Parametric surfaces

Brian Curless
CSEP 557
Winter 2013

Mathematical surface representations

- Explicit \( z = f(x, y) \) (aka a “height field”)
  - what if the curve isn’t a function, like a sphere?

- Implicit \( g(x, y, z) = 0 \)

\[
\begin{align*}
  x^2 + y^2 + z^2 &= r^2 \\
  g(x, y, z) &= x^2 + y^2 + z^2 - r^2
\end{align*}
\]

- Parametric \( S(u, v) = (x(u, v), y(u, v), z(u, v)) \)
  - For the sphere:
    \[
    \begin{align*}
    x(u, v) &= r \cos u \sin v \\
    y(u, v) &= r \sin u \sin v \\
    z(u, v) &= r \cos v
    \end{align*}
    \]
  - As with curves, we'll focus on parametric surfaces.

Surfaces of revolution

Recall that surfaces of revolution are based on the idea of rotating about an axis...

- Given: A set of points \( C(i) \) on a curve in the x-y plane:

  \[
  C(i) = \begin{bmatrix} x_0[i] \\ y_0[i] \\ 0 \end{bmatrix}
  \]

  where \( i \in [0, N-1] \)

- Let \( R_n \) be rotation about the y-axis angle \( \theta_j \)

- Find: A set of points \( S(j) \) on the surface formed by rotating \( C(i) \) rotated about the y-axis. Assume \( j \in [0, M-1] \)

  Solution:

\[
S[j] = R_n[R_1[C[i]]]
\]

\[
= R_1[R_2[C[i]]]C[j]
\]
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 $T(v)$.

![Diagram of sweep surfaces]

More specifically:
- Suppose that $C(u)$ lies in an $(x,y,z)$ coordinate system with origin $O_z$.
- For every point along $T(v)$, lay $C(u)$ so that $O_z$ 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 $O_z$ along $T(v)$.

![Diagram of fixed orientation]

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.

![Diagram of Frenet frame]

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

Tangent: $t(v) = \text{normalize}(T'(v))$
Binormal: $b(v) = \text{normalize}(T'(v) \times T''(v))$
Normal: $n(v) = b(v) \times t(v)$

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.
- Place $O_z$ on $T(v)$.
- Align $x_c$ for $C(u)$ with $b$.
- Align $y_c$ for $C(u)$ with $-n$.

![Diagram of Frenet swept surfaces]

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

Let's look back at where we computed the coordinate frames from curve derivatives:

Where might these frames be ambiguous or undetermined?

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 \( \tilde{C}(u) \) as it moves along \( T(v) \).
- ...

Tensor product Bézier surfaces

Given a grid of control points \( V \), forming a **control net**, construct a surface \( S(u,v) \) by:

- treating rows of \( V \) (the matrix consisting of the \( V_j \)) as control points for curves \( V_j(\omega), \ldots, V_j(\nu) \).
- treating \( V_0(\mu), \ldots, V_r(\nu) \) as control points for a curve parameterized by \( V \).

Tensor product Bézier 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?
**Polynomial form of Bézier surfaces**

Recall that cubic Bézier curves can be written in terms of the Bernstein polynomials:

\[ Q(t) = \sum_{i=0}^{3} b_i(t) \]

A tensor product Bézier surface can be written as:

\[ S(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} b_i(u) b_j(v) \]

In the previous slide, we constructed curves along \( u \) and then along \( v \). This corresponds to re-grouping the terms like so:

\[ S(u, v) = \sum_{i=0}^{3} \left( \sum_{j=0}^{3} b_j(v) \right) b_i(u) \]

But, we could have constructed them along \( v \), then \( u \):

\[ S(u, v) = \sum_{j=0}^{3} \left( \sum_{i=0}^{3} b_i(u) \right) b_j(v) \]

**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:

- 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:

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

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

Trimmed 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.

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