

Nonobtuse Remeshing and Mesh Decimation

J. Y. S. Li and H. Zhang

GrUVi Lab, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada

Abstract

Quality meshing in 2D and 3D domains is an important problem in geometric modeling and scientific computing. We are concerned with triangle meshes having only nonobtuse angles. Specifically, we propose a solution for guaranteed nonobtuse remeshing and nonobtuse mesh decimation. Our strategy for the remeshing problem is to first convert an input mesh, using a modified Marching Cubes algorithm, into a rough approximate mesh that is guaranteed to be nonobtuse. We then apply iterative “deform-to-fit” via constrained optimization to obtain a high-quality approximation, where the search space is restricted to be the set of nonobtuse meshes having a fixed connectivity. With a detailed nonobtuse mesh in hand, we apply constrained optimization again, driven by a quadric-based error, to obtain a hierarchy of nonobtuse meshes via mesh decimation.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

A *nonobtuse triangle mesh* is composed of a set of nonobtuse triangles, in which every angle is less than or equal to 90° . The problem of nonobtuse remeshing can be formulated as follows. Given an input mesh M , construct a nonobtuse mesh \hat{M} , with *any connectivity*, which smoothly and accurately approximates M . To make effective use of nonobtuse meshes in an interactive setting, it is often desirable to construct a hierarchy of nonobtuse meshes, via decimation. We refer to this problem as that of nonobtuse mesh decimation.

Nonobtuse meshes are of interest in several contexts. First, a nonobtuse triangulation is necessarily a Delaunay one, both in 2D [BMR94] and for surface meshing [BS06]. Secondly, nonobtuse meshes are shown to possess more desirable numerical properties for finite element methods, e.g., in producing Stieltjes matrices [BGR88] or in ensuring faster convergence when physical properties over the discretized domain vary enormously [BMR94]. Also, nonobtuse meshes result in more efficient and more accurate geodesic estimations via fast marching [KS98]. Finally, nonobtusity ensures the validity of planar mesh embedding via discrete Harmonic maps [EDD*95]; it also implies that certain key properties of the well-known discrete Laplacian-Beltrami operator would be analogous to those of the classical Laplacian-Beltrami operator on a surface with Riemannian metric [BS06].

To the best of our knowledge, no known algorithms are guaranteed to produce a nonobtuse mesh which either interpolates or accurately and smoothly approximates a point cloud. The same holds for the remeshing problem. However, there have been studies on nonobtuse 2D triangulation and a great deal of work on remeshing of curved surfaces with angle or other quality criteria [AUGA05]. Bern et al. [BMR94] give an $O(n \log^2 n)$ algorithm for nonobtusely triangulating a n -sided polygon using $O(n)$ triangles. Chew [Che93] develops a refinement scheme based on constrained Delaunay triangulation (DT) to ensure an angle bound of $[30^\circ, 120^\circ]$. Cheng and Shi [CS05] use restricted union of balls to generate an ϵ -sampling of a surface and extract a mesh from the DT of the sample points. A lower bound on the minimum angle can be as high as 30° with proper choice of parameters, but results in a high triangle count. Neither algorithm provides a nonobtuse guarantee; this has been identified as an open problem by Gu and Yau [GY03], and so far only some simple heuristics [GY03] have been suggested.

In this paper, we present a solution for guaranteed nonobtuse remeshing and mesh decimation. To make our presentation concise, we shall only deal with closed, 2-manifold meshes. Handling of mesh boundaries is discussed in an extended version of this paper [LZ06]. The framework we develop is quite general and flexible. It allows us to generate nonobtuse meshes from point clouds with slight modifica-

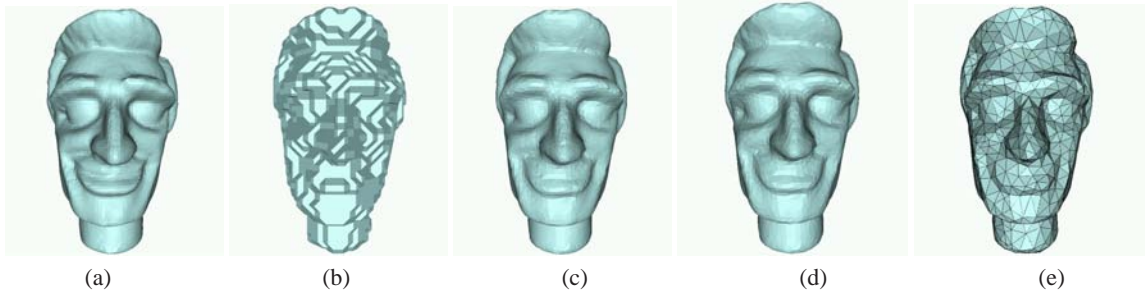


Figure 1: Major components of our approach. (a) Original mesh (34,712 faces). (b) Result of our midpoint-based Marching Cubes algorithm (15,380 faces) shows visible artifacts. (c) After constrained optimization, the approximation error from Metro [CRS98] is $\varepsilon = 0.78\%$, which is the Hausdorff distance as a percentage of the bounding box diagonal. (d) After decimation of 50% of the vertices nonobtusely. Error against (c): $\varepsilon = 0.107\%$. (e) 90% decimated. Error against (c): $\varepsilon = 0.61\%$.

tions to our existing algorithm. Angle bounds may also be enforced, although the cost of the constrained optimization will increase. Both of these issues are elaborated in [LZ06].

Our strategy for remeshing is to construct an initial nonobtuse mesh and iteratively refine it in a constrained manner to obtain a high-quality approximation of the input mesh M . Utilizing a modified Marching Cubes algorithm, we build a close but rough approximation of M . An added advantage of using Marching Cubes is that we can bound the mesh size by adjusting the grid resolution. Next, we iteratively deform the mesh, in a greedy fashion, to reduce approximation error. The deformation is carried out via quadratic programming with the constraint that each vertex must move within its *feasible region*. Once a final, fine-detailed nonobtuse mesh is generated, we can perform nonobtuse decimation via edge-collapse under similar nonobtuse constraints. See Figure 1.

2. Initial nonobtuse mesh via modified Marching Cubes

To construct the initial nonobtuse mesh \hat{M}_0 , we rely on a *midpoint-based* Marching Cubes algorithm. Originally appeared in the seminal paper of Lorensen and Cline [LC87], the Marching Cubes algorithm produces a triangle mesh which tessellates the zero level-set of a given 3D scalar field. Given an input mesh M , we generate a signed distance field sampled at the vertices of a regular cubical grid. Distance field generation for open meshes is more difficult, but it can be achieved with extra care [LZ06]. For our purpose, only the signs of the distance values matter. Specifically, instead of using linear interpolation to compute vertex locations [LC87], we insist that each mesh vertex along a cube edge be at the edge's midpoint. With a few additional modifications to mesh connectivity, e.g., with new vertices at cube centers created in some cases, the resulting triangle mesh is provably nonobtuse. Closer examination reveals that the resulting mesh also has no angle smaller than 30° . Thus it is possible to ensure an angle bound by placing tighter constraints. All these issues are discussed in length in [LZ06].

3. Deform-to-fit via constrained optimization

Given the initial nonobtuse mesh \hat{M}_0 , we wish to

$$\min. \sum_{v \in V(\hat{M})} \mathcal{L}(v, M), \quad \text{subject to } \hat{M} \in \mathcal{N}\mathcal{O}(\hat{M}_0), \quad (1)$$

where $V(\hat{M})$ is the set of vertices in \hat{M} , $\mathcal{N}\mathcal{O}(\hat{M}_0)$ is the set of nonobtuse meshes with the same connectivity as \hat{M}_0 , and

$$\mathcal{L}(v, M) = \alpha \cdot \mathcal{D}(v, M) + (1 - \alpha) \cdot \mathcal{S}(v). \quad (2)$$

The point-to-surface distance $\mathcal{D}(v, M)$, where M is the original input mesh, is a *quadratic error* [GH97], $\mathcal{S}(v)$ is a smoothness (regularization) term, and $0 \leq \alpha \leq 1$ is a free parameter for the trade-off between error reduction and smoothness. We define $\mathcal{D}(v, M) = Q_v(v)/k$ where in general, $Q_v(x)$, the quadratic error [GH97], is the sum of squared distances from x to a set of planes associated with v . In our case, this set is composed of k supporting planes, including that of the triangle T of M that is closest to v and those of the r -ring neighbor triangles of T in M . In our experiments, we set $r = 1$. At vertex v , the smoothness term $\mathcal{S}(v) = \|v - \mathcal{C}(v)\|^2$ measures the squared distance between v and the centroid $\mathcal{C}(v)$ of its one-ring neighbors. Solution to the optimization (1) is likely intractable, we thus resort to heuristics. Specifically, we move vertices one at a time in a greedy fashion, under a set of linearized and convexified nonobtuse constraints.

Given a vertex v , let its one-ring vertices be v_0, v_1, \dots , and v_{k-1} , in order. Consider the one-ring edge $e = (v_i, v_{i+1})$. Let U and V be the two planes orthogonal to e , passing through v_i and v_{i+1} , respectively. Let $sphere(e)$ be the sphere centered at the midpoint q of e with diameter $|e|$ and let p be the intersection between \overline{vq} and $sphere(e)$; refer to Figure 2. Finally, let R be the tangent plane of $sphere(e)$ through p . We define the *feasible region* $\mathcal{F}(e, v)$ of edge e with respect to v as the intersection of the front half spaces defined by U , V , and R , with their normals shown in the figure. The feasible region $\mathcal{F}(v)$ of v is then given by $\mathcal{F}(v) = \bigcap_{i=0}^{k-1} \mathcal{F}(v_i, v_{i+1}, v)$, which is linear and convex. Note that by construction, $\mathcal{F}(v)$ is always nonempty, since $v \in \mathcal{F}(v)$.

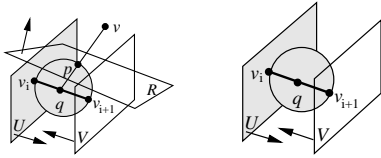


Figure 2: Left: feasible region $\mathcal{F}[(v_i, v_{i+1}), v]$ is the space delimited by planes U , V , and R . Right: any point between U and V and outside the sphere makes a nonobtuse triangle with v_i and v_{i+1} . Thus the feasible region is a linearized and convexified subset of the actual “nonobtuse region”.

To solve (1), we iteratively deform the mesh to obtain progressively better approximation to M . For each vertex v in the current mesh \hat{M} , we compute an optimal position v^* which minimizes $\mathcal{L}(v, M)$ subject to $v \in \mathcal{F}(v)$. To solve this convex quadratic programming problem, we rely on the OQP solver of Gertz and Wright [GW03]. Then a priority $\mathcal{H}(v)$, given by the improvement made by moving v to v^* : $\mathcal{H}(v) = \mathcal{L}(v, M) - \mathcal{L}(v^*, M)$, is computed. The vertex with the highest priority is moved to its optimal position and all vertices influenced by the move will have their priorities recomputed. To update the quadric for a vertex v that has just moved, we need to find a triangle T' in M that is closest to v . In our implementation, we execute a local search within the 3-ring neighborhood of the previous closest triangle T . The greedy optimization stops when $\mathcal{H}(v) \leq 0, \forall v \in V(\hat{M})$.

Experimentally, the nonobtuse meshes produced by optimization alone may be slightly rough. This can be attributed to the fact that our optimization is highly localized. To this end, we suggest to alternate between mesh optimization and constrained Laplacian smoothing, with the former being the first and the last steps of our “deform-to-fit” procedure and the latter encouraging tangential movement of the vertices to smooth out any roughness. In the smoothing step, vertices are processed one at a time with no particular ordering. The objective function is simply the smoothness term $\mathcal{S}(v)$.

4. Nonobtuse mesh decimation

Analogous to nonobtuse remeshing, nonobtuse mesh decimation can also be formulated via constrained optimization and solved in an iterative greedy fashion via edge collapse. For each candidate edge $e = (u, v)$, we compute the optimal position $w^* = \operatorname{argmin}_{w \in \mathcal{F}(e)} [Q_u(w) + Q_v(w)]$, for the unified vertex w , where $Q_u(w)$ and $Q_v(w)$ are quadric errors defined as before and the feasible region $\mathcal{F}(e)$ for edge e is determined similarly as in the vertex case, but by the one-ring neighbors of both u and v . The cost for collapsing edge (u, v) is $Q_u(w^*) + Q_v(w^*)$; this is set to ∞ if $\mathcal{F}(e) = \emptyset$. We iteratively collapse the edge with the lowest cost, and update the costs for the new edges as well as those edges that are influenced due to changes to their feasible regions.

5. Experimental results

Now we demonstrate several characteristics of our approach and the quality of the nonobtuse meshes produced. Unless otherwise specified, we choose $\alpha = 0.5$ for mesh optimization and only one step of Laplacian smoothing is applied. The approximation error ϵ , as defined for Figure 1, measures against the original mesh for remeshing and against the full-resolution nonobtuse mesh in decimation.

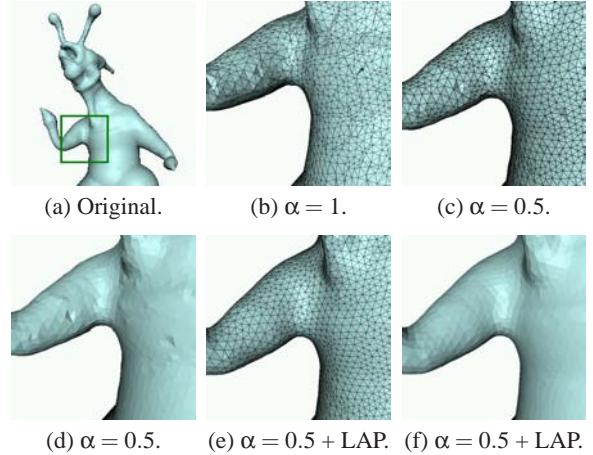


Figure 3: Importance of smoothness term and smoothing: Without the smoothness term in the objective function, we suffer from poor angle quality, shown in (b). Percentage of small angles (angles less than 30°) is $\gamma = 7.74\%$. (c) and (d): After weighing in the smoothness term: $\gamma = 0.12\%$. But geometric roughness is quite visible. (e) and (f): With constrained Laplacian smoothing added: $\gamma = 0.03\%$. And the rough geometric features are smoothed out.

In Figure 3, we show the importance of the smoothness term and smoothing. By comparing (b) and (c), it is apparent that including the smoothness term in the objective function dramatically improves the overall angle quality of the mesh. An additional smoothing step further improves the mesh quality as the roughness is smoothed out, as shown in (f). In Table 1, we report mesh quality and performance statistics from nonobtuse remeshing of six models. As it can be observed in conjunction with Figure 4, our algorithm is capable of producing good angle distributions and approximation quality. In particular, smoothing does help reduce the number of 90° angles as shown in the middle column of Figure 4. In our current implementation of nonobtuse decimation, no smoothness term is incorporated into the objective function. Hence, we see some spikes in the 90° bins.

6. Discussion and future work

In this paper, we propose a solution for guaranteed nonobtuse remeshing and mesh decimation. We design a modified, midpoint-based Marching Cubes algorithm to con-

Input mesh (#F)	#F	#V	Min. \angle	% small \angle 's	Metro ϵ	1: %VM	2: %VM	3: %VM
Armhand (50K)	18798	9401	26.14	0.05	1.00	310.6	397.9	487.3
Bigsmile (34.7K)	15380	7692	19.49	0.04	0.78	308.3	312.9	N/A
Baldhead (15.8K)	18704	9356	20.68	0.06	1.86	407.1	399.8	393.0
Horse (40K)	19880	9944	15.03	0.18	1.31	258.3	266.4	N/A
Man (29K)	68252	34122	10.52	0.14	0.55	328.7	285.1	N/A
Monster (32.5K)	48226	24115	19.79	0.03	0.64	294.8	374.3	N/A

Table 1: This table shows quality measures and performance statistics for nonobtuse remeshing. We report, from left to right: input mesh and its face count (#F); face (#F) and vertex (#V) counts of the output nonobtuse mesh; minimum angle (Min. \angle) in the nonobtuse mesh; percentage of angles less than 30° (% small \angle); approximation error (Metro ϵ); the number of vertices moved in each mesh optimization step (1, 2, and 3) as a percentage (“1:%VM”, “2:%VM”, and “3:%VM”) of (#V). Note that in many cases, only two optimization steps are needed; in that case, the “3:%VM” column will be marked by “N/A”.

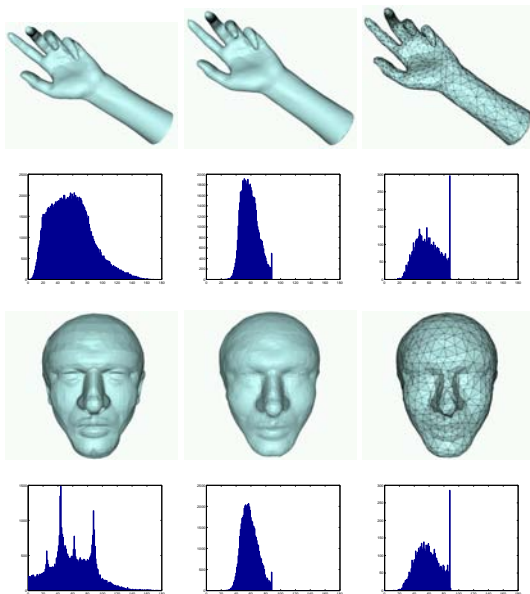


Figure 4: Visual results and histogram plots of angle distributions. From left to right: Original model; remeshed model; model after 90% of vertices decimated. The histogram plot associated with a particular model is given right below the model. Top: Armhand model. Bottom: Baldhead.

struct an initial nonobtuse mesh from which a series of local constrained optimizations are performed. Our framework is quite general and flexible and it provides several directions for future work. In particular, we wish to achieve efficient nonobtuse remeshing with angle bounds, nonobtuse or acute mesh generation from point clouds, and effective handling of mesh boundaries. In addition, lazy evaluations, early stopping, and adaptive smoothing and error reduction procedures can be used to improve efficiency. Other more challenging improvements, such as the use of adaptive grids in Marching Cubes and the preservation of sharp features in remeshing, are also worth investigating.

References

- [AUGA05] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: Recent advances in remeshing of surfaces, 2005. Part of the state-of-the-art report of the AIM@SHAPE EU network.
- [BGR88] BAKER B. S., GROSSE E., RAFFERY C. S.: Nonobtuse triangulation of polygons. *Disc. & Comp. Geom.* 3 (1988), 147–168.
- [BMR94] BERN M., MITCHELL S., RUPPERT J.: Linear-size nonobtuse triangulation of polygons. In *ACM Symp. on Comp. Geom. (SoCG)* (1994), pp. 221–230.
- [BS06] BOBENKO A. I., SPRINGBORN B. A.: A discrete laplacian-beltrami operator for simplicial surfaces, February 2006. preprint: arXiv:math.DG/0503219.
- [Che93] CHEW P.: Guaranteed-quality mesh generation for curved surfaces. In *ACM SoCG* (1993), pp. 274–280.
- [CRS98] CIGNONI P., ROCCHINI C., SCOPIGNO R.: Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* 17, 2 (1998), 167–174.
- [CS05] CHENG H.-L., SHI X.: Quality mesh generation for molecular skin surfaces using restricted union of balls. In *IEEE Visualization* (2005), pp. 399–405.
- [EDD*95] ECK M., DEROSE T., DUCHAMP T., HOPPE H., LOUNSBERRY M., STUETZLE W.: Multiresolution analysis of arbitrary meshes. In *SIGGRAPH* (1995), pp. 173–182.
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH* (1997), pp. 209–216.
- [GW03] GERTZ E. M., WRIGHT S. J.: Object-oriented software for quadratic programming. *ACM Trans. on Math. Software* 29 (2003), 58–81.
- [GY03] GU X., YAU S.-T.: Global conformal surface parameterization. In *Proc. of SGP* (2003), pp. 127–137.
- [KS98] KIMMEL R., SETHIAN J. A.: Computing geodesic paths on manifolds. *Proc. of National Academy of Science USA* 95 (1998), 8431–8435.
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH* (1987), pp. 163–169.
- [LZ06] LI J. Y. S., ZHANG H.: Guaranteed nonobtuse meshes via constrained optimization, 2006. Technical Report TR 2006-13, School of Computing Science, Simon Fraser University.