17. Surfaces

Reading

Recommended:

- Hearn and Baker, sections 10.8, 10.9, 10.14.
- 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?

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 *z*-axis.

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

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** $T(v)$.

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 $T(v)$.

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.

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 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!

What happens at inflection points?

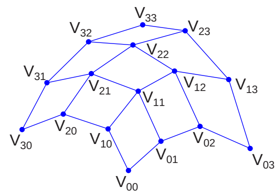
Variations

Several variations are possible:

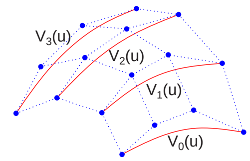
- Scale $C(u)$ as it moves, possibly using length of $T(v)$ as a scale factor.
- Morph $C(u)$ into some other curve $C'(u)$ as it moves along $T(v)$.
- …

Tensor product surfaces, cont.

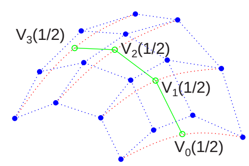
Let's walk through the steps:



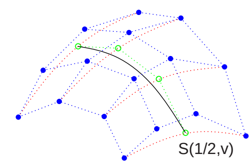
Control net



Control curves in *u*



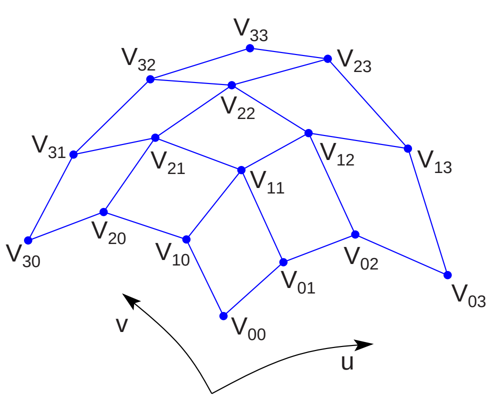
Control polygon at u=1/2



Curve at $S(1/2,v)$

Which control points are interpolated by the surface?

Tensor product Bézier surfaces



Given a grid of control points V_{ij}, forming a **control net**, contruct a surface $S(u,v)$ by:

- treating rows of V as control points for curves $V_0(u),\ldots, V_n(u)$.
- treating $V_0(u),\ldots, V_n(u)$ as control points for a curve parameterized by v.

Matrix form

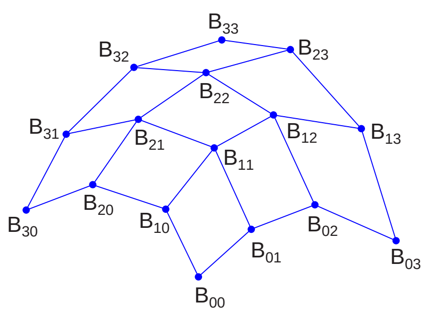
Tensor product surfaces can be written out explicitly:

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} V_{ij} B_i^n(u) B_j^n(v)$$

$$= \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} M_{Bézier}\, V\, M_{Bézier}^T \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

Tensor product B-spline surfaces

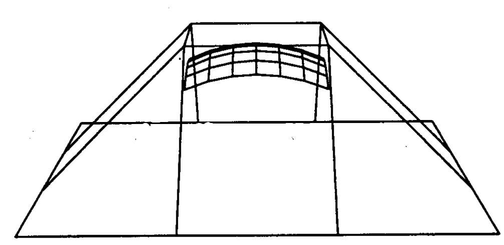
As with spline curves, we can piece together a sequence of Bézier surfaces to make a spline surface. If we enforce C2 continuity and local control, we get B-spline curves:



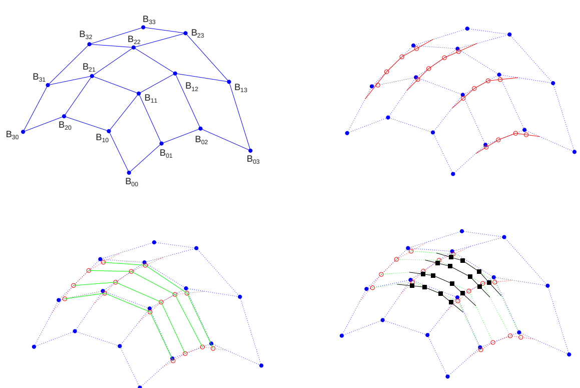
- treat rows of *B* as control points to generate Bézier control points in *u*.
- treat Bézier control points in *u* as B-spline control points in *v*.
- treat B-spline control points in *v* to generate Bézier control points in *u*.

Tensor product B-splines, cont.

Another example:



Tensor product B-splines, cont.



Which B-spline control points are interpolated by the surface?

Trimmed NURBS surfaces

Uniform B-spline surfaces are a special case of NURBS surfaces.

Sometimes, we want to have control over which parts of a NURBS surface get drawn.

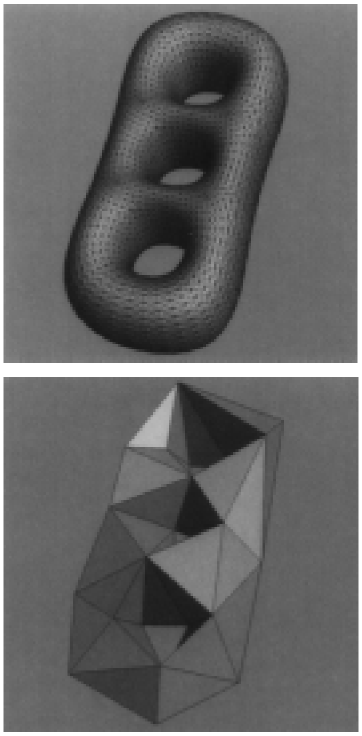
For example:

We can do this by **trimming** the *u*-*v* domain.

- Define a closed curve in the *u*-*v* domain (a **trim curve**)
- Do not draw the surface points inside of this curve.

It's really hard to maintain continuity in these regions, especially while animating.

Building complex models



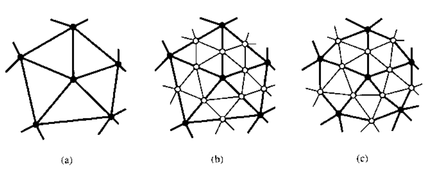
Subdivision surfaces

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.

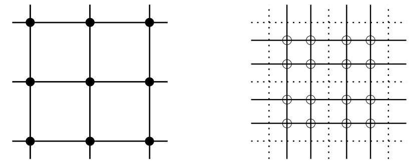


There are two types of splitting steps:

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

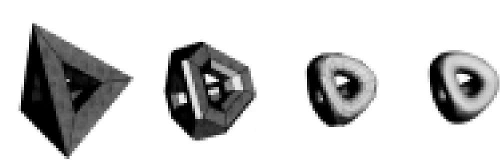
Vertex schemes

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



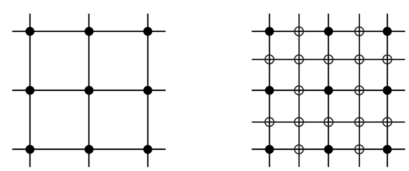
Original　　After splitting

Doo-Sabin subdivision:



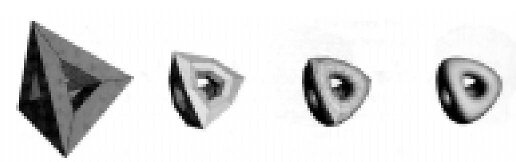
Face schemes

Each quadrilateral face is split into four subfaces:



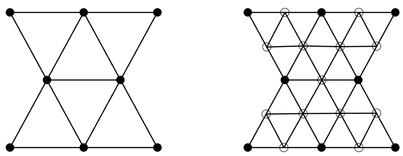
Original　　After splitting

Catmull-Clark subdivision:



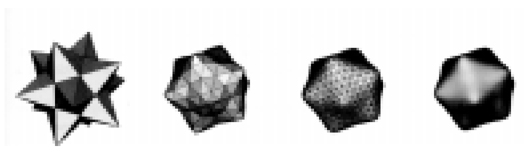
Face schemes, cont.

Each triangular face is split into four subfaces:



Original After splitting

Loop subdivision:



Averaging step

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

where

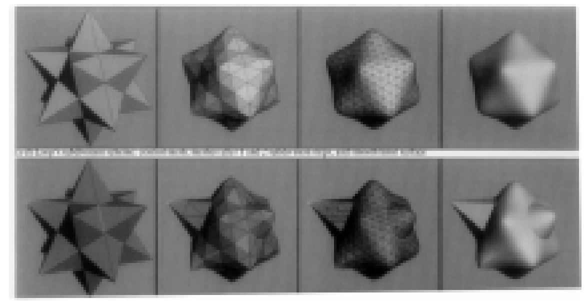
$$\alpha(n) = \frac{n(1-\beta(n))}{\beta(n)} \qquad \beta(n) = \frac{5}{4} - \frac{(3+2\cos(2\pi/n))^2}{32}$$

(carefully chosen to ensure smoothness.)

Adding creases without trim curves

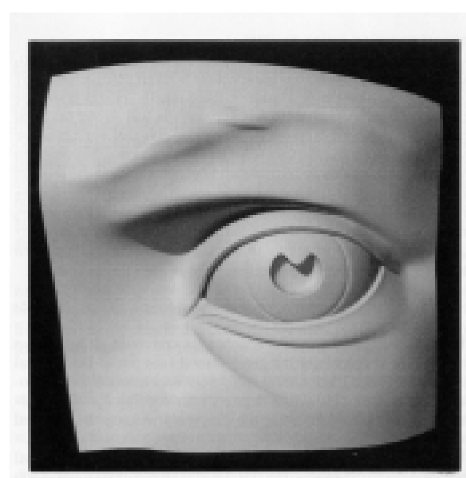
In some cases, we want a particular feature such as a crease to be preserved. With NURBS surfaces, this required the use of trim curves.

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



Creases with trim curves, cont.

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



Interpolating subdivision surfaces

Interpolating schemes are defined by

- splitting
- averaging only new vertices

Summary

What to take home:

- How to construct swept surfaces from a profile and trajectory curve:
  - with a fixed frame
  - with a Frenet frame
- How to construct tensor product Bézier surfaces
- How to construct tensor product B-spline surfaces
- How to construct subdivision surfaces from their averaging masks