CS-184: Computer Graphics

Lecture #12: Curves and Surfaces

Prof. James O'Brien
University of California, Berkeley

V2008-F-12-1.0

1

Today

- General curve and surface representations
- Splines and other polynomial bases

2

2

Geometry Representations

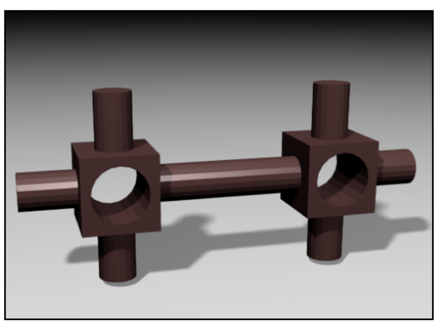
- Constructive Solid Geometry (CSG)
- Parametric
  - Polygons
  - Subdivision surfaces
- Implicit Surfaces
- Point-based Surface

- Not always clear distinctions
  - *i.e.* CSG done with implicits

3

3

Geometry Representations



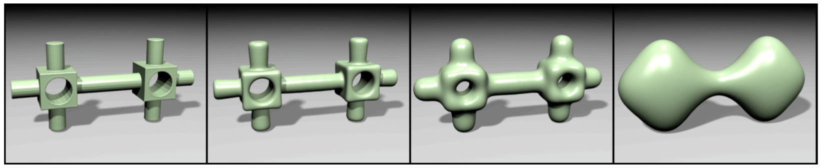
Object made by CSG
Converted to polygons

4

4

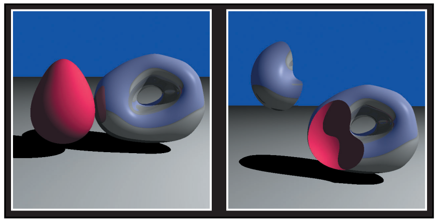
Geometry Representations

Object made by CSG
Converted to polygons
Converted to implicit surface



5

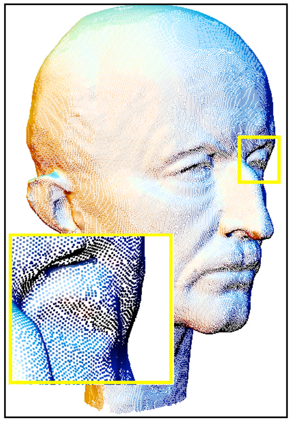
Geometry Representations



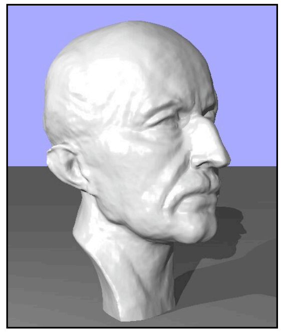
CSG on implicit surfaces

6

Geometry Representations



Point-based surface descriptions



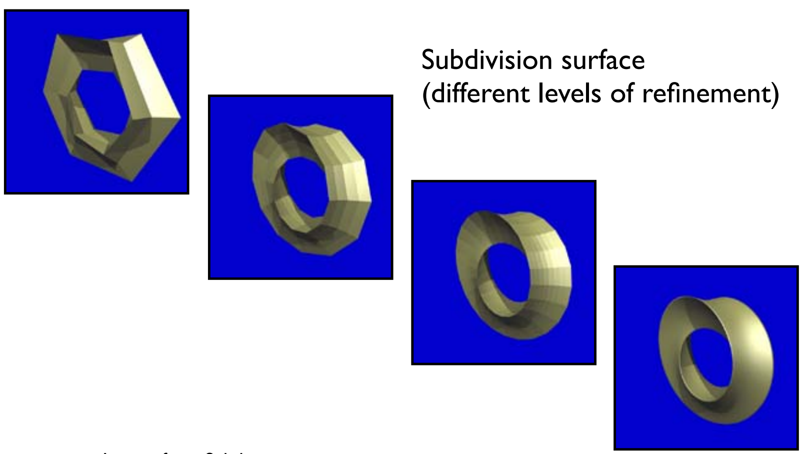
Ohtake, *et al.*, SIGGRAPH 2003

7

7

Geometry Representations

Subdivision surface
(different levels of refinement)



Images from Subdivision.org

8

8

Geometry Representations

- Various strengths and weaknesses
  - Ease of use for design
  - Ease/speed for rendering
  - Simplicity
  - Smoothness
  - Collision detection
  - Flexibility (in more than one sense)
  - Suitability for simulation
  - *many others...*

9

9

Parametric Representations

Curves: $\boldsymbol{x} = \boldsymbol{x}(u) \quad \boldsymbol{x} \in \Re^n \quad u \in \Re$

Surfaces: $\boldsymbol{x} = \boldsymbol{x}(u, v) \quad \boldsymbol{x} \in \Re^n \quad u, v \in \Re$

$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{u}) \quad \boldsymbol{u} \in \Re^2$

Volumes: $\boldsymbol{x} = \boldsymbol{x}(u, v, w) \quad \boldsymbol{x} \in \Re^n \quad u, v, w \in \Re$

$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{u}) \quad \boldsymbol{u} \in \Re^3$

and so on...

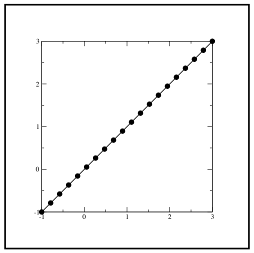
Note: a vector function is really n scalar functions

10

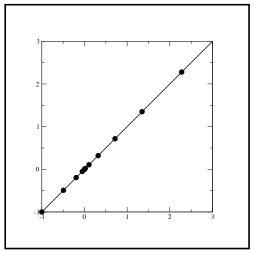
10

Parametric Rep. Non-unique

- Same curve/surface may have multiple formulae



$\boldsymbol{x}(u) = [u, u]$



$\boldsymbol{x}(u) = [u^3, u^3]$

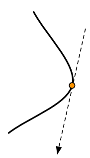
11

11

Simple Differential Geometry

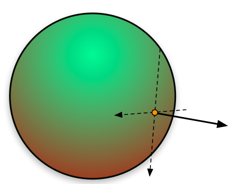
- Tangent to curve

$$\boldsymbol{t}(u) = \left.\frac{\partial \boldsymbol{x}}{\partial u}\right|_u$$



- Tangents to surface

$$\boldsymbol{t}_u(u,v) = \left.\frac{\partial \boldsymbol{x}}{\partial u}\right|_{u,v} \qquad \boldsymbol{t}_v(u,v) = \left.\frac{\partial \boldsymbol{x}}{\partial v}\right|_{u,v}$$



- Normal of surface

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{t}_u \times \boldsymbol{t}_v}{||\boldsymbol{t}_u \times \boldsymbol{t}_v||}$$

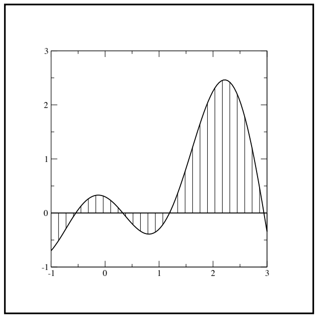
- Also: curvature, curve normals, curve bi-normal, *others...*
- Degeneracies: $\partial \boldsymbol{x} / \partial u = 0$ or $\boldsymbol{t}_u \times \boldsymbol{t}_v = 0$

12

12

Discretization

- Arbitrary curves have an uncountable number of parameters



i.e. specify function value at all points on real number line

13

13

Discretization

- Arbitrary curves have an uncountable number of parameters
- Pick *complete* set of basis functions
  - Polynomials, Fourier series, *etc.*

$$x(u) = \sum_{i=0}^{\infty} c_i \phi_i(u)$$

- Truncate set at some reasonable point

$$x(u) = \sum_{i=0}^{3} c_i \phi_i(u) = \sum_{i=0}^{3} c_i \, u^i$$

- Function represented by the vector (list) of c_i
- The c_i may themselves be vectors

$$\boldsymbol{x}(u) = \sum_{i=0}^{3} \boldsymbol{c}_i \phi_i(u)$$

14

14

Polynomial Basis

- Power Basis

$$x(u) = \sum_{i=0}^{d} c_i\, u^i$$

$$x(u) = \boldsymbol{C} \cdot \boldsymbol{\mathcal{P}}^d \qquad \boldsymbol{C} = [c_0, c_1, c_2, \ldots, c_d] \quad \boldsymbol{\mathcal{P}}^d = [1, u, u^2, \ldots, u^d]$$

The elements of $\mathcal{P}^d$ are *linearly independant*

i.e. no good approximation

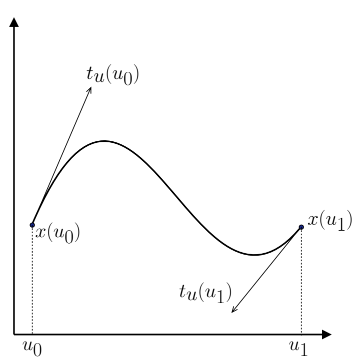
$$u^k \not\approx \sum_{i \neq k} c_i\, u^i$$

Skipping something would lead to bad results... odd stiffness

Specifying a Curve

Given desired values (constraints) how do we determine the coefficients for cubic power basis?

For now, assume
$u_0 = 0 \quad u_1 = 1$



Specifying a Curve

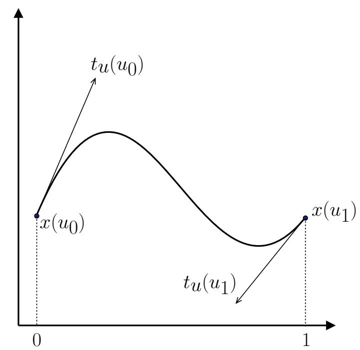
Given desired values (constraints) how do we determine the coefficients for cubic power basis?

$$x(0) = c_0 \quad = x_0$$

$$x(1) = \Sigma\, c_i \;= x_1$$

$$x'(0) = c_1 \quad = x'_0$$

$$x'(1) = \Sigma\, i\, c_i = x'_1$$



17

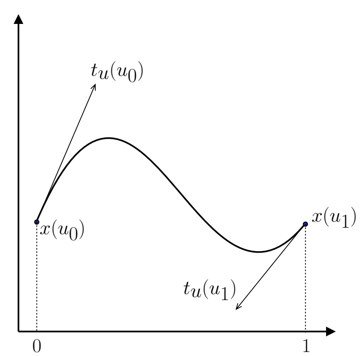
17

Specifying a Curve

Given desired values (constraints) how do we determine the coefficients for cubic power basis?

$$\begin{bmatrix} x_0 \\ x_1 \\ x'_0 \\ x'_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$\mathbf{p} = \mathbf{B} \cdot \mathbf{c}$$



18

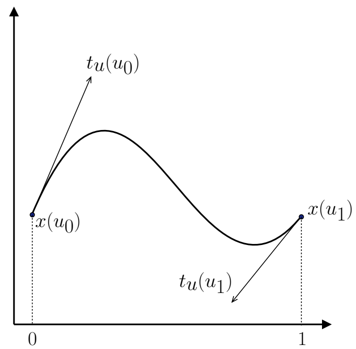
18

Specifying a Curve

Given desired values (constraints) how do we determine the coefficients for cubic power basis?

$$\mathbf{c} = \beta_{\text{H}} \cdot \mathbf{p}$$

$$\boldsymbol{\beta}_{\text{H}} = \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & 1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$



19

19

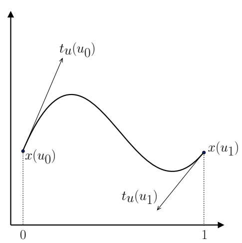
Specifying a Curve

Given desired values (constraints) how do we determine the coefficients for cubic power basis?

$$\mathbf{c} = \beta_{\text{H}} \cdot \mathbf{p}$$

$$x(u) = \mathcal{P}^3 \cdot \mathbf{c} = \boxed{\mathcal{P}^3 \boldsymbol{\beta}_{\text{H}}} \mathbf{p}$$

$$= \begin{bmatrix} 1 + 0u - 3u^2 + 2u^3 \\ 0 + 0u + 3u^2 - 2u^3 \\ 0 + 1u - 2u^2 + 1u^3 \\ 0 + 0u - 1u^2 + 1u^3 \end{bmatrix} \mathbf{p}$$



$$\boldsymbol{\beta}_{\text{H}} = \mathbf{B}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & 1 \\ 2 & -2 & 1 & 1 \end{bmatrix}$$

20

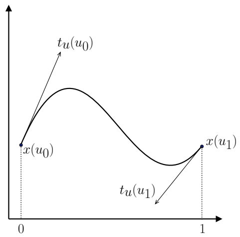
20

Specifying a Curve

Given desired values (constraints) how do we determine the coefficients for cubic power basis?

$$\mathbf{c} = \beta_{\text{H}} \cdot \mathbf{p}$$

$$x(u) = \begin{bmatrix} 1 + 0u - 3u^2 + 2u^3 \\ 0 + 0u + 3u^2 - 2u^3 \\ 0 + 1u - 2u^2 + 1u^3 \\ 0 + 0u - 1u^2 + 1u^3 \end{bmatrix} \mathbf{p}$$

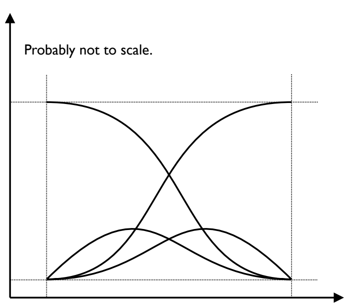


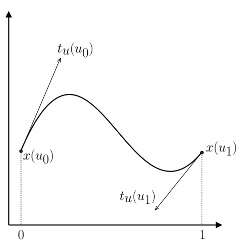
$$x(u) = \sum_{i=0}^{3} p_i b_i(u)$$

Hermite basis functions

Specifying a Curve

Given desired values (constraints) how do we determine the coefficients for cubic power basis?





$$x(u) = \sum_{i=0}^{3} p_i b_i(u)$$

Hermite basis functions

Hermite Basis

- Specify curve by
  - Endpoint values
  - Endpoint tangents (derivatives)
- Parameter interval is arbitrary (most times)
  - Don't need to recompute basis functions
- These are *cubic* Hermite
  - Could do construction for any odd degree
  - $(d-1)/2$ derivatives at end points

23

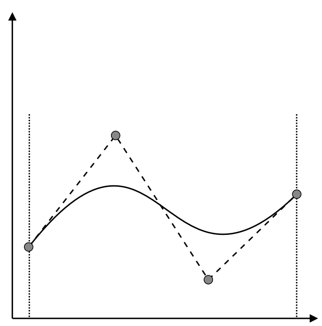
23

Cubic Bézier

- Similar to Hermite, but specify tangents indirectly

$$x_0 = p_0$$
$$x_1 = p_3$$
$$x_0' = 3(p_1 - p_0)$$
$$x_1' = 3(p_3 - p_2)$$

Note: all the control points are points in space, no tangents.



24

24

Cubic Bézier

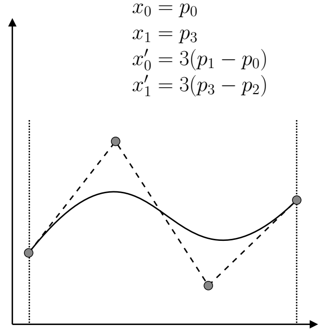
- Similar to Hermite, but specify tangents indirectly

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \mathbf{c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \mathbf{p}$$

$$\mathbf{c} = \boxed{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}} \mathbf{p}$$

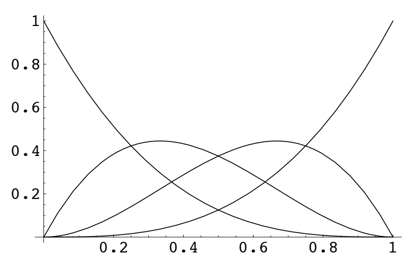
$$\mathbf{c} = \boldsymbol{\beta}_Z \, \mathbf{p}$$

$$x_0 = p_0$$
$$x_1 = p_3$$
$$x_0' = 3(p_1 - p_0)$$
$$x_1' = 3(p_3 - p_2)$$



Cubic Bézier

- Plot of Bézier basis functions



Changing Bases

- Power basis, Hermite, and Bézier all are still just cubic polynomials
  - The three basis sets all span the same space
  - Like different axes in $\Re^{X}$ $\Re^4$
- Changing basis

$$\mathbf{c} = \boldsymbol{\beta}_{\mathrm{Z}}\,\mathbf{p}_{\mathrm{Z}}$$

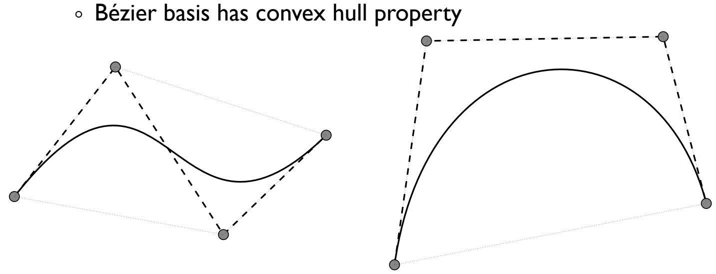
$$\mathbf{c} = \boldsymbol{\beta}_{\mathrm{H}}\,\mathbf{p}_{\mathrm{H}}$$

$$\mathbf{p}_{\mathrm{Z}} = \boldsymbol{\beta}_{\mathrm{Z}}^{-1}\,\boldsymbol{\beta}_{\mathrm{H}}\,\mathbf{p}_{\mathrm{H}}$$

27

Useful Properties of a Basis

- Convex Hull
  - All points on curve inside convex hull of control points
    - $\sum_i b_i(u) = 1 \qquad b_i(u) \geq 0 \qquad \forall u \in \Omega$
    - Bézier basis has convex hull property



28

Useful Properties of a Basis

- Invariance under class of transforms
  - Transforming curve is same as transforming control points
    - $\boldsymbol{x}(u) = \sum_i \boldsymbol{p}_i b_i(u) \Leftrightarrow \mathcal{T}\boldsymbol{x}(u) = \sum_i (\mathcal{T}\boldsymbol{p}_i) b_i(u)$
    - Bézier basis invariant for affine transforms
    - Bézier basis NOT invariant for perspective transforms
      - NURBS are though...

29

29

Useful Properties of a Basis

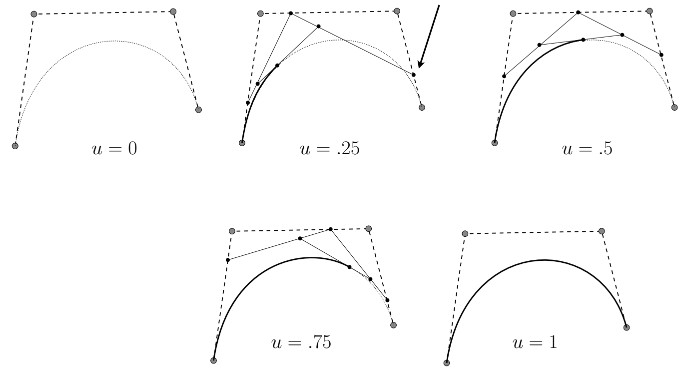
- Local support
  - Changing one control point has limited impact on entire curve
  - Nice subdivision rules
  - Orthogonality ($\int_\Omega b_i(u) b_j(u) \mathrm{d}u = \delta_{ij}$)
  - Fast evaluation scheme
  - Interpolation -vs- approximation

30

30

DeCasteljau Evaluation

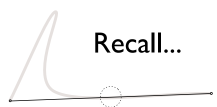
- A geometric evaluation scheme for Bézier



Adaptive Tessellation

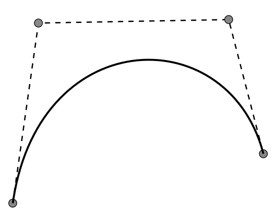
- Midpoint test subdivision
- Possible problem
  - Simple solution if curve basis has *convex hull* property

Recall...



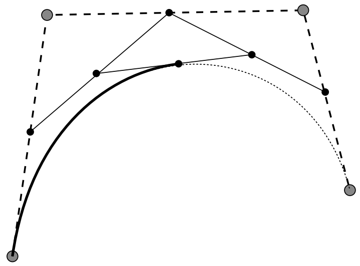
If curve inside convex hull and the convex hull is nearly flat: curve is nearly flat and can be drawn as straight line

Better: draw convex hull
Works for Bézier because the ends are interpolated



Bézier Subdivision

- Form control polygon for half of curve by evaluating at u=0.5

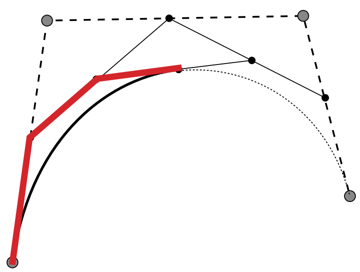


33

33

Bézier Subdivision

- Form control polygon for half of curve by evaluating at u=0.5



33

33

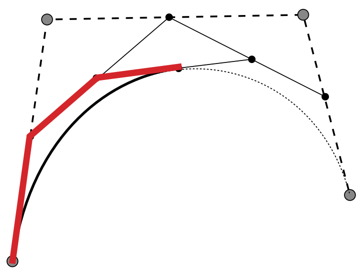
Bézier Subdivision

- Form control polygon for half of curve by evaluating at u=0.5

Repeated subdivision makes smaller/flatter segments

Also works for surfaces...

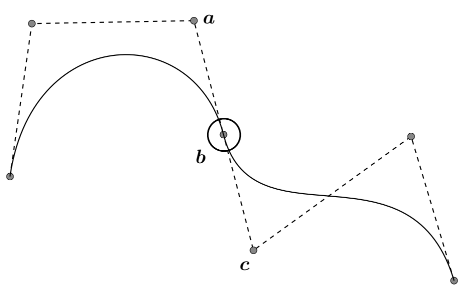
We'll extend this idea later on...



33

33

Joining



$\mathcal{C}^0 \Leftrightarrow \boldsymbol{b} = \boldsymbol{b}$

$\mathcal{C}^1 \Leftrightarrow \boldsymbol{b} - \boldsymbol{a} = \boldsymbol{c} - \boldsymbol{b}$

$\mathcal{G}^1 \Leftrightarrow \dfrac{\boldsymbol{b} - \boldsymbol{a}}{||\boldsymbol{b} - \boldsymbol{a}||} = \dfrac{\boldsymbol{c} - \boldsymbol{b}}{||\boldsymbol{c} - \boldsymbol{b}||}$

If you change $\boldsymbol{a}$, $\boldsymbol{b}$, or $\boldsymbol{c}$ you must change the others

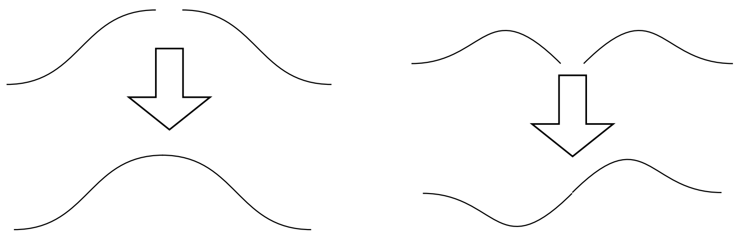
But if you change $\boldsymbol{a}$, $\boldsymbol{b}$, or $\boldsymbol{c}$ you do not have to change beyond those three. *LOCAL SUPPORT*

34

34

“Hump” Functions

- Constraints at joining can be built in to make new basis



35

Tensor-Product Surfaces

- Surface is a curve swept through space
- Replace control points of curve with other curves

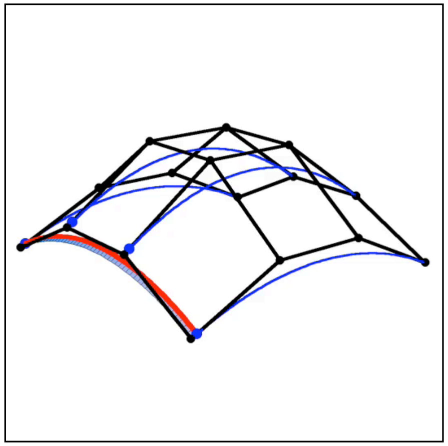
$$x(u,v) = \Sigma_i \, p_i \, b_i(u)$$
$$\Sigma_i \, q_i(v) \, b_i(u) \qquad q_i(v) = \Sigma_j \, p_{ji} \, b_j(v)$$

$$x(u,v) = \sum_{ij} p_{ij} b_i(u) b_j(v) \qquad b_{ij}(u,v) = b_i(u) b_j(v)$$

$$x(u,v) = \sum_{ij} p_{ij} b_{ij}(u,v)$$

36

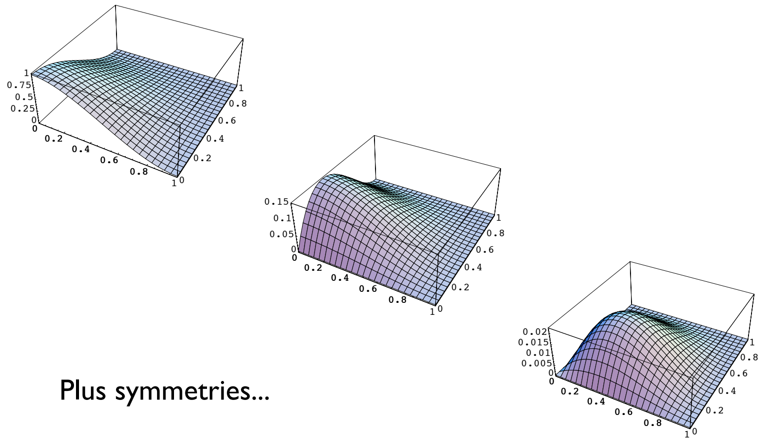
Tensor-Product Surfaces



37

37

Hermite Surface Bases



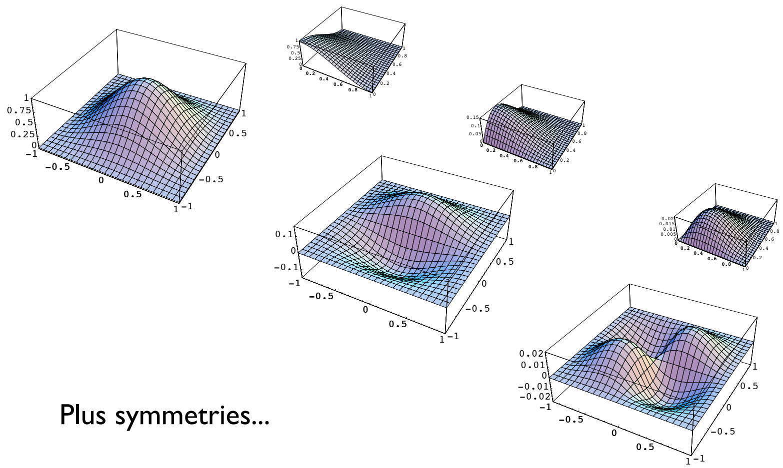
Plus symmetries...

38

38

Wednesday, October 15, 2008

Hermite Surface Hump Functions

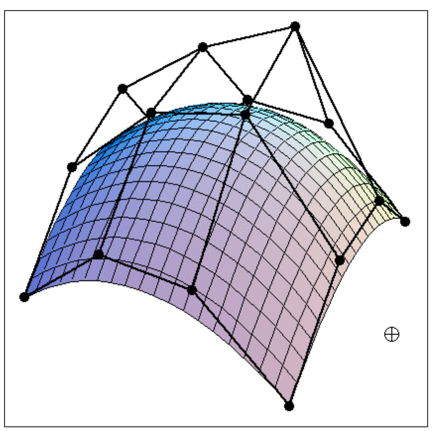


Plus symmetries...

39

39

Bézier Surface Patch

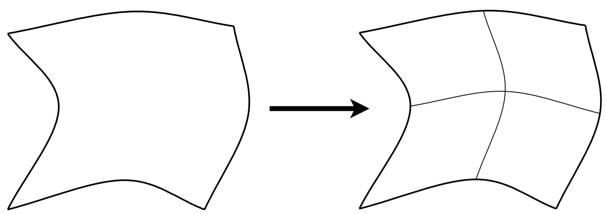


40

40

Adaptive Tessellation

- Given surface patch
  - If close to flat: draw it
  - Else subdivide 4 ways

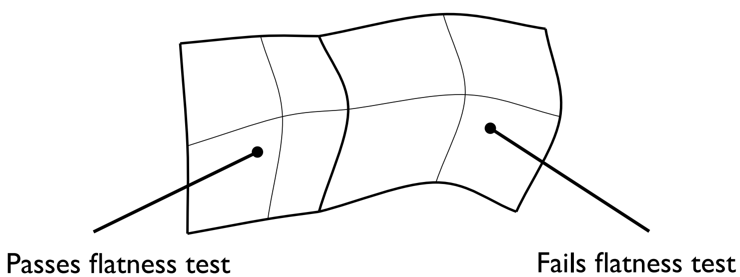


41

41

Adaptive Tessellation

- Avoid cracking

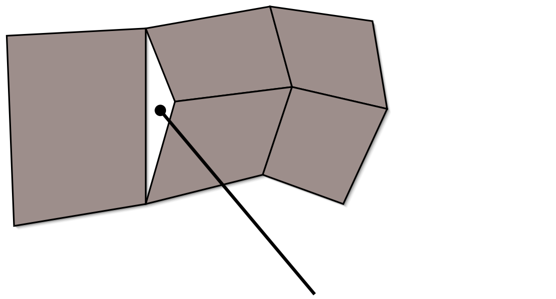


42

42

Adaptive Tessellation

- Avoid cracking



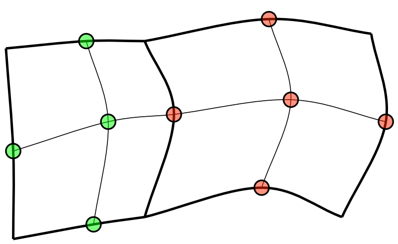
Cracks may be okay in some contexts...

43

43

Adaptive Tessellation

- Avoid cracking

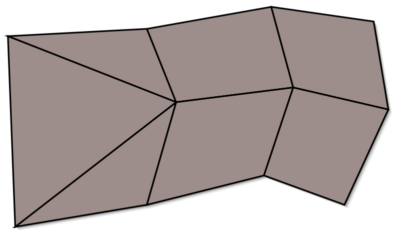


44

44

Adaptive Tessellation

- Avoid cracking



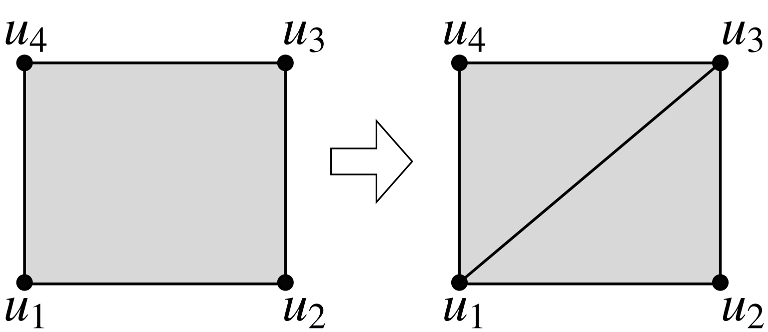
Test interior and boundary of patch
Split boundary based on boundary test
Table of polygon patterns
May wish to avoid "slivers"

45

45

Adaptive Tessellation

- Triangle Based Method (no cracks)

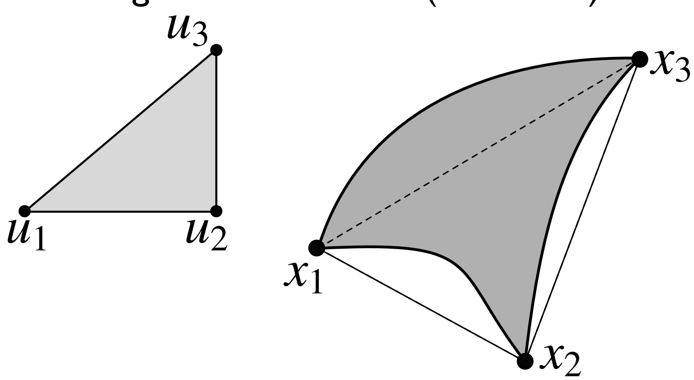


46

46

Adaptive Tessellation

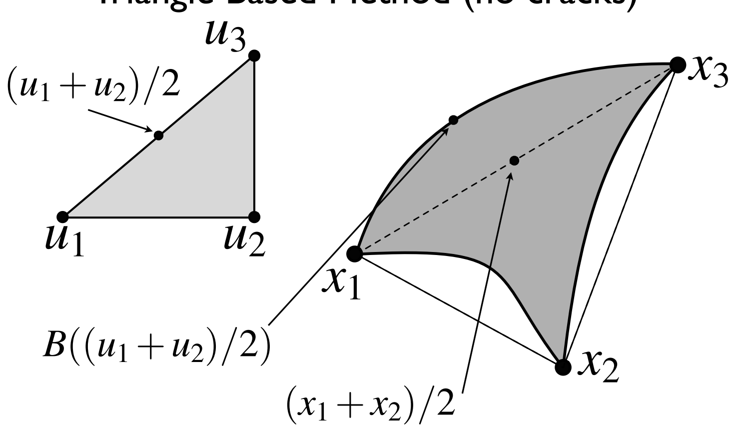
- Triangle Based Method (no cracks)



47

Adaptive Tessellation

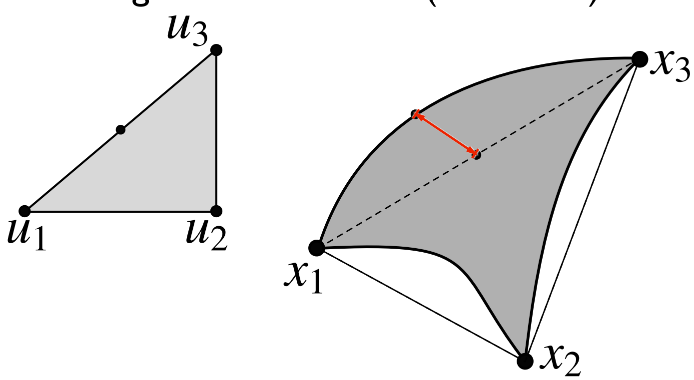
- Triangle Based Method (no cracks)



48

Adaptive Tessellation

- Triangle Based Method (no cracks)



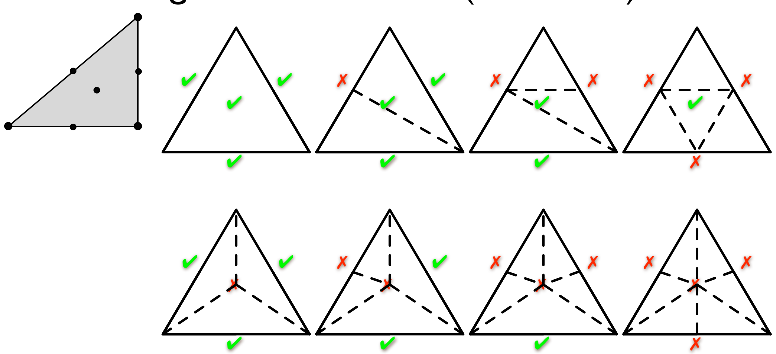
$||B((u_1+u_2)/2)-(x_1+x_2)/2|| < \tau$?

49

49

Adaptive Tessellation

- Triangle Based Method (no cracks)

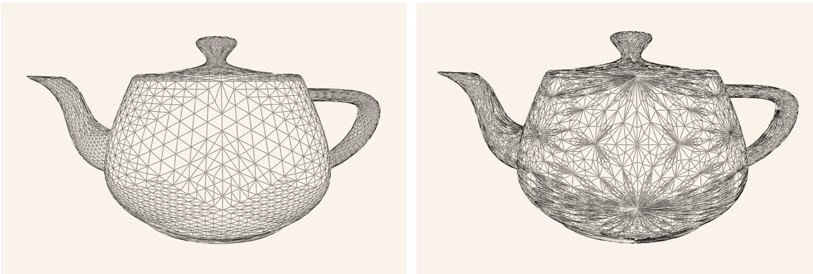


Center test tends to generate slivers.
Often better to leave it out.

50

50

Adaptive Tessellation



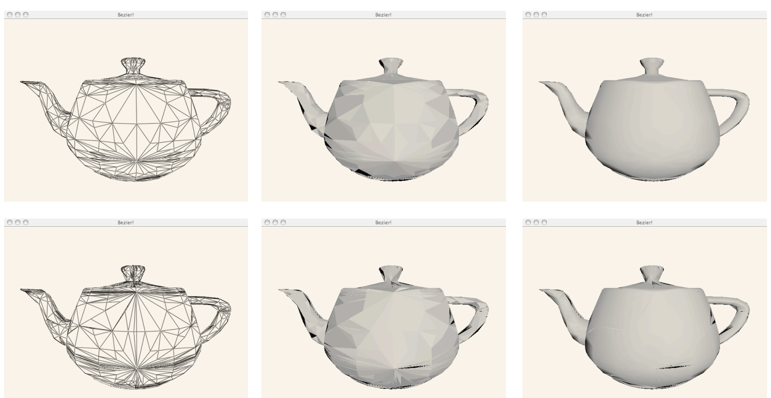
Without center test | With center test

Yiding Jia, CS184 S08

51

51

Adaptive Tessellation



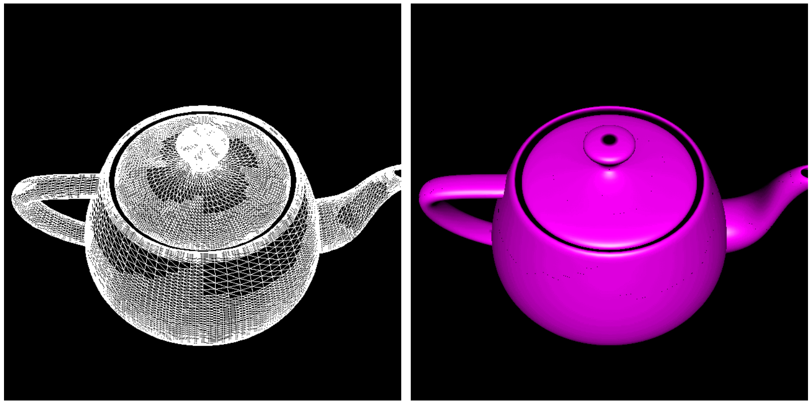
Second row shows typical error of swapping tests.

Yiding Jia, CS184 S08 -- I broke his code to make this example.

52

52

Adaptive Tessellation



Visible artifacts from cracks.

Apollo Ellis, CS184 S08

53