

16. Subdivision curves and surfaces

Reading

Recommended:

- ♦ Stollnitz, DeRose, and Salesin. *Wavelets for Computer Graphics: Theory and Applications*, 1996, section 6.1-6.3, 10.2, A.5.

Subdivision curves

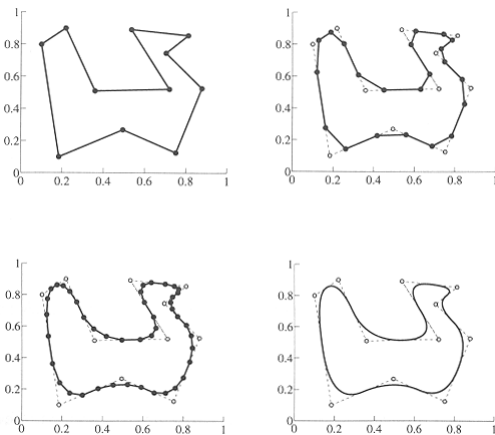
Idea:

- ♦ repeatedly refine the control polygon

$$P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \dots$$

- ♦ curve is the limit of an infinite process

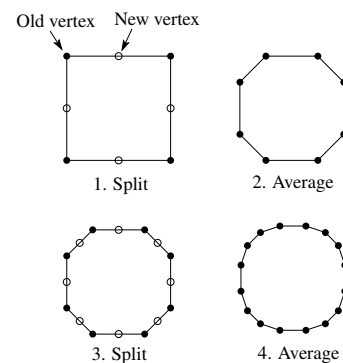
$$C = \lim_{i \rightarrow \infty} P_i$$



Chaikin's algorithm (1974)

Chaikin introduced the following "corner-cutting" scheme in 1974:

- ♦ Start with a piecewise linear curve
- ♦ Insert new vertices at the midpoints (the **splitting step**)
- ♦ Average each vertex with the "next" neighbor (the **averaging step**)
- ♦ Go to the splitting step



Averaging masks

The limit curve is a quadratic B-spline!

Instead of averaging with the nearest neighbor, we can generalize by applying an **averaging mask** during the averaging step:

$$r = (\dots, r_{-1}, r_0, r_1, \dots)$$

In the case of Chaikin's algorithm:

$$r =$$

Lane-Riesenfeld algorithm (1980)

Use averaging masks from Pascal's triangle:

$$r = \frac{1}{2^n} \left(\binom{n}{0}, \binom{n}{1}, \dots, \binom{n}{n} \right)$$

Gives B-splines of degree $n+1$.

$n=0$:

$n=1$:

$n=2$:

Subdivide ad nauseum?

After each split-average step, we are closer to the **limit curve**.

How many steps until we reach the final (limit) position?

Can we push a vertex to its limit position without infinite subdivision? Yes!

Recipe for subdivision curves

After subdividing and averaging a few times, we can push each vertex to its limit position by applying an **evaluation mask**.

Each subdivision scheme has its own evaluation mask, mathematically determined by analyzing the subdivision and averaging rules.

For Lane-Riesenfeld cubic B-spline subdivision, we get:

$$\frac{1}{6} (1 \quad 4 \quad 1)$$

Now we can cook up a simple procedure for creating subdivision curves:

- ◆ Subdivide (split+average) the control polygon a few times. Use the averaging mask.
- ◆ Push the resulting points to the limit positions. Use the evaluation mask.

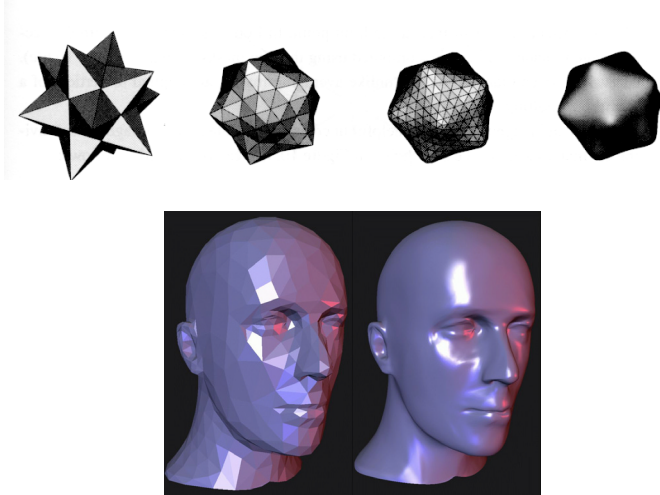
Subdivision surfaces

Chaikin's use of subdivision for curves inspired similar techniques for subdivision surfaces.

Iteratively refine a **control polyhedron** (or **control mesh**) to produce the limit surface

$$\sigma = \lim_{j \rightarrow \infty} M^j$$

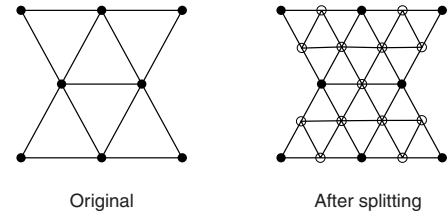
using splitting and averaging steps.



Triangular subdivision

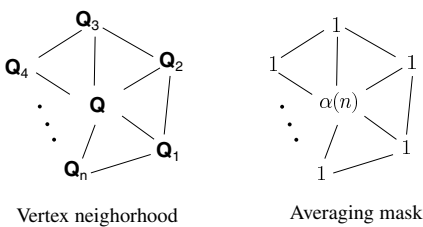
There are a variety of ways to subdivide a polygon mesh.

A common choice for triangle meshes is 4:1 subdivision – each triangular face is split into four subfaces:



Loop averaging step

Once again we can use **masks** for the averaging step:



$$\mathbf{Q} \leftarrow \frac{\alpha(n)\mathbf{Q} + \mathbf{Q}_1 + \dots + \mathbf{Q}_n}{\alpha(n) + n}$$

where

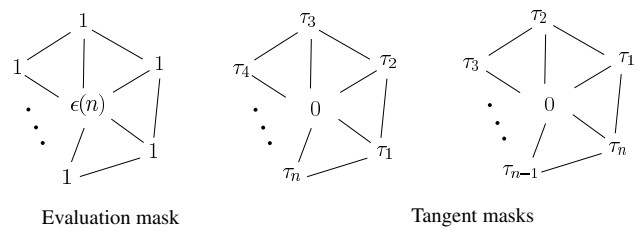
$$\alpha(n) = \frac{n(1-\beta(n))}{\beta(n)} \quad \beta(n) = \frac{5}{4} - \frac{(3+2\cos(2\pi/n))^2}{32}$$

These values, due to Charles Loop, are carefully chosen to ensure smoothness – namely, tangent plane or normal continuity.

Note: tangent plane continuity is also known as G^1 continuity.

Loop evaluation and tangent masks

As with subdivision curves, we can split and average a number of times and then push the points to their limit positions.



$$\mathbf{Q}^\infty = \frac{\epsilon(n)\mathbf{Q} + \mathbf{Q}_1 + \dots + \mathbf{Q}_n}{\epsilon(n) + n}$$

$$\mathbf{T}_1^\infty = \tau_1(n)\mathbf{Q}_1 + \tau_2(n)\mathbf{Q}_2 + \dots + \tau_n(n)\mathbf{Q}_n$$

$$\mathbf{T}_2^\infty = \tau_n(n)\mathbf{Q}_1 + \tau_1(n)\mathbf{Q}_2 + \dots + \tau_{n-1}(n)\mathbf{Q}_n$$

where

$$\epsilon(n) = \frac{3n}{\beta(n)} \quad \tau_i(n) = \cos(2\pi i/n)$$

How do we compute the normal?

Recipe for subdivision surfaces

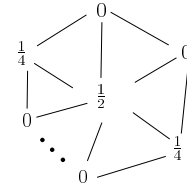
As with subdivision curves, we can now describe a recipe for creating and rendering subdivision surfaces:

- ◆ Subdivide (split+average) the control polyhedron a few times. Use the averaging mask.
- ◆ Compute two tangent vectors using the tangent masks.
- ◆ Compute the normal from the tangent vectors.
- ◆ Push the resulting points to the limit positions. Use the evaluation mask.
- ◆ Render!

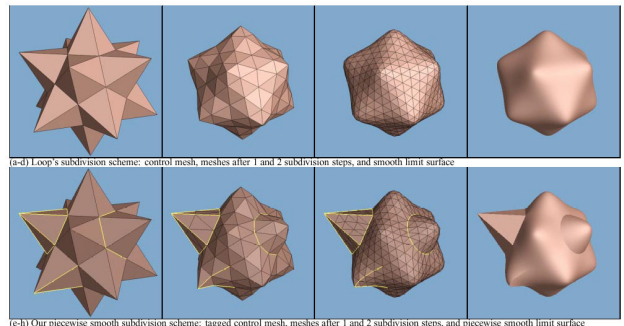
Adding creases without trim curves

In some cases, we want a particular feature such as a crease to be preserved. With NURBS surfaces, this required the use of trim curves.

For subdivision surfaces, we can just modify the subdivision mask:



This gives rise to G^0 continuous surfaces (i.e., having positional but not tangent plane continuity)



Creases without trim curves, cont.

Here's an example using Catmull-Clark surfaces (based on subdividing quadrilateral meshes):



Summary

What to take home:

- ◆ The meanings of all the **boldfaced** terms.
- ◆ How to perform the splitting and averaging steps on subdivision curves.
- ◆ The various kinds of mesh splitting steps for subdivision surfaces, especially Loop.
- ◆ How to construct and render subdivision surfaces from their averaging masks, evaluation masks, and tangent masks.