Surfaces

Reading

Required:

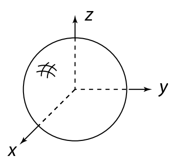
- Watt, 2.1.4, 3.4-3.5.

Optional

- Watt, 3.6.
- Bartels, Beatty, and Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling,* 1987.

Mathematical surface representations

- Explicit $z=f(x,y)$ (a.k.a., a "height field")
  - what if the surface isn't a function?



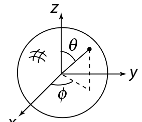
- Implicit $g(x,y,z) = 0$

- Parametric $(x(u,v),y(u,v),z(u,v))$
  - For the sphere:

    $x(u,v) = r \cos 2\pi v \sin \pi u$

    $y(u,v) = r \sin 2\pi v \sin \pi u$

    $z(u,v) = r \cos \pi u$



We'll focus mostly on parametric surfaces.

Surfaces of revolution

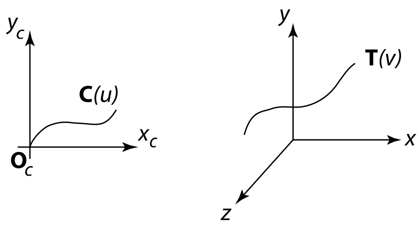
Idea: rotate a 2D **profile curve** around an axis.

What kinds of shapes can you model this way?

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 $\mathbf{O}_c$.
- For every point along **T**(v), lay **C**(u) so that $\mathbf{O}_c$ coincides with **T**(v).

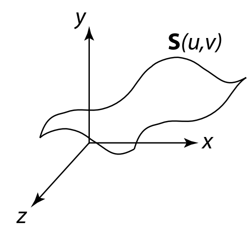
Orientation

The big issue:

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

Here are two options:

1. **Fixed** (or **static**): Just translate $\mathbf{O}_c$ along **T**(v).



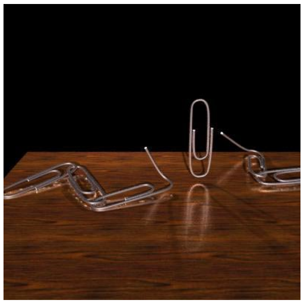
2. Moving. Reorient as you move along, based on orientation of **T**(v

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

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 $\tilde{\mathbf{C}}(u)$ as it moves along **T**(v)



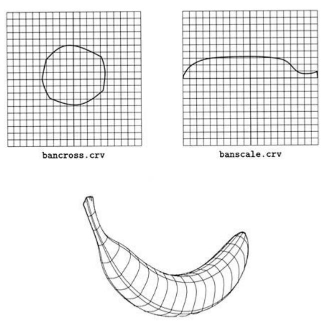
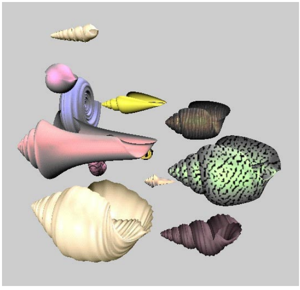
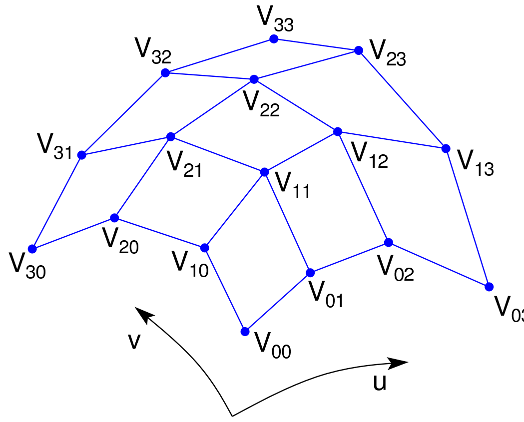


Figure 3.8: Banana example. A banana is represented by an affine transformation surface. The cross section is scaled, translated along z from −1 to 1, and rotated around the y axis.



Tensor product Bézier surfaces

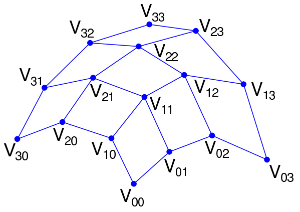


Given a grid of control points $\mathbf{V}_{ij}$, forming a **control net**, contruct a surface **S**(u,v) by:

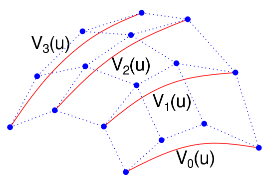
- treating rows of **V** (the matrix consisting of the $\mathbf{V}_{ij}$) as control points for curves $\mathbf{V}_0(u)$,…, $\mathbf{V}_n(u)$.
- treating $\mathbf{V}_0(u)$,…, $\mathbf{V}_n(u)$ as control points for a curve parameterized by v.

Tensor product Bézier surfaces, cont.

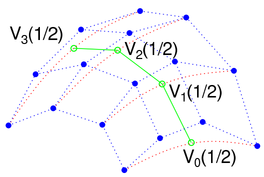
Let's walk through the steps:



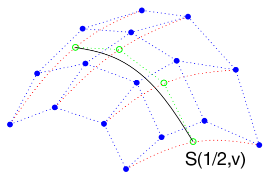
Control net



Control curves in *u*



Control polygon at *u*=1/2



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

Which control points are interpolated by the surface?

Matrix form of Bézier surfaces

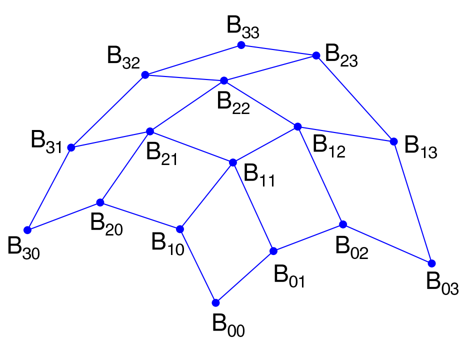
Tensor product surfaces can be written explicitly:

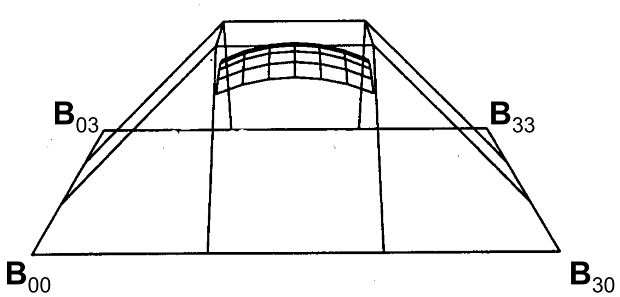
$$\mathbf{S}(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{n}\mathbf{V}_{ij}P_i^n(u)P_j^n(v)$$

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

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

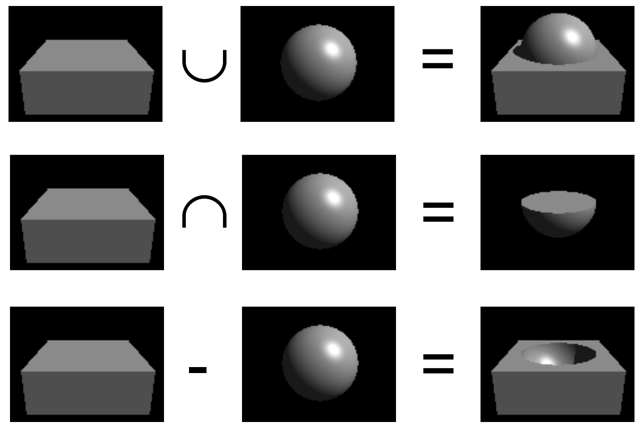




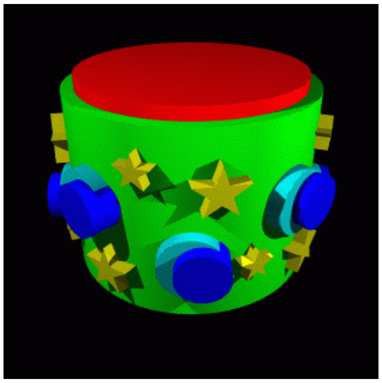
Constructive solid geometry

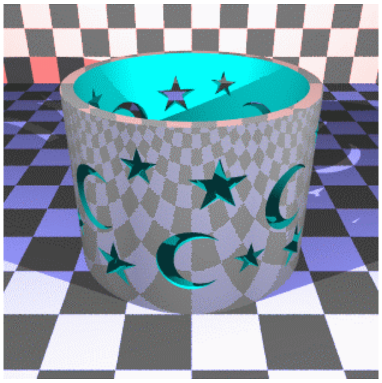
Simple shapes can be combined together to make more complex shapes. This process is called **constructive solid geometry (CSG)**

- glue pieces together
- saw parts off, drill holes



CSG, cont.





CSG with implicit functions

CSG operations are easier to implement with implicit functions.

Let f(x,y,z) and g(x,y,z) be implicit representations of two shapes where

- $f(x,y,z) < 0$ if (x,y,z) is inside the shape
- $f(x,y,z) > 0$ if (x,y,z) is outside the shape
- $f(x,y,z) = 0$ if (x,y,z) is on the surface

h = **union** of f and g

- $h(x,y,z) = \min(f(x,y,z), g(x,y,z))$

h = **intersection** of f and g

- $h(x,y,z) =$

h = f – g

- $h(x,y,z) =$

Summary

What to take home:

- How to construct swept surfaces from a profile and trajectory curve
- How to construct tensor product Bézier surfaces
- CSG with implicit functions