Computer Graphics

Freeform Surfaces II

Mark Pauly

Geometric Computing Laboratory

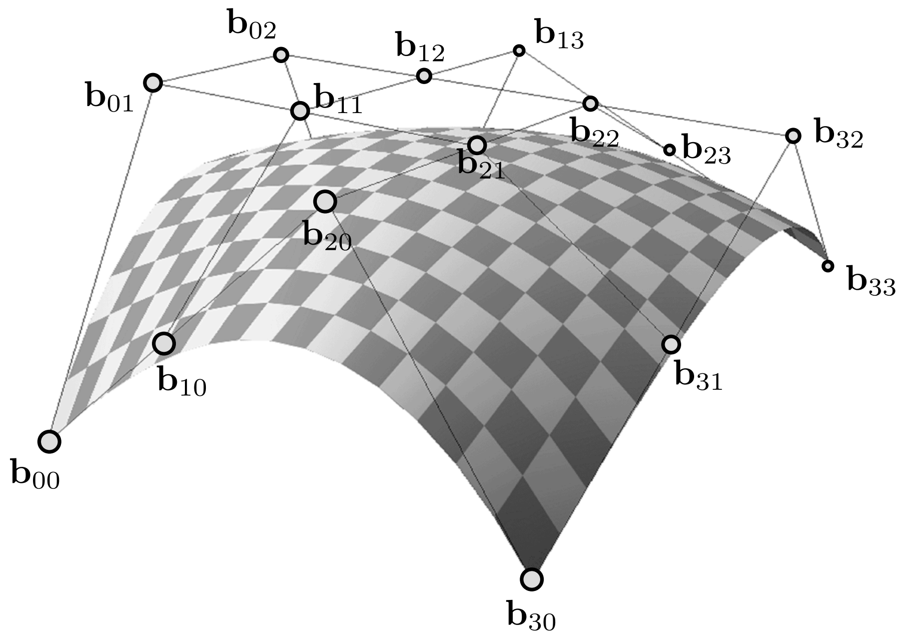
Tensor Product Bezier Surfaces

- Let a Bezier curve be given as
  $\mathbf{x}(u) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(u)$
- Control points $\mathbf{b}_i$ move on Bezier curves $\mathbf{b}_i(v)$

$$\mathbf{b}_i(v) = \sum_{j=0}^{n} \mathbf{b}_{i,j} B_j^n(v)$$

- This defines a tensor product Bezier patch

$$\mathbf{x}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{b}_{i,j} B_i^m(u) B_j^n(v)$$



Derivatives and Normals

- Partial derivatives in u and v

$$\frac{\partial}{\partial u}\mathbf{x}(u,v) = m \cdot \sum_{i=0}^{m-1}\sum_{j=0}^{n}(\mathbf{b}_{i+1,j} - \mathbf{b}_{i,j})B_i^{m-1}(u)B_j^n(v)$$

$$\frac{\partial}{\partial v}\mathbf{x}(u,v) = n \cdot \sum_{i=0}^{m}\sum_{j=0}^{n-1}(\mathbf{b}_{i,j+1} - \mathbf{b}_{i,j})B_i^{m}(u)B_j^{n-1}(v)$$

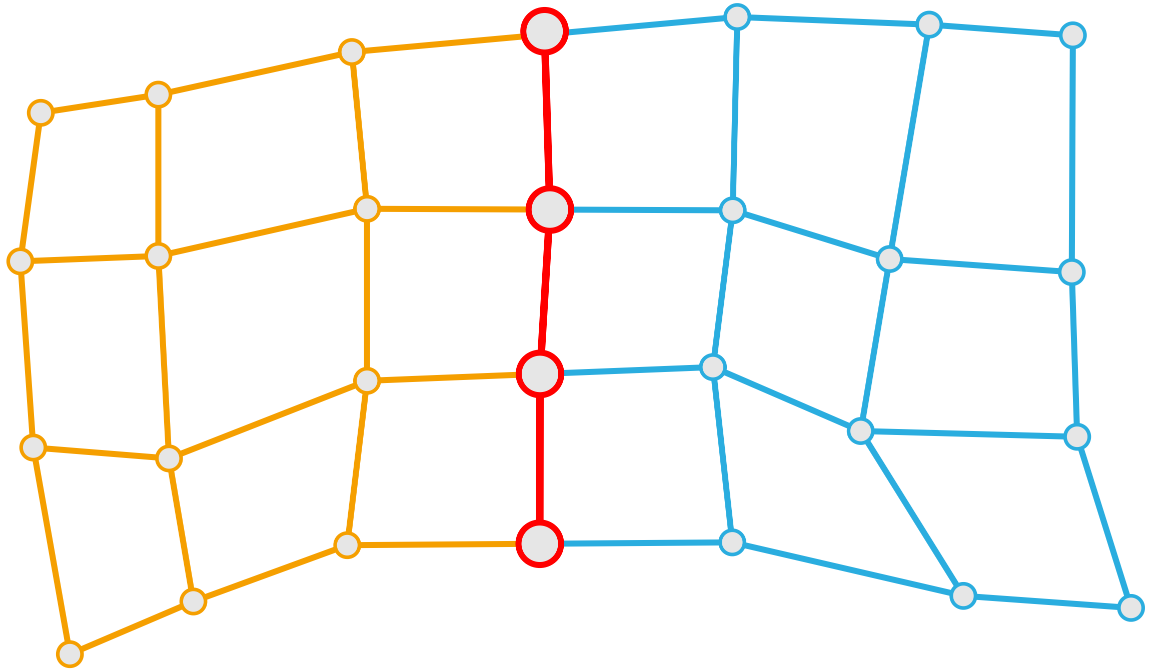
- Normal vector as cross-product of tangents

$$\mathbf{n}(u,v) = \frac{\frac{\partial}{\partial u}\mathbf{x}(u,v) \times \frac{\partial}{\partial v}\mathbf{x}(u,v)}{\left\| \frac{\partial}{\partial u}\mathbf{x}(u,v) \times \frac{\partial}{\partial v}\mathbf{x}(u,v) \right\|}$$

Try it yourself!

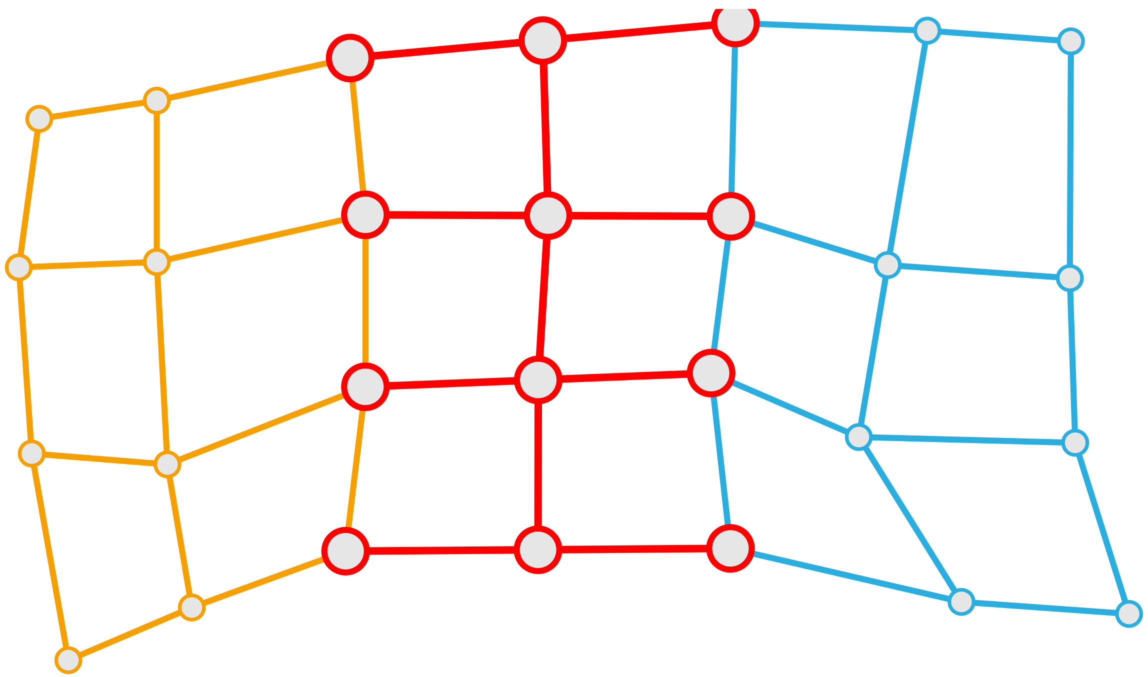
Try it yourself!

Piecewise Bezier Surfaces



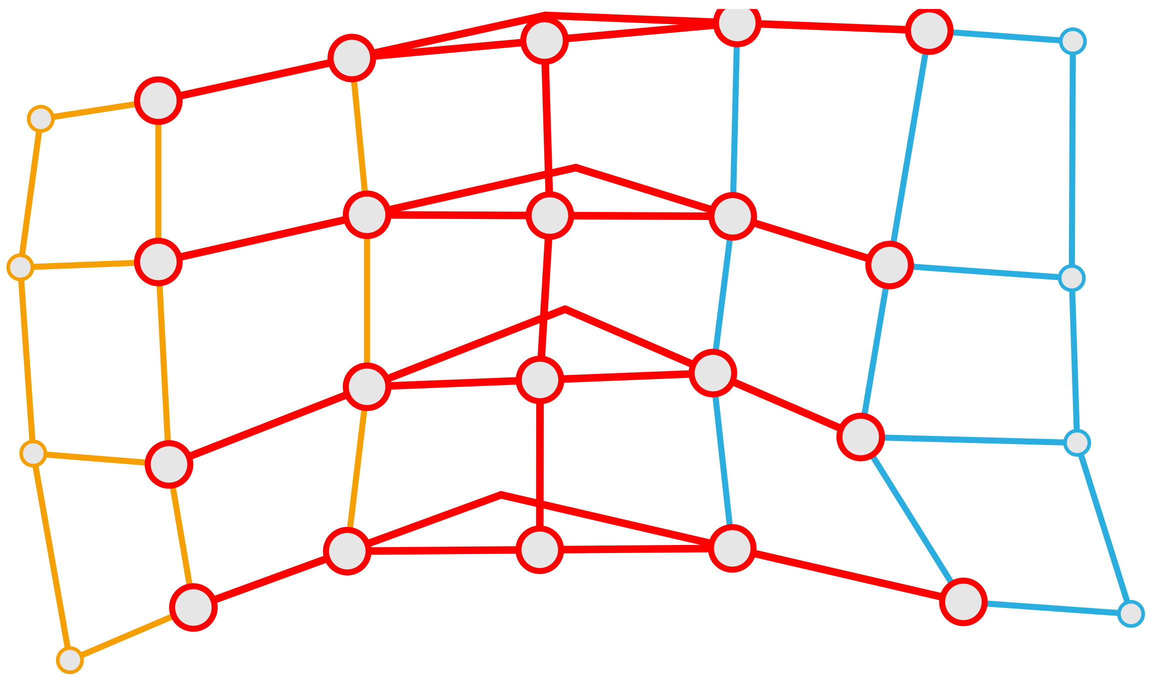
C^0 continuity: Boundary curves

Piecewise Bezier Surfaces



C^1 *continuity: Collinearity*

Piecewise Bezier Surfaces

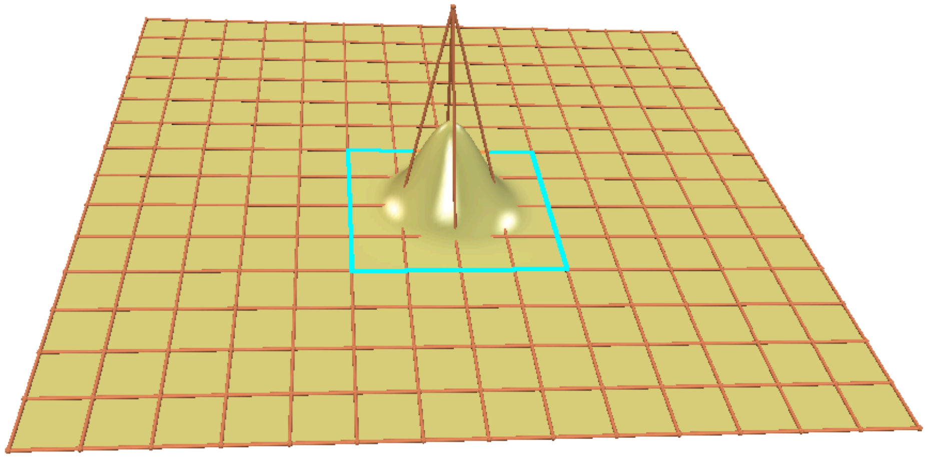


C^2 continuity: A-frames

B-Spline Surfaces

- Tensor product B-spline surfaces

$$\mathbf{x}(u,v) = \sum_{i=0}^{L} \sum_{j=0}^{K} \mathbf{d}_{i,j} N_i^m(u) N_j^n(v)$$



- Properties
  - Affine combination
  - Convex hull property
  - Local control
  - Maximal smoothness C^{n-1}
  - ~~Corner interpolation~~
  - ~~Boundary curve interpolation~~

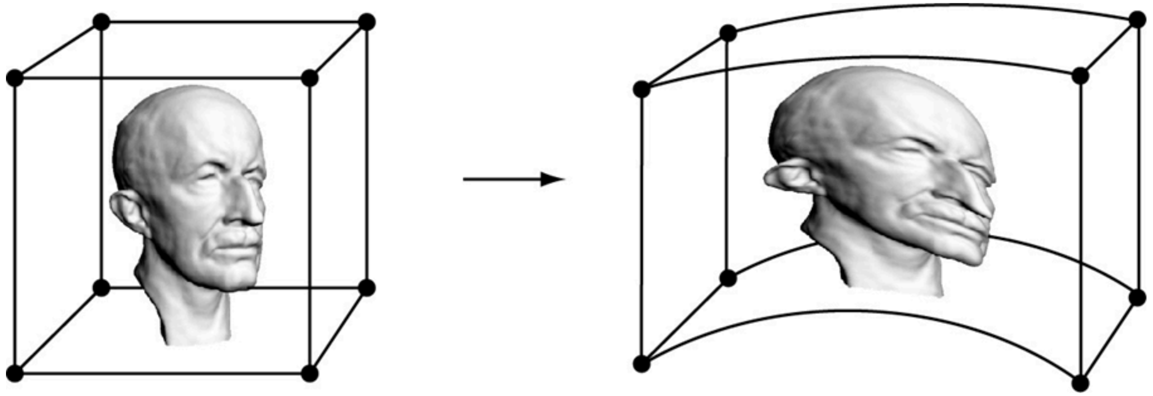
Freeform Deformation

Freeform Deformation

- Deformation of arbitrary geometric objects
  - Polygonal meshes
  - Bezier / spline curves & surfaces
  - Point-sampled objects
- How to deal with different representations?
  $\Rightarrow$ Deform embedding space around objects

Freeform Deformation

- Space warp function $\mathbf{d}: \mathbb{R}^3 \to \mathbb{R}^3$
  - Deform each 3D point: $\mathbf{p} \mapsto \mathbf{d}(\mathbf{p})$
  - Deform each mesh vertex
  - Deform control points of freeform curves/surfaces

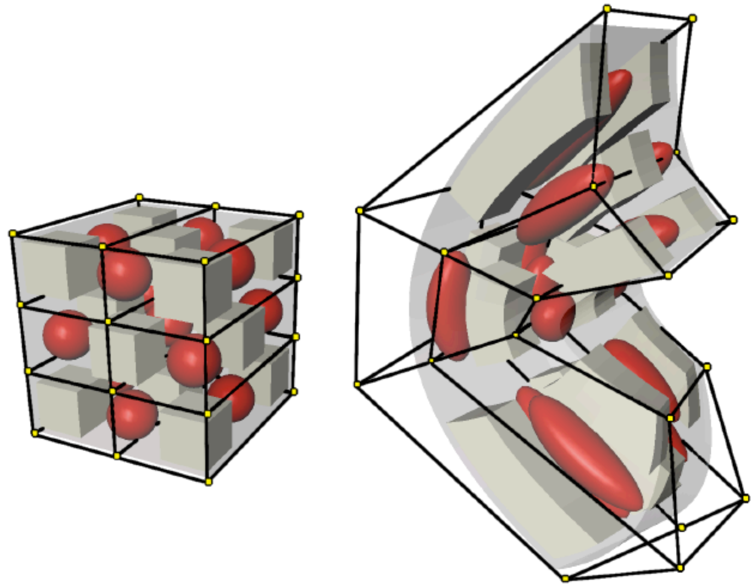


How to represent deformation function?

Freeform Deformation

- Use Bezier or spline framework!
  - Manipulate through control points
  - Local / global deformations
  - Continuity of deformations
- Tensor product Bezier deformation

$$\mathbf{d}(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \mathbf{b}_{i,j,k} B_i^l(u) B_j^m(v) B_k^n(w)$$



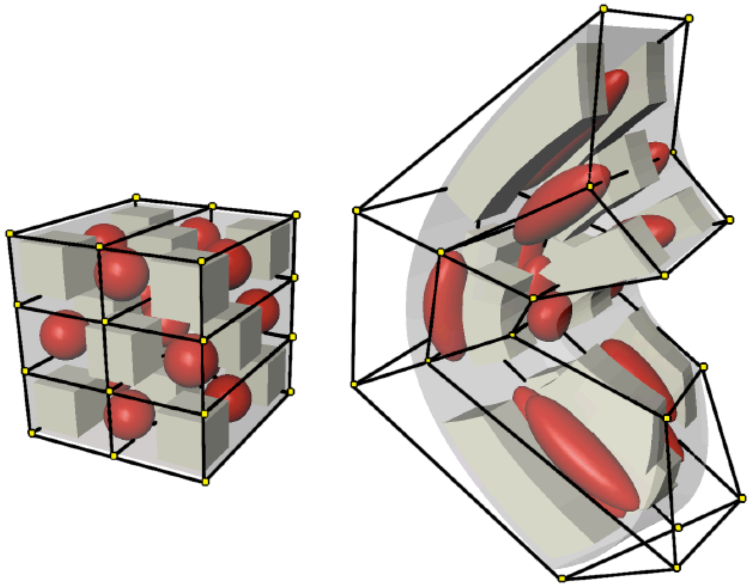
Sederberg & Perry, SIGGRAPH 1986

Freeform Deformation

- Setting up a *regular* grid of control points

$$\mathbf{b}_{i,j,k} = \mathbf{o} + \frac{i}{l}\mathbf{u} + \frac{j}{m}\mathbf{v} + \frac{k}{n}\mathbf{w}$$

initializes function $\mathbf{d}$ to be the identity

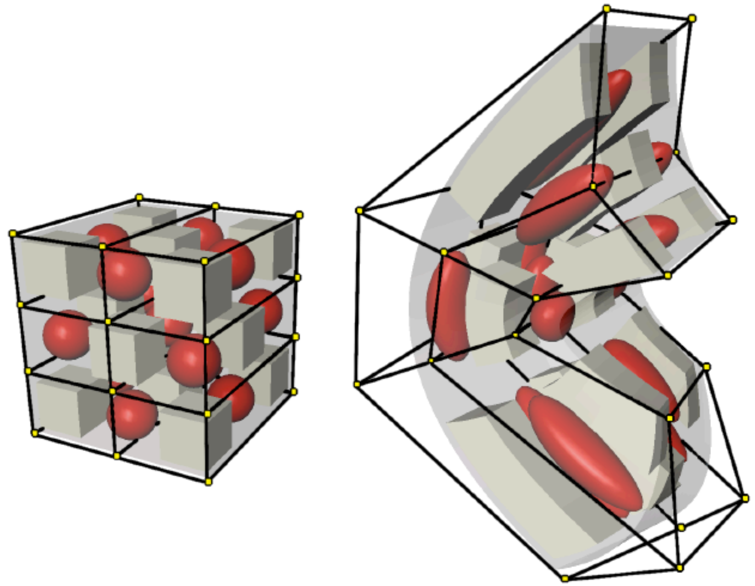


Sederberg & Perry, SIGGRAPH 1986

Freeform Deformation

- Precompute local coordinates (u, v, w) for each object point $\mathbf{x}$, such that

$$\mathbf{x} = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \mathbf{b}_{i,j,k} B_i^l(u) B_j^m(v) B_k^n(w)$$



- For Bezier deformation, this yields

$$u = \frac{\mathbf{v} \times \mathbf{w} \cdot (\mathbf{x} - \mathbf{o})}{\mathbf{v} \times \mathbf{w} \cdot \mathbf{u}}$$

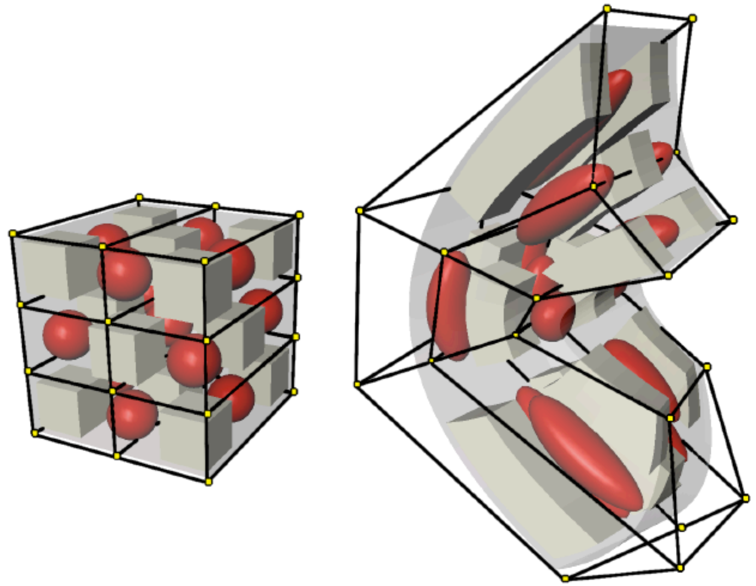
$$v = \frac{\mathbf{w} \times \mathbf{u} \cdot (\mathbf{x} - \mathbf{o})}{\mathbf{w} \times \mathbf{u} \cdot \mathbf{v}}$$

$$w = \frac{\mathbf{u} \times \mathbf{v} \cdot (\mathbf{x} - \mathbf{o})}{\mathbf{u} \times \mathbf{v} \cdot \mathbf{w}}$$

Sederberg & Perry, SIGGRAPH 1986

Freeform Deformation

- Deform object by dragging control points
- Compute deformed position $\mathbf{d}(\mathbf{x})$ for all points
    1. Use precomputed local coordinates: $\mathbf{x} \mapsto (u, v, w)$
    2. Evaluate deformation: $(u, v, w) \mapsto \mathbf{d}(u, v, w)$

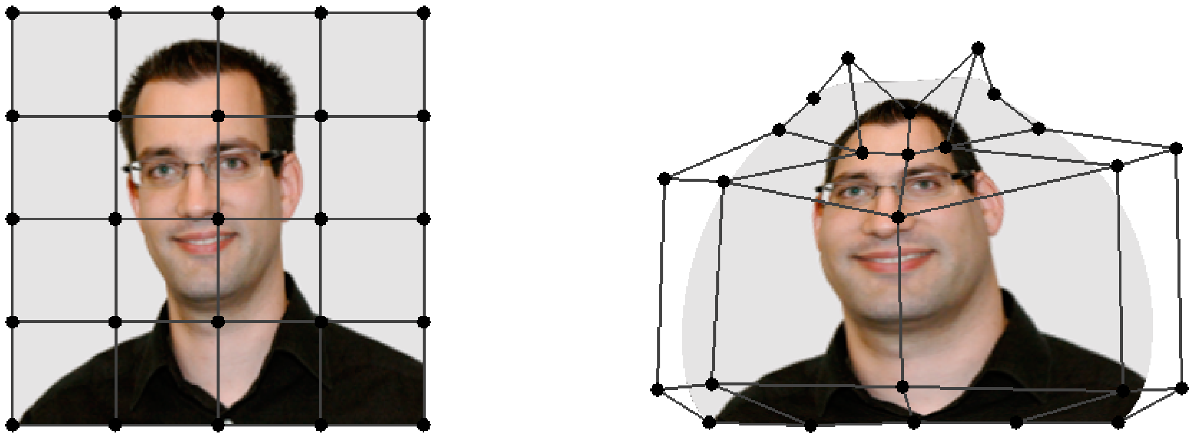


Sederberg & Perry, SIGGRAPH 1986

Freeform Deformation

2D freeform deformation for image manipulation

$$\mathbf{d}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{b}_{i,j} B_i^m(u) B_j^n(v)$$



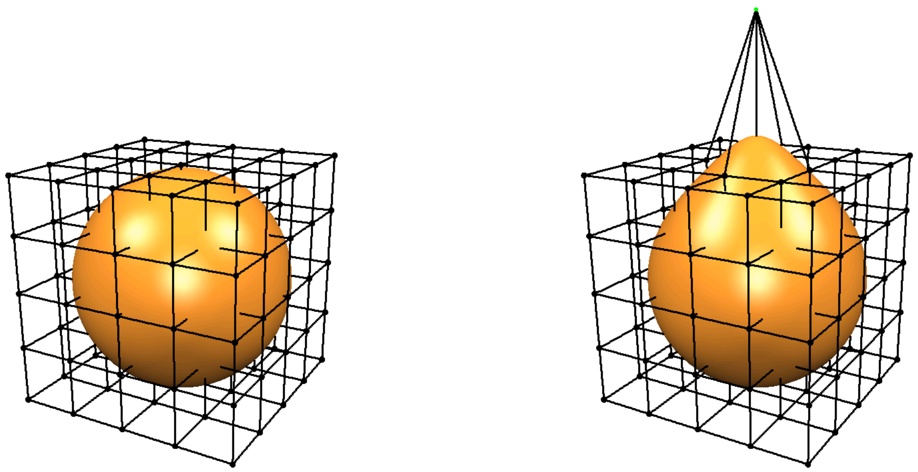
Let’s try: Image Freeform Deformation

Freeform Deformation

- Tensor product *spline* deformations allow for *local control*

$$\mathbf{d}(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n}\mathbf{d}_{i,j,k}N_i^l(u)N_j^m(v)N_k^n(w)$$

  - Computing local coordinates (u,v,w) is much harder!

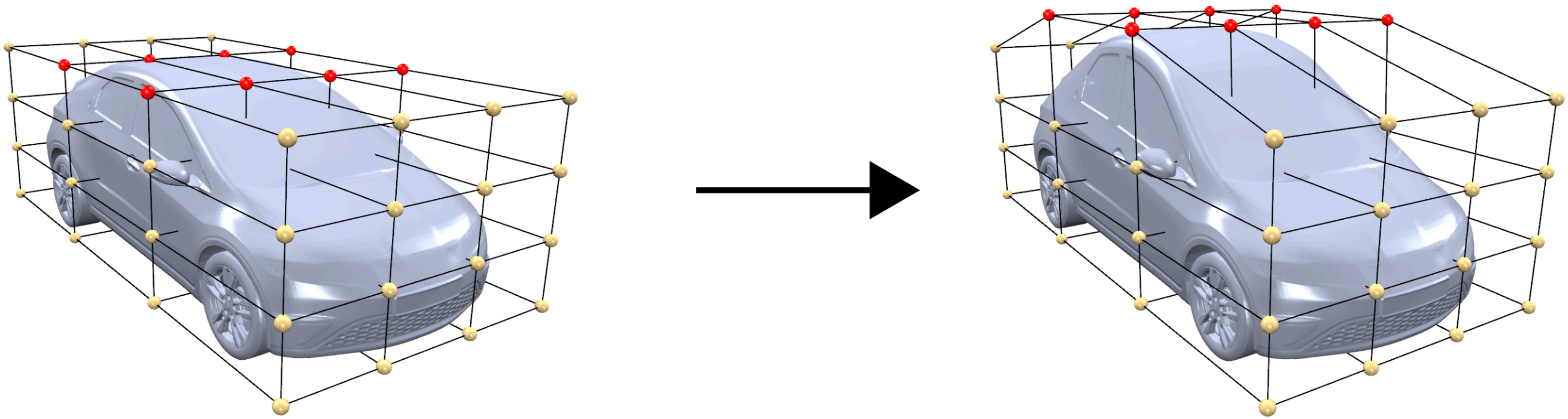


Sederberg & Perry, SIGGRAPH 1986

Freeform Deformation

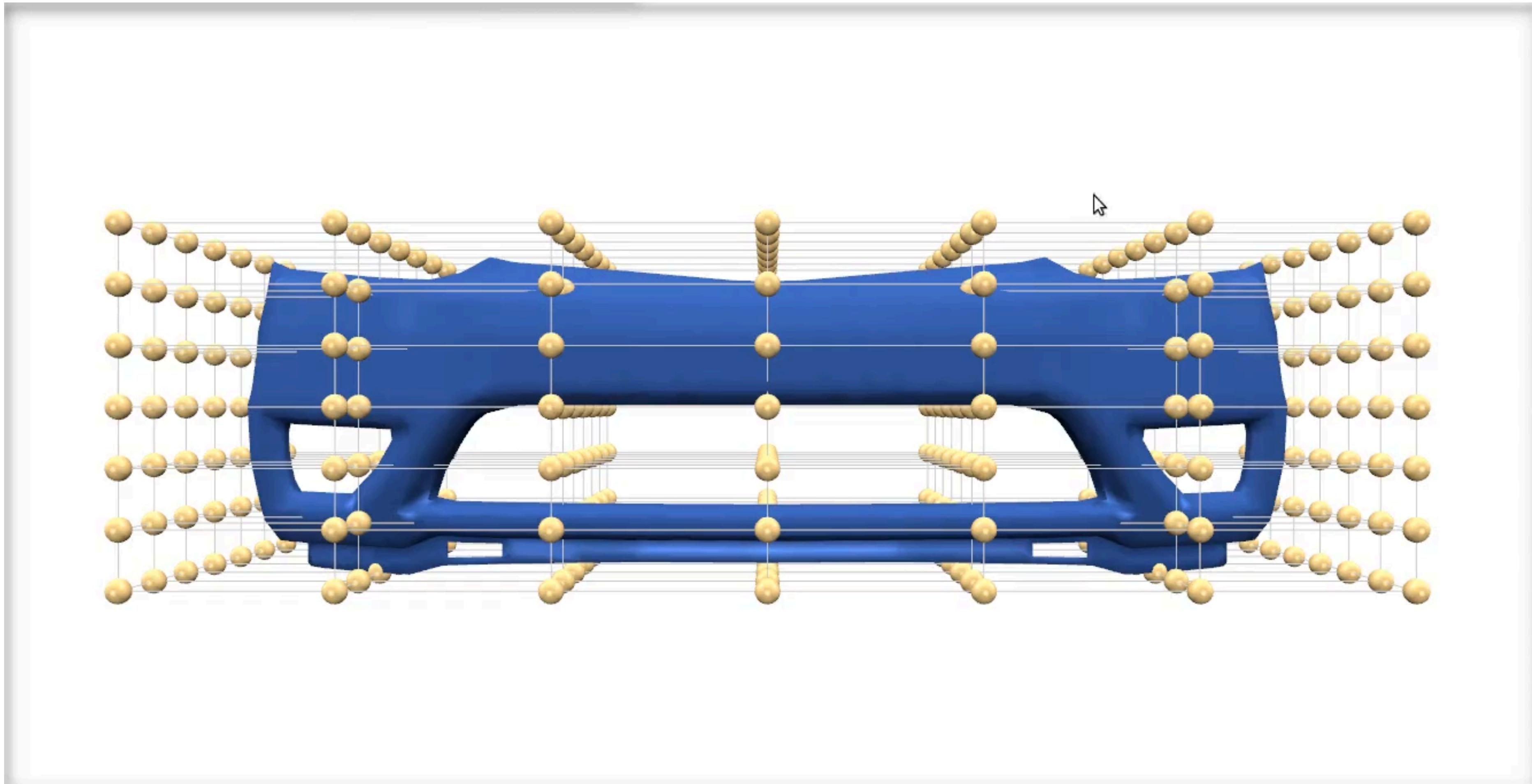
- Tensor product *spline* deformations allow for *local control*

$$\mathbf{d}(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \mathbf{d}_{i,j,k} N_i^l(u) N_j^m(v) N_k^n(w)$$



Sederberg & Perry, SIGGRAPH 1986

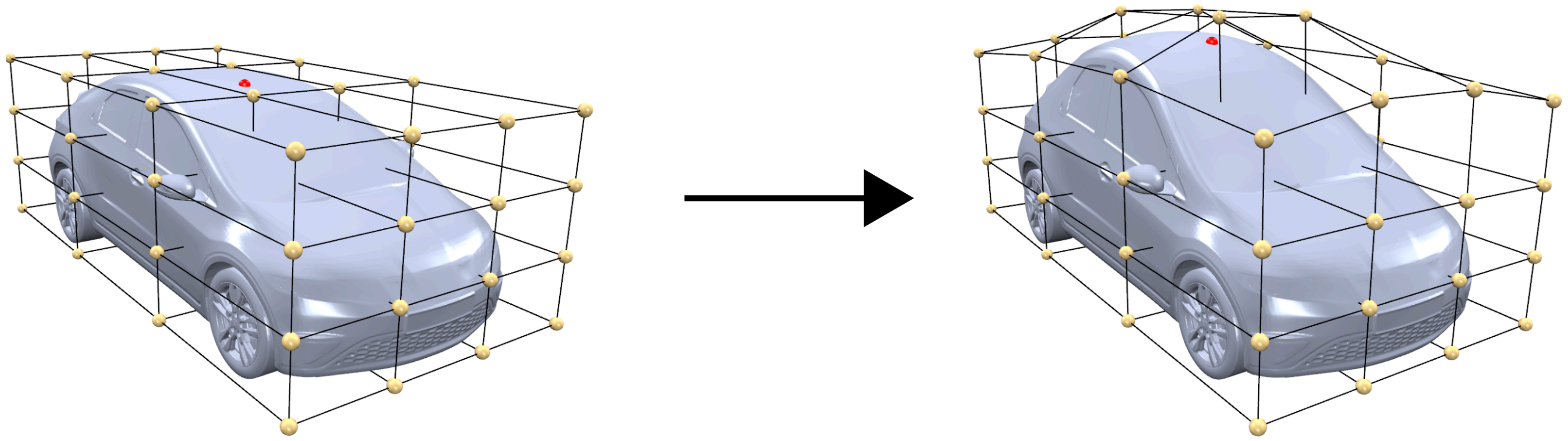
Spline-Based FFD



Sederberg & Perry, SIGGRAPH 1986

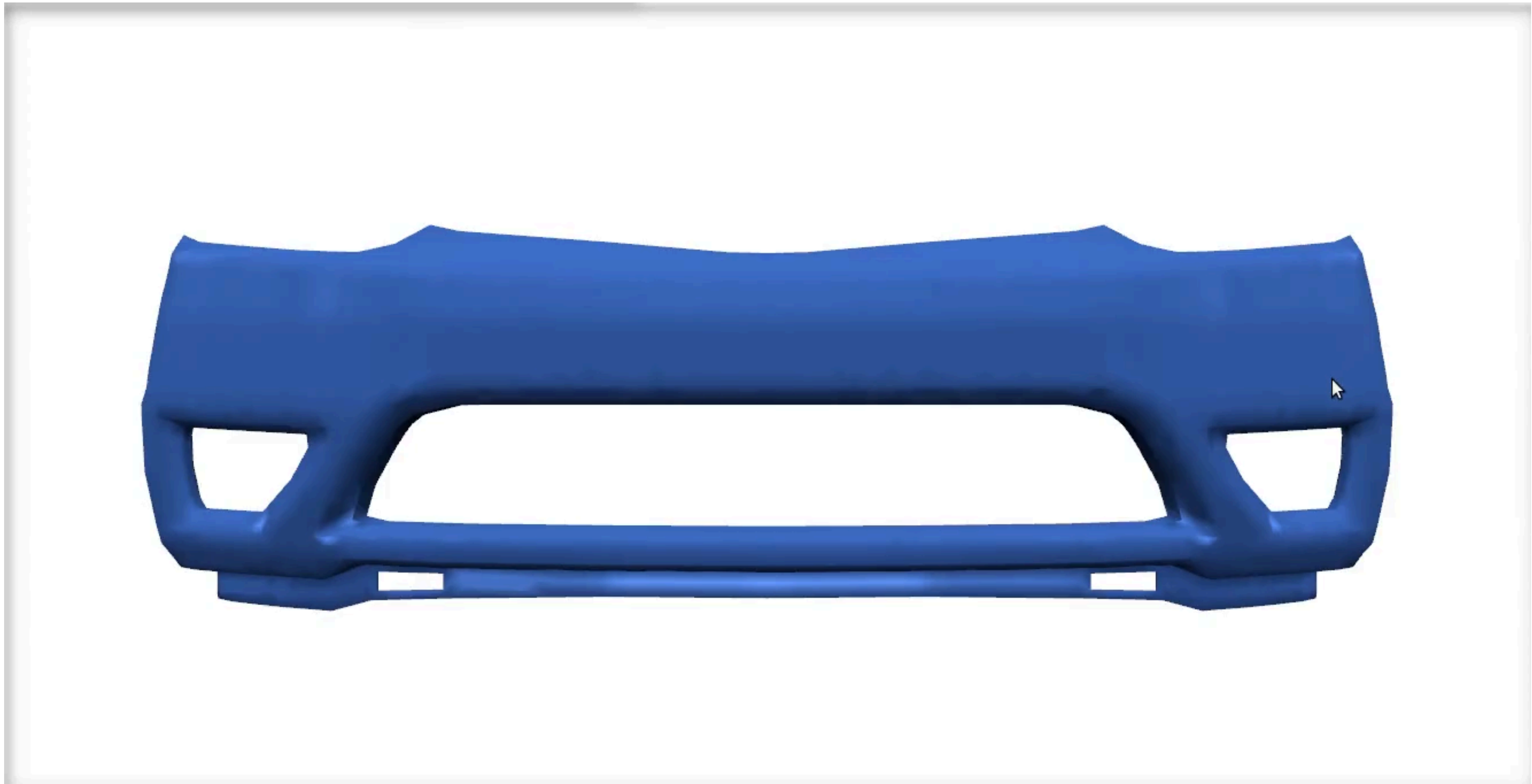
Direct Manipulation FFD

- How to prescribe displacement constraints?
  - Solve linear system for control points
  - Can be over- or under-determined
  - Pseudo-inverse: least squares, least norm



Hsu et al, SIGGRAPH 1992

Direct Manipulation FFD



Hsu et al, SIGGRAPH 1992

RBF Deformation

Deformation based on Radial Basis Functions

Summary

- **Freeform Surfaces**
  - Extend univariate curves to bivariate surfaces
  - Tensor product surfaces are “curves on curves”
  - 2D grid of control points
  - High quality smooth surfaces
- **Freeform Deformation**
  - Extend bivariate surfaces to trivariate volumes
  - Same tensor product idea
  - 3D grid of control points
  - Can deform arbitrary explicit representations

Quiz: Bezier/Spline Surfaces

We want to apply some scaling and rotation to a Bezier/spline surface.
Do we get the same result if we...

1. apply the transformation to the control points and then tesselate it to a triangle mesh
2. tesselate it to a triangle mesh and then apply transformation to the mesh vertices

A: Yes, same result.

B: No, different results.

Quiz: Freeform Deformation

We want to manipulate a Bezier/spline surface using freeform deformation.
Do we get the same result if we...

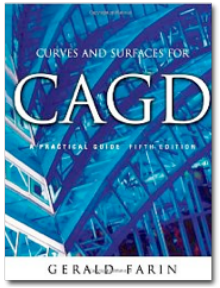
1. apply the deformation to the control points and then tesselate it to a triangle mesh
2. tesselate it to a triangle mesh and then apply deformation to the mesh vertices

A: Yes, same result.

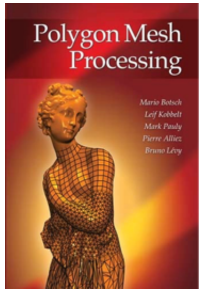
B: No, different results.

Literature

- Farin: *Curves and Surfaces for CAGD. A Practical Guide*, Morgan Kaufmann, 2001
  - Chapters 14, 16



- Botsch et al., *Polygon Mesh Processing*, AK Peters, 2010
  - Chapter 9.5



- Sederberg & Parry, Free-Form Deformations of Solid Geometric Models, SIGGRAPH 1986
- Hsu et al., Direct Manipulation of Free-Form Deformations, SIGGRAPH 1992