Mesh Simplification With Curvature Error Metric

C. Michaud[1,2], N. Mellado[1] and M. Paulin[1]

[1]IRIT - Université de Toulouse, [2]Oktal-SE

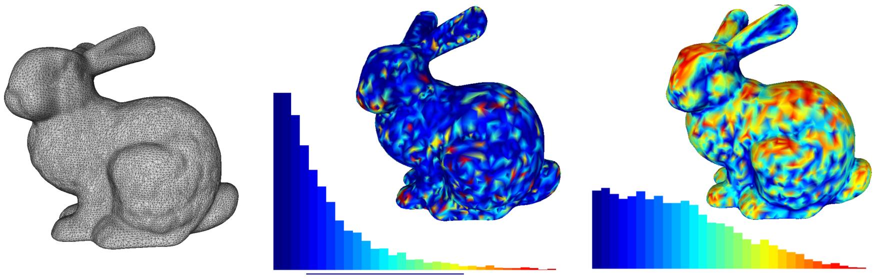

Figure 1: *The 40k faces Stanford bunny (left most). In the middle our simplification down to 12k faces and the corresponding error distribution. On the right the classic QEM simplification (12k faces) and its error distribution.*

Abstract

Progressive meshes algorithms aim at computing levels of detail from a highly detailed mesh. Many of these algorithms are based on a mesh decimation technique, generating coarse triangulation while optimizing for a particular metric which minimizes the distance to the original shape. However these metrics do not robustly handle high curvature regions, sharp features, boundaries or noise. We propose a novel error metric, based on algebraic spheres as a measure of the curvature of the mesh, to preserve curvature along the simplification process. This metric is compact, does not require extra input from the user, and is as simple to implement as a conventional quadric error metric.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Curve, surface, solid, and object representations

1. Introduction

Mesh simplification is a standard processing step in polygon mesh processing, which aims at adapting the level of detail of a mesh w.r.t. an input criterion, e.g. a target number of faces.

The major class of simplification algorithms is based on an iterative process collapsing edges in a resulting position minimizing an error metric. This error was first defined by Hoppe [Hop96] as the distance from the target point to the faces adjacent to the collapsed edge (one-ring).

Garland and Heckbert [GH97] introduced the Quadric Error Metric (QEM) to improve the performances of the *point-to-one-ring* distance minimization. Based on a vertex-to-plane squared distance error, the QEM is expressed as a quadric form, making it easy to minimize. The main drawback of QEM and its variants is their locality, and their limited description power for curved shapes, as the curvature information of collapsed faces is lost throughout the simplification process. In order to better represent smooth and curved shapes, Thiery et al. [TGB13] proposed a new metric in which the volume of the input mesh is described by a set of spheres attached to the mesh vertices and interpolated along the edges. Even though the overall volume and the curvatures are explicitly described and thus preserved, they do not handle sharp features and holes. The representation of the mesh as a set of pre-computed canonical proxies has been recently studied [SLA15] as a way of better preserving the structure of the mesh. A more global error metric is proposed, by extending the QEM evaluation to quadrics generated from both the mesh geometry and fitted planar proxies. The robustness of the approach is deferred to the primitive fitting step, where noise and sharp features might be handled by dedicated algorithms.

In this poster, we propose to constrain the simplification process w.r.t. the curvature of the input detailed mesh. We introduce a curvature measure based on Algebraic Point Set Surfaces [GG07], evaluated from the original mesh. During the simplification, this curvature information is propagated by interpolating spheres. We define a new error metric that aims at representing and preserving the curvature information inside the faces. Our approach follows the standard simplification scheme used by QEM and its variants, which ease its adoption in existing geometry processing pipelines.

DOI: 10.2312/egp.20171040

2. Curvature error metric

2.1. Mesh simplification algorithm

We run through our simplification algorithm in a similar way to [GH97]. It can be summarized as follows:

- Estimate curvature for all the initial vertices $\mathbf{x}_i$.
- Compute the optimal contraction $\mathbf{x}_\alpha$ for each edge $(\mathbf{x}_1, \mathbf{x}_2)$ and cost of contracting that pair.
- Place all the pairs in a heap keyed on cost with the minimum cost pair at the top.
- Iteratively remove the pair $(\mathbf{x}_1, \mathbf{x}_2)$ of least cost from the heap, contract this pair, and assign to the target $\mathbf{x}_\alpha$ the curvature information best fitting the former surface.

We now present how we estimate local curvature and how we find the optimal edge contraction point and its associated cost.

2.2. Curvature estimation

We estimate curvature from Algebraic Point Set Surfaces (APSS [GG07]), which defines Moving Least Squares surfaces with algebraic sphere primitives. APSS elegantly handles curved areas as well as planar regions and inflection points since the algebraic sphere naturally degenerates to a plane, as opposed to the conventional sphere definition (radius and center) used in [TGB13]. APSS surfaces can be reconstructed with varying fitting kernel sizes to handle noise and sharp features.

From the input mesh and its attached APSS surface, we sample the algebraic sphere S_i for each vertex $\mathbf{x}_i$ of the mesh. Computing the curvature at any point consists in interpolating the spheres along the edges and faces, e.g. for the edge $(\mathbf{x}_1, \mathbf{x}_2)$, we have $S_\alpha = S_1 + \alpha(S_2 - S_1)$ (Figure 2).

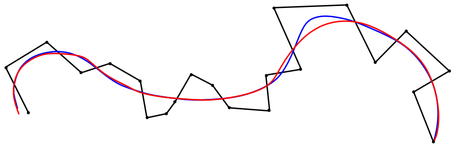

Figure 2: *Reconstructed APSS curve (blue) from the polyline (black). We show in red the curve generated by the spheres interpolated with our approach.*

2.3. Cost of contraction and optimal target position

In contrast to point-plane distance [Hop96], we measure the cost of contracting an edge taking into account the local curvature, thus preserving curved features of the mesh. More specifically, the contraction cost is defined as the volume between the sphere representing the curvature and the mesh. Thanks to the algebraic formulation, the distance from a point $\mathbf{x}$ to the sphere is obtained by computing the field value $S(\mathbf{x})$. We compute the cost by integrating the distance along edges and faces. For an edge $(\mathbf{x}_1, \mathbf{x}_2)$, the area is:

$$\int_0^1 S(\mathbf{x}_1 + \alpha(\mathbf{x}_2 - \mathbf{x}_1))d\alpha \, \|\mathbf{x}_2 - \mathbf{x}_1\|$$

Even if we would ideally compute a target position in an arbitrary position, we consider in this work the resulting position $\mathbf{x}_\alpha$ on $(\mathbf{x}_1, \mathbf{x}_2)$. We obtain it by minimizing the following energy (Figure 3):

$$\underset{\alpha}{\operatorname{argmin}}\Big(\int_0^1 S_\alpha(\mathbf{x}_0 + \alpha(\mathbf{x}_\alpha - x_0))d\alpha \, \|\mathbf{x}_\alpha - \mathbf{x}_0\| + \int_0^1 S_\alpha(\mathbf{x}_\alpha + \alpha(\mathbf{x}_3 - \mathbf{x}_\alpha))d\alpha \, \|\mathbf{x}_3 - \mathbf{x}_\alpha\| \Big)$$

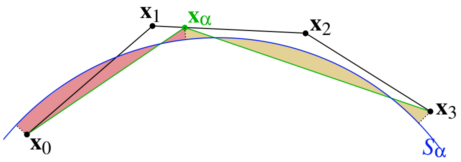

Figure 3: *Error caused by the contraction of* $(\mathbf{x}_1, \mathbf{x}_2)$ *on* $\mathbf{x}_\alpha$*. In red the distance area between* $(\mathbf{x}_0, \mathbf{x}_\alpha)$ *and* S_α*, in yellow the distance area between* $(\mathbf{x}_\alpha, \mathbf{x}_3)$ *and* S_α*.*

Practically, we explicitly search for the global minimum of the energy along the edge and pick the associated α. After a collapse, we assign to the new point $\mathbf{x}_\alpha$ the primitive $S_{\alpha_{min}} = S_1 + \alpha_{min}(S_2 - S_1)$. By interpolating the spheres we avoid the re-evaluation of the MLS surface.

3. Results

We used the standard APSS surface on smooth objects, and an other variant to handle sharp edges (see Figure 4). We show in Figures 1 and 4 that with the same polygon count, our approach better preserves the overall shape than simplification using QEM, since we have more vertices with a low error.

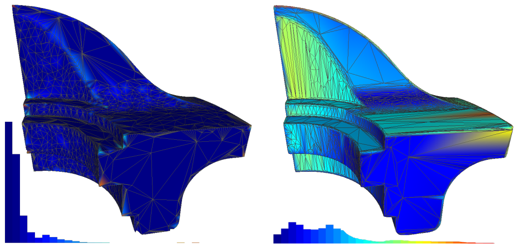

Figure 4: *Simplification of Fandisk. On the left our simplification based on algebraic sphere. On the right the QEM simplification. The histograms show the associated error distributions.*

4. Discussion

We presented a new error metric for mesh simplification which preserves local curvature. Thanks to the properties of interpolated algebraic sphere, the curvature is easily computed. While currently we limit the resulting vertex position on the collapsing edge, future work will include finding the optimal position in 3D by minimizing a volume. We also want to investigate adaptive kernel size when computing the algebraic spheres w.r.t. the surface features.

References

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 1, 2

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 209–216. 1, 2

[Hop96] HOPPE H.: Progressive meshes. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 99–108. 1, 2

[SLA15] SALINAS D., LAFARGE F., ALLIEZ P.: Structure-Aware Mesh Decimation. *Computer Graphics Forum* (Jan. 2015), 20. 1

[TGB13] THIERY J.-M., GUY E., BOUBEKEUR T.: Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 178:1–178:12. 1, 2