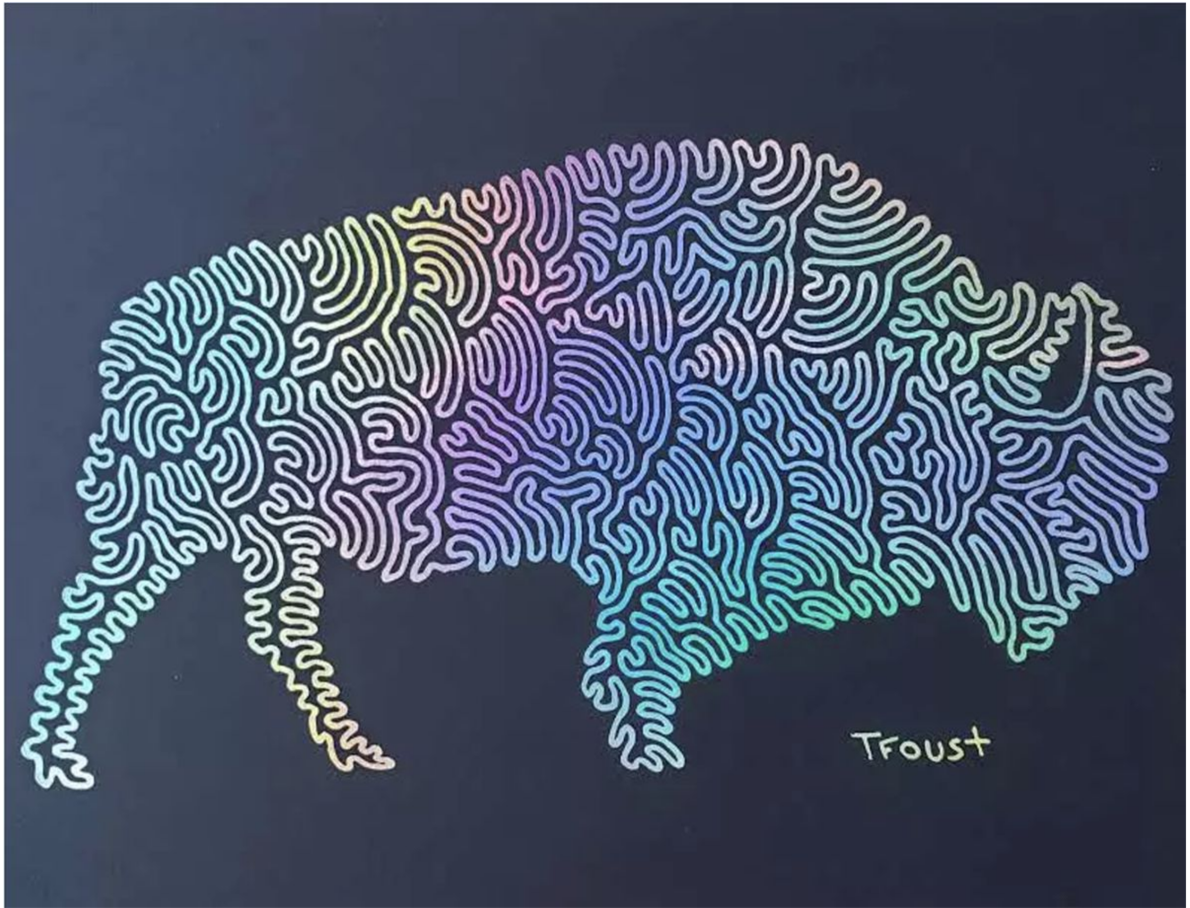
CSCI 4530/6530 Advanced Computer Graphics

`https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S25/`

Lecture 3: Curves & Surfaces

https://www.moillusions.com/glass-water-optical-illusion/

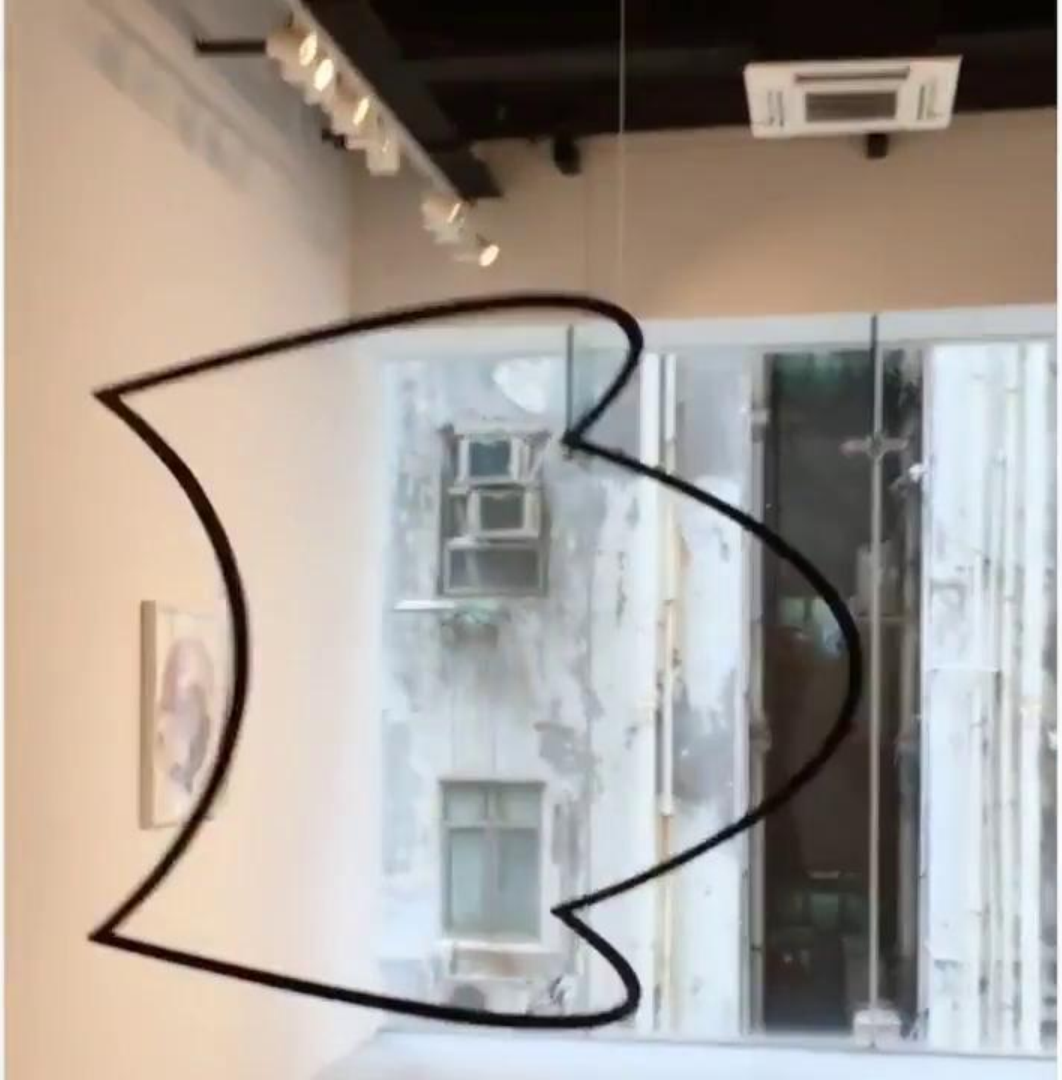
One Line Bison, Tyler Foust, 2020



https://www.tylerfoust.com/

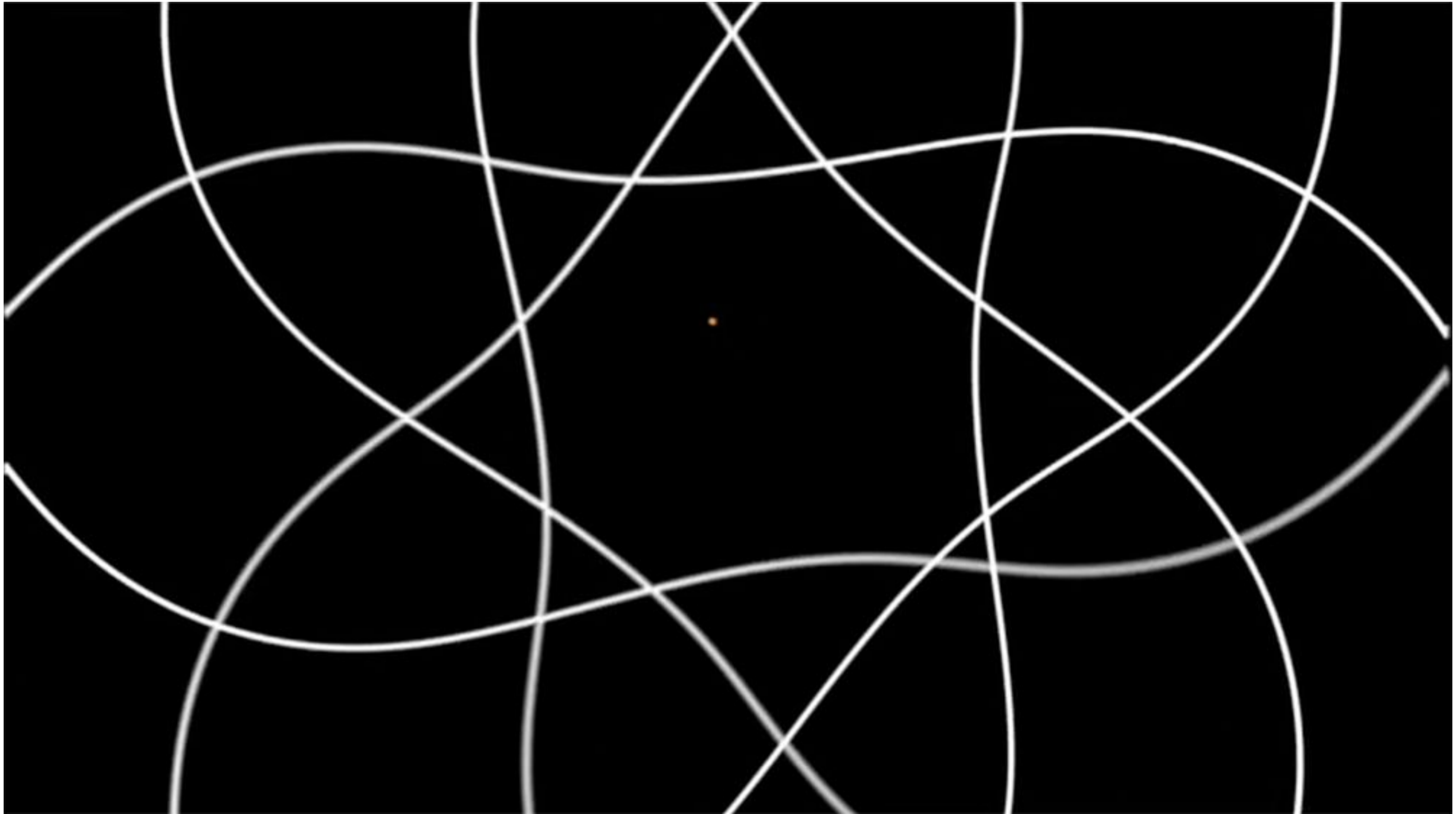
Squaring the Circle, Troika, 2013

http://troika.uk.com/work/troika-squaring-the-circle/



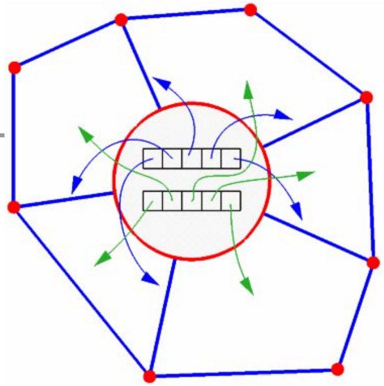
Herbstlaub

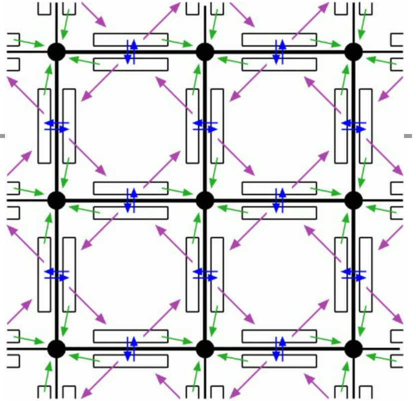
Oliver Vogel, Siggraph 2007

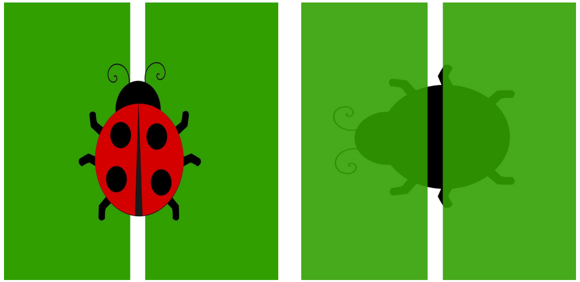


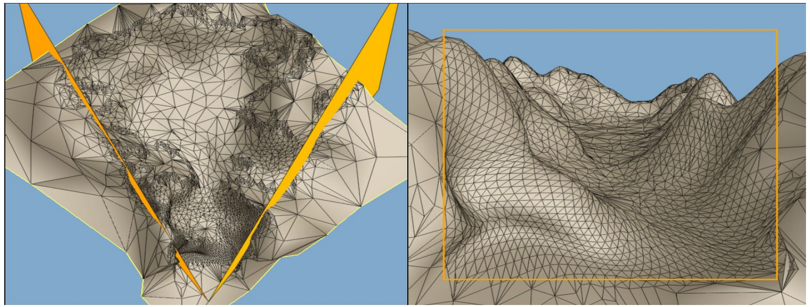
Last Time?

- Adjacency Data Structures
  - Geometric & topologic information
  - Dynamic allocation
  - Efficiency of access
- Mesh Simplification
  - edge collapse/vertex split
  - geomorphs
  - progressive transmission
  - view-dependent refinement







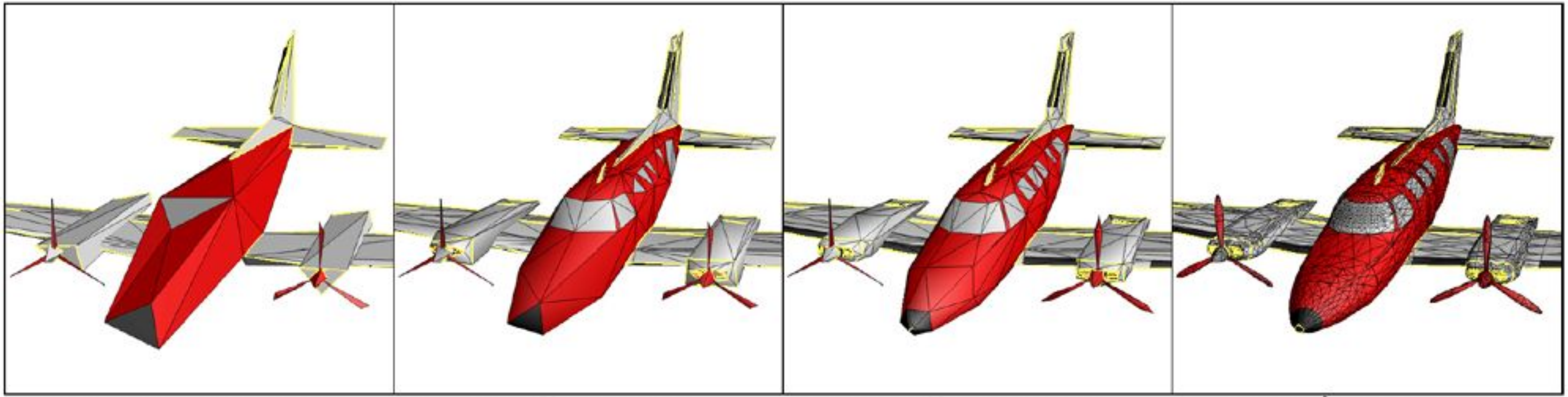


Today

- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Today’s Reading:

- Hugues Hoppe “Progressive Meshes” SIGGRAPH 1996



(a) Base mesh M^0 (150 faces) (b) Mesh M^{175} (500 faces) (c) Mesh M^{425} (1,000 faces) (d) Original $\hat{M}=M^n$ (13,546 faces)

Progressive Meshes

- Mesh Simplification
  - vertex split / edge collapse
  - geometry & discrete/scalar attributes
  - priority queue
- Level of Detail
  - geomorphs
- Progressive Transmission
- Mesh Compression
- Selective Refinement
  - view dependent

Selective Refinement

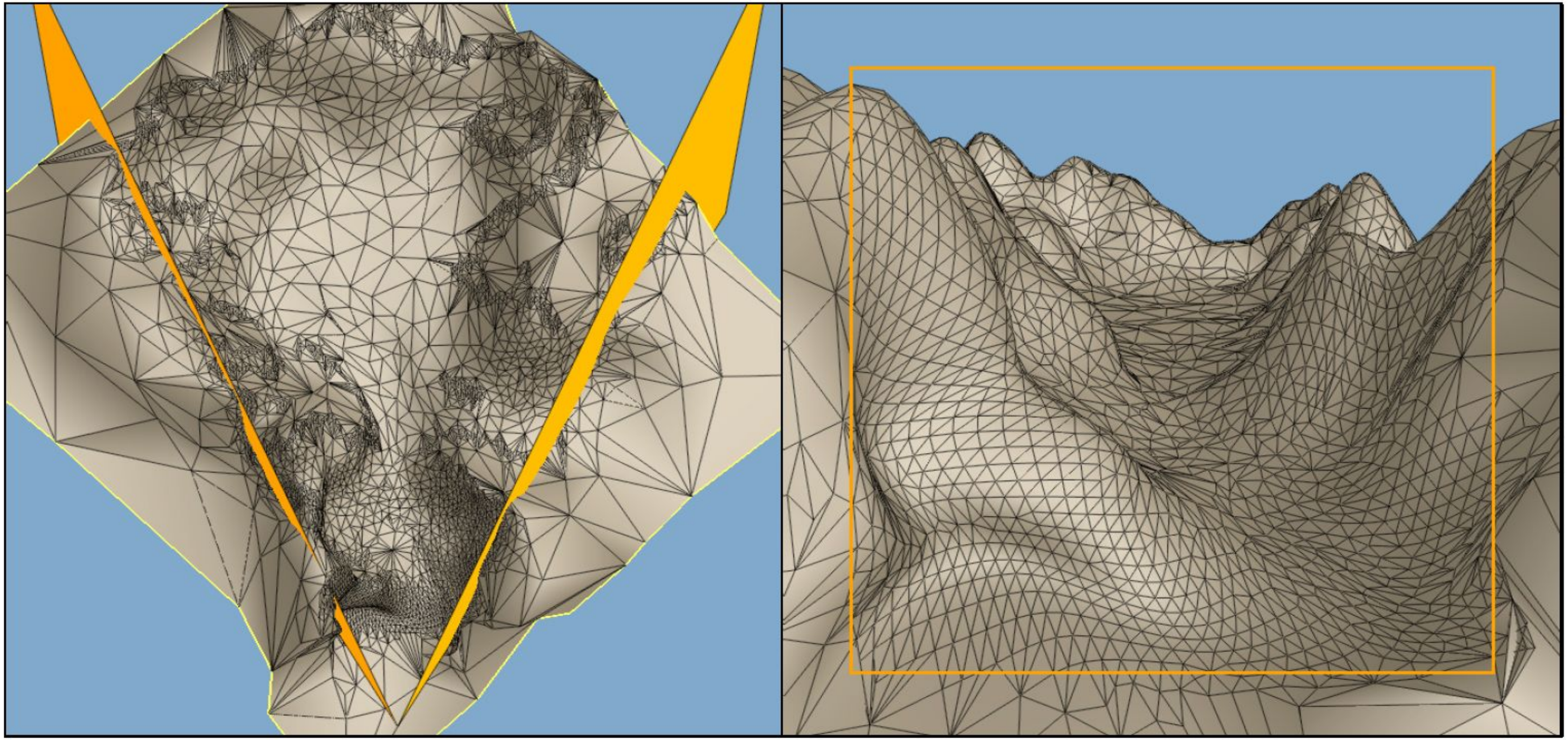
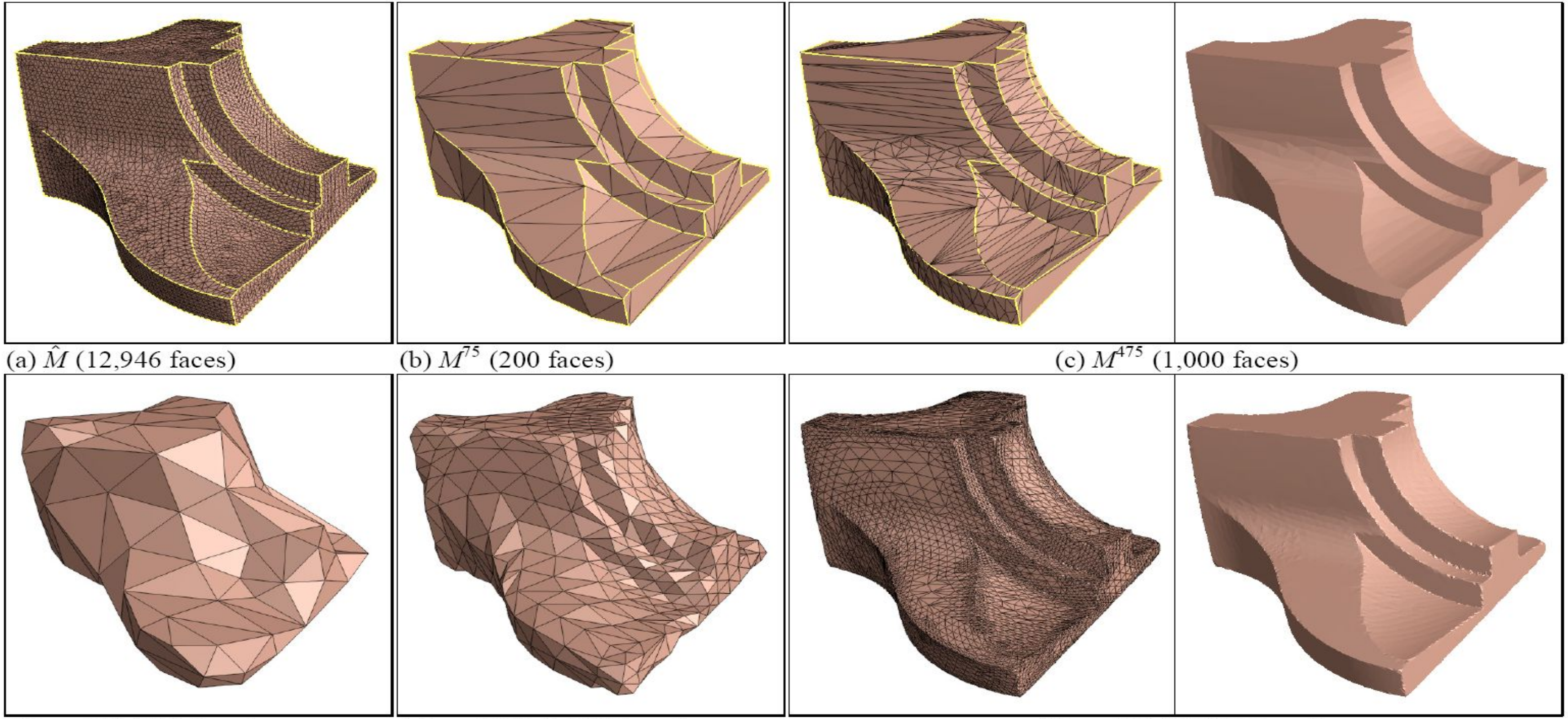


Figure 10: Selective refinement of a terrain mesh taking into account view frustum, silhouette regions, and projected screen size of faces (7,438 faces).

Preserving Discontinuity Curves



(a) $\hat{M}$ (12,946 faces) (b) M^{75} (200 faces) (c) M^{475} (1,000 faces)

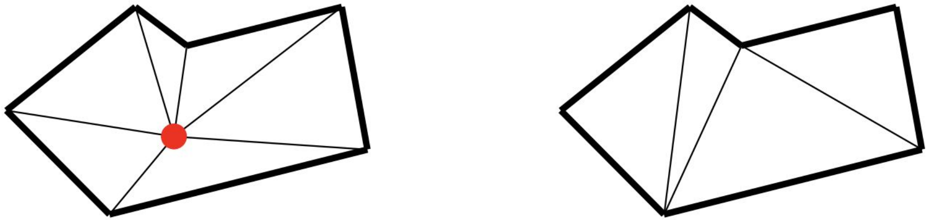
(d) $\epsilon = 9.0$ (192 faces) (e) $\epsilon = 2.75$ (1,070 faces) (f) $\epsilon = 0.1$ (15,842 faces)

Figure 12: Approximations of a mesh $\hat{M}$ using (b–c) the PM representation, and (d–f) the MRA scheme of Eck et al. [7]. As demonstrated, MRA cannot recover $\hat{M}$ exactly, cannot deal effectively with surface creases, and produces approximating meshes of inferior quality.

- Problematic / visible “popping” between LODs, geomorphing
- Discrete vs continuous LOD – is continuous necessary?
  Progressive transmission, progressive refinement
- Lossless / invertible
- Research: appreciate original context, iterating/extending prior work, hybrid techniques, future work
- Research directly used by / influencing games/other industry?
- Triangles vs quads, collapse vs. other ops (split, swap, etc)
- Expensive cost? Precompute vs on-the fly?
  Can we reduce this by approximation? Or parallelize it?
- Mesh formalism, Energy function (springs?) to select edge
  – how it works not immediately intuitive
- Limitations? Incorrectly, preserve unimportant details,
  store unnecessary high resolution? Can’t use on animated meshes

Other Simplification Strategies

- Remove a vertex & surrounding triangles, re-triangulate the hole



- Merge Nearby Vertices (*will likely change the topology)*

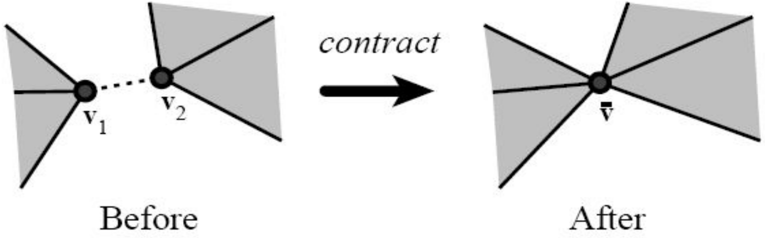
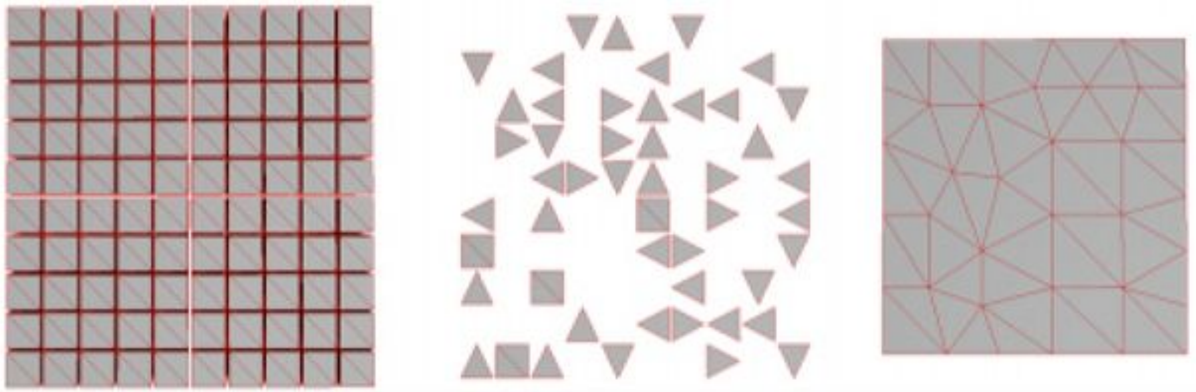


Figure 2: **Non-edge contraction.** When non-edge pairs are contracted, unconnected sections of the model are joined. The dashed line indicates the two vertices being contracted together.

Garland & Heckbert, “Surface Simplification Using Quadric Error Metrics” SIGGRAPH 1997

Is it Important to Preserve Original Topology?



Garland & Heckbert,
“Surface Simplification
Using Quadric Error Metrics”
SIGGRAPH 1997

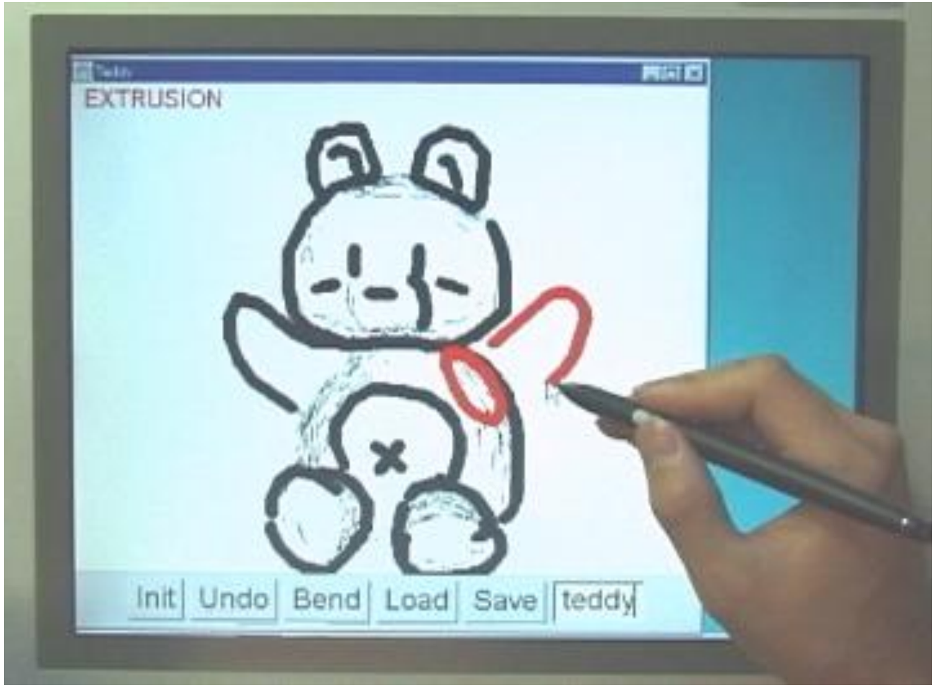
Figure 3: On the left is a regular grid of 100 closely spaced cubes. In the middle, an approximation built using only edge contractions demonstrates unacceptable fragmentation. On the right, the result of using more general pair contractions to achieve aggregation is an approximation much closer to the original.

Today

- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Reading for Today

"Teddy:
A Sketching
Interface for
3D Freeform
Design",
Igarashi et al.,
SIGGRAPH
1999



How do we represent objects that don't have flat polygonal faces & sharp corners?
What are the right tools to design/construct digital models of blobby, round, or soft things?
What makes a user interface intuitive, quick, and easy-to-use for beginners?

- Attention to UI, lowering the barrier to entry for novices
- Simple algorithm
- Limitation: does not work for non-spherical base shapes
- Challenge: making 3D shape with 2D input
- Tradeoff: simplicity vs. fully-featured modeling software
- “Direct manipulation” – draw contours on screen rather than typing numbers into boxes physically separated from visual result
- Has Teddy made an impact on modeling software? If not, why not?

What NOT to write about the assigned readings:

- "There was alot of math in the paper.
  Math is hard. I didn't understand the math."
- "This paper was published in the dark ages using slow computers.
  I wonder how fast it would be with a GPU."
- "The pictures were pretty. I liked watching the video."
- “Now that we have AI/ML, the results will be much better.”

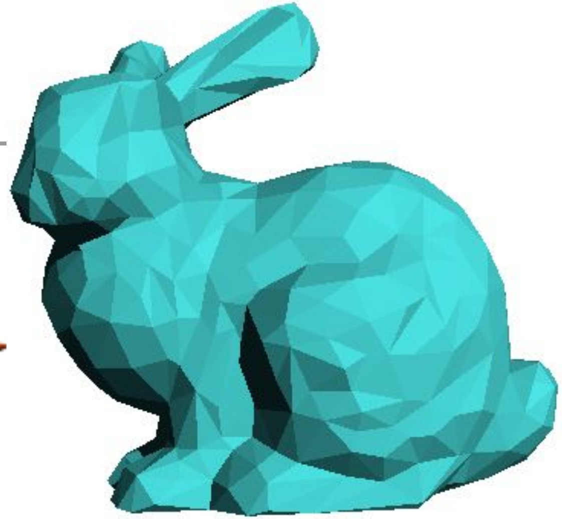
Focus on the technical details / algorithms described in the paper
What additional info would you need to implement this method from scratch?
How could the writing / presentation / examples be improved?

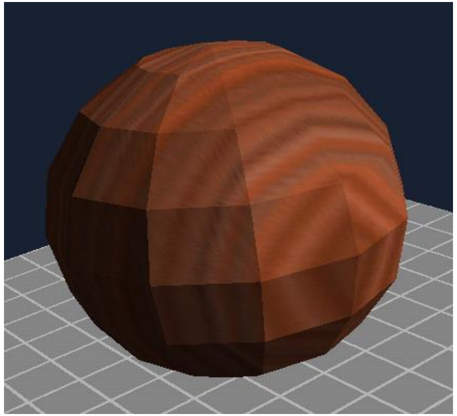
Today

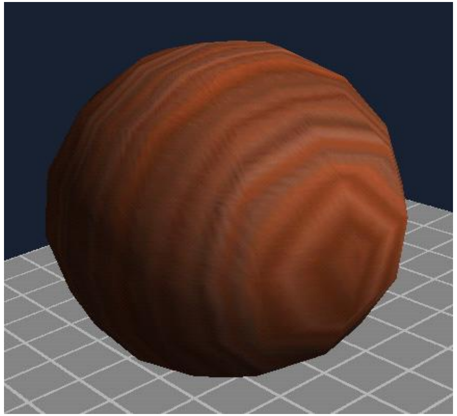
- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
  - Interpolating Color & Normals in OpenGL
  - Some Modeling Tools & Definitions
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Limitations of Polygonal Meshes

- Planar facets (& silhouettes)
- Fixed resolution
- Deformation is difficult
- No natural parameterization (for texture mapping)
- Incorrect collision detection
- Solid texturing problems

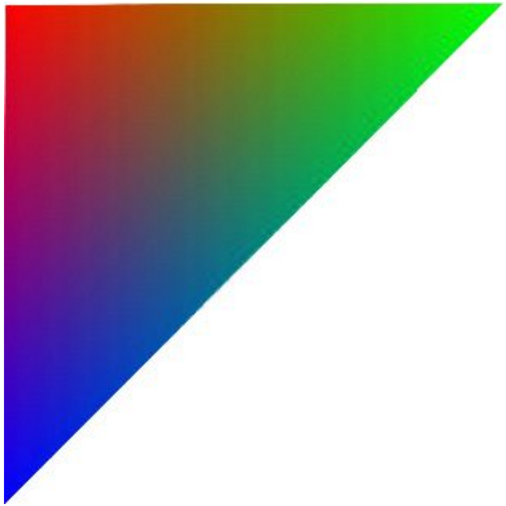


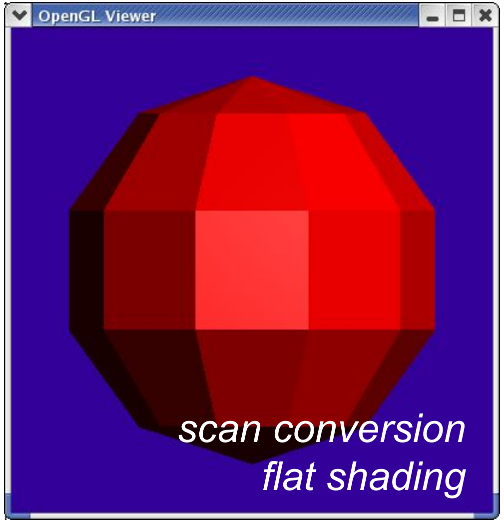


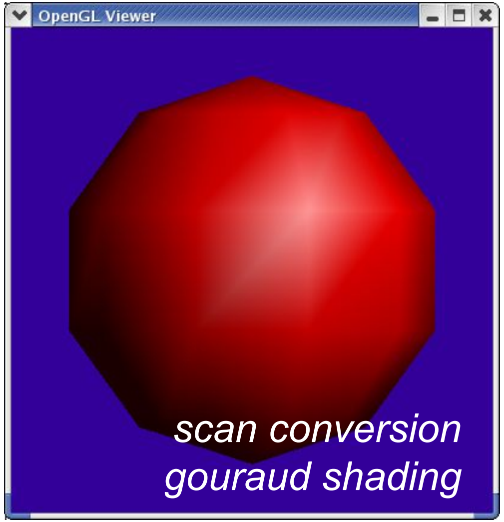


Color & Normal Interpolation

- It's easy in OpenGL to specify different colors and/or normals at the vertices of triangles:
- Why is this useful?

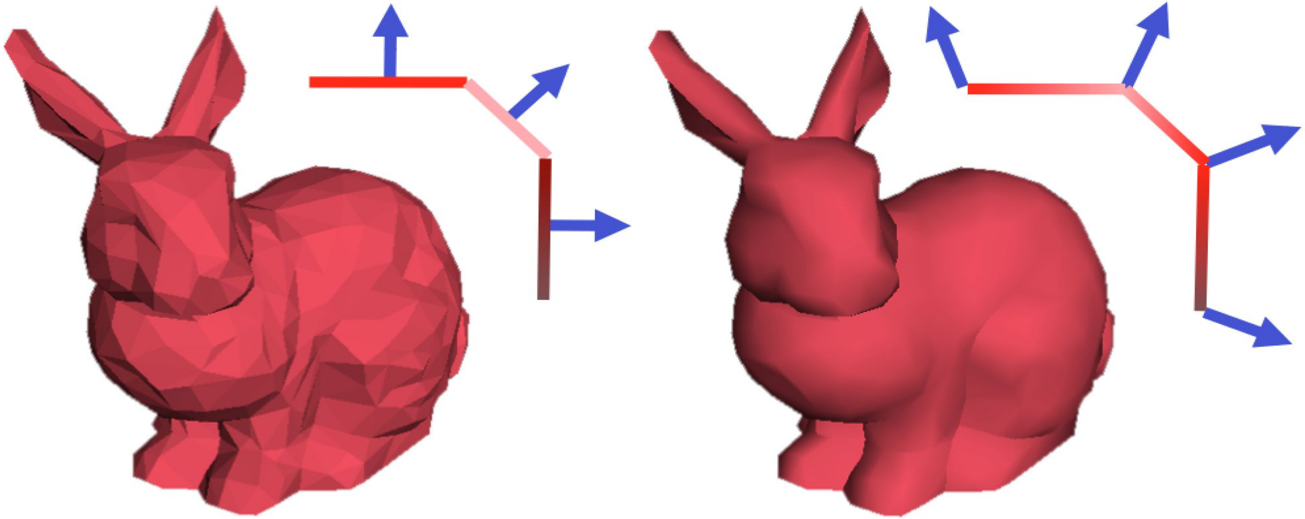






What is Gouraud Shading?

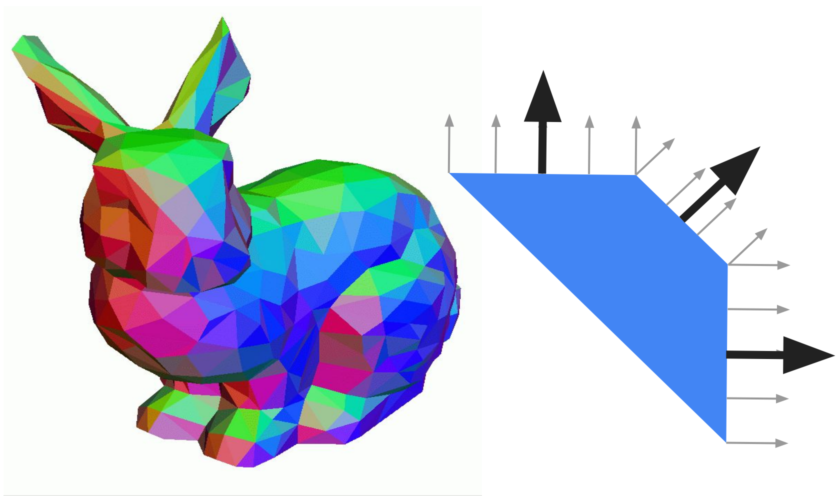
- Instead of shading with the normal of the triangle, we'll shade the vertices with the *average normal* at the vertex and *interpolate the shaded color* across each face. This gives the *illusion of a smooth surface* with smoothly varying normals.
- How do we compute Average Normals? Is it expensive??

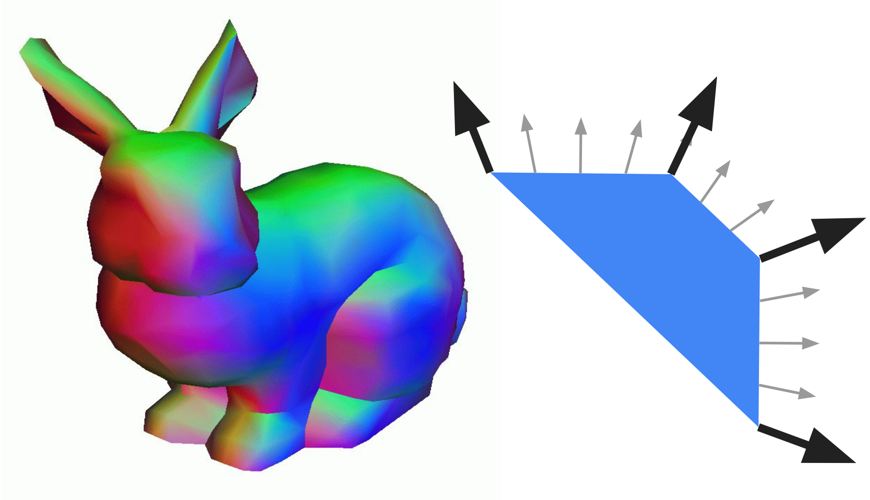


Phong Normal Interpolation

(Not Phong *Shading*)

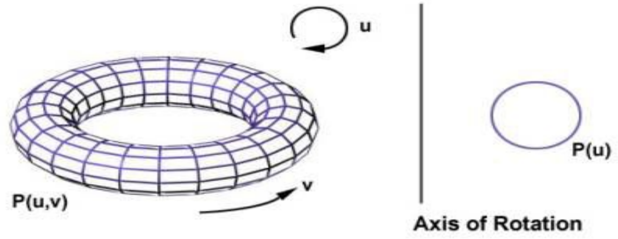
- *Interpolate the average vertex normals* across the face and compute *per-pixel shading*
  - Normals should be re-normalized (ensure length=1)
- Before shaders, per-pixel shading was not possible in hardware (Gouraud shading is actually a decent substitute!)



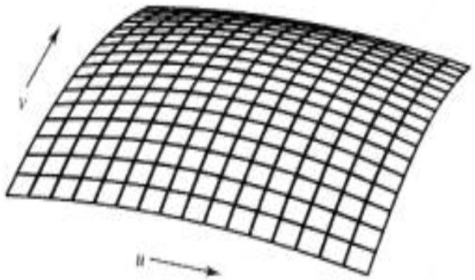


Some Non-Polygonal Modeling Tools

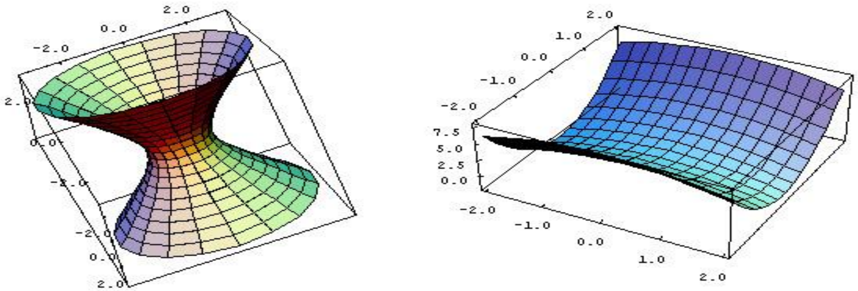
Extrusion



Surface of Revolution

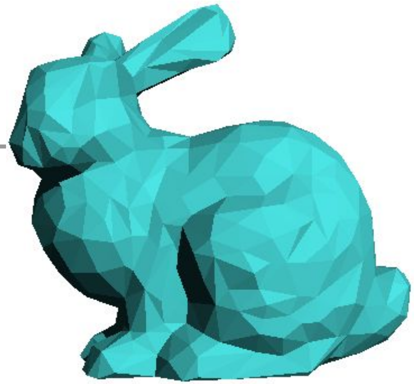


Spline Surfaces/Patches

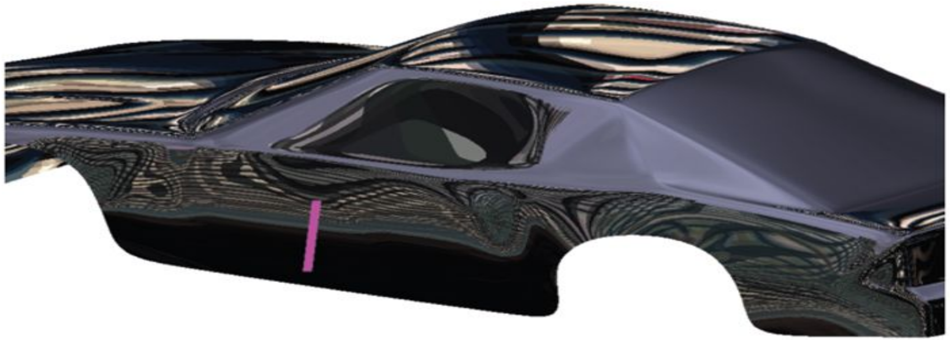


Quadrics and other
Implicit Polynomials

Continuity Definitions:

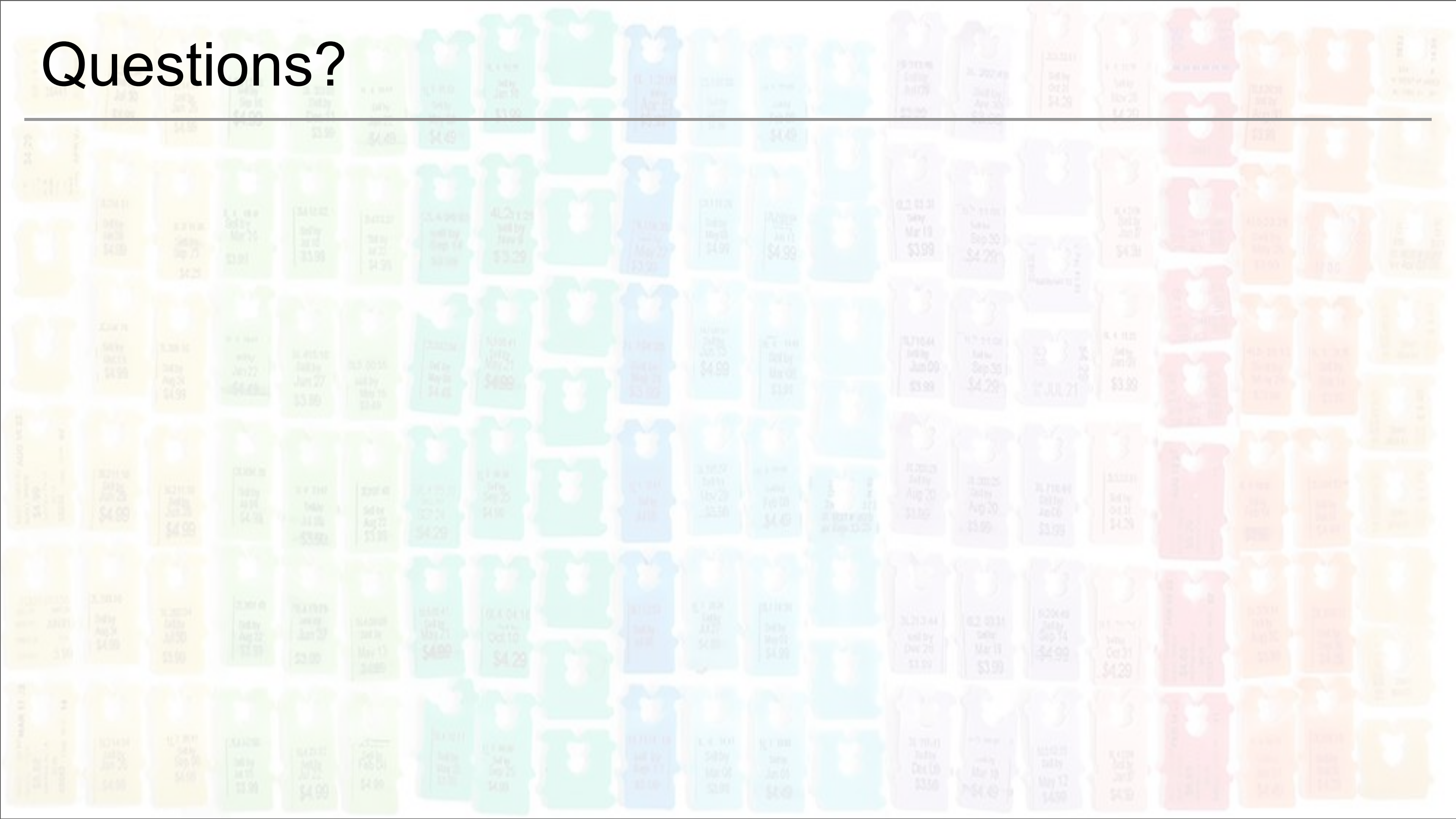


- C^0 continuous
  - curve/surface has no breaks/gaps/holes
- G^1 continuous
  - tangent at joint has same direction
- C^1 continuous
  - curve/surface derivative is continuous
  - tangent at joint has same direction *and* magnitude
- C^n continuous
  - curve/surface through n^{th} derivative is continuous
  - important for shading



"Shape Optimization Using Reflection Lines", Tosun et al., 2007

Questions?

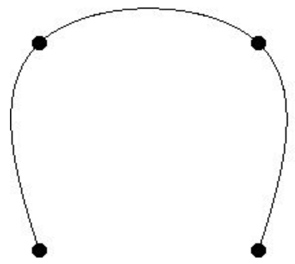


Today

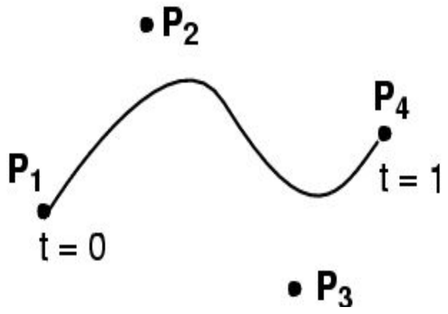
- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Definition: What's a Spline?

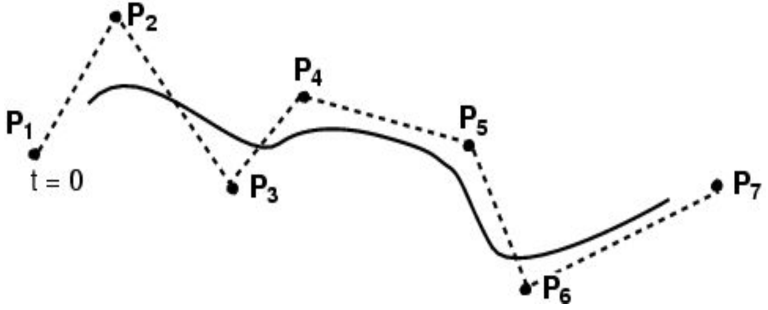
- Smooth curve defined by some control points
- Moving the control points changes the curve



Interpolation

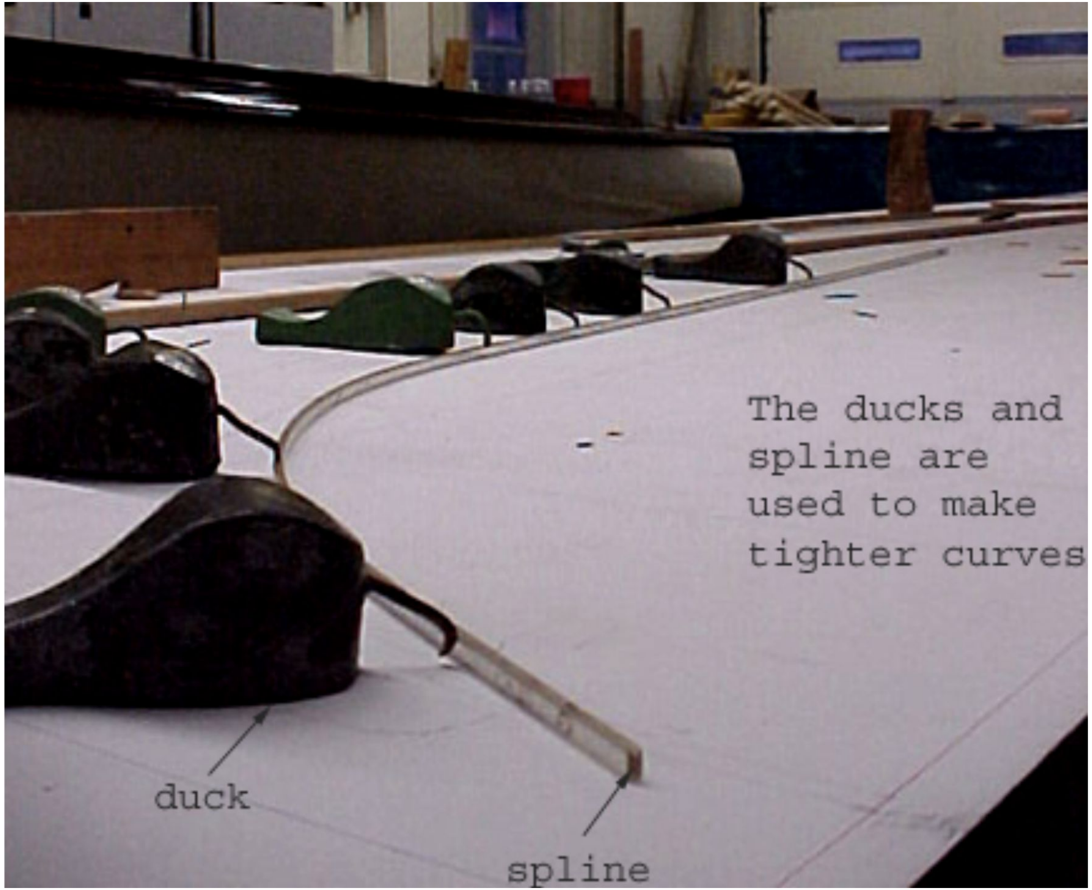


Bézier (approximation)



BSpline (approximation)

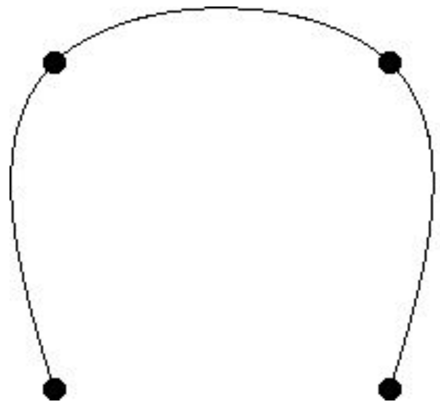
Interpolation Curves / Splines



www.abm.org

Interpolation Curves

- Curve is constrained to pass through all control points



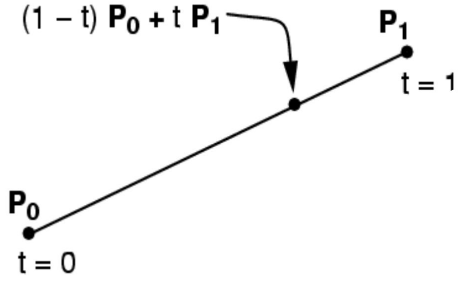
- Given points P_0, P_1, ... P_n, find lowest degree polynomial which passes through the points

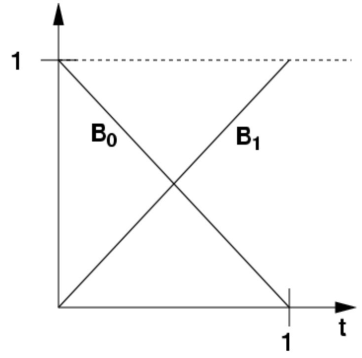
$$x(t) = a_{n-1} t^{n-1} + ... + a_2 t^2 + a_1 t + a_0$$
$$y(t) = b_{n-1} t^{n-1} + ... + b_2 t^2 + b_1 t + b_0$$

Linear Interpolation

- Simplest "curve" between two points

  $Q(t) = (1 - t)\,\boldsymbol{P_0} + t\,\boldsymbol{P_1}$





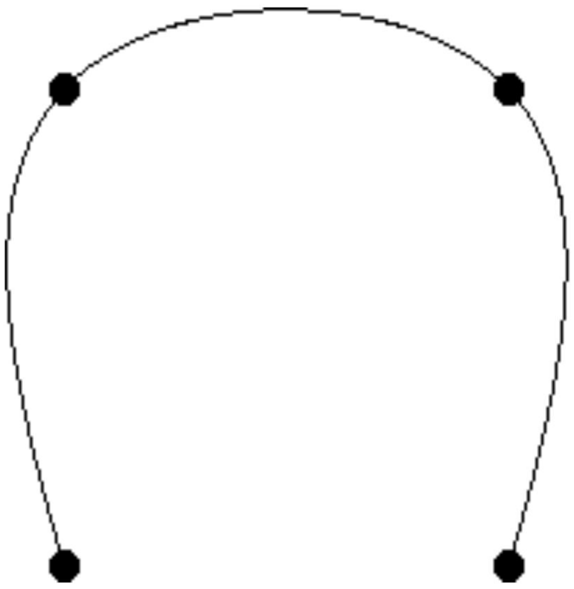
Spline Basis Functions

a.k.a. Blending Functions

$$Q(t) = \begin{pmatrix} Q_x(t) \\ Q_y(t) \\ Q_z(t) \end{pmatrix} = \begin{pmatrix} (P_0) & (P_1) \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}$$

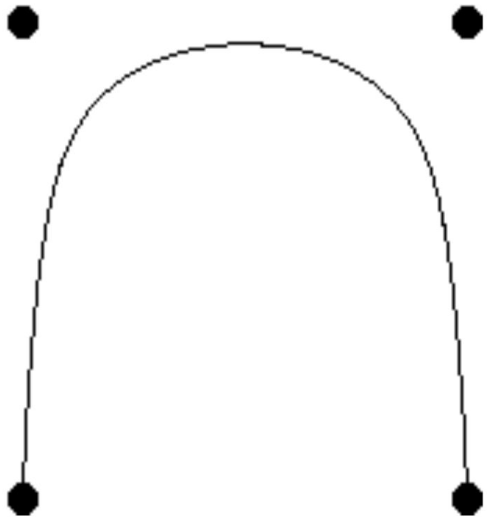
Q(t) = Geometry **G** · Spline Basis **B** · Power Basis **T**(*t*)

Interpolation vs. Approximation Curves



Interpolation

curve must pass through control points

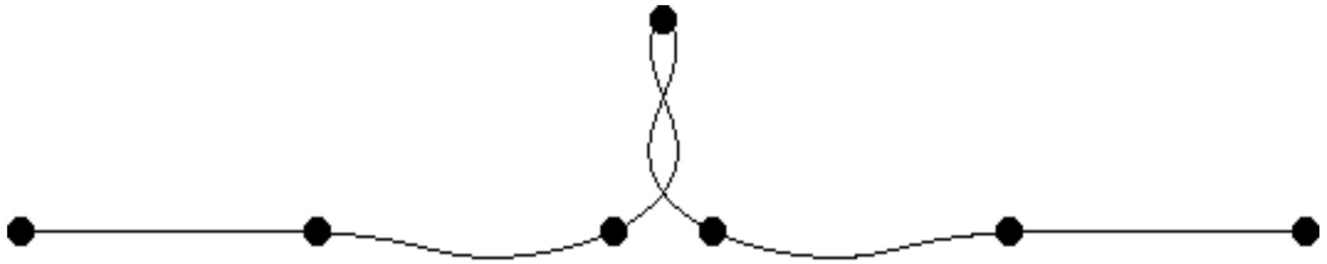


Approximation

curve is influenced by control points

Interpolation vs. Approximation Curves

- Interpolation Curve:
  over constrained → lots of (undesirable?) oscillations



- Approximation Curve – more reasonable?

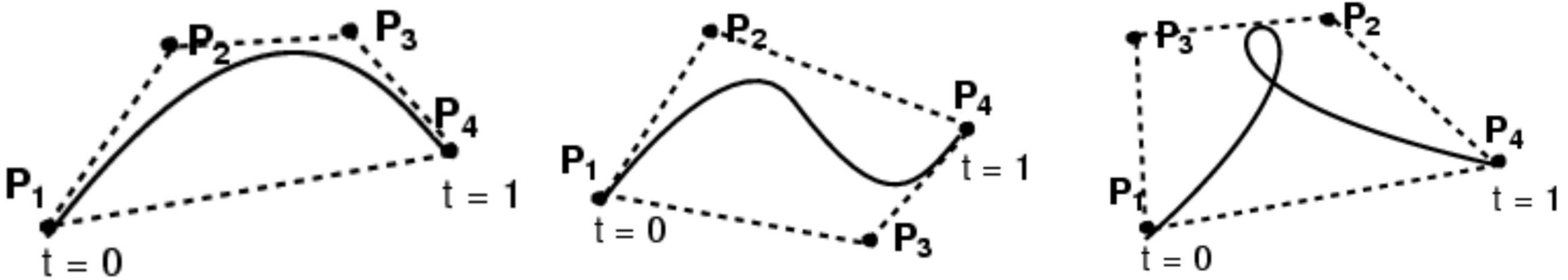
Questions?

Today

- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Cubic Bézier Curve

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at $\mathbf{P_1}$ to $(\mathbf{P_2}-\mathbf{P_1})$ and at $\mathbf{P_4}$ to $(\mathbf{P_4}-\mathbf{P_3})$

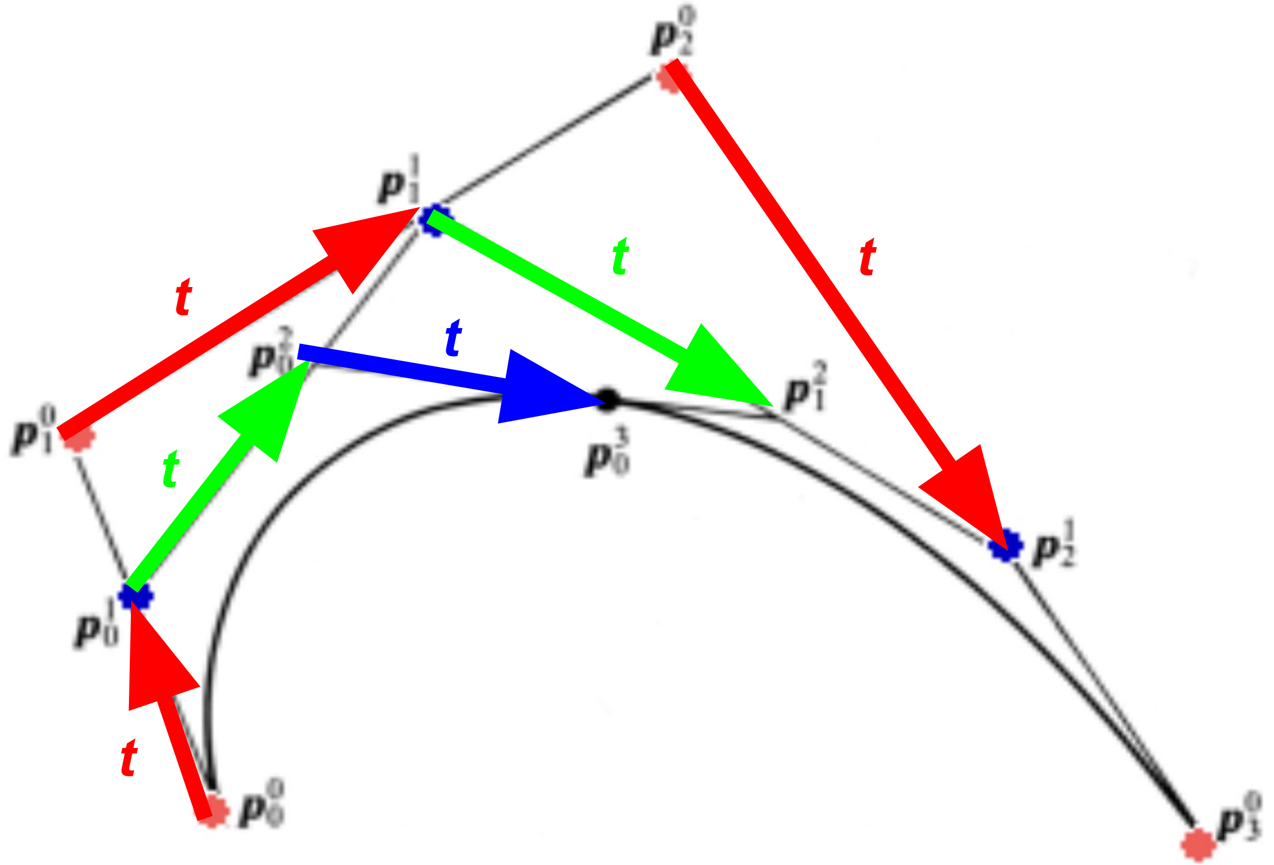


A Bézier curve is bounded by the convex hull of its control points.

Cubic Bézier Curve

- de Casteljau's algorithm for constructing Bézier curves:

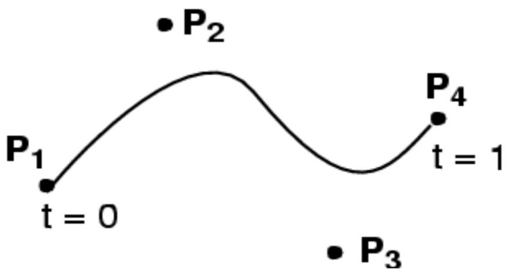
- For any *t*

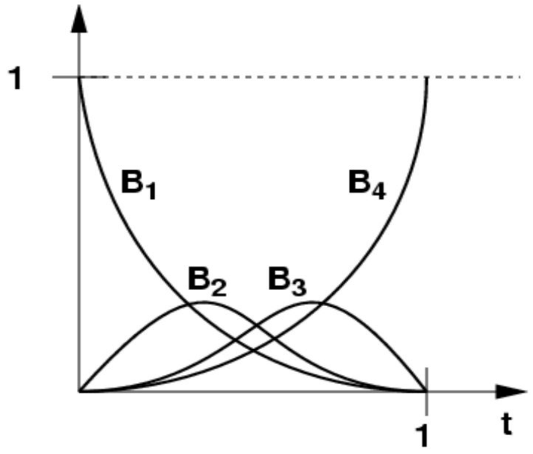


Cubic Bézier Curve

$Q(t)$ = Geometry **G** · Spline Basis **B** · Power Basis $\mathbf{T}(t)$ = $\mathbf{GBT}(t)$

$$Q(t) = (1-t)^3 \mathbf{P_1} + 3t(1-t)^2 \mathbf{P_2} + 3t^2(1-t) \mathbf{P_3} + t^3 \mathbf{P_4}$$





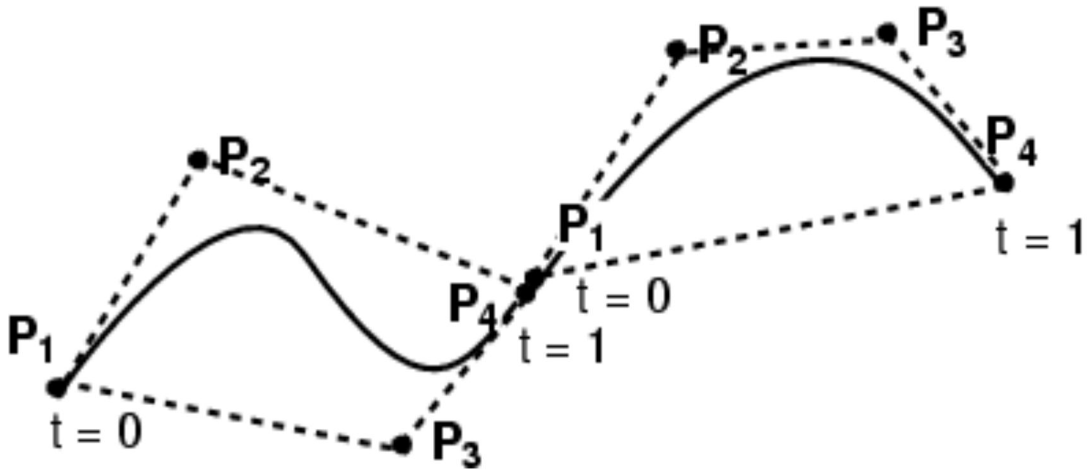
$$B_{Bezier} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Bernstein Polynomials

$$B_1(t) = (1-t)^3;\ B_2(t) = 3t(1-t)^2;\ B_3(t) = 3t^2(1-t);\ B_4(t) = t^3$$

Connecting Cubic Bézier Curves

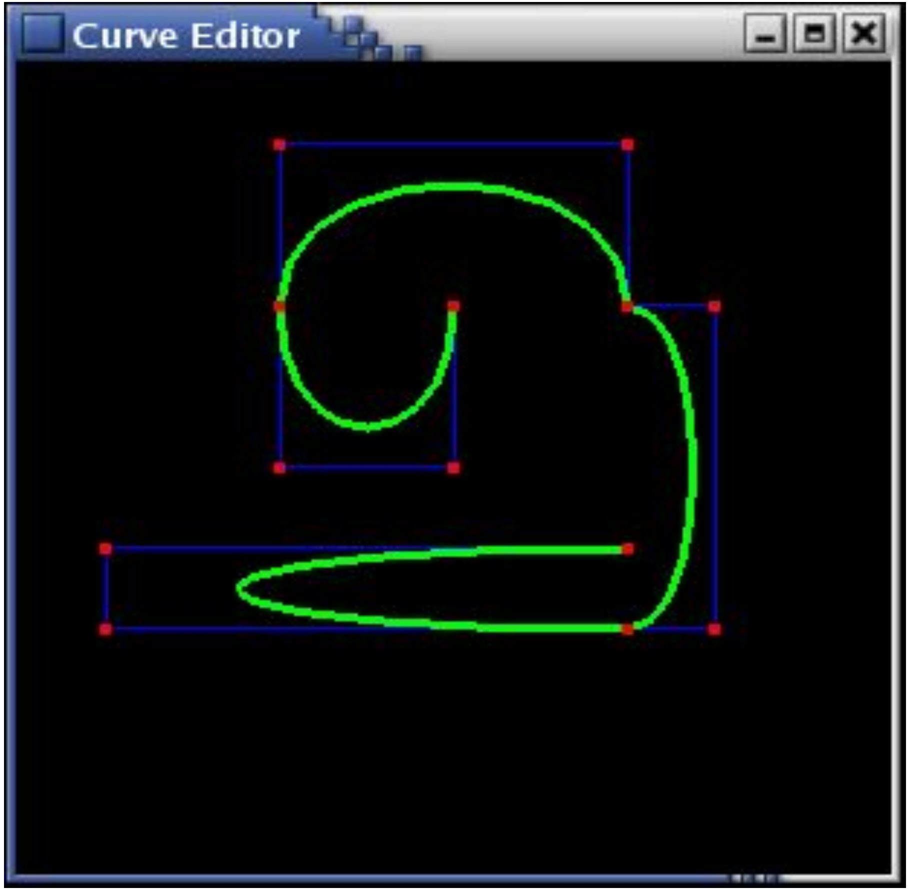
- How can we guarantee C^0 continuity?
- How can we guarantee G^1 continuity?
- How can we guarantee C^1 continuity?
- Can't guarantee higher C^2 or higher continuity



Asymmetric: Curve goes through some control points but misses others

Connecting Cubic Bézier Curves

- Where is this curve
  - C^0 continuous?
  - G^1 continuous?
  - C^1 continuous?
- What's the relationship between:
  - the # of control points, and
  - the # of cubic Bézier subcurves?



Higher-Order Bézier Curves

- > 4 control points
- Bernstein Polynomials as the basis functions

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}, \qquad 0 \le i \le n$$

- Every control point affects the entire curve
  - Not simply a local effect
  - More difficult to control for modeling

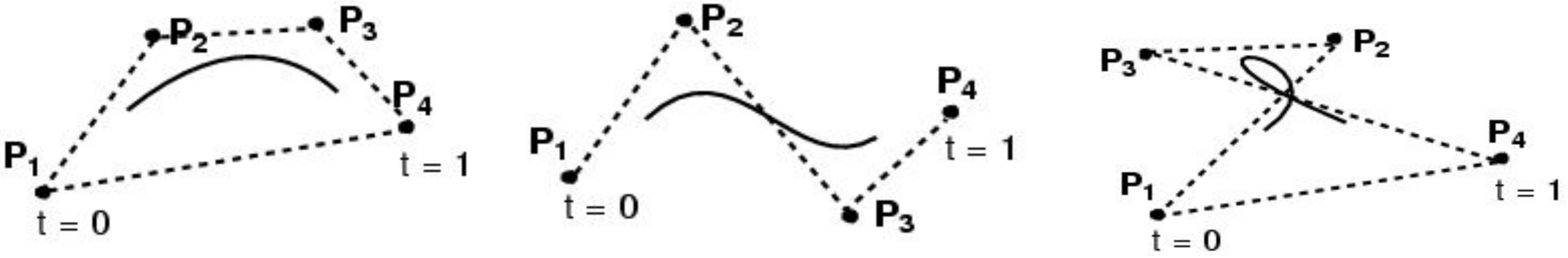
Questions?

Today

- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Cubic BSplines

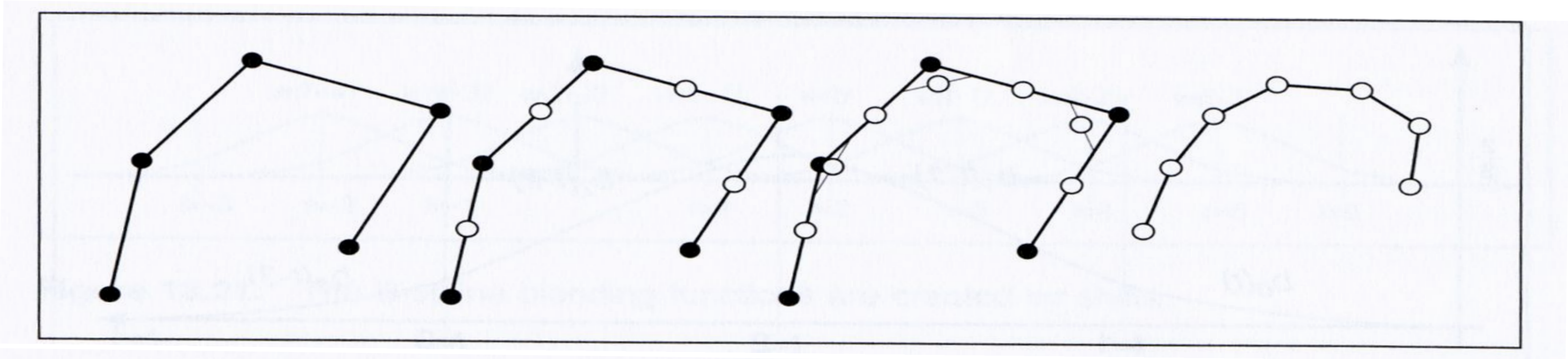
- ≥ 4 control points
- Locally cubic
- Curve is not constrained to pass through any control points

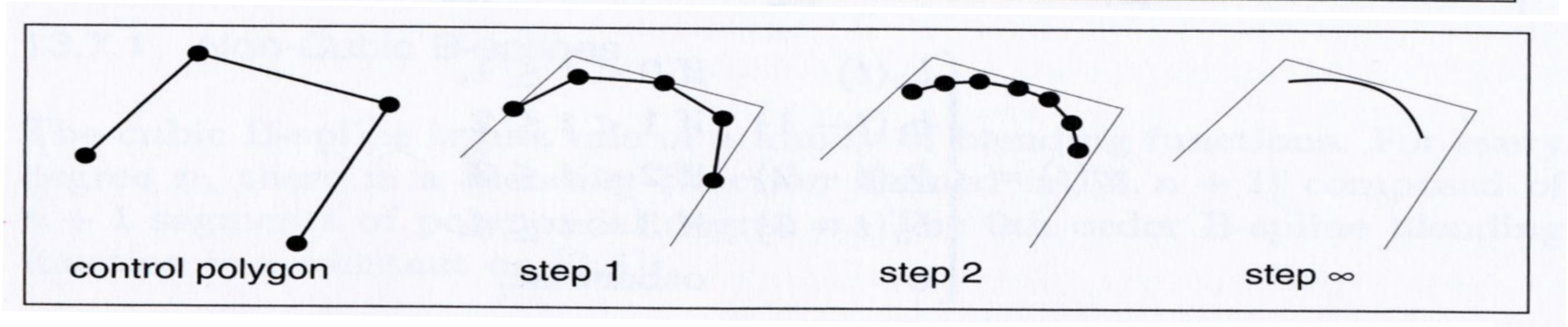


A BSpline curve is also bounded by the convex hull of its control points.

Cubic BSplines

- Iterative method for constructing BSplines



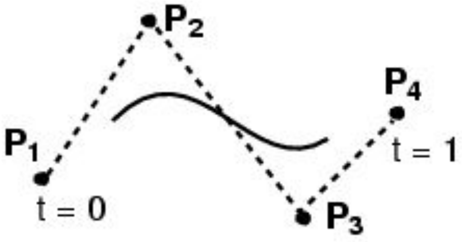


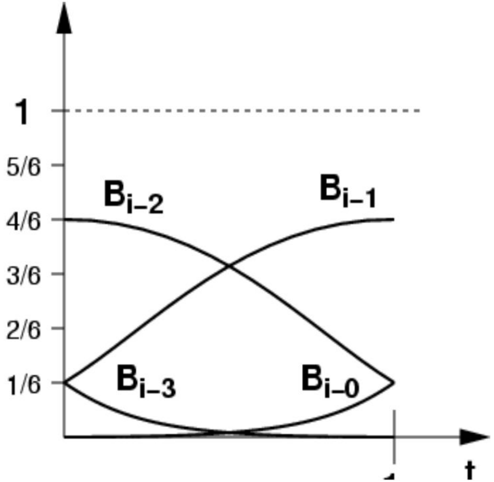
Shirley, Fundamentals of Computer Graphics

Cubic BSplines

$Q(t)$ = Geometry **G** · Spline Basis **B** · Power Basis $\mathbf{T}(t)$ = $\mathbf{GBT}(t)$

$$Q(t) = \frac{(1-t)^3}{6}P_{i-3} + \frac{3t^3 - 6t^2 + 4}{6}P_{i-2} + \frac{-3t^3 + 3t^2 + 3t + 1}{6}P_{i-1} + \frac{t^3}{6}P_i$$

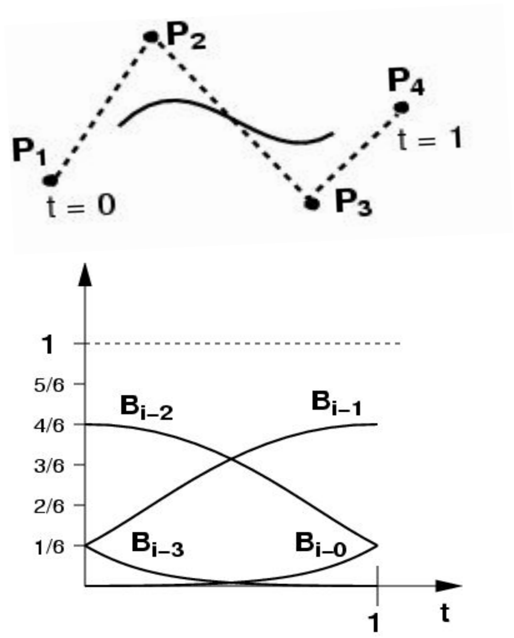


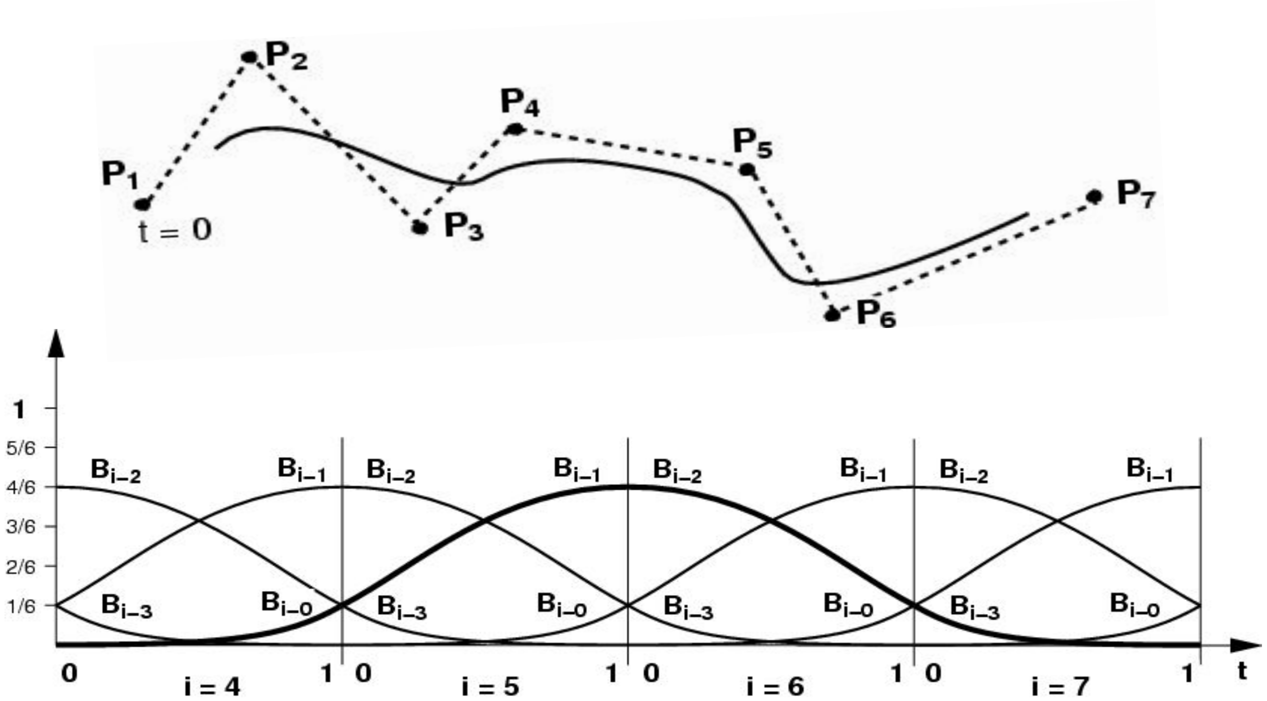


$$B_{B-Spline} = \frac{1}{6}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Connecting Cubic BSpline Curves

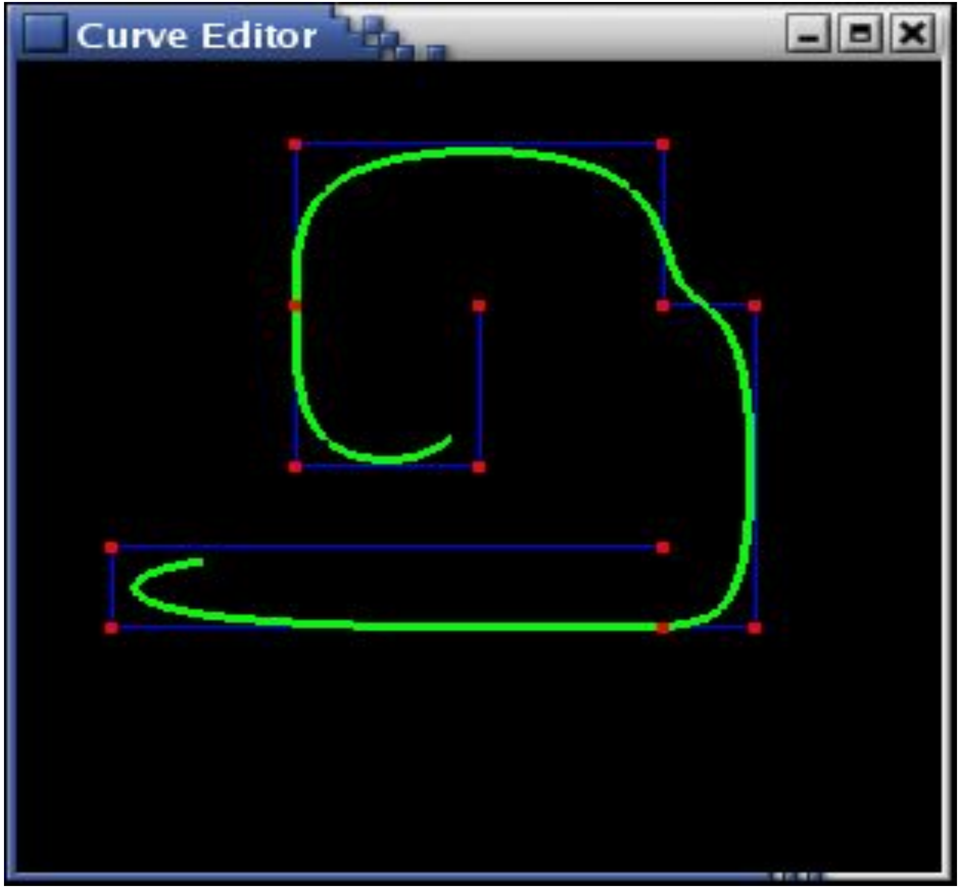
- Can be chained together
- Better control locally (windowing)



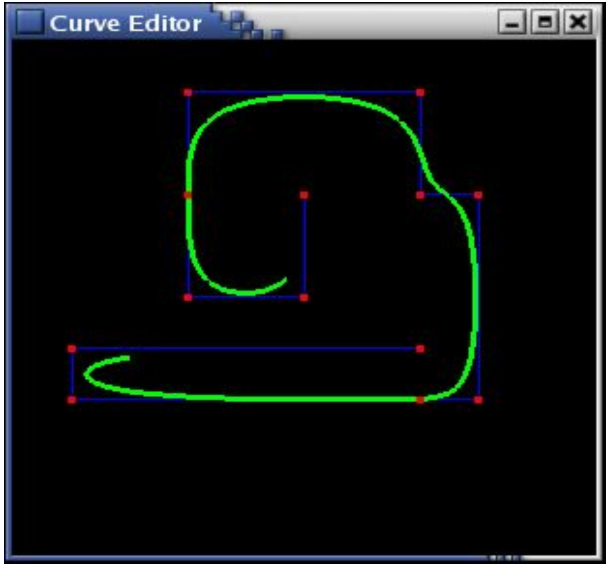


Connecting Cubic BSpline Curves

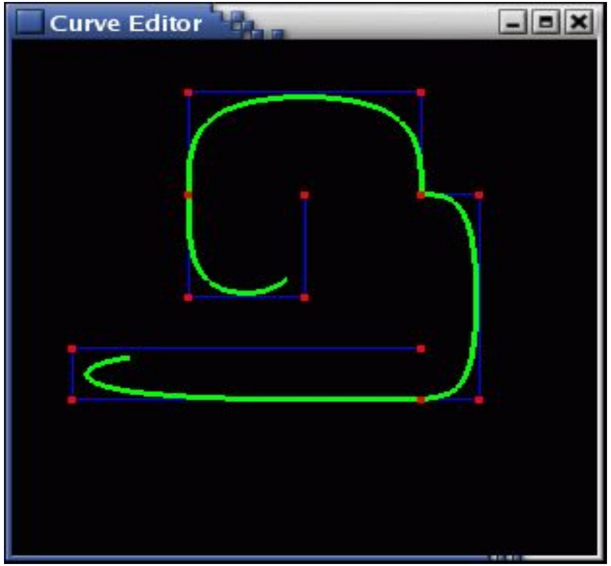
- What's the relationship between:
  - the # of control points, and
  - the # of cubic BSpline subcurves?



BSpline Curve Control Points

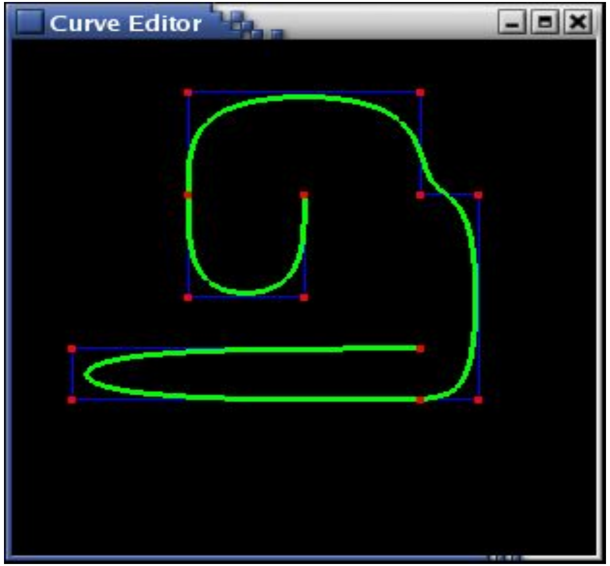


Default BSpline



BSpline with Discontinuity

Repeat interior control point



BSpline which passes through end points

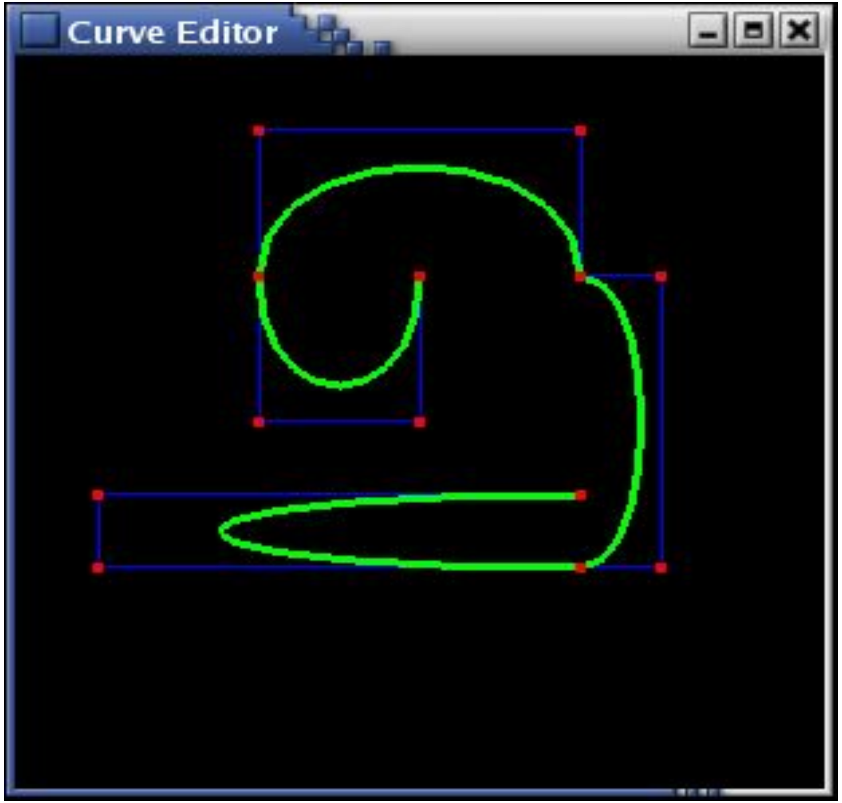
Repeat end points

Questions?

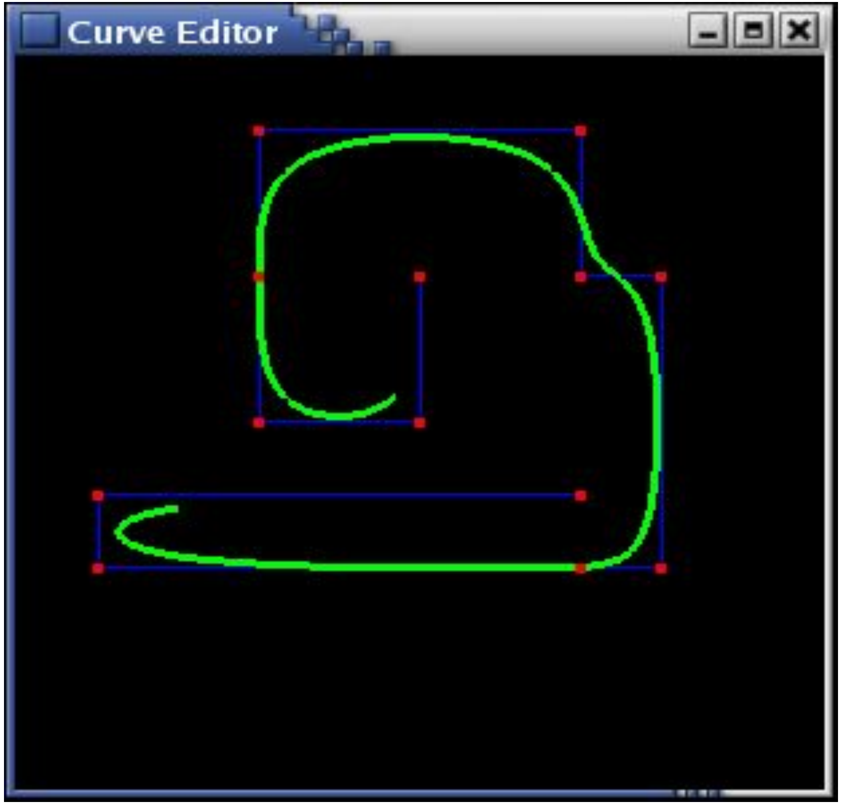
Today

- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Bézier is not the same as BSpline



Bézier

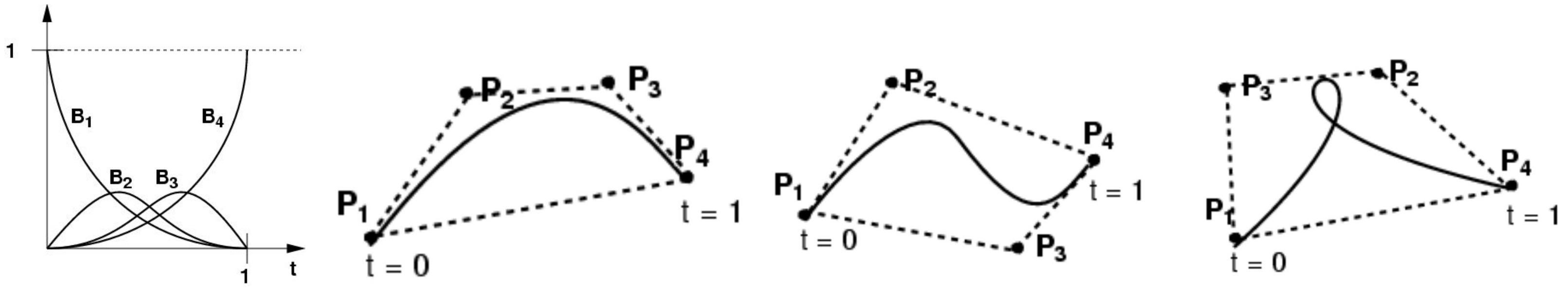


BSpline

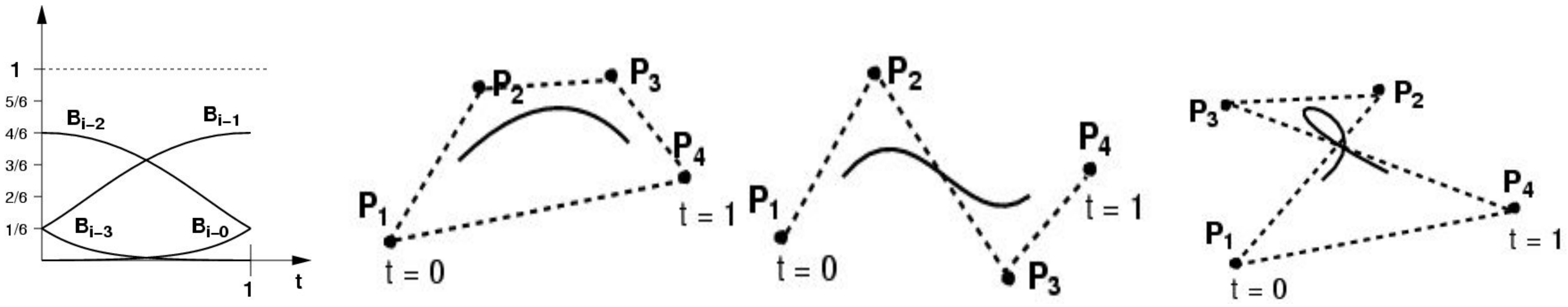
Bézier is not the same as BSpline

- Relationship to the control points is different

Bézier

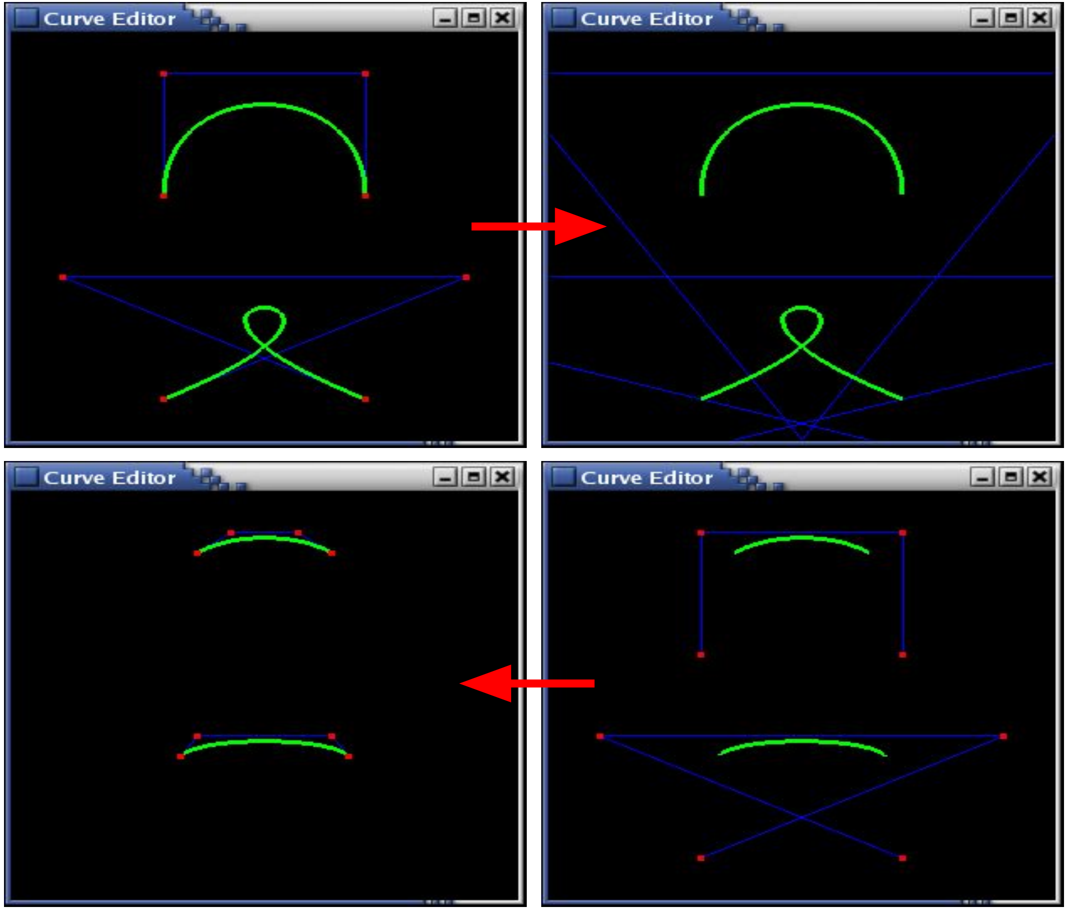


BSpline



Converting between Bézier & BSpline

original control points as Bézier



new BSpline control points to match Bézier

new Bézier control points to match BSpline

original control points as BSpline

Converting between Bézier & BSpline

Using the basis functions:

$Q(t)$ = Geometry **G** · Spline Basis **B** · Power Basis $\mathbf{T}(t)$ = $\mathbf{GBT}(t)$

$$G_{Bezier} \cdot B_{Bezier} \cdot T = G_{BSpline} \cdot B_{BSpline} \cdot T$$

$$G_{Bezier} = \frac{G_{BSpline} \cdot B_{BSpline} \cdot T}{B_{Bezier} \cdot T}$$

$$B_{Bezier} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$B_{B-Spline} = \frac{1}{6}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

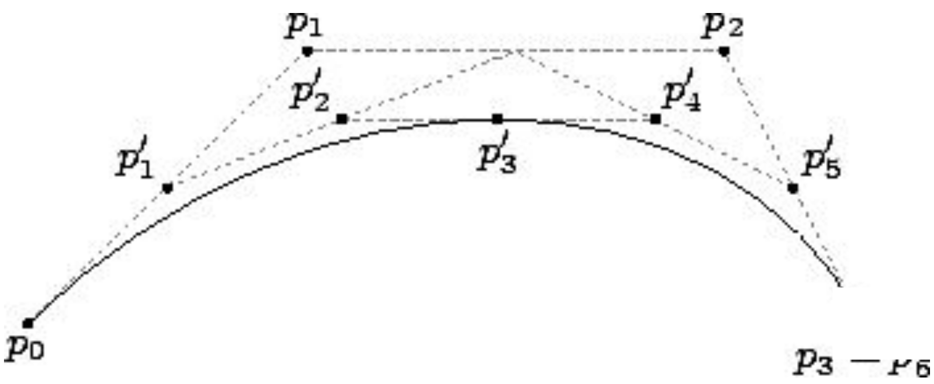
NURBS (generalized BSplines)

- BSpline: uniform cubic BSpline

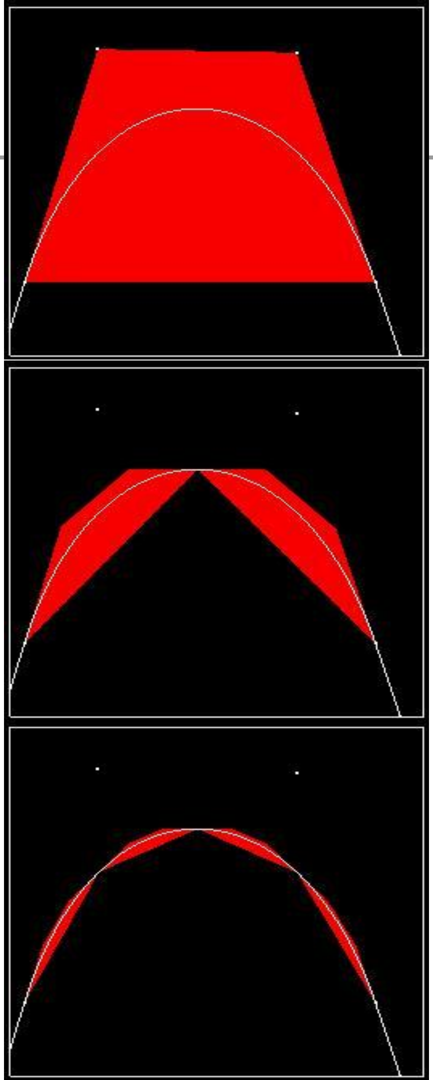
- NURBS: Non-Uniform Rational BSpline
  - non-uniform = different spacing between the blending functions, a.k.a. knots
  - rational = ratio of polynomials (instead of cubic)

Neat Bezier Spline Trick

- A Bezier curve with 4 control points:
  - $P_0 \quad P_1 \quad P_2 \quad P_3$
- Can be split into 2 new Bezier curves:
  - $P_0 \quad P'_1 \quad P'_2 \quad P'_3$
  - $P'_3 \quad P'_4 \quad P'_5 \quad P_3$



A Bézier curve is bounded by the convex hull of its control points.



Today

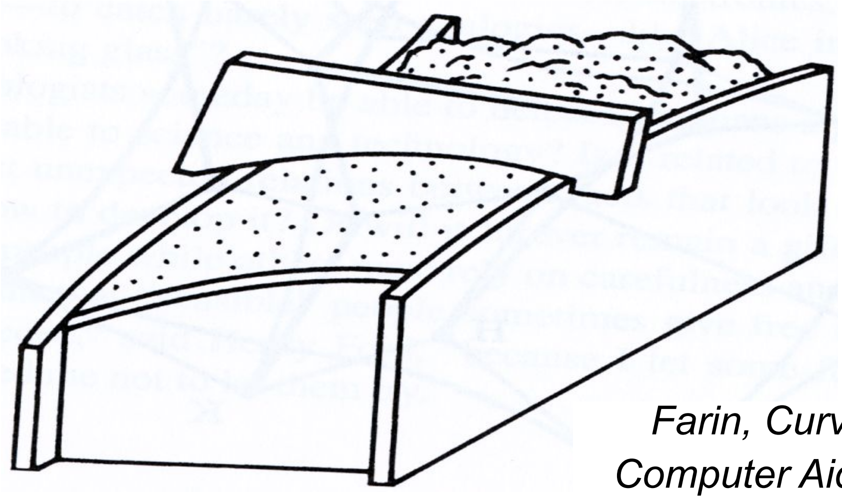
- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Spline Surface via Tensor Product

- Of two vectors:

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \otimes \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_2b_1 & a_3b_1 \\ a_1b_2 & a_2b_2 & a_3b_2 \\ a_1b_3 & a_2b_3 & a_3b_3 \\ a_1b_4 & a_2b_4 & a_3b_4 \end{bmatrix}$$

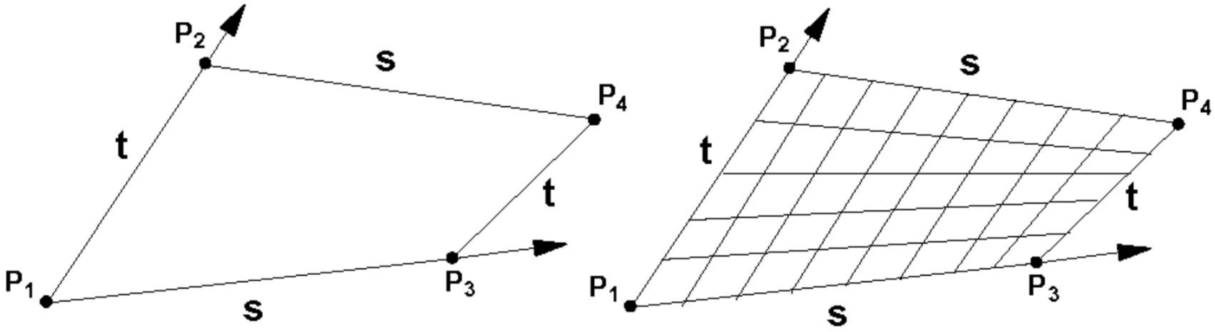
- Similarly, we can define a surface as the tensor product of two curves....



Farin, Curves and Surfaces for Computer Aided Geometric Design

Bilinear Patch

- 1D Linear Interpolation: $\mathbf{L}\,(P_1, P_2, \alpha) = (1 - \alpha)\, P_1 + \alpha\, P_2$
- 2D Bilinear Interpolation: $\mathbf{Q}\,(s,t) = \mathbf{L}\,(\,\mathbf{L}\,(P_1, P_2, t),\ \mathbf{L}(P_3, P_4, t), s)$



- Bilinear Interpolation creates non-planar quadrilaterals (if P_1, P_2, P_3, P_4 are not co-planar)
- But will this help us model smooth surfaces?
- Do we have control of the derivative at the edges?

Ruled Surfaces in Art & Architecture

http://www.bergenwood.no/wp-content/media/images/frozenmusic.jpg

Chiras Iulia
Astri Isabella
Matiss Shteinerts

http://www.lonelyplanetimages.com/images/399954

Antoni Gaudi
Children's School
Barcelona

Bicubic Bézier Patch

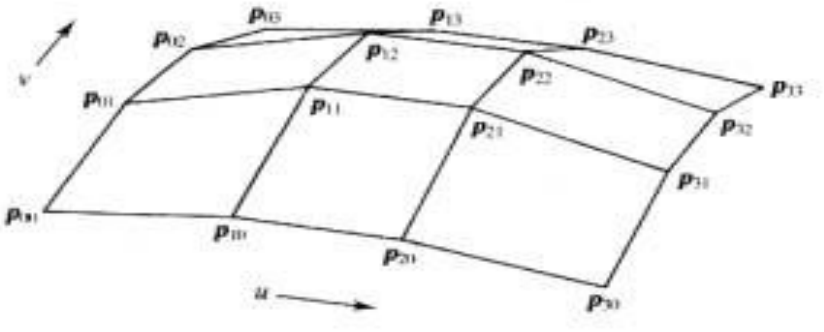
Bézier Curve: **CB** $(P_1, P_2, P_3, P_4, \alpha)$

Bézier Surface: $\boldsymbol{Q}(s,t) = \boldsymbol{CB}($ $\boldsymbol{CB}(P1, P2, P3, P4, t),$

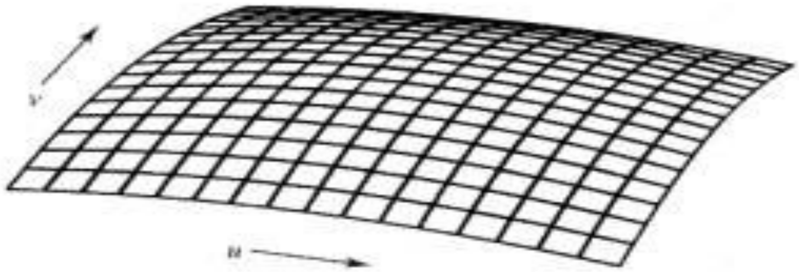
$(\boldsymbol{CB}(P1, P2, P3, P4, t),$

$(\boldsymbol{CB}(P1, P2, P3, P4, t),$

$(\boldsymbol{CB}(P1, P2, P3, P4, t), s)$

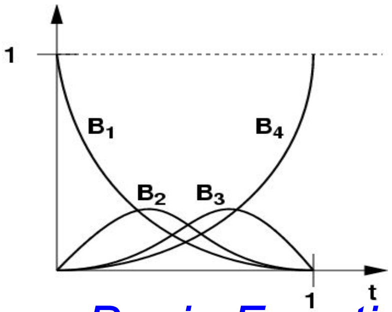


(a)

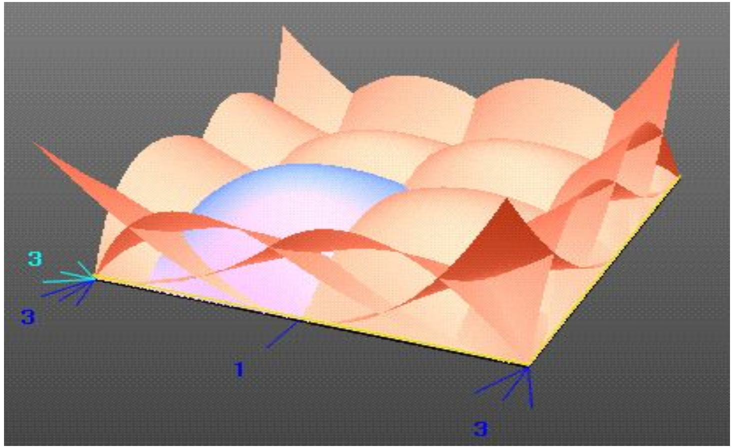


(b)

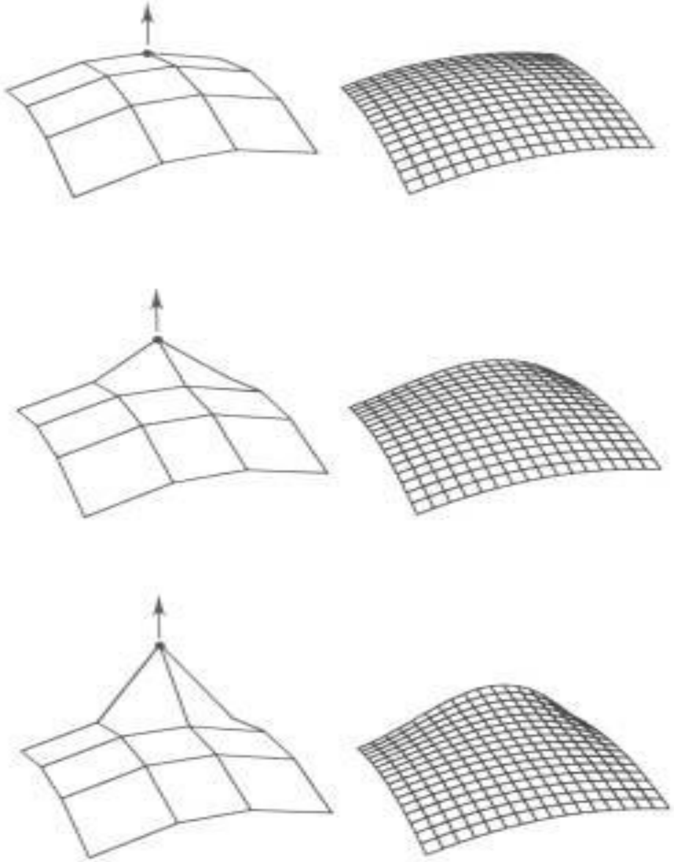
Editing Bicubic Bezier Patches



Curve Basis Functions

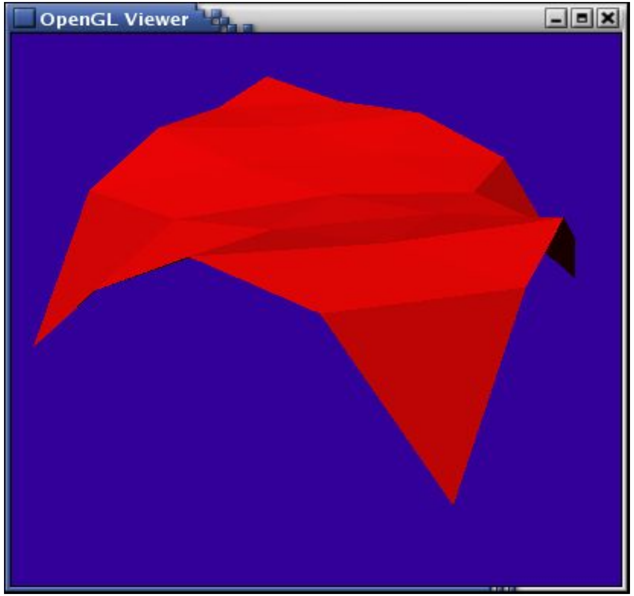


Surface Basis Functions

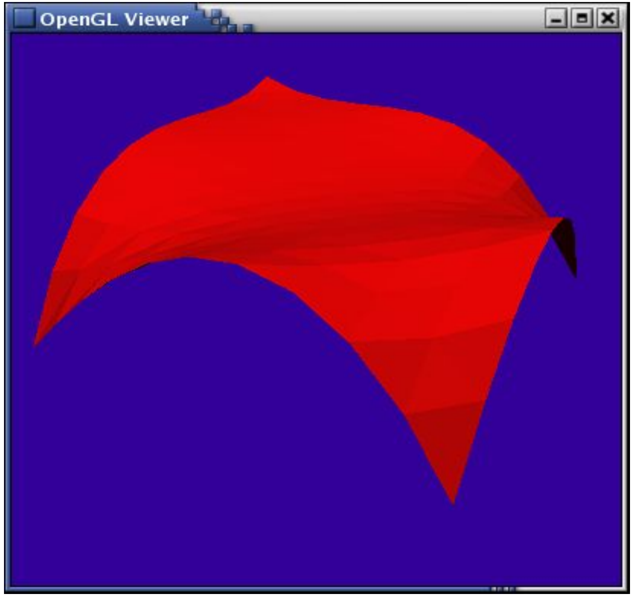


Bicubic Bezier Patch Tessellation

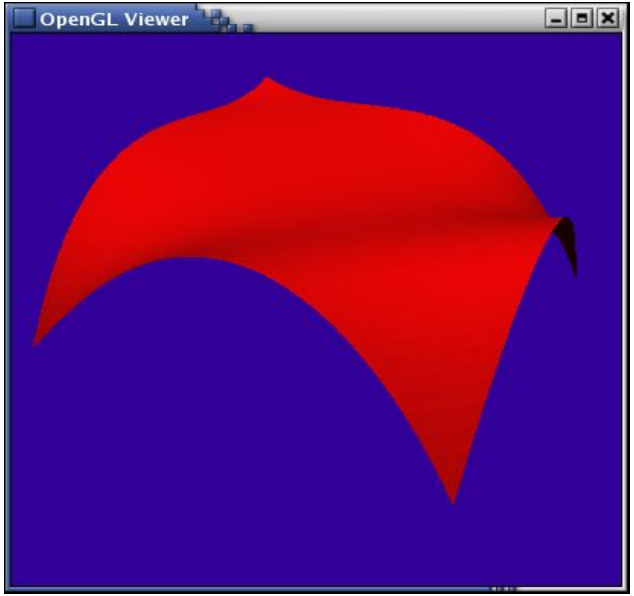
- Given 16 control points and a tessellation resolution, we can create a triangle mesh



resolution: 5x5 vertices



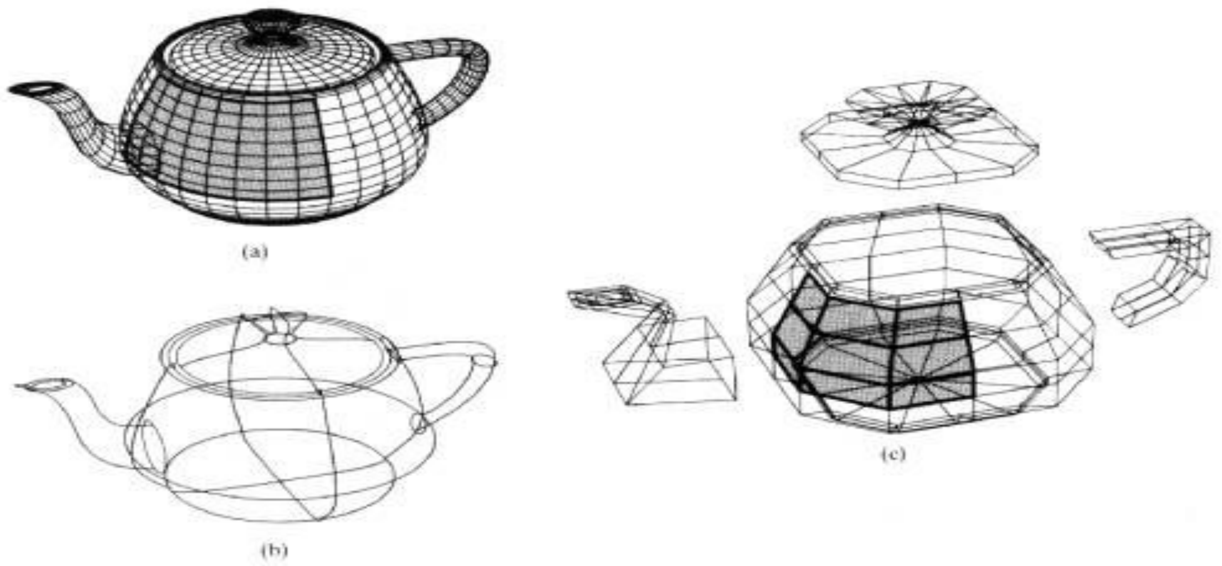
resolution: 11x11 vertices



resolution: 41x41 vertices

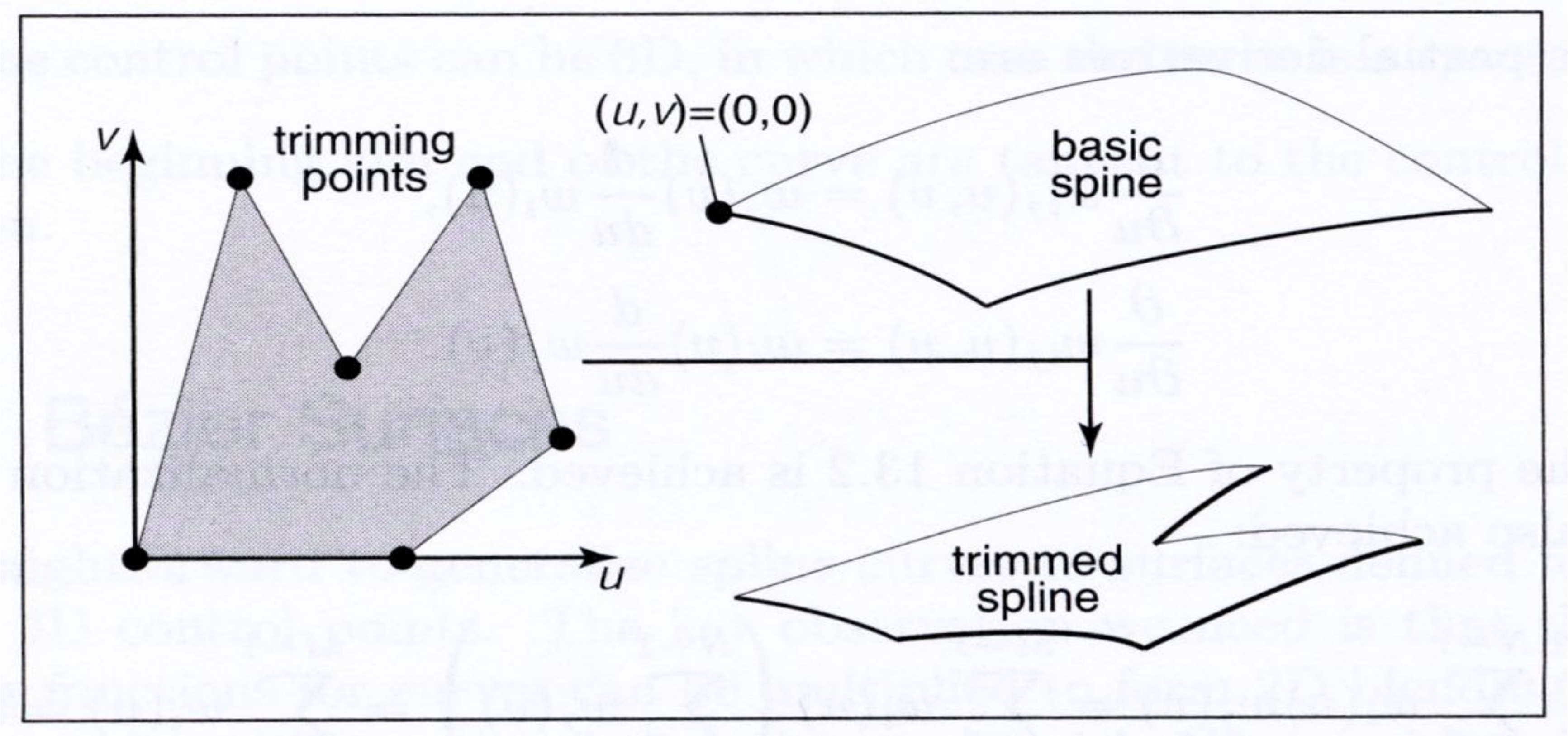
Modeling with Bicubic Bezier Patches

- Original Teapot specified with Bezier Patches:



- But it’s not "watertight": it has intersecting surfaces at spout & handle, no bottom, a hole at the spout tip, a gap between lid & base

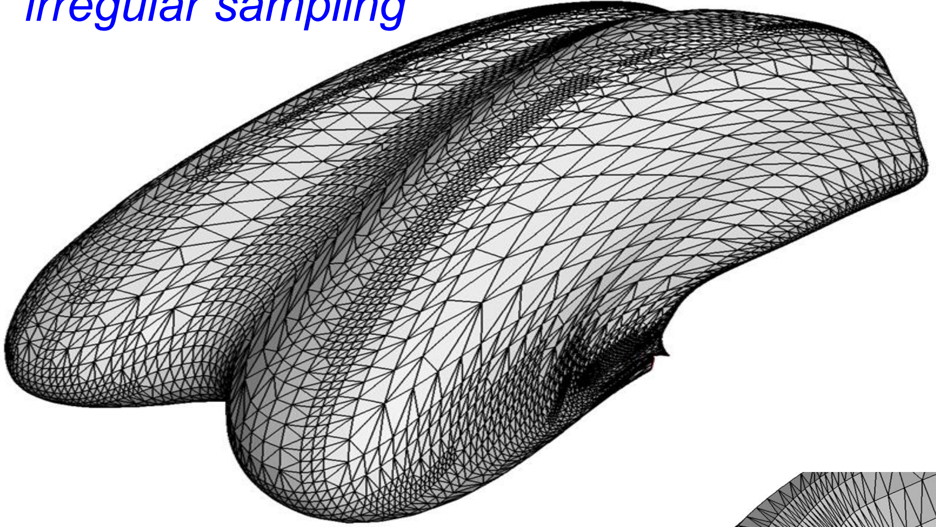
Trimming Curves for Patches

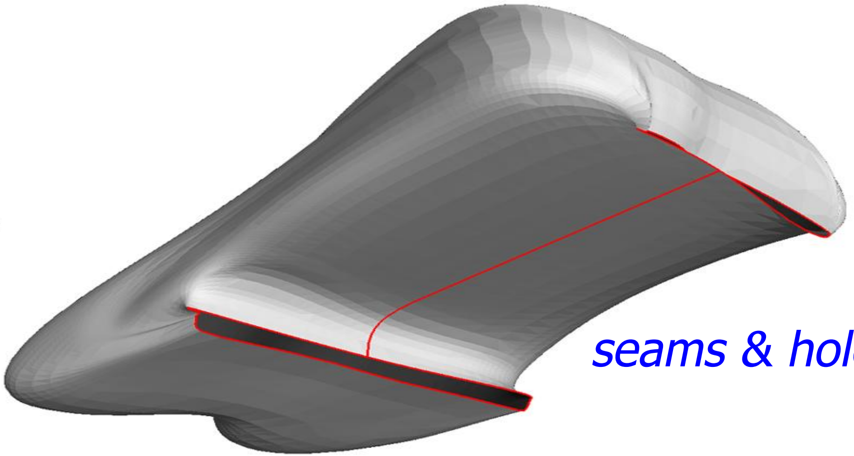


Shirley, Fundamentals of Computer Graphics

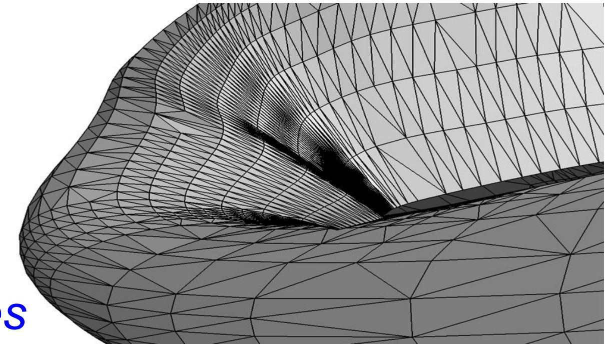
Spline-Based Modeling Headaches

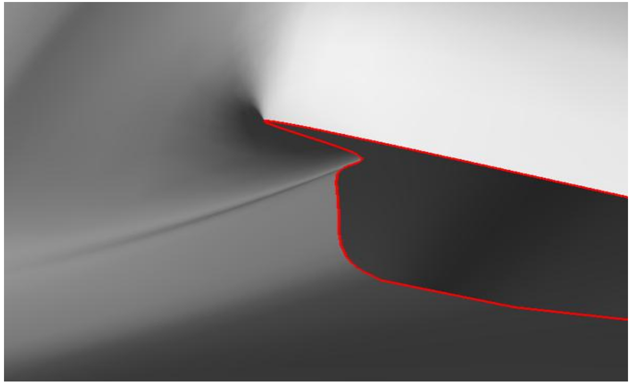
irregular sampling





seams & holes





“pinched” surfaces

Questions?

- Bezier Patches?

  or

- Triangle Mesh?

Henrik Wann Jensen

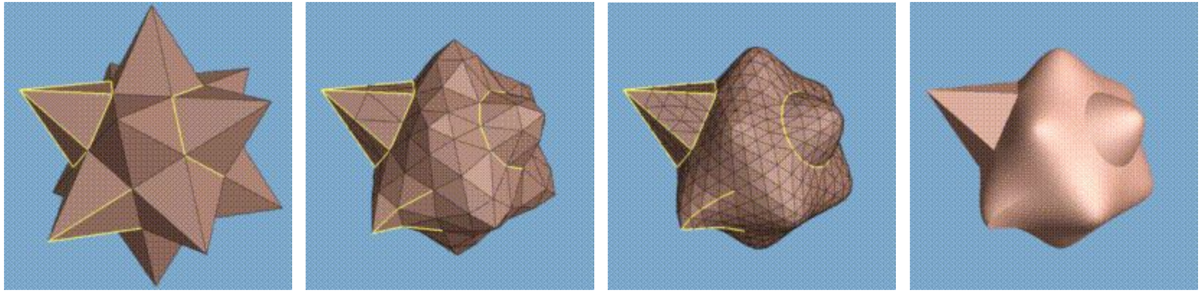
Today

- Reading (from last time): “Progressive Meshes”
- Reading: “Teddy: A Sketching Interface for 3D Freeform Design”
- Limitations of Polygonal Models
- What's a Spline?
- Bézier Spline
- BSpline (NURBS)
- Bézier ≠ BSpline!
- Extending to Surfaces
- Papers for Friday

Readings for Next Time *(pick one)*

- Hoppe et al., “Piecewise Smooth Surface Reconstruction” SIGGRAPH 1994

Triangle Meshes
directly applies to HW1!



- DeRose, Kass, & Truong, "Subdivision Surfaces in Character Animation", SIGGRAPH 1998

Quad Meshes
more common in artistic practice
(e.g. Pixar’s Geri’s Game)

Homework 1