Subdivision curves and surfaces

Brian Curless
CSE 457
Spring 2016

Reading

Recommended:

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

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

$$\mathbf{M}\mathbf{V} = \mathbf{V}\mathrm{L}$$

This is already fixed in the handout.

Subdivision curves

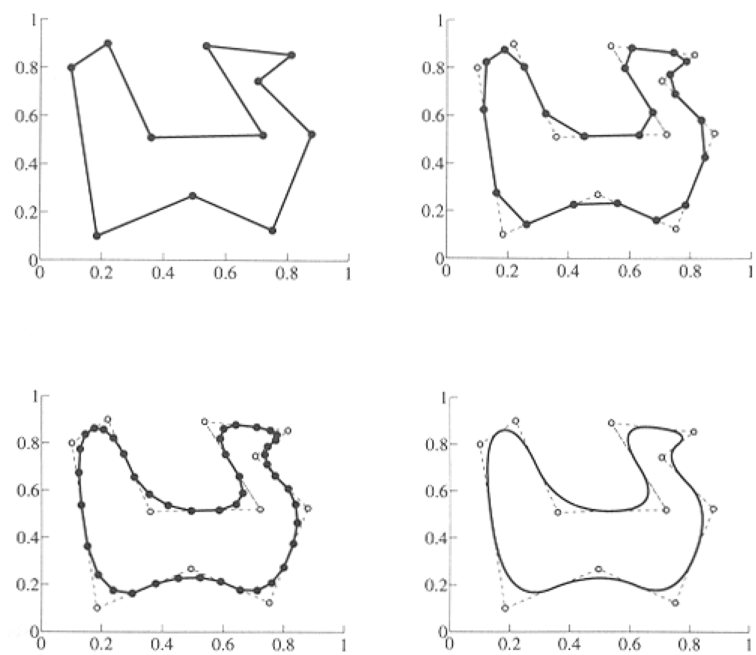
Idea:

- repeatedly refine the control polygon

$$P_1 \to P_2 \to P_3 \to \cdots$$

- curve is the limit of an infinite process

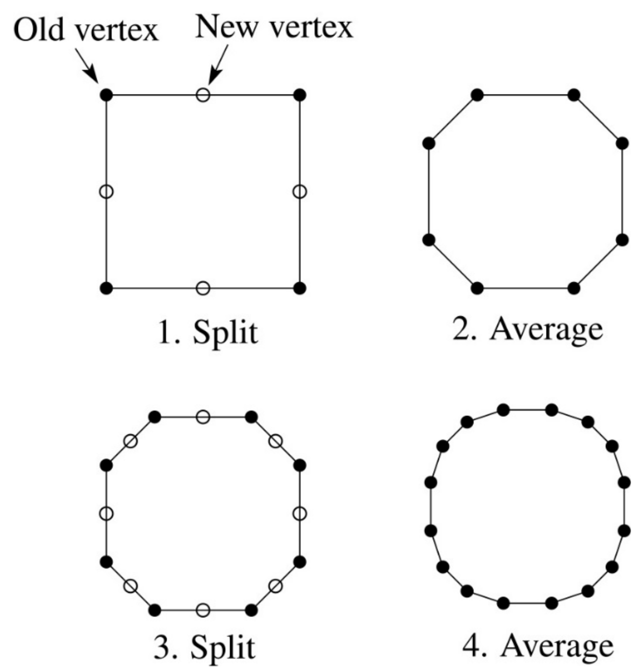
$$Q = \lim_{j \to \infty} P_j$$



Chaikin's algorithm

Chakin 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



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:

$$a = (\ldots, a_{-1}, a_0, a_1, \ldots)$$

In the case of Chaikin's algorithm:

$$a = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Different averaging masks lead to different curves.

For example,

$$a = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

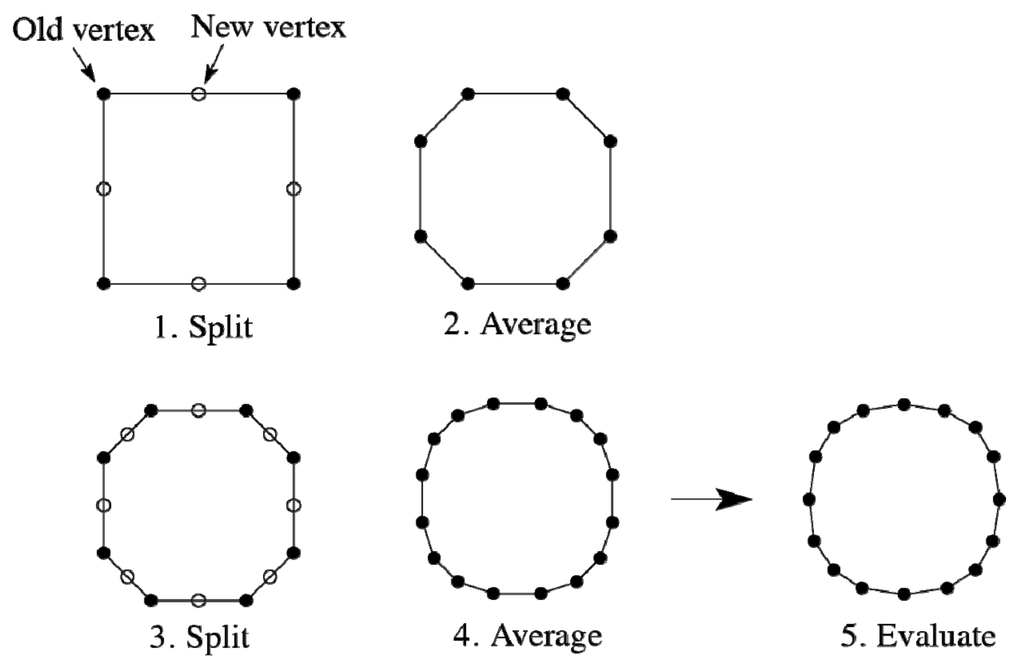
Leads to ***cubic*** B-spline curves.

Limit curves and evaluation masks

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

We can stop after a number of split-average steps and apply an **evaluation mask** to push the vertices onto the limit curve.

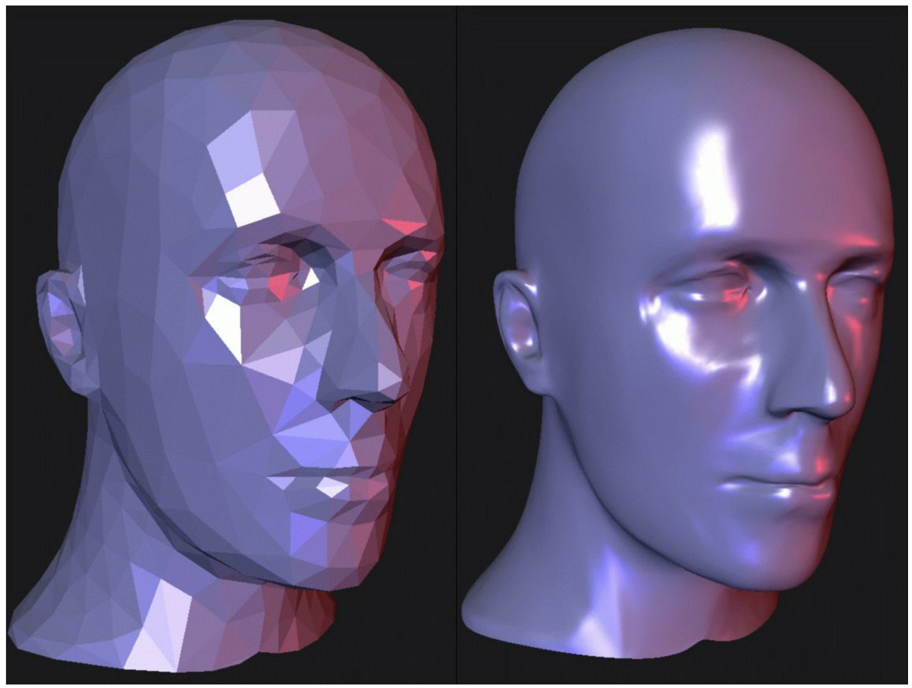
For Chaikin's algorithm, the evaluation masks is: $e = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$



For cubic subdivision, the evaluation masks is: $e = \begin{pmatrix} \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{pmatrix}$

Building complex models

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



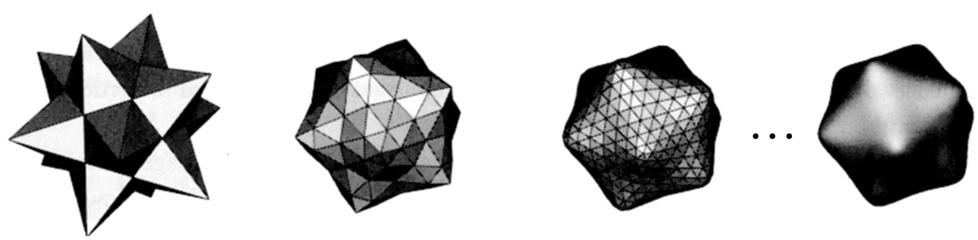
Subdivision surfaces

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.

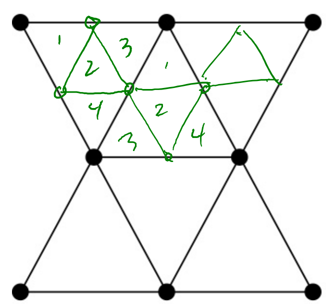


P_0 P_1 P_2 … P_∞

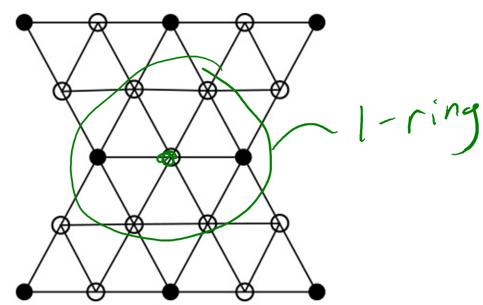
Triangular subdivision

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

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



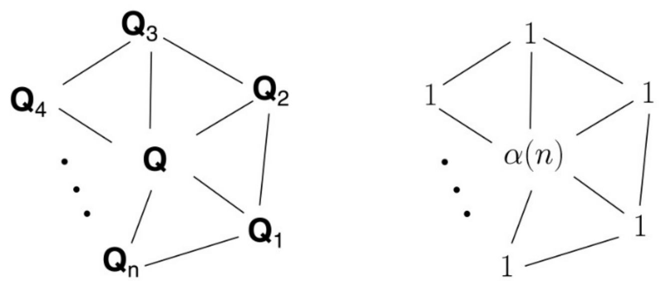
Original



After splitting

Loop averaging step

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



Vertex neighorhood

Averaging mask
(before affine normalization)

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

where

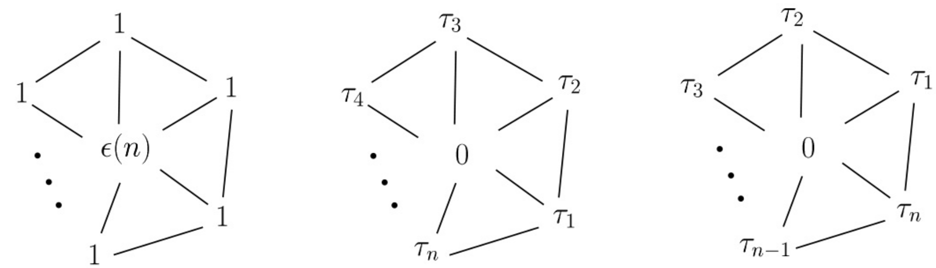
$$\alpha(n) = \frac{n(1-\beta(n))}{\beta(n)} \quad \beta(n) = \frac{5}{4} - \frac{(3+2\cos(2\pi/n))^2}{32}$$

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

Note: tangent plane continuity is also 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

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

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

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

where

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

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:

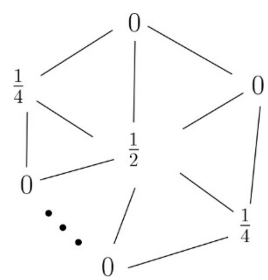
- Subdivide (split+average) the control polyhedron a few times. Use the averaging mask.
- Compute two tangent vectors using the tangent masks.
- Compute the normal from the tangent vectors.
- Push the resulting points to the limit positions. Use the evaluation mask.
- Render!

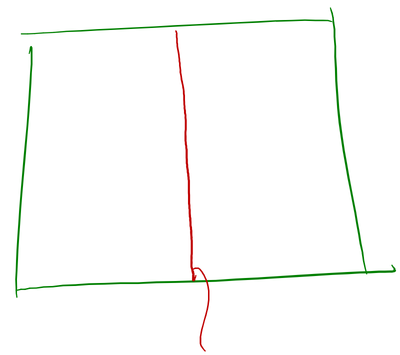
Adding creases without trim curves

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

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

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):

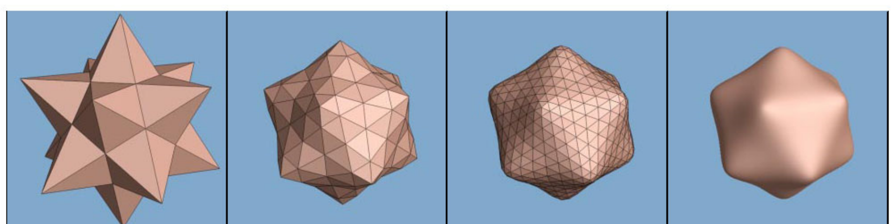


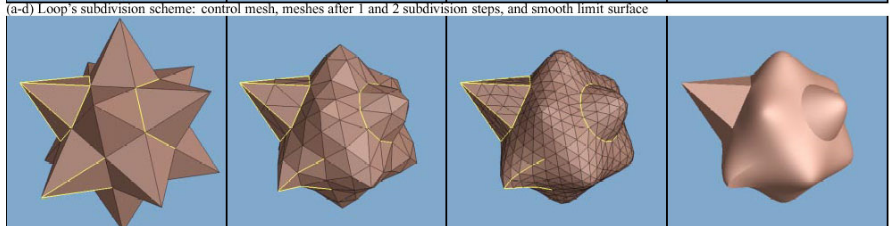


$$\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

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

[Hoppe, SIGGRAPH 1994]

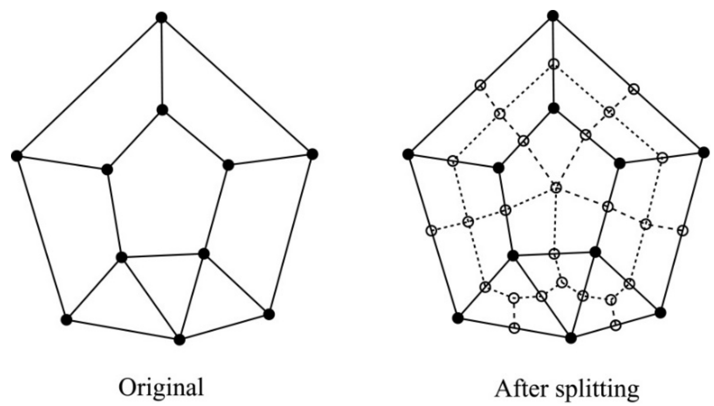

(a-d) Loop's subdivision scheme: control mesh, meshes after 1 and 2 subdivision steps, and smooth limit surface


(e-h) Our piecewise smooth subdivision scheme: tagged control mesh, meshes after 1 and 2 subdivision steps, and piecewise smooth limit surface

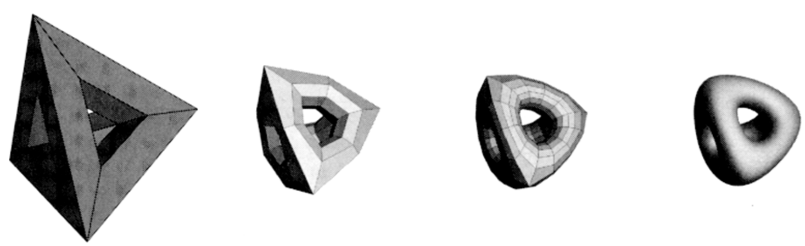
Catmull-Clark subdivision

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:



Note: after the first subdivision, all polygons are quadilaterals 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.