Subdivision curves and surfaces

Brian Curless
CSE 457
Spring 2014

Subdivision curves

Idea:
• repeatedly refine the control polygon
  $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow \cdots$
• 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) neighboring 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:

$$MV = V \Lambda$$

This is already fixed in the handout.
**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:

\[ r = \left( 0, \frac{1}{2}, \frac{1}{2} \right) \]

Different averaging masks lead to different curves.

For example,

\[ r = \left( \frac{1}{4}, \frac{1}{2}, \frac{1}{4} \right) \]

Leads to cubic B-spline curves.

**Recipe for subdivision curves**

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

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 cubic B-spline subdivision, we get:

\[ r = \left( \frac{1}{6}, \frac{2}{3}, \frac{1}{6} \right) \]

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

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

**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

\[ S = \lim_{j \to \infty} P_j \]

using splitting and averaging steps.

\[ P_0 \quad P_1 \quad P_2 \quad \ldots \quad P_\infty \]

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 smaller triangles:

Original

After splitting

Loop averaging step

Once again we can use masks for the averaging step:

\[ Q = \frac{1}{\alpha(n)} \sum Q_k \]

where

\[ \alpha(n) = \frac{n(n-1) - \beta(n)}{\beta(n)} \quad \beta(n) = \frac{5}{4} \left[ 3 + 2\cos(2\pi / n) \right]^2 \]

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

Note: tangent plane continuity is also know as \( G^1 \) continuity for surfaces.

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.

Evaluation mask (before affine normalization)

Tangent masks

\[ Q^e = \frac{1}{\alpha(n)} Q_0 + \ldots + Q_k \]

\[ T^e = \tau(n) Q_0 + \tau(n) Q_2 + \ldots + \tau(n) Q_k \]

where

\[ \alpha(n) = \frac{3n}{\beta(n)} \quad \tau(n) = \cos(2\pi / n) \]

How do we compute the normal?

\[ \mathbf{N} \approx \mathbf{T} \times \mathbf{T}_n \]
**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**

For NURBS surfaces, adding sharp features like creases required the use of trim curves.

For subdivision surfaces, we can just modify the subdivision masks. E.g., we can mark some edges and vertices as “creases” and modify the subdivision mask for them (and their children):

This gives rise to $G^2$ 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: Original](image)

![Catmull-Clark subdivision: After splitting](image)

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

**Catmull-Clark subdivision (cont’d)**

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

This particular example uses the hybrid technique of DeRose, et al., which applies sharp subdivision rules at some creases for a finite number of steps, and then switches to smooth subdivision, giving more gentle creases. This technique was used in Geri’s Game.
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.