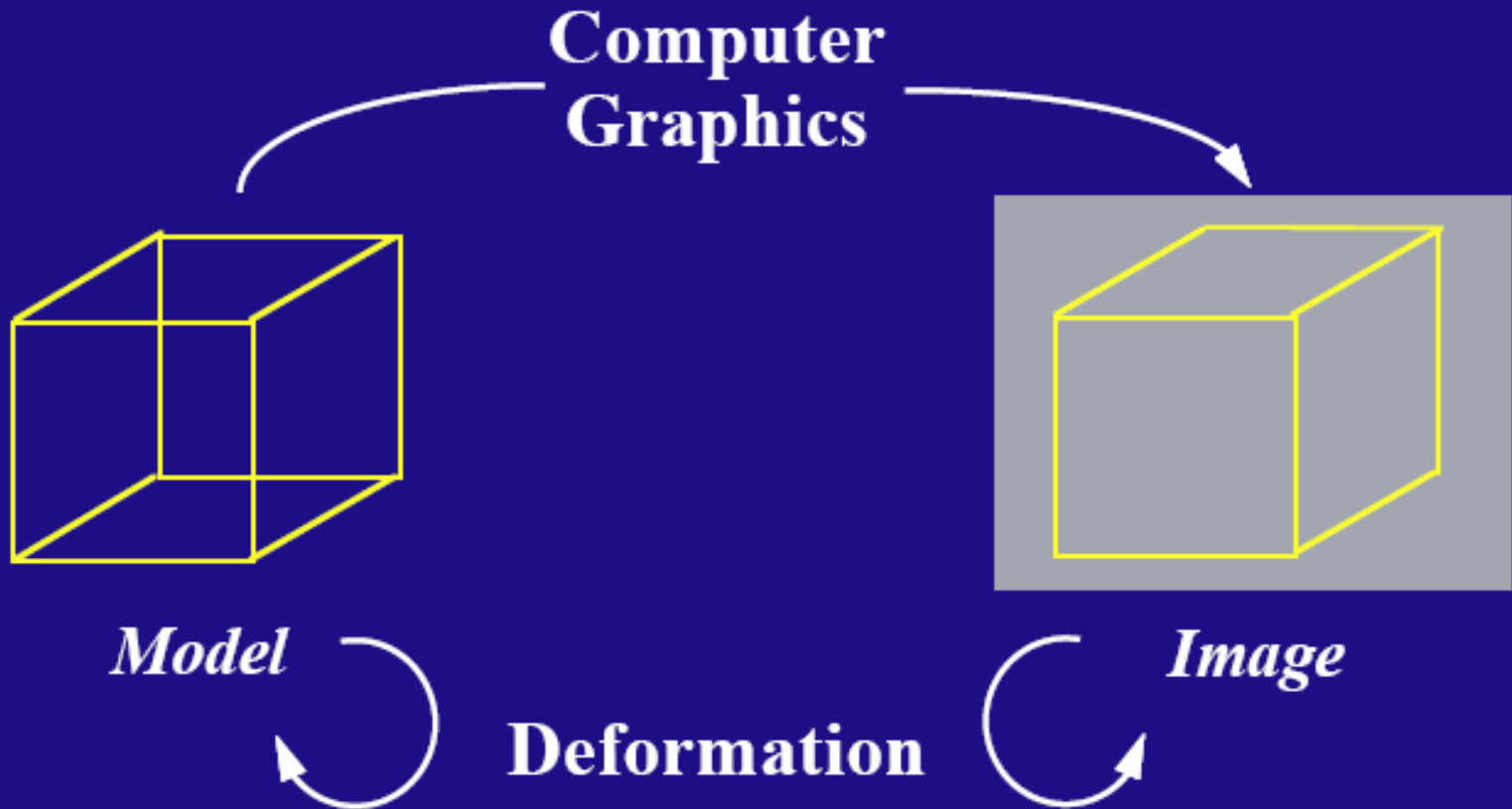


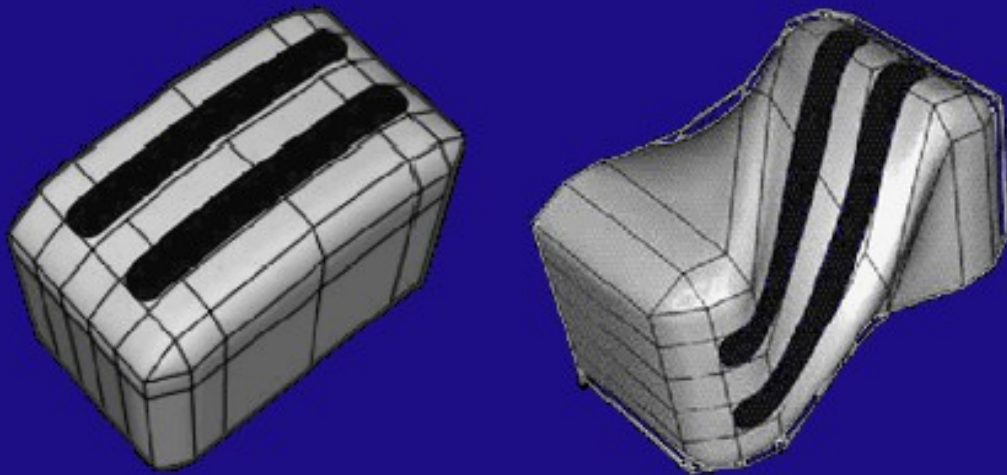
# 3D krivljenje (deformacija objektov)



# Metamorphosis and Graphics



# Model Deformation

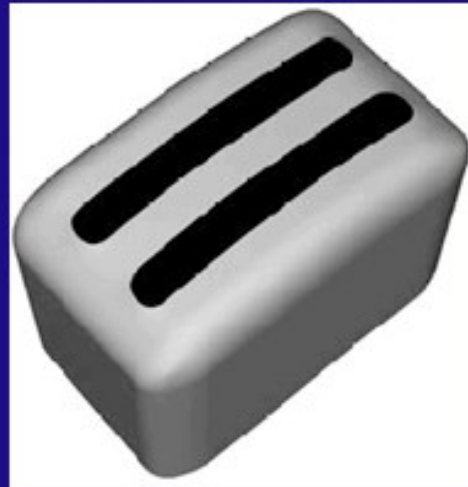
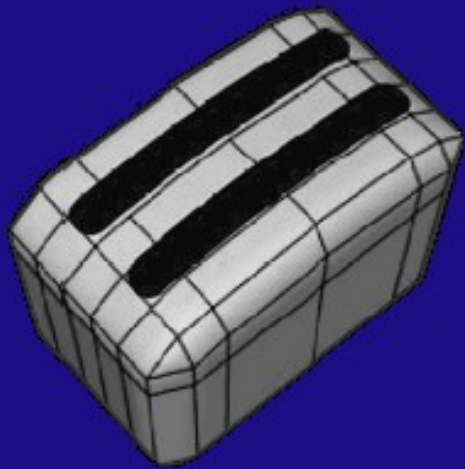


**Deform the Model**



**Render**

# Image Deformation



Render



Deform Image

# Preoblikovanje objektov

- Many objects are not rigid
  - jello
  - mud
  - gases/liquids
  - etc.
- Two main techniques:
  - Geometric deformations
  - Physically-based methods

# Preoblikovanje objektov

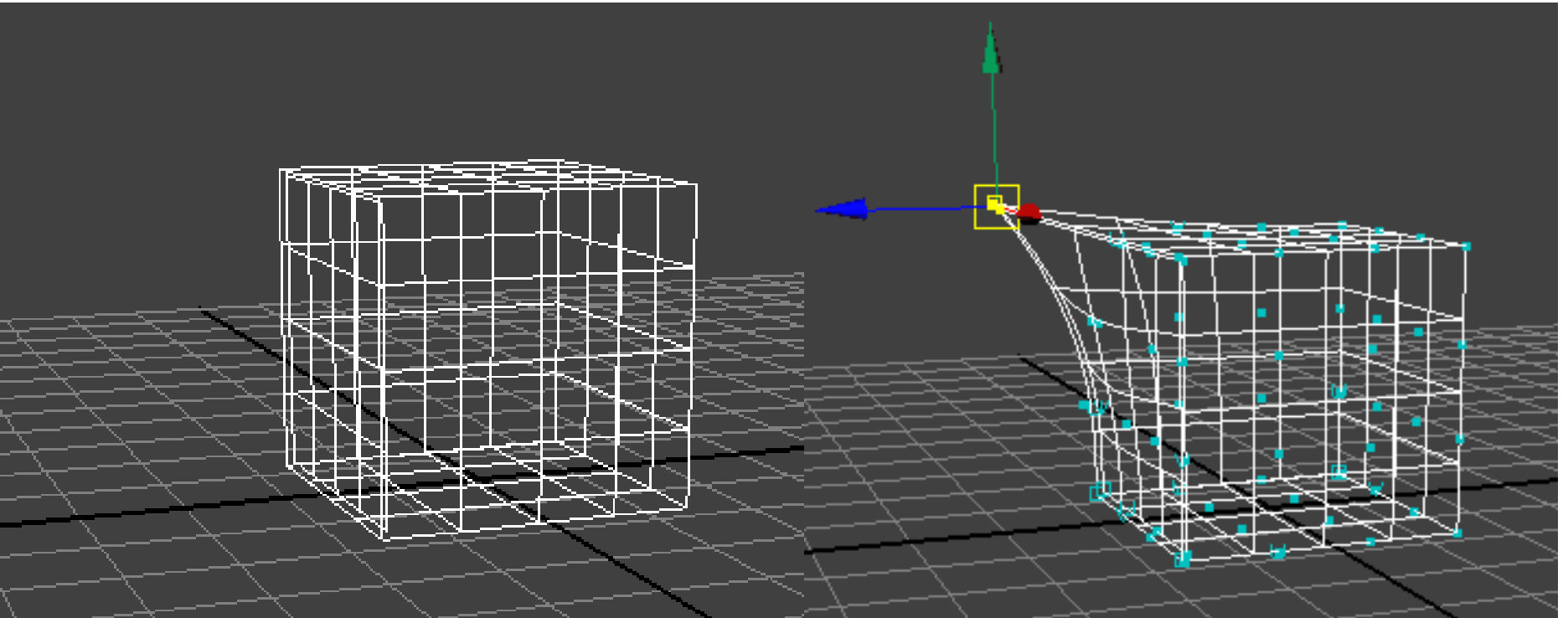
- Assume general geometric objects
- Morphing one object into another
  - If the same number of points, same edge connectivity (simple)
- Simple shape modifications
  - Non-uniform scale
  - Shearing
  - General affine transformation
  - Warping

Demo

# Geometrično preoblikovanje

- Deform the object's geometry directly
- Two main techniques:
  - control point / vertex manipulation
  - space warping

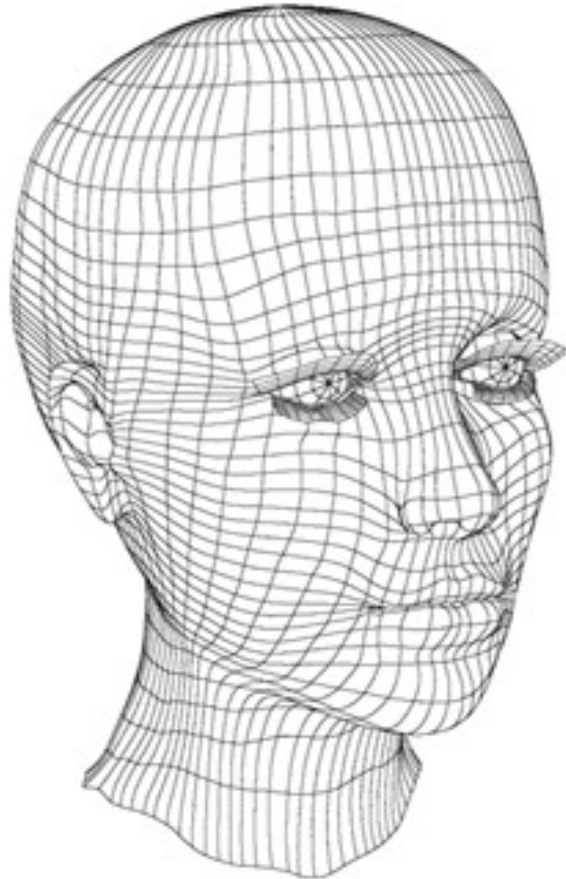
# Krivljenje 3D prostora



Demo



# Rokovanje s kontrolnimi točkami oziroma verteksi

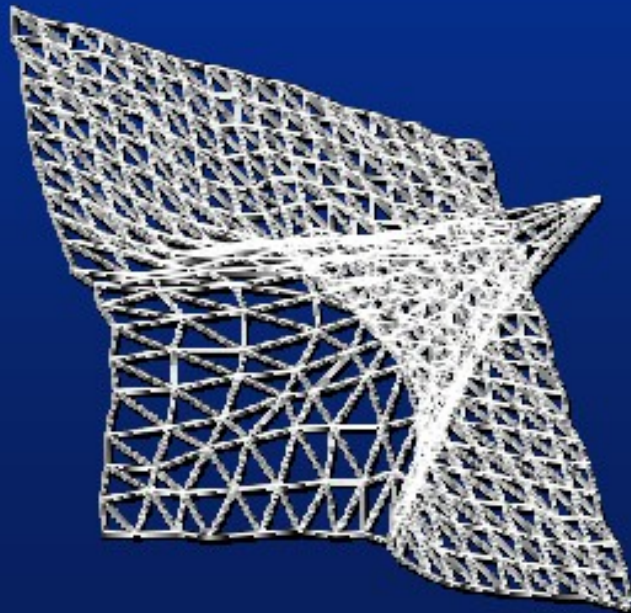


Edit the surface vertices or control points directly

# Mapping Polyhedral Objects

---

- Replace each point  $p$  by  $W(p)$



# Animacija sprememb oblike

- Per-vertex animation curves
  - beginning and end known
- Simply change parameters with time
  - Twist angle
  - Scaling constant
  - Seed vertex position
- All standard rules apply
  - Curve smoothness, motion controls, etc.

# Deformacija koordinatne mreže

- Establish local coordinate system
- Modify it in some way
- Object keeps original coordinates in the warped system
- Example: 2D grid deformation
  - Apply rectangular grid
    - Compute bilinear coefficients
  - Deform grid points
  - Use bilinear interpolation of new grid points with old coefficients

# Krivljenje prostora (Space Warping)

Deform the object by deforming the space it is in

## **Two main techniques:**

- Nonlinear Deformation
- Free Form Deformation (FFD)

**Independent of object representation**

# Nelinearno globalno preoblikovanje

Objects are defined in a local object space

Deform this space using a combination of:

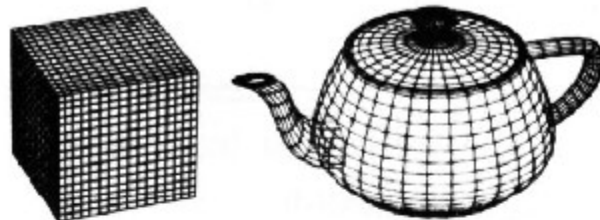
- Non-uniform Scaling
- Tapering
- Twisting
- Bending

Combine these to produce complex shapes

Can produce quite interesting results

# Nelinearno globalno preoblikovanje

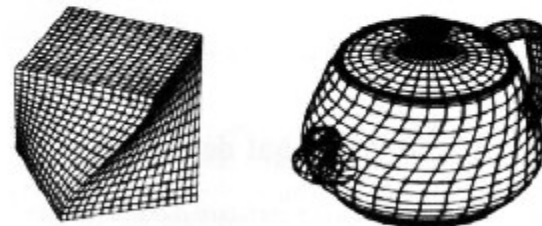
- original



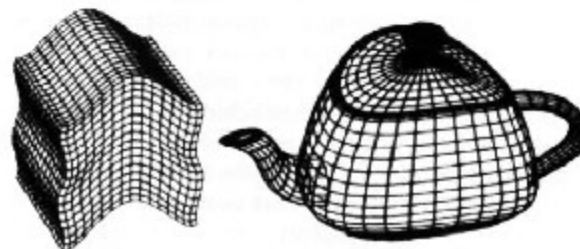
- tapering



- twisting

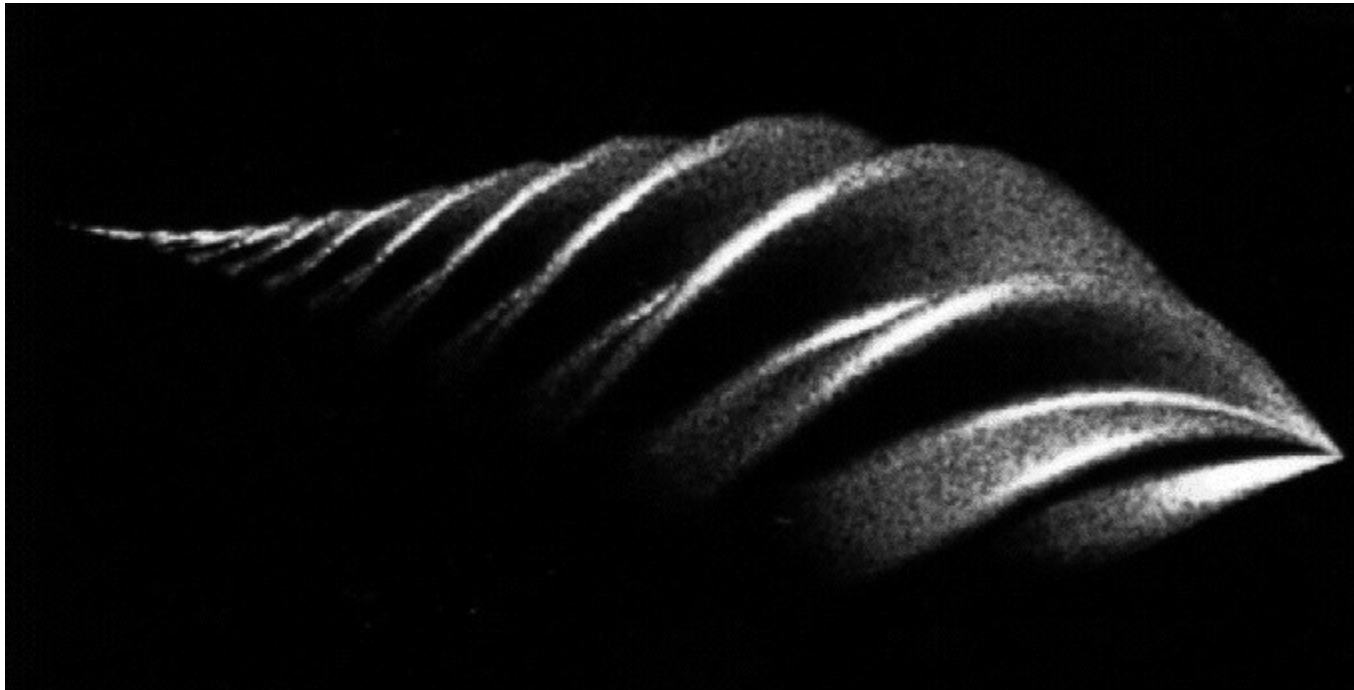


- bending



# Nelinearno globalno preoblikovanje

Good for modeling [Barr 87]

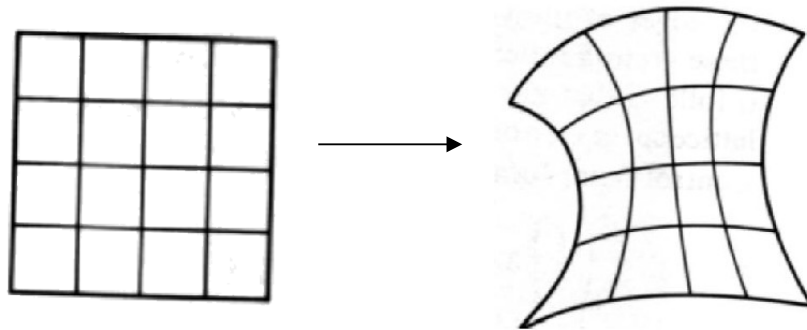


Animation is harder



# Free Form Deformation (FFD)

Deform space by deforming a lattice around an object

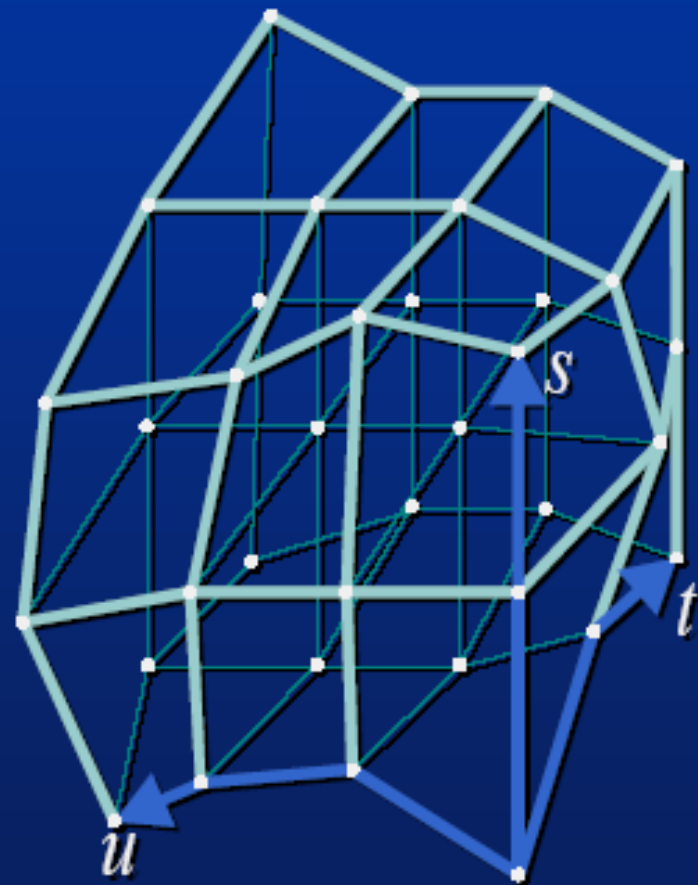
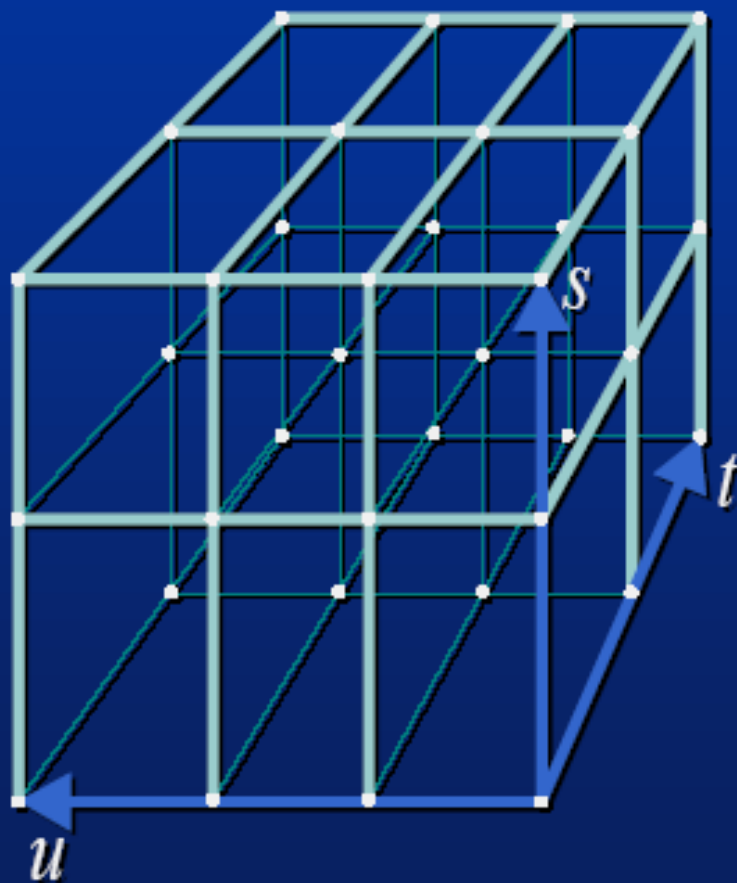


The deformation is defined by moving the control points

Imagine it as if the object were encased in rubber

# Free-form Deformation - FFD

---



# Free Form Deformation (FFD)

- Local coord system: (S,T,U)
- Point P coordinate along S:
- Same for T,U  $s = (T \times U)(P - P_0) / ((T \times U)S)$
- Algorithm:
  - Introduce fine grid
  - Deform grid points
  - Use Bezier interpolation to get new position
    - Treat new grid points as control points

# Bezier solids

- Bezier curves: 1D entity
- Bezier patches: 2D
  - Have a surface connectivity control mesh
  - Repeat DeCasteljau twice, get a point
- Bezier solids (3D): Same, but three times
- Each  $(s,t,u)$  point can be written through control points
  - Cubic polynomial in  $s, t, u$  (for cubic Bezier)
- New position:
  - same expression, same  $s,t,u$
  - new c.p. positions

# Free Form Deformation (FFD)

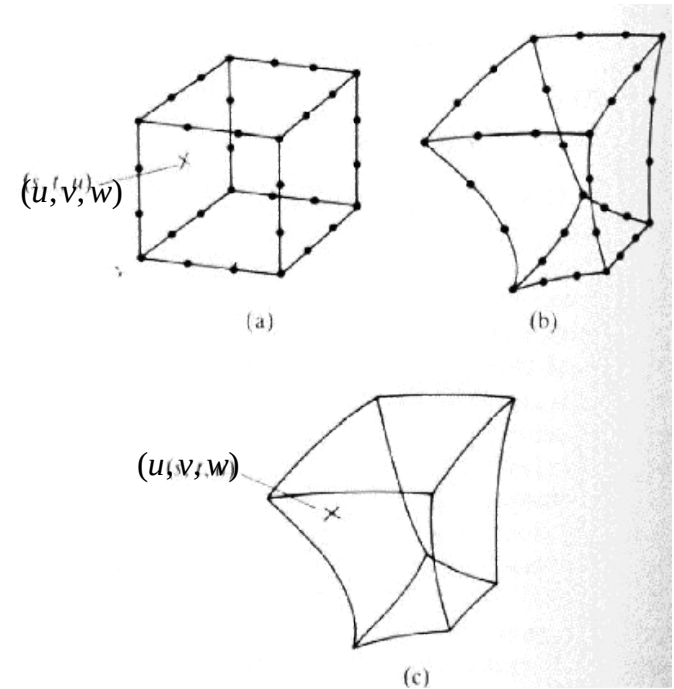
The lattice defines a Bezier volume

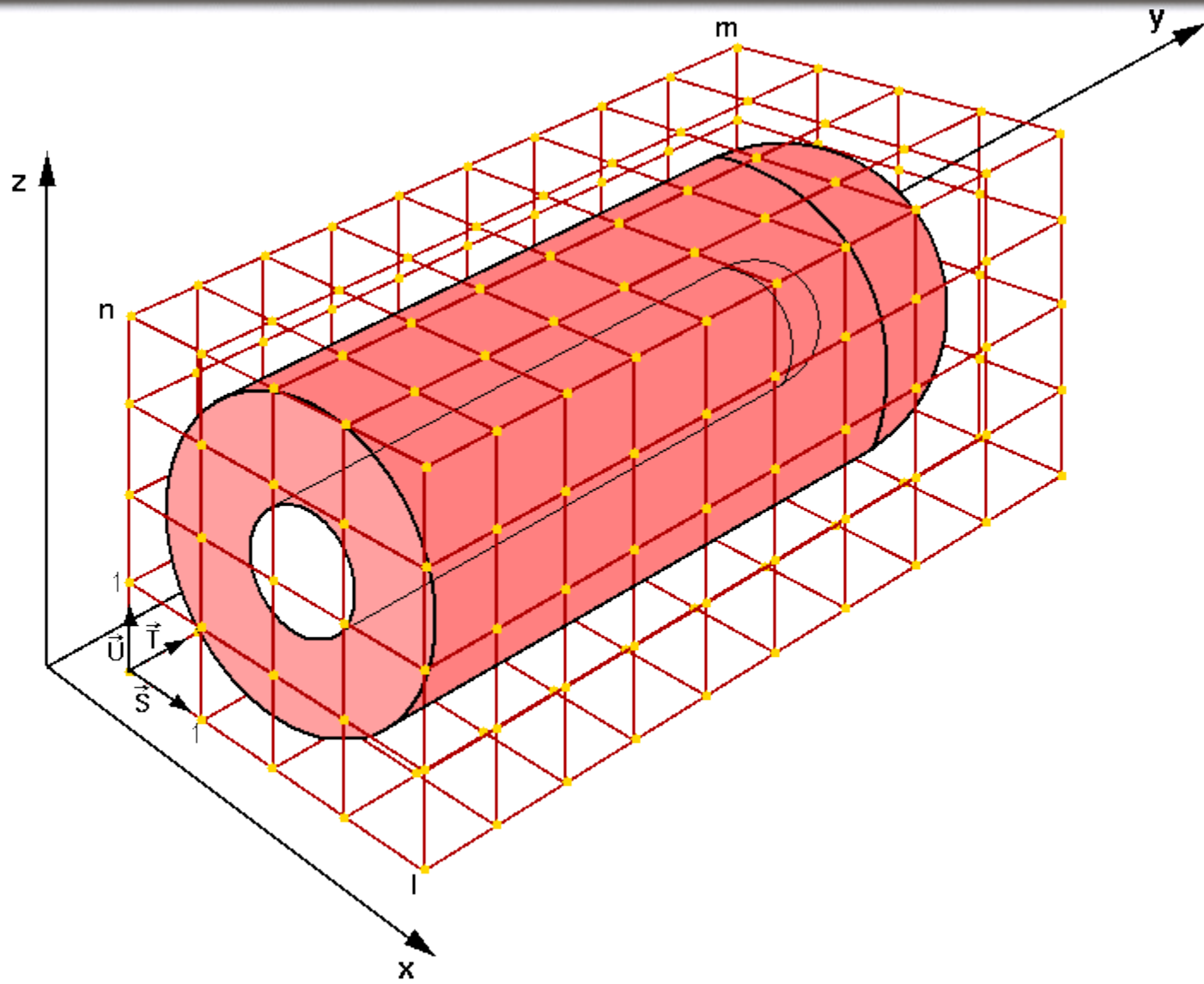
$$\mathbf{Q}(u, v, w) = \sum_{ijk} \mathbf{P}_{ijk} B(u)B(v)B(w)$$

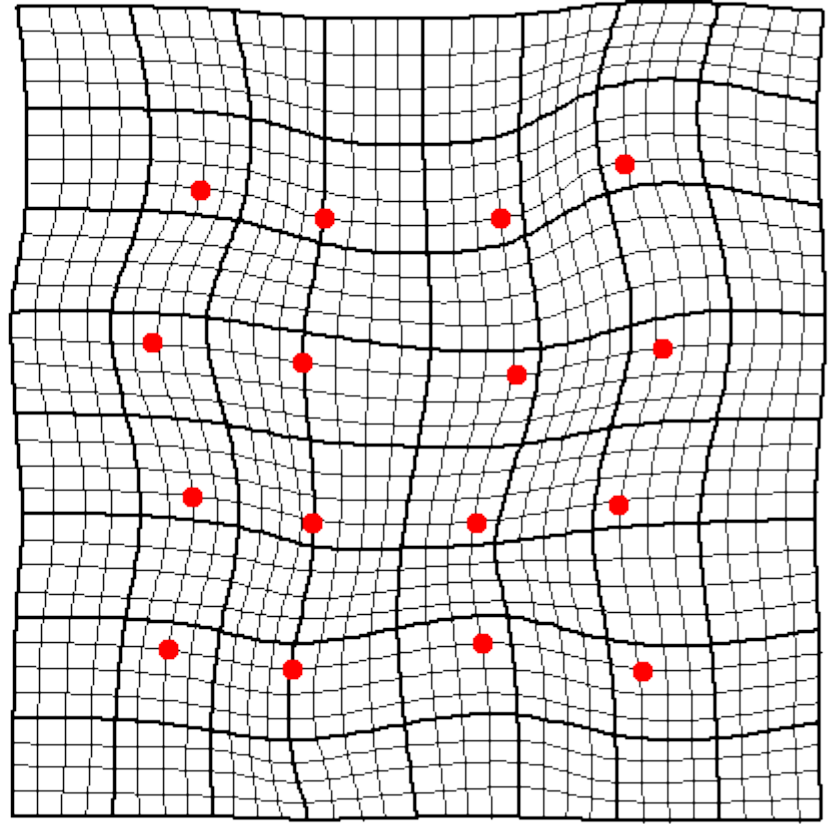
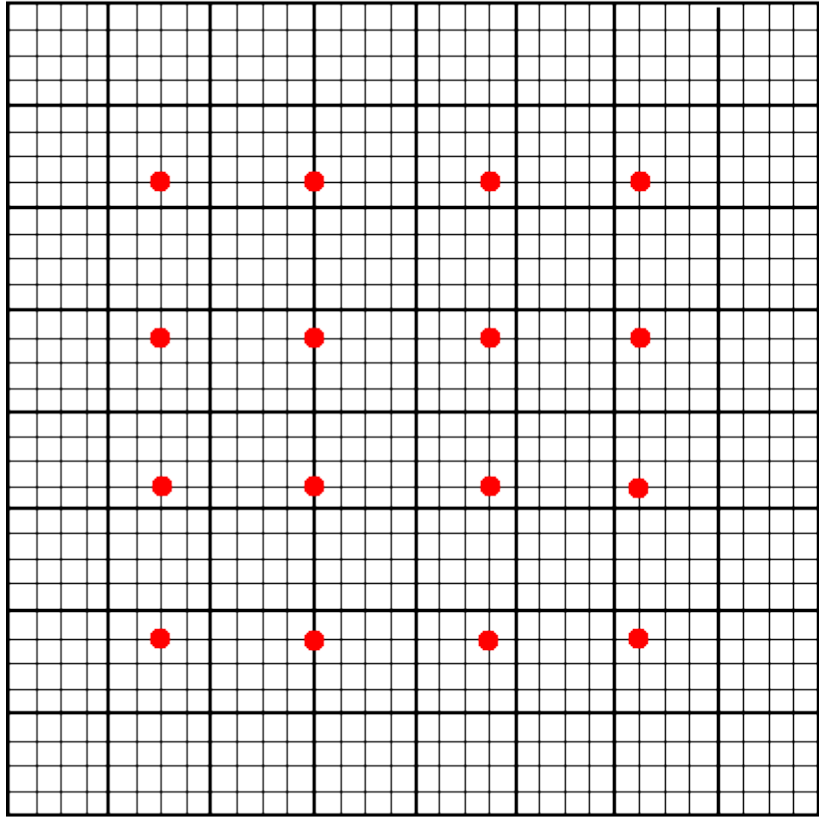
Compute lattice coordinates  
 $(u, v, w)$

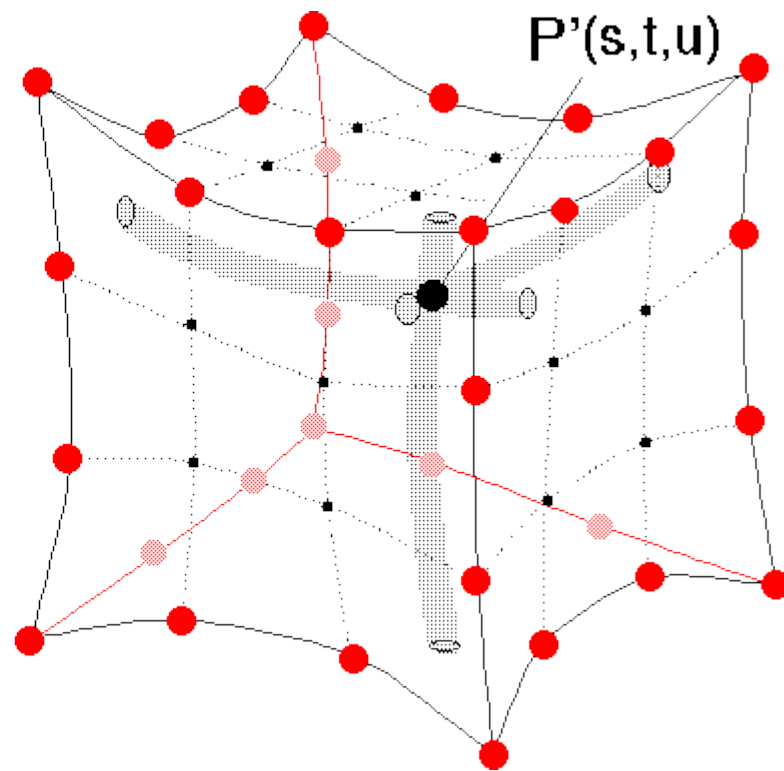
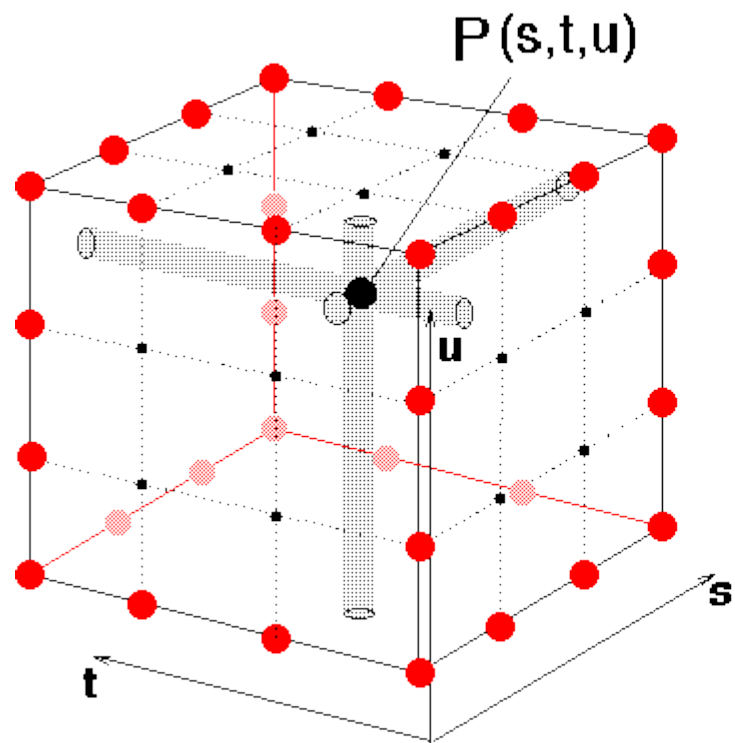
Alter the control points  
 $\mathbf{P}_{ijk}$

Compute the deformed points  
 $\mathbf{Q}(u, v, w)$

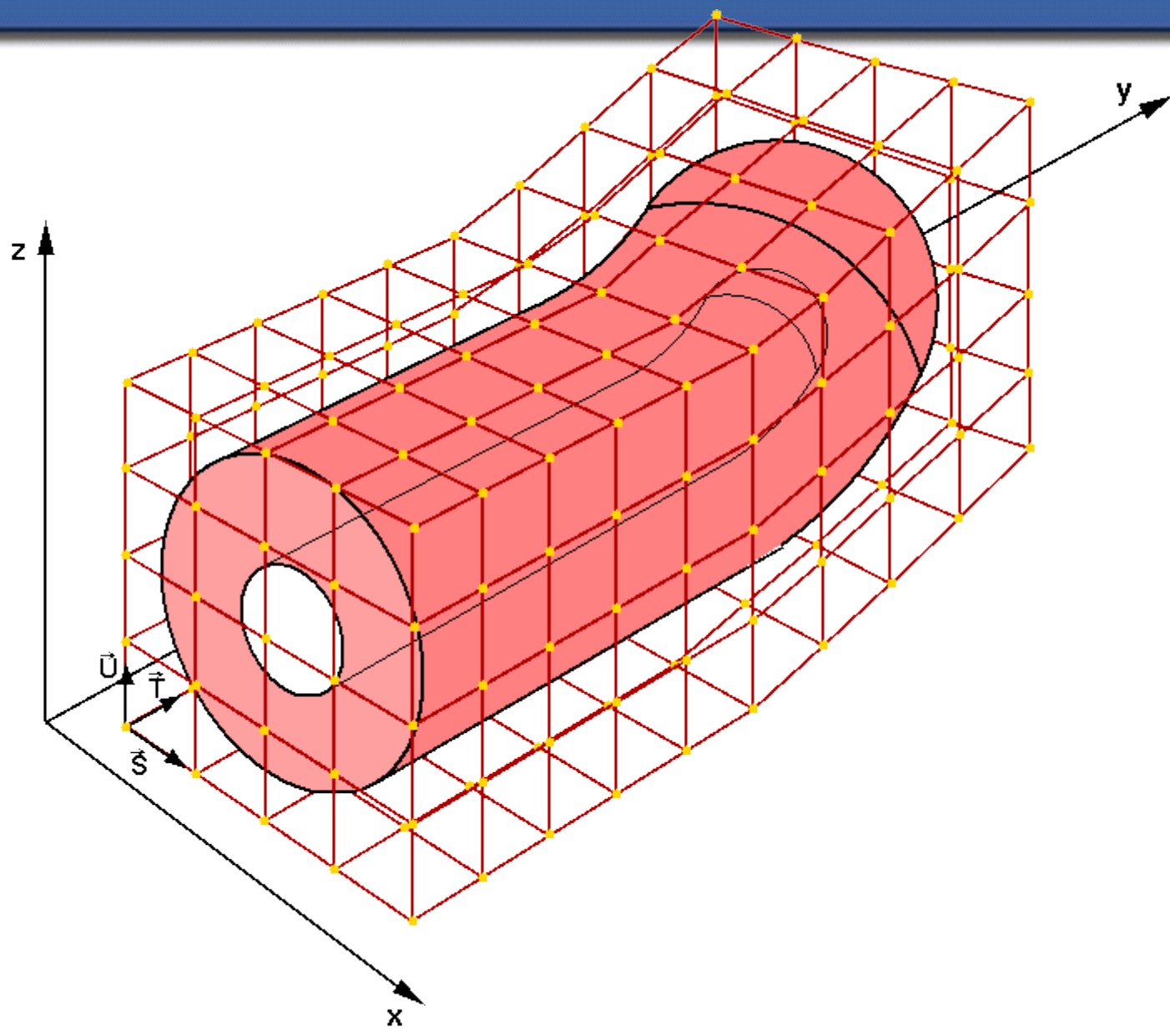


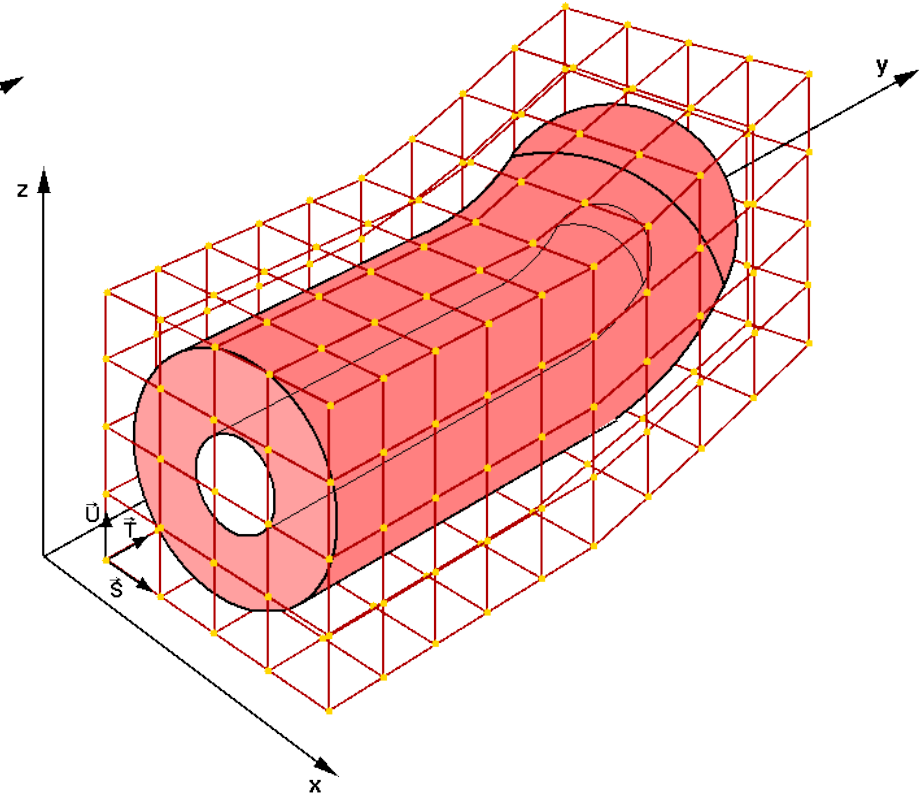
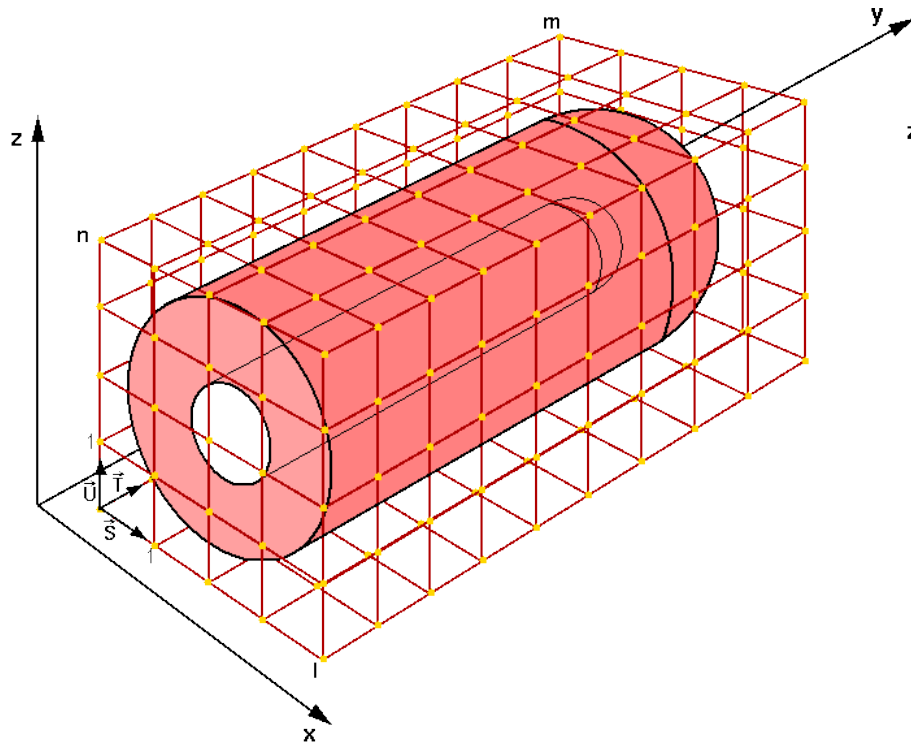




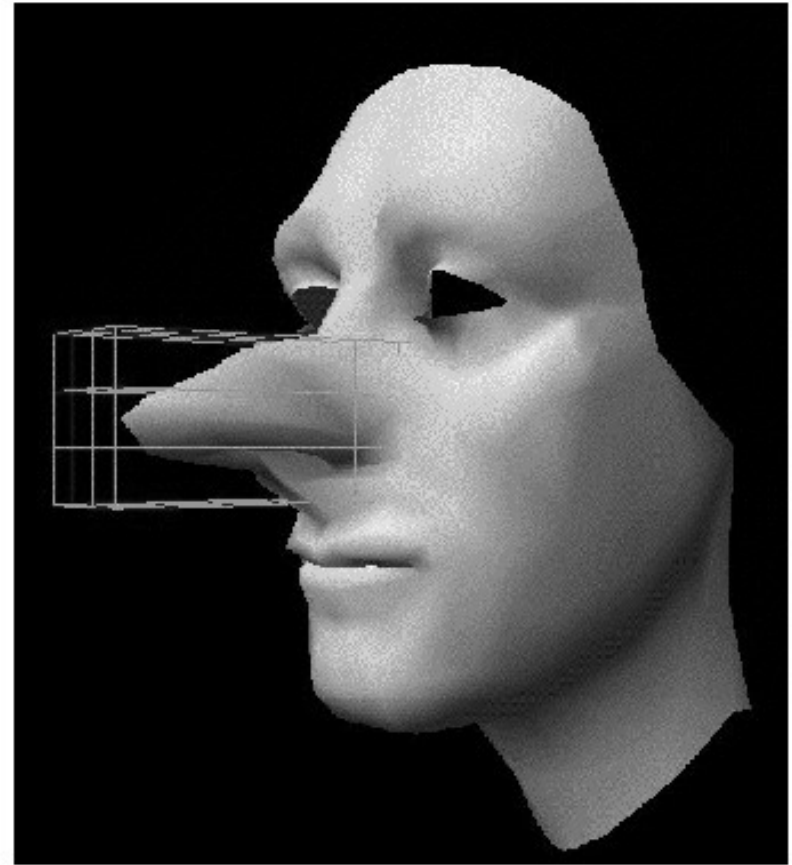
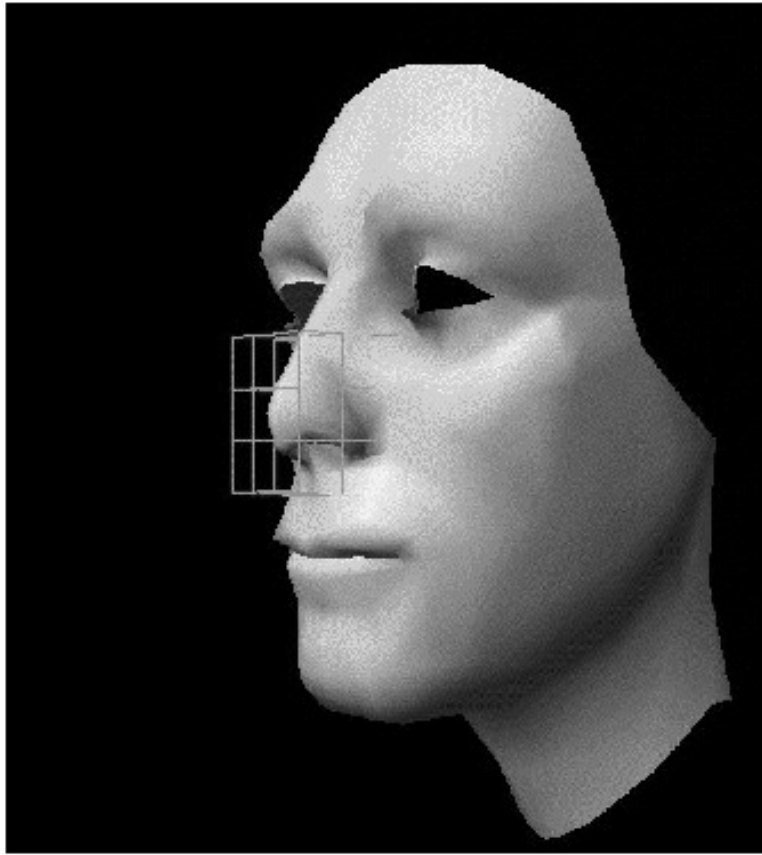




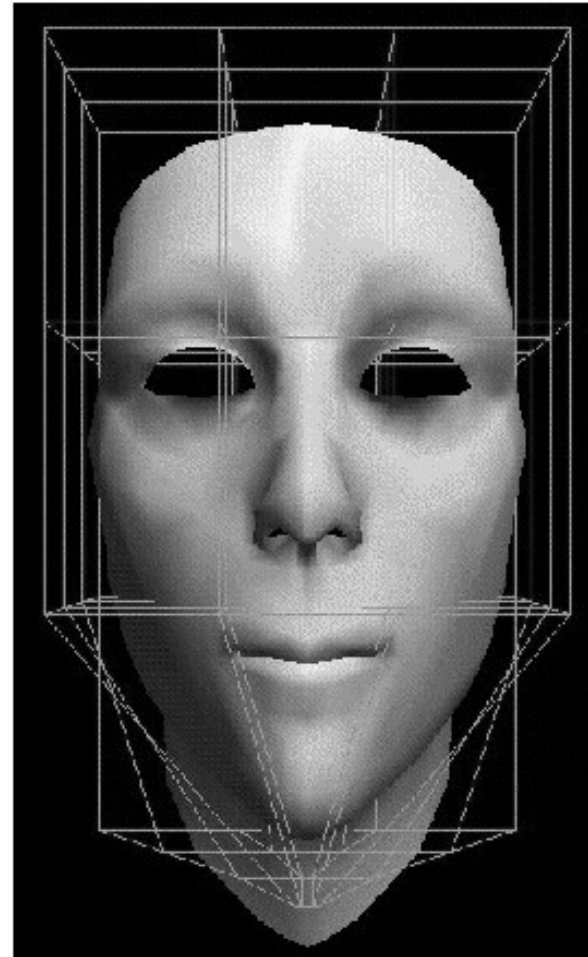
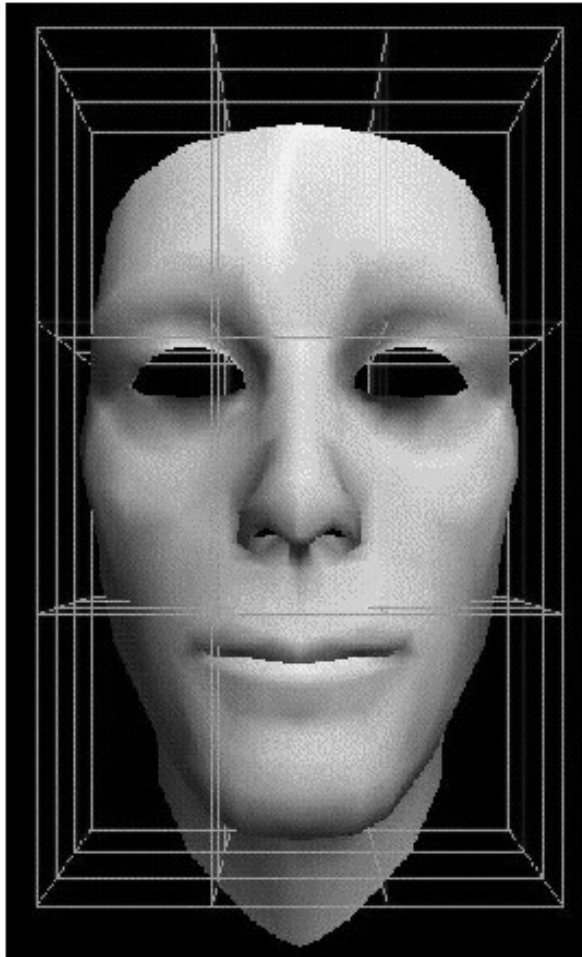




# FFD Example



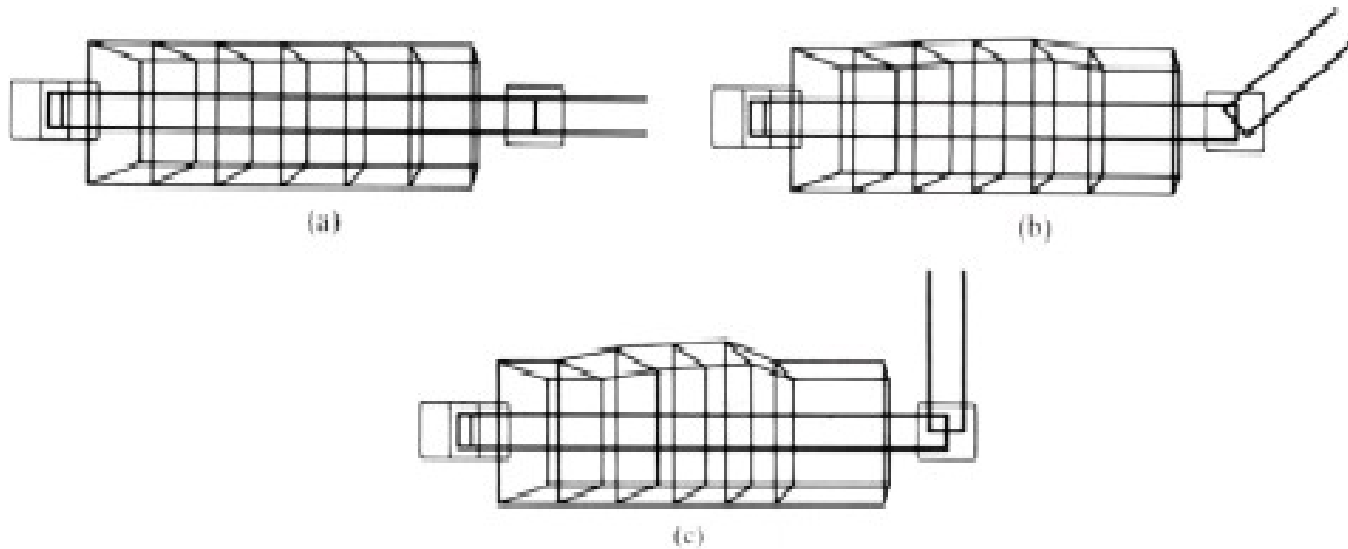
# FFD Example



# Animacija s FFD

- **Hierarchical FFD**
  - Coarse level FFD modifies vertices and finer FFD grids
- **Moving object through deformation tool**
  - Can move the tool itself
- **Modifying control points of FFD**
  - Any technique applies
    - Key frame
    - Physics based
- **Example:**
  - FFD is relative to wire skeleton
  - Moves of skeleton re-position FFD grid
  - Skin position is computed within new FFD

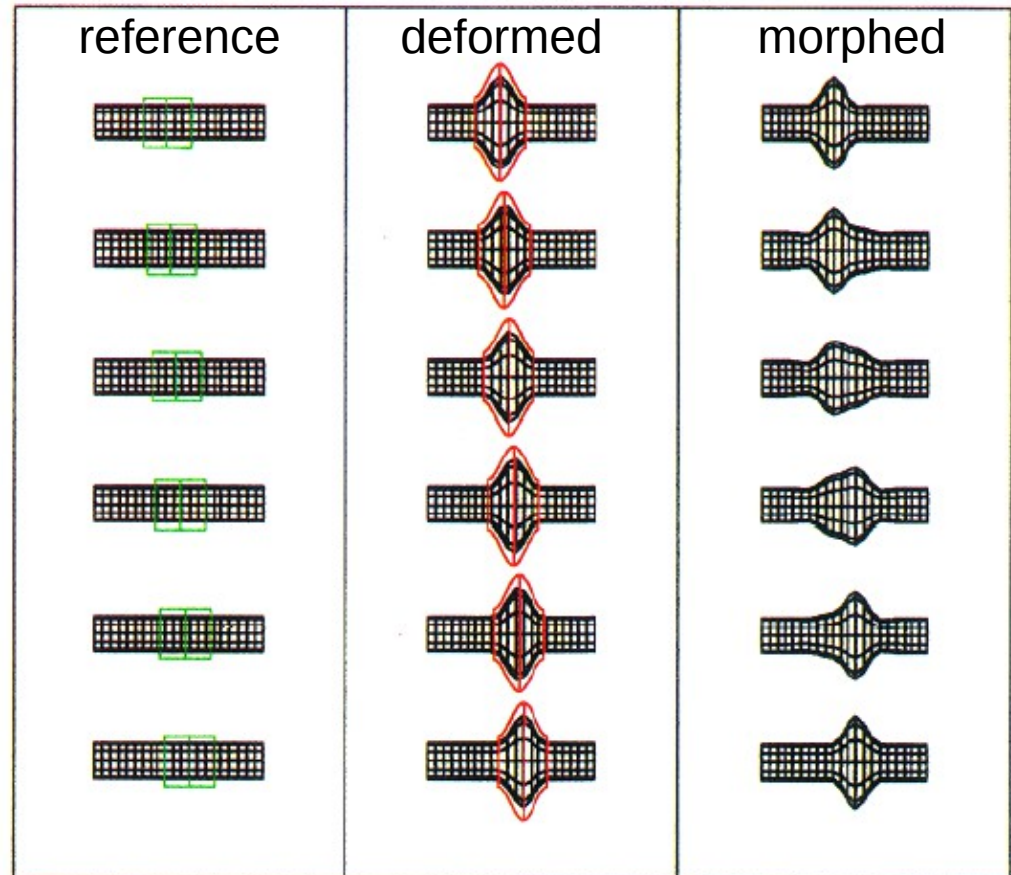
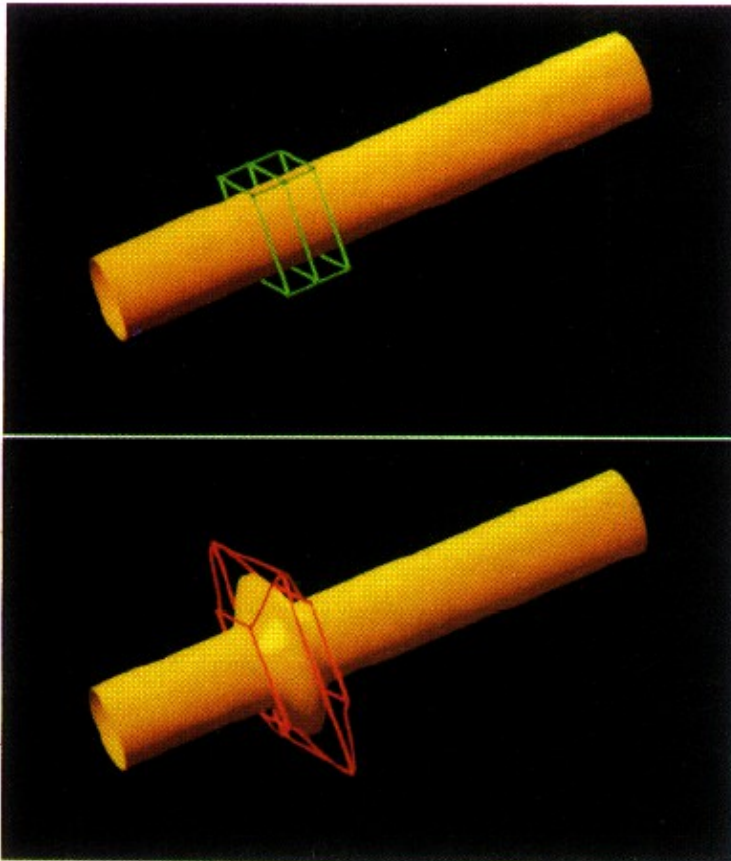
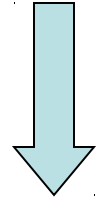
# Ad-hoc muscle



# FFD Animation

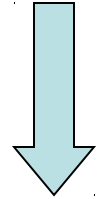
(next lecture!)

Animate a reference and a deformed lattice

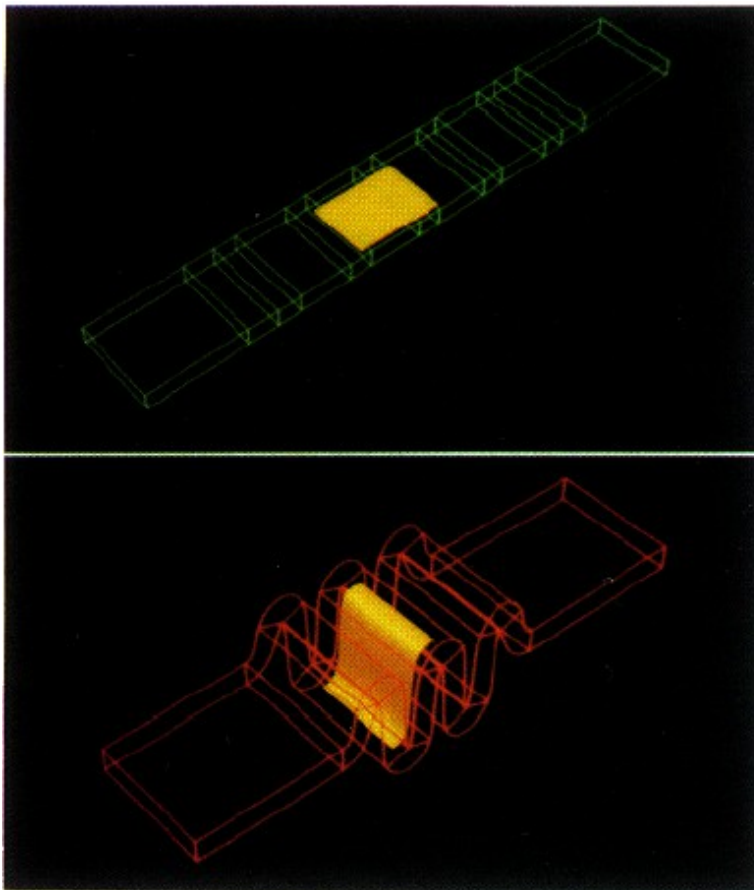


# FFD Animation

(next lecture!)

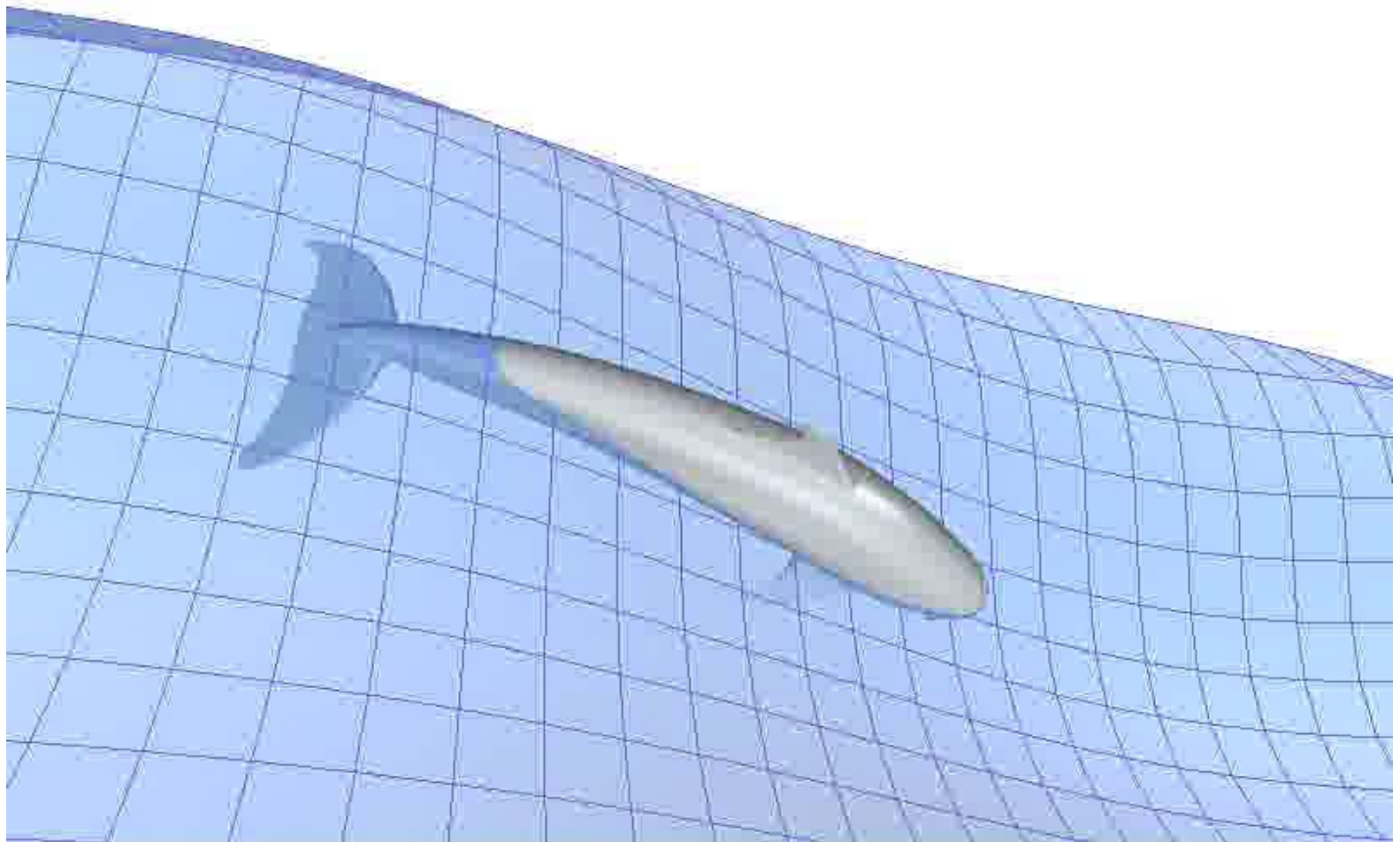


Animate the object through the lattice



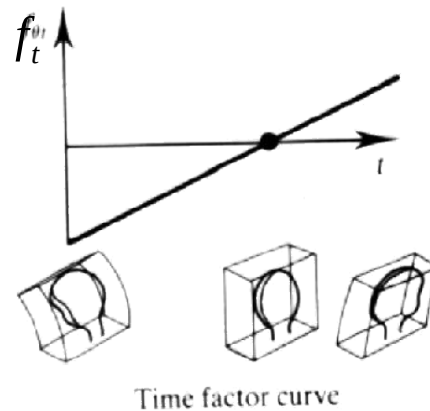
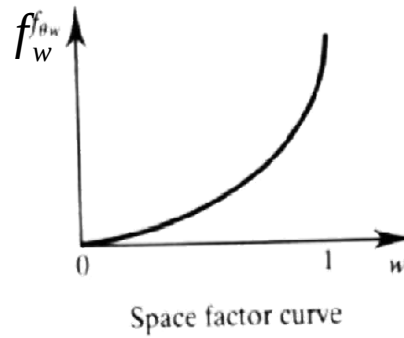
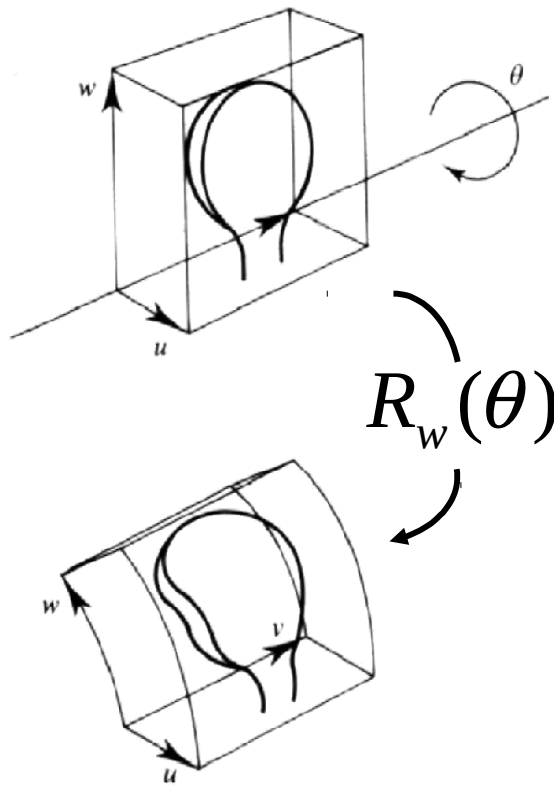
reference	deformed	morphed





# Factor Curves

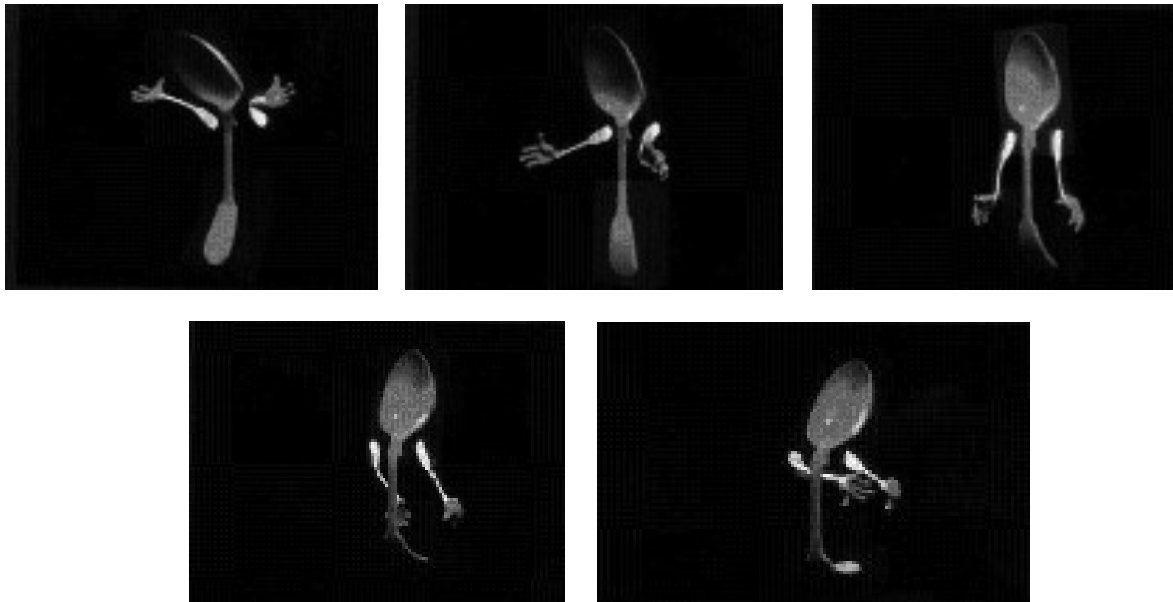
Modify the transformation applied to the object based on where and when it is applied



$$\theta = \theta_0 f_w(w) f_t(t)$$

# Factor Curves

Scripted animation can lead to complex motions  
(depending on animator skill)



Deformations can be nested

# Krivljenje, temelječe na fiziki

## ■ Deformation derived from physical models:

- ◆ Springs
- ◆ Fluids
- ◆ Cloth
- ◆ etc.

