Efficient substitutes of Subdivision Surfaces

Jorg Peters
University of Florida

Overview

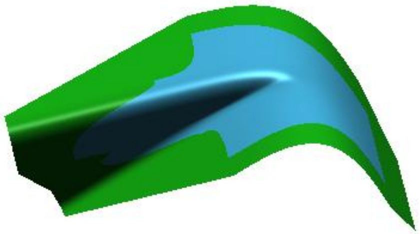
This course section develops the background knowledge.

It motivates and defines

- surface smoothness *vs* position and normal channel
- `exact' fitting *vs* approximation
- splines in B-spline form *vs* Bernstein Bezier form
- subdivision surfaces
- efficient substitutes: approximate and smooth

Smoothness: Why do we want smooth surfaces?

- Continuously changing normals are important both artistically and to avoid errors in downstream algorithms.
  - Gouraud shading, Phong shading, silhouette
  - Bump mapping, Normal mapping,
  - Displacement mapping,…



Smoothness and creases

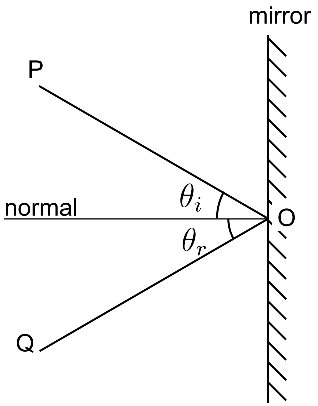
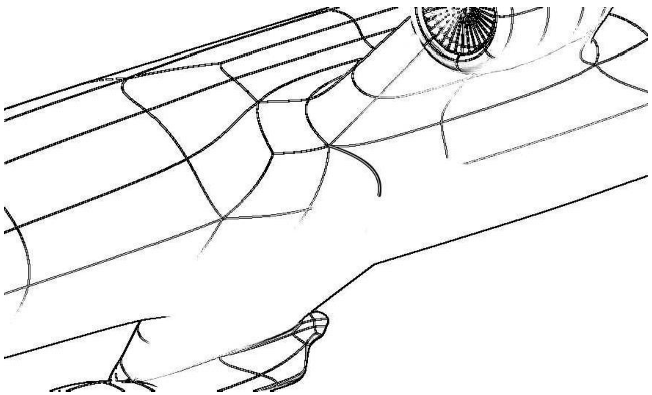
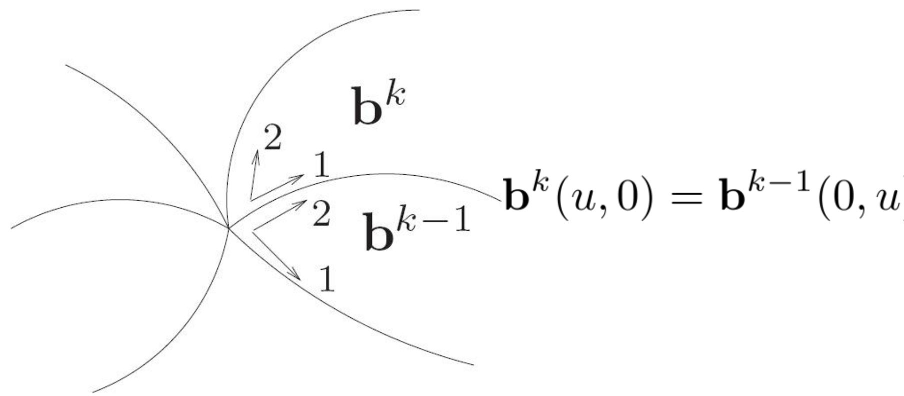


Diagram of specular reflection
(Image courtesy of wikipedia)

Surface smoothness = unique normal

- Mathematically: smoothly connected patches

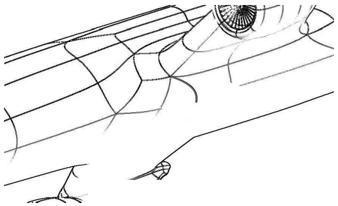




$$\mathbf{b}^k : \square \in R^2 \to R^3$$

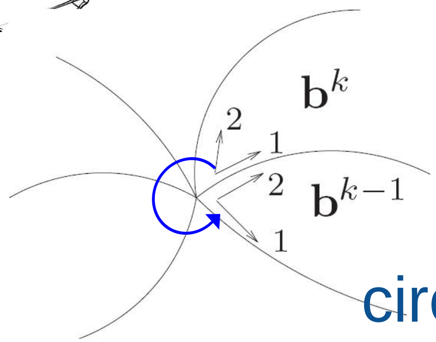
Surface smoothness – unique normal

G^1 continuity
= equal derivatives ∂ after change of variables ρ^k



$$\partial \mathbf{b}^k(u,0) = \partial\left(\mathbf{b}^{k-1} \circ \rho^k\right)(u,0)$$

$$\rho^k : \square \in R^2 \to R^2$$



circulant constraints:

Consistency of ρ^k

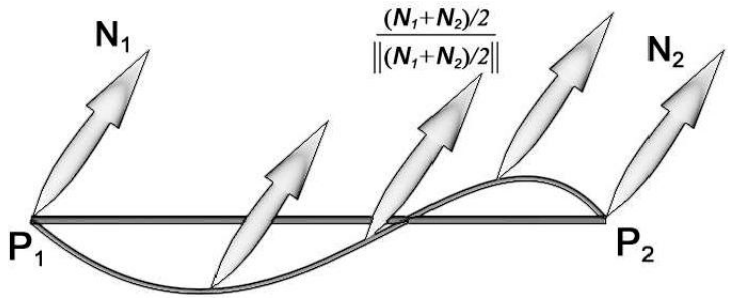
vertex enclosure $\partial_1 \partial_2 \mathbf{b}^k(0,0)$

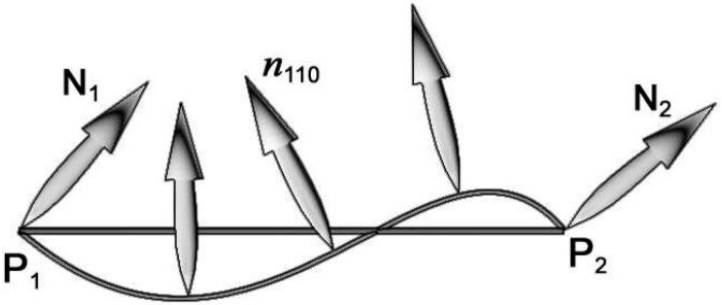
Tricky ! (justification for approximate smoothness)

Filling the normal channel

Algorithmically:

- Separation of the position and the normal channel
- Fill normal channel with directions not necessarily orthogonal to the surface





(from PN triangles)

Smoothness -- Summary

Unique normal

- Is needed for art and downstream processing
- Mathematically: G^1 continuity
- Algorithmically: position channel, normal channel

Overview

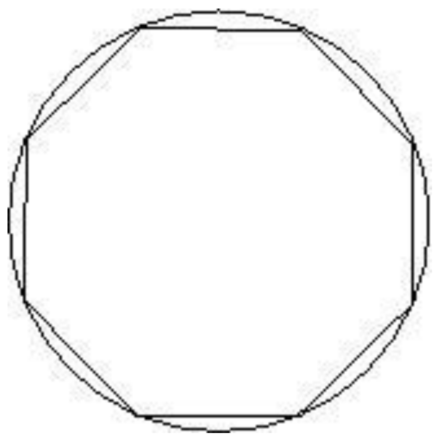
This course section develops the background knowledge.

It motivates and defines

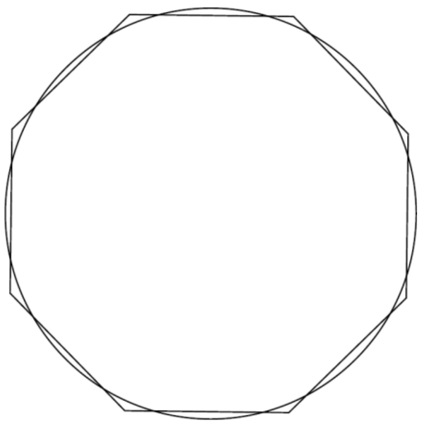
- surface smoothness *vs* position and normal channel
- `exact' fitting *vs* approximation
- splines in B-spline and Bernstein Bezier form
- subdivision surfaces
- efficient substitutes: approximate and smooth

Exact vs approximate fitting

- Which polygon represents the circle better?



Interpolation
`exact' fitting at points

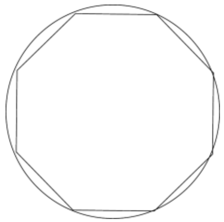


Approximation
`mid'-structure

In Computer Graphics:

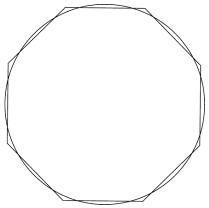
- pixel level: bi-linear polygonalization
- Tesselation: triangulation of surface

Exact vs approximate fitting



Interpolation

`exact' fitting at points



Approximation

`mid'-structure

- *Watertight (no pixel dropout):* Two adjacent patches tessellated independently should yield the same boundary points in $\mathbf{R}^3$
- *Matter of convention!* Interpolation or approximation will do

Fitting (evaluation) -- Summary

- Exact fitting (evaluation) is generally no better than approximate fitting
- Consistent fitting is crucial

Overview

This course section develops the background knowledge.

It motivates and defines

- surface smoothness -- position and normal channel
- debates ‘exact’ fitting vs approximation
- Splines: B-spline form *vs* Bernstein Bezier form
- subdivision surfaces
- efficient substitutes: approximate and smooth

Splines: B-spline form

- Degree 2: no inflections -- flat spots for higher-order saddles (Euler's Thm)
- Cubic (degree 3) spline
  - $\mathbf{x}(t) := \sum_\ell q_\ell N_\ell(t)$
- Fast stable evaluation= "knot insertion"

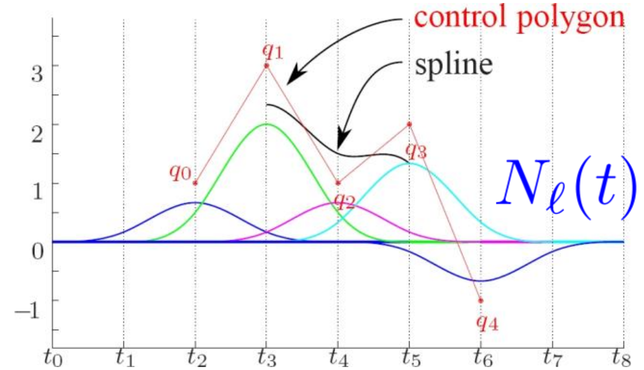
  = control point averaging

  - de Boor's algorithm

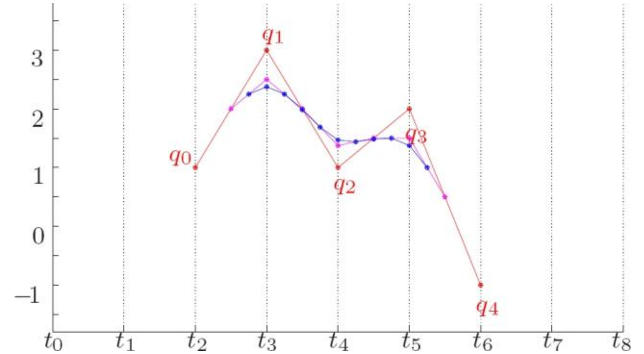
  = depth first insertion at same parameter

  - subdivision

  = breadth first insertion, uniform



Univariate uniform cubic spline



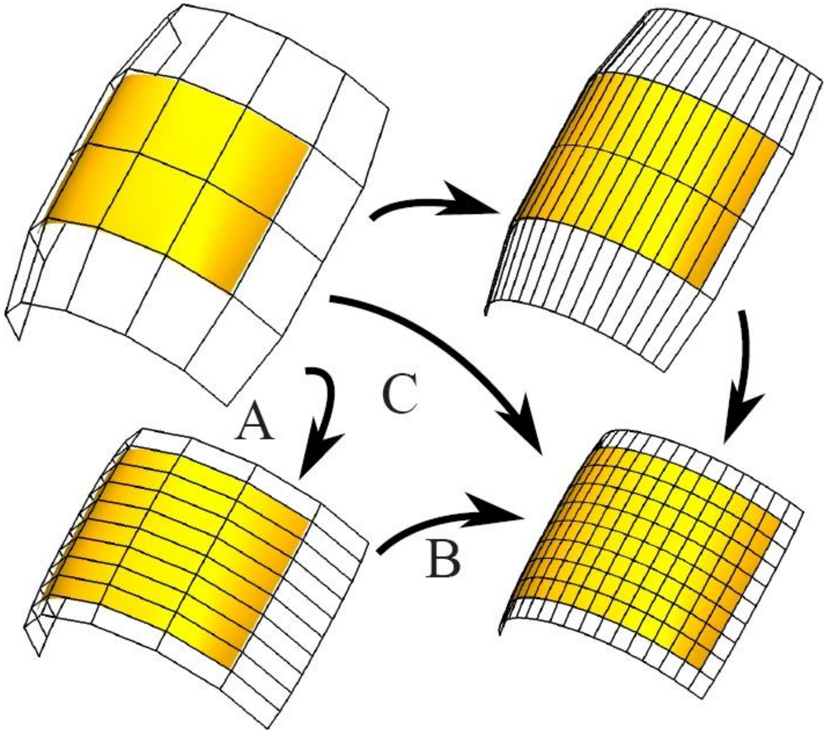
Subdivision

Splines: B-spline form

- Bi-3 splines

$$\sum_{i=0}^{3}\sum_{j=0}^{3} q_{i,j} N_i(u) N_j(v).$$

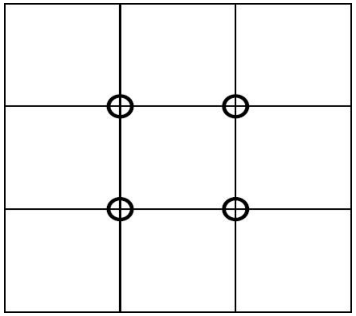
  - gluNurbsSurface
  - De Boor's algorithm
  - subdivision

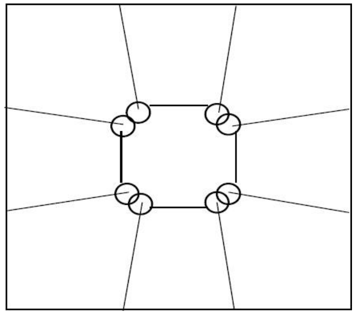


Commutativity

Splines: Bernstein-Bezier form (BB-form)

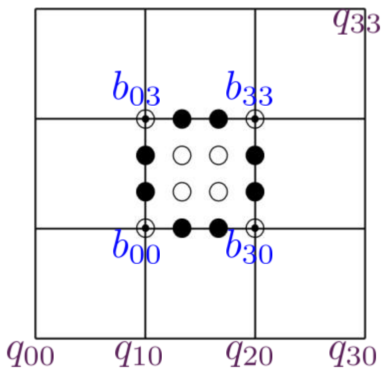
- Cubic spline $\sum_{i=0}^{3} b_i \; B_i(u) \quad B_k(t) := \frac{3!}{(3-k)!k!}(1-t)^{3-k} t^k$
  - glMap1
  - De Casteljau's algorithm
- Bi-3 splines $\sum_{i=0}^{3} \sum_{j=0}^{3} b_{i,j} \; B_i(u) \; B_j(v)$
  - glMap2
- Bi-3 patch rational **Gregory** patch expensive normal





Splines: B-spline to BB-form conversion

$$q \longrightarrow b$$

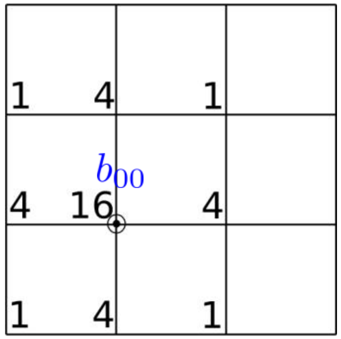


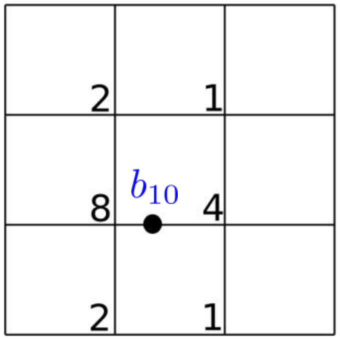
$$36b_{00} := 16q_{11} + 4(q_{21} + q_{12} + q_{01} + q_{10}) + q_{22} + q_{02} + q_{00} + q_{20}$$

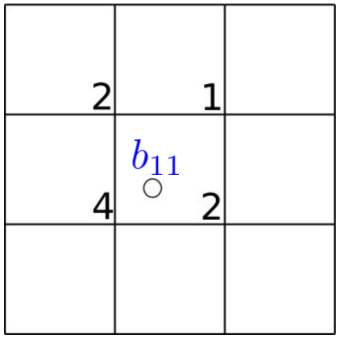
$$18b_{10} := 8q_{11} + 2q_{10} + 2q_{12} + 4q_{21} + q_{20} + q_{22}$$

$$9b_{11} := 4q_{11} + 2(q_{12} + 2q_{21}) + q_{22}$$

B-splines and the BB-form are equally powerful!







Splines -- Summary

Splines

- are piecewise polynomials
- typically smoothly connected and
- can be represented in B-spline form or BB-form
  - Use the (uniform) B-spline form to have built-in smoothness
  - Use the BB-form to interpolate at corners

Overview

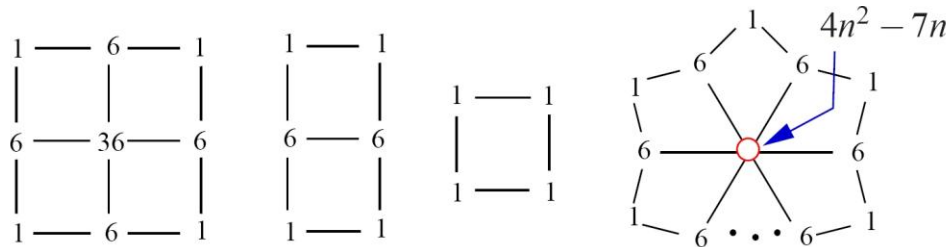
This course section develops the background knowledge.

It motivates and defines

- surface smoothness -- position and normal channel
- debates ‘exact’ evaluation vs approximation
- splines in B-spline and Bernstein Bezier form
- subdivision surfaces
- efficient substitutes: approximate and smooth

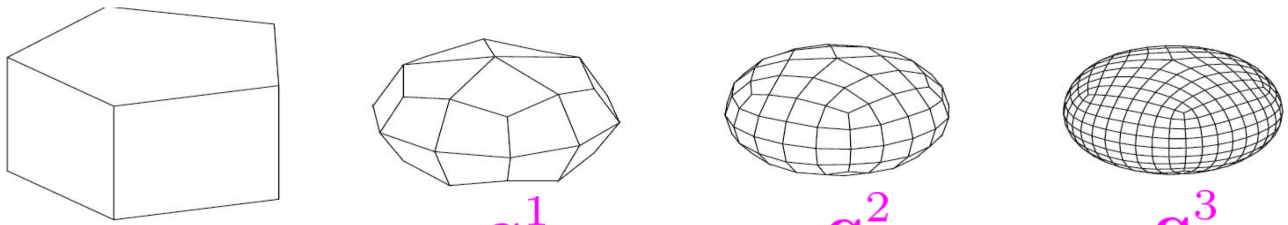
Subdivision surfaces

- *Algorithmically*, a mesh refinement procedure that applies rules to determine
  - the position of new mesh points from old ones : $\mathbf{q}^m = A^m \mathbf{q}$
  - the new connectivity



Stencil (mask)

n = valence



$\mathbf{q} = \mathbf{q}^0$ $\mathbf{q}^1$ $\mathbf{q}^2$ $\mathbf{q}^3$

Catmull-Clark algorithm

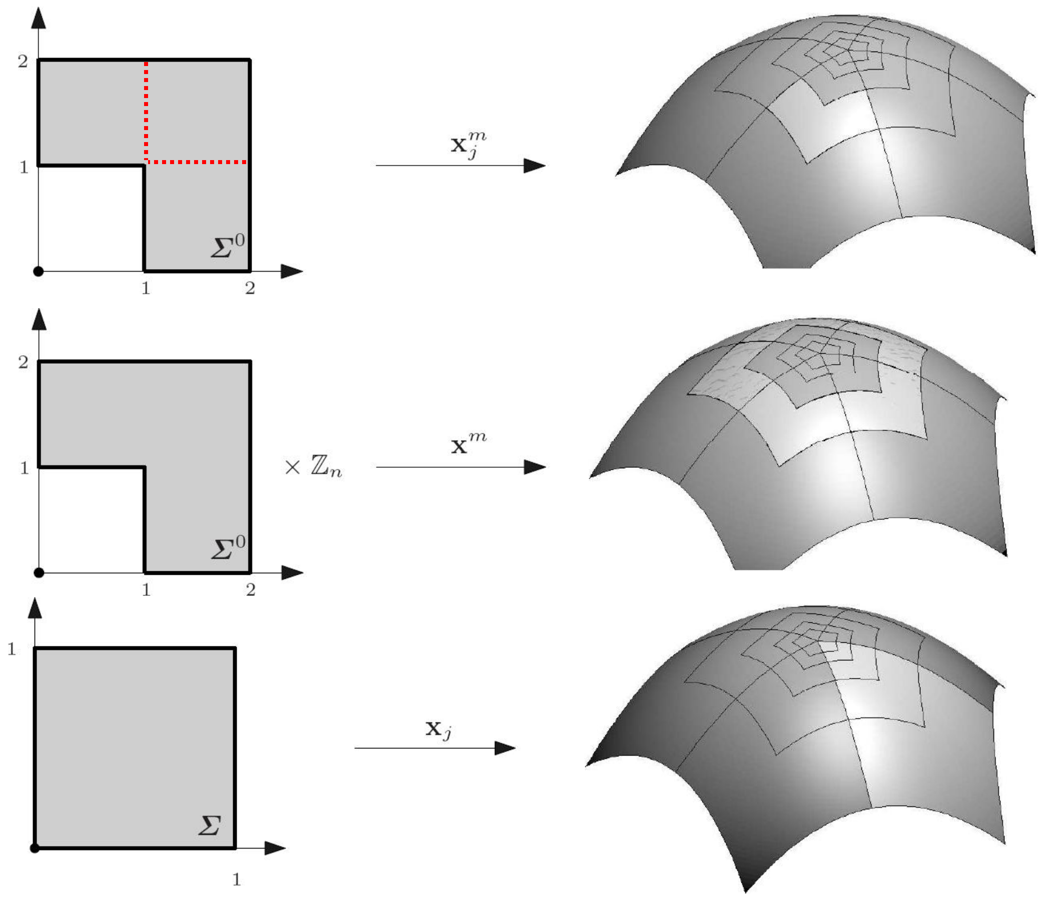
Subdivision Surfaces

Mathematically, a spline surface with isolated singularities.

From

[Peters, Reif08]

Subdivision Surfaces



Segment
=
3 spline
patches

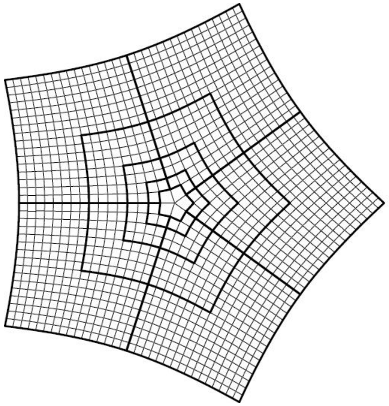
Nested
rings

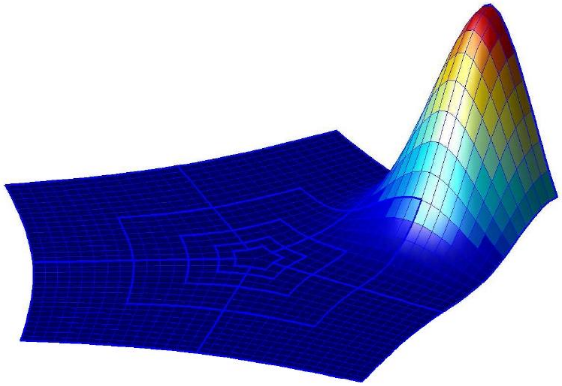
Subdivision
surface
patch

Evaluation of Subdivision Surfaces

- Standard Evaluation
  1. determine the ring m (by taking the logarithm of u,v base 2);
  2. apply m subdivision steps (either by matrix A^m or stencil);
  3. interpret the resulting control net at level m
     as L-shaped segments in B-spline form;
  4. evaluate the bi-3 spline (by de Boor's algorithm).
- Usually most efficient!
- Especially for grids
- parameterization

$$\sum_{i=0}^{3}\sum_{j=0}^{3} A^m q_{i,j} N_i(u) N_j(v).$$



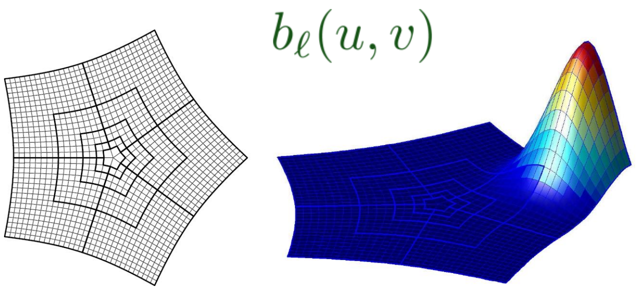


Evaluation of Subdivision Surfaces

- Standard Evaluation:

  1.determine m; 2.apply m steps;

  3.get spline; 4.evaluate spline



- Alternatives for special cases only !
  - Tabulation of Generating Functions [Bolz,…]
    - No general creases, fixed max-level of refinement

$$\mathbf{x}(u,v) = \sum_{\ell}^{L} \mathbf{q}_\ell \, b_\ell(u,v)$$

  - Patch selection 2. in eigenspace [DooSabin,Stam]
    - Encode-decode $A = VJV^{-1}$ $\quad \mathbf{p} := V^{-1}\mathbf{q}$
    - No creases, Isolated point, high m (level of refinement) $A^m\mathbf{q} \rightarrow VJ^m\mathbf{p}$
  - Eigensystem evaluation [Micchelli, WarrenSchaefer,…]
    - No creases, solve system of eq's

Subdivision -- Summary

Subdivision Surfaces

- are (infinitely many) piecewise polynomials
- Algorithmically: mesh refinement,

  Mathematically: spline with singularity
- Has a natural parametrization
- Best evaluated by standard technique (using the parameterization)

Overview

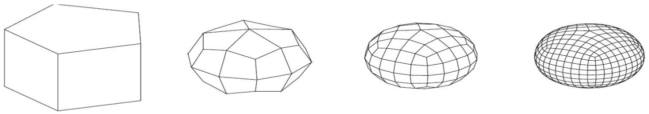
This course section develops the background knowledge.

It motivates and defines

- surface smoothness -- position and normal channel
- debates ‘exact’ evaluation vs approximation
- splines in B-spline and Bernstein Bezier form
- subdivision surfaces
- efficient substitutes: approximate and smooth

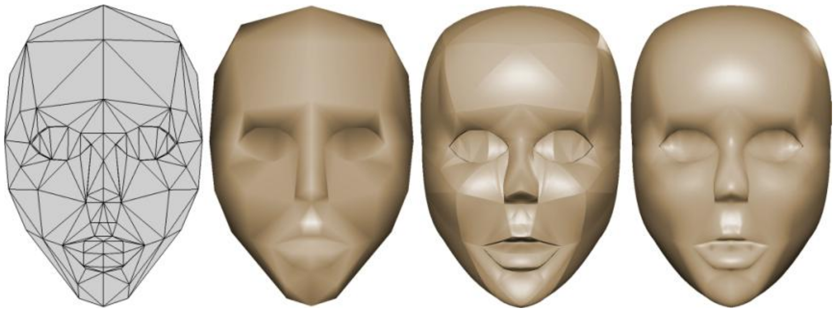
Efficient Substitutes: Can it be done simpler?

- Control polyhedra mid-structures, proxy splines

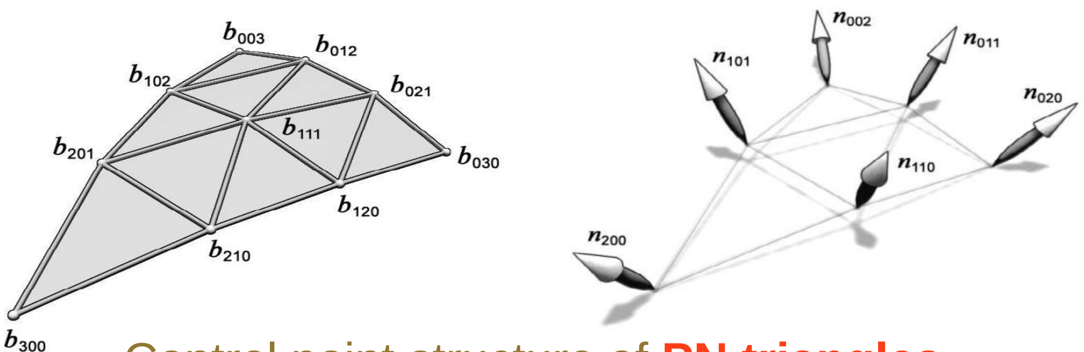


- Separate geometry and normal channels
  - Refine geometry, refine normals
    - *PN triangles* [VPBM01, BA08]
    - *PN quads*
    - (connectivity) **ACC** [LS08]

      B-spline to BB



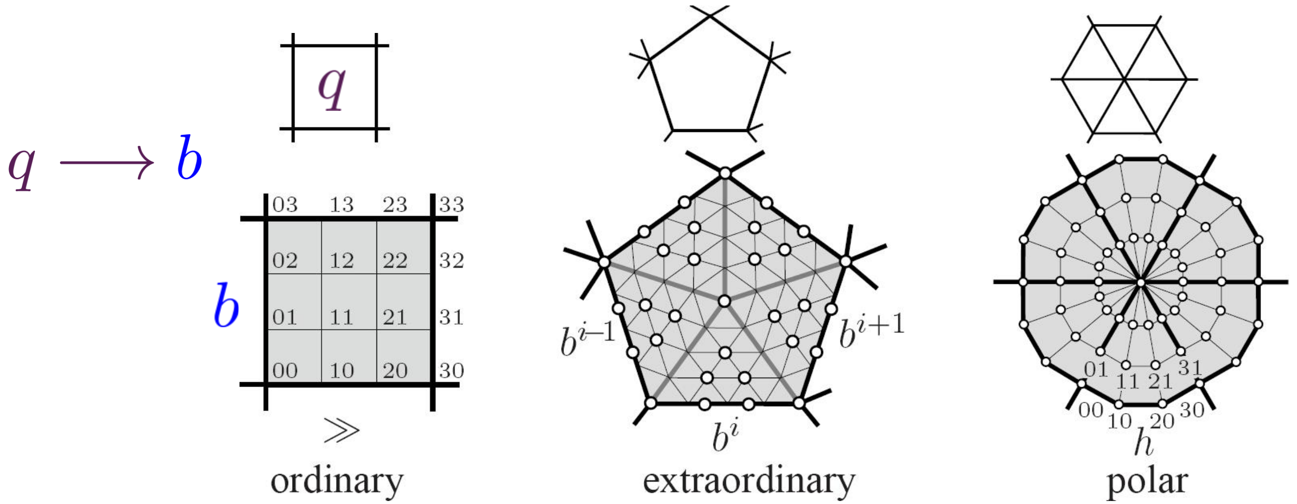
Input Gouraud geometry PN



Control point structure of **PN triangles**

Efficient Substitutes: Smooth surfaces

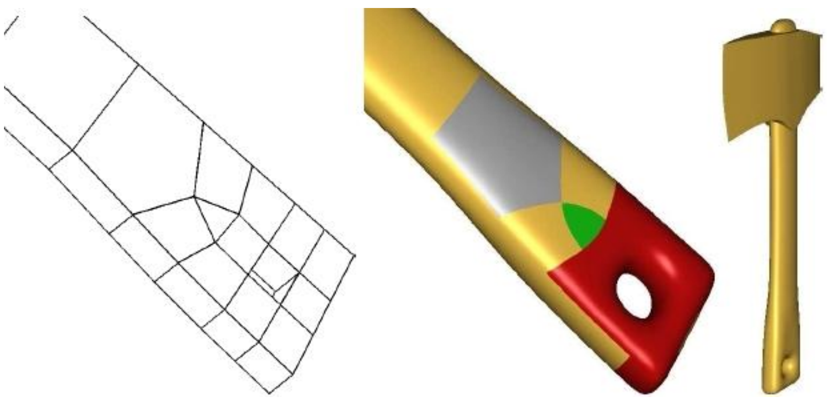
1. Control polyhedra
2. Separate geometry and normal channels (PN, ACC)
3. ***G^1*** surface constructions (smooth)
   - c-patches and many-sided P(m)-patches



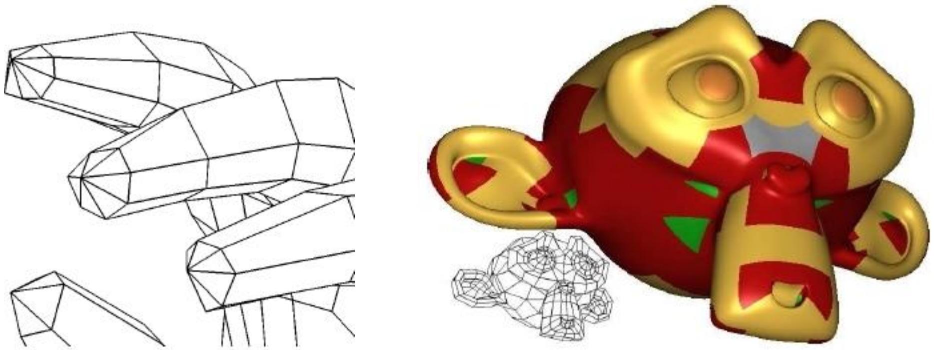
Mesh-to-patch conversion

C^1 surface constructions

c-patches and many-sided p(m)-patches



Quad/tri/pent model with
hemi-sharp blends/creases (blade)



Quad/tri/pent polar models

Efficient Substitutes -- Summary

- SubD surfaces = piecewise polynomial with infinitely many pieces
- Substitutes = piecewise polynomial with finitely many pieces

Besides (projected) control polyhedra there are two classes of efficient substitutes for subdivision surfaces:

- Approximately smooth (PN, ACC) and G^1 smooth
- Triangular, quad and polar patches are available
  - mimic both Catmull-Clark and triangle-based subdivision
- Representations in their own right!

Efficiency Comparison

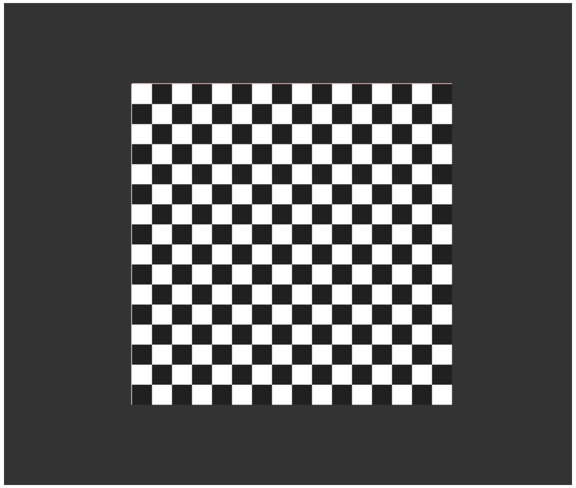
efficiency	**space**	**time**	**comment**
triangulation	_	_	fixed resolution
recursive subD	_		adaptivity?
non-recursive subD		_	creases?
tabulation	_	+	creases?
efficient substitutes	+	+	crease✓, adapt✓

Summary: Take home points

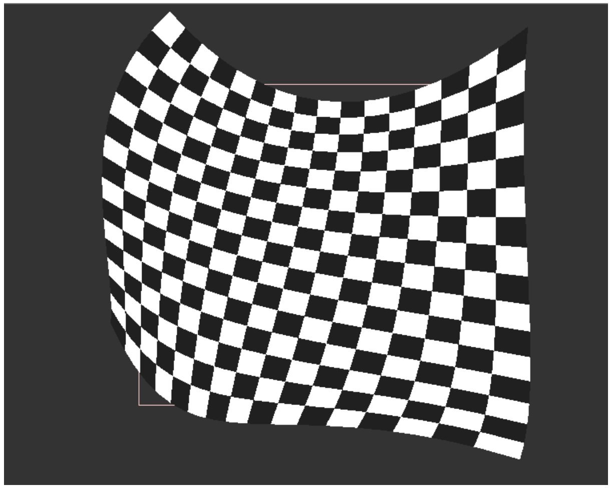
This course section developed the background knowledge.

- surface smoothness *vs* position and normal channel
- debates ‘exact’ evaluation *vs* approximation
- splines in B-spline form *vs* Bernstein Bezier form
- subdivision surfaces
- substitutes: approximate (PN) and G^1 smooth
- space and time efficiency is superior
- Thank you!
- Questions?

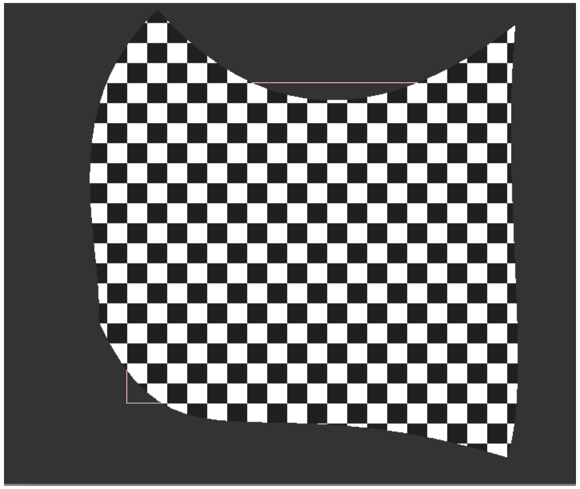
Semi sharp blends can preserve texture



Texture



After
surface perturbation

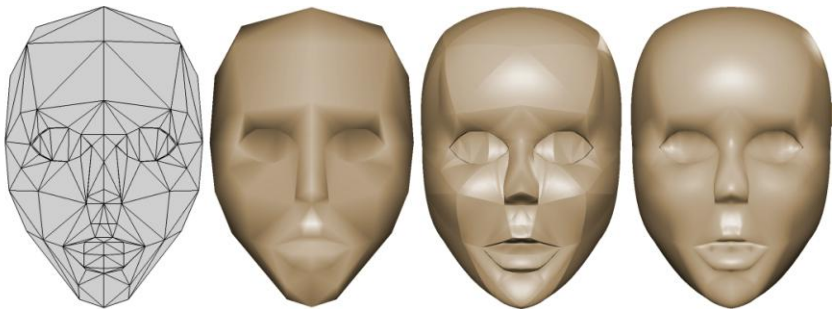


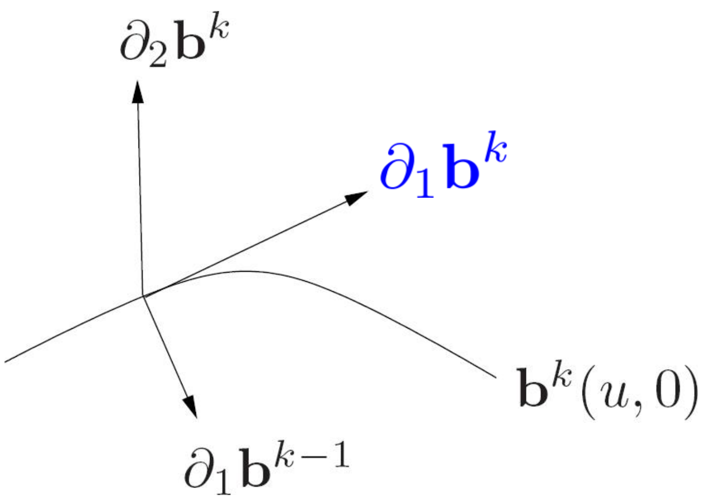
With texture coordinates
according to perturbation

Higher-quality surfaces?

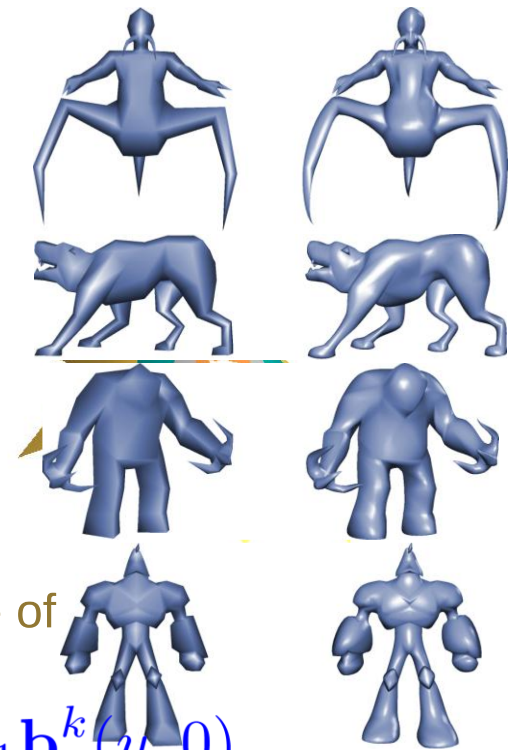
- For high-end design, C^1 continuity is not sufficient.
- Guided surfacing, L-S SGP08, bi-3 C^2 polar subdivision,…

extra





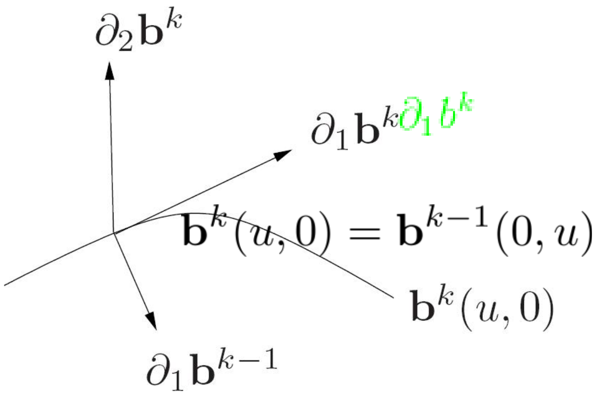
A nested sequence of

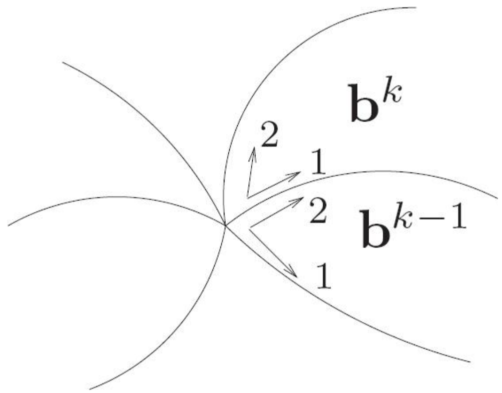


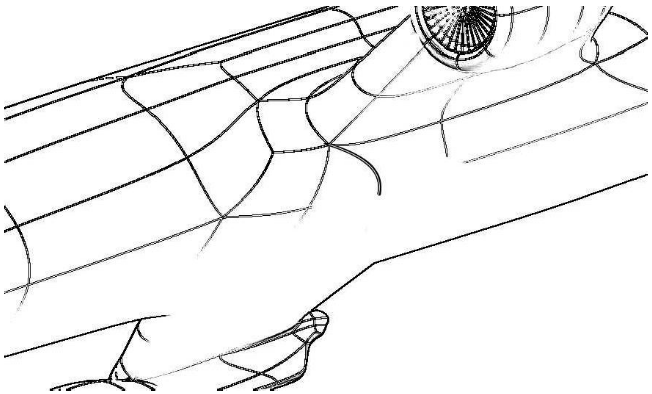
$$\beta^k(u)\partial_2 \mathbf{b}^k(u,0) + \gamma^k(u)\partial_1 \mathbf{b}^{k-1}(0,u) = \alpha^k(u)\partial_1 \mathbf{b}^k(u,0)$$

Surface smoothness = unique normal

- ***G*[1]** continuity: equal derivatives after change of variables







$$\partial_2 \mathbf{b}^k(u,0) + \partial_1 \mathbf{b}^{k-1}(0,u) = \alpha^k(u)\partial_1 \mathbf{b}^k(u,0)$$

circulant constraints
vertex enclosure
Consistency of reparam's