Surfaces and Solids

CS 4620 Lecture 30

Administration

- A4 and PPA2 demos
  - Today
- A5 due on Friday
- Dreamworks visiting Thu/Fri
- Rest of class
  - Surfaces, Animation, Rendering

Modeling in 3D

- Representing subsets of 3D space
  - volumes (3D subsets)
  - surfaces (2D subsets)
  - curves (1D subsets)
  - points (0D subsets)

Representing geometry

- In order of dimension…
- Points: trivial case
- Curves
  - – normally use parametric representation
  - – line—just a point and a vector (like ray in ray tracer)
    - polylines (approximation scheme for drawing)
  - – more general curves: usually use splines
  - **p**(*t*) is from R to R^3
  - **p** is defined by piecewise polynomial functions

Representing geometry

- Surfaces
  - implicit and parametric representations both useful
  - example: plane
    - implicit: vector from point perpendicular to normal
    - parametric: point plus scaled tangents
  - example: sphere
    - implicit: distance from center equals r
    - parametric: write out in spherical coordinates
      - messiness of parametric form not unusual

Specific surface representations

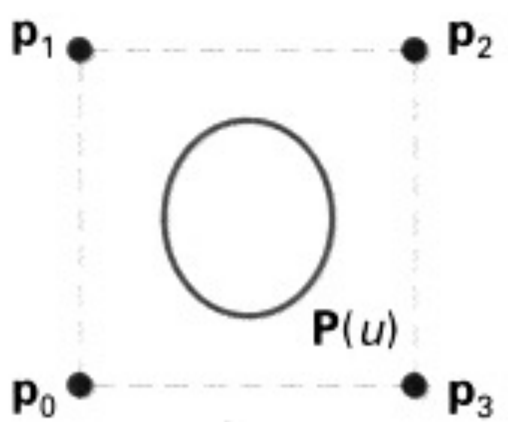
- Parametric spline surfaces
  - extrusions
  - surfaces of revolution
  - generalized cylinders
  - spline patches

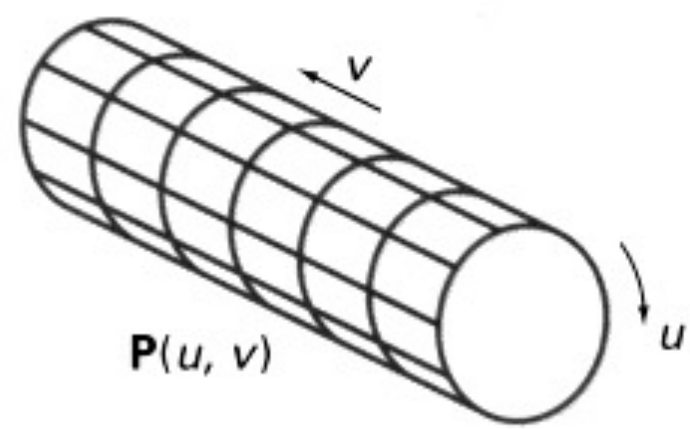
From curves to surfaces

- So far have discussed spline curves in 2D
  - it turns out that this already provides mathematical machinery for several ways of building curved surfaces
- Building surfaces from 2D curves
  - extrusions and surfaces of revolution
- Building surfaces from 2D and 3D curves
  - generalized swept surfaces
- Building surfaces from spline patches
  - generalizing spline curves to spline patches

Extrusions

- Given a spline curve $C \in \mathbb{R}^2$, define $S \in \mathbb{R}^3$ by
  $$S = C \times [a, b]$$
- This produces a "tube" with the given cross section
  - Circle: cylinder; "L": shelf bracket; "I": I beam
- It is parameterized by the spline *t* and the interval [*a, b*]
  $$s(t, s) = [c_x(t), c_y(t), s]^T$$



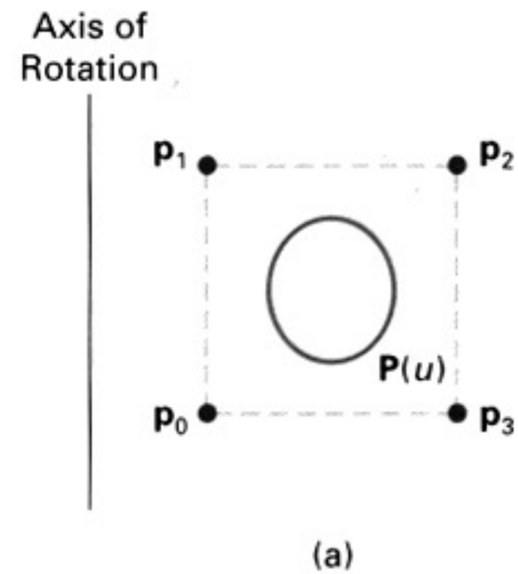


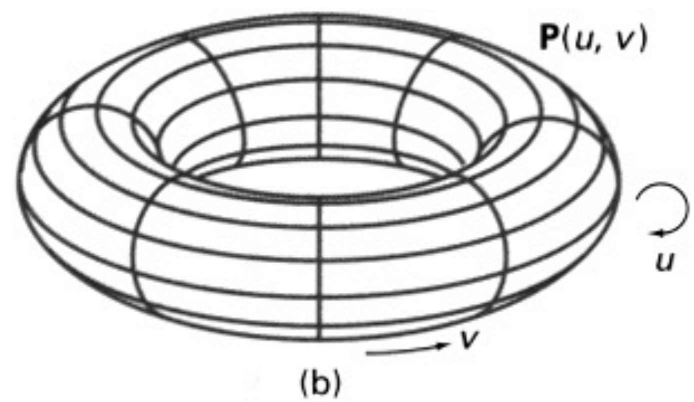
Surfaces of revolution

- Take a 2D curve and spin it around an axis
- Given curve **c**(*t*) in the plane, the surface is defined easily in cylindrical coordinates:

$$\mathbf{s}(t, s) = (r, \phi, z) = (c_x(t), s, c_y(t))$$

– the torus is an example in which the curve **c** is a circle

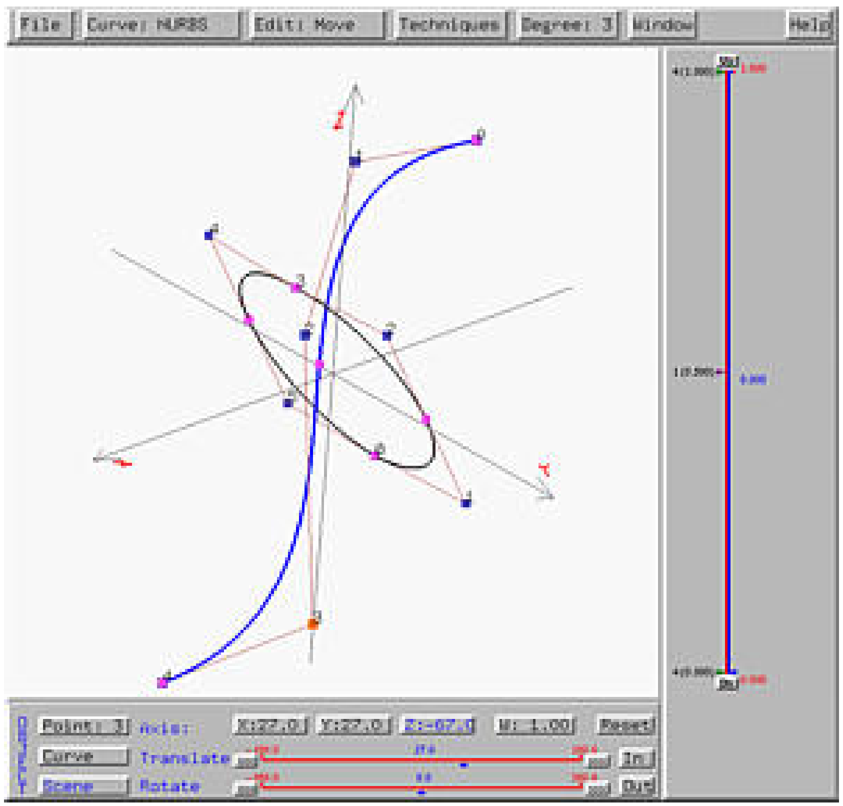


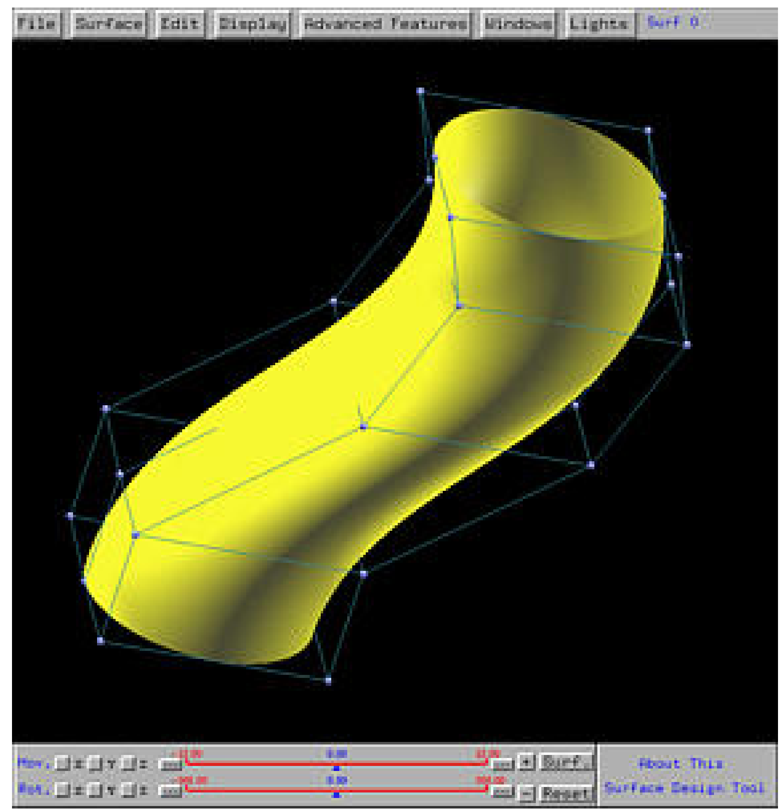


[Hearn & Baker]

Swept surfaces

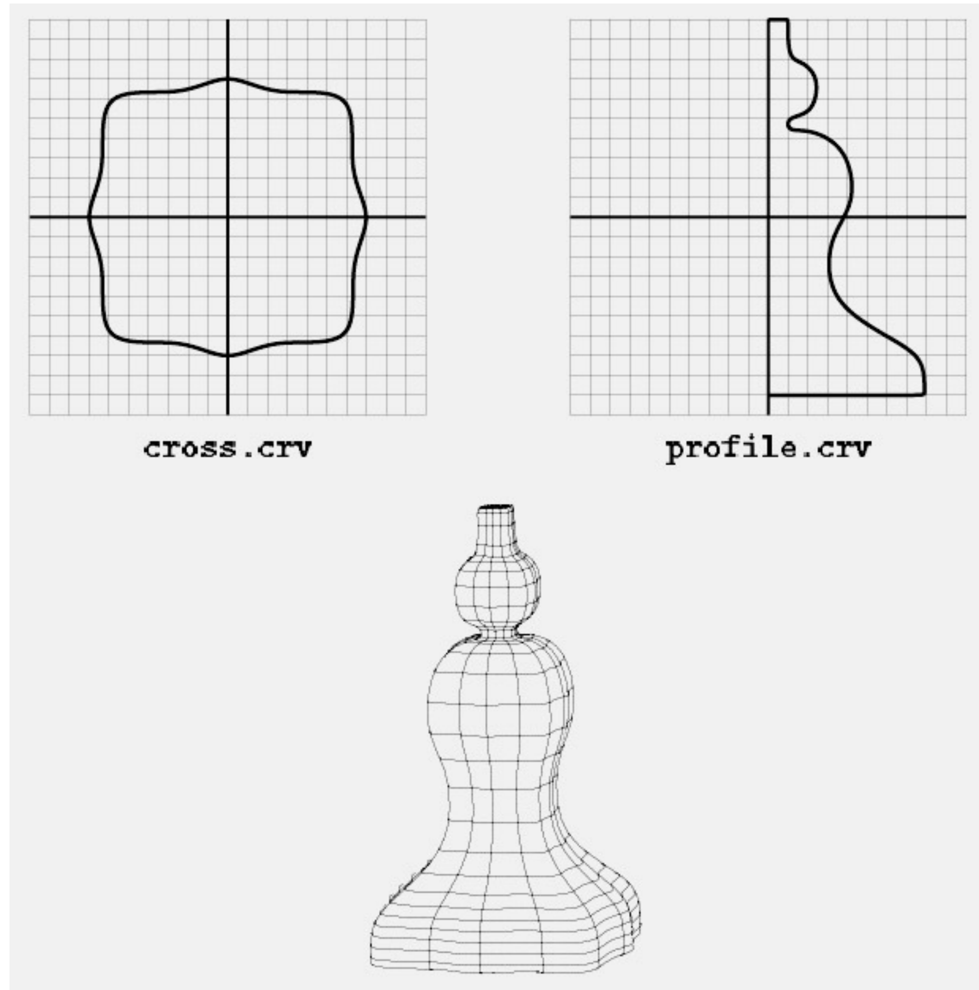
- Surface defined by a *cross section* moving along a *spine*
- Simple version: a single 3D curve for spine and a single 2D curve for the cross section





Generalized cylinders

- General swept surfaces
  – varying radius
  – varying cross-section
  – curved axis

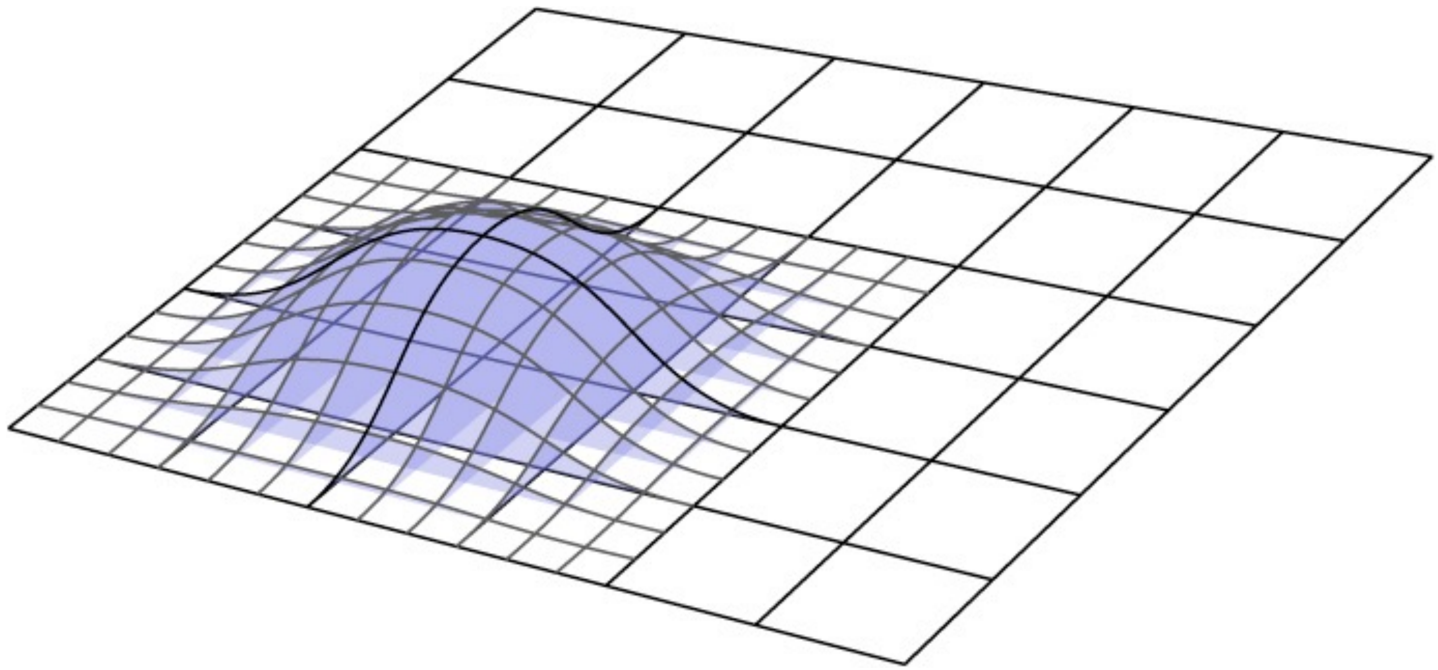


[Snyder 1992]

From curves to surface patches

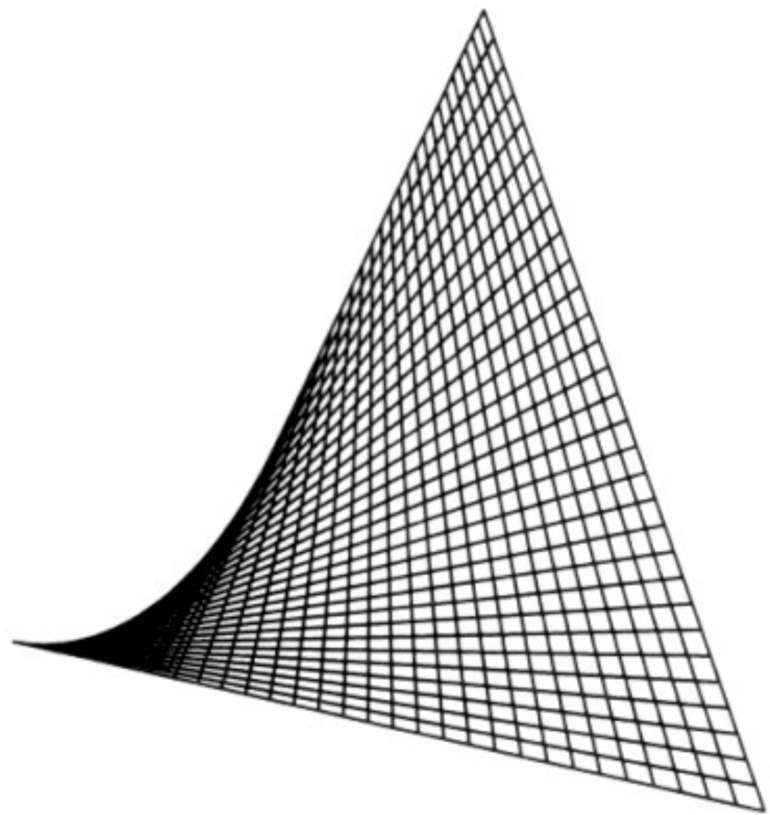
- Curve was sum of weighted 1D basis functions
- Surface is sum of weighted 2D basis functions
  - construct them as separable products of 1D fns.
  - choice of different splines
    - spline type
    - order
    - closed/open (B-spline)

Separable product construction



Bilinear patch

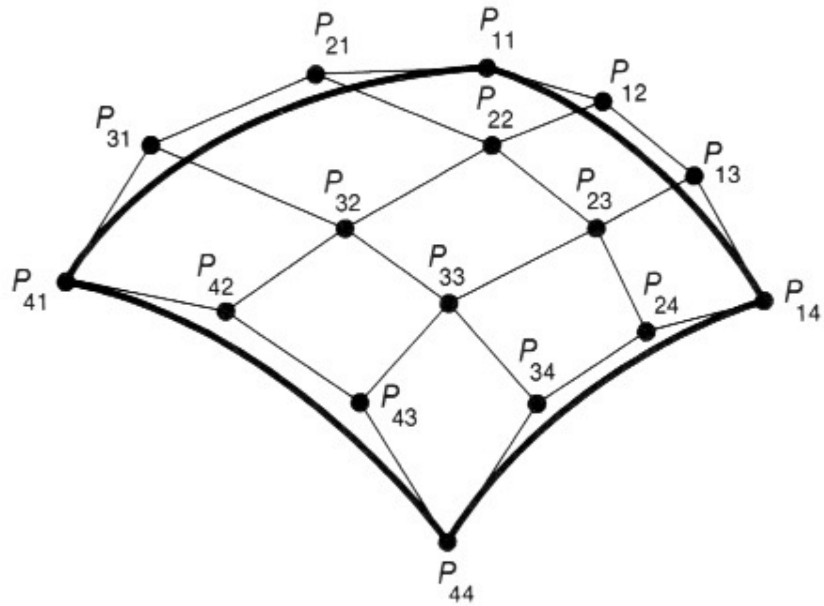
- Simplest case: 4 points, cross product of two linear segments



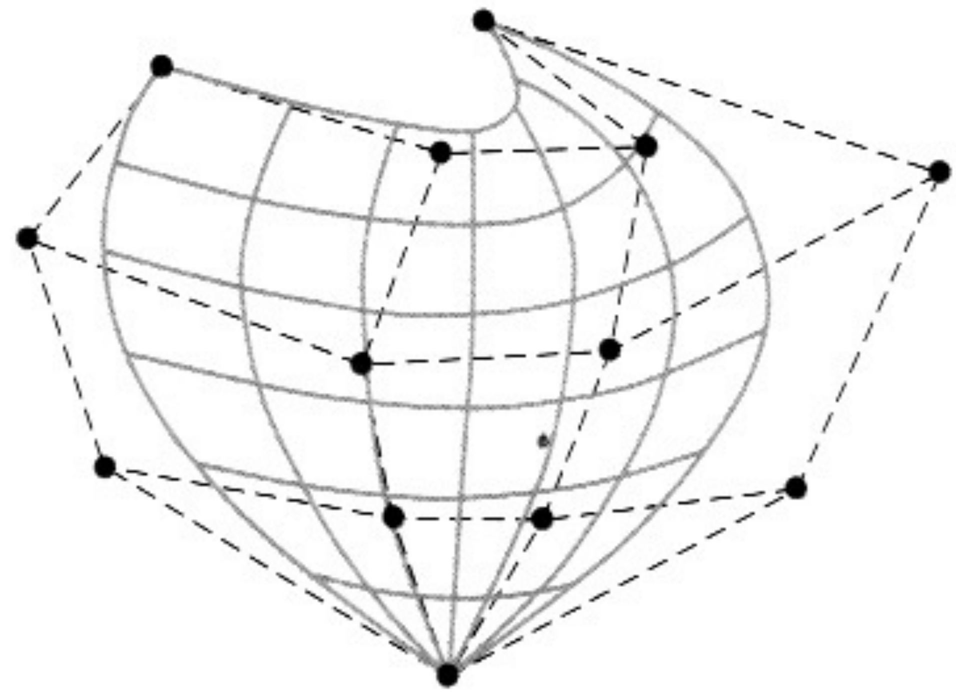
[Rogers]

Bicubic Bézier patch

- Cross product of two cubic Bézier segments



[Foley et al.]



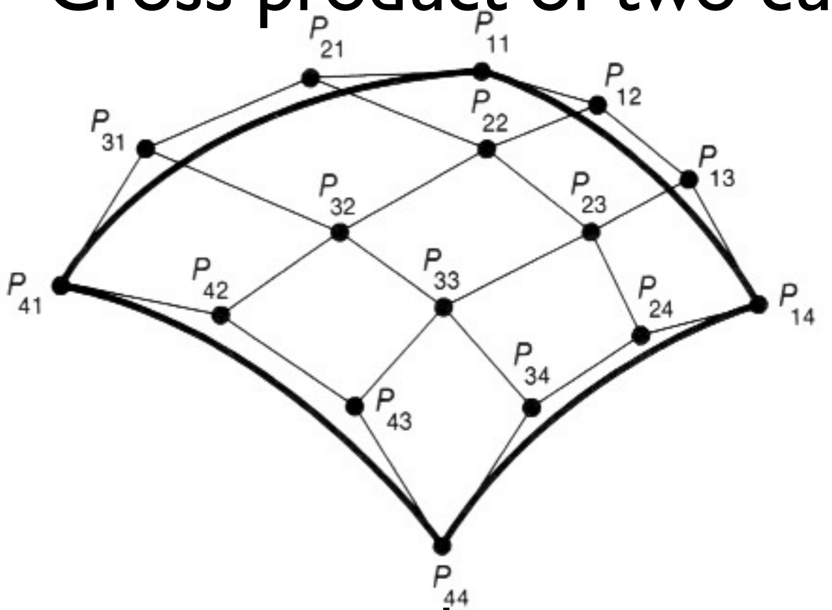
(b)

[Hearn & Baker]

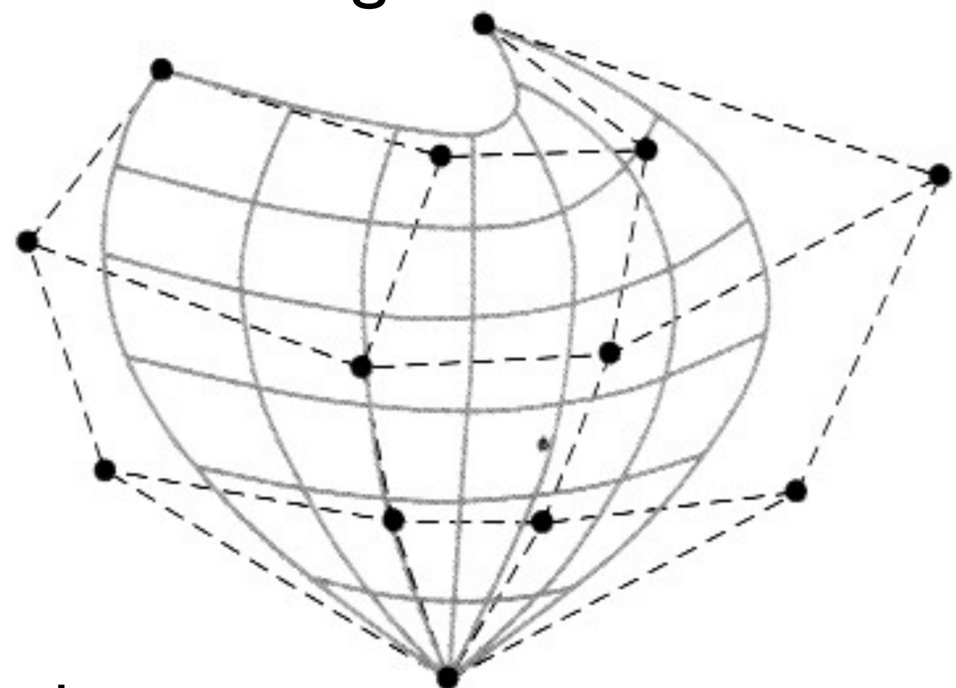
$$\mathbf{p}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u)\, B_j^m(v)\ P_{ij}$$

Bicubic Bézier patch

- Cross product of two cubic Bézier segments



[Foley et al.]



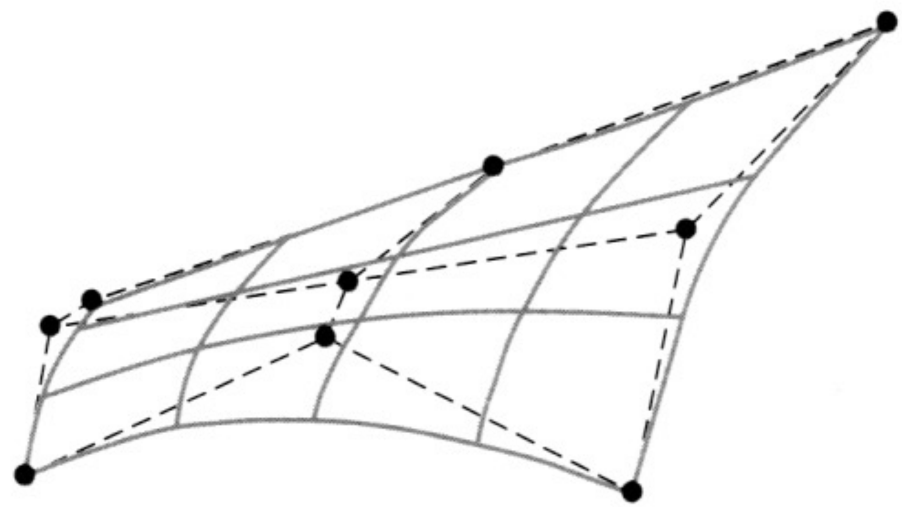
(b)

[Hearn & Baker]

  – properties that carry over
    - interpolation at corners, edges
    - tangency at corners, edges
    - convex hull

Biquadratic Bézier patch

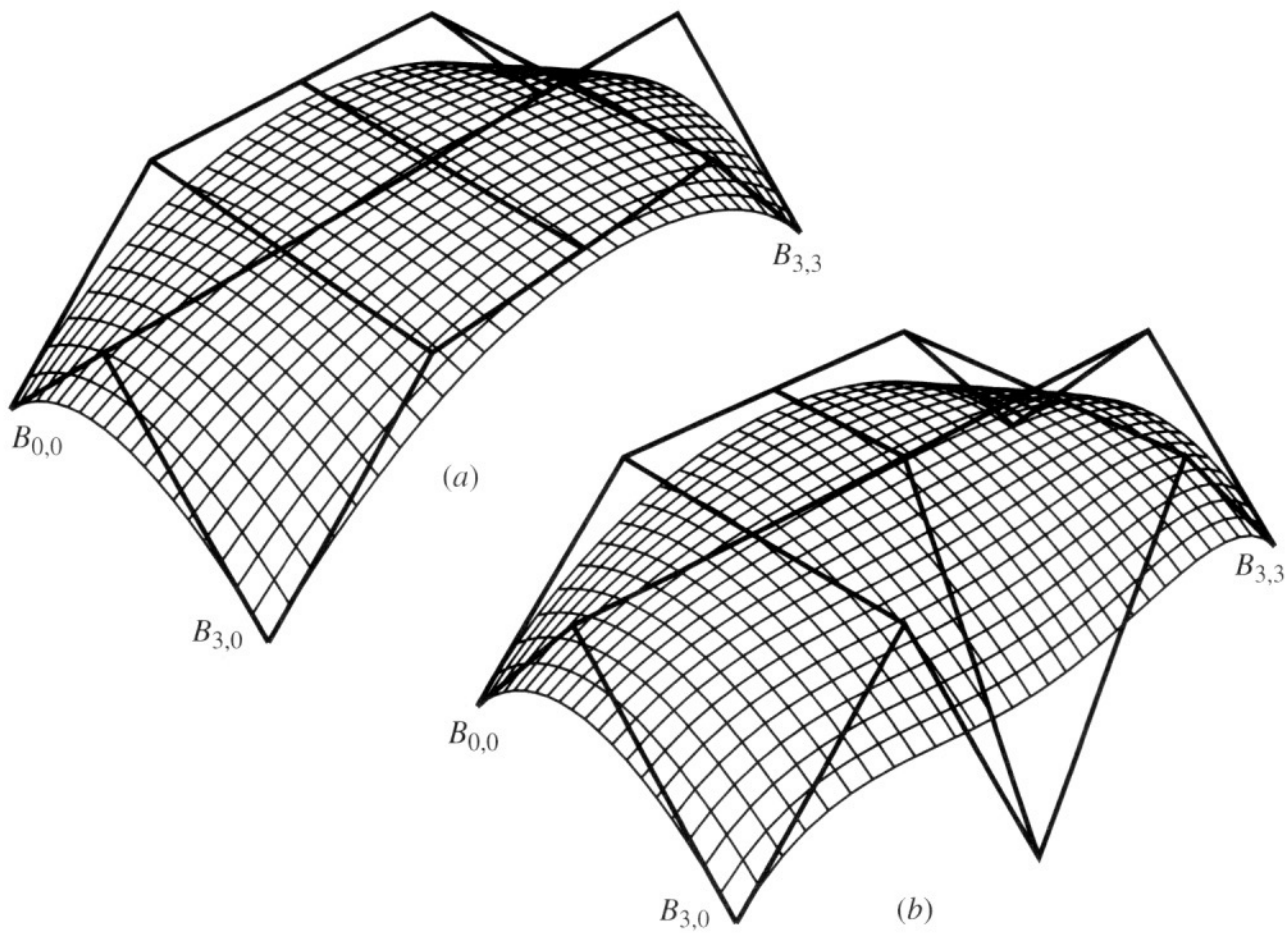
- Cross product of quadratic Bézier curves



[Hearn & Baker]

3x5 Bézier patch

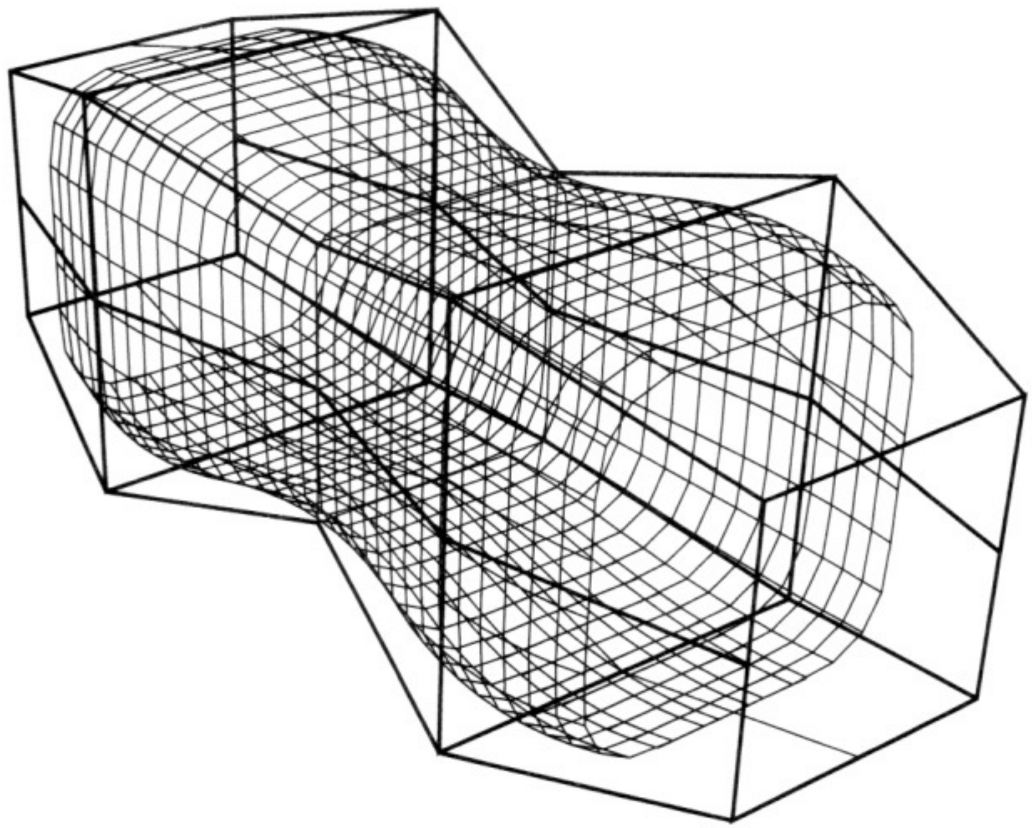
- Cross product of quadratic and quartic Béziers



[Rogers]

Cylindrical B-spline surfaces

- Cross product of closed and open cubic B-splines



(b)

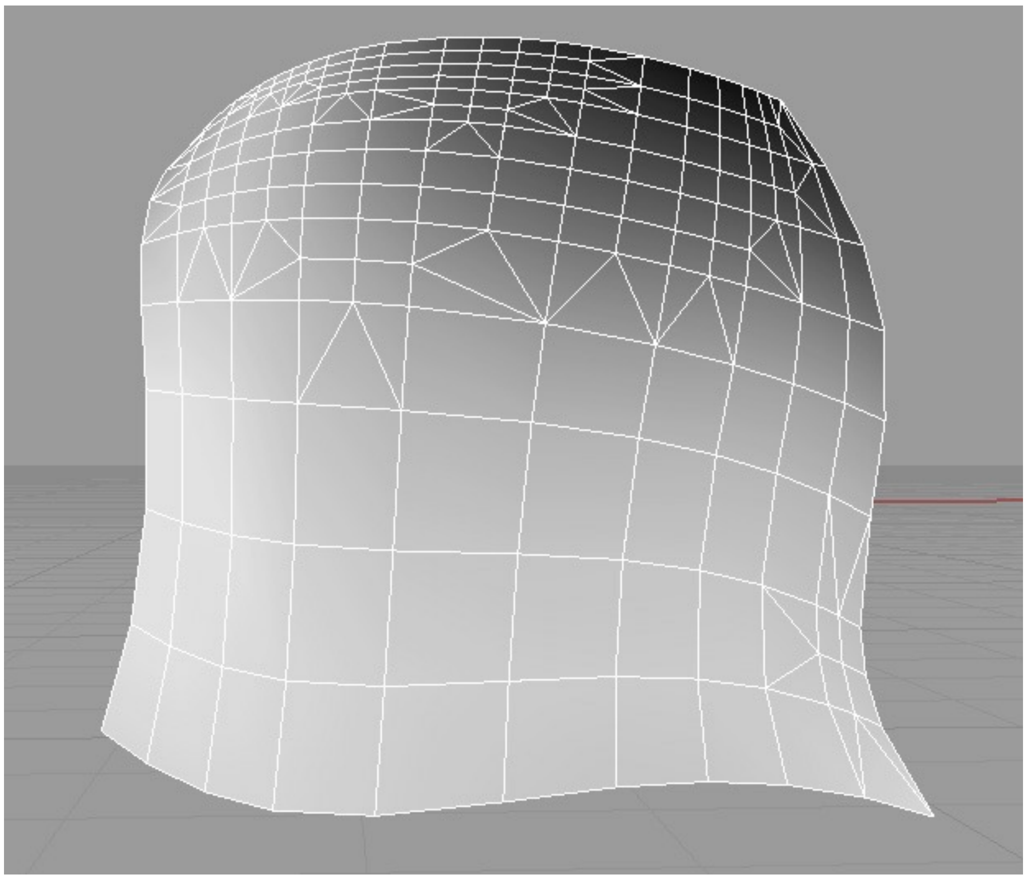
[Rogers]

Approximating spline surfaces

- Like curves, approximate with simple primitives
  - in surface case, triangles or quads
  - quads widely used because they fit in parameter space
    - generally eventually rendered as pairs of triangles
- adaptive subdivision
  - basic approach: recursively test flatness
    - if the patch is not flat enough, subdivide into four using curve subdivision twice, and recursively process each piece
  - as with curves, convex hull property is useful for termination testing (and is inherited from the curves)

Approximating spline surfaces

- With adaptive subdivision, must take care with cracks
  - (at the boundaries between degrees of subdivision)



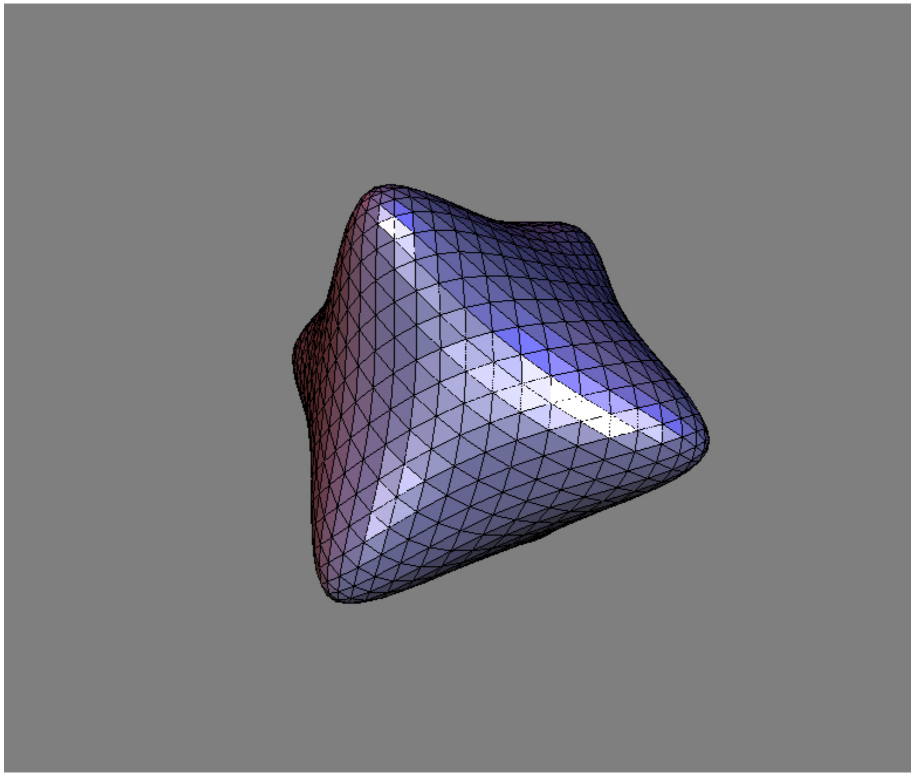
Geri’s Game

- Pixar short film to test subdivision in production
  - Catmull-Clark (quad mesh) surfaces
  - complex geometry
  - extensive use of creases
  - subdivision surfaces to support cloth dynamics

[DeRose et al. SIGGRAPH 1998]

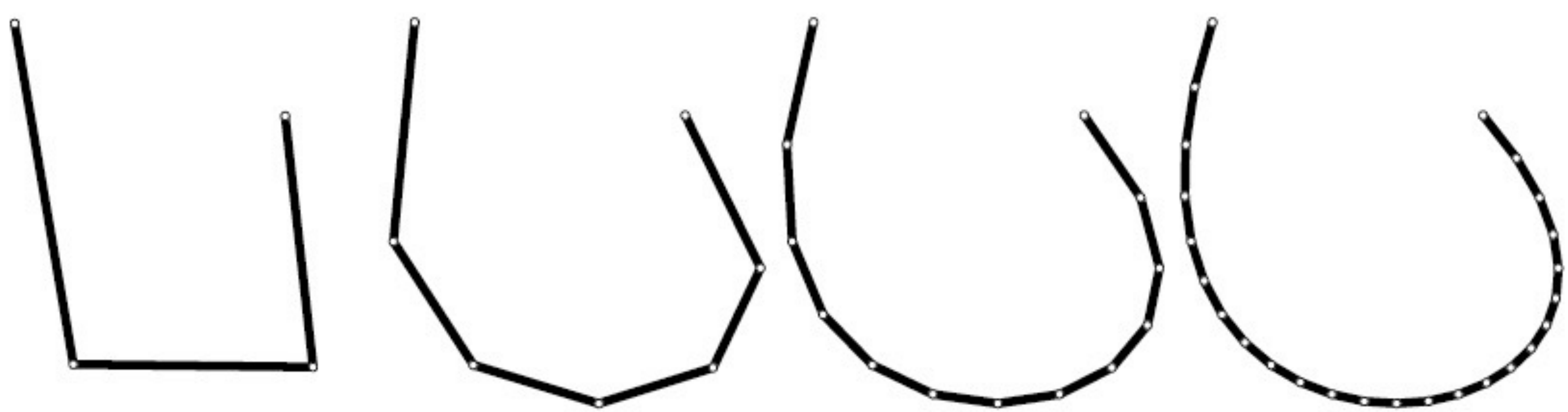
Specific surface representations

- Subdivision surfaces
  - based on polygon meshes (quads or triangles)
  - rules for subdividing surface by adding new vertices
  - converges to continuous limit surface



Subdivision curves

- Key idea: let go of the polynomials as the definition of the curve, and let the refinement rule define the curve
- Curve is defined as the *limit of a refinement process*
  – properties of curve depend on the rules
  – some rules make polynomial curves, some don't
  – complexity shifts from implementations to proofs



Subdivision surfaces

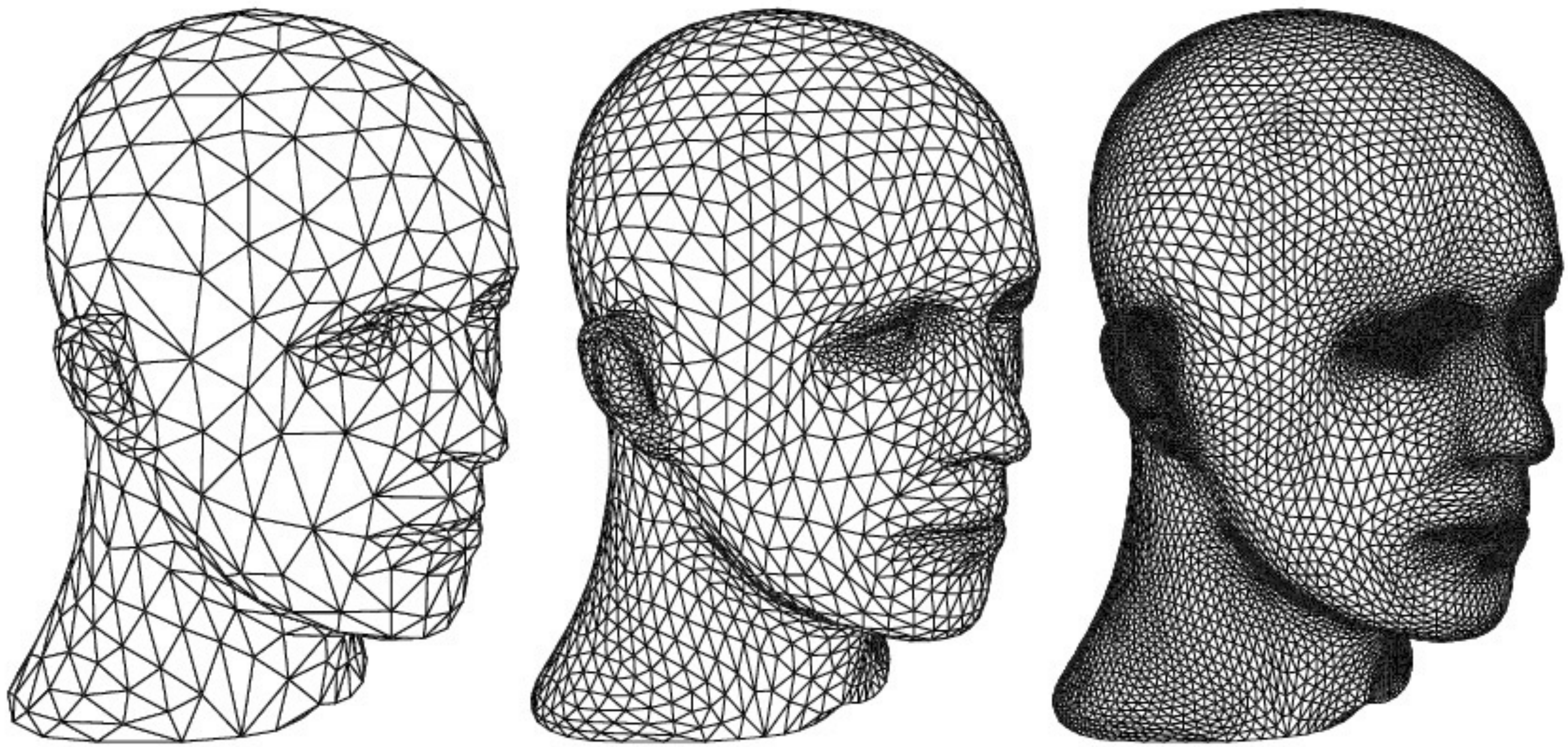


Figure 2.2: *Example of subdivision for a surface, showing 3 successive levels of refinement. On the left an initial triangular mesh approximating the surface. Each triangle is split into 4 according to a particular subdivision rule (middle). On the right the mesh is subdivided in this fashion once again.*

[Schröder & Zorin SIGGRAPH 2000 course 23]

