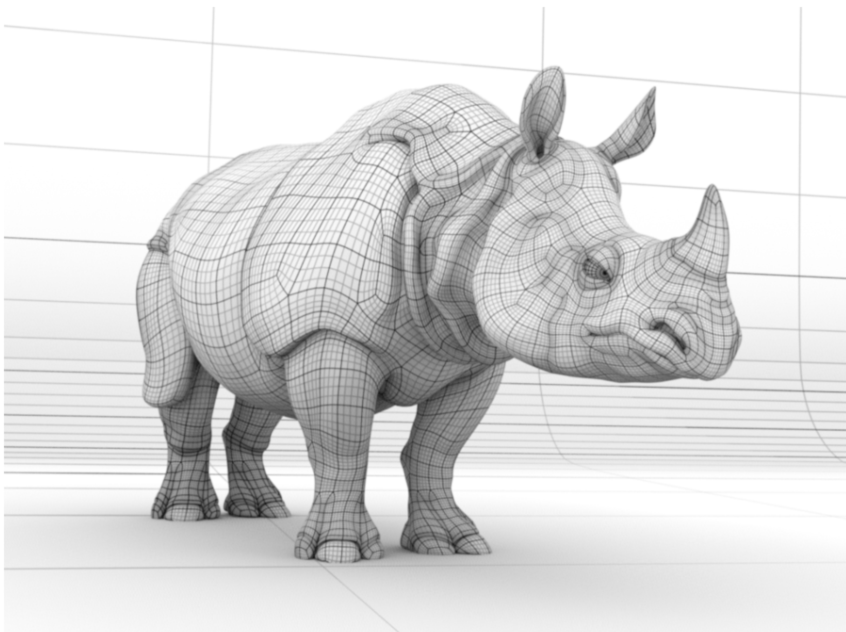
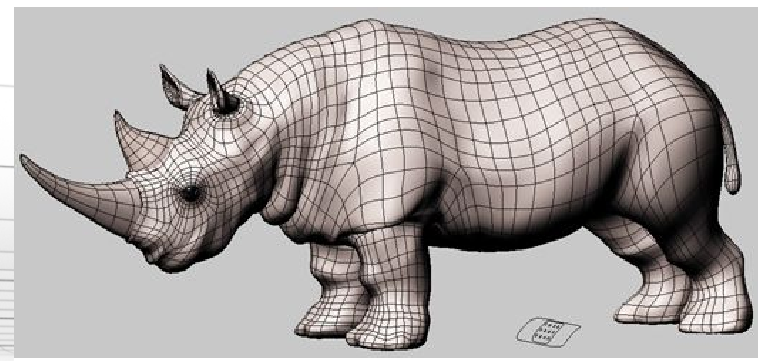
B-Spline and Subdivision Surfaces



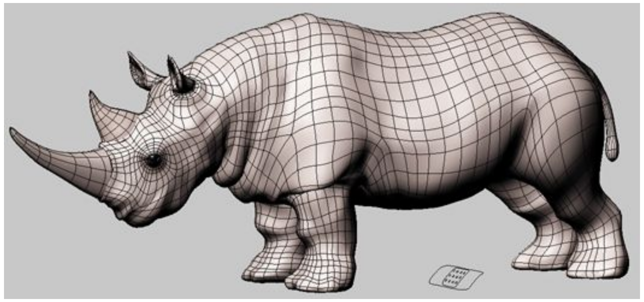


Slides from Mirela Ben-Chen

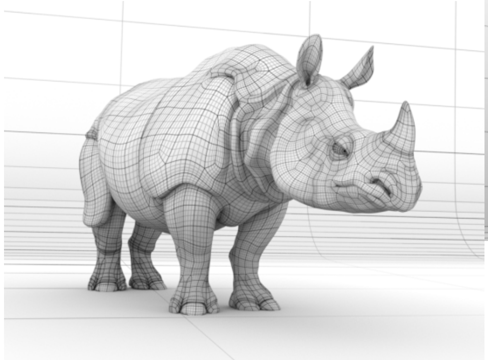
Images from: 3drender.com & sunflow.sourceforge.net

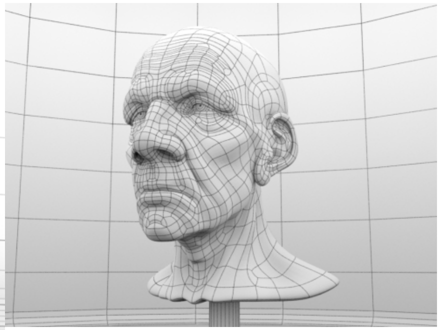
Surface Models

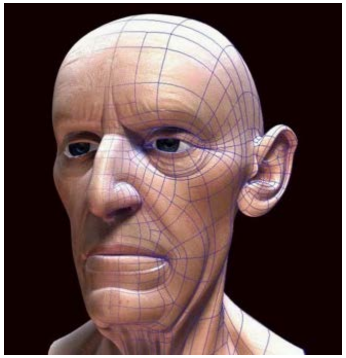
- B-Spline surfaces
  - NURBS surfaces



- Subdivision surfaces
  - Theory
  - Zoo







Images from: 3drender.com , sunflow.sourceforge.net , www.tsplines.com

Reminder: B-Spline Curves

B-Spline Curve

Decouple number of control points from degree of curve

"Glue" a few degree p Bézier curves, with continuity conditions

[Applet (Curve)]

Reminder: B-Spline Curves

B-Spline Curve

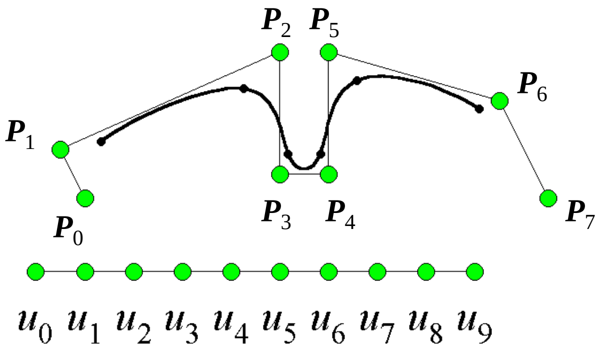
Building blocks:

$n + 1$ control points $\boldsymbol{P}_i$

Knot vector $U = \{ u_0 , u_1 , \ldots , u_m \}$

The degree p

m, n, p satisfy $m = n + p - 1$



$$C(t) = \sum_{i=0}^{n} N_{i,p}(t) \boldsymbol{P}_i \quad ; \quad N_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Applet (Basis functions)

B-Spline Surfaces

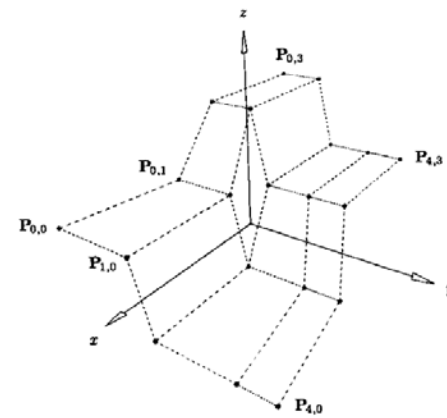
A collection of Bezier patches, with continuity conditions

Decoupling the degree and the number of control points

B-Spline Surfaces

B-Spline surface - tensor product surface of B-Spline curves

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) \boldsymbol{P}_{ij}$$



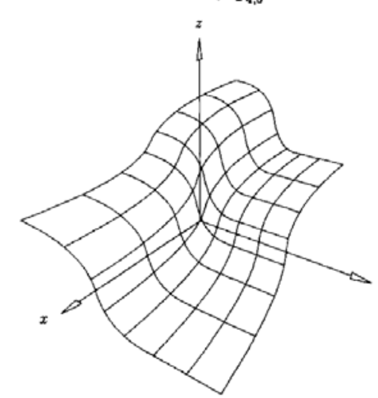
Building blocks:

Control net, $m + 1$ rows, $n + 1$ columns: $\boldsymbol{P}_{ij}$

Knot vectors $U = \{ u_0 , u_1 , \dots , u_h \}$,

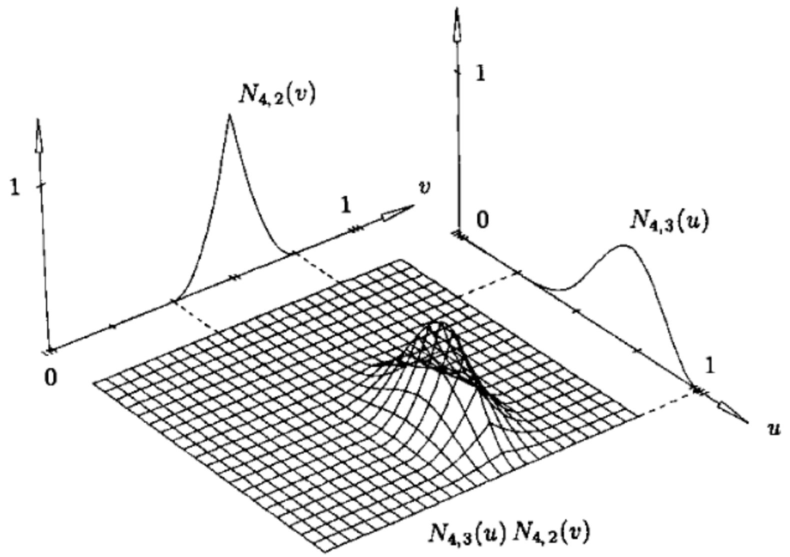
$V = \{ v_0 , v_1 , \dots , v_k \}$

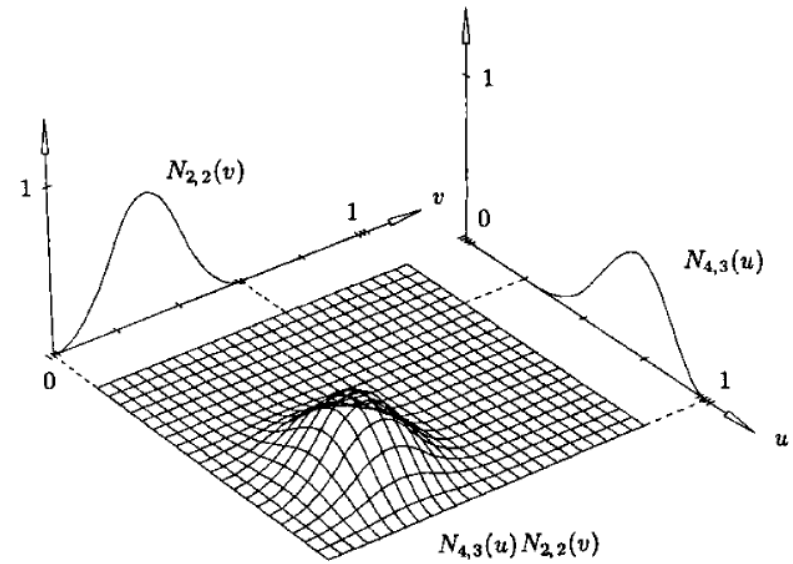
The degrees p and q for the u and v directions



Basis Functions

Cubic × Quadratic basis functions:





Properties

- Non negativity

$$N_{i,p}(u)N_{j,q}(v) \geq 0, \qquad \text{for all } i, j, p, q, u, v$$

- Partition of unity

$$\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v) = 1, \qquad \text{for all } (u,v) \in [0,1]\times[0,1]$$

- Affine invariance

$$A\boldsymbol{S}_{\{\boldsymbol{P}_{ij}\}}(u,v) + B = \boldsymbol{S}_{\{A\boldsymbol{P}_{ij}+B\}}(u,v)$$

$$\text{for all } A \in R^{3x3}, B \in R^{3}, (u,v) \in [0,1]\times[0,1]$$

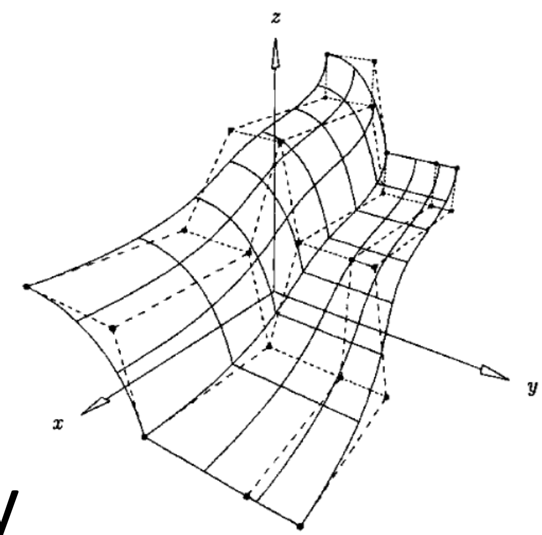
Properties

- If $n = p$, $m = q$, $U = \{ 0,\ldots,0,1,\ldots,1\}$ and $V = \{0,\ldots,0,1,\ldots,1\}$ then

$$N_{i,p}(u)N_{j,q}(v) = B_i^n(u)B_j^m(v)$$

  and $\boldsymbol{S}(u,v)$ is a Bézier surface

- $\boldsymbol{S}(u,v)$ is C^{p-k} continuous in the u direction at a u knot of multiplicity k, and similar for v direction



$U = \{0,0,0,0.5,0.5,1,1,1\}$

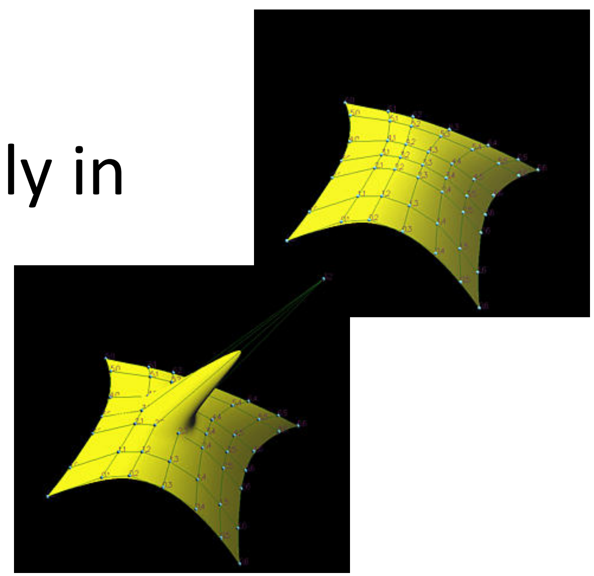
$V = \{0,0,0,0,0.5,1,1,1,1\}$

Properties

- Compact support

$$N_{i,p}(u)N_{j,q}(v) = 0, \qquad \text{for all } (u,v) \notin \left[u_i, u_{i+p+1}\right] \times \left[v_j, v_{j+q+1}\right]$$

- Local modification scheme
  - Moving $\boldsymbol{P}_{ij}$ affects the surface only in the rectangle $\left[u_i, u_{i+p+1}\right] \times \left[v_j, v_{j+q+1}\right]$



Properties

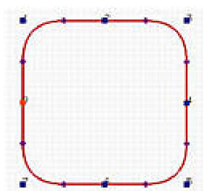
- Local definition
  - In any rectangle $\left[u_{i_0}, u_{i_0+1}\right] \times \left[v_{j_0}, v_{j_0+1}\right]$ the only non-zero basis functions are

$$N_{i,p}(u) N_{j,q}(v), \qquad \text{for } i_0 - p \le i \le i_0 \ \text{ and } \ j_0 - q \le j \le j_0$$

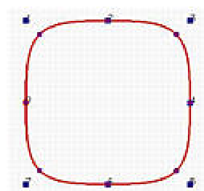
- Strong convex hull property
  - If $(u,v) \in \left[u_{i_0}, u_{i_0+1}\right] \times \left[v_{j_0}, v_{j_0+1}\right]$ then $\boldsymbol{S}(u,v)$ is in the convex hull of the control points $\boldsymbol{P}_{ij}$ $i_0 - p \le i \le i_0$ and $j_0 - q \le j \le j_0$

Reminder: NURBS Curves

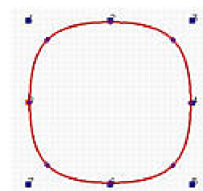
- B-spline curves cannot represent exactly circles and ellipses



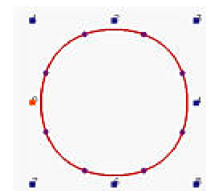
$p = 2$



$p = 3$



$p = 5$



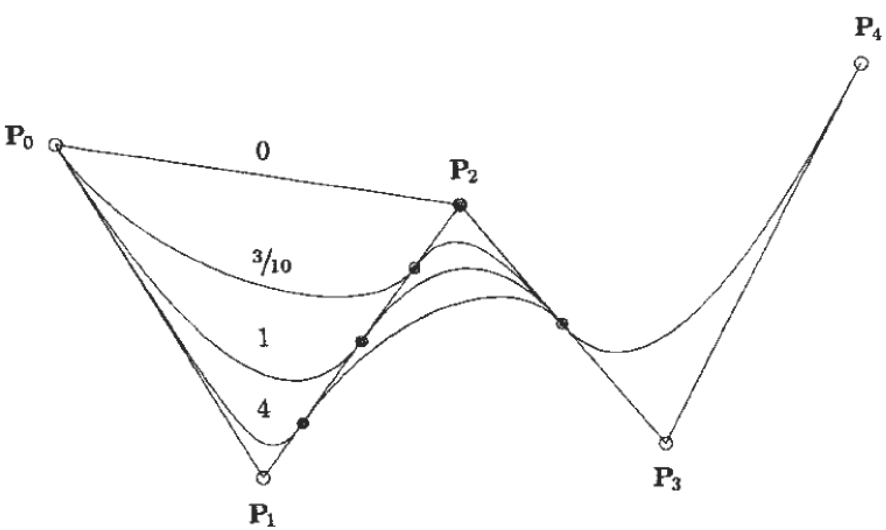
$p = 10$

- Generalize to *rational polynomials*

$$\boldsymbol{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \boldsymbol{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i}$$

Reminder: NURBS Curves

A weight per control point allows to change the influence of a point on the curve, without moving the point



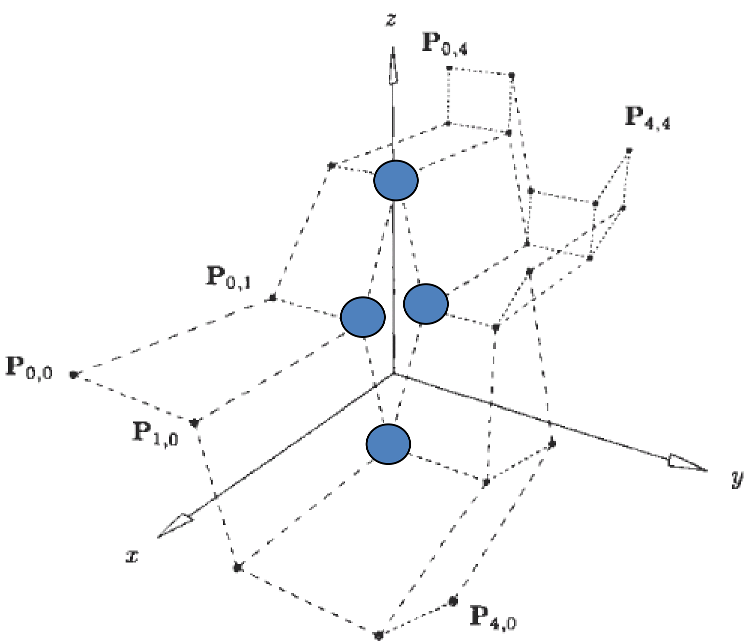
NURBS Surfaces

Add a weight for every control point of a B-spline surface, and normalize

$$\boldsymbol{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{ij} \boldsymbol{P}_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{ij}}$$

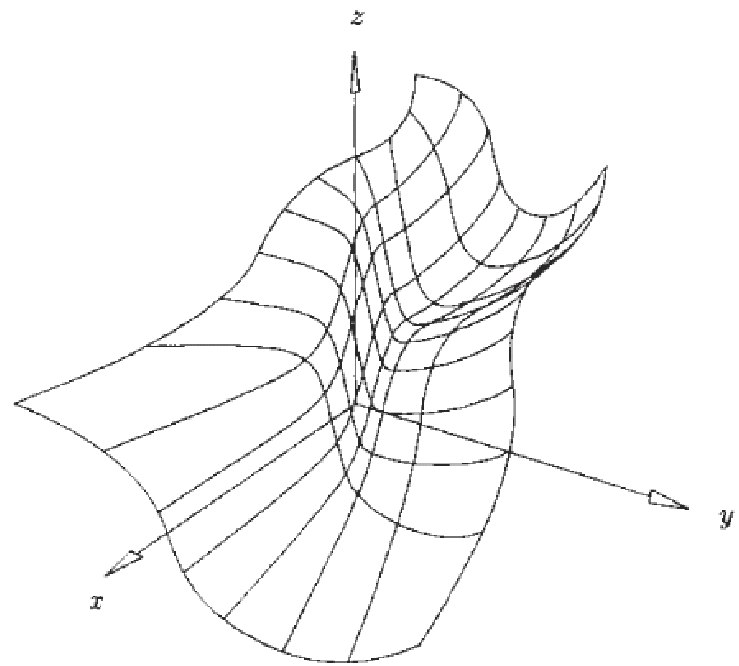
Is **not** a tensor product patch

NURBS Surface Example



Control net

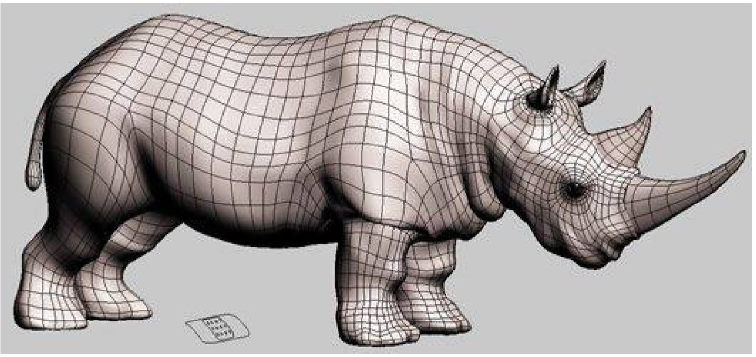
$$U = V = \{0, 0, 0, 1/3, 2/3, 1, 1, 1\}$$

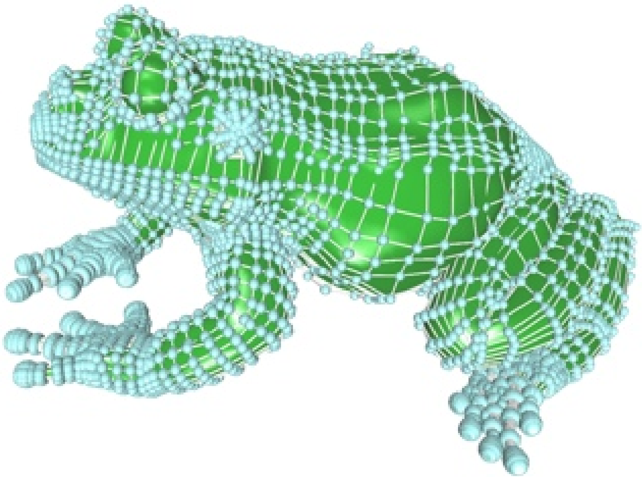


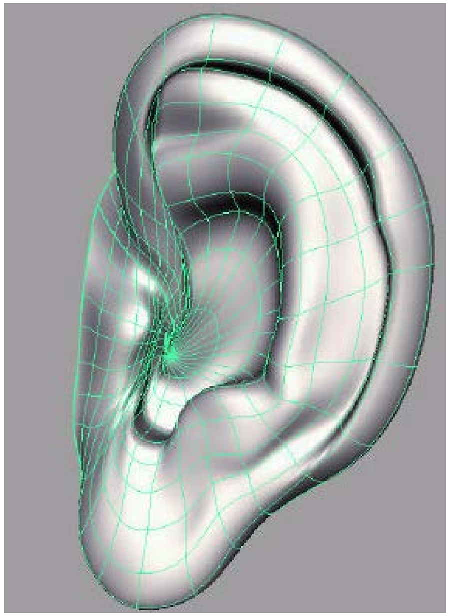
NURBS Surface

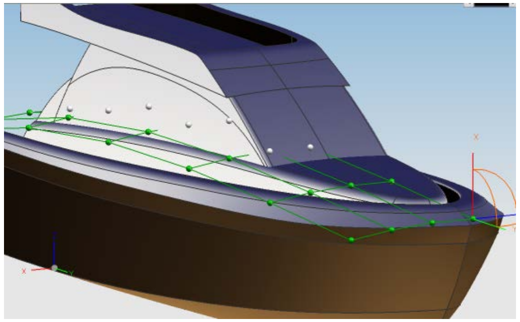
w_{ij}(●) = 10, w_{ij}(•) = 1

NURBS Surfaces





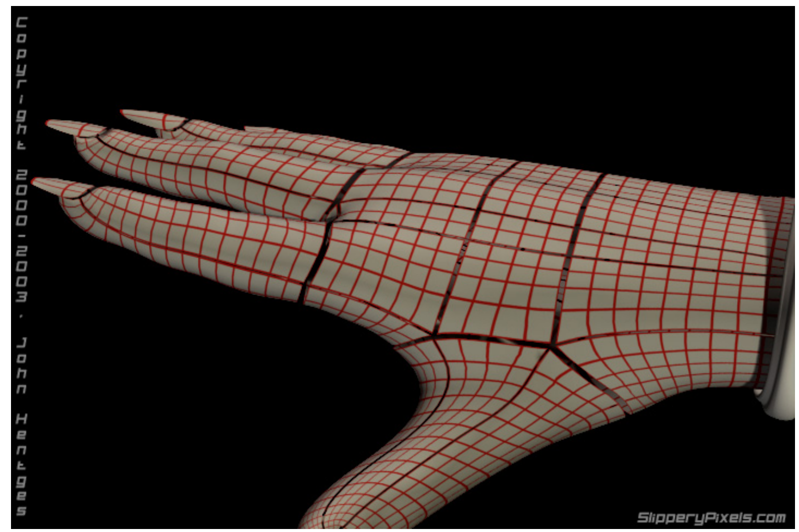




Problems with NURBS

- A single NURBS patch is either a topological disk, a tube or a torus

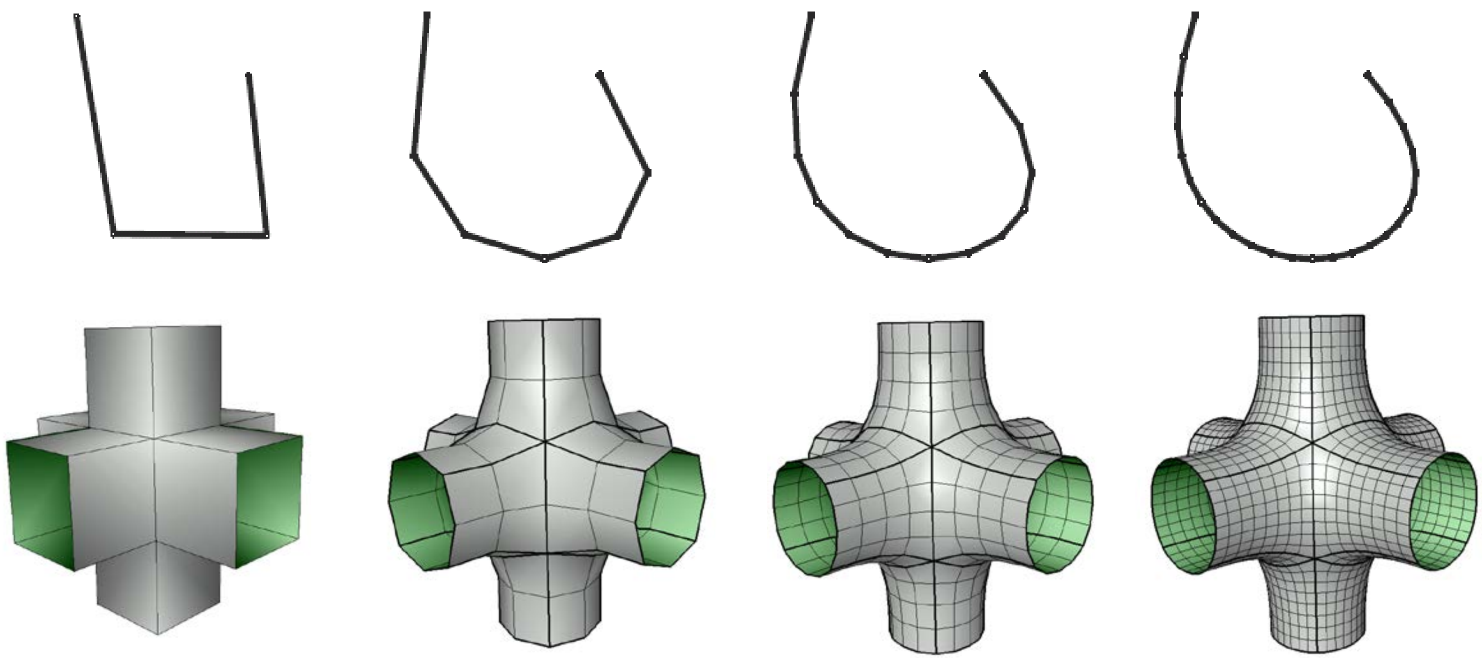
- Must use many NURBS patches to model complex geometry



- When deforming a surface made of NURBS patches, cracks arise at the seams

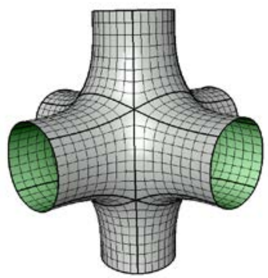
Subdivision

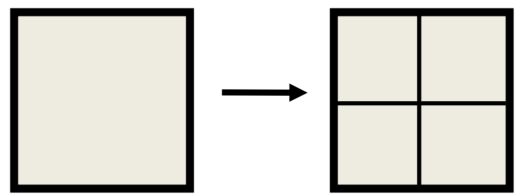
"Subdivision defines a smooth curve or surface as the limit of a sequence of successive refinements"



Subdivision Rules

- How the connectivity changes





- How the geometry changes
  - Old points
  - New points

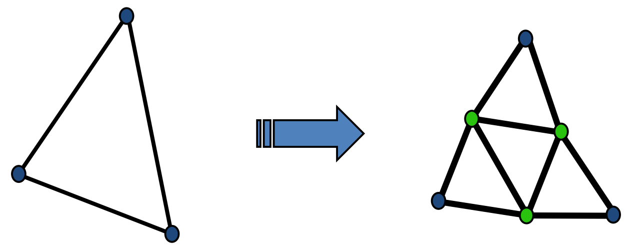
Design Goals for Subd Rules

- Efficiency
- Compact support
- Local definition
- Affine invariance
- Simplicity
- Smoothness

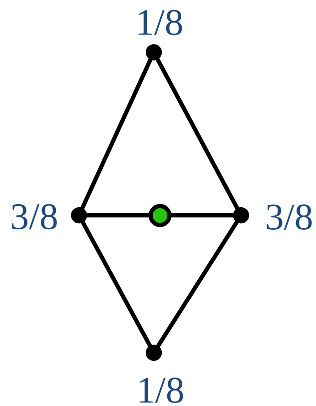
Same properties NURBS have, but will work for any topology

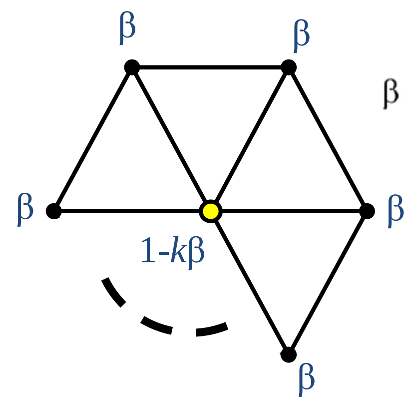
An Example – Loop Scheme

- Connectivity



- Geometry





$$\beta = \frac{1}{k}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2\right)$$

- Analysis?
  - Does it converge?
  - Is the limit surface smooth?
  - Any problems at extraordinary (valence != 6) vertices?

Parameterization of Subd Surfaces

- B-spline curves and surfaces are *parameterized*

$$\boldsymbol{S}(t):[t_0,t_m]\rightarrow R^2$$

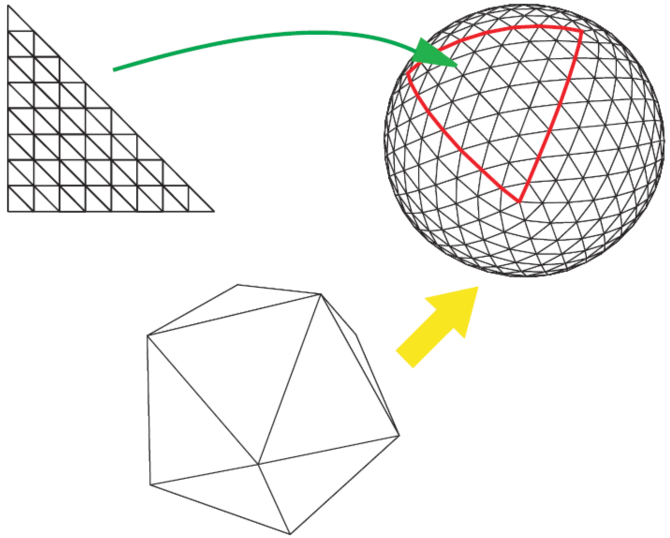
$$\boldsymbol{S}(u,v):[u_0,u_h]\times[v_0,v_k]\rightarrow R^3$$

- To analyze subd schemes, we need a similar parameterization
- Which domain to use? A planar rectangle cannot work
- Solution: Use initial control mesh as the domain

Parameterization of Subd Surfaces

- Apply subd rules to initial mesh, *without updating the geometry*

- Use resulting polyhedron as the domain



Smoothness of Surfaces
C^1 continuity

A surface $\boldsymbol{f} : |K| \rightarrow R^3$ is C^1 *continuous* if for every point $x \in |K|$ there exists a *regular parameterization* $\pi : D \rightarrow f(U_x)$, over a unit disk D in the plane, where U_x is the neighborhood in $|K|$ of x.

A *regular parameterization* π is one that is continuously differentiable, one-to-one, and has a Jacobi matrix of maximum rank.

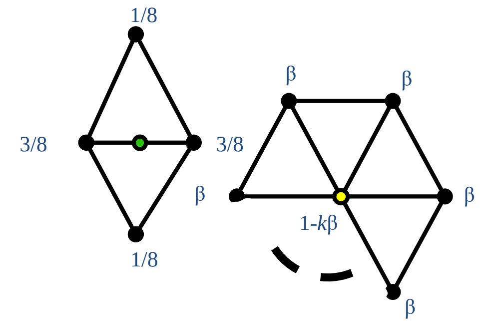
Subdivision Properties

- How can we prove properties of a subdivision scheme?

- Express the subdivision as a local matrix operation

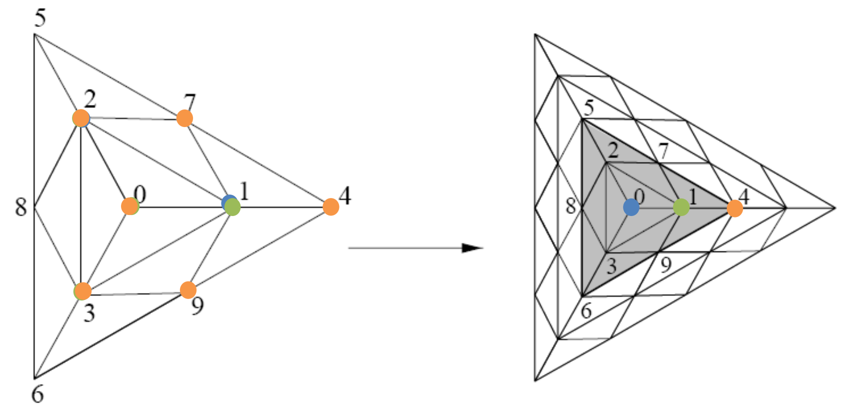
- Prove properties (convergence, continuity, affine invariance, etc.) using eigen-analysis.

Subdivision Matrix



- Look at the local neighborhood of an extraordinary vertex v

- Let U^j be the set of vertices in the 2-ring neighborhood of v after j Loop subdivision steps



Step j Step $j + 1$

- Let $\boldsymbol{p}_i^{\,j}$ be the corresponding control points

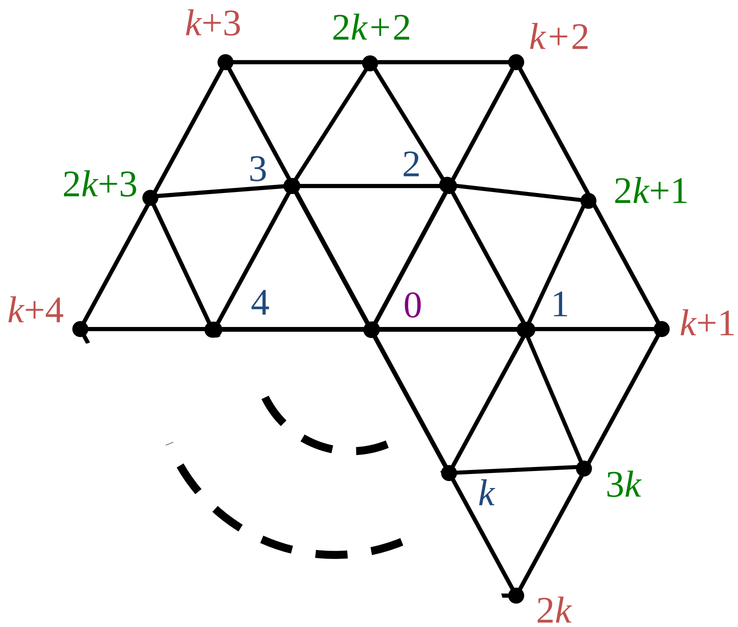
- We can compute $\boldsymbol{p}_i^{\,j+1}$ using only $\boldsymbol{p}_i^{\,j}$

Subdivision Matrix

- An extraordinary vertex of degree k has $\hat{k} = 3k + 1$ vertices in its 2-ring neighborhood

$$\begin{pmatrix} p_0^{j+1} \\ \vdots \\ p_{3k}^{j+1} \end{pmatrix}_{\hat{k}\times 3} = S_{\hat{k}\times\hat{k}} \begin{pmatrix} p_0^{j} \\ \vdots \\ p_{3k}^{j} \end{pmatrix}_{\hat{k}\times 3}$$

$$\boldsymbol{p}^{j+1} = S\boldsymbol{p}^{j}$$



- S depends on k

Subdivision Matrix

- Assume S has a full set of eigenvectors $\{\varphi_i\}$ with corresponding real eigenvalues $\{\lambda_i\}$, arranged in non-increasing order

- We can express $\boldsymbol{p}^0$ in terms of the eigenvectors of S:

$$\left(\boldsymbol{p}^0\right)_{\hat{k}\times 3} = \sum_i \left(\varphi_i\right)_{\hat{k}\times 1} \left(\boldsymbol{a}_i\right)_{1\times 3}$$

- Where:

$$\boldsymbol{a}_i = \varphi_i^{\,T} \boldsymbol{p}^0$$

Subdivision Matrix

- Now we can express $\boldsymbol{p}^j$ using only a_i and the eigenvalues and eigenvectors of S:

$$
\begin{aligned}
\boldsymbol{p}^j &= S^j \sum_i \varphi_i \boldsymbol{a}_i \\
&= \sum_i S^j \varphi_i \boldsymbol{a}_i \\
&= \sum_i (\lambda_i)^j \varphi_i \boldsymbol{a}_i
\end{aligned}
$$

- We used the fact: $S^j \varphi_i = (\lambda_i)^j \varphi_i$

$$\boldsymbol{p}^j = \sum_i (\lambda_i)^j \varphi_i \boldsymbol{a}_i$$

Subdivision Properties

- *Convergence:* We need $|\lambda_i| \leq 1$ for all i (and only one eigenvalue can be exactly 1)

- *Affine invariance:* If for all A,B:

$$S \cdot \left(\boldsymbol{p}^j \cdot A_{3\times3} + \mathbf{1}_{\hat{k}\times1} \cdot B_{1\times3} \right) = S \cdot \left(\boldsymbol{p}^j \right) \cdot A_{3\times3} + \mathbf{1}_{\hat{k}\times1} \cdot B_{1\times3}$$

$$S \cdot \left(\boldsymbol{p}^j \right) \cdot A + S \cdot \mathbf{1} \cdot B = S \cdot \left(\boldsymbol{p}^j \right) \cdot A + \mathbf{1} \cdot B$$

For affine invariance we need $S \cdot \mathbf{1} = \mathbf{1}$

→ $\lambda_0 = 1$

$$\boldsymbol{p}^j = \sum_i (\lambda_i)^j \varphi_i \boldsymbol{a}_i$$

Subdivision Properties

- *Limit position:*

$$\boldsymbol{p}^\infty = \lim_{j\to\infty} \boldsymbol{p}^j = \lim_{j\to\infty} \sum_i (\lambda_i)^j \varphi_i \boldsymbol{a}_i$$

Since $|\lambda_i| < 1$ for all $i > 0$, we have:

$$\boldsymbol{p}^\infty = (\lambda_0)^j \varphi_0 \boldsymbol{a}_0 = \mathbf{1} \cdot \boldsymbol{a}_0$$

In the limit the 2-neighborhood of *v* is mapped to the same position: $\boldsymbol{a}_0$

We can compute the limit positions without recursively applying the subdivision

Subdivision Properties

- *Limit tangent plane:*

  Since scheme is translation invariant, fix $\boldsymbol{a}_0 = \mathbf{0}$

  Assume $\lambda = \lambda_1 = \lambda_2 > \lambda_3$

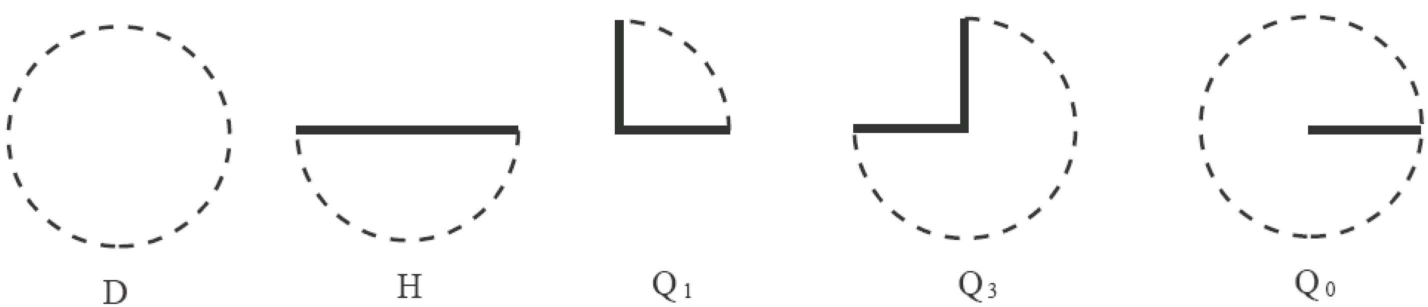
$$\frac{\boldsymbol{p}^j}{(\lambda)^j} = \varphi_1 \boldsymbol{a}_1 + \varphi_2 \boldsymbol{a}_2 + \sum_{i \geq 3} \left(\frac{\lambda_i}{\lambda} \right)^j \varphi_i \boldsymbol{a}_i$$

For large enough j, the 2-neighborhood of v is mapped to linear combinations of a_1 and a_2

a_1 and a_2 span the tangent plane of the limit surface at v

Piecewise Smooth Surfaces

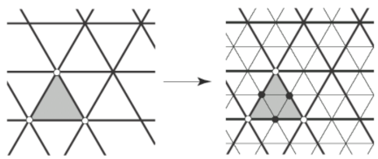
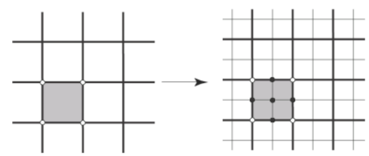
- So far, only considered closed smooth surfaces
- Surfaces have boundaries and creases
- A subdivision scheme should have rules for all the following cases:

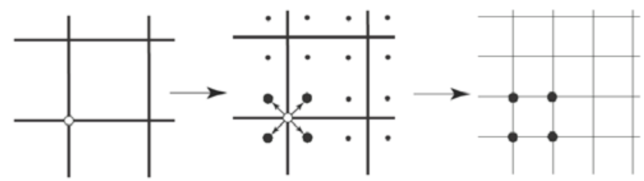


Interior | Smooth boundary | Convex boundary | Concave boundary | Crease

Subdivision Zoo

- Can be classified according to:

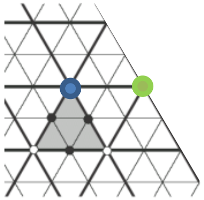
Primal (face split)		
	 *Triangular meshes*	 *Quad Meshes*
Approximating	Loop(C^2)	Catmull-Clark(C^2)
Interpolating	Mod. Butterfly (C^1)	Kobbelt (C^1)

 **Dual (vertex split)**
Doo-Sabin, Midedge(C^1)
Biquartic (C^2)

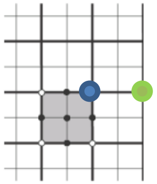
- Many more...

Terminology

- *Regular* vertices
  - Tri meshes
    - In the interior - degree 6
    - On the boundary – degree 4



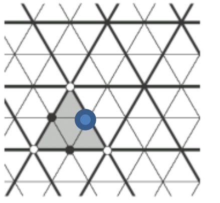
  - Quad meshes
    - In the interior - degree 4
    - On the boundary – degree 3



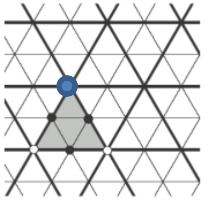
- *Extraordinary* vertices – all the rest

Terminology

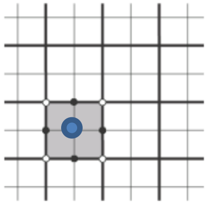
- *Odd* vertices – new vertices at current subdivision level



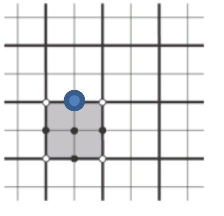
- *Even* vertices – vertices inherited from previous level



- *Face* vertices – odd vertices inserted in a face



- *Edge* vertices – odd vertices inserted on an edge



Boundaries and Creases

- Special subdivision rules will be given for each scheme for the boundary vertices

- The boundary curve of the limit surface:
  - Should not depend on interior control vertices
    - In case two surfaces will be merged along the boundary
  - Should be C^1 or C^2

- Use boundary rules for edges tagged as *creases*

Loop Scheme

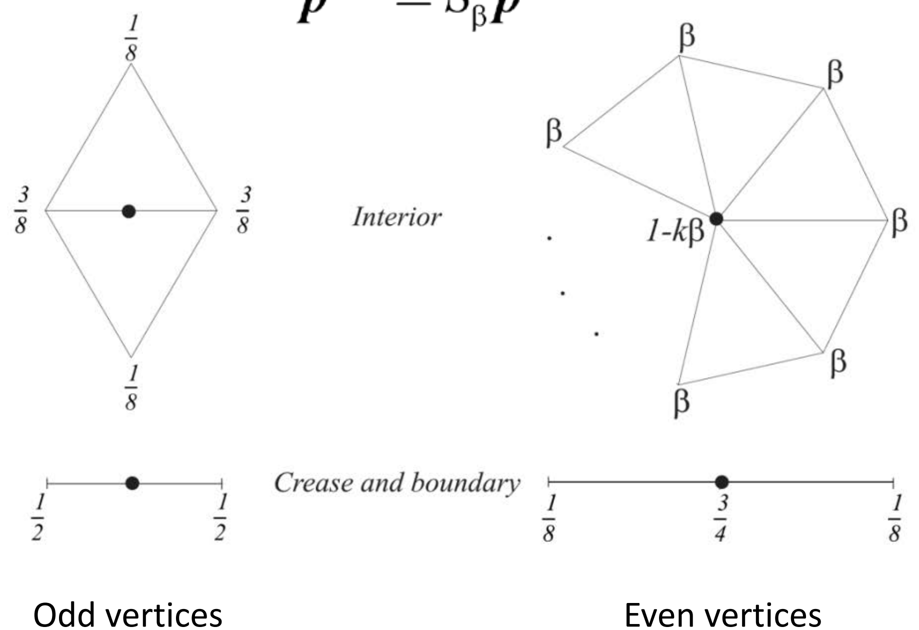
- Possible choices for β
  - Original (by Loop):

$$\beta = \frac{1}{k}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2\right)$$

  - Or (by Warren):

$$\beta = \begin{cases} \frac{3}{16} & n = 3 \\ \frac{3}{8n} & n > 3 \end{cases}$$

$$\boldsymbol{p}^{j+1} = S_\beta \boldsymbol{p}^j$$



Loop Scheme

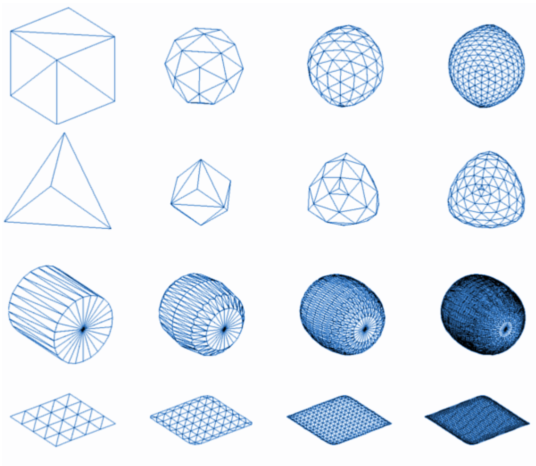
- Limits for interior vertex v (using control points from *any* subd level j)
  - Position

$$p^{\infty} = S_{\chi} p^{j} \quad ; \quad \chi = \frac{1}{3/8\beta + k}$$

  - Tangents

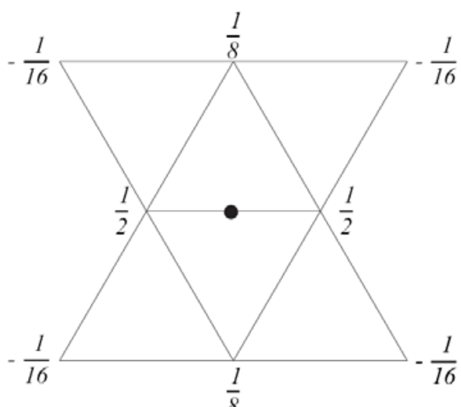
$$t_1 = \sum_{i=0}^{k-1} \cos \frac{2\pi i}{k} p_i^j \quad ; \quad t_2 = \sum_{i=0}^{k-1} \sin \frac{2\pi i}{k} p_i^j$$

Where v_i^j for $i = \{0,..,k-1\}$ are the one ring neighbors of the vertex v at subd level j, and p_i^j are the corresponding control points

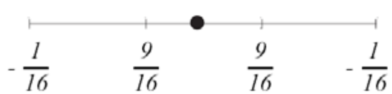


Modified Butterfly Scheme

- Interpolating scheme
  - Even vertices don't move

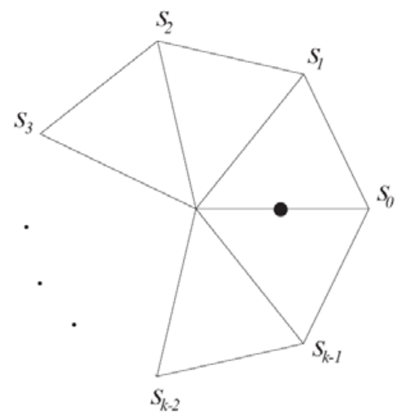


Mask for interior odd vertices with regular neighbors



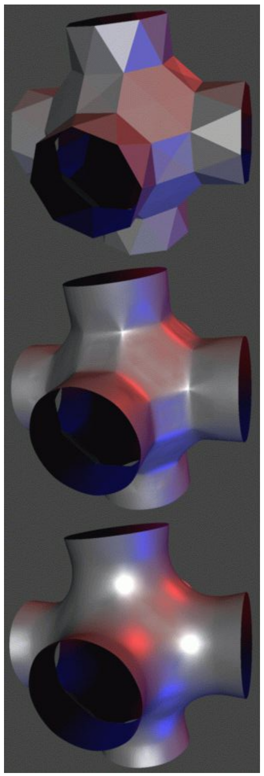
Mask for crease and boundary vertices

a. Masks for odd vertices



$$s_i = \begin{cases} \left\{\frac{5}{12}, -\frac{1}{12}, -\frac{1}{12}\right\} & k = 3 \\ \left\{\frac{3}{8}, 0, -\frac{1}{8}, 0\right\} & k = 4 \\ \frac{1}{k}\left(\frac{1}{4} + \cos\frac{2\pi i}{k} + \frac{1}{2}\cos\frac{4\pi i}{k}\right) & k \geq 5 \end{cases}$$

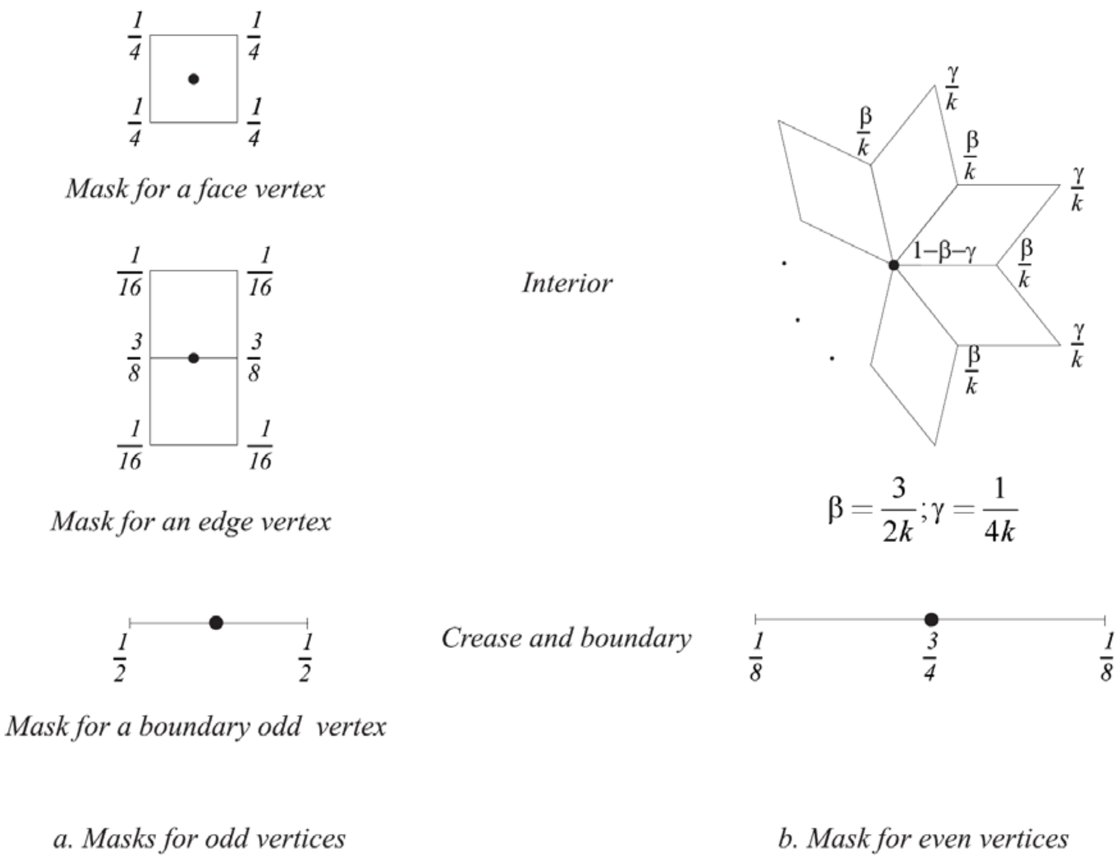
b. Mask for odd vertices adjacent to an extraordinary vertex

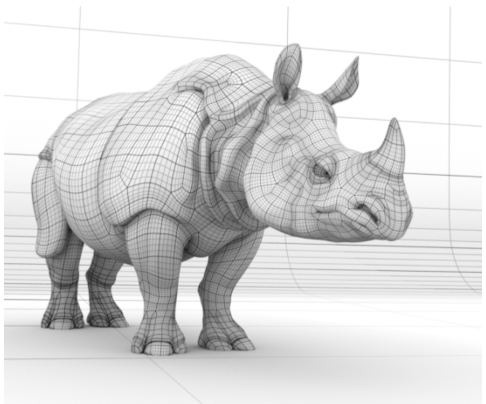


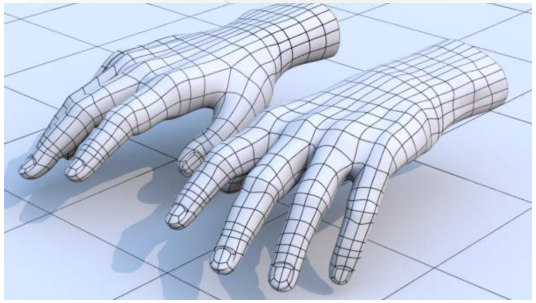
Butterfly (not C^1)

Modified Butterfly

Catmull-Clark Quad Scheme





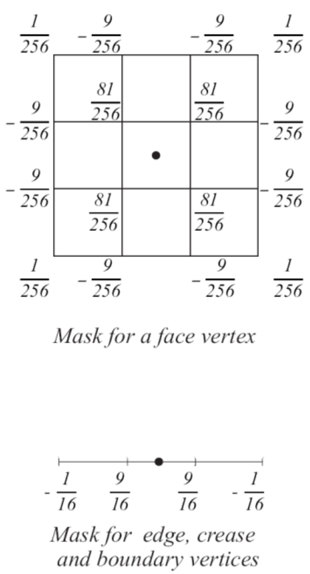


Can be modified to work on general polygons

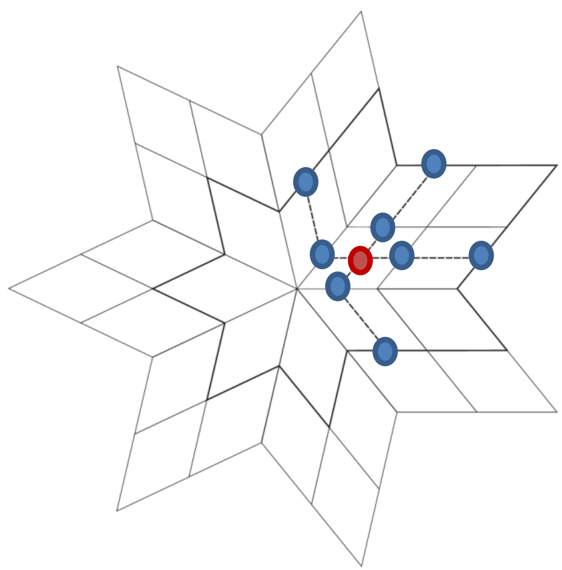
Kobbelt Scheme

Main observation - to compute a face control point:

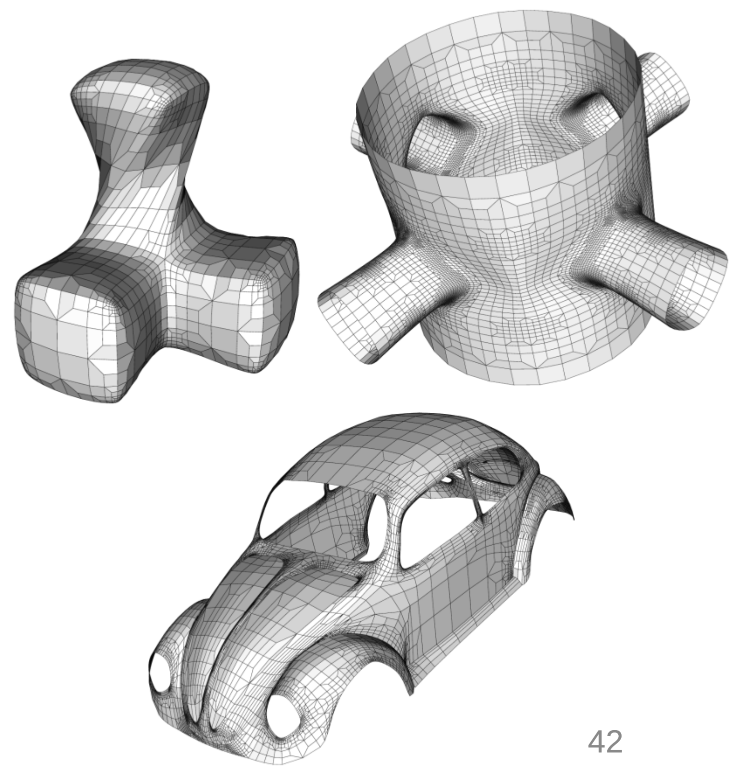
- compute all *edge control points*
- compute *face control points* using the *edge rule* applied to edge control points on same level



a. Regular masks

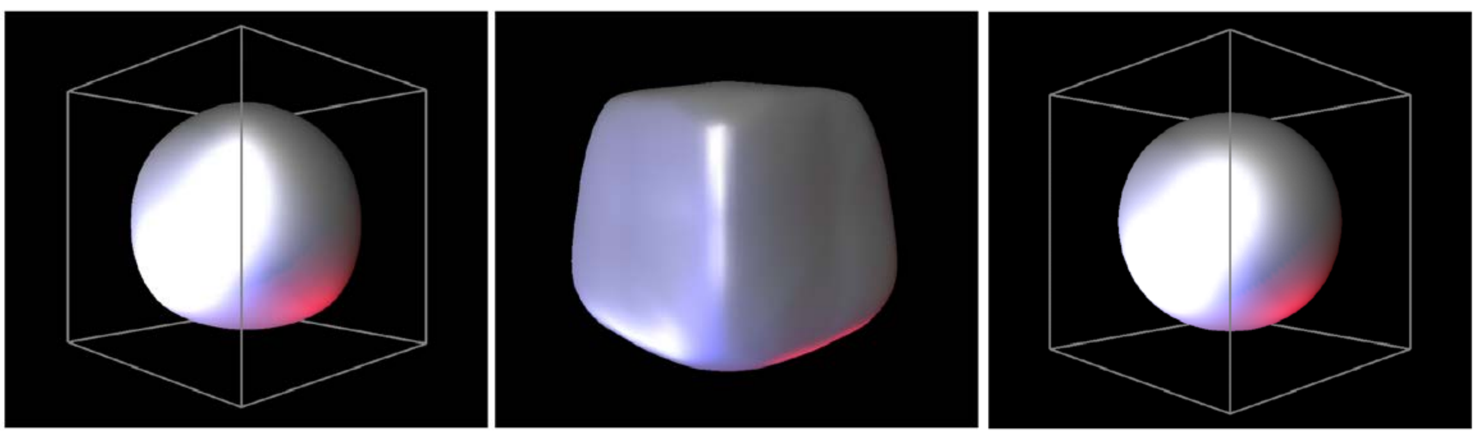


b. Computing a face vertex adjacent to an extraordinary vertex



Scheme Comparison

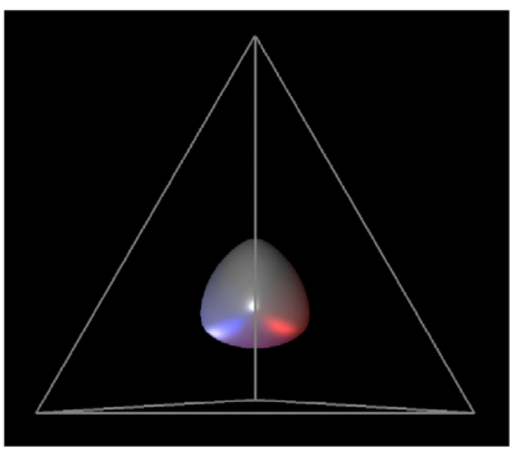
- Subdividing a cube
  - Loop result is asymetric, because cube was triangulated first
  - Both Loop and Catmull-Clark are better then Butterfly (C^2 vs. C^1)
  - Interpolation vs. smoothness



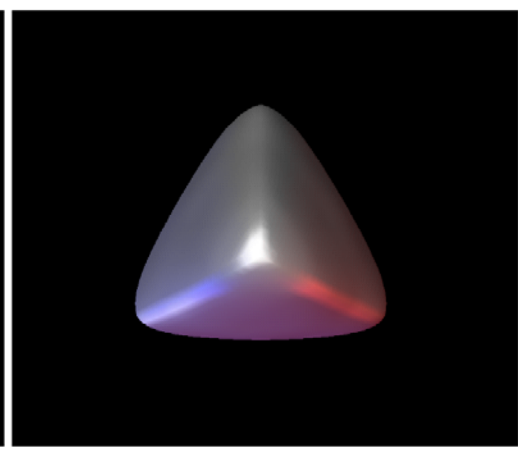
Loop *Butterfly* *Catmull-Clark*

Scheme Comparison

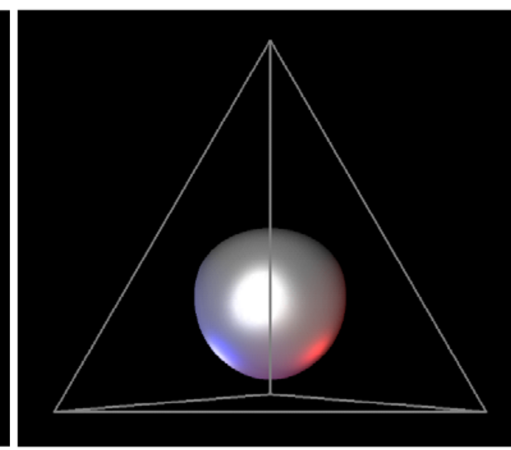
- Subdividing a tetrahedron
  - Same insights
  - Severe shrinking for approximating schemes



Loop



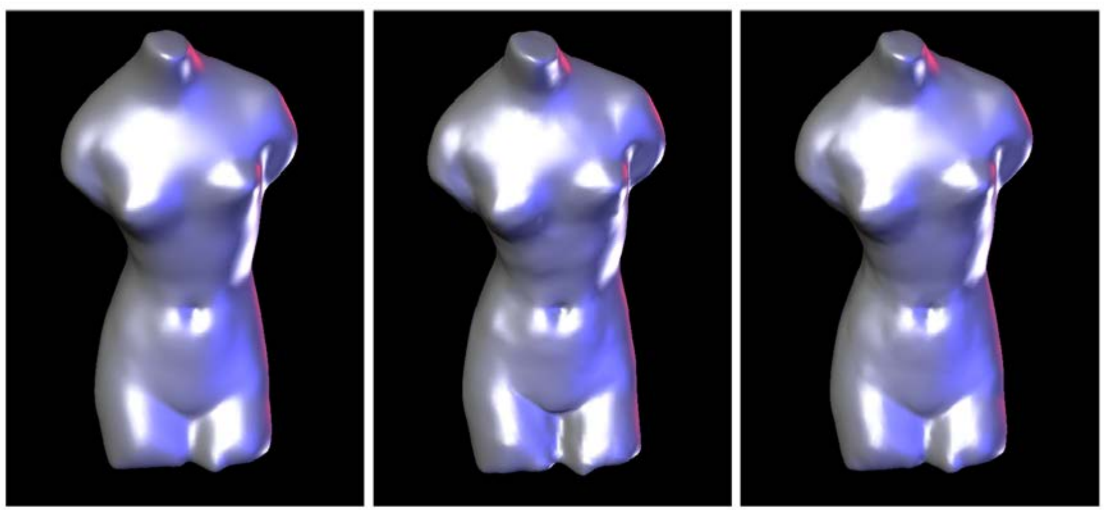
Butterfly



Catmull-Clark

Scheme Comparison

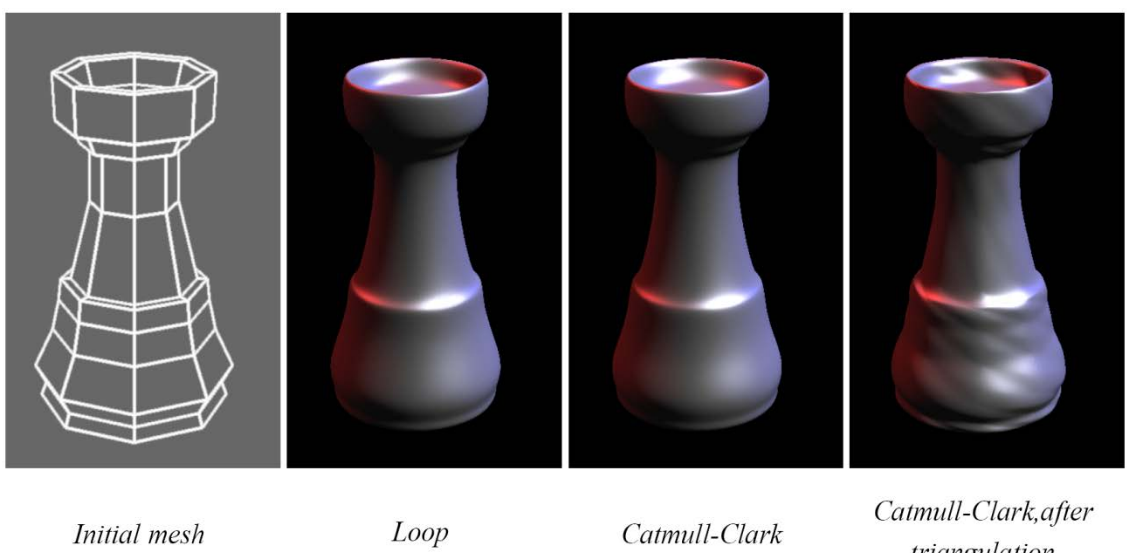
- Spot the difference?
- For smooth meshes with uniform triangle size, different schemes provide very similar results
- Beware of interpolating schemes for control polygons with sharp features



Loop *Butterfly* *Catmull-Clark*

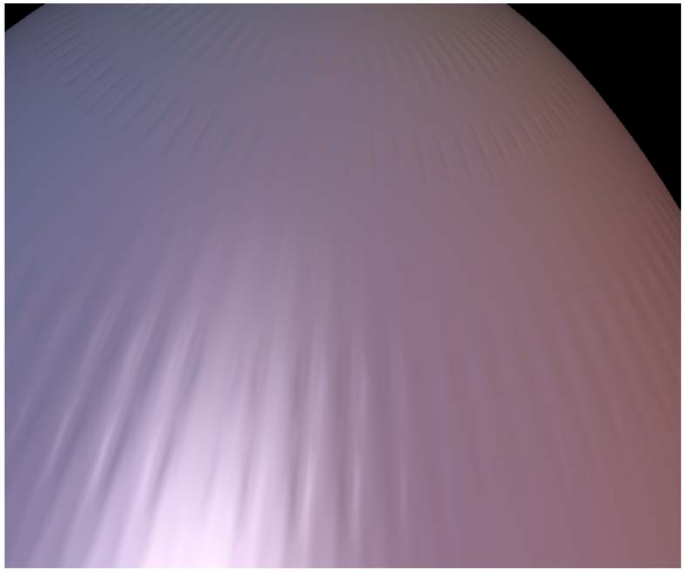
So Who Wins?

- Loop and Catmull-Clark best when interpolation is not required
- Loop best for triangular meshes
- Catmull-Clark best for quad meshes
  - Don't triangulate and then use Catmull-Clark



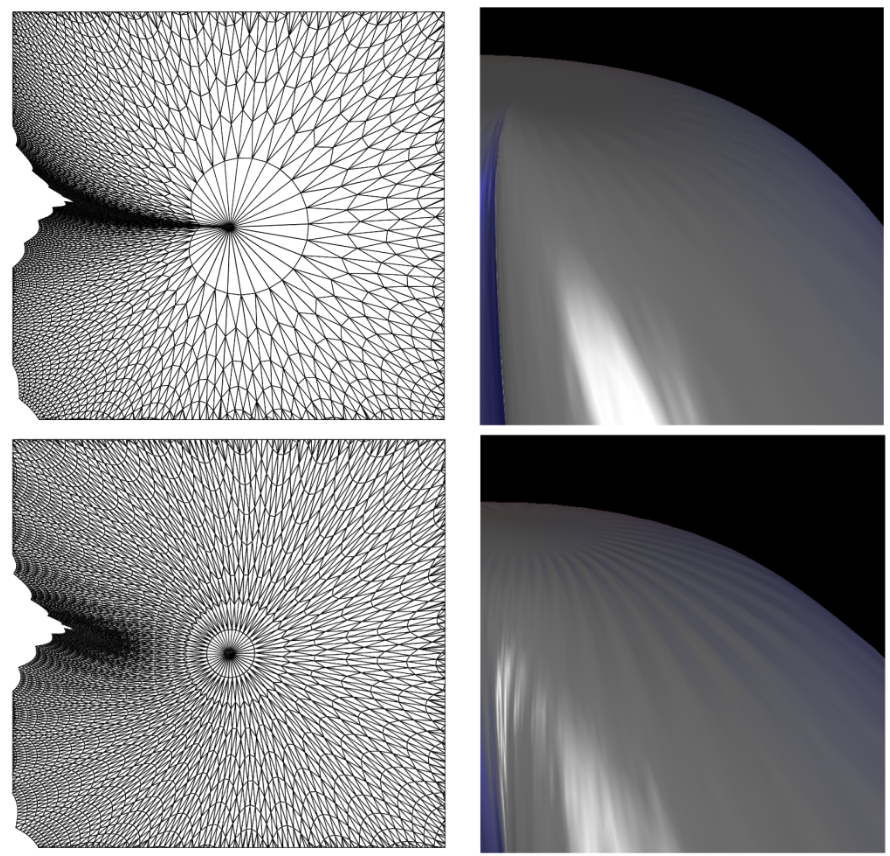
The Dark Side of Subd Surfaces

- Problems with curvature continuity
  - Requires either very large support, or forces 0 curvature at extraordinary vertices
  - Generates ripples near vertices of large degree



The Dark Side of Subd Surfaces

- Decreased smoothness with degree
  - For large degrees, third eigenvalue approaches second and first eigenvalues
  - Generates creases



- Can fix by modifying scheme
  - Creates more ripples