Lecture 20: Surface Modeling

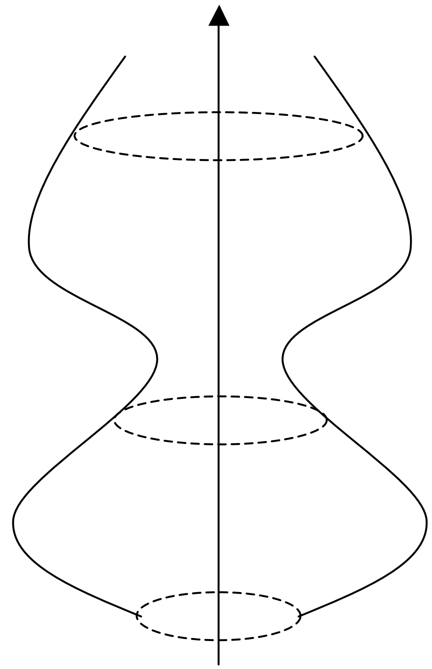
Reading

Hearn and Baker, sections 10.8, 10.9, 10.14.

Optional:

- Bartels, Beatty, and Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, 1987.
- Stollnitz, DeRose, and Salesin. *Wavelets for Computer Graphics: Theory and Applications*, 1996, section 10.2.

Surfaces of revolution



Idea: rotate a 2D **profile curve** around an axis.

What kinds of shapes can you model this way?

Variations

Several variations are possible:

- Scale $C(u)$ as it moves, possibly using length of $T(v)$ as a scale factor.
- Morph $C(u)$ into some other curve $C'(u)$ as it moves along $T(v)$.
- …

Constructing surfaces of revolution

Given: A curve $C(u)$ in the yz-plane:

$$C(u) = \begin{bmatrix} 0 \\ c_y(u) \\ c_z(u) \\ 1 \end{bmatrix}$$

Let $R_x(\theta)$ be a rotation about the x-axis.

Find: A surface $S(u,v)$ which is $C(u)$ rotated about the z-axis.

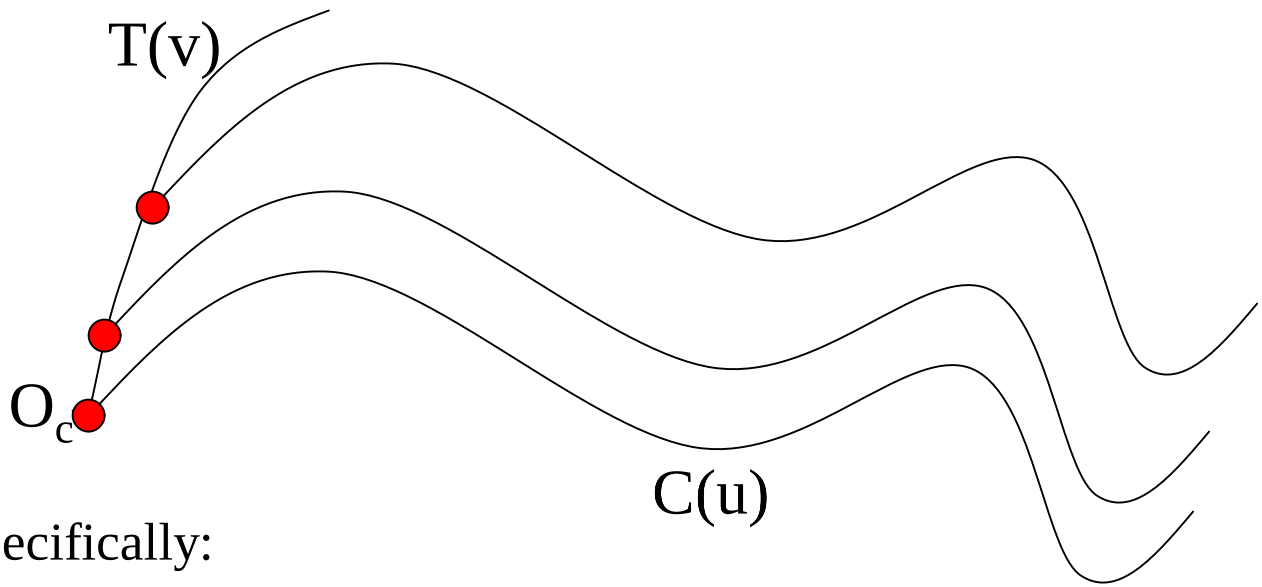
$$S(u,v) = \mathbf{R_x}(v) \cdot C(u)$$

General sweep surfaces

The **surface of revolution** is a special case of a **swept surface**.

Idea: Trace out surface $S(u,v)$ by moving a **profile curve** $C(u)$ along a **trajectory curve** $T(v)$.

$$S(u,v) = \mathbf{T}(T(v)) \cdot C(u)$$



More specifically:

- Suppose that $C(u)$ lies in an (x_c, y_c) coordinate system with origin O_c.
- For every point along $T(v)$, lay $C(u)$ so that O_c coincides with $T(v)$.

Orientation

The big issue:

- How to orient $C(u)$ as it moves along $T(v)$?

Here are two options:

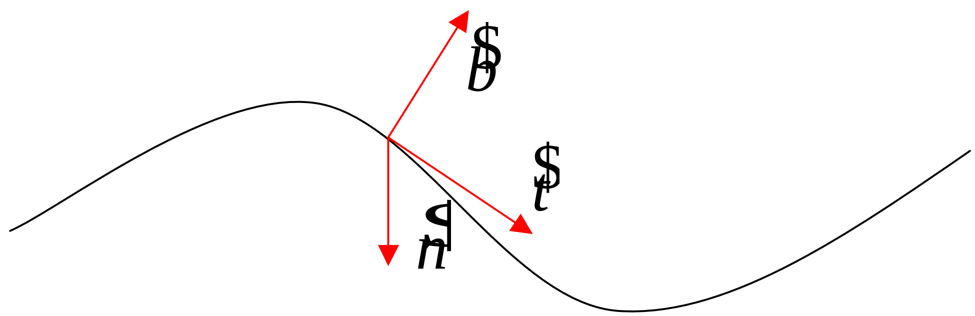
1. **Fixed** (or **static**): Just translate O_c along $T(v)$.

2. Moving. Use the **Frenet frame** of $T(v)$.

- Allows smoothly varying orientation.
- Permits surfaces of revolution, for example.

Frenet frames

Motivation: Given a curve $T(v)$, we want to attach a smoothly varying coordinate system.



To get a 3D coordinate system, we need 3 independent direction vectors.

$$\hat{t}(v) = normalize(T'(v))$$

$$\hat{b}(v) = normalize(T'(v) \times T''(v))$$

$$\hat{n}(v) = \hat{b}(v) \times \hat{t}(v)$$

As we move along $T(v)$, the Frenet frame (t,b,n) varies smoothly.

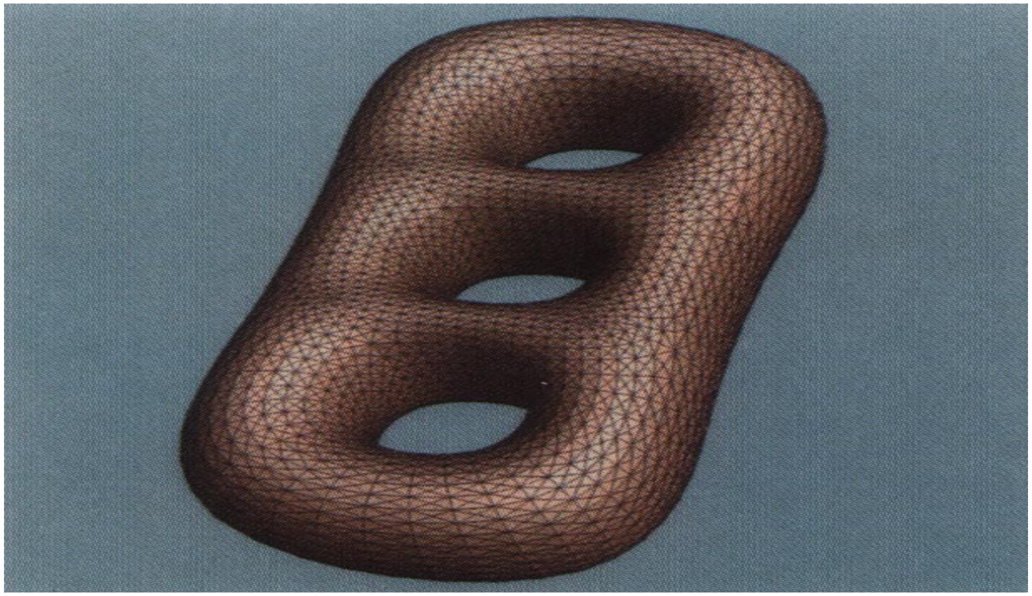
Frenet swept surfaces

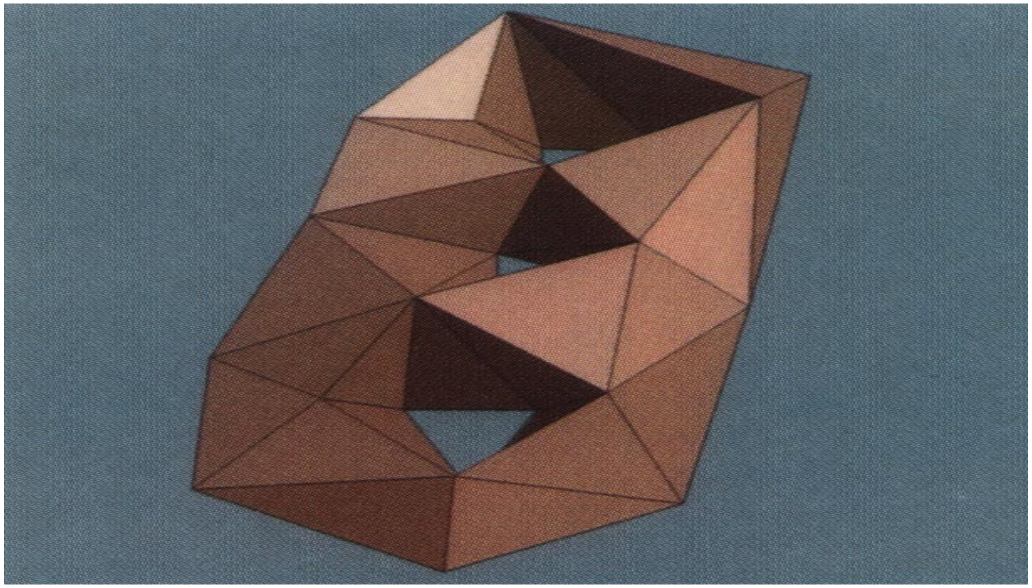
Orient the profile curve $C(u)$ using the Frenet frame of the trajectory $T(v)$:

- Put $C(u)$ in the **normal plane** *nb*.
- Place O_c on $T(v)$.
- Align x_c for $C(u)$ with $-n$.
- Align y_c for $C(u)$ with b.

If $T(v)$ is a circle, you get a surface of revolution exactly?

Building complex models





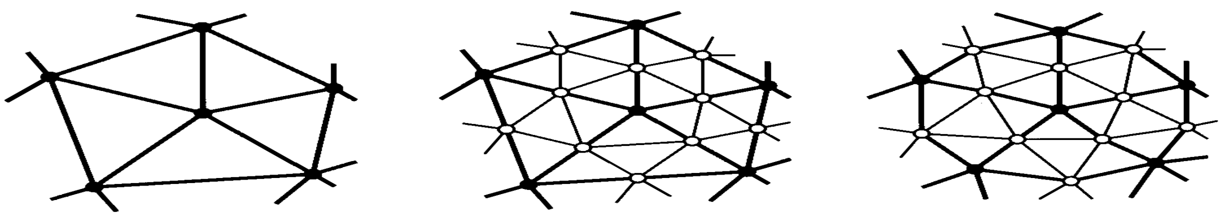
Subdivision surfaces

Chaikin’s use of subdivision for curves inspired similar techniques for subdivision.

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.



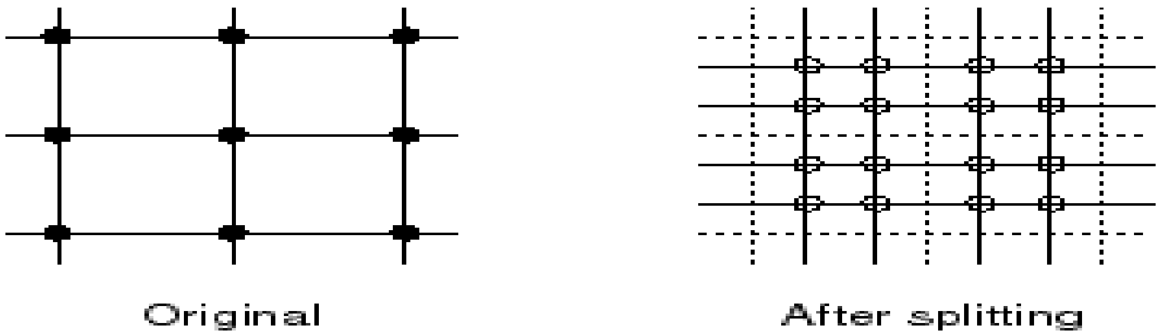
(a) (b) (c)

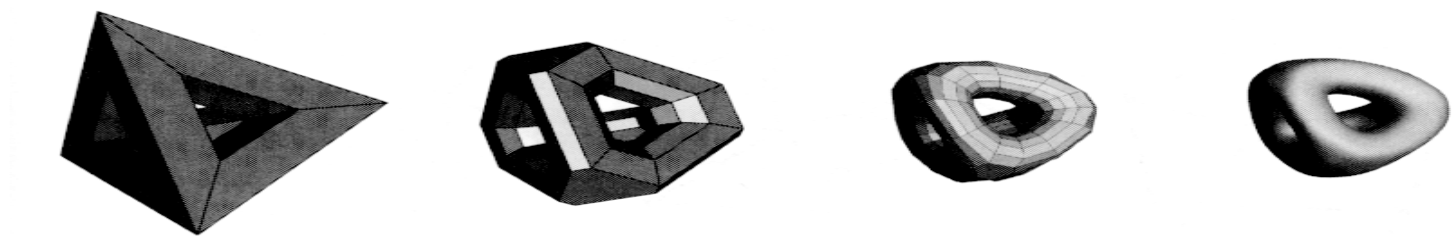
There are two types of splitting steps:

- **vertex schemes**
- **face schemes**

Vertex schemes

A vertex surrounded by *n* faces is split into *n* subvertices, one for each face:

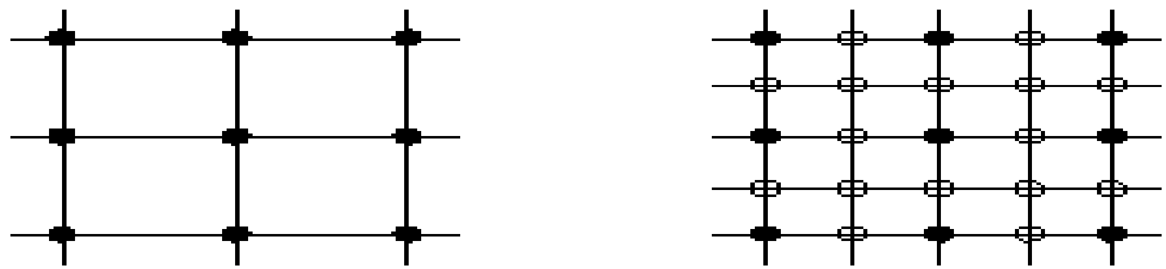




Doo-Sabin subdivision:

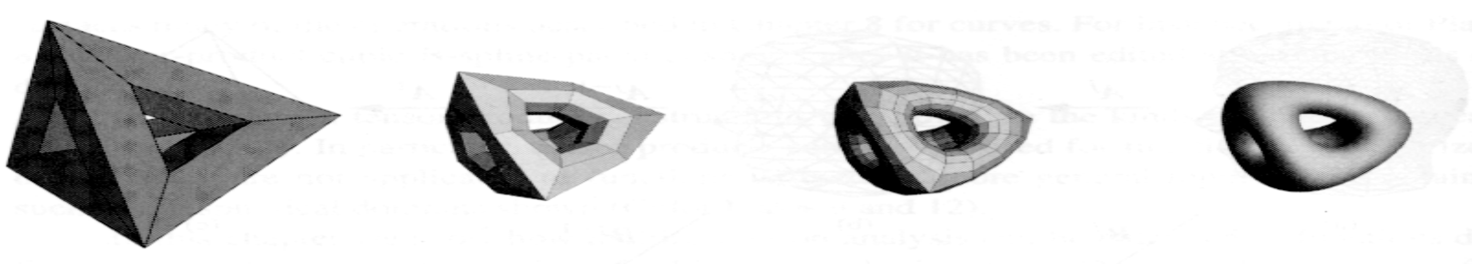
Face schemes

Each quadrilateral face is split into four subfaces:



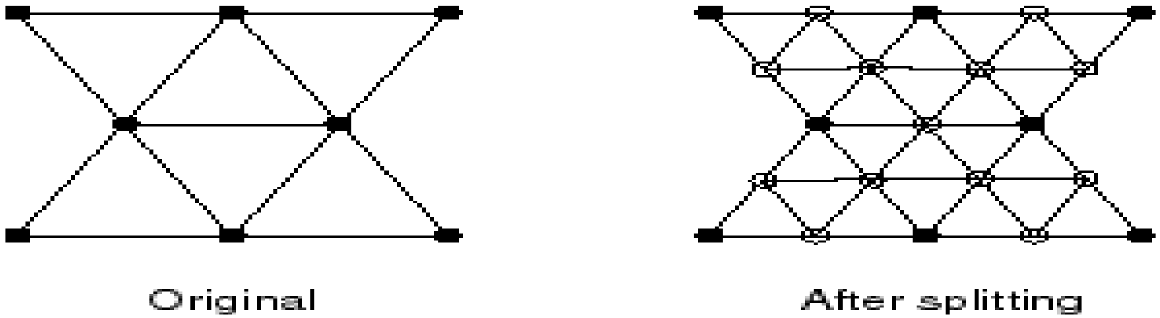
Original   After splitting

Catmull-Clark subdivision:

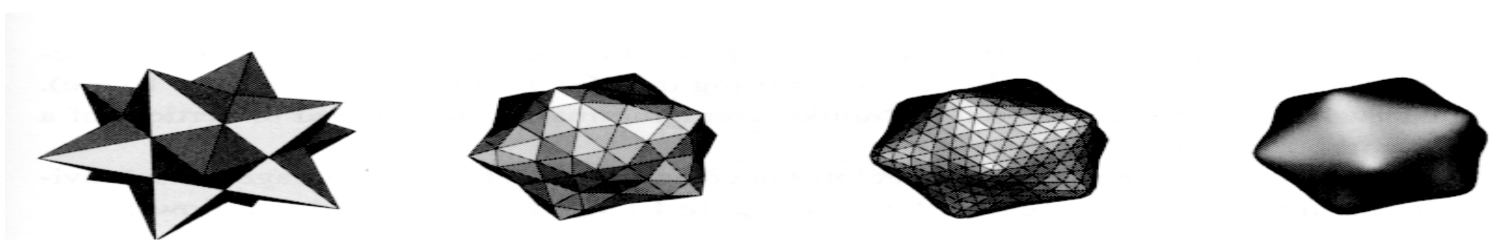


Face schemes, cont.

Each triangular face is split into four subfaces:



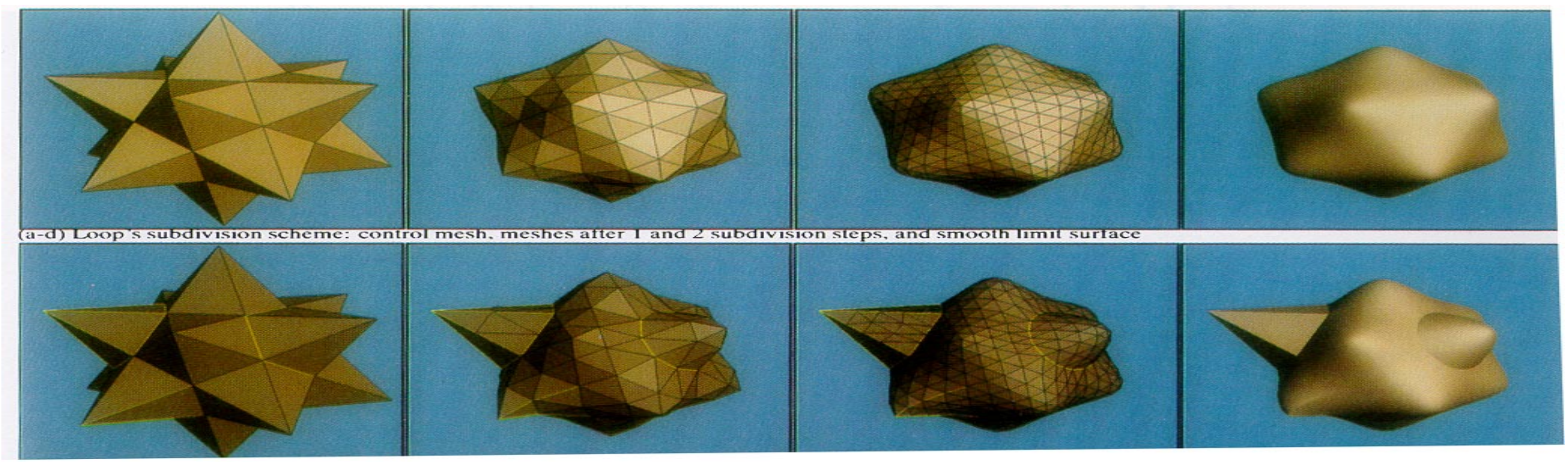
Loop subdivision:



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.

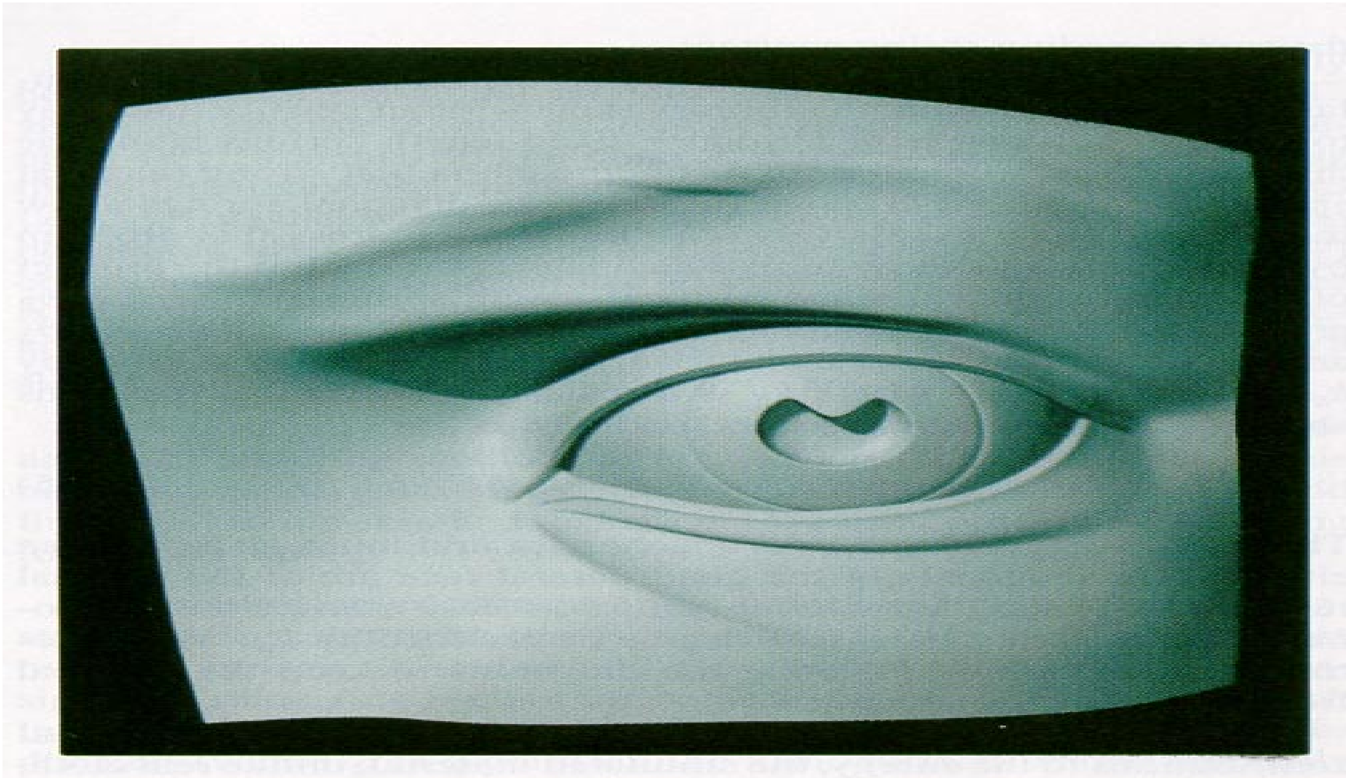
For subdivision surfaces, we can just modify the subdivision mask.



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

Creases with trim curves, cont.

Here's an example using Catmull-Clark surfaces of the kind found in Geri's Game:



Summary

What to take home:

- Surfaces of revolution
- How to construct swept surfaces from a profile and trajectory curve:
  - with a fixed frame
  - with a Frenet frame
- Subdivision surfaces