Subdivision curves and surfaces

Subdivision curves

Idea:
- repeatedly refine the control polygon
  \[ P^1 \rightarrow P^2 \rightarrow P^3 \rightarrow \ldots \]
- curve is the limit of an infinite process
  \[ Q = \lim_{j \to \infty} P^j \]

Chaikin’s algorithm

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” (clockwise) neighbor (the averaging step)
- Go to the splitting step

Reading

Recommended:

Note: there is an error in Stollnitz, et al., section A.5. Equation A.3 should read:

\[ MV = V \Lambda \]
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:

\[ r = (\ldots, r_{-1}, r_0, r_1, \ldots) \]

In the case of Chaikin’s algorithm:

\[ r = \]

---

Lane-Riesenfeld algorithm (1980)

Use averaging masks from Pascal’s triangle:

\[ r = \frac{1}{2^n} \binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n} \]

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.
**DLG interpolating scheme (1987)**

Slight modification to subdivision algorithm:
- splitting step introduces midpoints
- averaging step *only changes midpoints*

For DLG (Dyn-Levin-Gregory), use:

\[ r = \frac{1}{16} (-2, 5, 10, 5, -2) \]

Since we are only changing the midpoints, the points after the averaging step do not move.

---

**Building complex models**

We can extend the idea of subdivision from curves to surfaces…

---

**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

\[ \sigma = \lim_{j \to \infty} M^j \]

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:

\[
Q \leftarrow \frac{\alpha(n)Q + Q_1 + \cdots + Q_n}{\alpha(n) + n}
\]

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 for surfaces.

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)
Catmull-Clark subdivision

4:1 subdivision of triangles is sometimes called a face scheme for subdivision, as each face begets more faces.

An alternative face scheme starts with arbitrary polygon meshes and inserts vertices along edges and at face centroids:

Catmull-Clark subdivision:

[Diagram showing original and after splitting]

Note: after the first subdivision, all polygons are quadiilaterals in this scheme.

Interpolating subdivision surfaces

Interpolating schemes are defined by

- splitting
- averaging only new vertices

The following averaging mask is used in butterfly subdivision:

[Diagram of averaging mask]

Setting \( t=0 \) gives the original polyhedron, and increasing small values of \( t \) makes the surface smoother, until \( t=1/8 \) when the surface is provably \( G^1 \).

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.
- How to perform mesh splitting steps for subdivision surfaces, especially Loop.
- How to construct and render subdivision surfaces from their averaging masks, evaluation masks, and tangent masks.