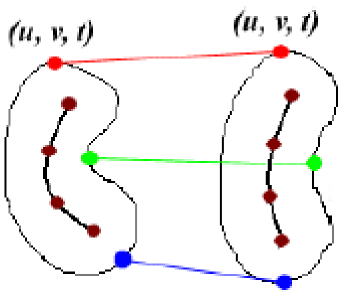
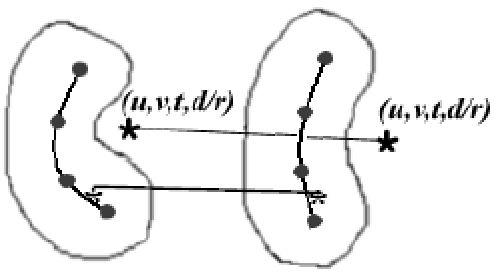
Free Form Deformation

Simulating Deformation
-- various approaches

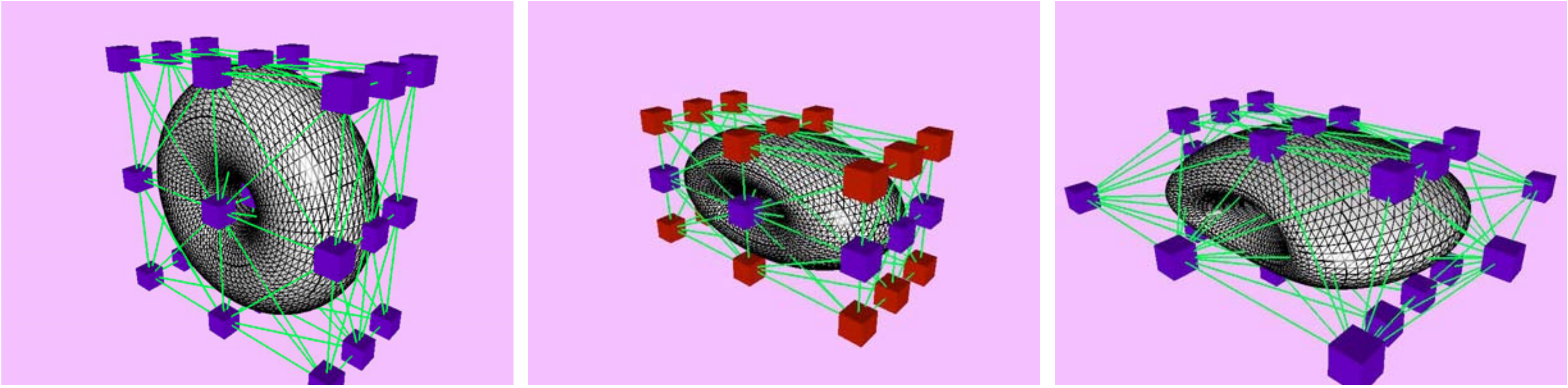
- Physics-based Simulations
  - Particle system, string system…
  - Finite differences, finite element, boundary element methods…
  - …
- …
- **Free form deformation**
- M-reps based deformation





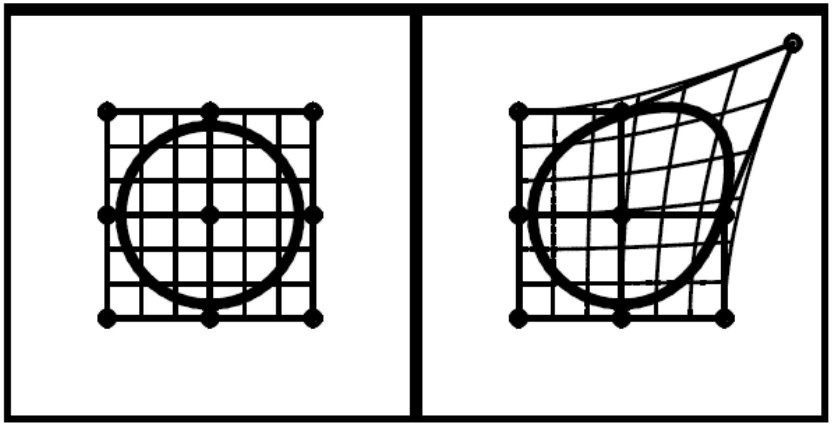
Free-form deformation (FFD)

- Embed the object into a domain that is more easily parametrized than the object.
- Advantages:
  - You can deform arbitrary objects
  - Independent of object representation



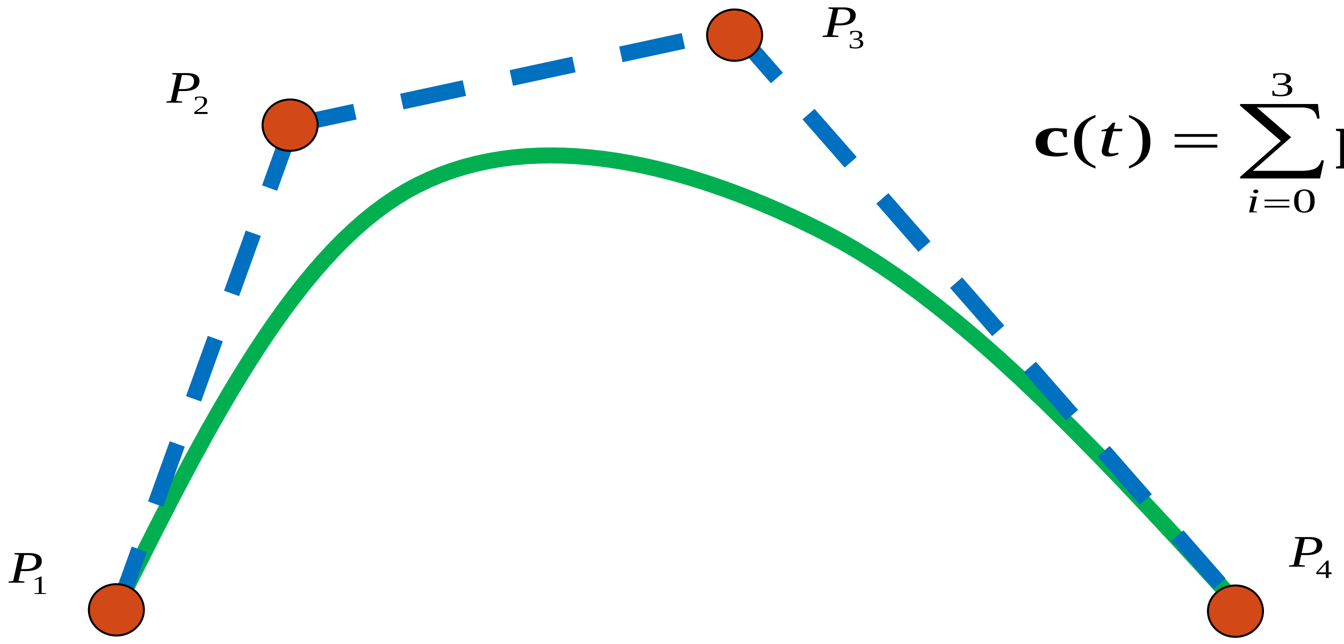
2D FFD

- 2D FFD = a map from R2 to R2
  - When the space deforms, defines a new position for every point in the new space
  - Any lines or curves that lie in that space are altered (see the figure)
  - The space deformation is easily manipulated by control points



Remind the "Bezier Curve"

Interpolate the two end control points,
and approximates the other two points:



$$\mathbf{c}(t) = \sum_{i=0}^{3} \mathbf{p}_i B_i^3(t)$$

$$Q'(0) = 3(P_2 - P_1); Q'(1) = 3(P_4 - P_3)$$

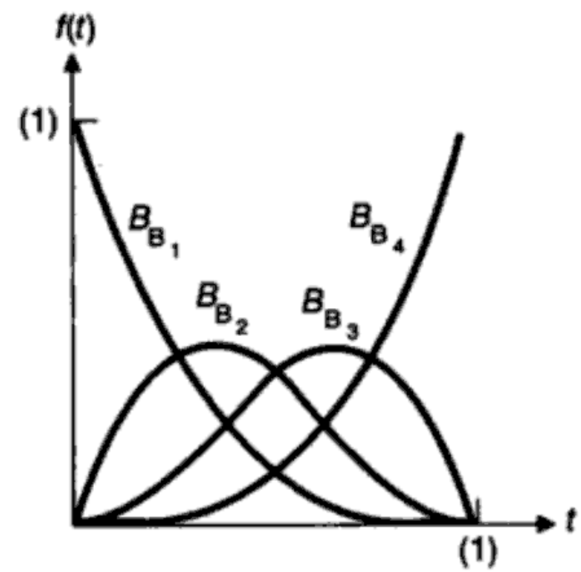
Review: Bernstein Polynomials

- Bezier curve

$$\mathbf{c}(t) = \sum_{i=0}^{3} \mathbf{p}_i B_i^3(t)$$

- Control points P and basis functions B

$$B_0^3(t) = (1-t)^3$$
$$B_1^3(t) = 3t(1-t)^2$$
$$B_2^3(t) = 3t^2(1-t)$$
$$B_3^3(t) = t^3$$



Review: B-Splines and NURBS

B-Spline ➡ $$\mathbf{c}(u) = \sum_{i=0}^{n} \mathbf{p}_i B_{i,k}(u)$$

- **NURBS** = Non Uniform Rational B-Splines

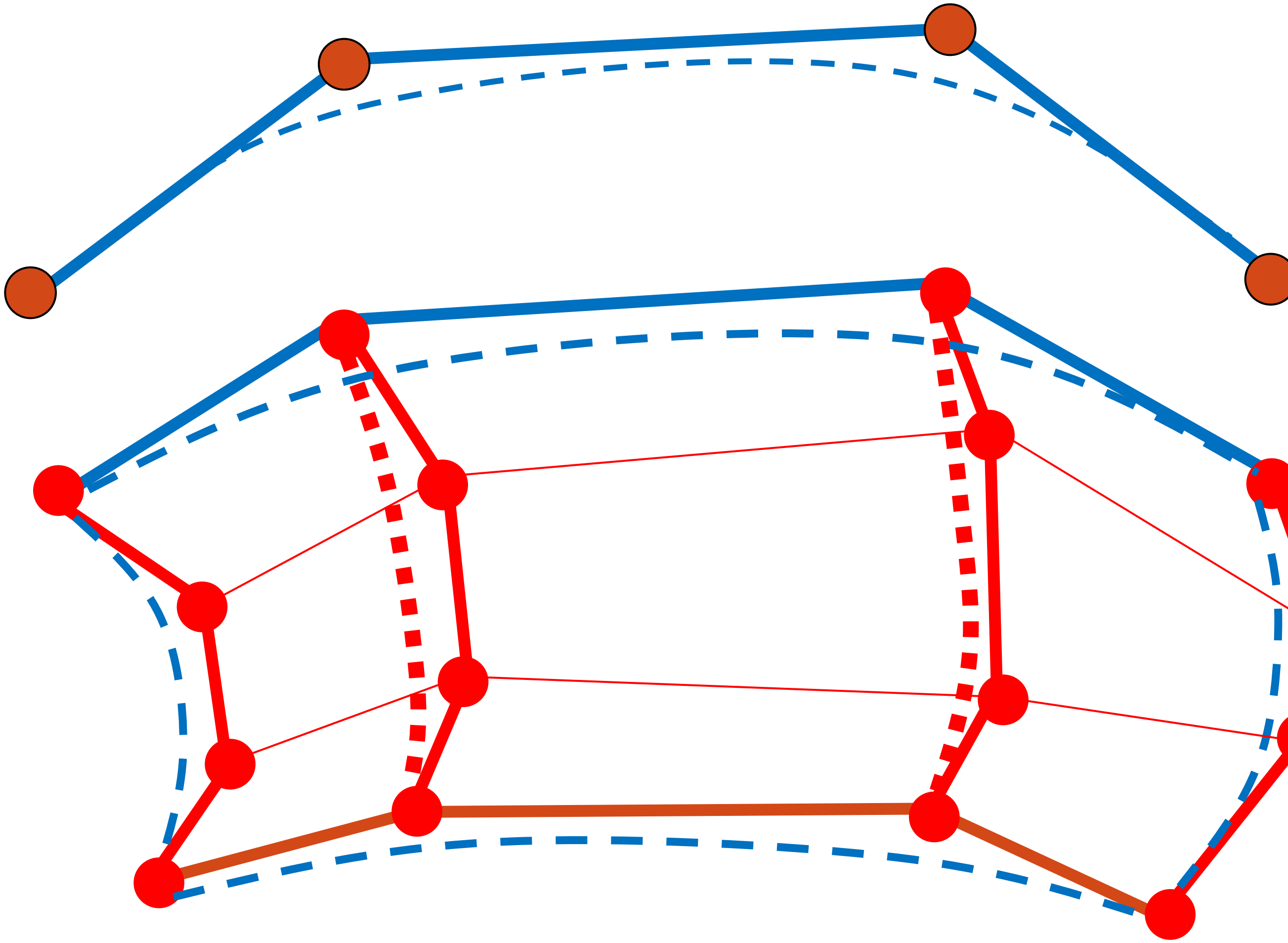
By generalizing B-splines using Homogeneous Coordinates

$$\mathbf{c}(u) = \sum_{i=0}^{n} \begin{bmatrix} \mathbf{p}_{i,x} w_i \\ \mathbf{p}_{i,y} w_i \\ \mathbf{p}_{i,z} w_i \\ w_i \end{bmatrix} B_{i,k}(u)$$

NURBS ➡ $$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{i=0}^{n} w_i B_{i,k}(u)}$$

1D→2D

B-spline Curves → B-spline Surfaces



Tensor Product Surfaces

Basis functions: bivariate functions of u and v
(constructed as products of univariate basis functions)

A tensor product surface

$$S^T(u,v) = (x(u,v), y(u,v), z(u,v)) = \sum_{i=0}^{n} \sum_{j=0}^{m} f_i(u) g_j(v) b_{i,j};$$

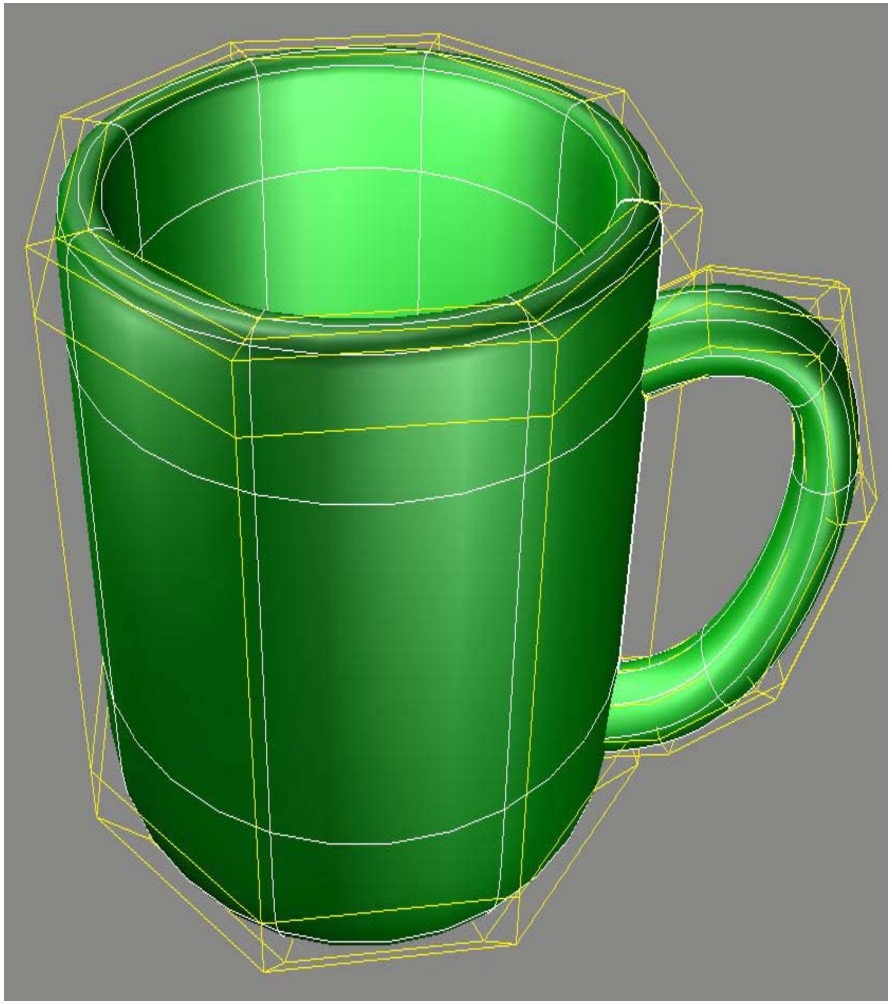
$$where \begin{cases} b_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ 0 \le u, v \le 1 \end{cases}$$

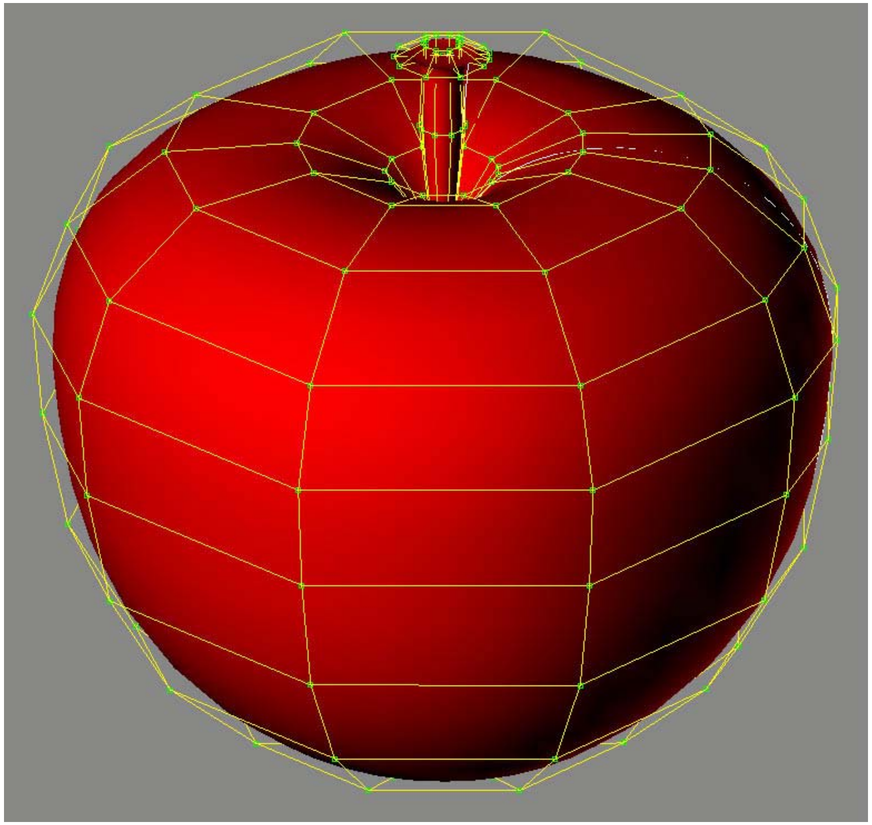
The (u,v) domain of this mapping is a square (rectangle)

$$S^T(u,v) = [f_i(u)]^T [b_{i,j}][g_j(u)]$$

(n+1)*(m+1) matrix of 3D points

NURBS Surface Examples





2D→3D

Parametric Solids

- Tricubic solid

$$\mathbf{p}(u,v,w)=\sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3}\mathbf{a}_{ijk}u^{i}v^{j}w^{k}$$

$$u,v,w\in[0,1]$$

- Bezier solid

$$\mathbf{p}(u,v,w)=\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}B_{i}(u)B_{j}(v)B_{k}(w)$$

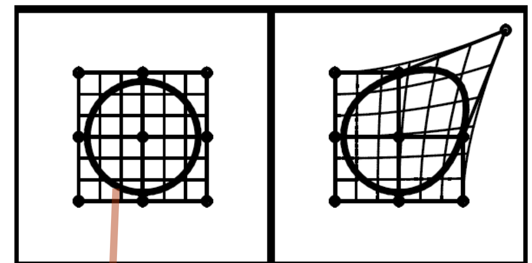
- B-spline solid

$$\mathbf{p}(u,v,w)=\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)$$

- NURBS solid

$$\mathbf{p}(u,v,w)=\frac{\sum_{i}\sum_{j}\sum_{k}\mathbf{p}_{ijk}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}{\sum_{i}\sum_{j}\sum_{k}q_{ijk}B_{i,I}(u)B_{j,J}(v)B_{k,K}(w)}$$

Formulation of 2D FFD

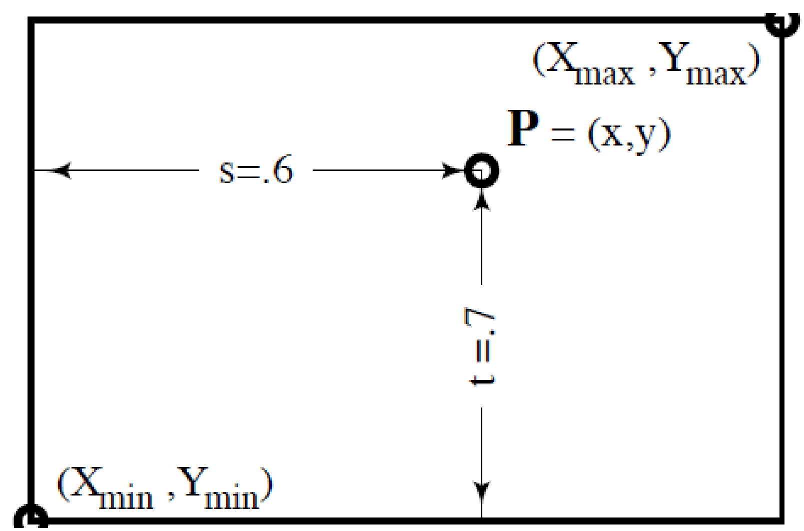


$$\mathbf{X}(s,t) = \frac{\sum_{j=0}^{n}\sum_{i=0}^{m} w_{ij} B_i^m(s) B_j^n(t) \mathbf{P}_{ij}}{\sum_{j=0}^{n}\sum_{i=0}^{m} w_{ij} B_i^m(s) B_j^n(t)}$$

where $B_i^n(t)$ and $B_j^m(s)$ are Bézier blending functions, and s and t are the local coordinates of a point with respect to the deformation region.

$\mathbf{P}_{ij}$ are the actual (x, y) coordinates of the displaced control point i, j.

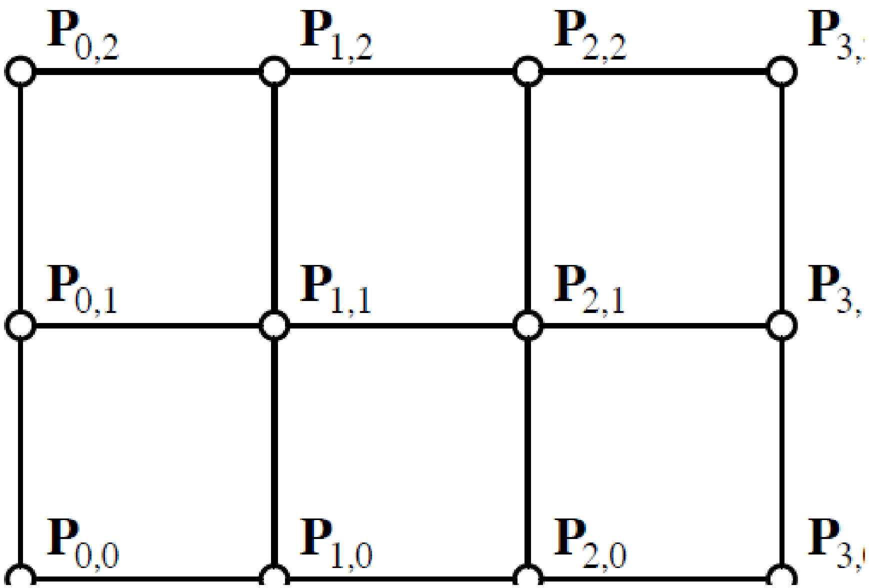
$$s = \frac{x - X_{min}}{X_{max} - X_{min}}, \quad t = \frac{y - Y_{min}}{Y_{max} - Y_{min}}.$$



Formulation of 2D FFD (cont.)

In their initial, undisplaced position, the control points form a rectangular grid:

$$\mathbf{P}_{i,j} = \left(X_{min} + \frac{i}{m}(X_{max} - X_{min}), Y_{min} + \frac{j}{n}(Y_{max} - Y_{min}) \right)$$

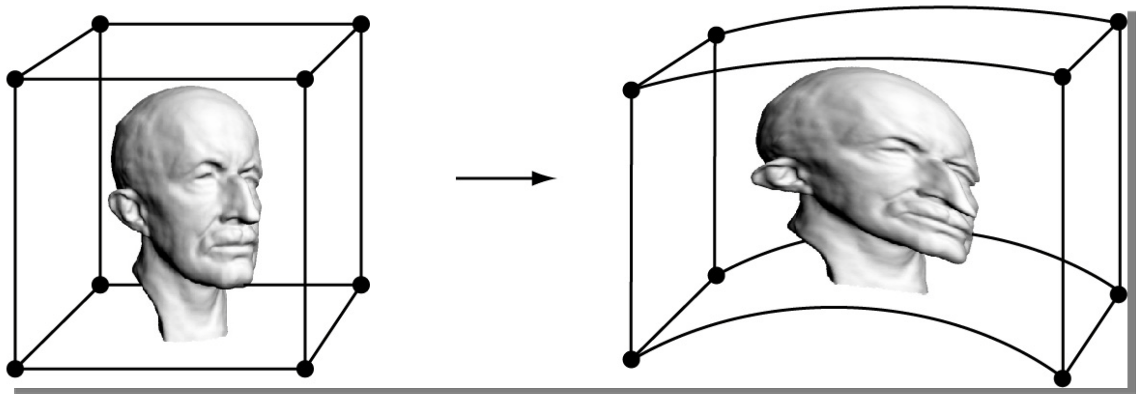


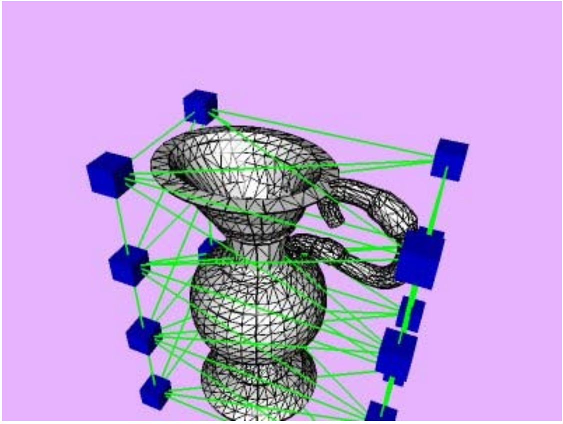
To deform the object:

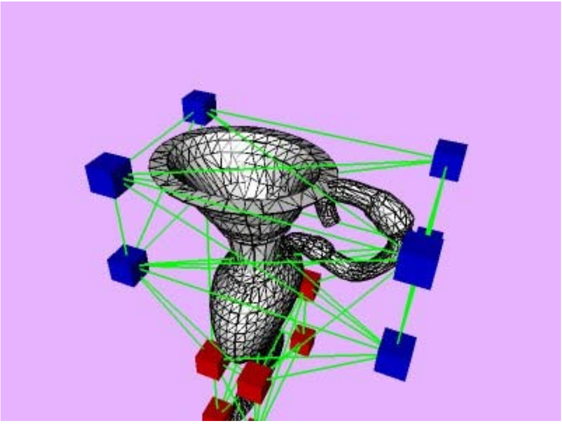
1. Generate control points net
2. Compute (s,t) coordinates of every sample point Q
3. Move the control points
4. Evaluate new position of Q using

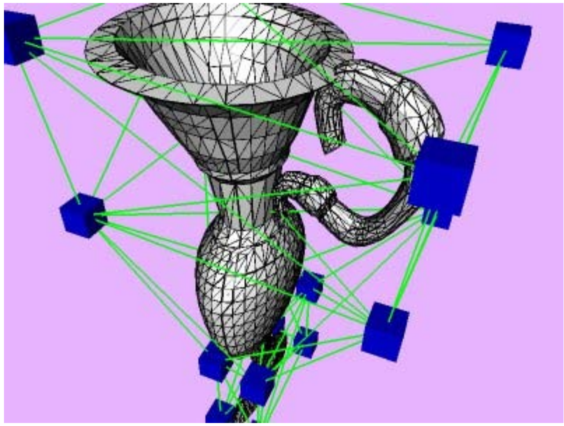
$$\mathbf{X}(s,t) = \frac{\sum_{j=0}^{n} \sum_{i=0}^{m} w_{ij} B_i^m(s) B_j^n(t) \mathbf{P}_{ij}}{\sum_{j=0}^{n} \sum_{i=0}^{m} w_{ij} B_i^m(s) B_j^n(t)}$$

Generalize to 3D easily









Another example and application