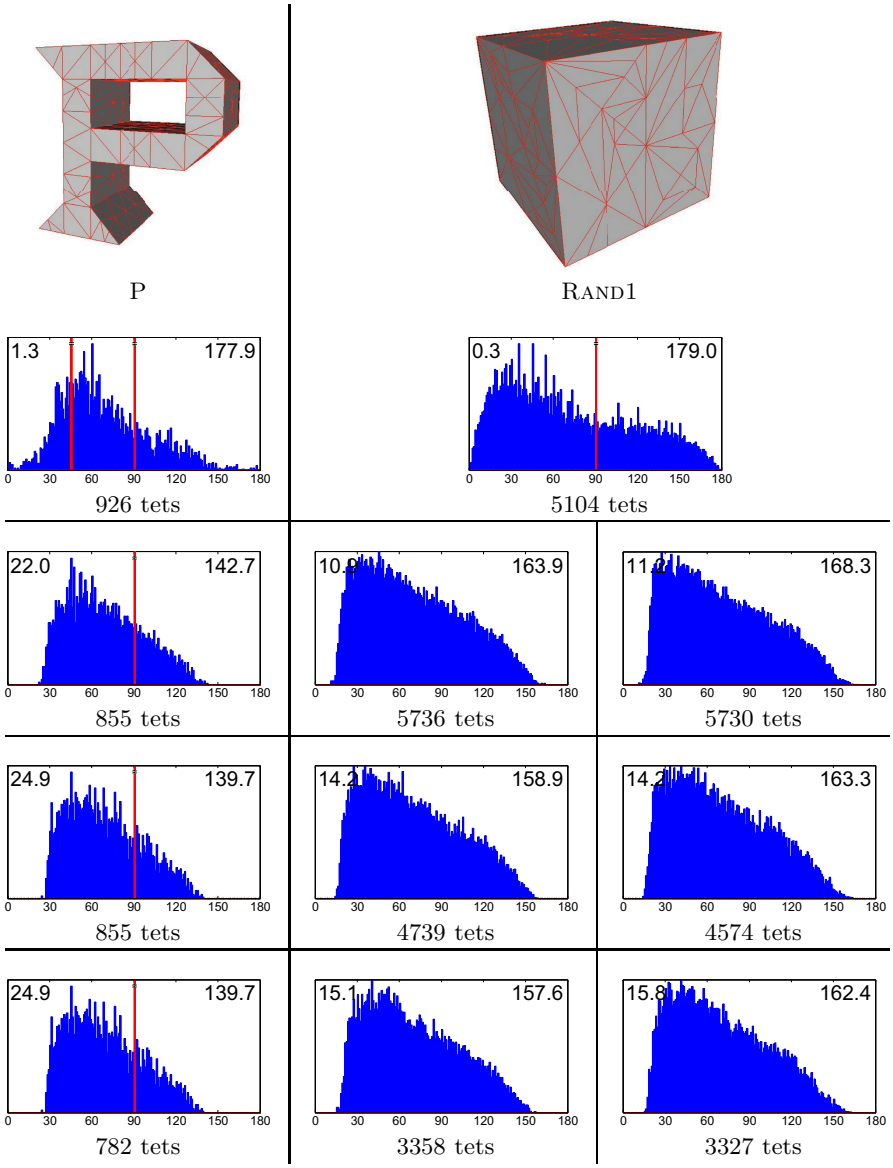
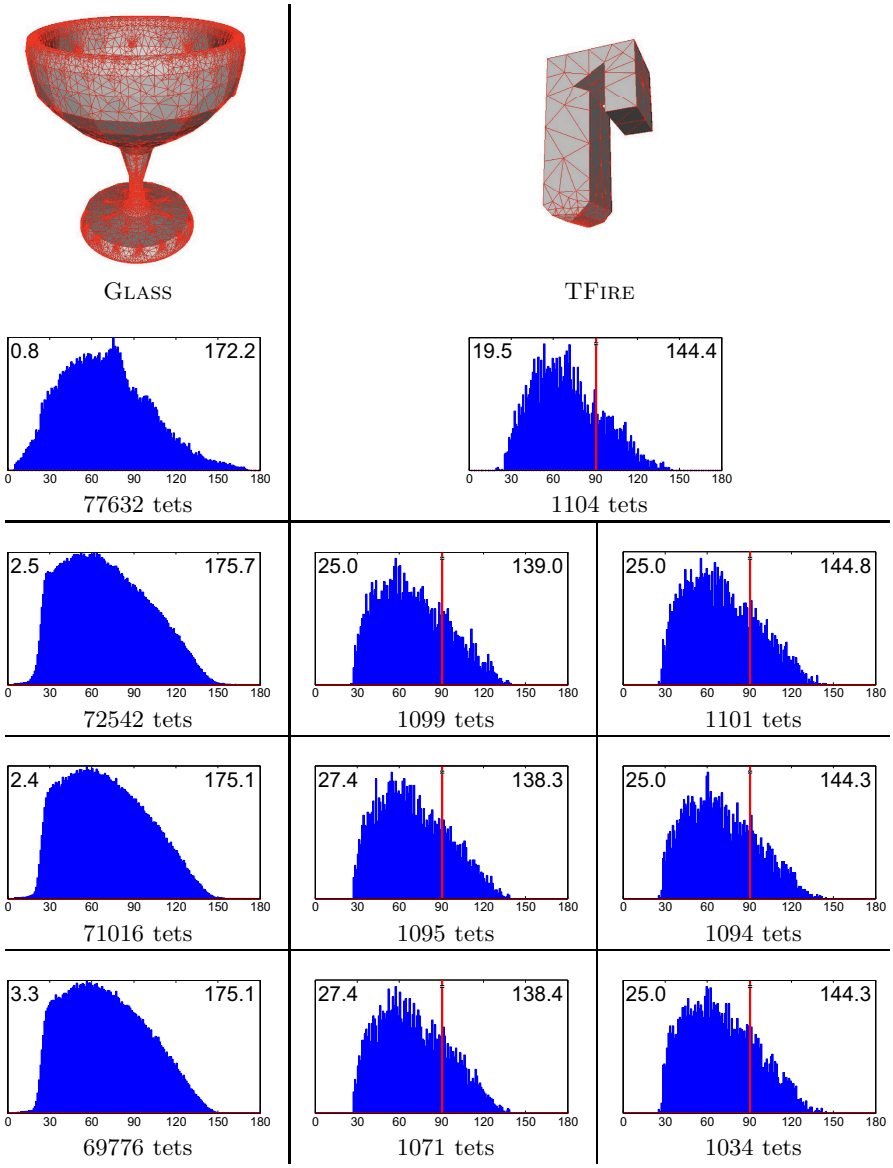


Table 3. Mesh quality improvement results for meshes: Glass (using minimum biased sine quality measure) and TFire (left column – using minimum biased sine quality measure, right column - using minimum sine quality measure). Histograms show, from the top to the bottom, the dihedral angle distribution in the original mesh, mesh improved without MFRT, mesh improved using MFRT, meshed improved using MFRT and thinning.



measure – ultimately we obtain 15.8°–162.4°, and by 2° for biased sine quality measure – ultimately we obtain 15.1°–157.6°. For comparison, the best results Freitag and Ollivier-Gooch [9] obtained for the same mesh was 15.0°–166.7° for minmax cosine quality measure (and 12.5°-167.3° for sine quality measure). Mesh P also benefits significantly from the use of MFRT – it narrows the dihedral range by 6°, but in this case the thinning pass does not improve the extreme quality values.

The TFire mesh also benefit from adding the MFRT and the edge collapse operation, although not as significantly as the previous ones. Still, our result 24.9°–139.7° is better than 21.3°–147.1° obtained by Klingner and Shewchuk [12].

In case of the Glass mesh, we can notice that our mesh improvement algorithm actually expands the dihedral angles range. This is due to the lack of extremely obtuse angles in the original mesh, and due to the fact, that the mesh operations we use choose to "sacrifice" good quality tetrahedra in order to locally improve the worst tetrahedron. However, we can notice that the we still benefit from inclusion of MFRT and thinning in the mesh improvement algorithm.

6 Discussion and Future Work

Our results show that using the multi-face retriangulation operation alongside smoothing and topological operations from the previous works can lead to better improvement of the dihedral angles and should be included in the standard repertoire of the topological operations for tetrahedral meshes. For the meshes we tested, we obtained a narrowing of the range of dihedral angles by up to 8° without inserting a single Steiner vertex. Additionally, edge collapse can also improve the worst dihedral angles and decrease the complexity of the mesh by up to 35%, esspecially when the initial quality of the mesh is very poor.

However, during our experiments we have noticed that the mesh improvement algorithm is still prone to get stuck in the local minima, even if we use multi-face retriangulation – in a few cases, running the algorithm with some mesh operations "switched off" (for instance operations on the boundary of the mesh) leads to better results than running the algorithm with the full repertoire of mesh operations. This is, of course, a consequence of using a greedy, hill-climbing approach. This could possibly be improved by applying a randomized approach.

It is also important to notice that our algorithm is designed for valid input meshes. If the initial mesh has inverted tetrahedra the algorithm might fail to remove them. Also, the tetrahedron quality measures we used are not particularily well suited for meshes with inverted tetrahedra, since they lose continuity as the tetrahedron gets inverted.

In the future we are going to further investigate the possibilities of mesh improvement without Steiner vertex insertion, also with other quality measures, such as the *volume-length* measure [18] V/l_{rms}^3, where V is the signed volume of a tetrahedron and l_{rms} is the root-mean-squared edge length.

Acknowledgments

We thank Frederik Gottlieb for the implementation of the core part of the data structure that we are using, Bryan Klingner, Jonathan Richard Shewchuk and Mads Fogtmann Hansen for meshes, geometric models and discussion. We would also like to thank Vedrana Andersen for helping us give this paper its final shape and anonymous reviewers for useful suggestions.

References

1. Amenta, N., Bern, M., Eppstein, D.: Optimal point placement for mesh smoothing. In: Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms, pp. 528–537. Society for Industrial and Applied Mathematics, Philadelphia (1997)
2. Chen, L., Xu, J.: Optimal delaunay triangulation. J. Comp. Math. 22, 299–308 (2004)
3. Chew, L.P.: Guaranteed-quality delaunay meshing in 3d (short version). In: Proceedings of the thirteenth annual symposium on Computational geometry, pp. 391–393. ACM, New York (1997)
4. Cutler, B., Dorsey, J., McMillan, L.: Simplification and improvement of tetrahedral models for simulation. In: SGP 2004: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 93–102. ACM, New York (2004)
5. de Cougny, H.L., Shephard, M.S.: Refinement, derefinement and optimization of tetrahedral geometric triangulations in three dimensions (1995) (Unpublished manuscript)
6. Du, Q., Wang, D.: Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations. Int. J. Numer. Meth. Eng 56, 1355–1373 (2002)
7. Freitag, L.A.: On combining laplacian and optimization-based mesh smoothing techniques. In: Trends in Unstructured Mesh Generation, pp. 37–43 (1997)
8. Freitag, L.A., Jones, M., Plassmann, P.: An efficient parallel algorithm for mesh smoothing. In: Proceedings of the Fourth International Meshing Roundtable (1995)
9. Freitag, L.A., Ollivier-Gooch, C.: Tetrahedral mesh improvement using swapping and smoothing. International Journal for Numerical Methods in Engineering 40, 3979–4002 (1997)
10. Hansen, M.F., Bærentzen, J.A., Larsen, R.: Generating quality tetrahedral meshes from binary volumes. In: Proceedings of VISAPP 2009 (2009)
11. Klincsek, G.T.: Minimal triangulations of polygonal domains. Annals of Discrete Mathematics 9, 121–123 (1980)
12. Klingner, B.M., Shewchuk, J.R.: Agressive tetrahedral mesh improvement. In: Proceedings of the 16th International Meshing Roundtable, October 2007, pp. 3–23 (2007)

13. Labelle, F., Shewchuk, J.R.: Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. ACM Transactions on Graphics 26(3), 57 (2007)
14. Liu, J., Sun, S.: Small polyhedron reconnection: A new way to eliminate poorly-shaped tetrahedra. In: Proceedings of the 15th International Meshing Roundtable, pp. 241–257 (2006)
15. Molino, N., Bridson, R., Teran, J., Fedkiw, R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In: Proc. International Meshing Roundtable (2003)
16. Möller, P., Hansbo, P.: On advancing front mesh generation in three dimensions. International Journal for Numerical Methods in Engineering 38(21), 3551–3569 (1995)
17. Natarajan, V., Edelsbrunner, H.: Simplication of three-dimensional density maps. IEEE Transactions on Visualization and Computer Graphics 10, 587–597 (2004)
18. Parthasarathy, V.N., Graichen, C.M., Hathaway, A.F.: A comparison of tetrahedron quality measures. Finite Elements in Analysis and Design 15(3), 255–261 (1994)
19. Shewchuk, J.R.: Tetrahedral mesh generation by Delaunay refinement. In: Proceedings of the fourteenth annual symposium on Computational geometry, pp. 86–95. ACM, New York (1998)
20. Shewchuk, J.R.: Two discrete optimization algorithms for the topological improvement of tetrahedral meshes (Unpublished manuscript) (2002)
21. Shewchuk, J.R.: What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (2002) (Unpublished manuscript)
22. Si, H.: Tetgen, a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, v1.3 user's manual. Technical report, WIAS (2004)