Albert-Ludwigs-University Freiburg
Faculty of Engineering
Chair of Computer Graphics

WS 2018/19

**Seminar paper**

# Surface Simplification Using Quadric Error Metrics

Felix Baumann

December 28, 2018

## Abstract

Nowadays, computer graphics applications often require highly detailed models to provide a convincing amount of realism. However, rendering these models needs a lot of computational power and storing them a significant amount of storage space.

Since this may prevent real time applications from running smoothly, one may want to decrease the complexity of models in situations where the high level of detail isn't really necessary. For example, using simpler models for objects that are far away from the observer may speed up rendering significantly.

Unfortunately, the creation of simplified models, which still preserve the original shape well, isn't trivial. Therefore, *surface simplification* has received a lot of attention.

Michael Garland and Paul S. Heckbert propose a surface simplification algorithm [3] that is claimed to simplify given models fast, while still preserving the shape of the models well. It is able though to change the models topology by iteratively contracting pairs of vertices. It uses a so called *Quadric Error Metric* to choose pairs for contraction that keep the damage to the original model low.

# Contents

# 1 Surface Simplification

While very detailed polygonal models may offer a high level of realism in computer graphics applications, they also require a lot of computational effort when being rendered. Especially in real time applications it is reasonable to try to reduce this effort, even if this comes with some optical quality loss.

A possible approach is using simpler models for objects with low importance in the current situation. For instance, it's likely that an object far away from the viewpoint may be rendered in a much lower resolution than an object close to the viewpoint, without harming the result a lot.

Therefore it is useful to store multiple versions of a model, each having a different level of detail, and always using the one that seems appropriate for the current situation. These versions can be created using *surface simplification* which is the process of altering the mesh of a model in a way that reduces its complexity. Obviously, these simplified models are supposed to preserve the shape of the original model well to avoid an unnecessary quality loss. On the other hand, they have to be computable within a reasonable amount of time. Otherwise, the creation would ruin the saved rendering effort. So some compromise has to be made between the quality of the simplification and the effort of creating it.

Figure 1 shows the so called *Stanford Bunny*, a model with thousands of faces and therefore a high level of detail. Surface simplification is supposed to create less complex models as shown in Figure 2. When watching the two versions of the model from far away as in Figure 3, the optical difference isn't that significant anymore.

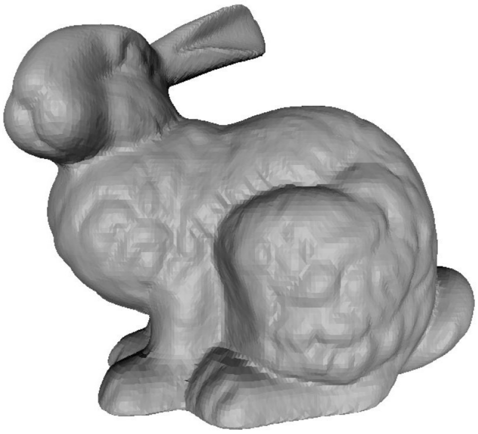

Figure 1: The *Stanford Bunny*, a detailed model with 69,000 faces. [4]

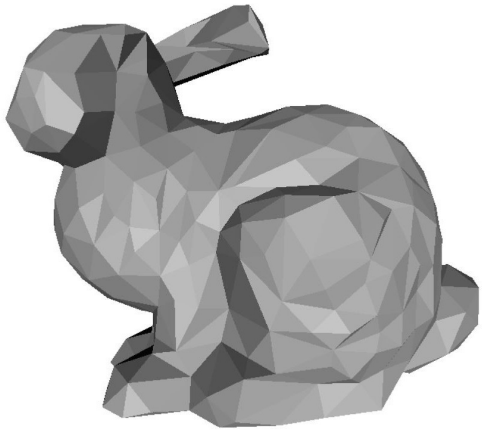

Figure 2: A simplified *Stanford Bunny* with only 1,000 faces. [4]

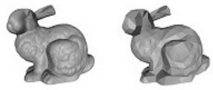

Figure 3: A detailed bunny and one simplified by 98%. From far away the difference isn't that obvious. [4]

## 2 Pair Contraction

Garland and Heckbert propose a surface simplification technique called *pair contraction* [3]. It iteratively contracts pairs of vertices of a given polygonal model reducing the total number of vertices and therefore the model's complexity. They consider it to be a generalization of the approach of contracting edges, since pair contraction isn't limited to pairs of vertices that share a common edge. Instead, pair contraction may be applied to any pair of vertices, which allows it to change the models topology. In many cases, this may not be desirable, yet there are also scenarios where the ability to join unconnected regions is valuable. For instance, flaws in the model like duplicate vertices at the same position can be removed. Furthermore, it allows a higher level of simplification. Nonetheless, caution is required, since it could also ruin a model quickly.

They describe the technique as follows: once a pair of vertices $(\mathbf{v_1}, \mathbf{v_2})$ is chosen, those vertices are both moved to a new position $\mathbf{v'}$ called the *contraction target*. All incident edges of the second vertex are reassigned to the first vertex. After that, the second vertex as well as any degenerate edges and faces of the model are removed.

Figure 4 shows a single pair contraction, including the movement of chosen vertices as well as the deletion of the second vertex and a degenerate edge.

Obviously, an effective simplification process takes more than one pair contraction. Figure 5 shows several steps of pair contraction with a fine outcome. Note how close the shape of the resulting model is to the original one.

Unfortunately, pair contraction doesn't necessarily produce such a satisfying outcome if left alone. Both, an arbitrary choice of a vertex pair and a bad choice for the contraction target may alter the shape of the model unfavorably as shown in figure 6.

Therefore a method is required that chooses pairs of vertices and contraction targets which preserve the shape of the model well. Garland and Heckbert propose the usage of their so called *Quadric Error Metric*.

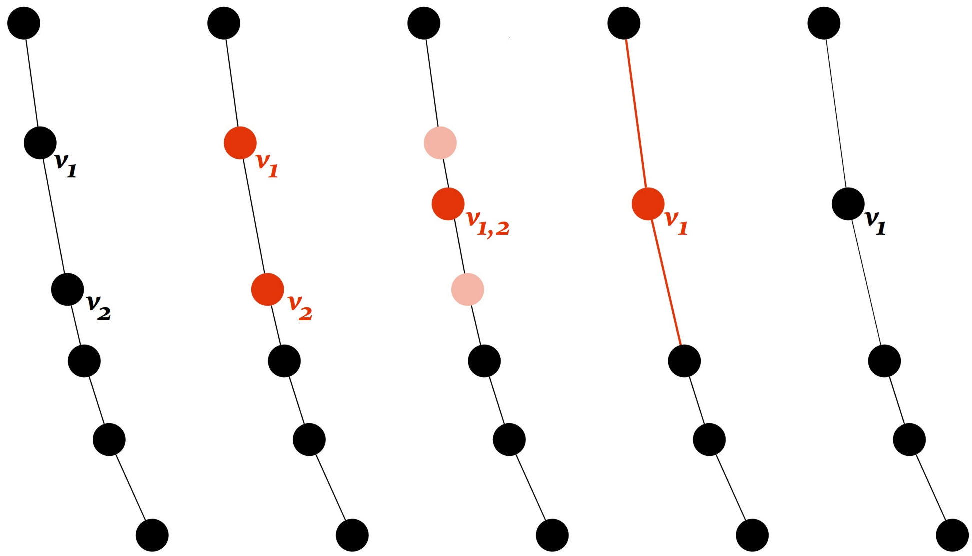


Figure 4: The steps of a single pair contraction.
First two vertices are chosen, second they're moved to a new position and third the second vertex as well as the edge between those are removed.

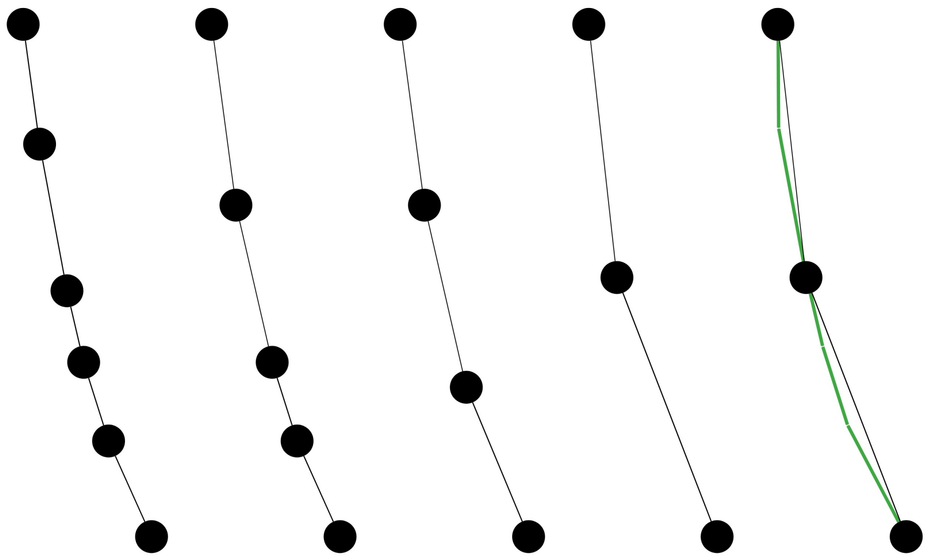

Figure 5: Multiple steps of pair contraction yielding a simplification of 50%.
The green line indicates the original shape.

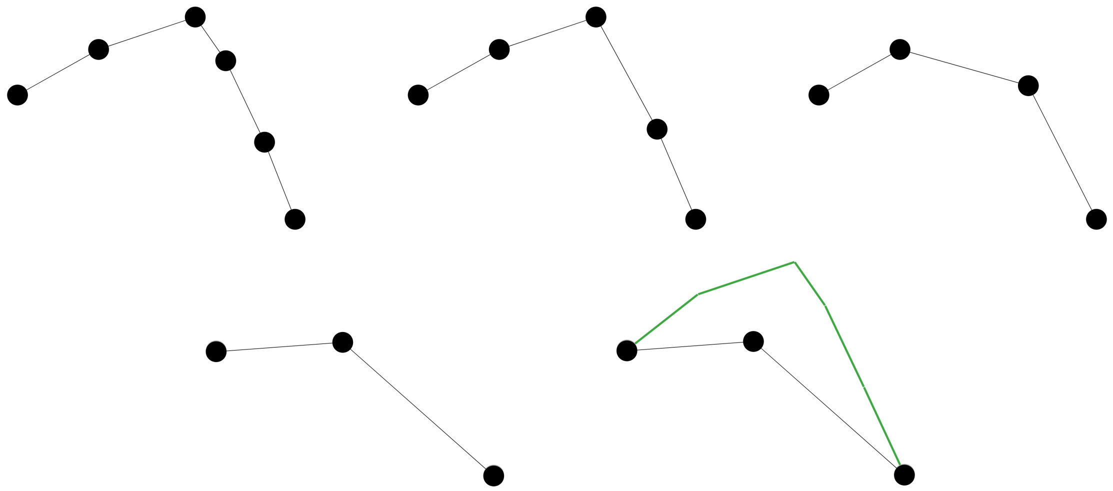

Figure 6: Multiple steps of pair contraction yielding a simplification of 50%. Again, the green line indicates the original shape. Due to a bad choice of contraction targets the shape of the model changed significantly.

# 3 Quadric Error Metric

## 3.1 Plane Intersection

In order to choose a vertex pair suitable for contraction, some metric is needed that measures the amount of damage dealt to the model by a potential contraction. Obviously, a contraction with a low damage value (*error*) is desirable.

In search of an indicator for this error, one may want to take a closer look at the pair contraction shown in figure 7. It's an example where the outcome of the contraction is really close to the shape in the beginning. The pair of contracted vertices is colored as well as all their incident edges. After extending those edges, one may observe that they have only small distances to the contraction target.

Surface simplification is usually applied to three dimensional models. Since this is only a two dimensional example, it can be considered as a cut through a three dimensional model. In this case, the edges represent entire *faces* of the model and the extended edges represent the planes of these faces.

Moreover, note that a vertex is located at the intersection of the planes of its adjacent faces [3]. Using this, it's reasonable to approximate the damage dealt to the shape of the model by a contraction by the distances from the contraction target to the planes that intersect at one of the contracted vertices.

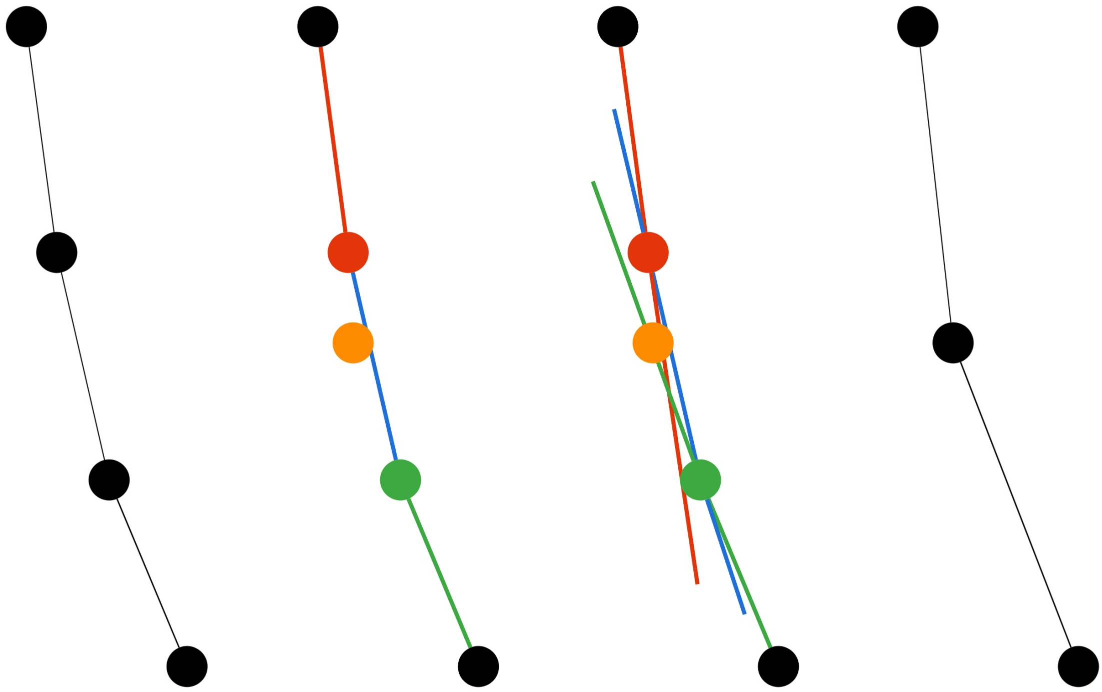

Figure 7: A pair contraction with a fine result.
The vertices marked in red and green are moved to the orange position. Extending the edges incident on those vertices shows that the contraction target (orange) is close to all of those extended edges.

While those distances are small at the pair contraction in figure 7 which yielded a fine result, the distances at another example in figure 8 are larger and the result is much worse.

## 3.2 Squared Distances

The *Quadric Error Metric* developed by Garland and Heckbert estimates the damage caused by a pair contraction. It uses the size of the squared distances from the contraction target to the planes that intersect at the contracted vertices [3]. The contraction target is supposed to be close to all of the planes. This motivates the use of squared distances instead of normal ones, since it penalizes a large distance to one plane more than moderate distances to all of the planes.

Figure 9 shows the distances considered by the Quadric Error Metric.

For the calculation of these squared distances, the position of the contraction target is required which is given as a vector $\mathbf{v}' = (x, y, z, 1)^T$ of coordinates in three dimensional space with the fourth argument being the homogeneous component.

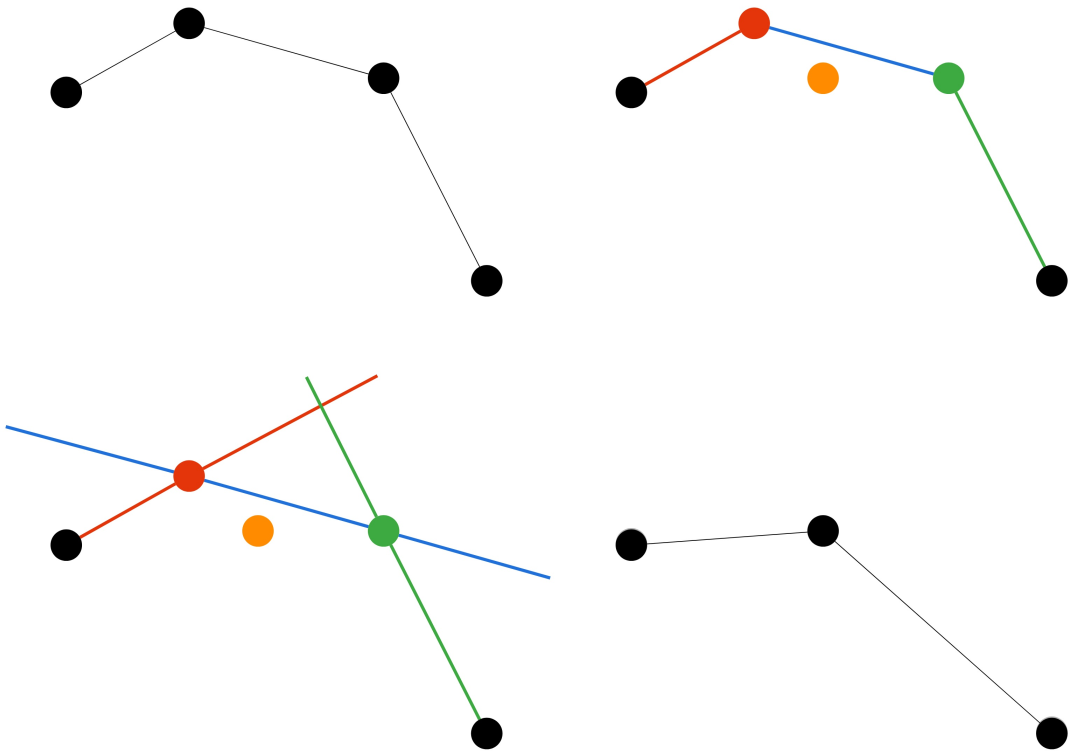

Figure 8: A pair contraction with an unsatisfying result.
The vertices marked in red and green are moved to the orange position. Extending the edges incident on those vertices shows that the contraction target (orange) is relatively far away from two of those extended edges.

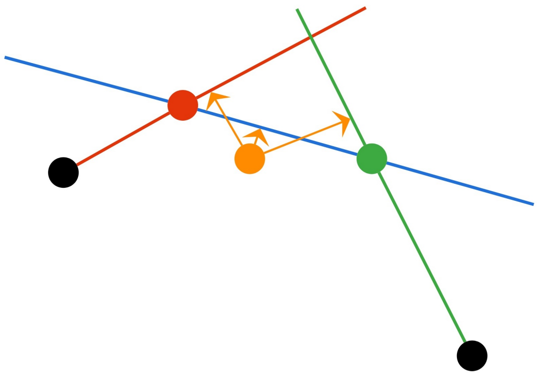

Figure 9: The Quadric Error Metric evaluates a contraction by the size of the distances (orange arrows) from the contraction target (orange dot) to the planes intersecting at one of the contracted vertices (colored lines).

Meanwhile, a plane $p$ can be expressed by a formula

$$ax + by + cz + d = 0, \tag{1}$$

where any point given by $x$, $y$ and $z$ that solves this equation is part of the plane. A simple normalization of the equation can make sure that the following holds:

$$a^2 + b^2 + c^2 = 1. \tag{2}$$

The distance between the *contraction target* $\mathbf{v}'$ and the plane $p$ can be calculated by the formula:

$$\frac{|ax + by + cz + d|}{\sqrt{a^2 + b^2 + c2}} \tag{3}$$

as Cheney et al. point out [1].

Therefore the squared distance between the contraction target $\mathbf{v}'$ and the plane $p$ is computed as:

$$\frac{(ax + by + cz + d)^2}{a^2 + b^2 + c2}. \tag{4}$$

Due to the normalization above (2) this yields the simpler formula

$$(ax + by + cz + d)^2, \tag{5}$$

which can be rewritten as a vector-matrix-vector-product as follows:

$$(ax + by + cz + d)^2 = \left( (a, b, c, d) \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \right)^2 \tag{6}$$

$$= (x, y, z, 1) \cdot \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{7}$$

$$= \mathbf{v}'^T \cdot \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix} \cdot \mathbf{v}' \tag{8}$$

Since the Quadric Error Metric doesn't only consider the distance from the contraction target to one plane, but to several ones as shown in figure 9, the product above (8) would have to be calculated for every plane.

Instead, the squared distances are computed all at a time by summing up the matrices of all those planes. That says the *error* $\Delta$ of a vertex $\mathbf{v}'$ is defined as:

$$\Delta(\mathbf{v}') = \mathbf{v}'^T \cdot Q \cdot \mathbf{v}' \quad [3] \tag{9}$$

with $Q$ being the sum of the matrices of all considered planes $p$, i.e.

$$Q = \sum_p \begin{pmatrix} a_p^2 & a_p b_p & a_p c_p & a_p d_p \\ a_p b_p & b_p^2 & b_p c_p & b_p d_p \\ a_p c_p & b_p c_p & c_p^2 & c_p d_p \\ a_p d_p & b_p d_p & c_p d_p & d_p^2 \end{pmatrix} \tag{10}$$

and each plane $p$ given as:

$$a_p x + b_p y + c_p z + d_p = 0. \tag{11}$$

# 4 Contraction Target

The quality of a contraction depends a lot on a fine choice of the *contraction target*. For instance, the result of the contraction in figure 8 could be improved by choosing a contraction target inside the triangle limited by the extended edges. This motivates computing the optimal target for a given contraction instead of choosing an arbitrary one.

Garland considers the target that minimizes the error as optimal [2].

Since the *error* is given by the formula

$$\Delta(\mathbf{v}') = \mathbf{v}'^T \cdot Q \cdot \mathbf{v}', \tag{12}$$

the optimal target can be identified as a position $\mathbf{v}'$ that minimizes this function.

Rewriting this yields:

$$min\left(\Delta(\mathbf{v}')\right) = min\left(\mathbf{v}'^T \cdot Q \cdot \mathbf{v}'\right) \tag{13}$$

$$= min\left( (x, y, z, 1) \cdot \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \right) \tag{14}$$

$$= min \Big( q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 + 2q_{23}yz + 2q_{24}y + q_{33}z^2 + 2q_{34}z + q_{44} \Big) \tag{15}$$

At a minimum all the partial derivatives have to be zero. Therefore the contraction target has to fulfill the following conditions:

$$\frac{\partial \Delta}{\partial x} = q_{11}x + q_{12}y + q_{13}z + q_{14} = 0 \tag{16}$$

$$\frac{\partial \Delta}{\partial y} = q_{21}x + q_{22}y + q_{23}z + q_{24} = 0 \tag{17}$$

$$\frac{\partial \Delta}{\partial z} = q_{31}x + q_{32}y + q_{33}z + q_{34} = 0 \tag{18}$$

Those can be combined into one single equation in matrix-vector form:

$$\begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{19}$$

Solving for the contraction target finally yields:

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{20}$$

In order to calculate the optimal contraction target, it's sufficient to replace the last row of the *Q-matrix* of this contraction by $(0, 0, 0, 1)$ and invert the resulting matrix. The optimal target will then be given as the matrix' last column.

The $Q$-matrix shall consist of the squared distances to all the planes that intersect in one of the two contracted vertices. Therefore it's reasonable to assign a matrix to every vertex of the model and in case of a contraction to simply sum up the matrices of the two contracted vertices.

If the vertices share a face, unfortunately, its plane will be considered twice. Yet Garland argues that this mistake is relatively small and acceptable since this method of implicitely tracking the planes using the matrices is way more efficient than tracking them explicitely [3].

# 5 Algorithm

Garland and Heckbert propose an algorithm for surface simplification that performs pair contractions iteratively. It uses the Quadric Error Metric to choose a suitable pair of vertices for contraction and for computing the optimal contraction target [3]. The following chapters explain their algorithm in detail.

## 5.1 Initial Matrices

First of all, an initial *Q-matrix* for every single vertex in the whole given model has to be computed. This requires the planes of the faces adjacent to the current vertex. While each plane $p$ is given by an equation $ax + by + cz + d = 0$, the squared-distance-matrix for this plane has the form

$$\begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix}. \tag{21}$$

Summing up these matrices for all planes of the faces adjacent to the current vertex $v$ yields its initial $Q$-matrix.

$$Q(v) = \sum_{p \in v} \begin{pmatrix} a_p^2 & a_p b_p & a_p c_p & a_p d_p \\ a_p b_p & b_p^2 & b_p c_p & b_p d_p \\ a_p c_p & b_p c_p & c_p^2 & c_p d_p \\ a_p d_p & b_p d_p & c_p d_p & d_p^2 \end{pmatrix} \tag{22}$$

Calculating the error of an initial vertex using its $Q$-matrix will naturally result in an error value of 0. This is reasonable since the vertex hasn't been moved yet and therefore didn't alter the shape of the model.

## 5.2 Pair Selection

In order to perform pair contractions, a set of pairs of vertices has to be found that are considered candidates for contraction.

Technically, any pair of vertices of the model would be a candidate, but contracting vertices far away from each other will usually result in large errors. Instead, the algorithm only considers pairs of vertices that either:

- share a common edge or
- are closer to each other than a given threshold.

For all vertex pairs $(\mathbf{v_1}, \mathbf{v_2})$ that fulfill at least one of these conditions, a new $Q$-matrix $Q'$ has to be calculated as the sum of the Q-matrices of the vertices of the pair.

$$Q' = Q_1 + Q_2 \tag{23}$$

In case the contraction of the pair $(\mathbf{v_1}, \mathbf{v_2})$ will actually be executed, this $Q$-matrix $Q'$ will be assigned to the contracted vertex $\mathbf{v}'$.

## 5.3 Contraction Targets

At this point, all candidates for pair contraction as well as their new $Q$-matrices are known. Using this, the *contraction targets* can be calculated as

$$\mathbf{v}' = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{24}$$

## 5.4 Errors

With the knowledge about all considered candidate vertex pairs, their new $Q$-matrices and their contraction targets, the *errors* caused by each contraction can be computed.

$$\Delta(\mathbf{v}') = \mathbf{v}'^T \cdot Q \cdot \mathbf{v}', \tag{25}$$

### 5.5 Pair Contractions

Now, the set of considered pair contractions as well as the errors those would cause are known. Creating a *heap* of the pair contractions with respect to their errors allows to efficiently access the contraction that currently yields the *smallest error*.

With this preparation, the algorithm now performs iterative pair contractions by always choosing the one with the smallest error using the heap.

Since a pair contraction $(\mathbf{v_1}, \mathbf{v_2}) \to \mathbf{v}'$ moves two vertices, all the remaining candidate pairs that include either $\mathbf{v_1}$ or $\mathbf{v_2}$ have to be updated using the contraction target $\mathbf{v}'$ instead. Also, this changes the $Q$-matrix, the contraction target and the error of those candidates which therefore have to be recalculated.

Because vertices usually have a small number of adjacent faces, this doesn't take that much effort though.

### 5.6 Stopping Condition

When performing pair contractions iteratively, the final result will be a set of vertices without edges or faces and a too large distance between each possible pair of vertices to contract them. Since this doesn't seem to be useful, one may want to stop the contraction process before the model is ruined entirely.

This motivates the use of a condition that breaks the contraction loop. For instance, a desired percentage of simplification may be given, a maximum number of vertices or faces, or a maximum acceptable error with respect to the shape of the original model.

## 6 Conclusion

Michael Garland and Paul S. Heckbert propose a *surface simplification algorithm* that uses their *Quadric Error Metric* to choose vertices for iterative *pair contraction* [3].

Garland showed that the algorithm is both fast and preserves the original shape of the model really well, even at a large rate of simplification [2].

A rather unusual approach is that the algorithm can change the topology of the model which Garland and Heckbert consider to be a benefit [3]. The threshold parameter defining what's close enough to contract, has to be chosen with some care though, since changing the topology heedlessly may easily ruin the entire model.

## References


[1] Ward Cheney and David Kincaid: *Linear Algebra: Theory and Applications.* Jones and Bartlett, 2010. Pages: 450, 451.

[2] Michael Garland: *Quadric-based polygonal surface simplification.* May 9, 1999. Ph.D. thesis, Carnegie Mellon University, CS Dept. Tech. Rept. CMU-CS-99-105, page 51.

[3] Michael Garland and Paul S. Heckbert: *Surface simplification using quadric error metrics.* In *Proceedings of SIGGRAPH 97. ACM SIGGRAPH*, pages 209–216. 1997.

[4] Greg Turk and Marc Levoy: *Bunny model*, 1994. Stanford University Computer Graphics Laboratory.