12. Parametric surfaces

Reading

Recommended:

- Foley, et al, Section 11.3.
- Bartels, Beatty, and Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, 1987.

Outline

1. Surfaces of revolution
2. General sweep surfaces
3. Tensor product surfaces

Surfaces of revolution

Idea: Rotate a 2D "profile" curve around an axis.

Q: What kinds of shapes can you model this way?

Constructing a surface of revolution

Given: A curve $C(u) : u \in [0,1]$ in the yz-plane:

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

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

Find: A surface $S(u,v) : u, v \in [0,1]$, which is $C(u)$ rotated about the x-axis.

Solution:

General sweep surfaces

The "surface of revolution" is a special case of a "sweep surface"....

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

More specifically:

- Suppose that $C(u)$ lies in an $(\widehat{\mathbf{x}}_\mathbf{c}, \widehat{\mathbf{y}}_\mathbf{c})$ coordinate system with origin $\mathcal{O}_c$.
- For every point along $T(v)$, lay in $C(u)$ so that $T(v)$ coincides with $\mathcal{O}_c$.

Orientation

The big issue:

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

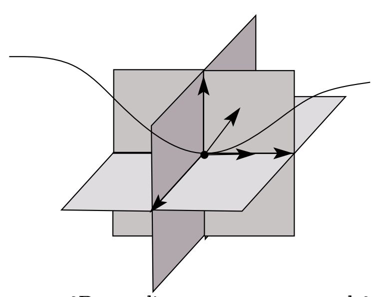
Lots of options. Here are two:

1. Fixed (or "static"): Just translate $\mathcal{O}_c$ 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.



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

- Tangent $\hat{\mathbf{t}}(\mathbf{v}) = \textit{normalize}(\mathbf{T}'(\mathbf{v}))$
- Bi-normal $\widehat{\mathbf{b}}(\mathbf{v}) = \textit{normalize}(\mathbf{T}'(\mathbf{v}) \times \mathbf{T}''(\mathbf{v}))$
- Normal $\widehat{\mathbf{n}}(\mathbf{v}) = \widehat{\mathbf{b}}(\mathbf{v}) \times \hat{\mathbf{t}}(\mathbf{v})$

As we move along $T(v)$, the Frenet frame $(\hat{\mathbf{t}}, \widehat{\mathbf{b}}, \widehat{\mathbf{n}})$ varies smoothly.

Frenet swept surfaces

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

- Put $C(u)$ in the "normal" plane span$(\widehat{\mathbf{n}}, \widehat{\mathbf{b}})$.
- Place $\mathcal{O}_c$ on $T(v)$.
- Align $\widehat{\mathbf{x}}_{\mathbf{c}}$ for $C(u)$ with $\widehat{\mathbf{b}}$.
- Align $\widehat{\mathbf{y}}_{\mathbf{c}}$ for $C(u)$ with $\widehat{\mathbf{n}}$.

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

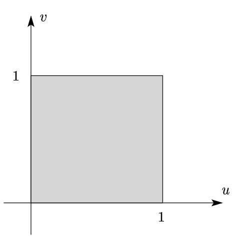
Problem: What do you do when the curvature changes sign or goes to 0?

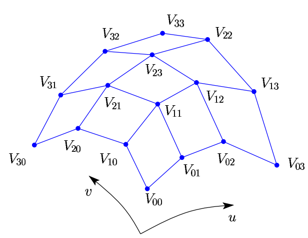
Variations

Several variations are possible:

- Scale $C(u)$ as it moves, possibly using length of $T(v)$ as a scale factor — great for seashells!
- Morph $C(u)$ into some other curve $\tilde{C}(u)$ as it moves along $T(v)$.
- Use your imagination....

Tensor product Bézier surfaces





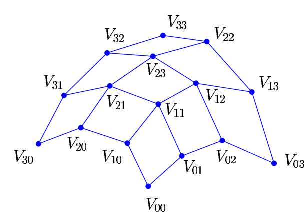
Given: A grid of control points $V_{ij} : i, j = 0, \ldots, n$. Connecting these together we get a "control net."

Construct: A surface $S(u, v)$:

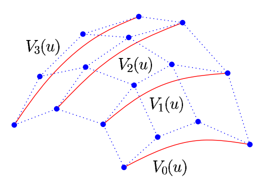
- Treat rows of V as control points for curves $V_0(u), \ldots, V_n(u)$.
- For a given value of u, treat $V_0(u), \ldots, V_n(u)$ as control points for a curve parameterized by v.

Tensor product Bézier surfaces, cont'd

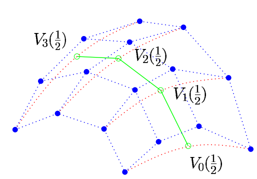
Let's walk through the steps:



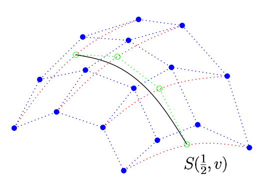
Control net



Control curves in u



Control polygon at $u = \frac{1}{2}$



Curve at $S(\frac{1}{2}, v)$

Q: Which control points are interpolated by the surface?

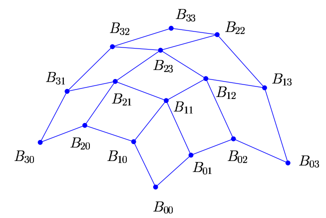
Matrix form

Tensor product Bézier surfaces can be written out explicitly:

$$
\begin{aligned}
S(u,v) &= \sum_{i,j=0,\ldots,n} V_{ij} B_i^n(u) B_j^n(v) \\
&= \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} M_{\text{Bézier}} \, V \, M_{\text{Bézier}}^{\mathrm{T}} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}
\end{aligned}
$$

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 C^2 continuity and local control, we get B-spline curves:



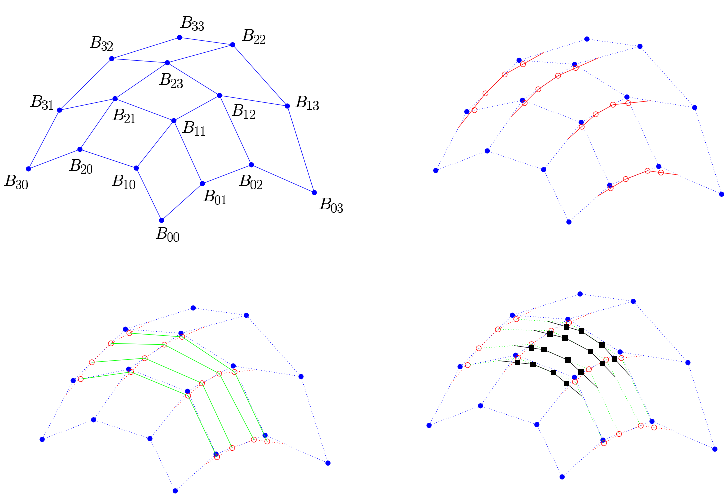
Given: A grid of control points $B_{ij} : i,j = 0,\ldots,n$.

Construct: A Bézier control net $V_{ij} : i,j = 0,\ldots,n$:

- 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 v.

Tensor product B-spline surfaces, cont'd

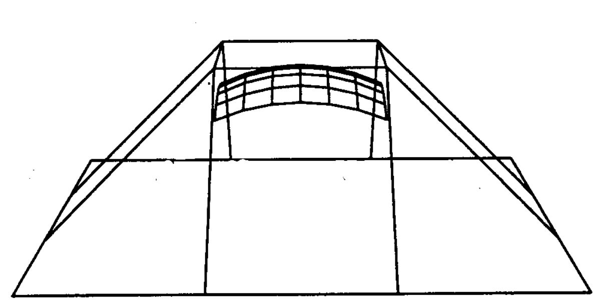
Let's walk through the steps:



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

Tensor product B-spline surfaces, cont'd

Another example:



B-spline surface and control net (Bartels, et al)

Matrix form

Tensor product surfaces can be written out explicitly:

$$S(u,v) = \sum_{i,j=0,\ldots,n} B_{ij} N_i^n(u) N_j^n(v)$$

$$= \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} M_{\text{Bézier}}\, M_{\text{B-spline}}\, B\, M_{\text{B-spline}}^{\mathrm{T}}\, M_{\text{Bézier}}^{\mathrm{T}} \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

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:

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

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