Adaptive Remeshing for Real-Time Mesh Deformation

Marion Dunyach[1,2], David Vanderhaeghe[2], Loïc Barthe[2], Mario Botsch[1]

[1]Computer Graphics & Geometry Processing, Bielefeld University, Germany
[2]IRIT, Université de Toulouse, France

Abstract

We present an adaptive isotropic remeshing technique that is fast enough to be used in interactive applications, such as mesh deformation or mesh sculpting. Previous real-time remeshing techniques are either not adaptive, hence requiring too many triangles, or make compromises in terms of triangle quality. High quality adaptive remeshing techniques are too slow for interactive applications. In this short paper we present a simple extension of a uniform remeshing approach that results in an efficient, yet high quality, curvature-adaptive remeshing.

1. Introduction

Triangle meshes are the predominant surface representation in computer graphics and geometric modeling. A high quality mesh on the one hand has to yield a sufficiently accurate approximation of the smooth underlying surface geometry, and on the other hand should consist of triangles that allow for numerically stable computations. This requires (a) to adapt the sampling density to the curvature of the underlying geometry and (b) to aim for equilateral triangles—both of which is the goal of *adaptive isotropic remeshing*.

During the last decade a large variety of remeshing approaches has been proposed. Below we discuss the most relevant works only, and refer the interested reader to the survey [AUGA08] and the book [BKP*10] for more details.

The approaches providing the highest mesh quality are typically based on the *Centroidal Voronoi Tessellation* (CVT): Earlier approaches make use of a 2D CVT computed in the parameter domain of a global parameterization [ACdVDI03] or a set of local parameterizations [SAG03, FAKG10]. Yan et al. [YLL*09] avoid costly parameterizations by computing the 3D CVT restricted to the surface. However, these approaches still are computationally involved, since they have to compute either the restricted Voronoi Diagram [YLL*09] or local/global planar parameterizations [SG03, SAG03, FAKG10], and therefore typically need a few minutes for processing moderately complex meshes. As a consequence, interactive applications involving large changes of the geometry, such as 3D mesh sculpting, are particularly challenging, since they require to frequently interleave remeshing with surface deformation while still providing *real-time feedback* to the designer.

A few real-time remeshing approaches for sculpting-like applications have been proposed. Most of these methods operate directly on the triangle mesh and use local modifications to optimize mesh connectivity and vertex distribution. However, they oftentimes trade mesh quality for processing speed. For instance, several methods only control edge lengths by splitting edges that are too long and collapsing edges that are too short [AWC04, KRK*06, SCC11]. While being very efficient, these methods neither optimize vertex valences nor vertex locations and therefore result in triangle shapes of sub-optimal quality. In order to address these issues, other approaches additionally incorporate edge flips for valence regularization and tangential relaxation for vertex placement [SG03, VRS03, BK04, vFTS06], yielding higher quality meshes with closer-to-equilateral triangles.

The majority of the above approaches concentrate on uniform remeshing, i.e., they try to achieve the same edge length for the whole mesh [VRS03,BK04,KRK*06,SCC11]. To be able to represent small geometric details, these approaches inevitably waste lots of triangles in less detailed surface regions. Other methods generate adaptive meshes by deciding for edge split/collapse not only based on edge length, but also on the angle between the endpoint normals [GD99, BK03, AWC04, vFTS06]. This, however, requires an additional threshold parameter and stretches triangles anisotropically along the direction of minimum curvature, which eventually degrades mesh quality.

In the following we present a simple curvature-adaptive remeshing algorithm, which is fast enough to be used in interactive applications and yields a mesh quality comparable to computationally much more expensive approaches.

DOI: 10.2312/conf/EG2013/short/029-032

2. Uniform Isotropic Remeshing

Our adaptive remeshing is a simple extension of the uniform remeshing framework proposed in [BK04, BKP*10]. Hence the general algorithm is identical to [BK04], which iterates the following operations 5–10 times:

1. **Edge Lengths:** Every edge e is collapsed if it is shorter than $4/3\,L$ and split if it is longer than $4/5\,L$, with L being the target edge length. The "heuristically optimal" thresholds $4/3L$ and $4/5L$ have been derived in [BK04].
2. **Vertex Valences:** Every edge is flipped if this operation decreases the squared difference of the valences of the four vertices of the two incident triangles to their optimal value, which is 6 in the interior and 4 on the boundary.
3. **Vertex Positions:** In order to improve the vertex distribution a tangential relaxation moves each vertex $\mathbf{x}_i$ to the weighted average $\mathbf{c}_i$ of its one-ring neighbors $\mathcal{N}_i$, projected into the tangent plane $(\mathbf{x}_i, \mathbf{n}_i)$:

$$\mathbf{c}_i = \frac{\sum_{j\in\mathcal{N}_i} w_j \mathbf{x}_j}{\sum_{j\in\mathcal{N}_i} w_j}, \quad \mathbf{x}_i \leftarrow \mathbf{c}_i + \mathbf{n}\mathbf{n}^T(\mathbf{x}_i - \mathbf{c}_i). \qquad (1)$$

The weights w_j typically are chosen as $w_j = 1$, i.e., $\mathbf{c}_i$ is simply the barycenter of the one-ring neighbors. The new position $\mathbf{x}_i$ can optionally be projected back onto the original surface, for which we efficiently find the surface triangle closest to $\mathbf{x}_i$ using a kD-tree.

3. Adaptive Isotropic Remeshing

Our main contribution is to replace the constant target edge length L of [BK04] by an adaptive sizing field $L(\mathbf{x})$ that is intuitive to control, simple to implement, and efficient to compute. As discussed in the introduction, splitting edges based on their length and the angle of their endpoint normals leads to anisotropically stretched triangles and requires one more threshold parameter for the allowed deviation of endpoint normals.

In contrast, our remeshing is based on just one very intuitive parameter: the approximation tolerance ε, i.e., the maximally allowed geometric deviation of the triangle mesh from the underlying smooth surface geometry. We first compute the curvature field of the input mesh and then derive the optimal local edge lengths, i.e., the sizing field $L(\mathbf{x})$, from the maximum curvature and the approximation tolerance.

As shown in Figure 1, left, in 2D the maximum edge length l for approximating a circular arc up to an error ε can simply be computed from the Pythagorean theorem, as was also observed by Alliez et al. [ACSD*03]:

$$r^2 = (r-\varepsilon)^2 + \left(\frac{l}{2}\right)^2 \quad \Leftrightarrow \quad l = 2\sqrt{2r\varepsilon - \varepsilon^2}. \qquad (2)$$

When approximating a general planar curve by a poly-line, the radius r corresponds to the radius of the osculating circle, i.e., the inverse of the local curvature $r = 1/\kappa$ [dC76].

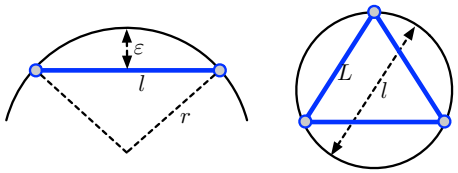

Figure 1: Left: *How to determine the edge length l for approximating a circular arc up to an error tolerance ε.* Right: *For surfaces the edge length l has to be scaled to yield the target edge length L for an equilateral triangle. In the surface case, the left and right figures can be considered a cross-section or a top view of each other.*

This result can be transferred from 2D curves to 3D surfaces by considering the 2D configuration of Figure 1 to be the planar cross-section used to compute the normal curvatures [dC76]. Since we want an *isotropic* remeshing, the sizing field $L(\mathbf{x}_i)$ at a vertex $\mathbf{x}_i$ should be direction-independent. We therefore conservatively pick the cross-section resulting in the smallest edge length l, which is nothing else than using the maximum absolute curvature $\kappa = \max\{|\kappa_{\min}|, |\kappa_{\max}|\}$ to determine the radius r in (2).

As shown in Figure 1, the cross-section edge length l has to be scaled by $\sqrt[3]{12}$ to get the edge length for an equilateral triangle with circum-diameter l. This finally gives a simple equation to compute the sizing value for each vertex from its maximum absolute curvature κ_i and the error tolerance ε:

$$L(\mathbf{x}_i) = \sqrt{6\varepsilon/\kappa_i - 3\varepsilon^2}.$$

In order to avoid overly large or small triangles, the resulting sizing field can be clamped to user-specified bounds, such that $L(\mathbf{x}_i) \in [L_{\min}, L_{\max}]$.

For computing the discrete maximum absolute curvature per vertex $\mathbf{x}_i$ we employ the standard cotangent discretization, and compute the maximum absolute curvature κ from mean curvature H and Gaussian curvature K [MDSB03]:

$$H_i = \frac{1}{2}\|\Delta\mathbf{x}_i\|, \qquad (3)$$

$$K_i = \frac{1}{A_i}\left(2\pi - \sum_{j\in\mathcal{N}(i)} \theta_i\right), \qquad (4)$$

$$\kappa_i = H_i + \sqrt{H_i^2 - K_i}, \qquad (5)$$

In the above equation θ_i are the incident triangle angles around vertex $\mathbf{x}_i$ and A_i is its Voronoi area.

This yields a sizing value for each vertex, and we conservatively determine the sizing value $L(e)$ for an edge $e = (\mathbf{x}_1, \mathbf{x}_2)$ as the minimum of the sizing of both endpoints:

$$L(e) = \min\{L(\mathbf{x}_1), L(\mathbf{x}_2)\}.$$

This value $L(e)$ is to replace the constant target length L in the edge length correction (Step 1 in Section 2).

As the target edge lengths are not constant anymore, the tangential relaxation (Step 3 in Section 2) has to be adjusted in order to preserve the relative edge sizing. To this end the barycenter computation of (1) is replaced by:

$$\mathbf{c}_i = \frac{\sum_{j \in \mathcal{T}_i} |t_j| \, L(\mathbf{b}_j) \, \mathbf{b}_j}{\sum_{j \in \mathcal{T}_i} |t_j| \, L(\mathbf{b}_j)} . \quad (6)$$

Here we adapt the smoothing scheme for computing a so-called *Optimal Delaunay Triangulation* [CH11]: The point $\mathbf{c}_i$ is computed as the average of the barycenters $\mathbf{b}_j$ of the incident triangles $t_j \in \mathcal{T}_i$, weighted by the triangle area and the sizing field at the barycenter $\mathbf{b}_j$ (average of the sizing field of the triangle vertices). This is similar to a 2D version of the tetrahedral meshing of Alliez et al. [ACSYD05], but we use the element barycenters instead of their circumcenters for reasons of robustness and simplicity [CH11].

These two updates to steps 1 and 3 of the uniform remeshing of Section 2 yield an efficient and high quality adaptive remeshing scheme for smooth surfaces. If feature edges (e.g., edges with large dihedral angle) are to be preserved, only some minor adjustments are necessary [VRS03, BKP*10]: First, edge splits, collapses, and flips that would destroy feature edges have to be discarded. Second, corner vertices (>2 incident feature edges) do not move, feature vertices (2 incident feature edges) only move along their feature lines. For feature-preserving *adaptive* remeshing, two more adjustments are required:

1. In order to avoid high sampling densities near feature edges, the sizing values for feature vertices are computed as the average of their non-feature neighbors.
2. The tangential relaxation in (6) is performed only along the feature line by replacing the incident triangles by the two incident feature edges, and triangle areas/barycenters by edge lengths and midpoints.

4. Mesh Sculpting

The development of our remeshing framework was motivated by its application in an interactive mesh modeling application, where every small step of mesh deformation is followed by one iteration of adaptive remeshing. While we describe our particular mesh sculpting method below, we point out that our remeshing technique can be used in combination with any interactive mesh deformation method.

Our sculpting approach follows the standard handle-based metaphor [BS08]: The user selects a handle point or a handle region, and the deformable region of interest (ROI) is defined as all vertices within a specified geodesic distance from the handle. Through mouse movements the user intuitively controls the affine transformation of the handle, consisting of translation, rotation, and scaling. In order to smoothly blend this transformation within the ROI we compute a smooth bi-harmonic scalar field that is 1 at the handle and 0 at the fixed boundary of the ROI, which amounts to solving a sparse bi-Laplacian linear system [BS08]. The translation and scale components of the handle transformation are then blended linearly, while the rotation is blended using spherical linear interpolation [BK03]. In contrast to most deformation approaches, we recompute the ROI and the scalar field in each step, since this allows for large-scale edits.

Model	#Vert.	Time	Min ∠	Ave ∠
Feline [SG03]	11k	74s	7.4°	48°
Feline [SAG03]	21k	120s	3.9°	49°
Feline [vFTS06]	24k	2.9s	0.6°	36°
Feline [SCC11]	21k	0.15s	2.1°	41°
Feline [ours]	21k	1.4s	26°	51°
Elk [YLL*09]	31k	132s	36°	54°
Elk [ours]	31k	2.0s	32°	51°
Horse [SG03]	5.6k	28s	9.1°	50°
Horse [FAKG10]	6k	16s	30°	52°
Horse [ours]	6k	0.94s	28°	51°
Joint [YLL*09]	3.1k	49s	32°	53°
Joint [ours]	3.1k	0.64s	15°	47°
Fandisk [SG03]	5.1k	17s	17°	49°
Fandisk [YLL*09]	3k	40s	24°	52°
Fandisk [FAKG10]	4k	3s	21°	53°
Fandisk [ours]	4k	0.25s	18°	49°

Table 1: *Comparison with other remeshing approaches, listing model complexity after remeshing, processing time, minimum angle, and average minimum angle over all triangles. Our timings were measured on a Dell T7500 workstation with Intel Xeon E5645 2.4 GHz CPU and 6GB RAM.*

In every sculpting step we ensure that the mesh has a sufficient resolution to properly represent the desired deformation. To this end we tentatively split each edge of the ROI at its midpoint and test whether the deformed midpoint deviates by more than our threshold ε from the midpoint of the deformed edge endpoints. If it does, we accept the split, otherwise it is canceled. This concept has been successfully used in other approaches [AWC04, vFTS06], and has the advantage that it does not require an additional parameter, but instead relies on the only parameter ε.

5. Results and Discussion

Table 1 compares our remeshing (5 iterations, with back-to-surface projection) to several other approaches, for which models and/or results are known. Since the experiments have been performed on very different machines the timings are not directly comparable. However, it can still be observed that our method is at least one order of magnitude faster than other high quality remeshers—while being comparable in terms of mesh quality. Furthermore, our remeshing yields a significantly higher quality—as measured in particular by the smallest angle—than (our implementation of) other real-time remeshers [vFTS06, SCC11].

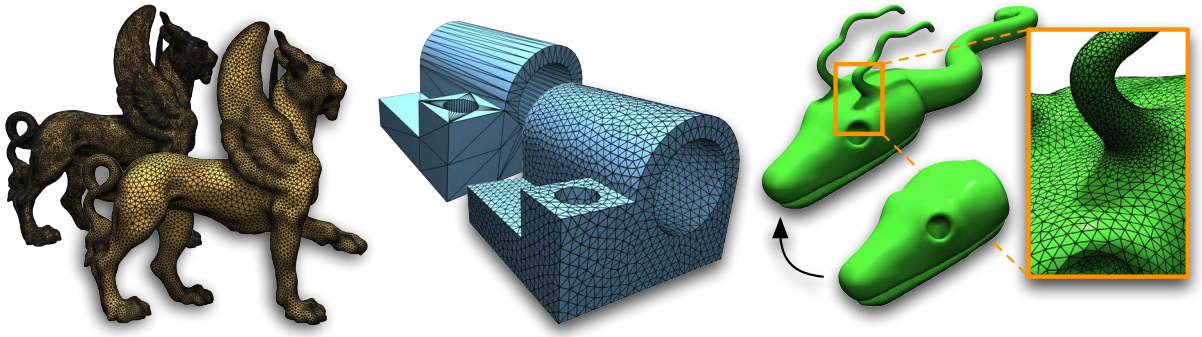

Figure 2: *Adaptive remeshing of the Feline model (left). Uniform remeshing of the Joint model (center). Starting from a snake head, a new model is sculpted by adding tail and horns (right). The snake modeling session is also shown in the video.*

In our sculpting application, each deformation step only slightly modifies the geometry while generating a smooth surface. Hence, alternating each small deformation step with one remeshing iteration is sufficient to produce a high quality mesh. In addition, performance can be increased by disabling the back-to-surface projection without visual loss of quality. The accompanying video shows live screen captures for twisting a bar while preserving its sharp features and for modeling the snake model of Figure 2.

In terms of limitations we note that our approximation tolerance ε is not satisfied exactly, mainly due to the discrete nature of the curvature and sizing field computation. As a consequence, while the RMS error is below ε, the maximum Hausdorff error typically is not. Our current implementation is running on a single core only. In the future we therefore plan to further increase performance through multi-core parallelization, as also done in [FAKG10].

References

[ACdVDI03] ALLIEZ P., COLIN DE VERDIÈRE E., DEVILLERS O., ISENBURG M.: Isotropic surface remeshing. In *Proceedings of the Shape Modeling International* (2003). 1

[ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. *ACM Transactions on Graphics 22*, 3 (2003). 2

[ACSYD05] ALLIEZ P., COHEN-STEINER D., YVINEC M., DESBRUN M.: Variational tetrahedral meshing. *ACM Transactions on Graphics 24*, 3 (2005). 3

[AUGA08] ALLIEZ P., UCELLI G., GOTSMAN C., ATTENE M.: Recent advances in remeshing of surfaces. In *Shape Analysis and Structuring, Mathematics and Visualization* (2008), Springer. 1

[AWC04] ANGELIDIS A., WYVILL G., CANI M.-P.: Sweepers: Swept user-defined tools for modeling by deformation. In *Shape Modeling and Applications* (2004). 1, 3

[BK03] BENDELS G., KLEIN R.: Mesh forging: Editing of 3d-meshes using implicitly defined occluders. In *Symposium on Geometry Processing* (2003). 1, 3

[BK04] BOTSCH M., KOBBELT L.: A remeshing approach to multiresolution modeling. In *Symposium on Geometry Processing* (2004). 1, 2

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. A K Peters, 2010. 1, 2, 3

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008). 3

[CH11] CHEN L., HOLST M.: Efficient mesh optimization schemes based on optimal delaunay triangulations. *Computer Methods in Applied Mechanics and Engineering 200* (2011). 3

[dC76] DO CARMO M. P.: *Differential Geometry of Curves and Surfaces*. Prentice Hall, Englewood Cliffs, NJ, 1976. 2

[FAKG10] FUHRMANN S., ACKERMANN J., KALBE T., GOESELE M.: Direct resampling for isotropic surface remeshing. In *Vision, Modeling, and Visualization* (2010). 1, 3, 4

[GD99] GAIN J. E., DODGSON N. A.: Adaptive refinement and decimation under free-form deformation. *Eurographics UK 99, Cambridge 17* (1999). 1

[KRK*06] KIL Y., RENZULLI P., KREYLOS O., HAMANN B., MONNO G., STAADT O.: 3d warp brush modeling. *Computers & Graphics 30*, 4 (2006). 1

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer-Verlag, 2003. 2

[SAG03] SURAZHSKY V., ALLIEZ P., GOTSMAN C.: Isotropic remeshing of surfaces: A local parameterization approach. In *In Proceedings of 12th International Meshing Roundtable* (2003). 1, 3

[SCC11] STĂNCULESCU L., CHAINE R., CANI M.-P.: Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics 35*, 3 (2011). 1, 3

[SG03] SURAZHSKY V., GOTSMAN C.: Explicit surface remeshing. In *Symposium on Geometry processing* (2003). 1, 3

[vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM Transactions on Graphics 25*, 3 (2006). 1, 3

[VRS03] VORSATZ J., RÖSSL C., SEIDEL H.-P.: Dynamic remeshing and applications. In *Symposium on Solid Modeling and Applications* (2003). 1, 3

[YLL*09] YAN D.-M., LÉVY B., LIU Y., SUN F., WANG W.: Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Computer Graphics Forum 28*, 5 (2009). 1, 3