

# Subdivision Surface Simplification

Won-Ki Jeong, Kolja Kähler, and Hans-Peter Seidel

Max-Planck-Institut für Informatik, Saarbrücken, Germany

E-mail: {jeong, kaehler, hpseidel}@mpi-sb.mpg.de

## Abstract

*A modified quadric error metric (QEM) for simplification of Loop subdivision surfaces is presented. The suggested error metric not only measures the geometric difference but also controls the smoothness and well-shapedness of the triangles that result from the decimation process. Minimizing the error with respect to the original limit surface, our method allows for drastic simplification of Loop control meshes with convenient control over the reproduction of sharp features.*

**Keywords:** subdivision surface, mesh simplification, Loop scheme, quadric error metric

## 1 Introduction

Subdivision surfaces have become increasingly popular due to their ability to bridge the gap between polygonal meshes and higher-order surfaces. Since they are constructed by repeated refinement of triangle meshes up to an arbitrarily close approximation to their smooth limit surface, they also provide an effective means to control accuracy and efficiency of display in a systematic manner. Since a subdivision surface is always smooth in the limit, an overly complex control mesh can often be drastically simplified with only minor changes to the look of the surface.

In this paper, we introduce a simplification error metric with respect to a Loop limit surface, based on the quadric error metric (QEM) method [7], to simplify the control mesh of a Loop surface. We also include the possibility to control the smoothness and well-shapedness of the triangles at a given target triangle count. With a single parameter, the user can adjust the simplification of the subdivision surface for accurate reproduction of surface features, or for a smoother representation. This is useful to produce visually more pleasing results or enable further processing such as re-sampling for displacement computation.

## 2 Related Work

Subdivision surfaces combine the easy handling of meshes with the well-defined properties of a parametric surface. Defined over an initial control mesh, an arbitrarily close approximation to a smooth limit surface can be generated by repeatedly refining the mesh using simple rules. Many subdivision schemes are inspired by splines, and generalize to spline curves and surfaces: Catmull and Clark described a generalization of bicubic tensor product B-splines [2], and the Doo-Sabin scheme generalizes to biquadratic tensor product B-splines [5]. The Loop scheme [14] is based on the three-directional box spline. Many other schemes were proposed until today, among these the extensions of the Loop scheme [1, 9], the Butterfly scheme [6], and variational schemes [12].

Mesh decimation reduces the complexity of a given input mesh by removal of detail information, i.e. vertices and triangles. Some typical areas of application are oversampled scan data, over-tessellated surfaces, level-of-detail rendering, and progressive transmission. The problem of mesh decimation is well understood and extensive literature exists on the subject [3, 4, 7, 8, 13, 15]. Modifications of the quadric error metric [7] in the context of subdivision surface fitting have been proposed. To reconstruct a B-Spline surface from a triangle mesh, the metric has been adapted to the Doo-Sabin scheme [16]. In [11], a triangle mesh is gradually converted to a Loop control mesh while simultaneously simplifying it.

## 3 QEM for Loop surfaces

### 3.1 Notation

In the following, mesh vertices, edges, and triangles will be identified by an index, usually  $v$ ,  $e$  and  $t$ , respectively. The *mesh* position of a vertex  $v$  is written as  $\mathbf{p}_v$ , or just as  $\mathbf{p}$ , if no particular vertex is referenced. For a control mesh vertex  $v$ , its *limit* position on the subdivision surface is written  $\mathbf{p}_{v,\infty}$ . In the following discussion, all vertex positions are given as a vector in homogeneous coordinates in  $\mathbb{R}^4$ , whose

elements are identified by a superscript  $\in \{0, \dots, 3\}$ , as in  $\mathbf{p}_v^0$ . For a vertex  $i$ ,  $\text{fan}(i)$  returns the fan of triangles around  $v$ .  $\text{adj}(v)$  returns the set of vertices directly connected to  $v$  by an edge, and  $\text{val}(v)$  returns the size of this set.

### 3.2 QEM-based mesh simplification

The QEM method was originally defined for simplification of triangle meshes. Each vertex of the original mesh sits at the intersection of the planes of its adjacent triangles, which amounts to a summed distance of zero to all these planes. The squared distance of a vertex with position  $\mathbf{p}$  to one such plane  $\mathbf{P} = [a \ b \ c \ d]$  is elegantly expressed in the quadratic form  $d(\mathbf{p}) = \mathbf{p}^\top \mathbf{Q} \mathbf{p}$ , where

$$\mathbf{Q} = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & dc \\ ad & bd & cd & d^2 \end{bmatrix}$$

Summing up a set of quadrics  $\mathbf{Q}_i$  results in a new quadric  $\tilde{\mathbf{Q}} = \sum_i \mathbf{Q}_i$ , which can be used to measure the sum of squared distances to the set of planes defined by the  $\mathbf{Q}_i$ .

The QEM simplification algorithm is initialized by assigning to each mesh vertex the sum of quadrics representing the planes of its adjacent triangles. For evaluating a prospective collapse of edge  $e$  with end vertices  $u$  and  $v$ , the quadrics  $\mathbf{Q}_u$  and  $\mathbf{Q}_v$  are summed up:  $\mathbf{Q}_{uv} = \mathbf{Q}_u + \mathbf{Q}_v$ . The position of the collapsed vertex  $\mathbf{p}_{uv}$  that minimizes the squared distance to the set of planes defined by  $\mathbf{Q}_{uv}$  is then calculated by solving the linear equation  $\nabla d(\mathbf{p}_{uv}) = 0$ . The value of  $d(\mathbf{p}_{uv})$  represents the estimated surface approximation error, and is used to rank this collapse operation. In the main loop of the algorithm, when the current “best collapse” is read from the queue and carried out, the new vertex position is set to the optimal position, and the potential collapses along the changed edges are re-evaluated. Due to the simple and fast evaluation of the error metric, QEM simplification is very fast, and produces generally pleasing results.

### 3.3 Setup for Loop surface simplification

The original QEM error metric can be adapted for use with a subdivision control mesh. A quadric does now not measure the distance of a vertex to the mesh surface itself, but the distance to the limit surface  $\mathcal{S}$  defined by the Loop scheme on the mesh.

We thus define the initial quadric of a control vertex  $v$  through an approximation of the limit surface around that vertex, whose triangles are used to set up the plane equations. First, we subdivide the control mesh  $\mathcal{M}_c$  once to obtain the subdivided mesh  $\mathcal{M}_s$ . We then project the vertices  $\mathbf{p}_v$  to their limit positions  $\mathbf{p}_{v,\infty}$ . To set up the initial quadric

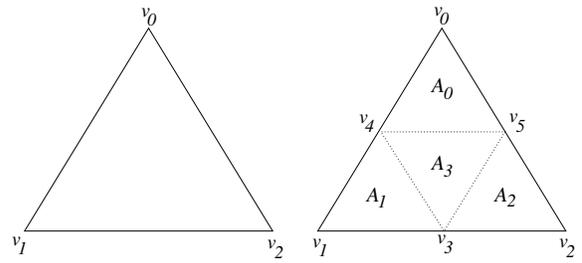


Figure 1. Initial subdivision of a control mesh triangle for setting up the quadric error metric

$\mathbf{Q}_v$  for a mesh vertex  $v$ , we first initialize all  $\mathbf{Q}_v$  to  $\mathbf{0}$ . For each control mesh triangle  $t$ , the sub-triangles in  $t$  are labelled  $A_{i \in \{0, \dots, 3\}}^t$  according to Figure 1. Iterating over all  $t$ , the *fundamental quadric* [7] computed for each triangle  $A_i^t$  is then added to vertex  $i$  of  $t$ .  $\mathbf{Q}_v$  thus contains the contributions of the subdivided and projected sub-triangles adjacent to  $v$ . Each  $\mathbf{Q}_v$  is now used to define an error metric with respect to the limit position of its associated control mesh vertex  $v$ . Since the planes that are used to build the  $\mathbf{Q}_v$  in fact include  $\mathbf{p}_{v,\infty}$ , the error will initially be zero. The error quadric for the center piece  $A_3^t$  of each triangle  $t$  is initially assigned to the adjacent edges of the mesh. Such an “edge quadric” is added to the quadric of the vertex that results from a collapse of that edge.

### 3.4 Distance metric

The error quadric that has been defined in the previous section can be used to compute the surface difference after an edge collapse. For the vertex  $v$  after the edge collapse depicted in Figure 2, a distance value  $d_v$ , representing the deviation of the limit position of control vertex  $v$  from its original position, is computed:

$$d_v = \mathbf{p}_{v,\infty}^\top \mathbf{Q}_v \mathbf{p}_{v,\infty}.$$

To measure this error, given the position  $\mathbf{p}_v$  on the control mesh  $\mathcal{M}_c$ , we express  $\mathbf{p}_{v,\infty}$  as a linear combination of  $\mathbf{p}_v$ :

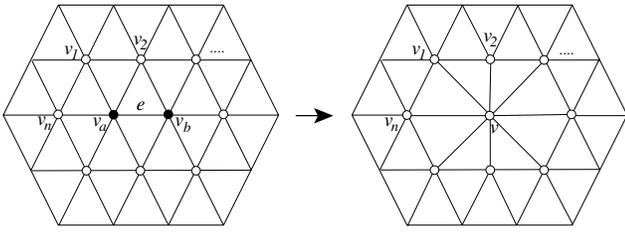
$$d_v = (\lambda_v \mathbf{p}_v + \mathbf{q}_v)^\top \mathbf{Q}_v (\lambda_v \mathbf{p}_v + \mathbf{q}_v),$$

where  $\lambda_v \in \mathbb{R}$ , and  $\mathbf{q}_v \in \mathbb{R}^4$  are computed from the one-neighborhood of  $v$ . If  $v$  is an inner vertex, and given  $n = \text{val}(v)$ , we set

$$\varphi := \frac{1}{n} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right), \chi := \frac{1}{3/8\varphi + n},$$

and obtain

$$\lambda_v = (1 - n\chi), \mathbf{q}_v = \chi \sum_{w \in \text{adj}(v)} \mathbf{p}_w.$$



**Figure 2. collapsing an edge  $e = (v_a, v_b)$  into a vertex  $v$  influences the two-neighborhood of  $v$ . Left: before edge collapse. Right: after edge collapse. The dots mark the vertices where  $Q$  has to be evaluated.**

If  $v$  is on a boundary, then the limit position is evaluated using the two neighboring boundary vertices  $u$  and  $w$ .  $\lambda_v$  is set to  $2/3$ , and  $\mathbf{q}_v = 1/5(\mathbf{p}_u + \mathbf{p}_w)$ . By expressing the linear combination as a matrix

$$\mathbf{T}_v = \begin{bmatrix} \lambda_v & 0 & 0 & \mathbf{q}_v^0 \\ 0 & \lambda_v & 0 & \mathbf{q}_v^1 \\ 0 & 0 & \lambda_v & \mathbf{q}_v^2 \\ 0 & 0 & 0 & \mathbf{q}_v^3 \end{bmatrix},$$

this can be expressed more concisely as

$$d_v = \mathbf{p}_v^T \mathbf{T}_v^T \mathbf{Q}_v \mathbf{T}_v \mathbf{p}_v.$$

Since the change of a vertex on the control mesh influences its two-neighbor region according to the support of the Loop scheme, we take into account not only the center vertex  $v$  but also its neighbor vertices to measure the distance difference, as shown in Figure 2 right. Matrices  $\mathbf{T}_w$  are computed for the vertices  $w \in \text{adj}(v)$ :

$$\lambda_w = \chi, \mathbf{q}_w = (1 - n\chi)\mathbf{p}_w + \chi \sum_{u \in \text{adj}(w)/v} \mathbf{p}_u.$$

Finally, the distance error metric is defined as

$$\tilde{\mathbf{Q}}_v = \sum_{w \in \text{adj}(v)} \mathbf{T}_w^T \mathbf{Q}_w \mathbf{T}_w + \mathbf{T}_v^T \mathbf{Q}_v \mathbf{T}_v. \quad (1)$$

### 3.5 Smoothness metric

When we use the quadric error metric as derived above, it shows that the surface resulting from simplification will in some cases result in very sharply preserved features on an otherwise smooth surface. Hence, we suggest a simple but reasonable measure for smoothness similar to [10]. Essentially, we compute the new position for a collapsed vertex by a weighted average of its optimum limit position as

obtained from the distance error metric and the mean limit position of its adjacent control vertices.

Similar to what was discussed in 3.4, the limit position of a vertex  $w \in \text{adj}(v)$  can be computed by applying an affine transformation  $\mathbf{T}_w$  to  $\mathbf{p}_v$ :

$$\mathbf{p}_{w,\infty} = \mathbf{T}_w \mathbf{p}_v.$$

The average limit position  $\bar{\mathbf{p}}_{v,\infty}$  computed from the vertices adjacent to  $v$  is then:

$$\bar{\mathbf{p}}_{v,\infty} = \frac{1}{\text{val}(v)} \sum_{w \in \text{adj}(v)} \mathbf{p}_{w,\infty} = \frac{1}{\text{val}(v)} \sum_{w \in \text{adj}(v)} (\mathbf{T}_w \mathbf{p}_v).$$

Combining the above two equations, we obtain a matrix  $\tilde{\mathbf{T}}_v$ , that transforms  $\mathbf{p}_v$  to the difference vector from  $\mathbf{p}_{v,\infty}$  to  $\bar{\mathbf{p}}_{v,\infty}$ :

$$\begin{aligned} \bar{\mathbf{p}}_{v,\infty} - \mathbf{p}_{v,\infty} &= \frac{1}{\text{val}(v)} \sum_{w \in \text{adj}(v)} (\mathbf{T}_w \mathbf{p}_v) - \mathbf{T}_v \mathbf{p}_v \\ &= \left( \frac{1}{\text{val}(v)} \sum_{w \in \text{adj}(v)} \mathbf{T}_w - \mathbf{T}_v \right) \mathbf{p}_v \\ &= \tilde{\mathbf{T}}_v \mathbf{p}_v. \end{aligned}$$

The squared distance between  $\mathbf{p}_{v,\infty}$  and  $\bar{\mathbf{p}}_{v,\infty}$  is thus easily calculated as  $\mathbf{p}_v^T \tilde{\mathbf{T}}_v^T \tilde{\mathbf{T}}_v \mathbf{p}_v$ . Finally, the quadric smoothness metric for vertex  $v$  is now defined as

$$\mathbf{Q}_v^s = \tilde{\mathbf{T}}_v^T \tilde{\mathbf{T}}_v. \quad (2)$$

### 3.6 Simplification Algorithm

We define the final error quadric  $\tilde{\mathbf{Q}}_v^s$  by combining Equations 1 and 2 :

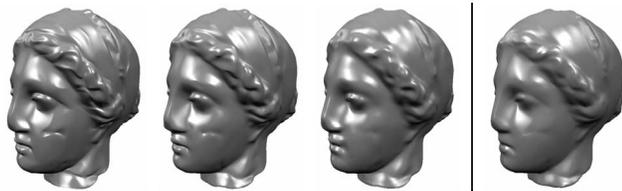
$$\tilde{\mathbf{Q}}_v^s := \delta \tilde{\mathbf{Q}}_v + (1 - \delta) \mathbf{Q}_v^s, \quad (3)$$

where  $\delta \in [0, 1]$  is a smoothing parameter that blends the contributions of the two quadrics. This parameter influences the surface smoothness as shown in Figure 3. With the new error quadric, our Loop surface simplification is carried out as follows:

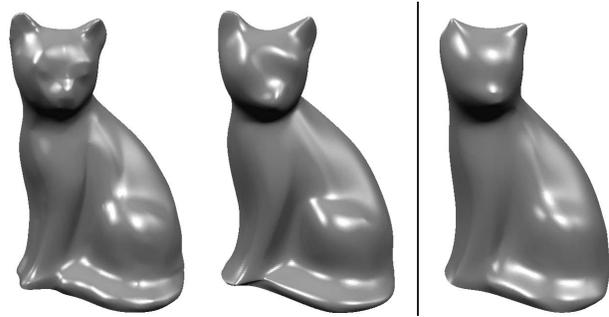
**Initialization:** Quadrics are computed as approximations of the limit surface, by subdividing and projecting the control mesh to its limit position.

**Priority evaluation:** For every edge in the mesh, the error that results from collapsing this edge into a vertex  $v$  is calculated according to  $\tilde{\mathbf{Q}}_v^s$ , and the priority queue is built.

**Update:** Due to the support of the Loop subdivision scheme, we update the priority for all edges in the two-neighborhood of a vertex after the collapse of an edge on top of the queue (cf. Figure 2).



**Figure 3. Simplified subdivision surfaces using different smoothing parameters (from left to right: 1.0, 0.5, 0.1). The rightmost image shows the result of the original QEM method.**



**Figure 4. Comparison of simplified subdivision surfaces with different simplification schemes.**

## 4 Results and Conclusion

Figure 3 compares the results obtained with our algorithm using different blending parameters to simplification of the control mesh with the original QEM algorithm. All control meshes were simplified from 134345 vertices down to 2000 vertices. Surface features are well preserved on the leftmost image even on a smooth subdivision surface by our error metric. The middle left and middle right images show a smoother result while some features are still preserved better in comparison to the rightmost image, which was created using the original QEM algorithm. Figure 4 shows the difference between standard simplification of a control mesh versus using our metric. The left image shows a subdivision surface created from a complex control mesh with 5600 vertices. The surface resulting from simplification to 150 vertices using our method is displayed in the center image. The right image shows the result from simplification to the same number of vertices using the original QEM method. Clearly, our method preserves the details of the limit surface much better at a given target vertex count.

We have presented a quadric error metric for simplifying Loop surfaces, in combination with a parameter that allows the user to control surface smoothness by a scalar parameter. For future work, we plan to make use of the smoothness parameter for generation of domain surfaces for displaced subdivision surfaces, and integrate other functionals into the error metric, i.e. curvature, surface normals, etc.

## References

- [1] H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *SIGGRAPH 2000 Conference Proceedings*, 2000.
- [2] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.
- [3] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno. Multiresolution decimation based on global error. *The Visual Computer*, 13(5):228–246, 1997.
- [4] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers & Graphics*, 22(1):37–54, 1998.
- [5] D. Doo and M. Sabin. Behaviour of recursive subdivision surfaces near extraordinary points. *Computer Aided Design*, 10:356–360, 1978.
- [6] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, 1990.
- [7] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, pages 209–216, 1997.
- [8] A. Guéziec. Surface simplification with variable tolerance. In *Second Annual Intl. Symp. on Medical Robotics and Computer Assisted Surgery*, pages 132–139, 1995.
- [9] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH 94 Conference Proceedings*, pages 295–302, 1994.
- [10] W.-K. Jeong, K. Kähler, J. Haber, and H.-P. Seidel. Automatic generation of subdivision surface head models from point cloud data. In *Graphics Interface 2002 Conf. Proc.*, pages 181–188, 2002.
- [11] T. Kanai. MeshToSS: Converting Subdivision Surfaces from Dense Meshes. *Proc. 6th International Workshop on Vision, Modeling and Visualization 2001*, pages 325–332, 2001.
- [12] L. Kobbelt. A variational approach to subdivision. *Computer Aided Geometric Design*, 13(8):743–761, 1996.
- [13] L. Kobbelt, S. Campagna, and H.-P. Seidel. A general framework for mesh decimation. In *Graphics Interface*, pages 43–50, 1998.
- [14] C. T. Loop. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.
- [15] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, volume 26, pages 65–70, 1992.
- [16] S. Takeuchi, T. Kanai, H. Suzuki, K. Shimada, and F. Kimura. Subdivision Surface Fitting Using QEM-based Simplification and Reconstruction of Approximated B-spline Surfaces. In *8th Pacific Conference on Computer Graphics and Applications*, pages 202–212. IEEE, October 2000.