III.12 Edge Contraction

A triangulated surface is simplified by reducing the number of vertices. This section presents an algorithm that simplifies by repeated edge contraction. We discuss the operation, describe the algorithm, and introduce the error measure that controls which edges are contracted and in what sequence.

Edge contraction. Let K be a 2-complex and assume for the moment that $|K|$ is a 2-manifold. The contraction of an edge $ab \in K$ removes ab together with the two triangles abx, aby and it mends the hole by gluing xa to xb and ya to yb, as shown in Figure III.1. Vertices a and b are glued to form a new vertex c. All simplices in the star of c are new, and the rest of the complex stays the same. To express this more formally, we define the *cone* from a point x to a simplex τ as the union of line segments connecting x to points $p \in \tau$, $x \cdot \tau = \mathrm{conv}\,(\{x\} \cup \tau)$. It is defined only if x is not an affine combination of the vertices of τ. With this restriction, $x \cdot \tau$ is a simplex of one higher dimension, $\dim(x \cdot \tau) = 1 + \dim \tau$. For a *set* of simplices, the cone is defined if it is defined for each simplex, and in this case $x \cdot T = \{x \cdot \tau \mid \tau \in T\}$. We also need generalizations of the star and the link from a single simplex to a set of simplices. Denote the closure without the (-1)-simplex as $\overline{T} = \mathrm{Cl}\,T - \{\emptyset\}$. The *star* and *link* of T are

$$\begin{aligned} \mathrm{St}\,T &= \{\sigma \in K \mid \sigma \geq \tau \in T\}, \\ \mathrm{Lk}\,T &= \mathrm{Cl}\,\mathrm{St}\,T - \mathrm{St}\,\overline{T}. \end{aligned}$$

For closed sets T, the link is simply the boundary of the closed star. For example, in Figure III.1 the link of the set $\overline{ab} = \{ab, a, b\}$ is the cycle of dashed edges and hollow vertices bounding the closed star of $\overline{ab}$. The *contraction* of the edge ab is the operation that changes K to $L = K - \mathrm{St}\,\overline{ab} \cup c \cdot \mathrm{Lk}\,\overline{ab}$. This definition applies generally and does not assume that K is a manifold.

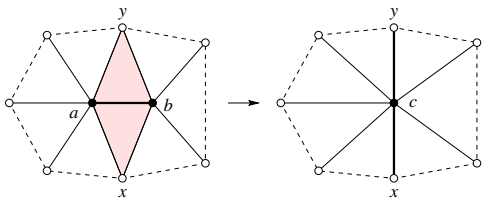

Figure III.1: The contraction of edge ab. Vertices a and b are glued to a new vertex c.

Algorithm. The surface represented by K is simplified by performing a sequence of edge contractions. To get a meaningful result, we prioritize the contractions by the numerical error they introduce. Contractions that change the topological type of the surface are rejected. Initially, all edges are evaluated and stored in a priority queue. The process continues until the number of vertices shrinks to the target number m. Let $n \geq m$ be the number of vertices in K.

```
while n > m and priority queue non-empty do
  remove top edge ab from priority queue;
  if contracting ab preserves topology then
    contract ab;  n--
  endif
endwhile.
```

The priority queue takes time $O(\log n)$ per operation. Besides extracting the edge whose contraction causes the minimum error, we remove edges that no longer belong to the surface and we add new edges. The number of edges removed and added during a single contraction is usually bounded by a small constant, but in the worst case it can be as large as $n - 1$. Before performing an edge contraction, we test whether or not it preserves the topological type of the surface. This is done by checking all edges and vertices in the link of $\overline{ab}$. Precise conditions to recognize edge contractions that preserve the type will be discussed in Section ??.

Hierarchy. We visualize the actions of the algorithm by drawing the vertices as the nodes of an upside-down forest. The contraction of the edge ab maps vertices a and b to a new vertex c. In the forest this is reflected by introducing c as a new node and declaring it the parent of a and b. The leaves of the forest are the vertices of K, and the roots are the vertices of the simplified complex L. The forest is illustrated in Figure III.2. We define a function $g : \mathrm{Vert}\,K \to \mathrm{Vert}\,L$ that maps each vertex $a \in K$ to the root $u = g(a)$ of the tree in which a is a leaf. The preimage of $u \in L$ is the set of leaves $g^{-1}(u) \subseteq K$ of the tree with root u. The preimages of the roots partition the set of leaves,

$$\mathrm{Vert}\,K = \bigcup_{u \in L} g^{-1}(u).$$

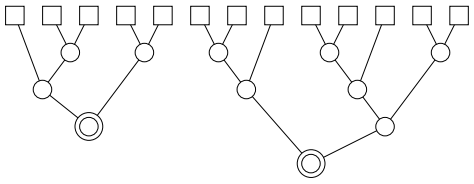

Figure III.2: Vertices of K are shown as square nodes, intermediate vertices as circle nodes, and vertices of the final complex L as double circle nodes.

Let ab be an edge in K and set $u = g(a)$, $v = g(b)$. If $u \neq v$ then ab still exists in L, or rather it corresponds to an edge, namely to $uv \in L$. Else, ab contracts to vertex $u = v \in L$. Similarly, a triangle $abd \in K$ corresponds to $uvw \in L$ if $u = g(a)$, $v = g(b)$, $w = g(d)$ are pairwise different. If two of u, v, w are the same then abd contracts to an edge, and if $u = v = w$ then abd contracts to this vertex in L.

Numerical error. As mentioned above, a vertex $u \in \operatorname{Vert} L$ represents a subset $g^{-1}(u) \subseteq \operatorname{Vert} K$. It makes sense to measure the numerical error at u by comparing u to the part of the original surface it represents. Specifically, we define the error as the sum of square distances of u from the planes spanned by triangles in the star of $g^{-1}(u)$. Figure III.3 illustrates this idea by showing a vertex $u \in L$ and the triangles in the star of $g^{-1}(u)$. The preimage of u is the collection of seven solid vertices in the right half of the figure. The star of the preimage contains the five shaded triangles and the ring of white triangles around them. The shaded triangles have all their vertices in $g^{-1}(u)$ and the white triangles have either one or two vertices in the preimage.

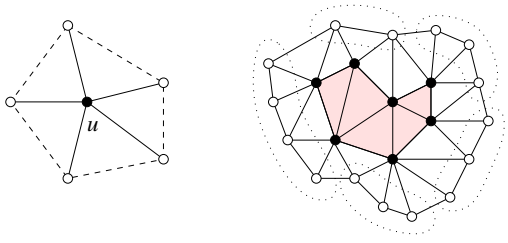

Figure III.3: Vertex u and its star to the left and the corresponding piece of K to the right. The solid vertices on the right are preimages of u and the hollow vertices are preimages of the neighbors of u.

Let H_u be the set of planes spanned by triangles in $\operatorname{St} g^{-1}(u)$. The sum of square distances is defined for every point in $\mathbb{R}^3$, so we can think of the error measure as a function $E_u : \mathbb{R}^3 \to \mathbb{R}$. This function has a unique minimum, unless the normals to the planes in H_u span less than $\mathbb{R}^3$. We can therefore choose u at the point in space where E_u attains its minimum. If the linear subspace spanned by the normals is 2-dimensional then there is a line of minima, and if it is 1-dimensional then there is a plane of minima. In both cases we add contraints to pin down u.

Inclusion-exclusion. We will see in Section **??** that E_u can be represented by a single symmetric 4-by-4 matrix $\mathbf{Q}_u$, no matter how many planes there are in H_u. Define $H_w = H_u \cup H_v$. We have $E_w = E_u + E_v - E_{uv}$, where $E_{uv} : \mathbb{R}^3 \to \mathbb{R}$ maps a point in $\mathbb{R}^3$ to the sum of square distances from planes in $H_{uv} = H_u \cap H_v$. We also have

$$\mathbf{Q}_w \quad = \quad \mathbf{Q}_u + \mathbf{Q}_v - \mathbf{Q}_{uv}.$$

It is generally not possible to construct $\mathbf{Q}_{uv}$ directly from $\mathbf{Q}_u$ and $\mathbf{Q}_v$. Constructing H_{uv} and computing $\mathbf{Q}_{uv}$ from this set can be expensive. Instead, we maintain matrices $\mathbf{Q}$ for all vertices, edges, triangles such that $\mathbf{Q}_{ab} = \mathbf{Q}_a \cap \mathbf{Q}_b$ and $\mathbf{Q}_{abd} = \mathbf{Q}_a \cap \mathbf{Q}_b \cap \mathbf{Q}_d$ for all edges ab and all triangles abd. We revisit

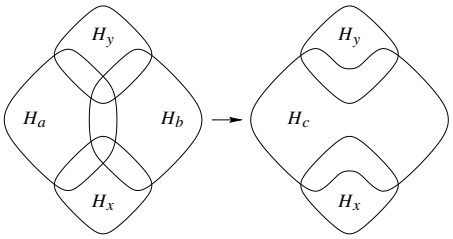

Figure III.4: Effect of edge contraction on sets of planes used in computing the error.

the contraction of the edge ab. The error function of the new vertex c is given by the matrix $\mathbf{Q}_c = \mathbf{Q}_a + \mathbf{Q}_b - \mathbf{Q}_{ab}$. For every vertex $x \in \operatorname{Lk} ab$ there is a new edge xc with error function represented by $\mathbf{Q}_{xc} = \mathbf{Q}_{xa} + \mathbf{Q}_{xb} - \mathbf{Q}_{xab}$, as illustrated in Figure III.4. We will see later that the matrices for edges are not just useful for correctly computing the matrices for vertices, but they represent meaningful geometric information about edges and their relation to the original surface triangulation.

Bibliographic notes. The problem of simplifying a triangulated surface has its origin in computer graphics, where rendering speed depends on the number of triangles used to represent a shape. The idea of using edge contractions for surface simplification appeared first in a paper by Hoppe et al. [5]. Edge contractions are used together with other local surface modification operations in an attempt to optimize a measure of distance between the original and the simplified surface. Hoppe [3] revisits the idea and shows how to use a given sequence of contractions for efficiently adjusting the level of detail of the surface representation. The idea of measuring error as the sum of square distances from gradually accumulating planes is due to Garland and Heckbert [1]. The good quality of the resulting simplifications has intensified the experimental research and lead to variations, such as error measures that account for color and texture of triangulated surfaces [2, 4].

[1] M. Garland and P. S. Heckbert. Surface simplification using quadratic error metrics. *Comput. Graphics*, Proc. siggraph 1997, 209–216.

[2] M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadratic error metrics. *In* "Proc. Visualization, 1998", IEEE Computer Society Press, Los Alamitos, California, 279–286.

[3] H. Hoppe. Progressive meshes. *Comput. Graphics*, Proc. siggraph 1996, 99–108.

[4] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. *In* "Proc. Visualization, 1999", IEEE Computer Society Press, Los Alamitos, California.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stützle. Mesh optimization. *Comput. Graphics*, Proc. siggraph 1993, 19–26.