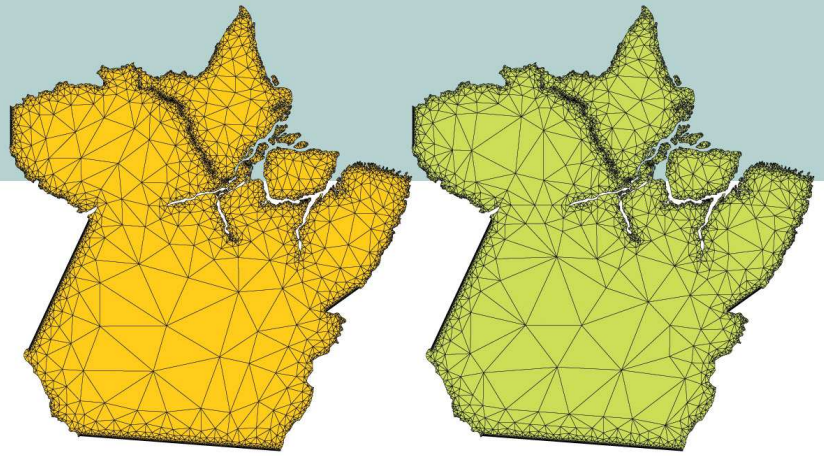
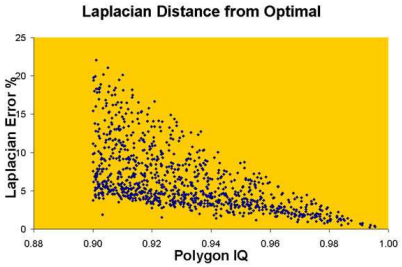
Optimal Smoothing in Polygons

Alexandre Cunha & Omar Ghattas
Carnegie Mellon University



Improved Mesh **Original Mesh**

We investigate a combined approach of Laplacian and optimization based smoothing for triangular meshes. The key contribution of this approach is to guarantee that at the end of each local smoothing step (moving a node inside its cointaining polygon) we are within a small threshold of the optimal solution, even when using just the inexpensive and simple Laplacian smoothing method.



Laplacian Smoothing

Although Laplacian smoothing does not offer a general guarantee of mesh quality, there are cases in which it perfoms well. When smoothing a node inside a regular polygon, it gives the optimal solution, i.e., the best possible triangles are created. For close to regular polygons, it gives close to optimal solutions. So, one might ask: ***how regular a polygon can be in order to have Laplacian smoothing produce good results ?***

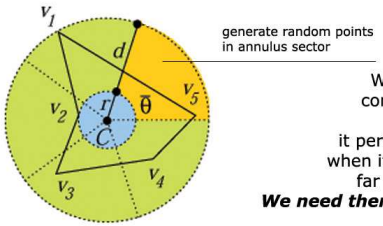
Polygon Regularity

We determine how regular a polygon is using the *isoperimetric quotient* (IQ) expression for polygons. The IQ is used to measure distortion of both the triangles inside the polygon and of the polygon itself. It is a normalized measure achieving a value of one for regular polygons and it approaches zero for highly distorted polygons. We studied the performance of Laplacian smoothing for a variety of polygons spanning the spectrum of IQ values.

$$\gamma = \frac{4 \tan \frac{\pi}{n} \mathcal{A}}{\sum {a_i}^2}$$

polygon area

edge length



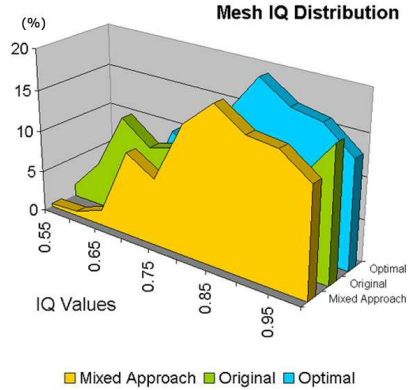
generate random points in annulus sector

Random Polygon Generation

We rely on a random polygon generator to build tens of millions of convex and non convex star-shaped polygons and assess how good Laplacian smoothing performs. As expected, we found that it performs reasonably within some range of IQ values. For example, when it is applied to hexagons with IQ > 0.94, results are at most 10% far from optimal. But how one knows how far from optimal we are ? ***We need then to compute the optimal solution to answer this question***.

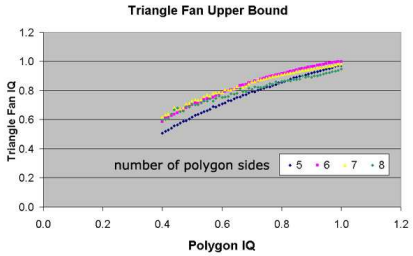
Efficiency

There are two ways to avoid the high cost of the SQP optmization smoother and still retain the "same level" of guaranteed quality. The first one is to check for regularity of containing polygons. For almost regular polygons, the inexpensive Laplacian smoothing can be safely used and optimal or close to optimal results are expected. For more distorted polygons, we might skip smoothing if the current local mesh quality is close enough to a possible maximum quality (the local upper bound).



Optmization

The optimal smoothing can be found using numerical optmization techniques. We employ an optimization package, CFSQP (A. Tits, et. al., University of Maryland), to compute the best location of a node inside a polygon. CFSQP uses the Feasible Sequential Quadratic Programming algorithm to solve the *minimax* problem. Since the input is a valid mesh and the isoperimetric quotient gives a convex function for every polygon, CFSQP always converges to a solution. For our problems, it converges on average in 3 or 4 iterations.
Besides its use to test Laplacian smoothing, we employ SQP optimization to find an upper bound for the local mesh quality with respect to the polygon IQ. This upper bound will be helpful to avoid unnecessary smoothing - when the current configuration is close enough to optimal, no further improvement is needed.



Features:

- **A simple random polygon generator that generates a rich variety of convex and non-convex star-shaped polygons as a test bed for smoothing algorithms**
- **A SQP method to do local mesh optimization**
- **An upper bound for the local mesh quality with respect to the polygon regularity**
- **Safety polygon IQ values to apply Laplacian smoothing and still produce good results**
- **A combined Laplacian and optmization-based algorithm for efficient local mesh smoothing in triangular meshes**
- **Straightforward extension for other mesh quality metrics and to 3D tetrahedral meshes.**

Carnegie Mellon

10th International Meshing Roundtable, 2001