CSE528 Computer Graphics: Theory, Algorithms, and Applications

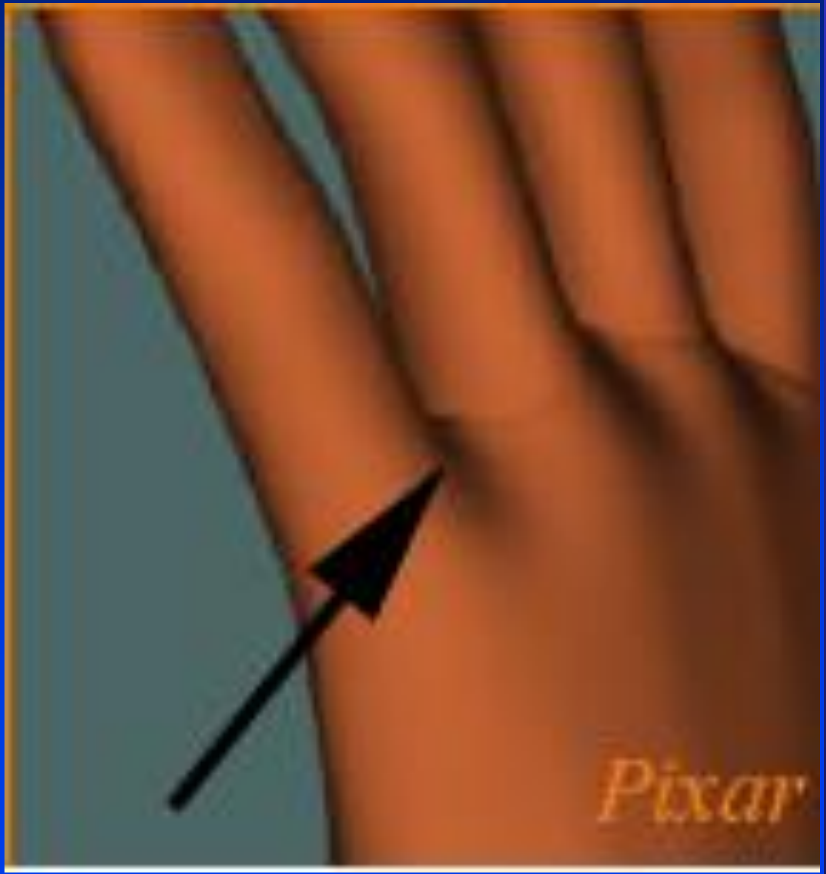
Hong Qin
Department of Computer Science
Stony Brook University (SUNY at Stony Brook)
Stony Brook, New York 11794-2424
Tel: (631)632-8450; Fax: (631)632-8334
qin@cs.stonybrook.edu
http://www.cs.stonybrook.edu/~qin

Non-Uniform Rational B-Splines

NURBS

Pixar Animation Character 'Woody' in Toy Story

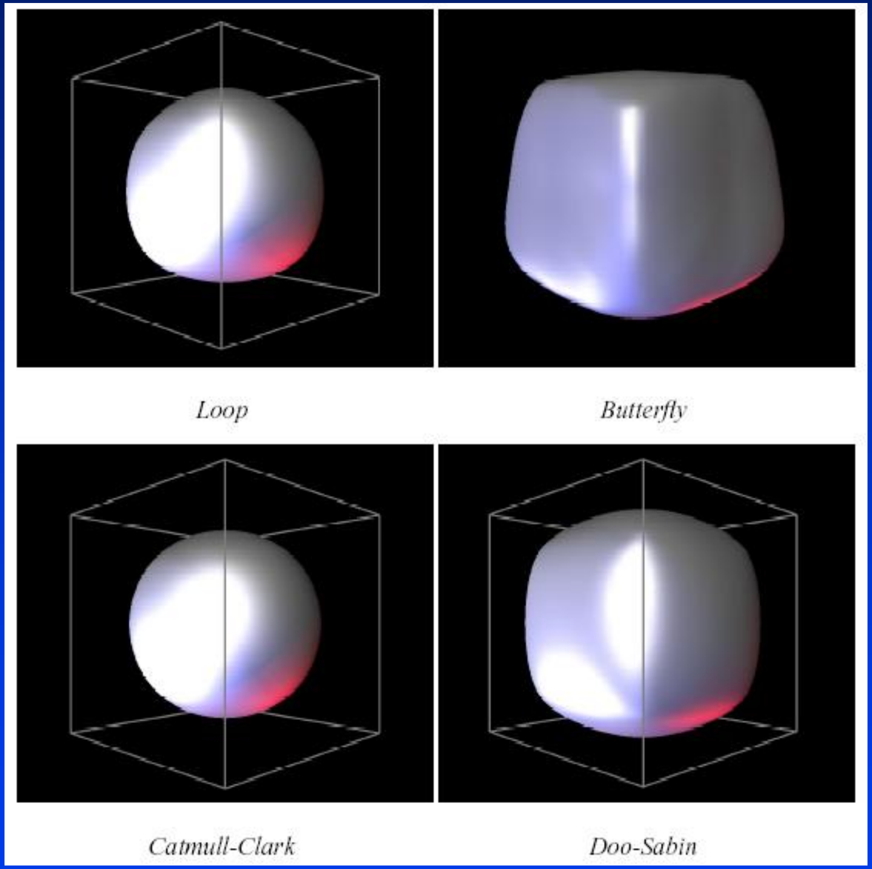
- Problems: Topological Restrictions Occur!
  - Trimming NURBS is expensive and can have numerical errors
  - When used in animation, very hard to hide seams



Subdivision Schemes for Interactive Surface Modeling

What is Subdivision

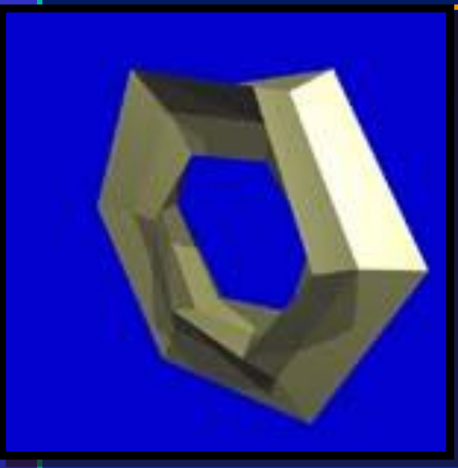
- Construct a surface from an arbitrary polyhedron
  - Subdivide each face of the polyhedron
- The limit will be a smooth surface

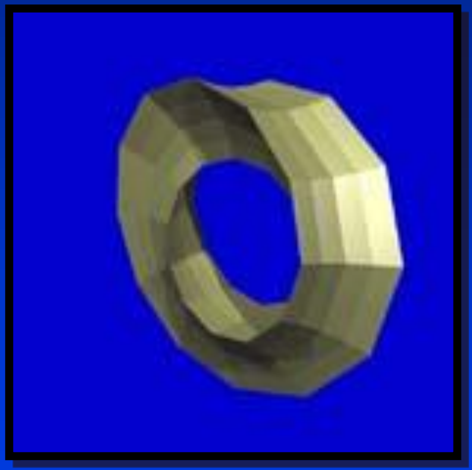


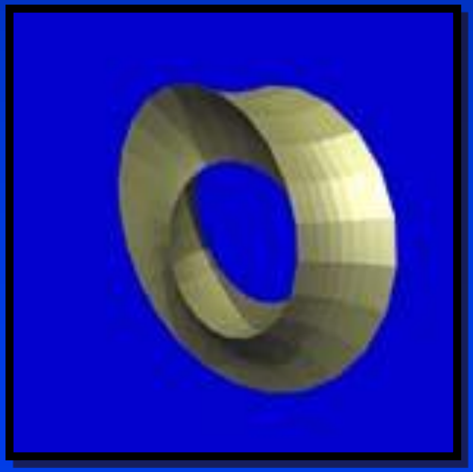
Subdivision Schemes

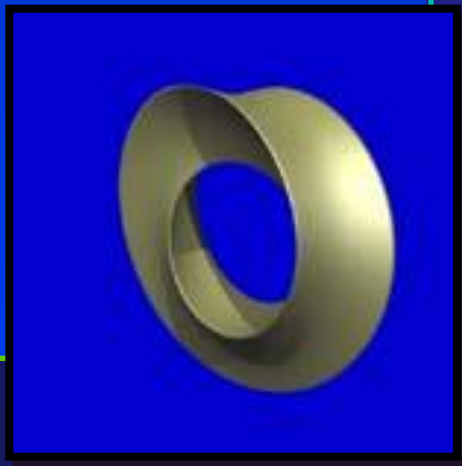
Subdivision Surfaces

Subdivision surface
(different levels of refinement)



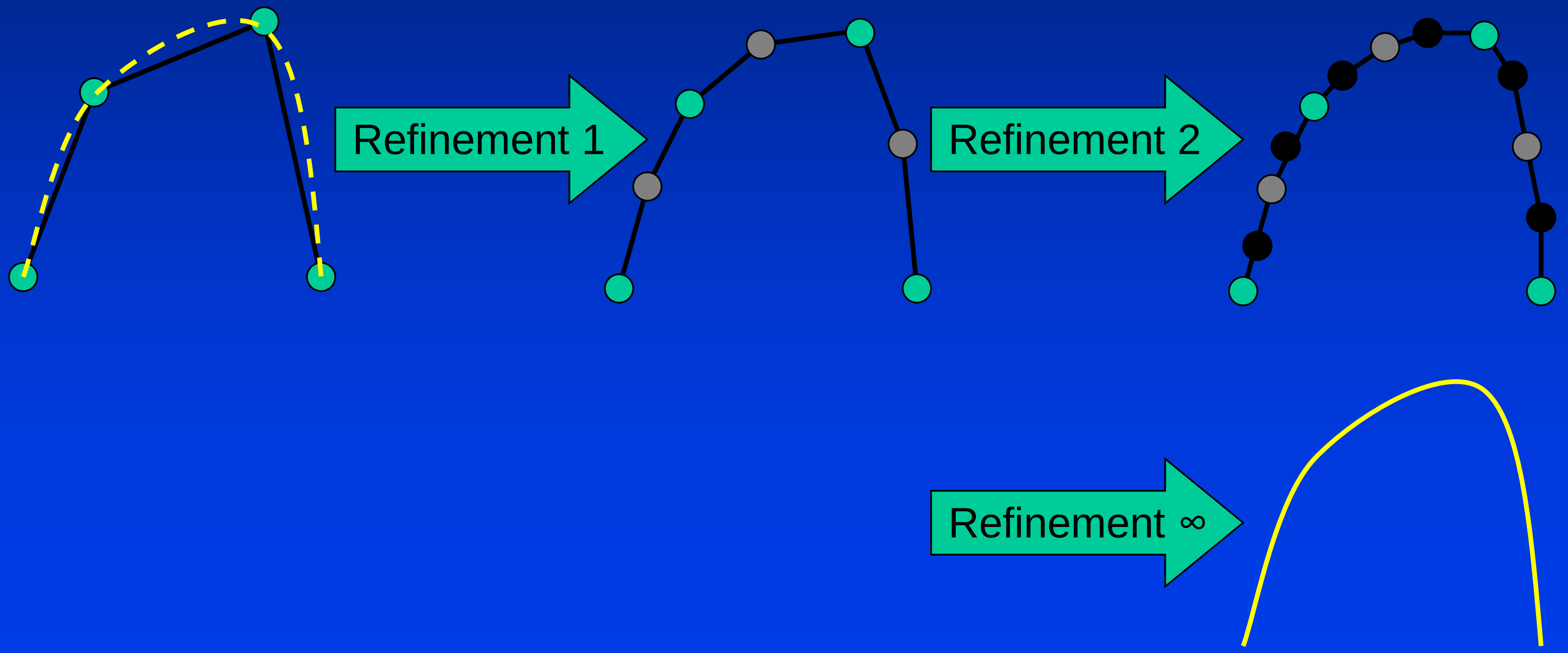




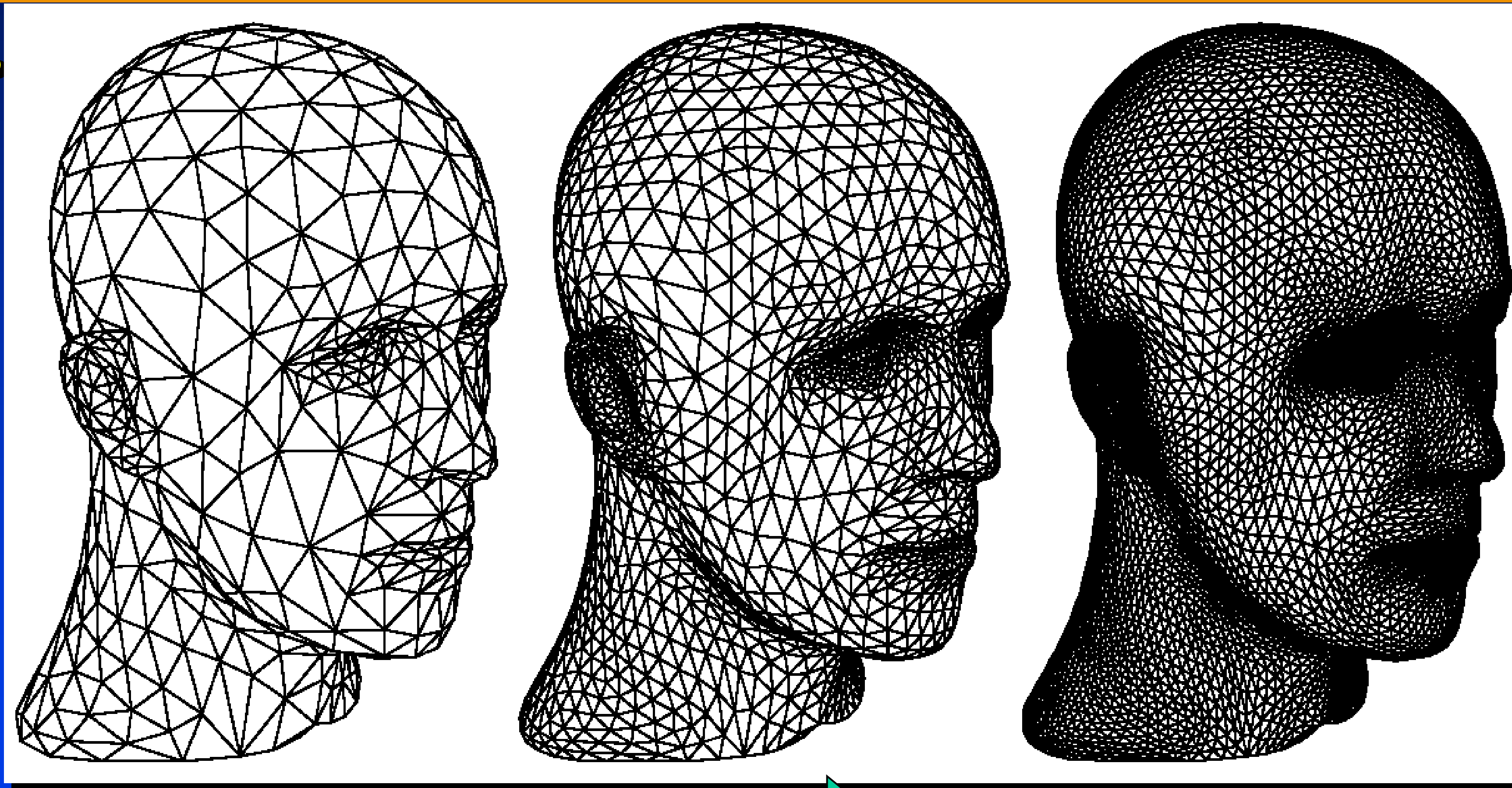


Subdivision: Key Idea

- Approach limit curve through an iterative refinement process



Subdivision in 3D

- 

Refinement

Subdivision Surfaces: Motivation

- How do we represent curved surfaces in the computer?
  - Efficiency of representation
  - Continuity
  - Affine invariance
  - Efficiency of rendering
- How do they relate to splines/patches?
- Why use subdivision rather than patches?

Subdivision Type

- Interpolating schemes
  - Limit surfaces/curve will pass through original set of data points.
- Approximating schemes
  - Limit surface will not necessarily pass through the original set of data points.

Subdivision in Production Environment

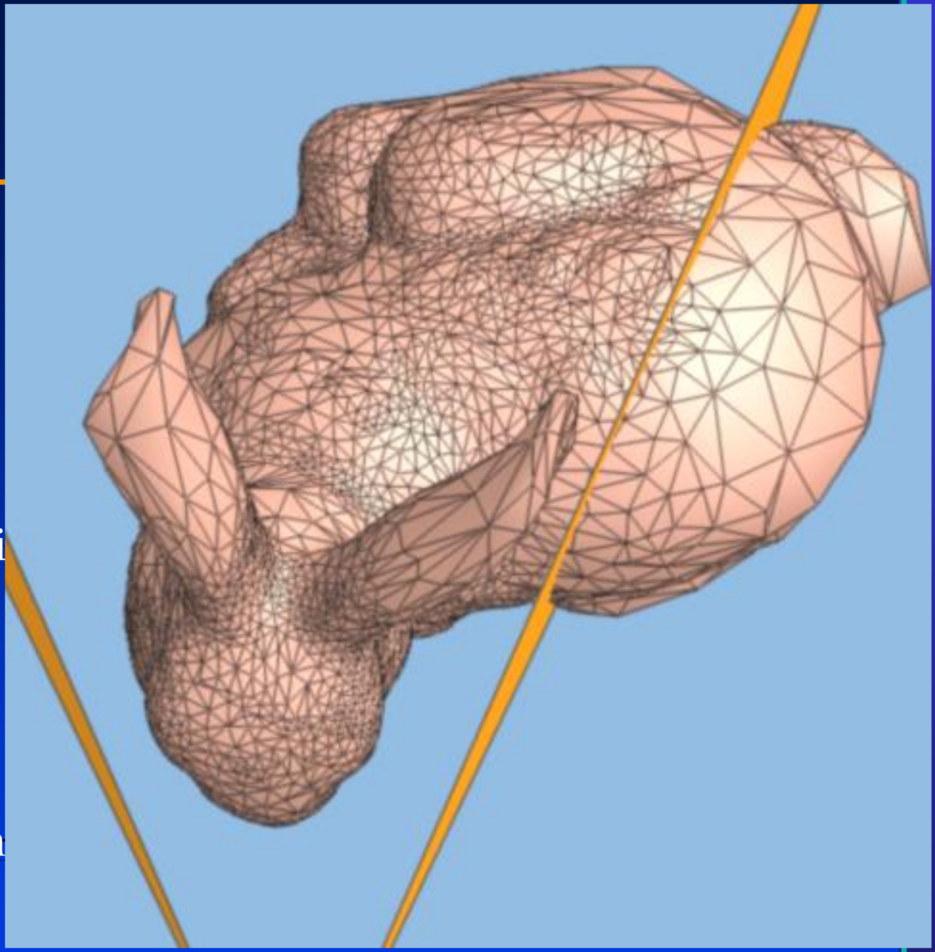
- Traditionally spline patches (NURBS) have been used in production for character animation.
- Difficult to control spline patch density in character modeling.

(Geri's Game, Pixar 1998)

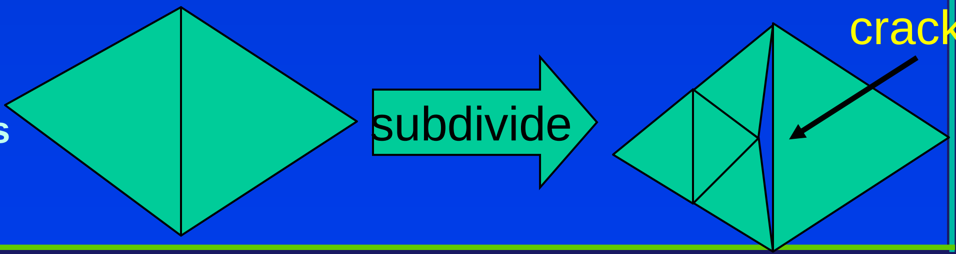
Subdivision in Character Animation
Tony Derose, Michael Kass, Tien Troung
(SIGGRAPH '98)

Adaptive Subdivision for Rendering

- Not all regions of a model need to be subdivided.
- Idea: Use some criteria and adaptively subdivide mesh where needed.
  - Curvature
  - Screen size (make triangles < size of pi
  - View dependence
    - Distance from viewer
    - Silhouettes
    - In view frustum
  - Careful! Must ensure that “cracks” aren



View-dependent refinement of progressive meshes
Hugues Hoppe.
(SIGGRAPH ’87)

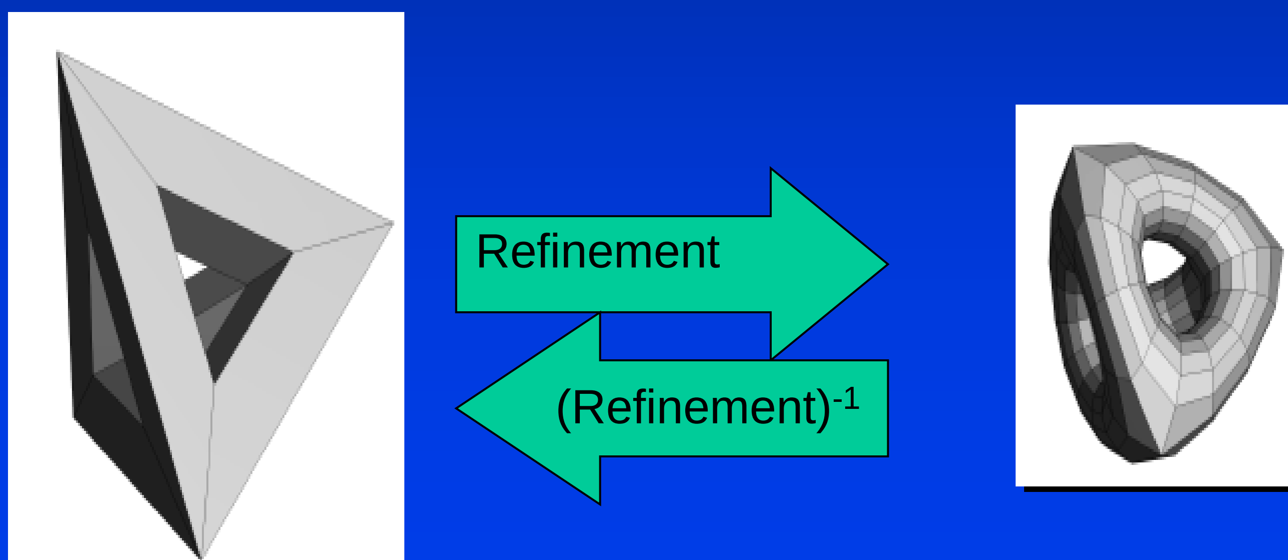


Subdivision for Compression

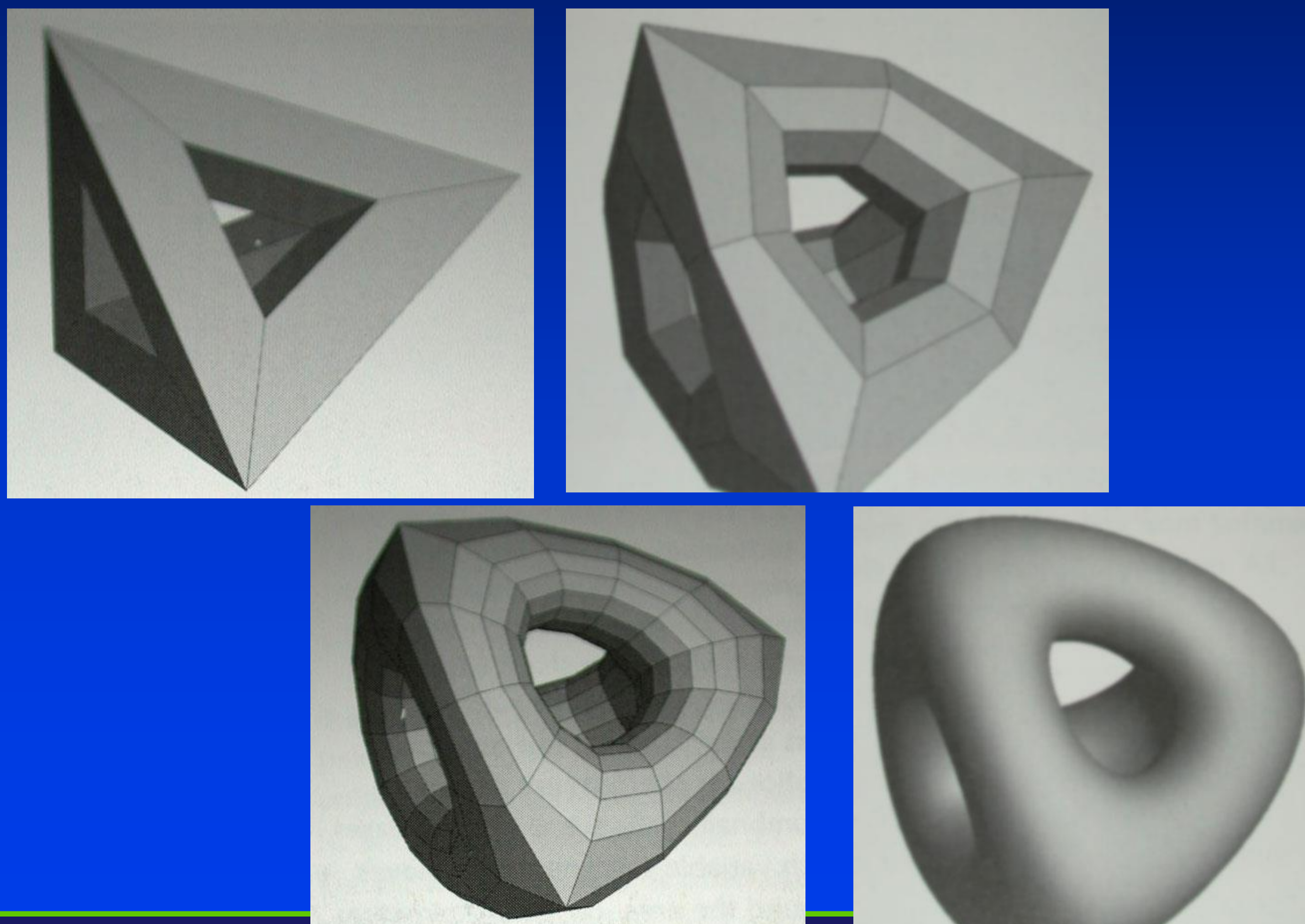
Progressive Geometry Compression

Andrei Khodakovsky, Peter Schröder and Wim Sweldens

(SIGGRAPH 2000)



Subdivision Surfaces



Introduction

- History of subdivision.
- What is subdivision?
- Why subdivision?

History of Subdivision Schemes

Stage I: Create smooth curves from arbitrary mesh

- de Rham, 1947.
- Chaikin, 1974.

Stage II: Generalize splines to arbitrary topology

- Catmull and Clark, 1978.
- Doo and Sabin, 1978.

Stage III: Applied in high end animation industry

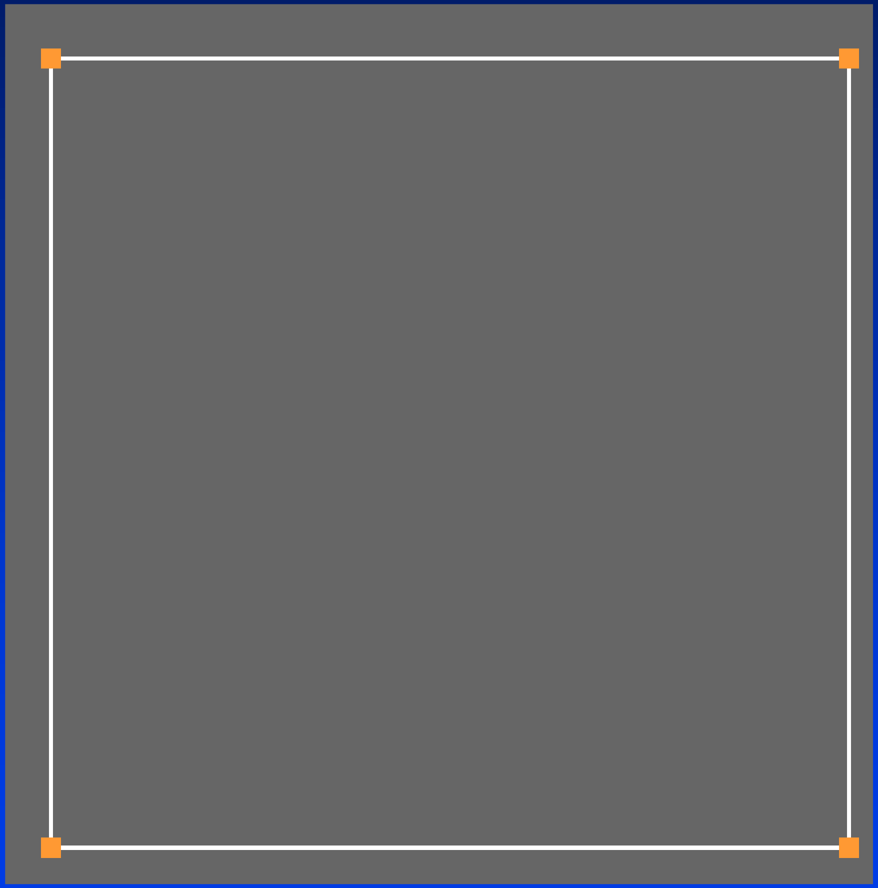
- Pixar Studio, "Geri's Game", 1998.

Stage IV: Applied in engineering design and CAD

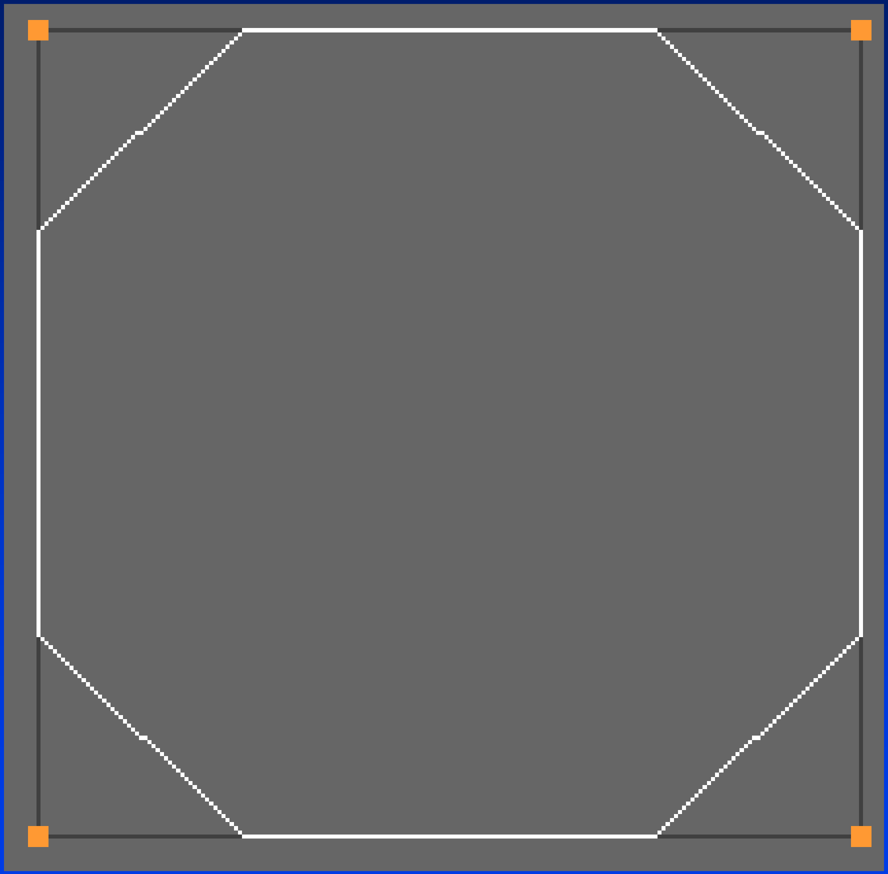
Basic Idea of subdivision

- Start from an initial control polygon.
- Recursively refine it by some rules.
- A smooth surface (curve) in the limit.

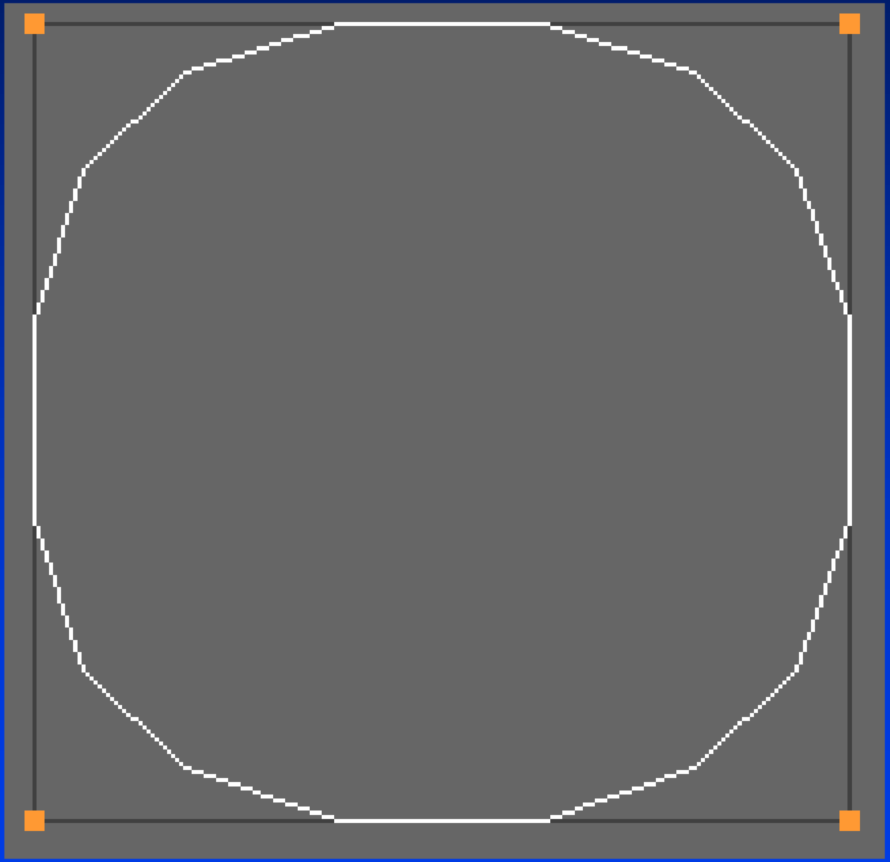
Chaikin's Corner Cutting Scheme



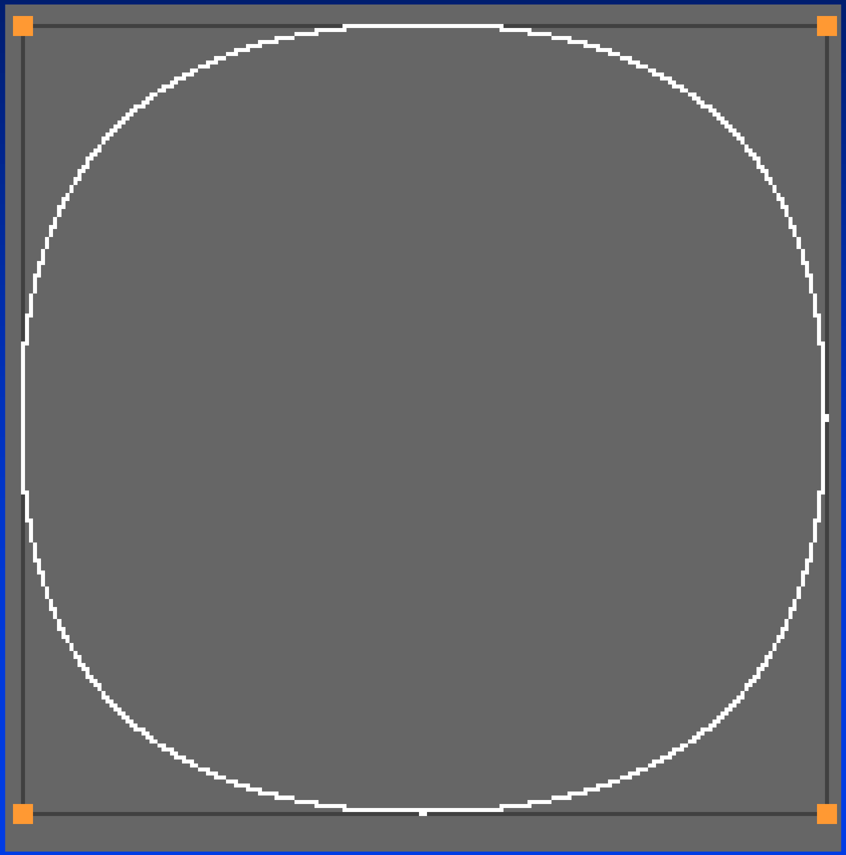
Chaikin's Corner Cutting Scheme



Chaikin's Corner Cutting Scheme



Chaikin's Corner Cutting Scheme



Chaikin's Algorithm

- A set of control points to define a polygon

$$\mathbf{p}_0^0, \mathbf{p}_1^0, \mathbf{p}_2^0, \ldots, \mathbf{p}_n^0$$

- Subdivision process (more control vertices)
- Rules (corner chopping)

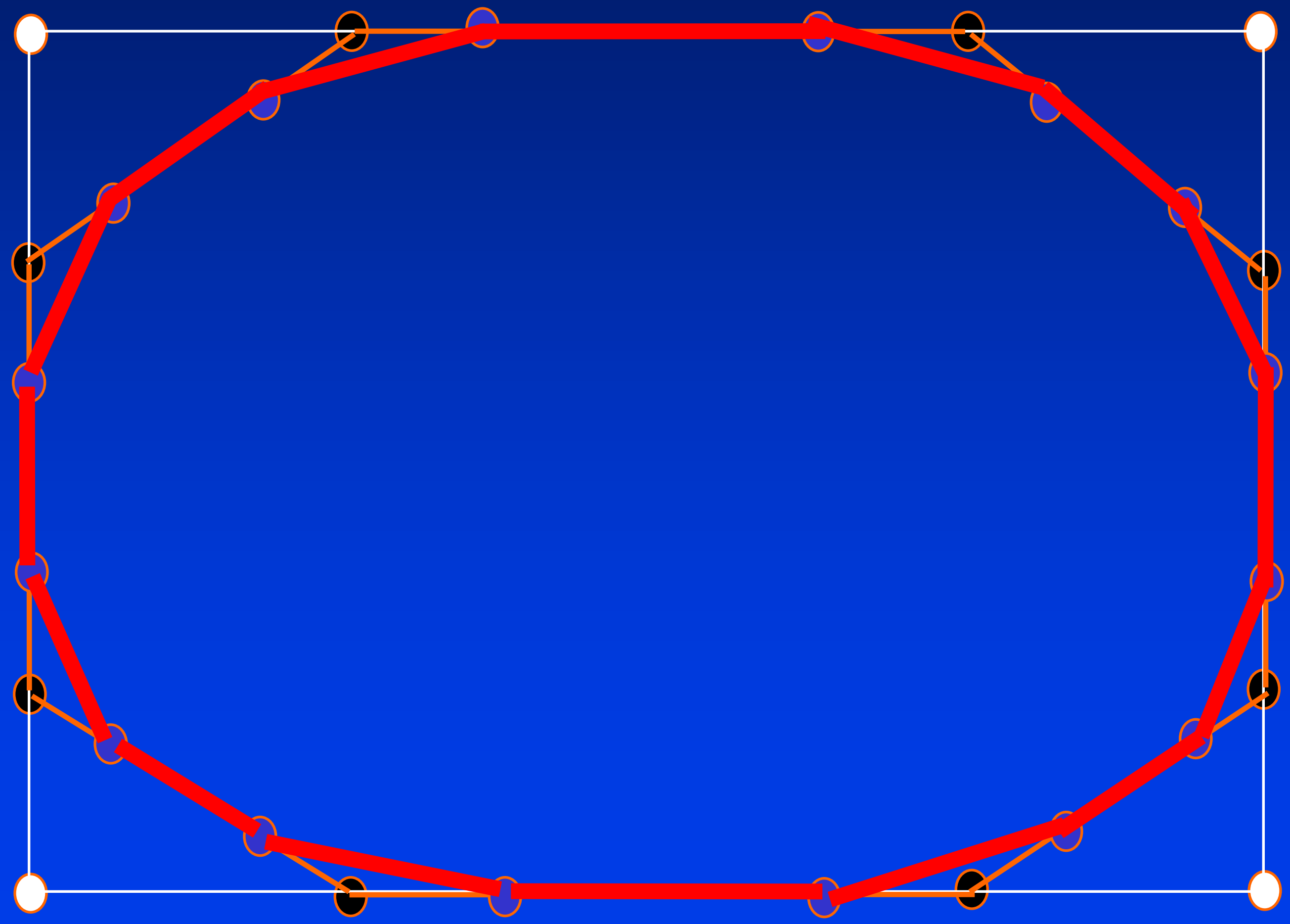
$$\mathbf{p}_0^k, \mathbf{p}_1^k, \mathbf{p}_2^k, \ldots \mathbf{p}_{2^k n}^k$$

$$\mathbf{p}_{2i}^{k+1} = \frac{3}{4}\mathbf{p}_i^k + \frac{1}{4}\mathbf{p}_{i+1}^k$$

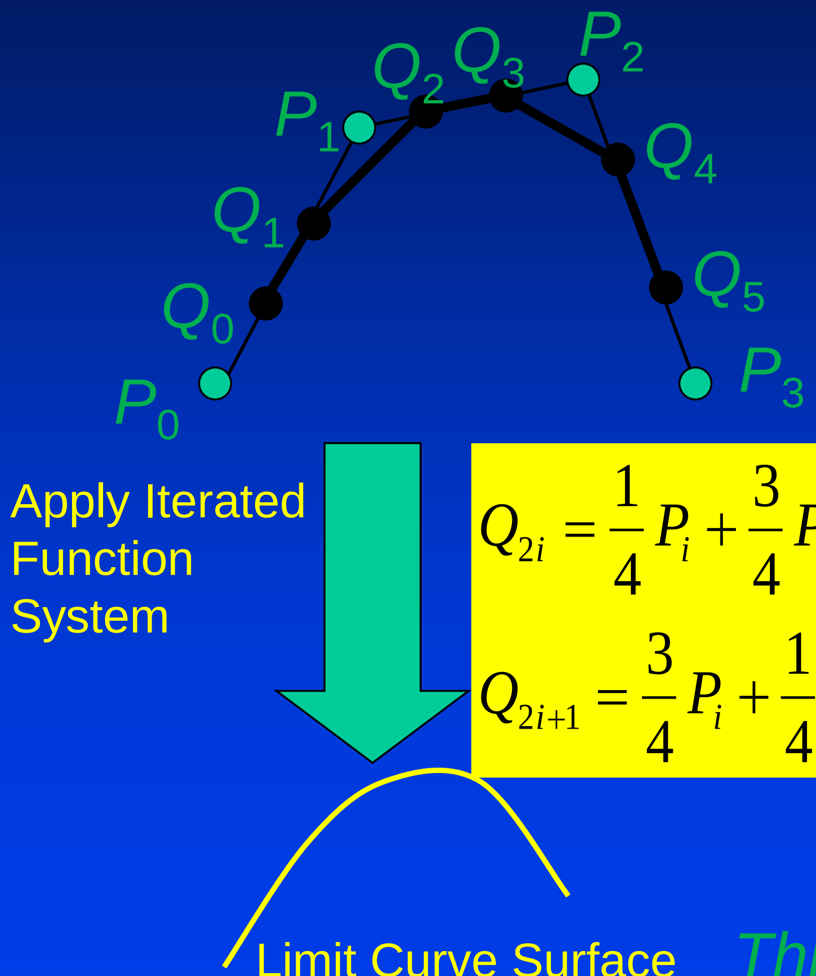
$$\mathbf{p}_{2i+1}^{k+1} = \frac{1}{4}\mathbf{p}_i^k + \frac{3}{4}\mathbf{p}_{i+1}^k$$

- Properties:
  – quadratic B-spline curve, C1 continuous, tangent to each edge at its mid-point

Chaikin's Algorithm



Chaiken's Algorithm – Another Example



$$Q_{2i} = \frac{1}{4}P_i + \frac{3}{4}P_{i+1}$$

$$Q_{2i+1} = \frac{3}{4}P_i + \frac{1}{4}P_{i+1}$$

Think Fractal!

$$Q_0 = \frac{1}{4}P_0 + \frac{3}{4}P_1$$

$$Q_1 = \frac{3}{4}P_0 + \frac{1}{4}P_1$$

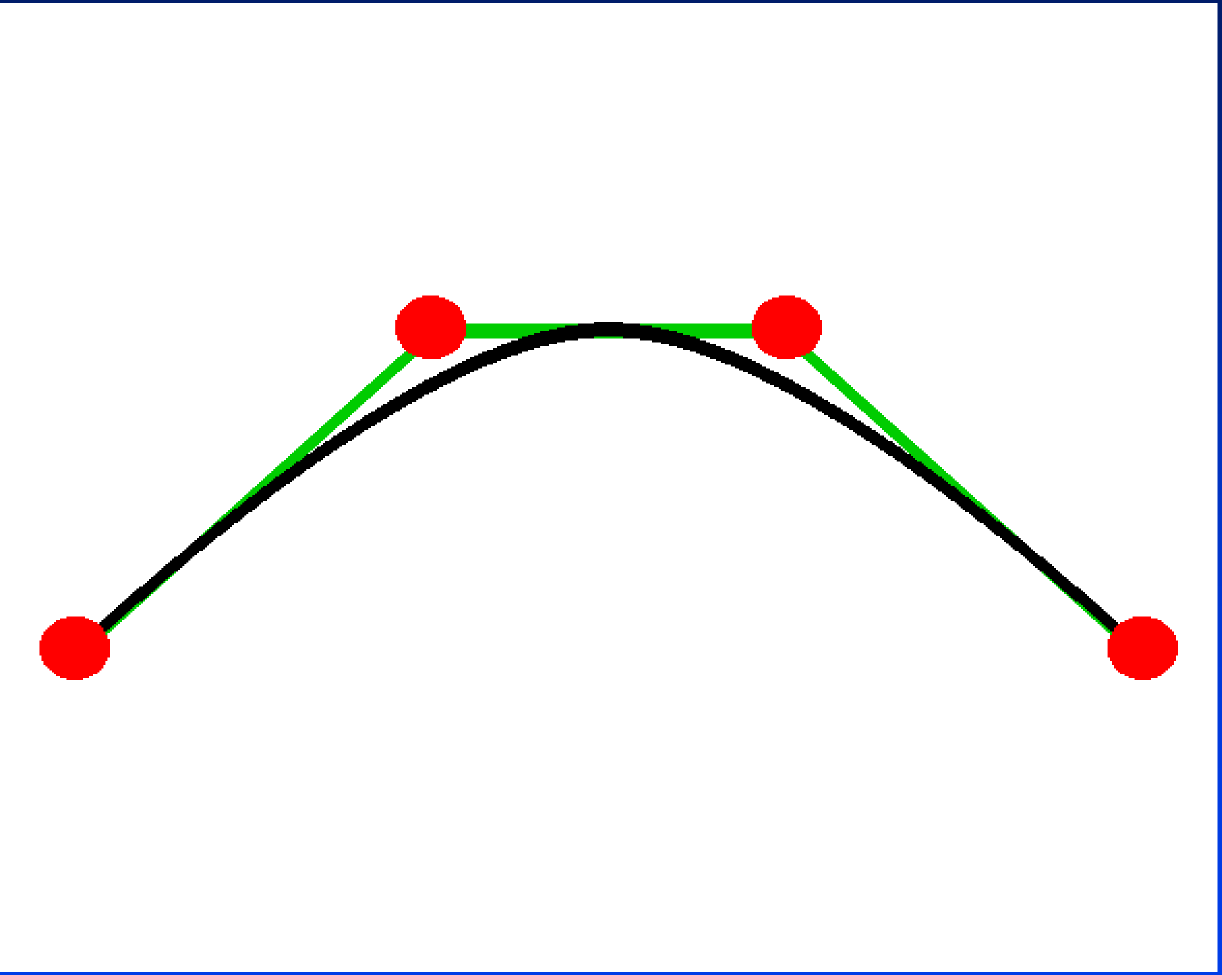
$$Q_2 = \frac{1}{4}P_1 + \frac{3}{4}P_2$$

$$Q_3 = \frac{3}{4}P_1 + \frac{1}{4}P_2$$

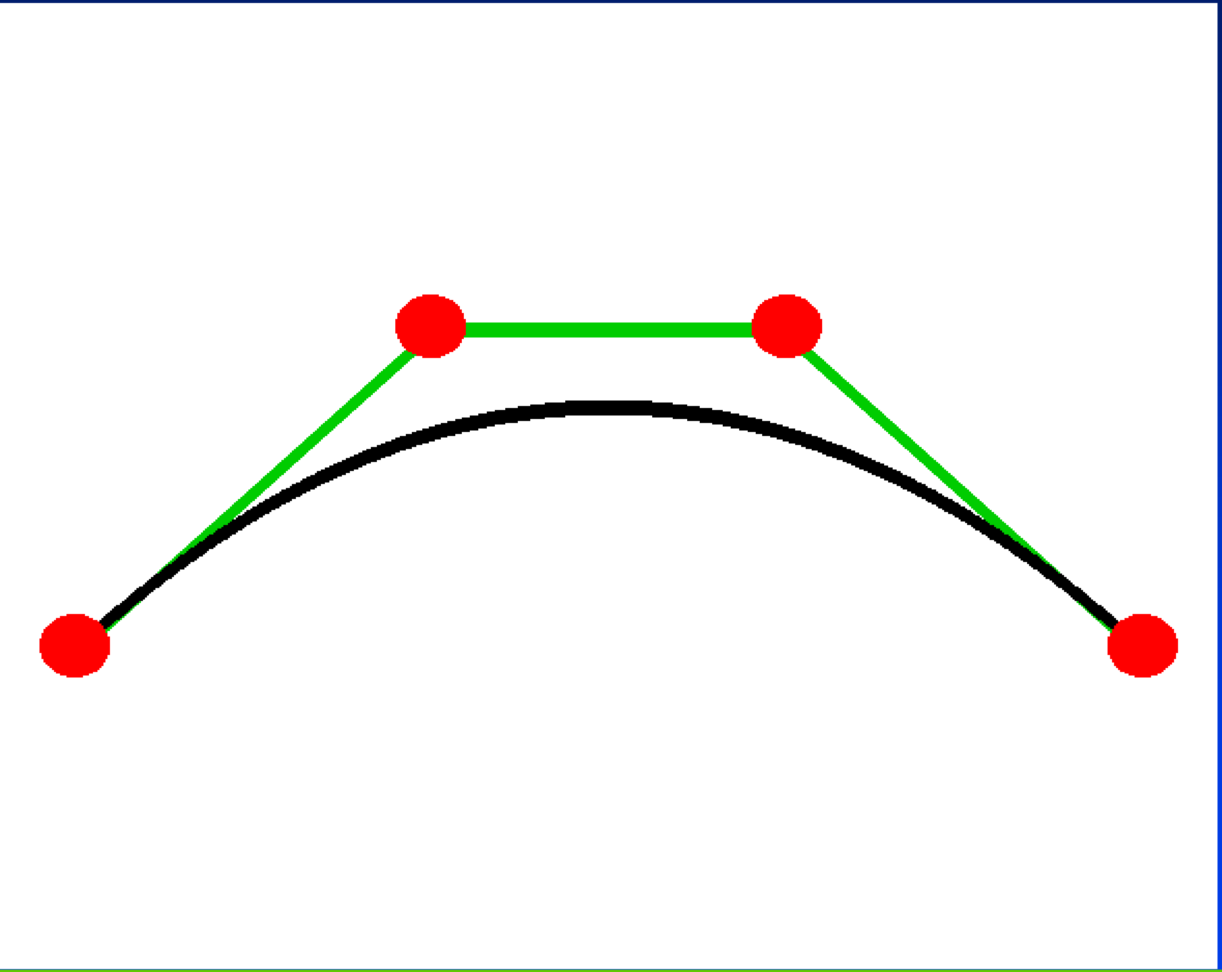
$$Q_4 = \frac{1}{4}P_2 + \frac{3}{4}P_3$$

$$Q_5 = \frac{3}{4}P_2 + \frac{1}{4}P_3$$

Quadratic Spline



Cubic Spline



Cubic Spline

- Subdivision rules

$$\mathbf{p}_{2i}^{k+1} = \frac{1}{2}\mathbf{p}_i^k + \frac{1}{2}\mathbf{p}_{i+1}^k$$

$$\mathbf{p}_{2i+1}^{k+1} = \frac{1}{4}(\frac{1}{2}\mathbf{p}_i^k + \frac{1}{2}\mathbf{p}_{i+2}^k) + \frac{3}{4}\mathbf{p}_{i+1}^k$$

- C2 cubic B-spline curve
- Corner-chopping
- No interpolation

Curve Interpolation

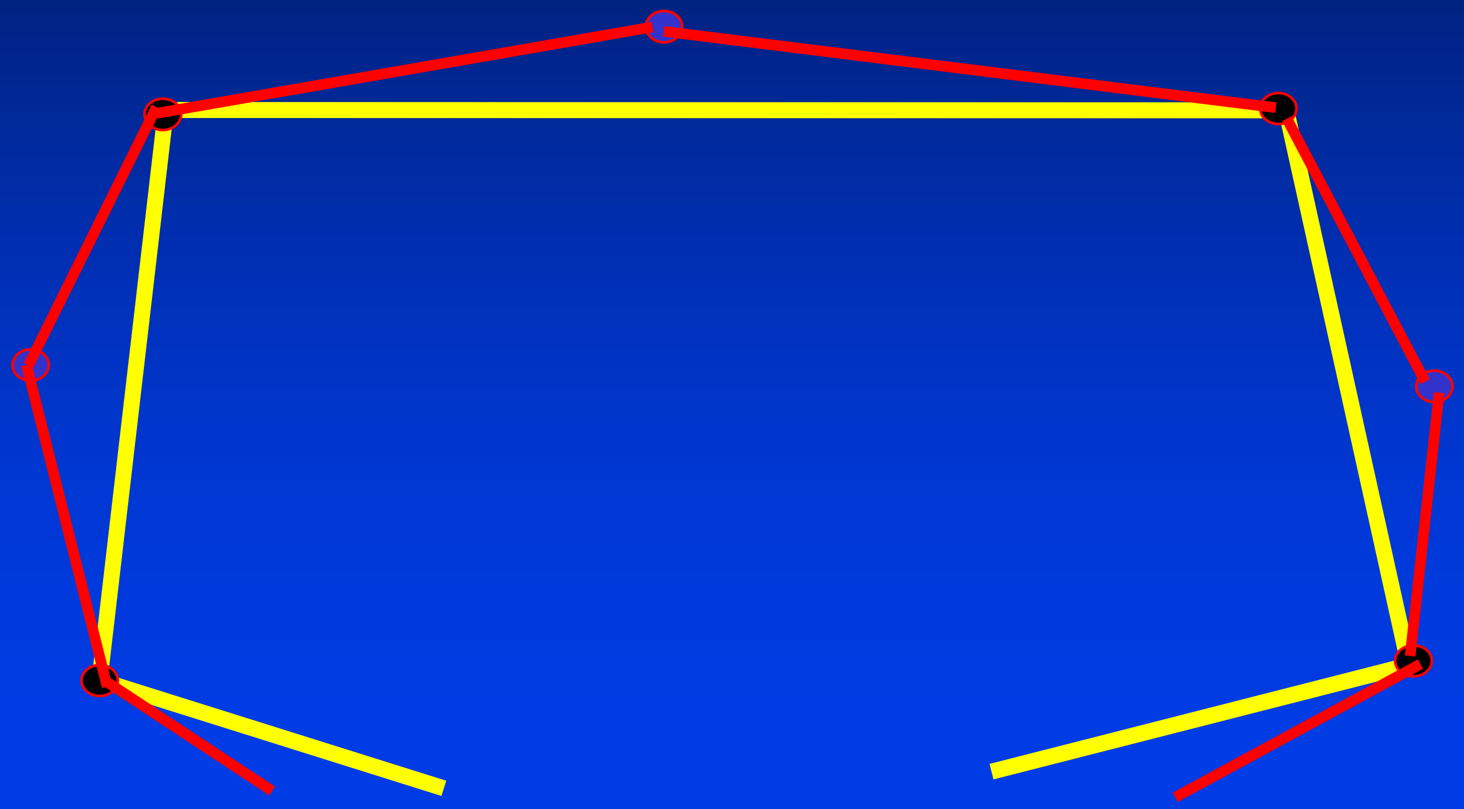
- Control points $\mathbf{p}_{-2}^{0}, \mathbf{p}_{-1}^{0}, \mathbf{p}_{0}^{0}, \cdots \mathbf{p}_{n+2}^{0}$
- Rules:

$$\mathbf{p}_{2i}^{k+1} = \mathbf{p}_{i}^{k}, -1 \le i \le 2^{k} n + 1$$

$$\mathbf{p}_{2i+1}^{k+1} = (\frac{1}{2} + w)(\mathbf{p}_{i}^{k} + \mathbf{p}_{i+1}^{k}) - w(\mathbf{p}_{i-1}^{k} + \mathbf{p}_{i+2}^{k}),$$
$$-1 \le i \le 2^{k} n$$

- At each stage, we keep all the OLD points and insert NEW points “in between” the OLD ones
- Interpolation!
- The behaviors and properties of the limit curve depend on the parameter w
- Generalize to SIX-point interpolatory scheme!

Curve Interpolation



Other Modeling Primitives

- Spline patches.
- Polygonal meshes.

Spline Patches

Advantages:

- High level control.
- Compact analytical representations.

Disadvantages:

- Difficult to maintain and manage inter-patch smoothness constraints.
- Expensive trimming needed to model features.
- Slow rendering for large models.

Polygonal Meshes

Advantages:

- Very general.
- Can describe very fine detail accurately.
- Direct hardware implementation.

Disadvantages:

- Heavy weight representation.
- A simplification algorithm is always needed.

Subdivision Schemes

Advantages:

- Arbitrary topology.
- Level of detail.
- Unified representation.

Disadvantages:

- Difficult for analysis of properties like smoothness and continuity.

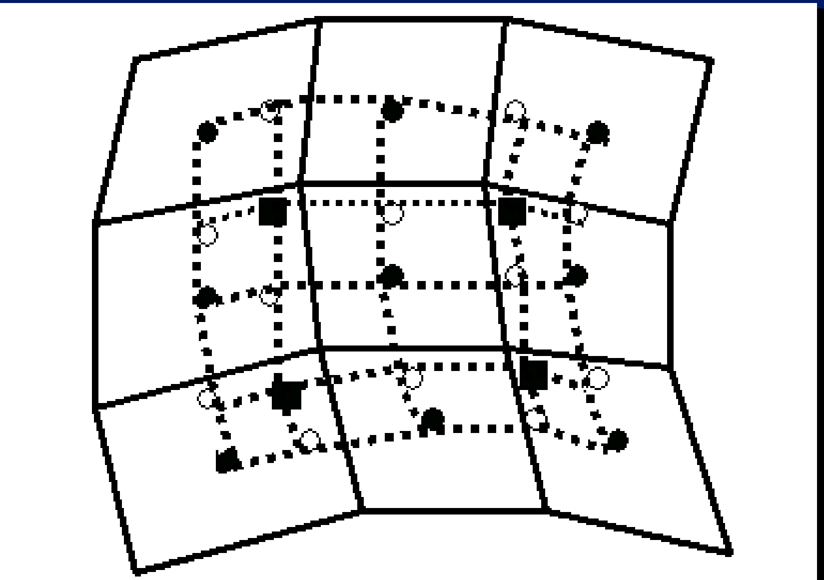
Uniform/Semi-uniform Schemes

- Catmull-Clark scheme
  - *Catmull and Clark, CAD 1978*
- Doo-Sabin scheme
  - *Doo and Sabin, CAD 1978*
- Loop scheme
  - *Loop, Master's Thesis, 1987*
- Butterfly scheme
  - *Dyn, Gregory and Levin, ACM TOG 1990.*
- Mid-edge scheme
  - *Habib and Warren, SIAM on Geometrrc Design 1995*
- Kobbelt scheme
  - *Kobbelt, Eurographics 1996*

Classification

- By Mesh type:
  - Triangular (Loop, Butterfly)
  - Quadrilateral (Catmull-Clark, Doo-Sabin, Mid-edge, Kobbelt)
- By Limit surface:
  - Approximating (Catmull-Clark, Loop, Doo-Sabin, Mid-edge)
  - Interpolating (Butterfly, Kobbelt)
- By Refinement rule:
  - Vertex insertion (Catmull-Clark, Loop, Butterfly, Kobbelt)
  - Corner cutting (Doo-Sabin, Mid-edge)

Catmull-Clark Scheme



- **Face point:**

  the average of all the points defining the old face.

- **Edge point:**

  the average of two old vertices and two new face points of the faces adjacent to the edge.

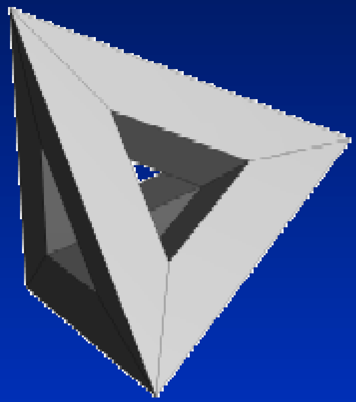
- **Vertex point:** $\left(F + 2E + (n-3)V\right)/n$

  F: the average of the new face points of all faces adjacent to the old vertex.

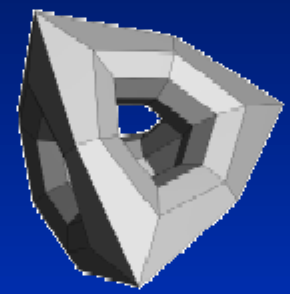
  E: the average of the midpoints of all adjacent edges.

  V: the old vertex.

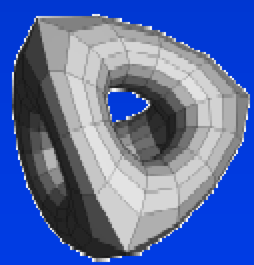
Catmull-Clark Scheme



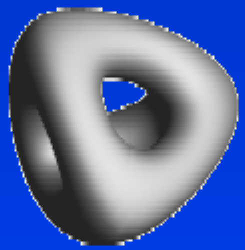
Initial mesh



Step 1



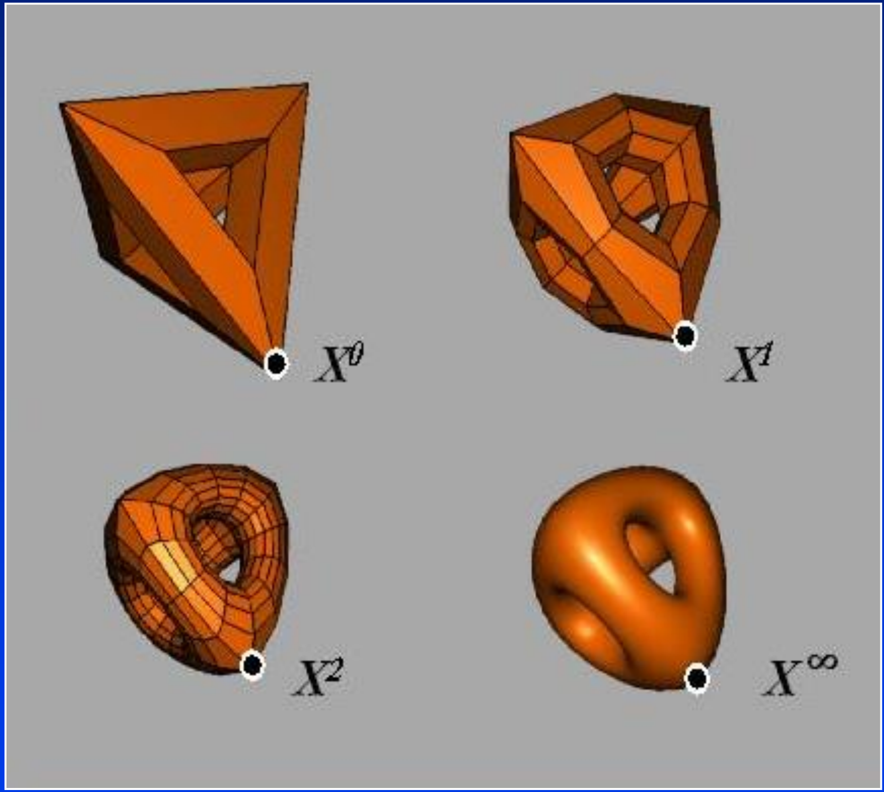
Step 2



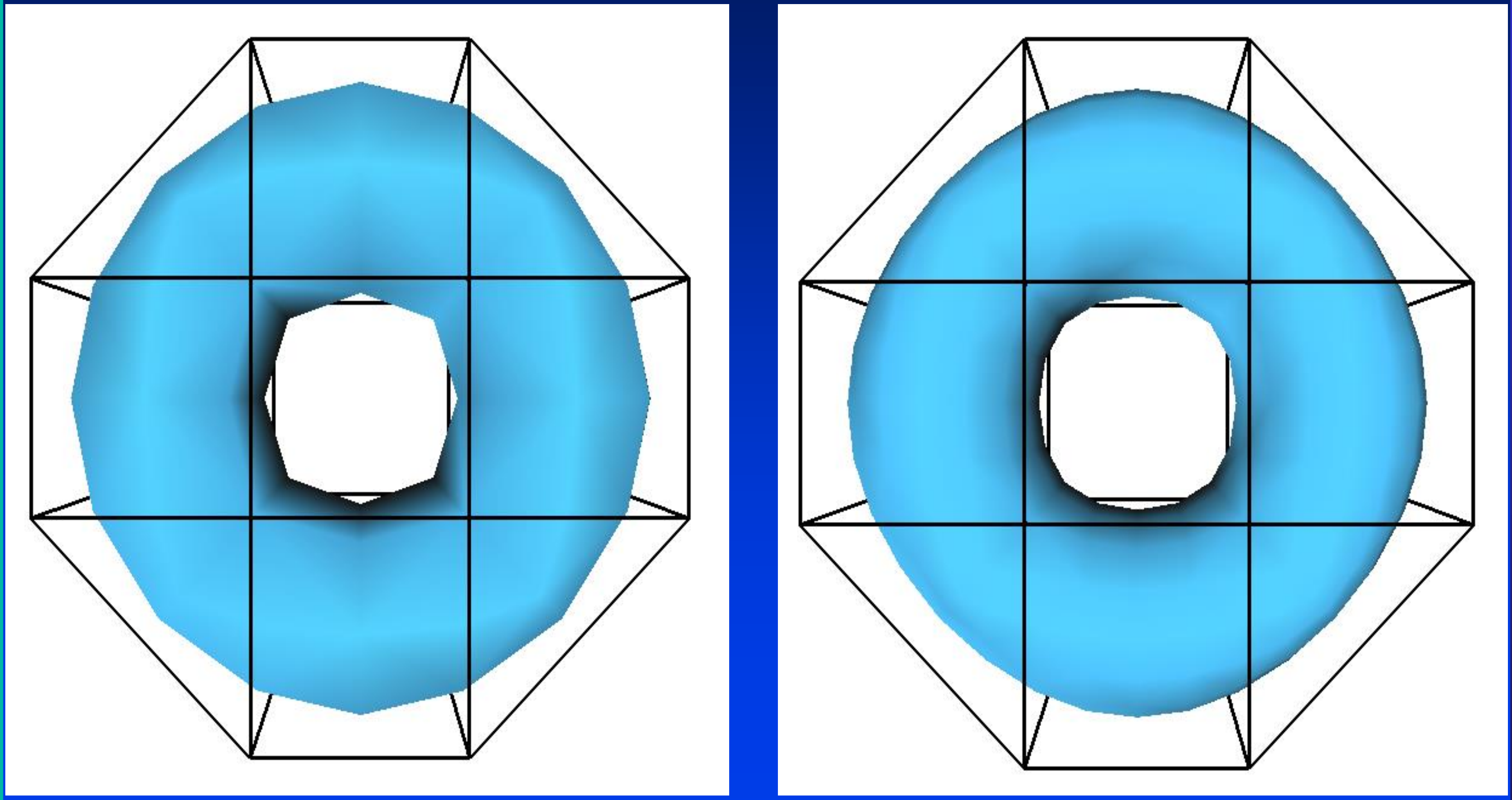
Limit surface

Modified Catmull-Clark

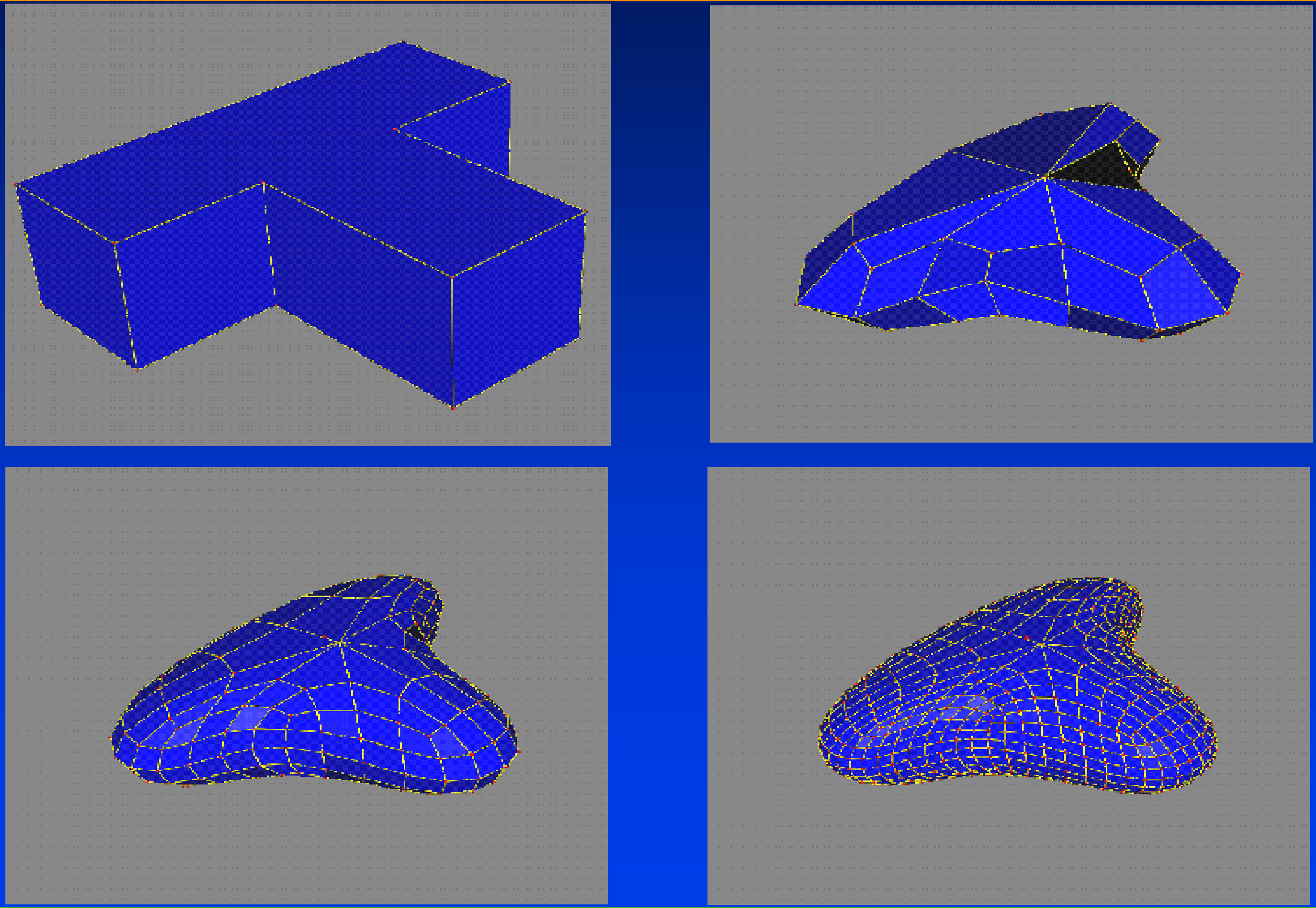
- Extend Cubic B-splines
  - Easier to implement with existing software
- Quadrilaterals are often better at capturing symmetry
  - Like human body parts
- Quads are convenient for cloth dynamics



Catmull-Clark Subdivision



Catmull-Clark Subdivision



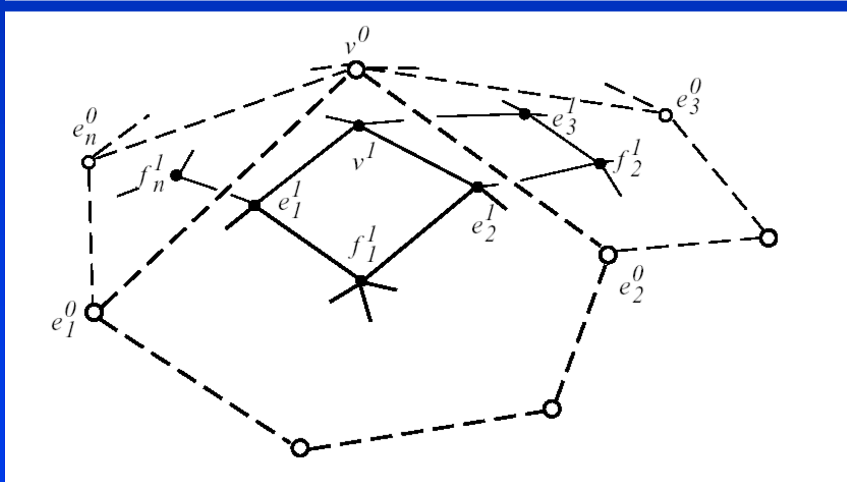
Catmull-Clark Subdivision

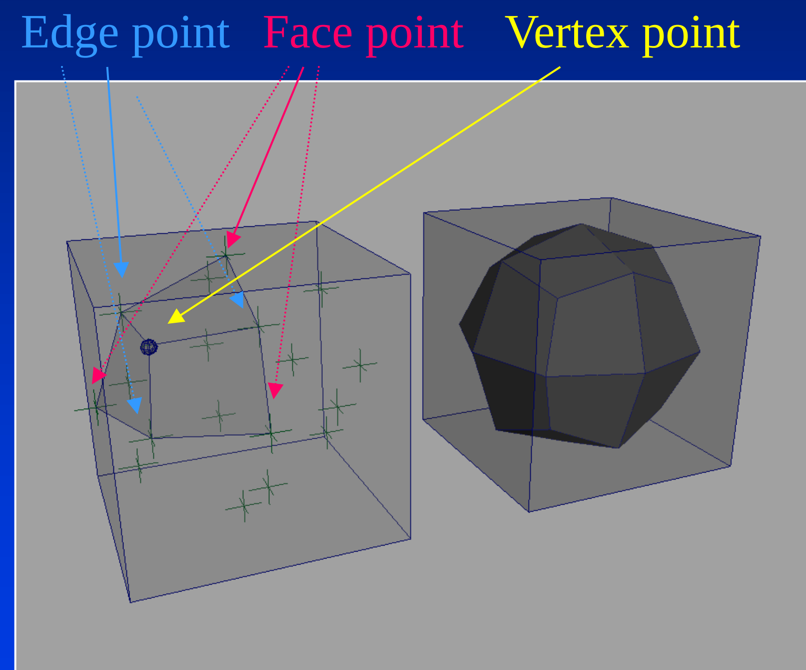
(1)

$$e_j^{i+1} = \frac{v^i + e_j^i + f_{j-1}^{i+1} + f_j^{i+1}}{4},$$

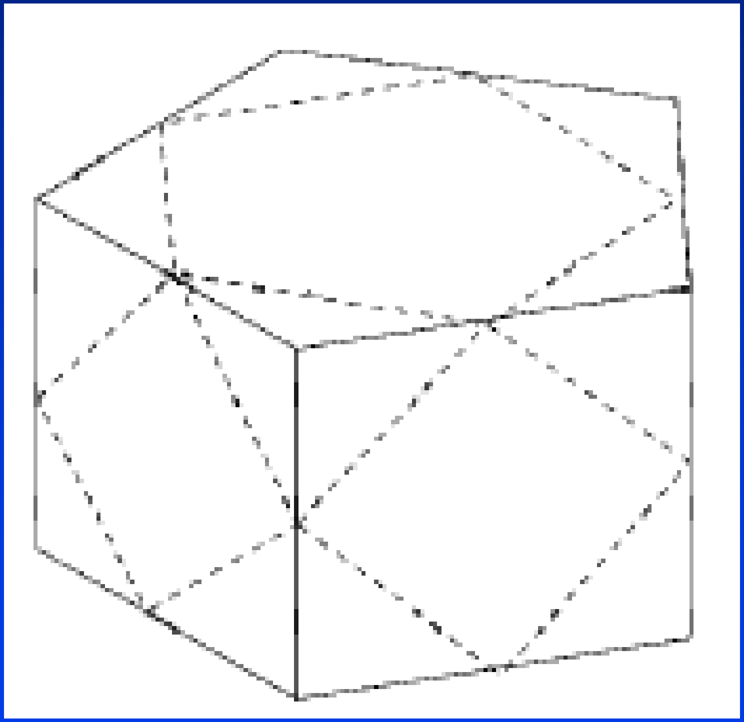
(2)

$$v^{i+1} = \frac{n-2}{n} v^i + \frac{1}{n^2} \sum_j e_j^i + \frac{1}{n^2} \sum_j f_j^{i+1}$$

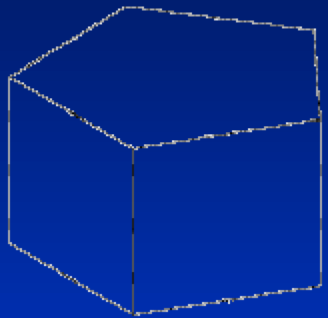




Mid-edge Scheme

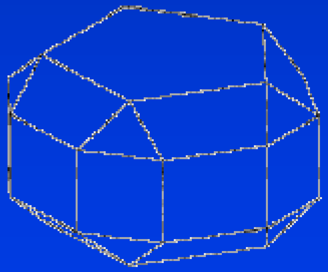


Mid-edge Scheme

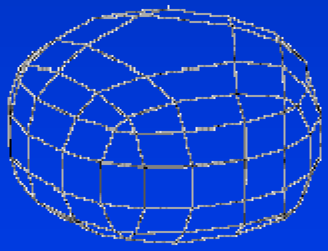


(a)

(b)



(c)



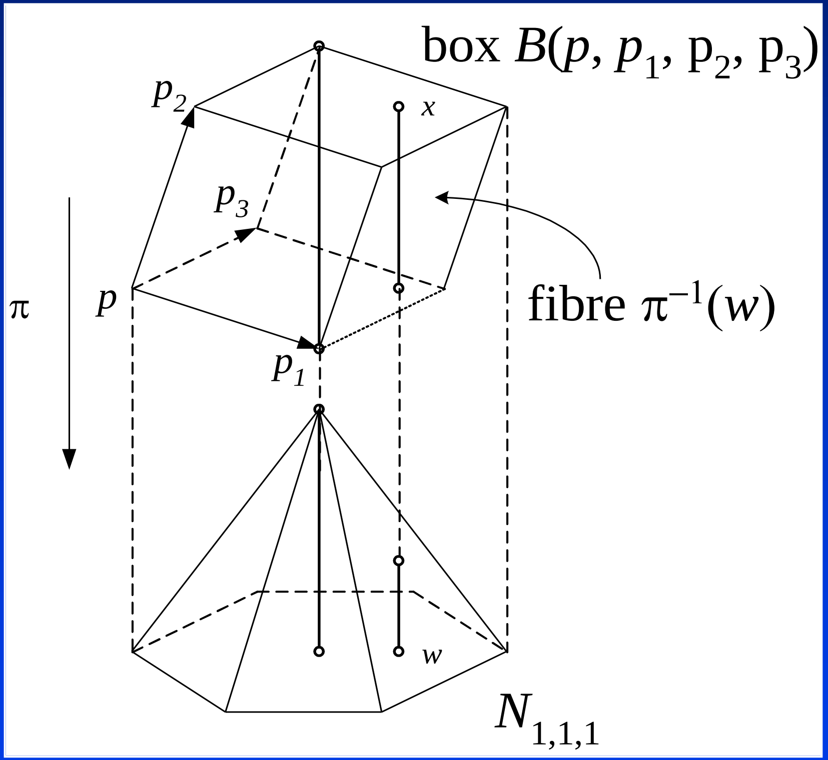
(d)

Loop Scheme

- Box splines
  - A projection of 6D box onto 2D
  - A quartic polynomial basis function
  - Triangular domain
- Works on triangular meshes
- Is an approximating scheme
- Non-tensor-product splines
- Loop scheme results from a generalization of box splines to arbitrary topology
- Guaranteed to be smooth everywhere except at extraordinary vertices

Box Spline Overview

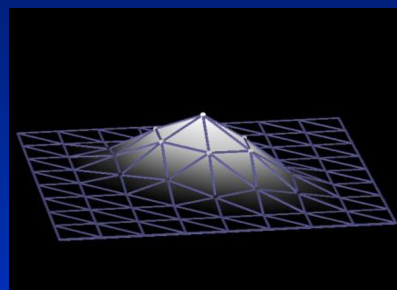
- Based on 2D Box Spline
  - Defined by projection of hypercube (in 6D) into 2D.
  - Satisfies many properties that B-spline has.
    - Recursive definition
    - Partition of unity
    - Truncated power
  - Natural splitting of a cube into sub-cubes provides the subdivision rule.



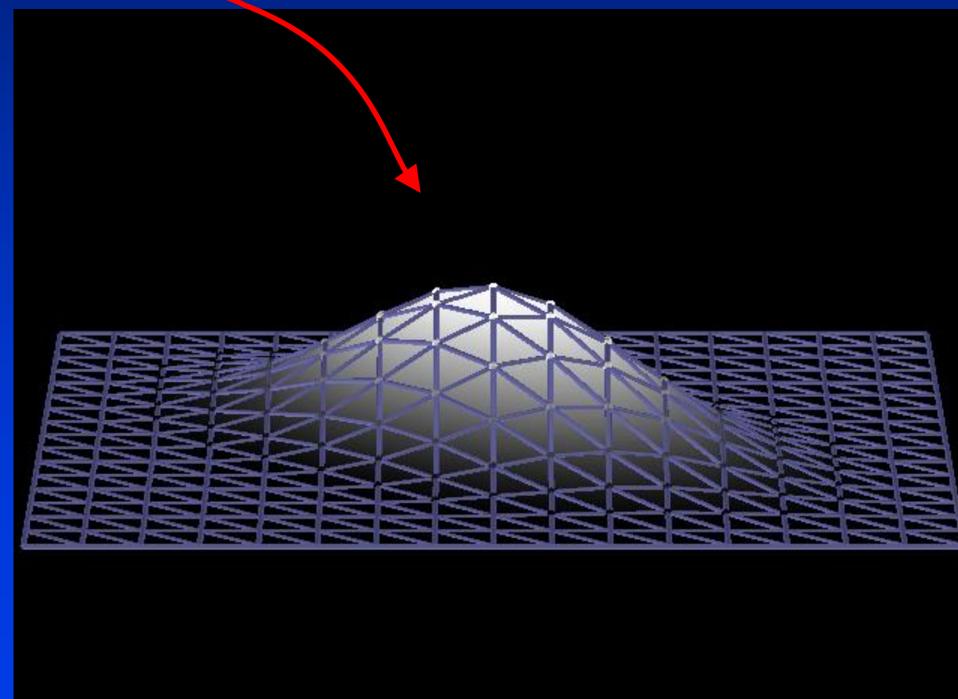
Basis Functions for Loop’s Scheme

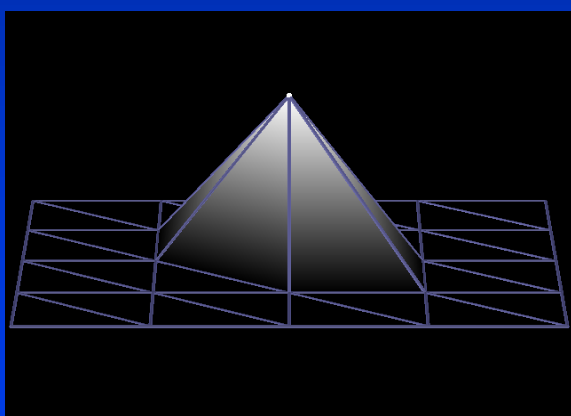
- Basis Function - Evaluation

Successive Subdivision



The Limit → $N_{2,2,2}$ Basis



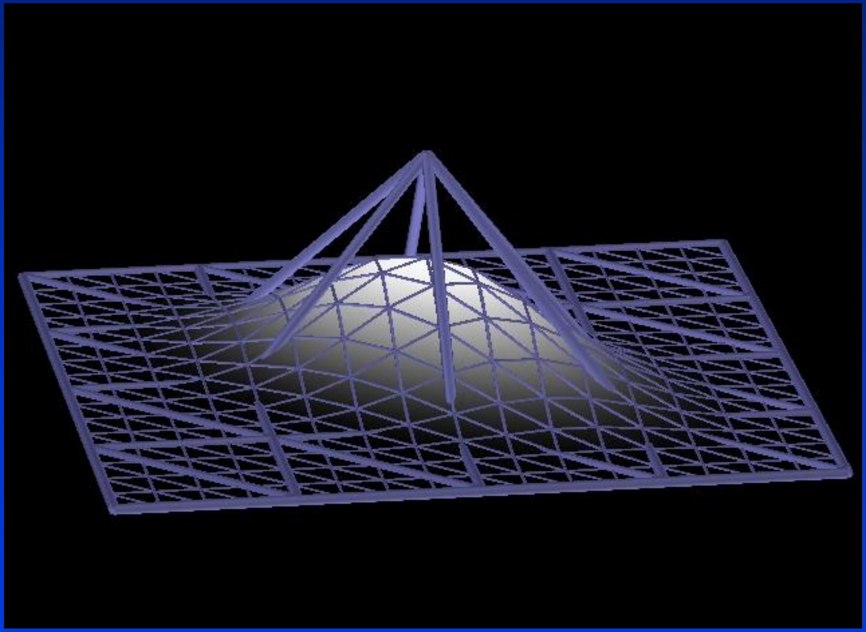


Assign unit weight to center, zero otherwise, over Z^2 lattice

Loop’s Scheme Properties

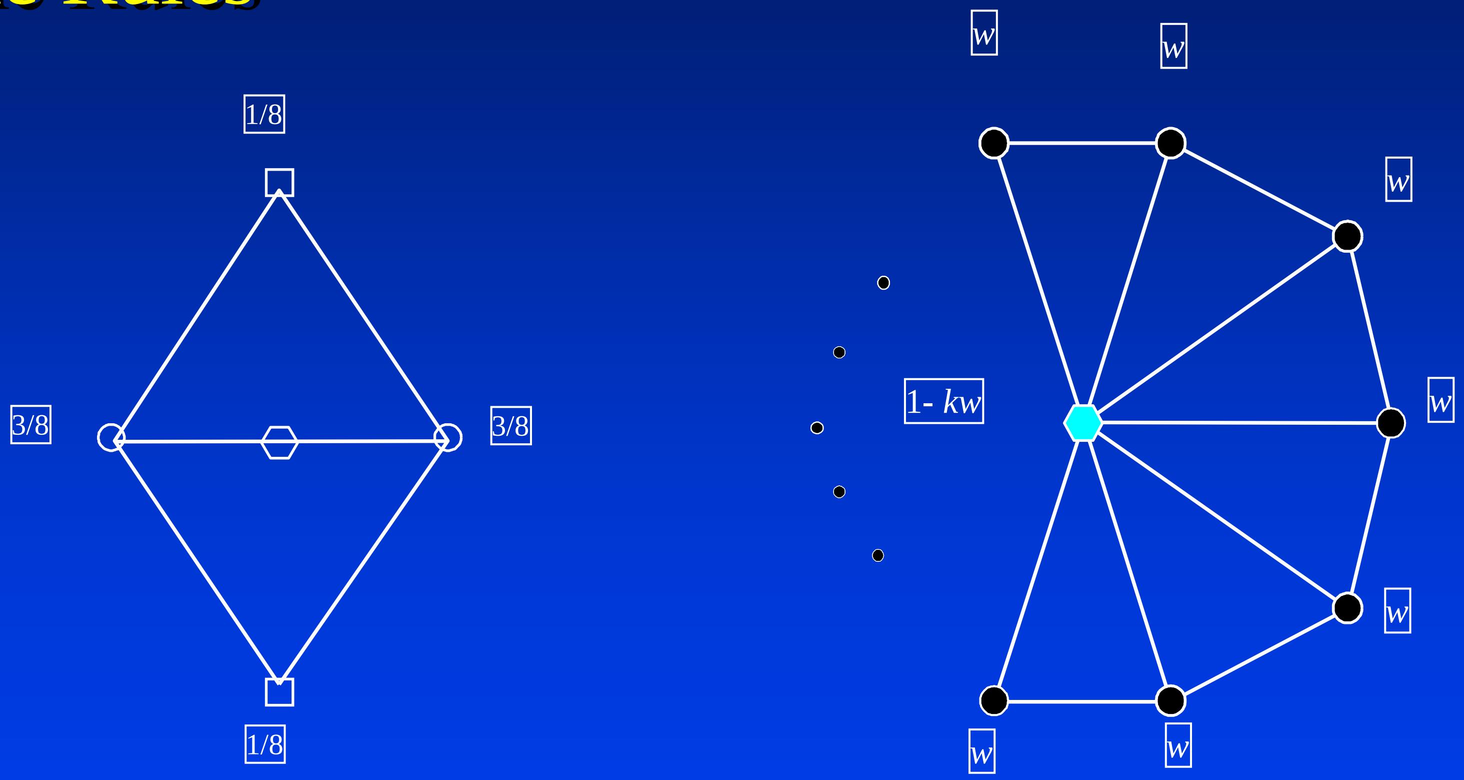
- Basis Function – Properties

1. Support → 2 neighbors from the center
2. C^4 continuity within the support
3. Piecewise polynomial
4. $N_{2,2,2}(\bullet - j)$, $j \in Z^2$ form a partition of unity i.e. $\Sigma N(x - j) = 1$

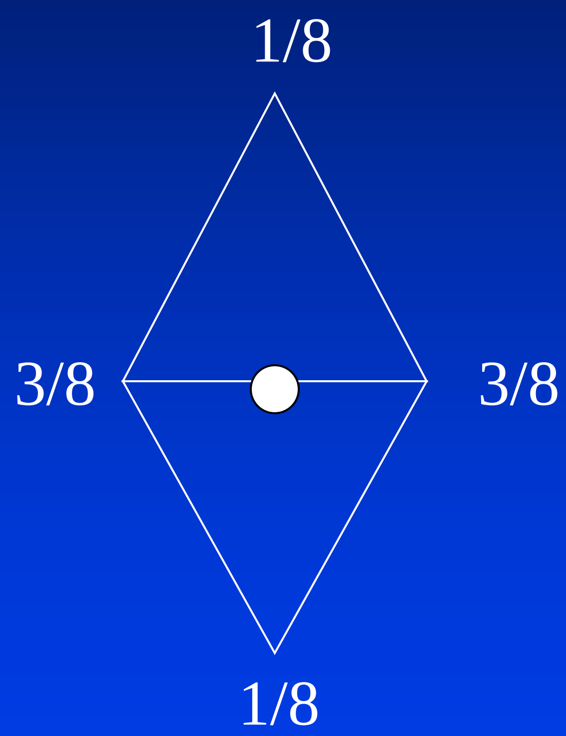


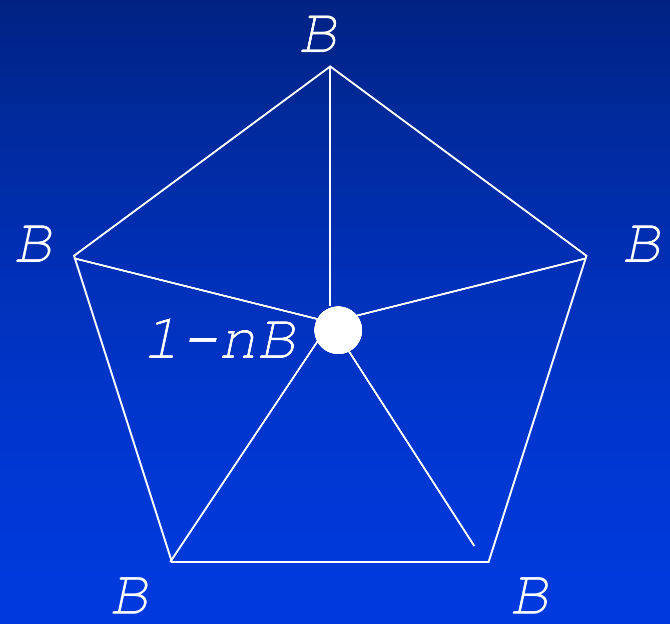
Loop’s Scheme Rules

- The Rules



Loop Scheme Rules

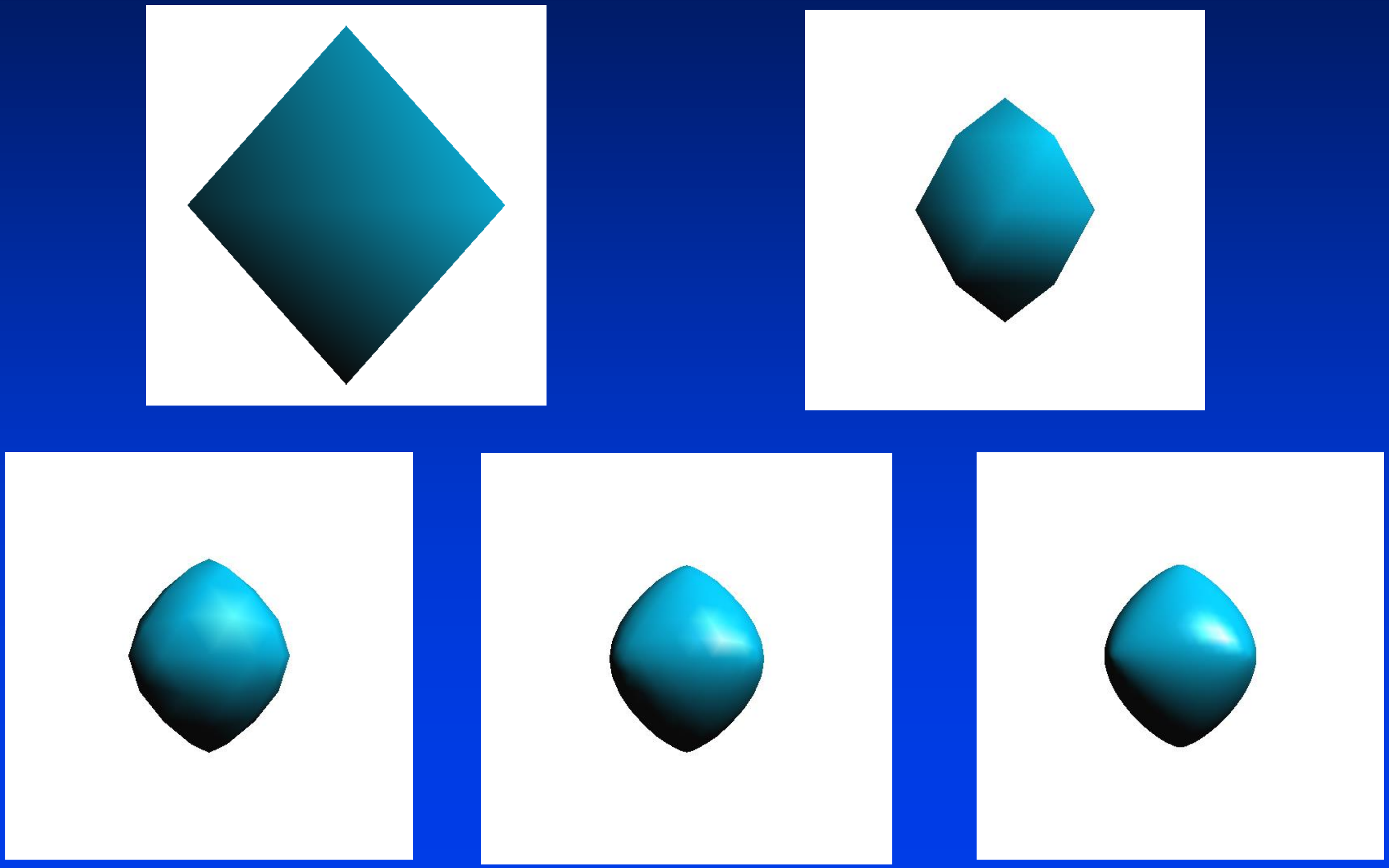




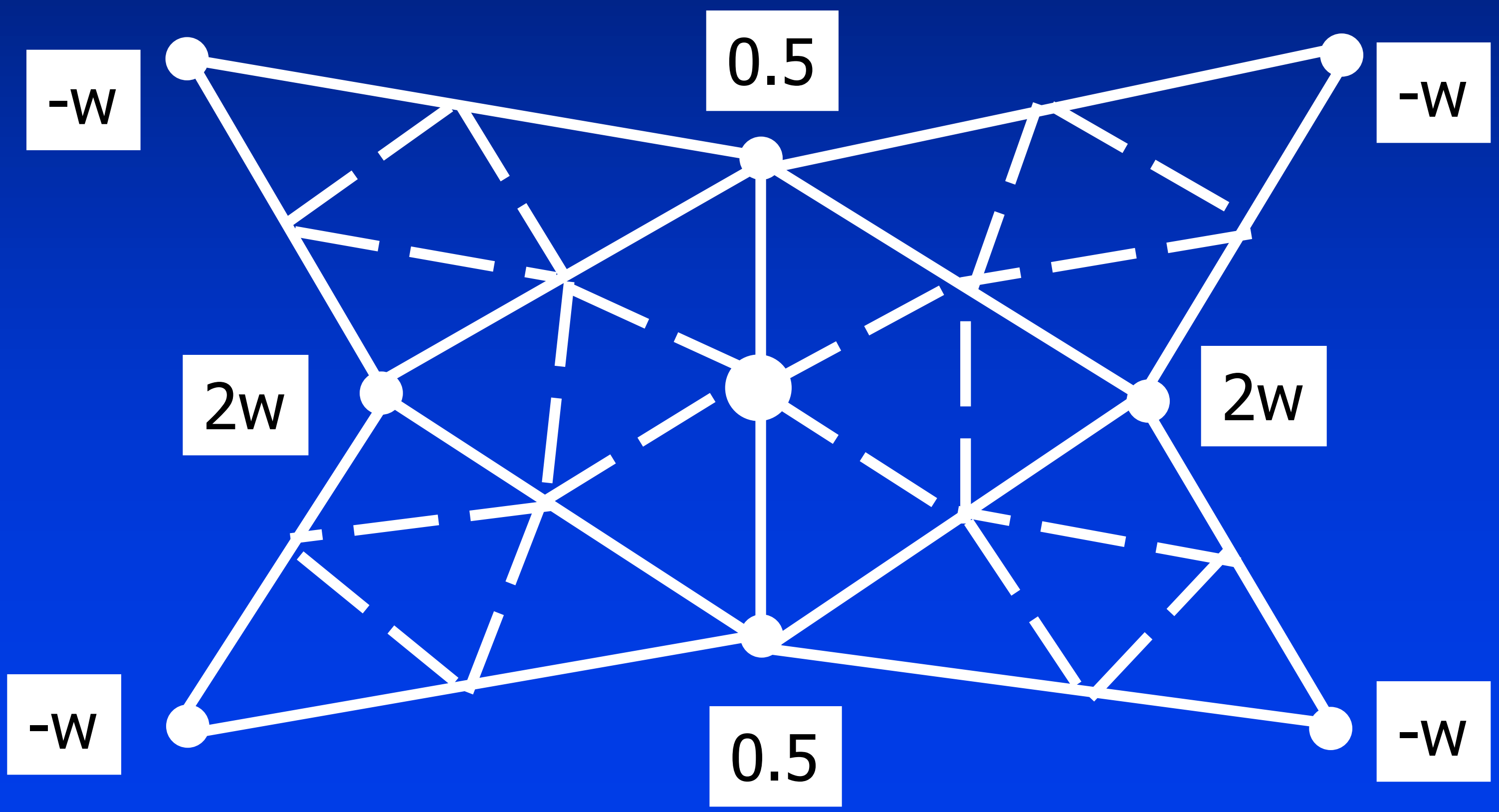
$B = 3/8k$, for $n>3$

$B = 3/16$, for $n=3$

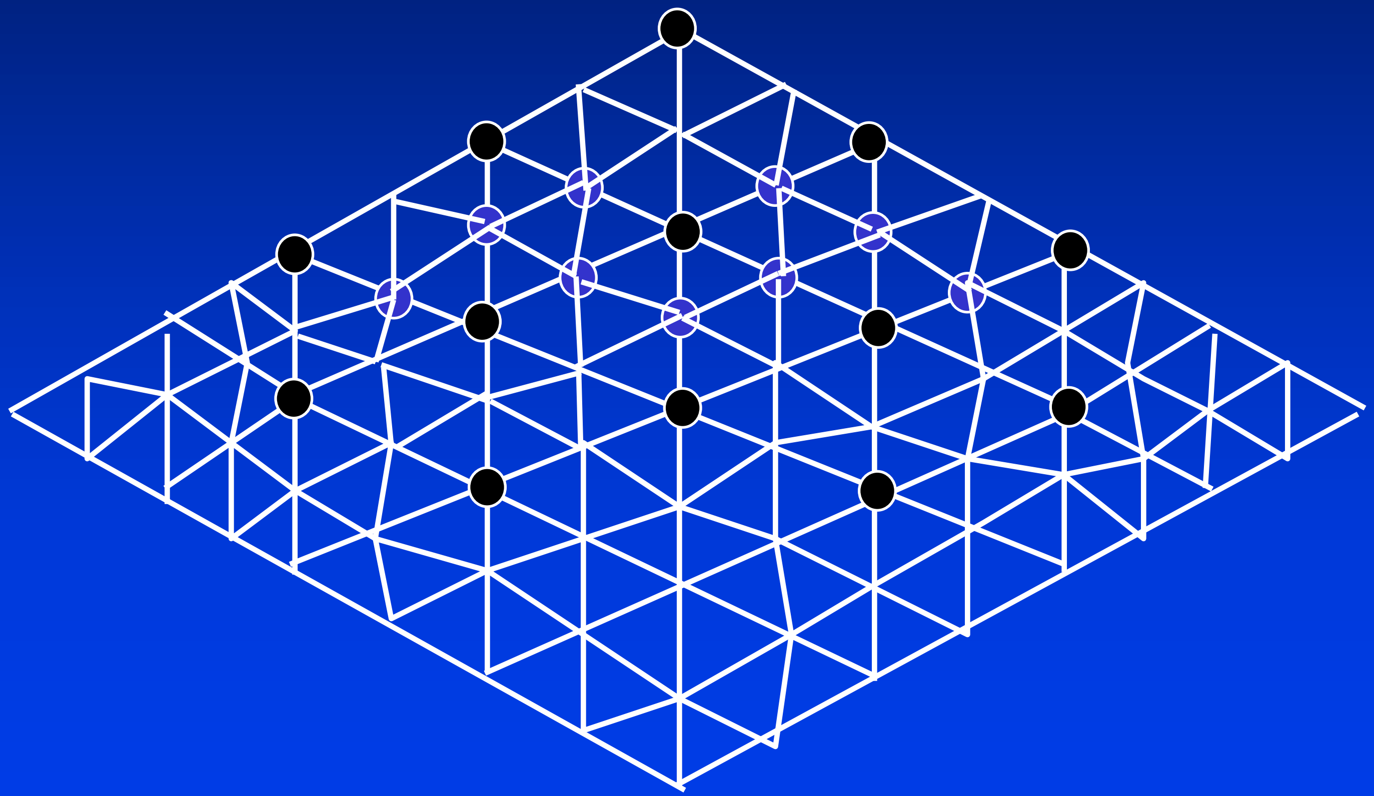
Loop Scheme Example



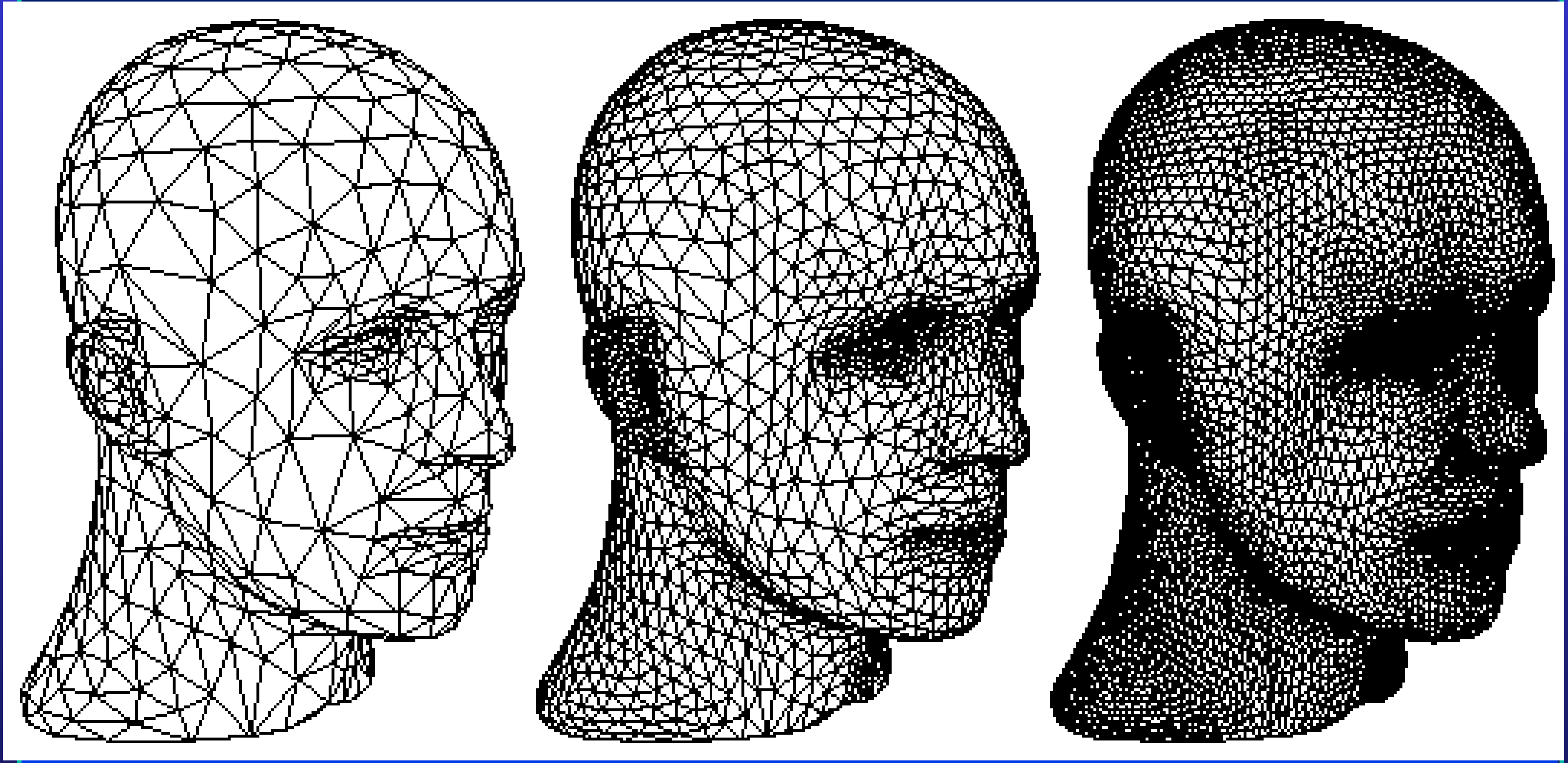
Butterfly Subdivision



Butterfly Scheme

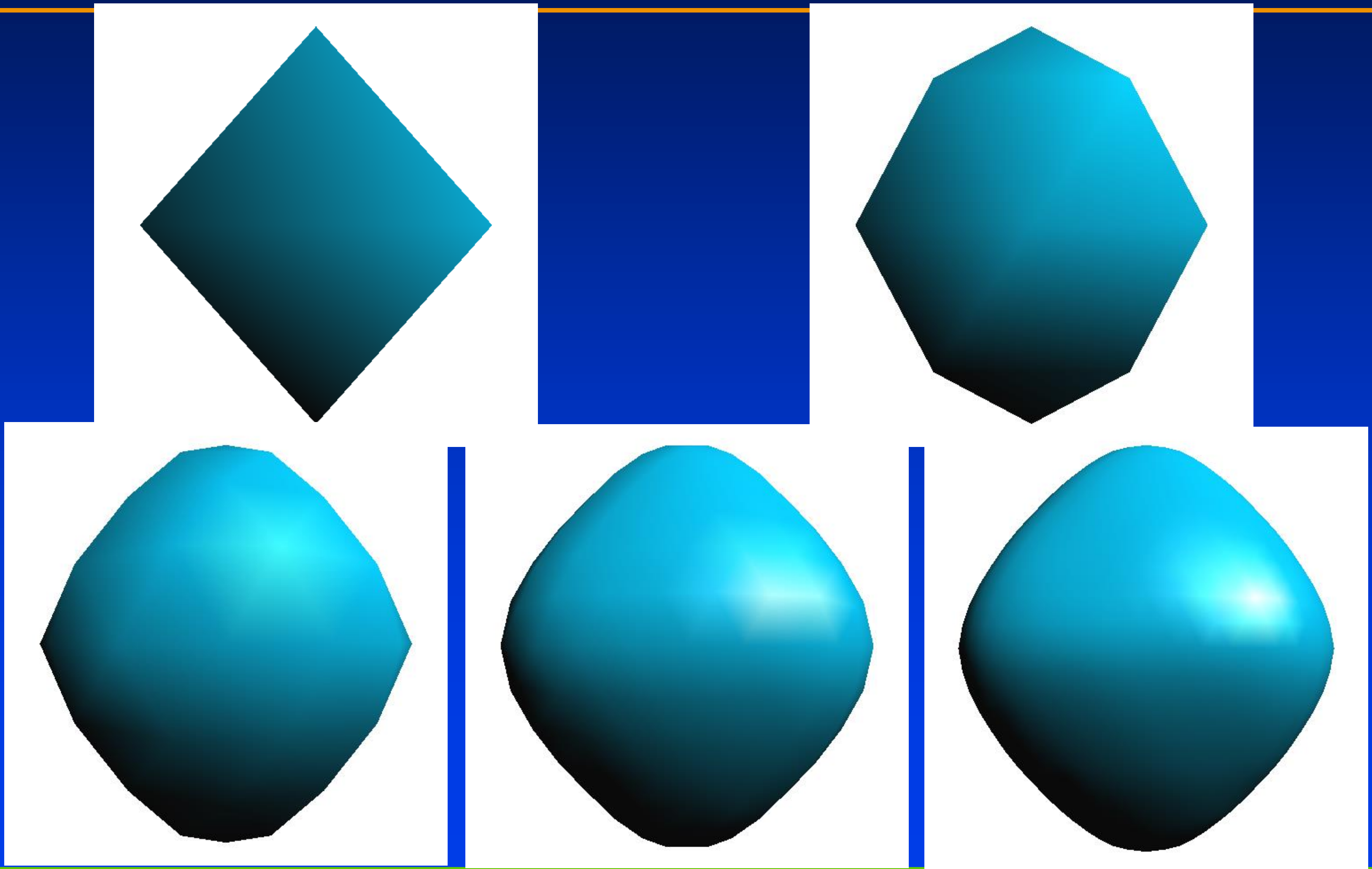


Modified Butterfly Scheme

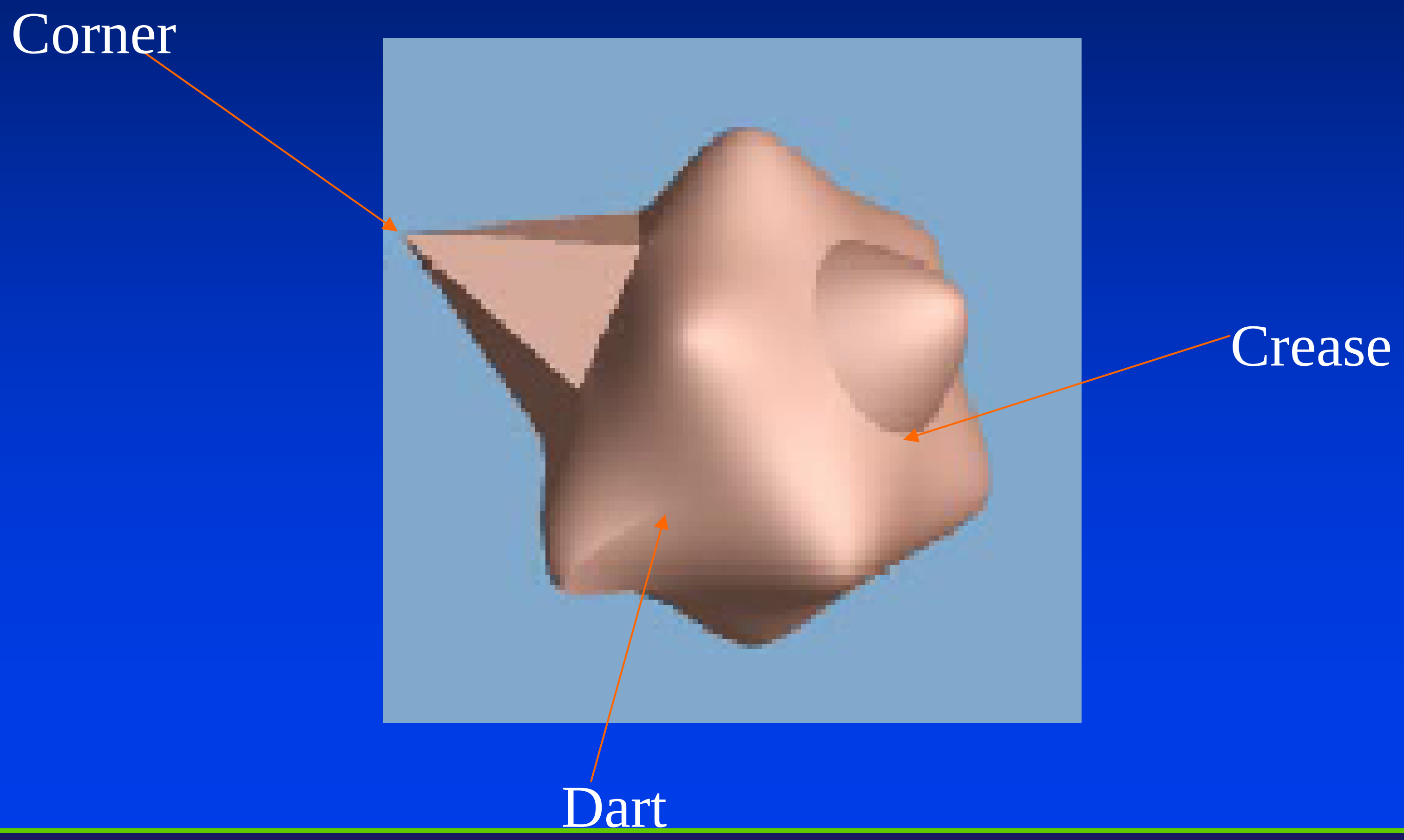


Initial mesh | One refinement step | Two refinement steps

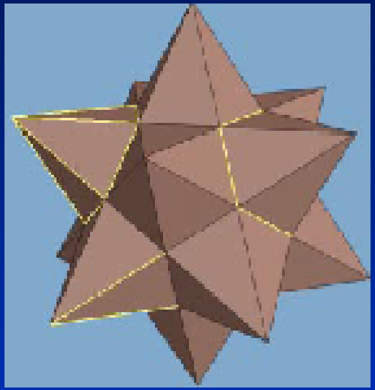
Modified Butterfly Example



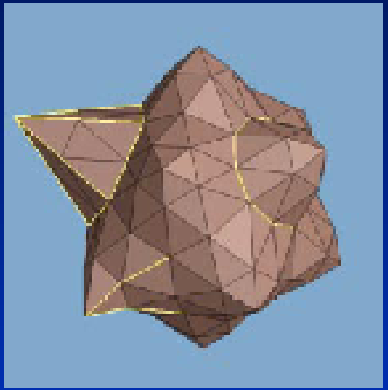
Modeling Sharp Features



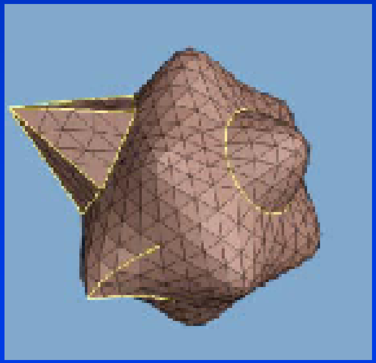
Piecewise Smooth Subdivision



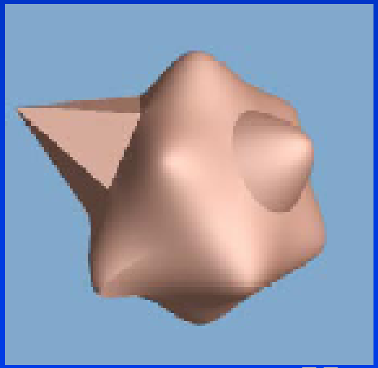
(a)



(b)



(c)

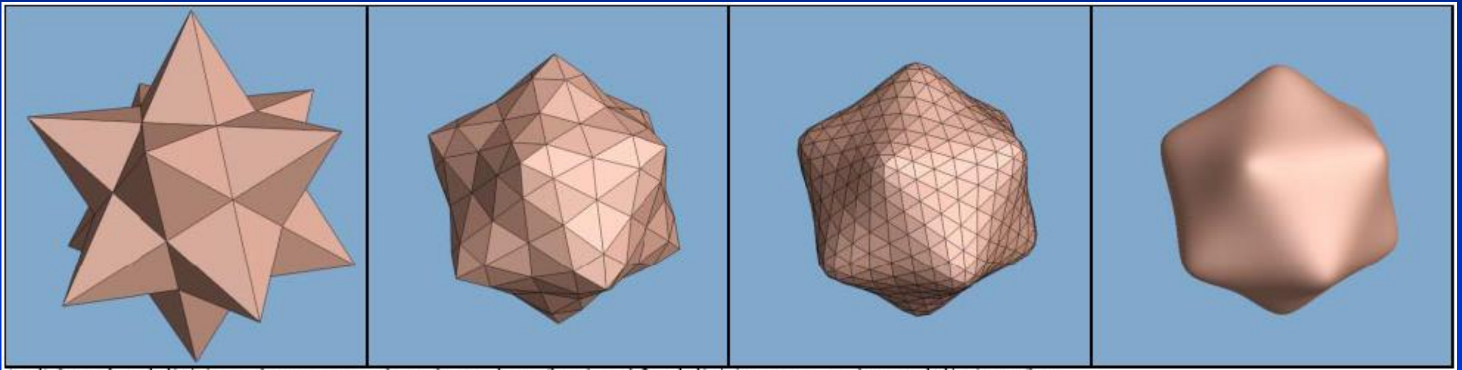


(d)

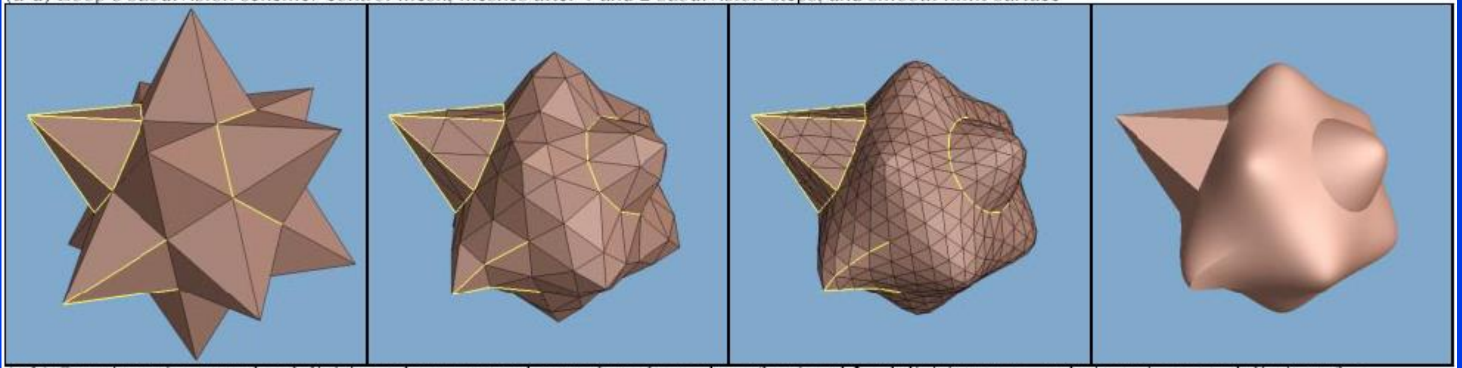
Hoppe et al. Siggraph 94

Piecewise Smooth Surface

- Piecewise C^1-continuous extension [Hoppe 94]
  - Extension of the Loop's scheme.



(a-d) Loop's subdivision scheme: control mesh, meshes after 1 and 2 subdivision steps, and smooth limit surface

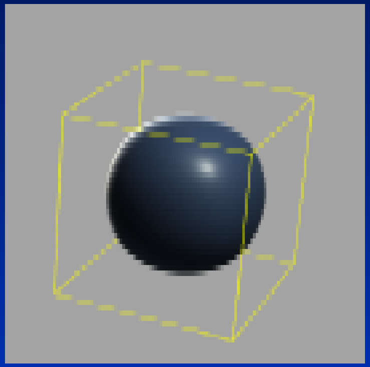


(e-h) Our piecewise smooth subdivision scheme: tagged control mesh, meshes after 1 and 2 subdivision steps, and piecewise smooth limit surface

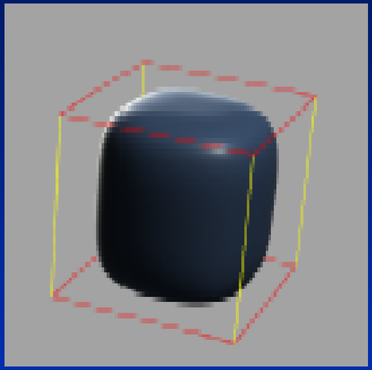
Non-uniform Subdivision Schemes

- Piecewise smooth subdivision schemes
  - *Hoppe et al. Siggraph 94*
- Hybrid scheme
  - *et al. Siggraph 98*
- NURSS scheme
  - *Sederburg et al. Siggraph 98*
- Combined scheme
  - *Levin Siggraph 99*
- Edge and vertex insertion scheme
  - *Habib et al. CAGD 99*

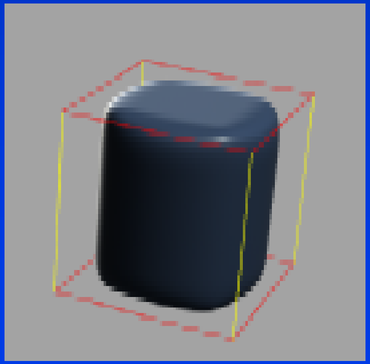
Hybrid Subdivision Scheme



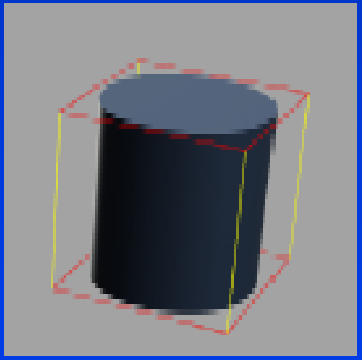
(a)



(b)



(c)



(d)

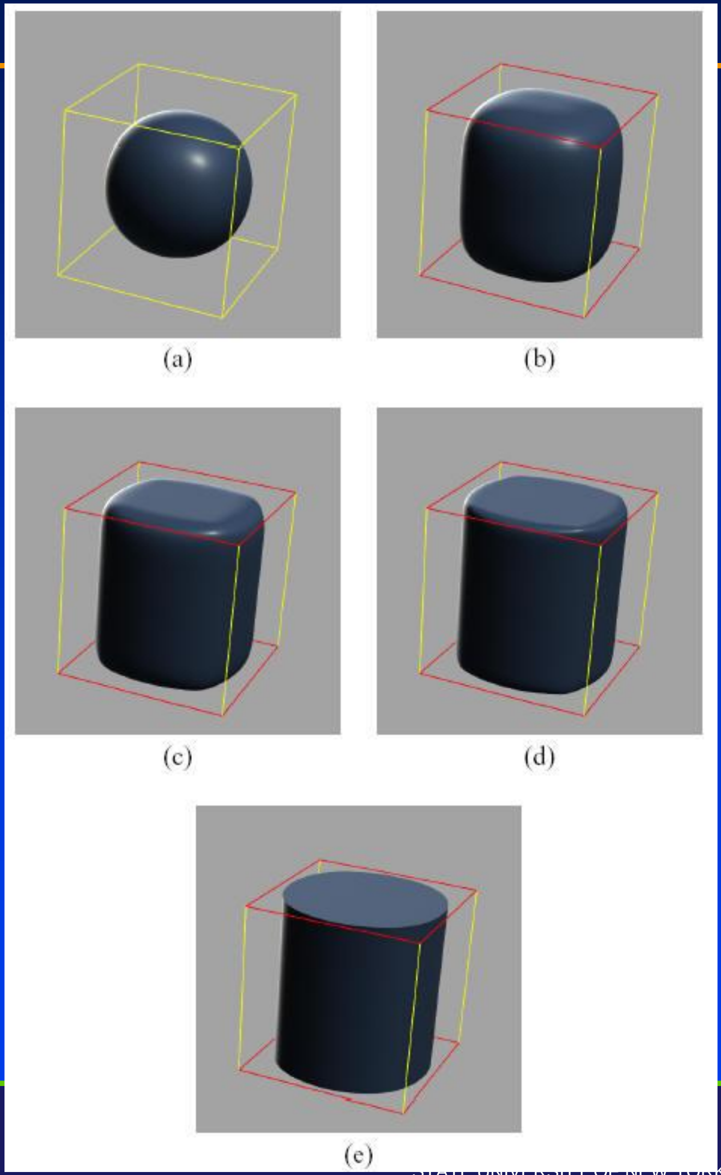
DeRose et al. Siggraph 98

Sharp Edges

1. Tag Edges as "**sharp**" or "not-sharp"
   - n = 0 – "**not sharp"**
   - n > 0 – sharp

During Subdivision,

2. if an edge is "sharp", use sharp subdivision rules. Newly created edges, are assigned a sharpness of n-1.
3. If an edge is "not-sharp", use normal smooth subdivision rules.


(a) (b) (c) (d) (e)

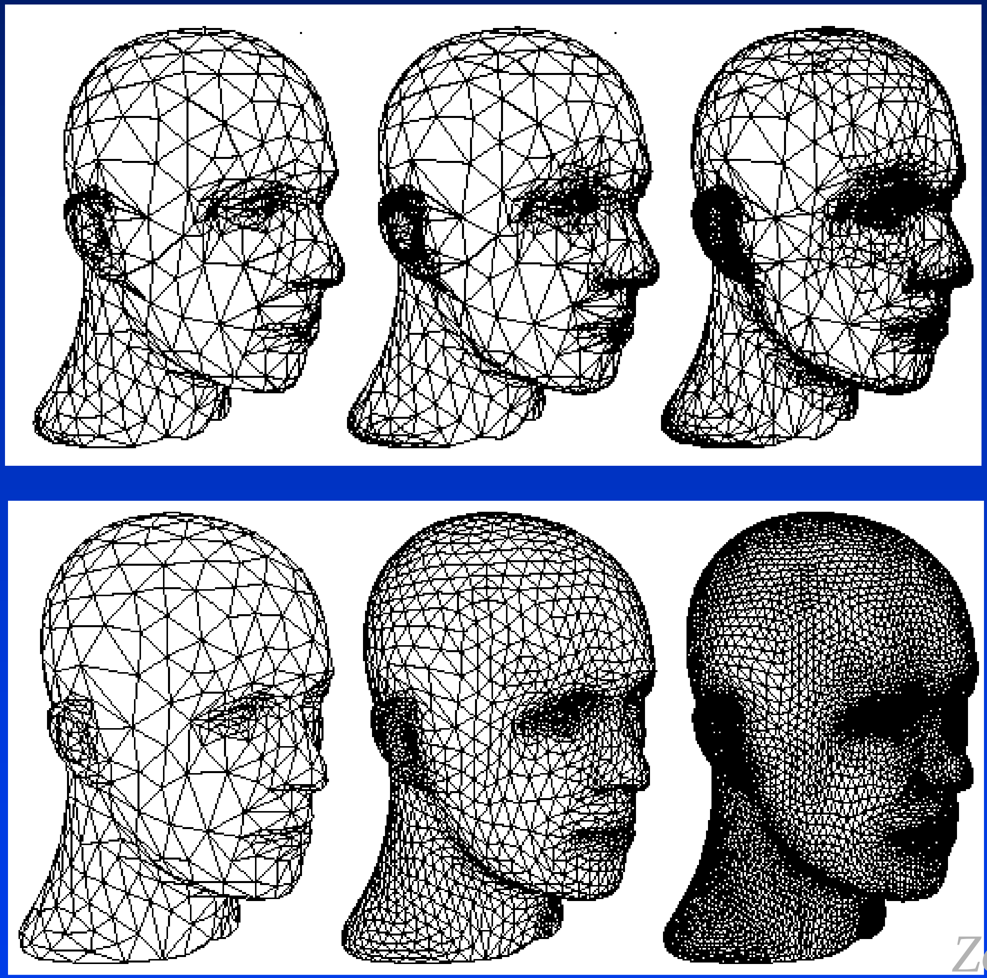
IDEA: Edges with a sharpness of "n" do not get subdivided smoothly for "n" iterations of the algorithm.

Non-Integer Sharpness

- Density of newly generated mesh increases rapidly.
- In practice, 2 or 3 iterations of subdivision is sufficient.
- Need better “control”.

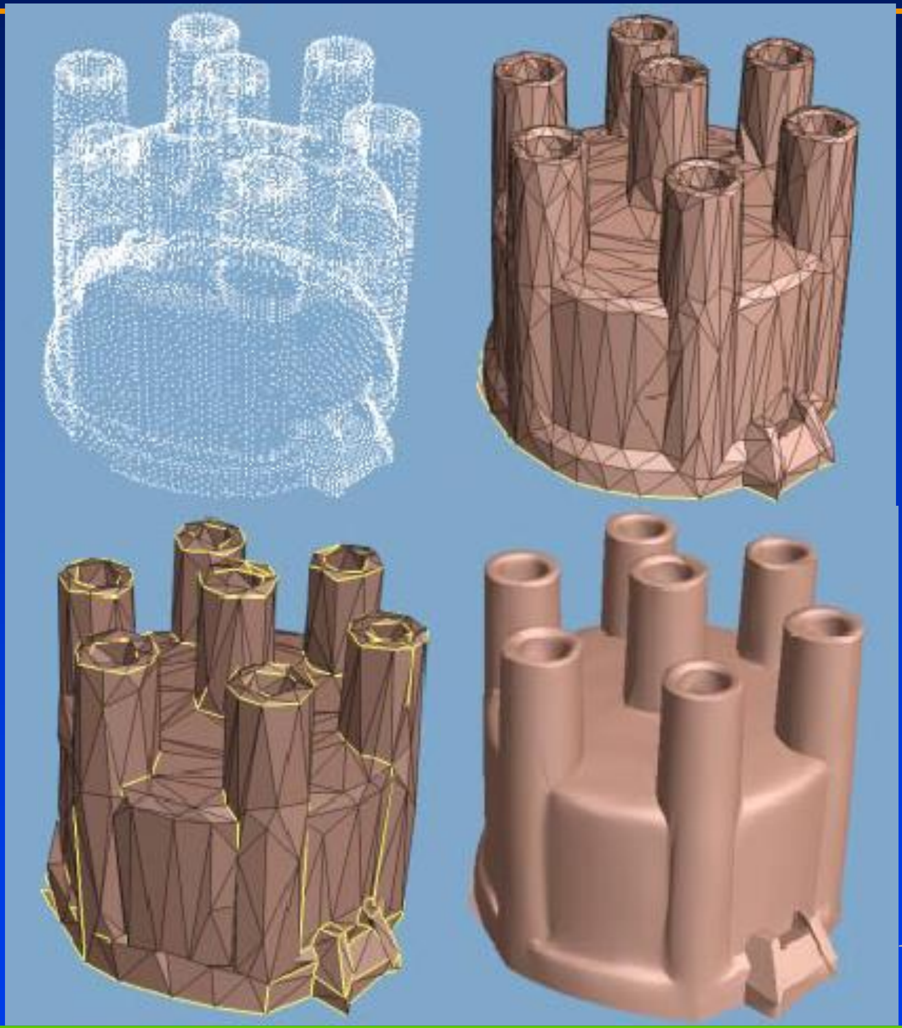
IDEA: Interpolate between smooth and sharp rules for non-integer sharpness values of n.

Hierarchical Editing



Zorin et al. Siggraph 97

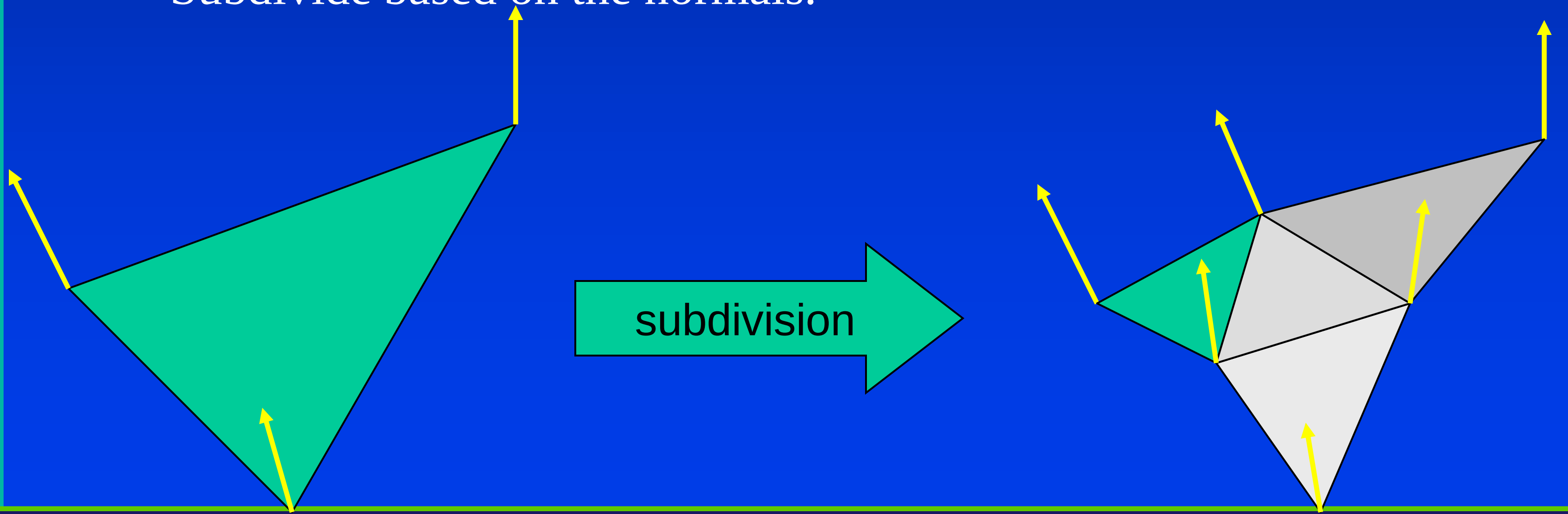
Surface Reconstruction



Hoppe et al. Siggraph 94

Local Subdivision Schemes

- Complex data structures required to perform subdivision.
  - Every polygon (triangle, quad, ..) must know its neighbors
  - Every vertex must know its neighbors
- Can we do something simpler?
  - Use vertex normal information to help "guess" about neighboring polygons.
  - Subdivide based on the normals.



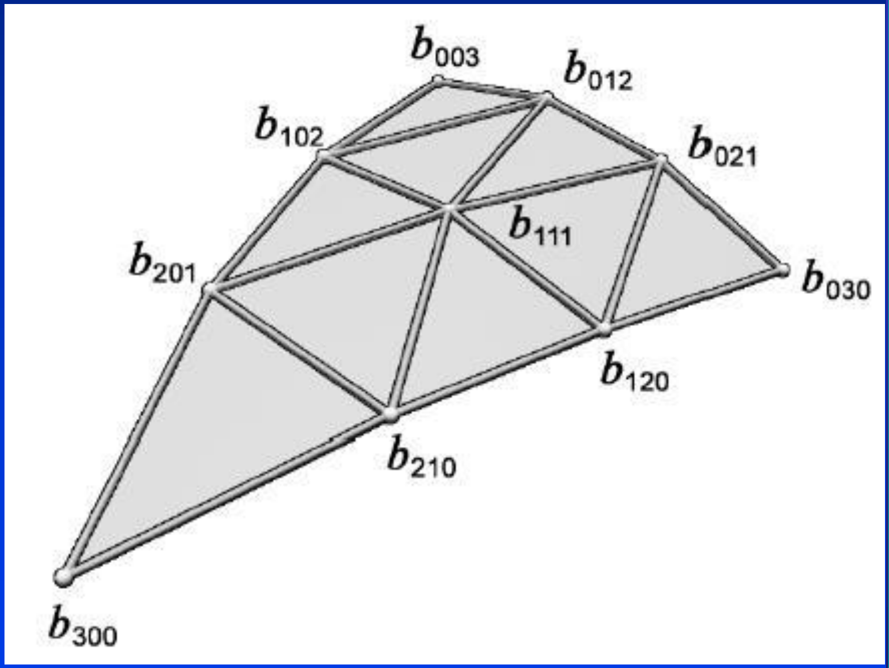
Local Subdivision (PN Triangles)

- Defined from “triangular bezier” patches.

u,v,w are barycentric coordinates
w=1-u-v, u,v,w≥1

$$b(u,v) = \sum_{i+j+k=3} b_{ijk} \frac{3!}{i!\,j!\,k!} u^i v^j w^k$$

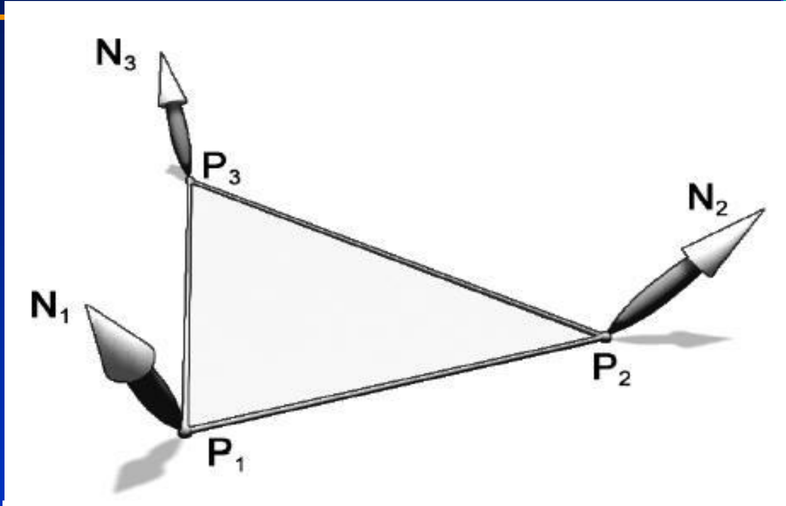
Bezier basis function

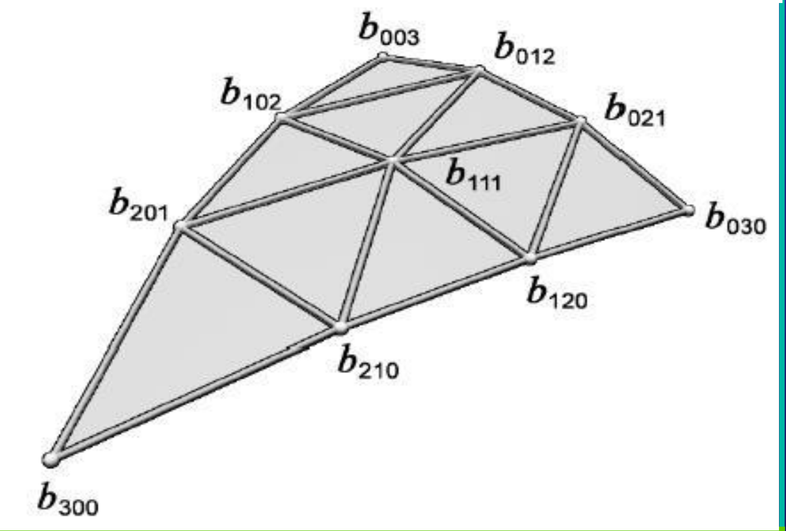


Curved PN Triangles
Alex Vlachos Joerg Peters Chas Boyd Jason Mitchel

Computing the Control Mesh

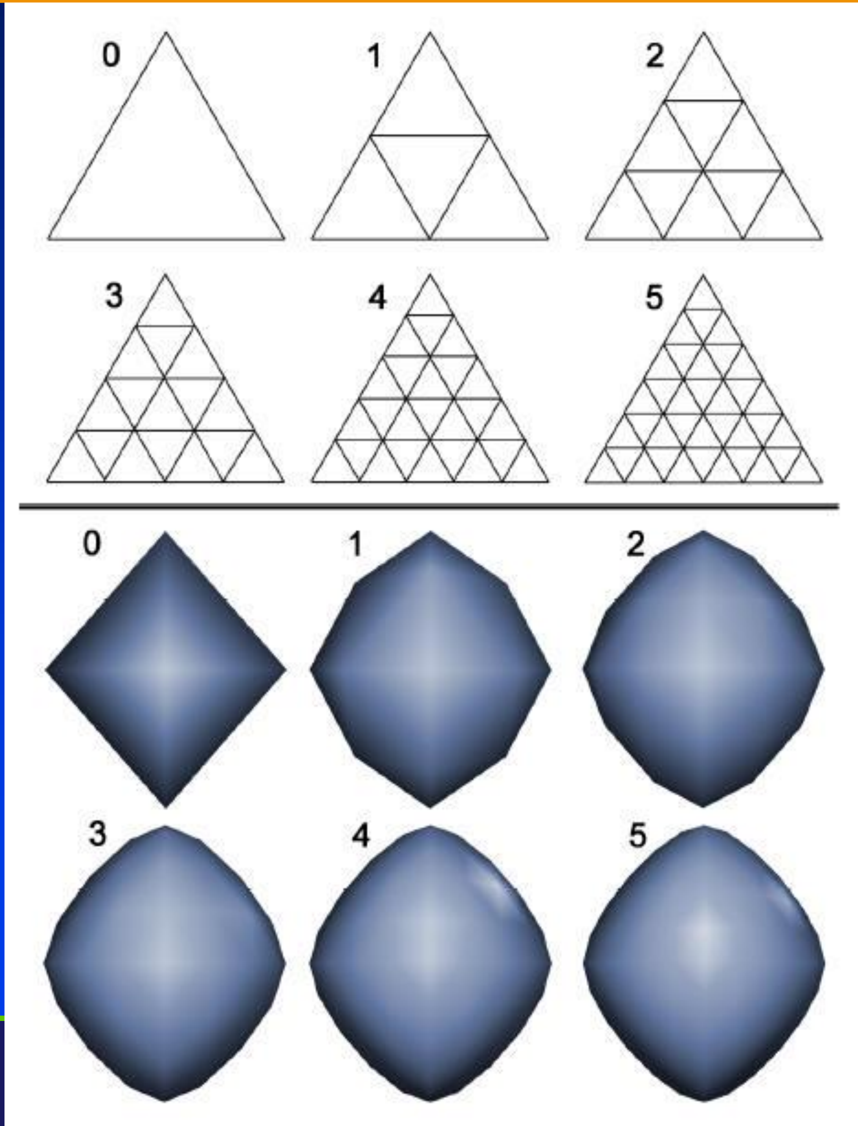
$$b_{300} = P_1$$
$$b_{030} = P_2$$
$$b_{003} = P_3$$
$$w_{ij} = (P_j - P_i) \bullet N_i$$
$$b_{210} = \left(2P_1 + P_2 - w_{12} N_1\right)/3$$
$$b_{120} = \left(2P_2 + P_1 - w_{21} N_2\right)/3$$
$$b_{021} = \left(2P_2 + P_3 - w_{23} N_2\right)/3$$
$$b_{012} = \left(2P_3 + P_2 - w_{32} N_3\right)/3$$
$$b_{102} = \left(2P_3 + P_1 - w_{31} N_3\right)/3$$
$$b_{201} = \left(2P_1 + P_3 - w_{13} N_1\right)/3$$
$$E = \left(b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{202}\right)/6$$
$$V = (P_1 + P_2 + P_3)/3$$
$$b_{111} = E + (E - V)/2$$





PN Triangles

- Interpolating Scheme.
- Example..



Local Subdivision

- Advantages
  - Easy to implement
    - No complex data structures
  - Easy to integrate into existing graphics applications
  - Hardware amenable
  - Looks good
- Disadvantages
  - No guarantees on higher level continuity.
  - Is limited in the amount of curvature it can provide.
  - In some sense it is a hack and not as “correct”.

Subdivision as Matrices

- Subdivision can be expressed as a matrix S_{mask} of weights w.
  - *S_{mask} is very sparse*
  - *Never Implement this way!*
  - Allows for analysis
    - Curvature
    - Limit Surface

$$S_{mask} P = \hat{P}$$

$$\begin{bmatrix} w_{00} & w_{01} & \cdots & 0 \\ w_{10} & w_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & w_{nj} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} \hat{p}_0 \\ \hat{p}_1 \\ \hat{p}_2 \\ \vdots \\ \hat{p}_0 \end{bmatrix}$$

S_{mask} Weights

Old Control Points

New Points

What about Continuity

- Subdivision mask weights *w* are derived from splines, such as B-Splines.
  - Subdivision surfaces converge to spline surfaces with C^2 continuity everywhere.**
  - Too lengthy to cover here, but there is lots of literature.

Subdivision Methods for Geometric Design

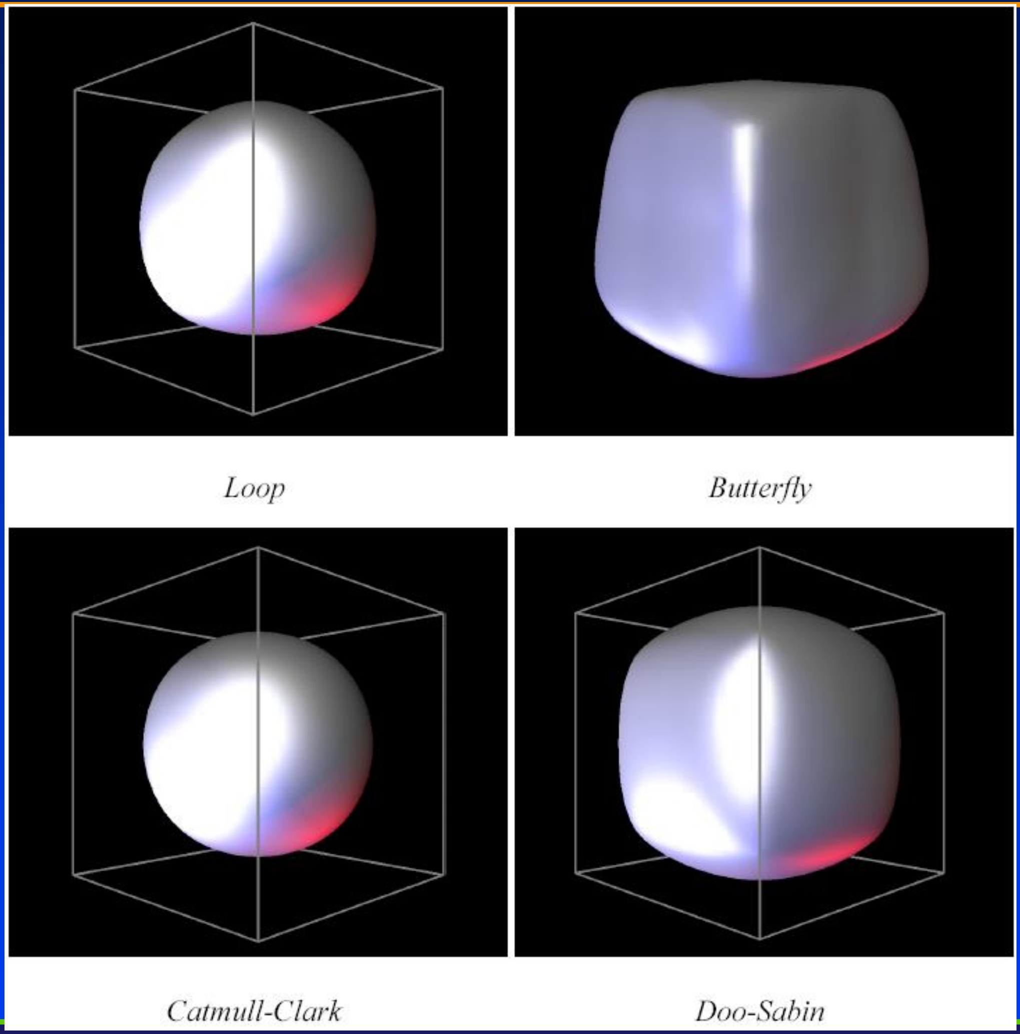
Joe Warren, Henrik Weimer. (2002)

Math works out except at "Extraordinary Vertices**".

Most Subdivision Schemes have and "ideal" valence for which it can be shown that the limit surface will converge to a spline surface.

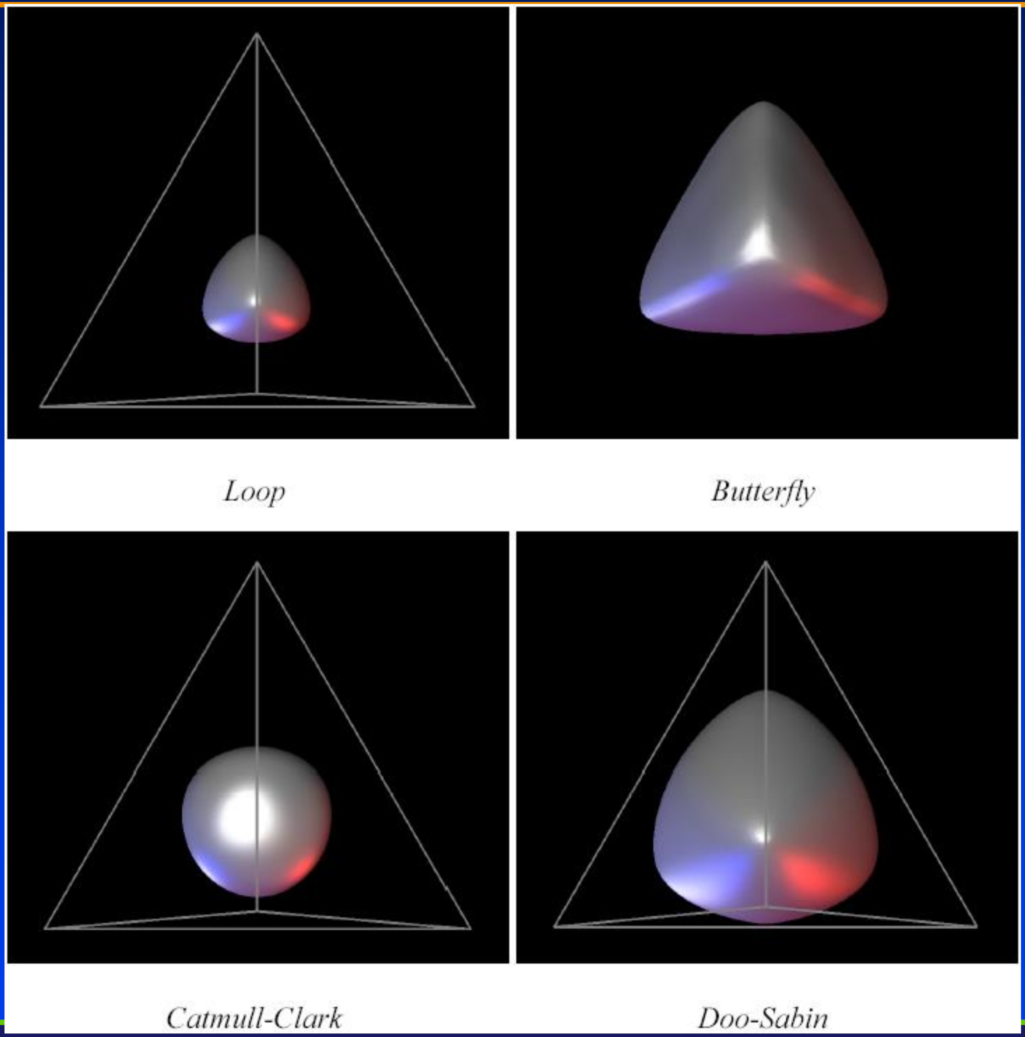
Comparison

- Catmull-Clark yields the nicest surface.
- Loop is more asymmetric.
- Mod. Butterfly is the worst.

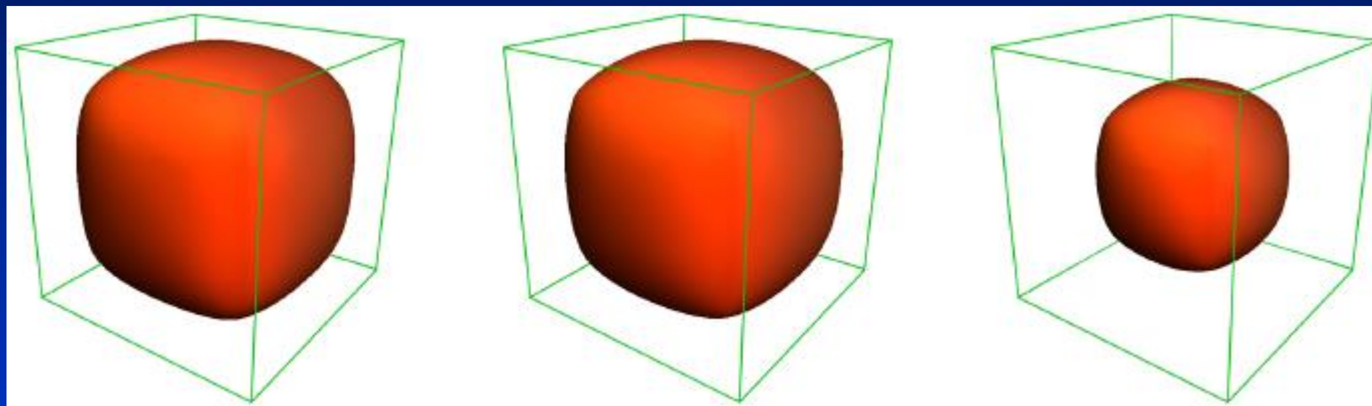


Comparison

- Extreme shrink for Loop and Catmull-Clark.

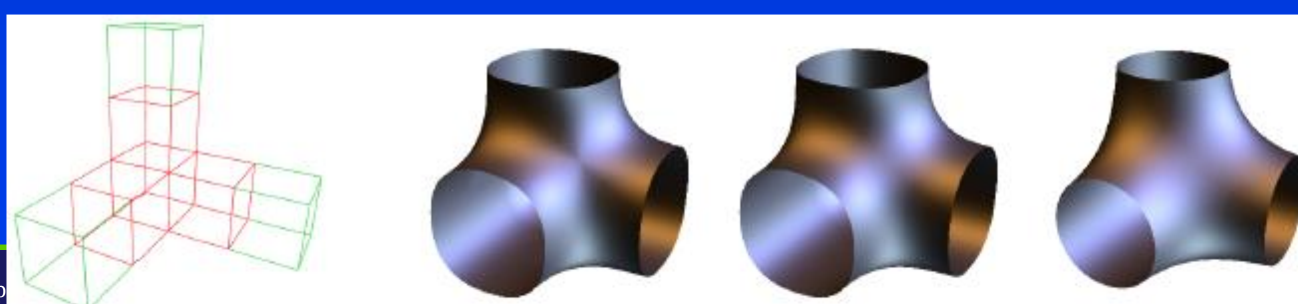


Comparison

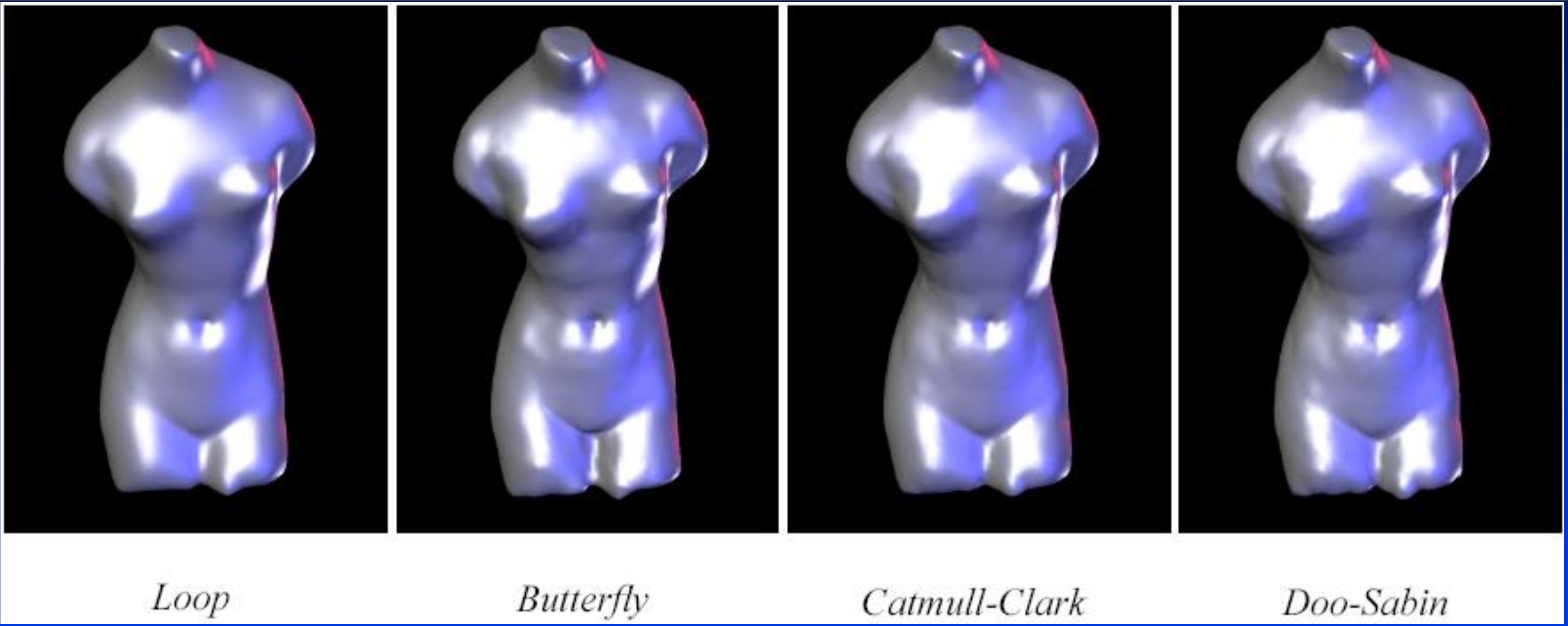


Midedge **Doo-Sabin** **Biquadric**

- The increasing shrinkage with increasing smoothness.

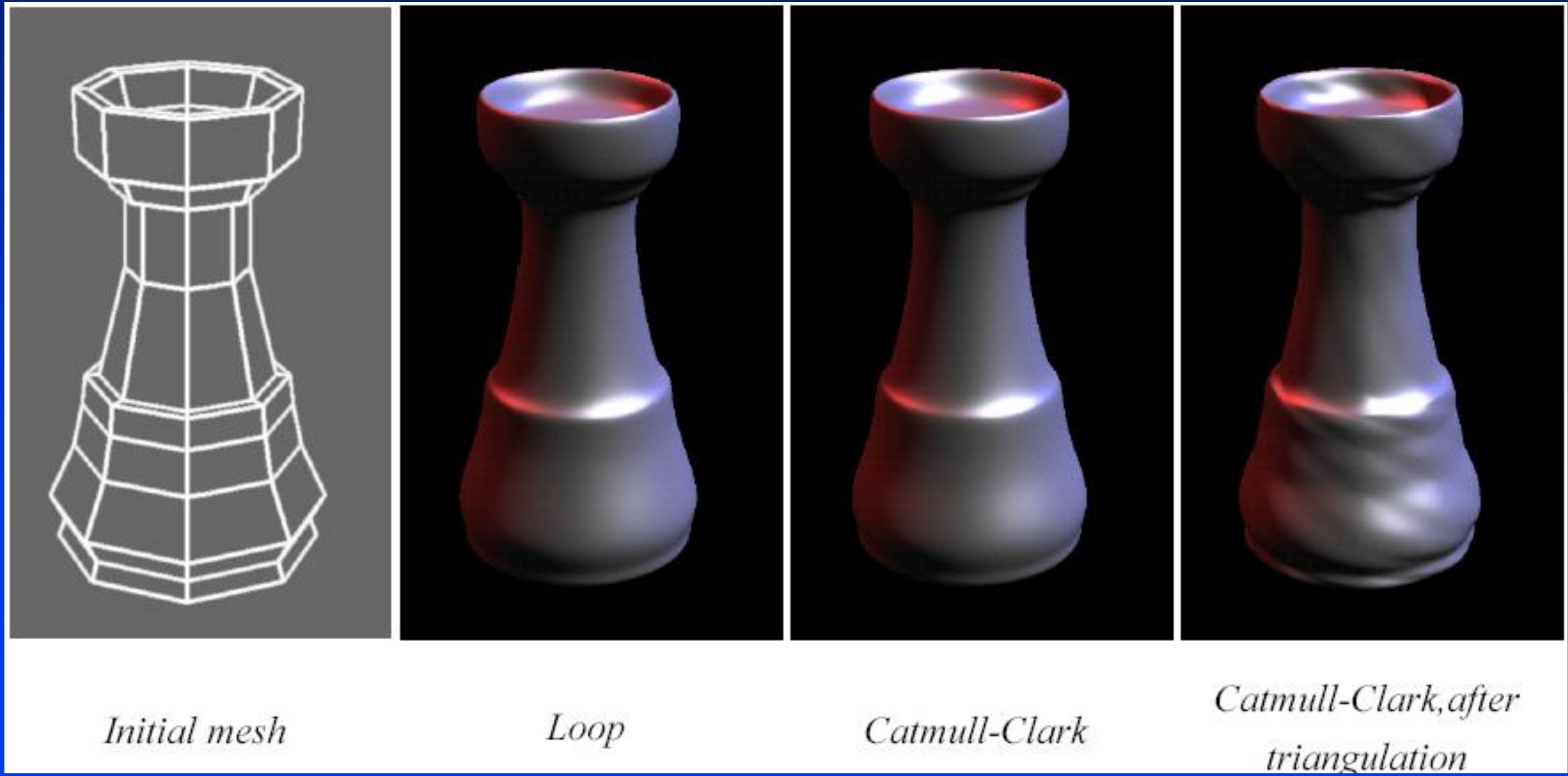


Comparison



- Similar results
- Interpolating schemes are sensitive to the presence of sharp features, and may produce low quality surfaces unless the initial mesh is smooth enough.

Comparison


Initial mesh
Loop
Catmull-Clark
Catmull-Clark, after triangulation

Comparison

- Loop and Catmull-Clark appear to be the best choices for most applications.
  - Loop seems to be more reliable.
- Quadrilateral scheme
  - Natural texture mapping for quads.
  - Natural number of symmetries?
- Curvature continuity
  - No C^1 with small support.

Subdivision

- Pro
  - No Trimming
  - Connectivity and Smoothness Guaranteed
- Con
  - Not much studies like NURBS

Subdivision Surface =

polygons	B-splines
+ flexible	-restrictive
-faceted	+smooth

"Geri's Game"

Subdivision Surfaces in the Making of Geri's Game

DeRose et al. Siggraph 98

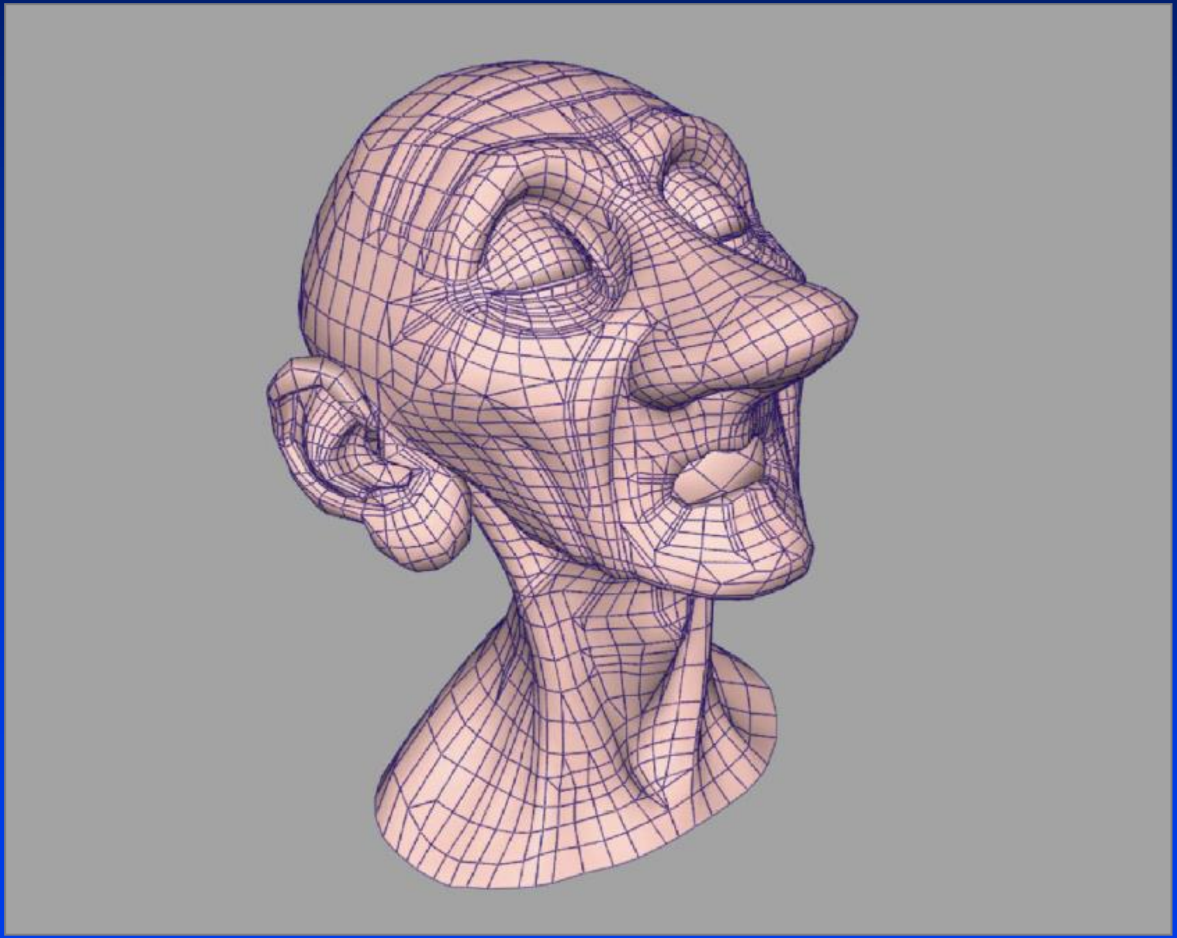
Subdivision in Production Environment

- Traditionally spline patches (NURBS) have been used in production for character animation.
- Difficult to control spline patch density in character modeling.

(Geri's Game, Pixar 1998)

Subdivision in Character Animation
Tony Derose, Michael Kass, Tien Troung
(SIGGRAPH '98)

Gery's Head



Catmull-Clark Surface Modeling

- Subdivision produces smooth continuous surfaces.
- How can “sharpness” and creases be controlled in a modeling environment?

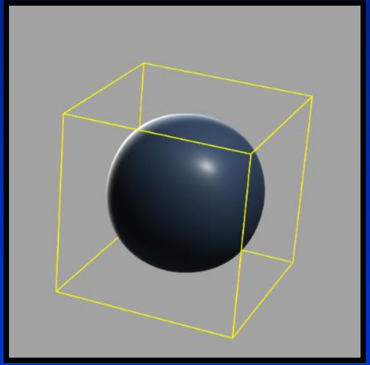
ANSWER: Define new subdivision rules for “creased” edges and vertices.

1. Tag Edges sharp edges.
2. If an edge is sharp, apply new sharp subdivision rules.
3. Otherwise subdivide with normal rules.

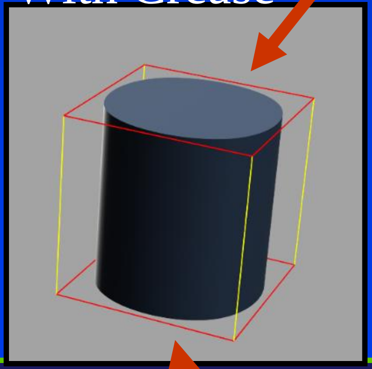
Modeling Fillets and Blends

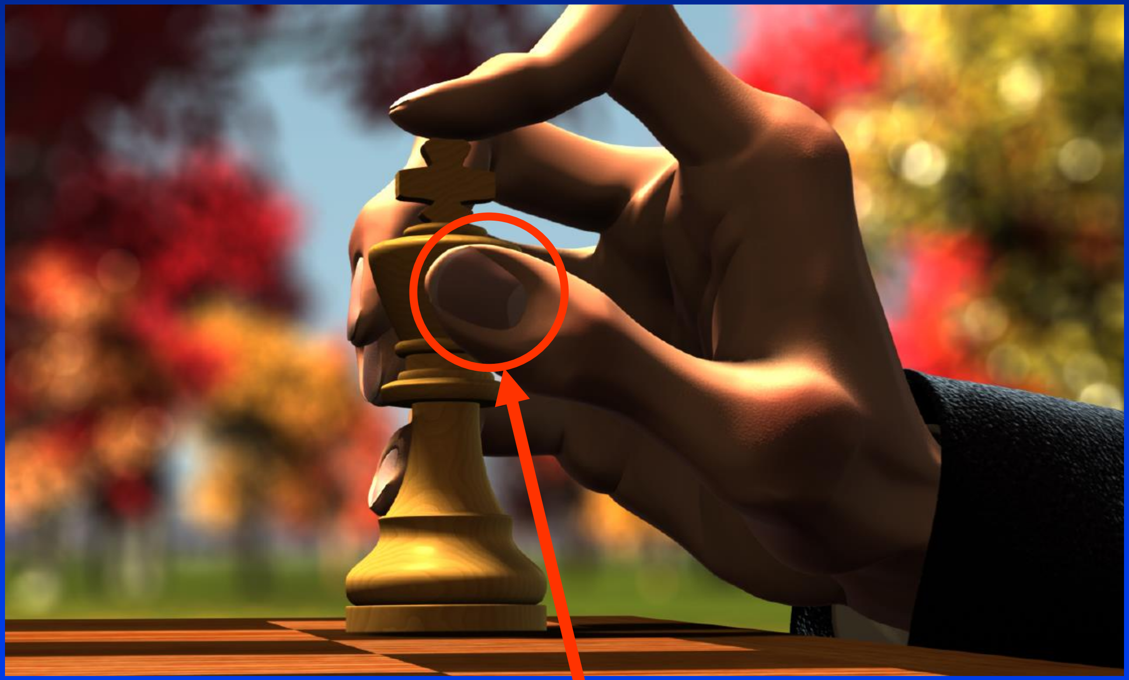
- Infinitely sharp creases

Without Crease

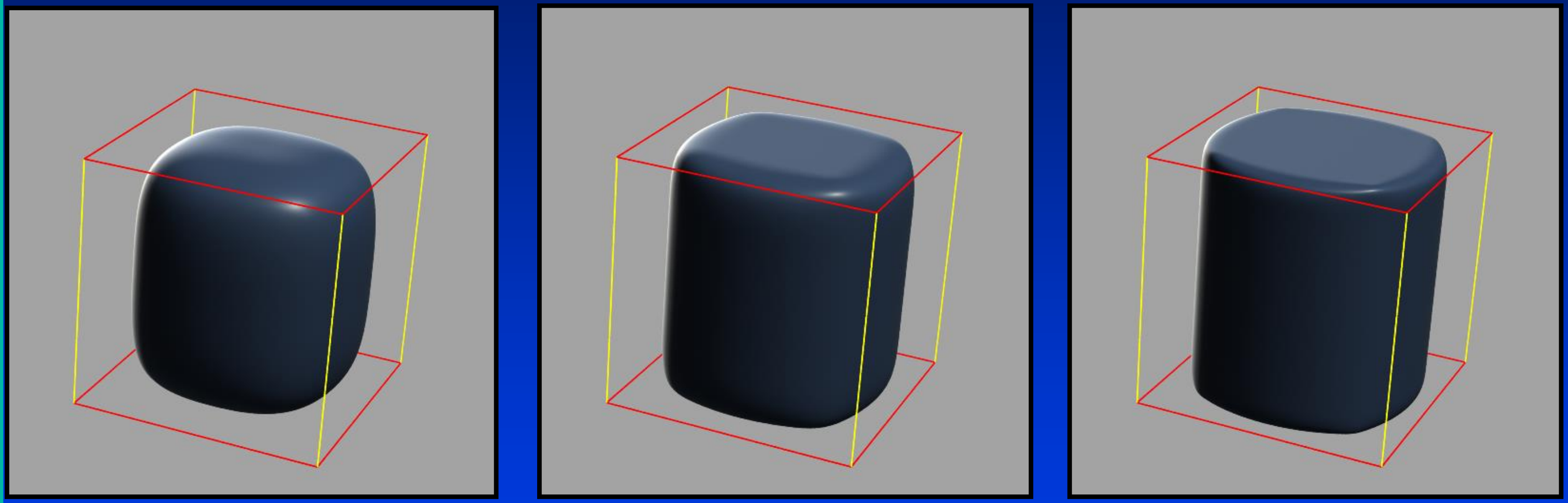


With Crease





Semi-sharp Creases

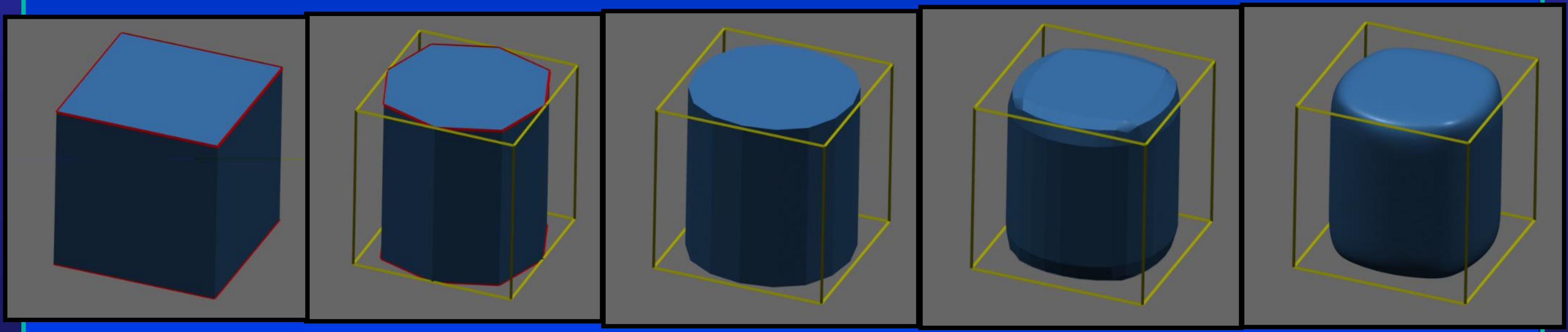


- Modify averaging rules
- Hybrid subdivision

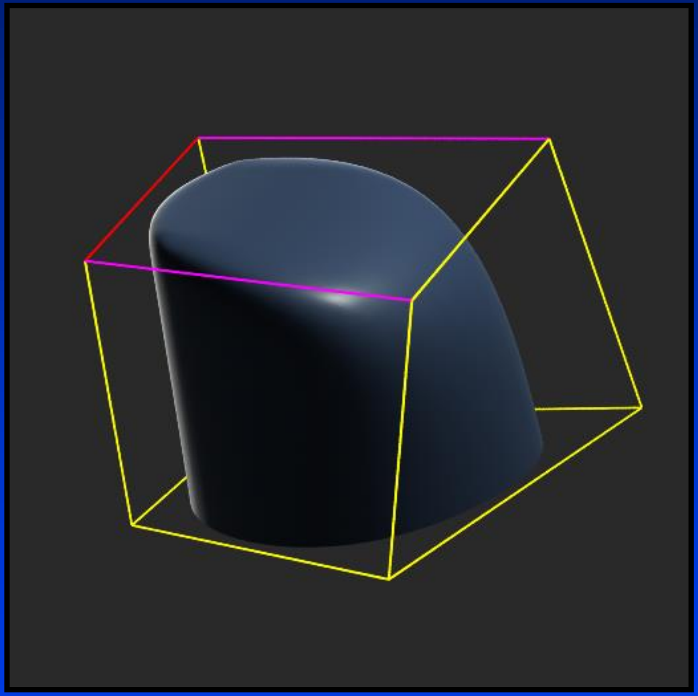
Integer Sharpness s

- Subdivide *s* times using sharp rules
- Use smooth rules to the limit surface

Example ***s = 2***



Variable Sharpness

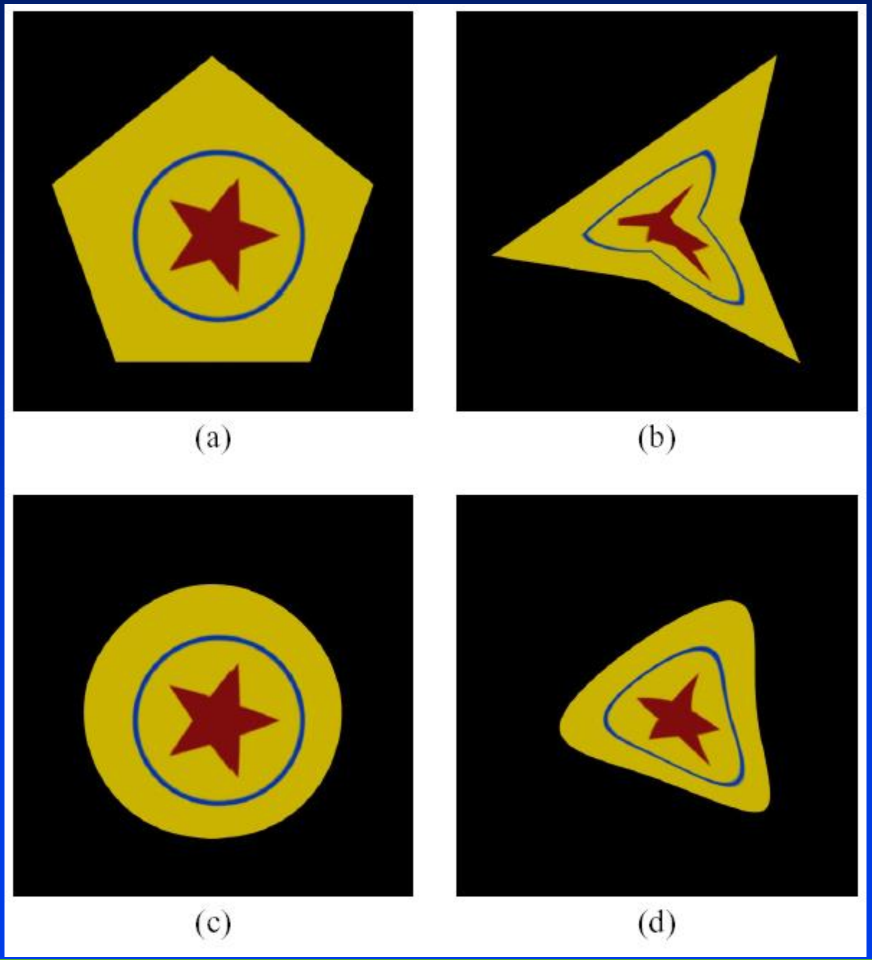


Model courtesy of
Jason Bickerstaff

Texturing

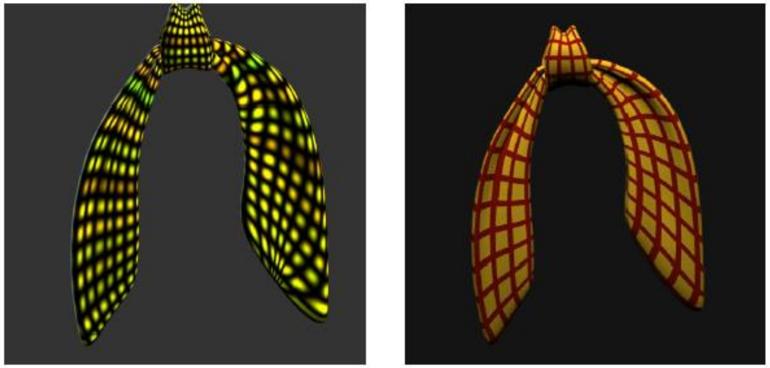
Scalar Fields provide texture coordinates.

$$S(s,t) = (x(s,t), y(s,t), z(s,t))$$



Texturing

- Specify parameters independent of subdivision level
- Assign parameters at control vertices.
- Subdivide using same rules.
- Interpolating using Laplacian smoothing or Painting an intensity map



(a) (b)

Implementation Issues

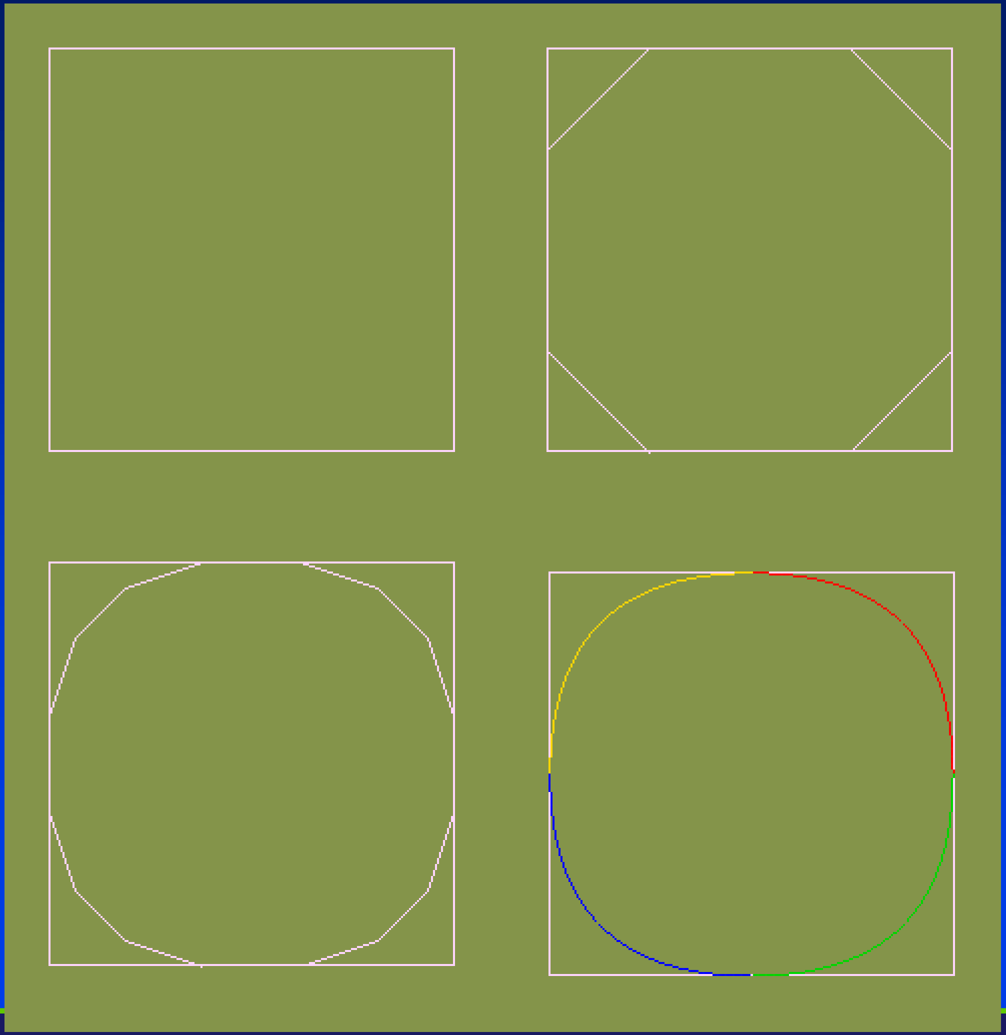
- Subdivision surfaces now implemented in RenderMan.
- Regular mesh regions -> B-splines.
- Using B-splines allows
  - Efficiency in memory usage
  - Reduce the total amount of splitting
  - Forward algorithms are available to dice B-spline patches
- An advantage of semi-sharp creases
  - Never tear

- Pixar Developments make subdivision surfaces very practical and useful
- Subdivision > NURBS
  - More control, accuracy
  - Time saved, To be refined locally
  - Remove two obstacles by developing semi-sharp creases and scalar fields
  - An efficient data structure and cloth energy function well suited to physical and cloth simulation
- Now part of Renderman

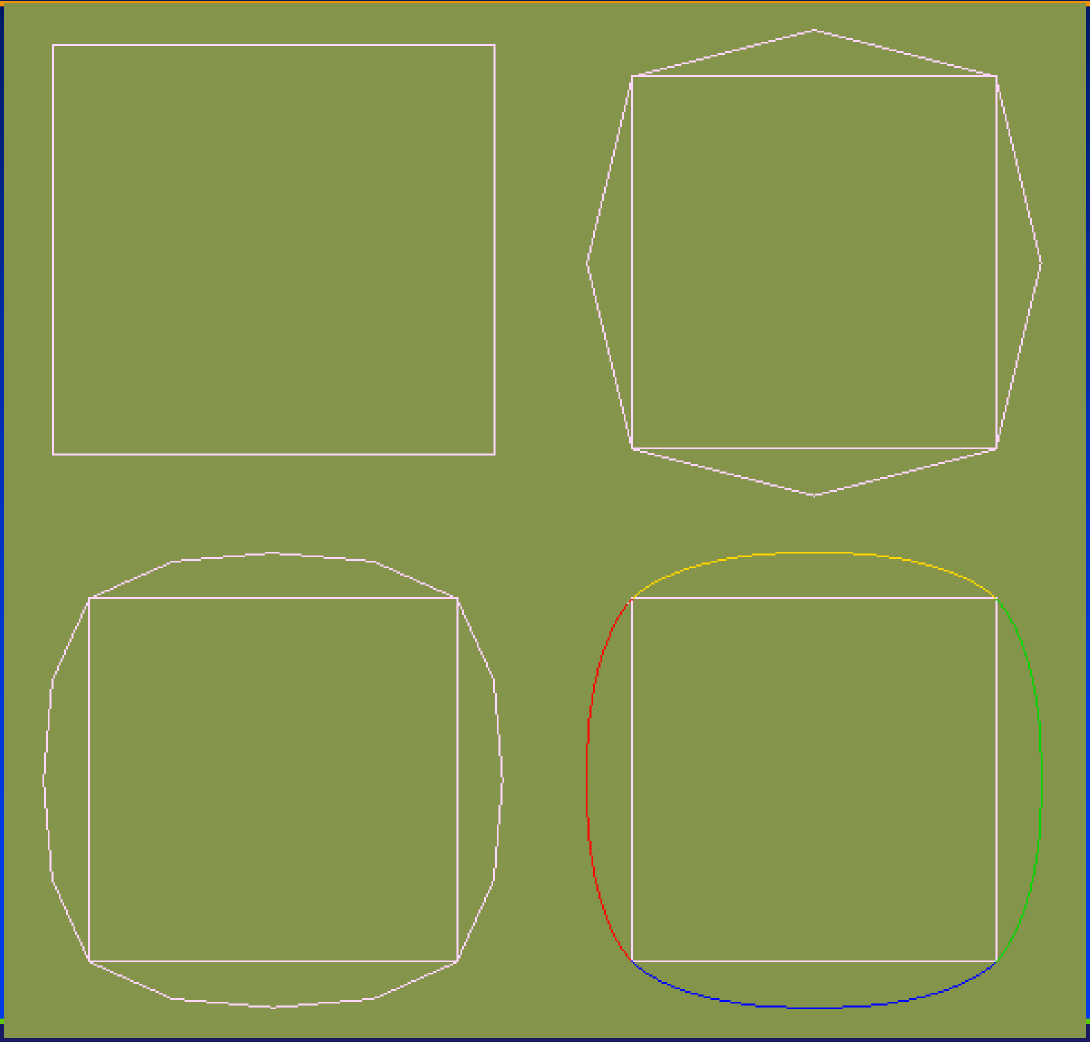
Subdivision Splines

- We treat subdivision as a novel method to produce spline-like models in the limit
- Key components for spline models
  - Control points, basis functions over their parametric domain, parameterization, piecewise decomposition
- Parameterization is done naturally via subdivision
- The initial control mesh serves as the parametric domain
- Basis functions are available for regular settings as well as irregular settings
- Control points for one patch are in the vicinity of its parametric domain from its initial control vertices
- Subdivision-based spline formulation is fundamental for physics-based geometric modeling and design, finite element analysis, simulation, and the entire CAD/CAM processes

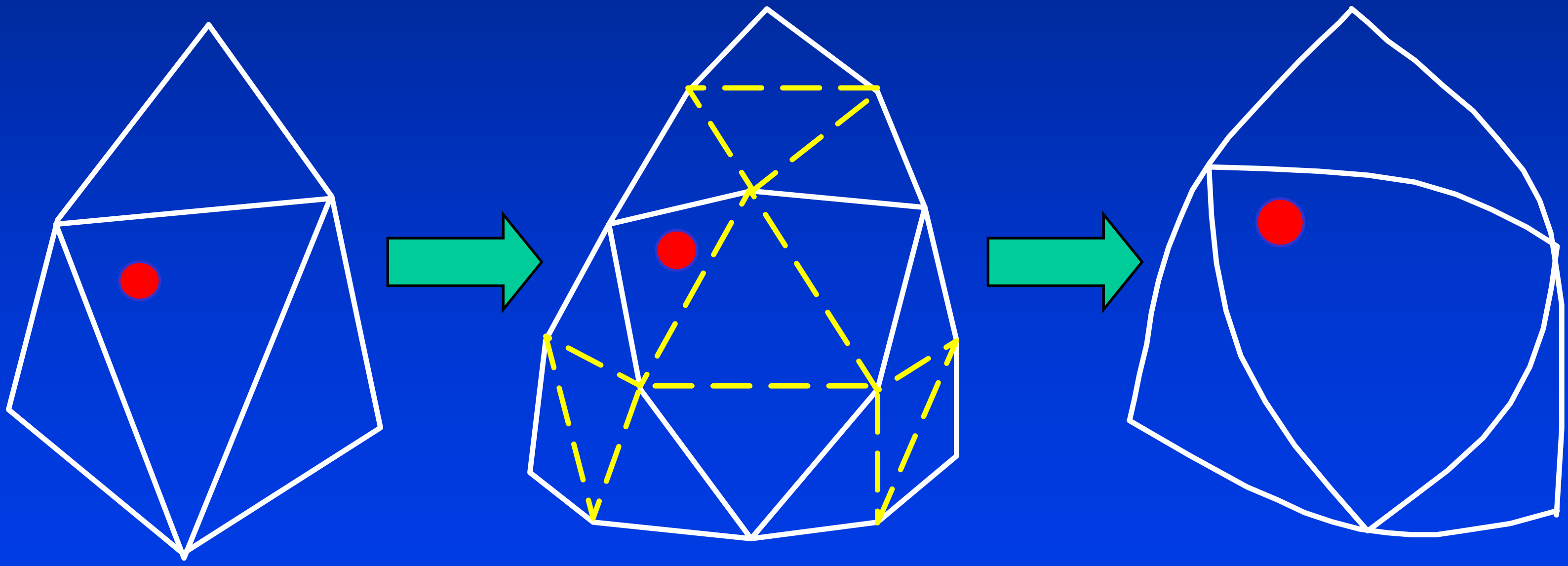
Chaikin Curve Example



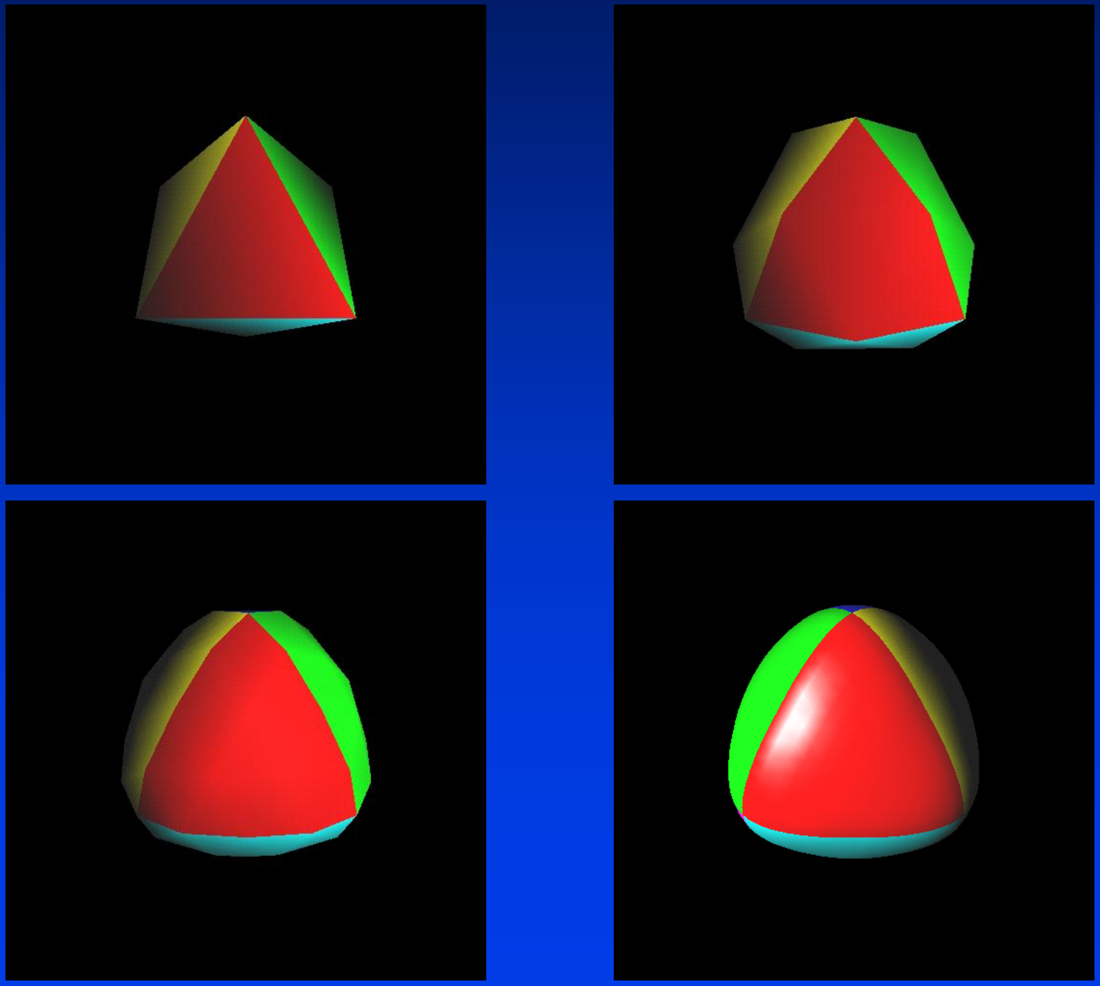
Interpolation Curve Example



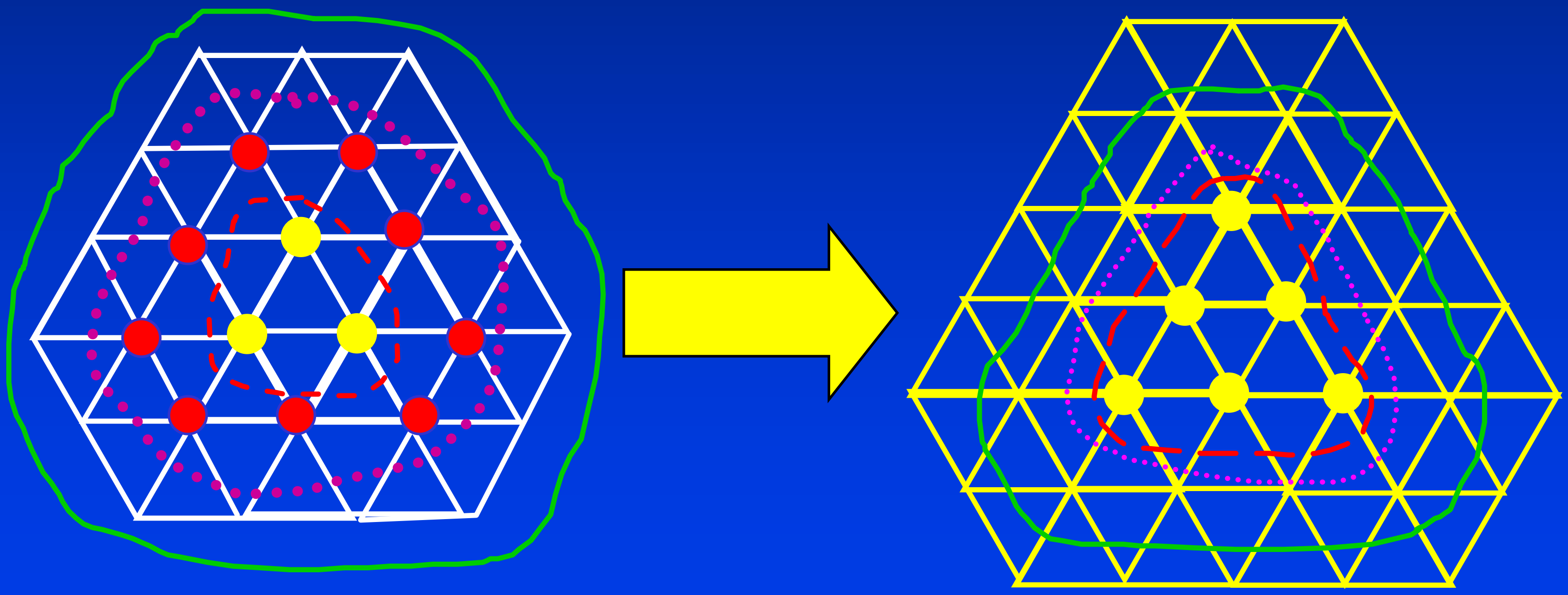
Parameterization



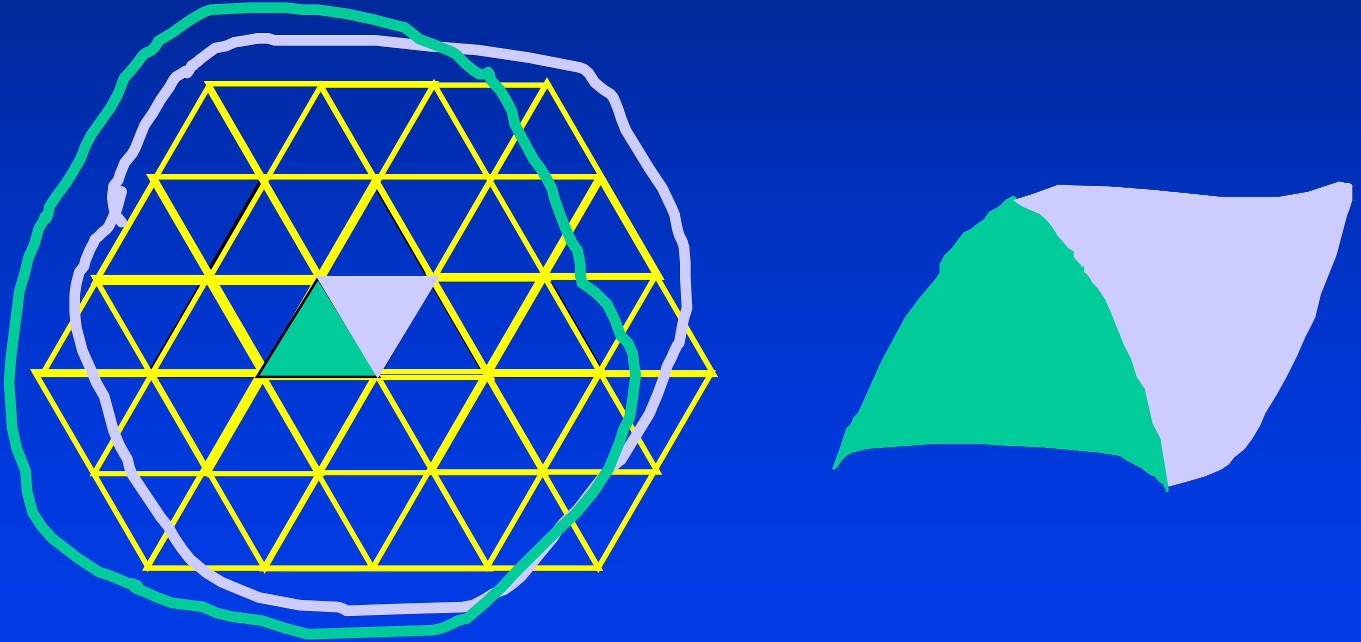
Butterfly Surface Example



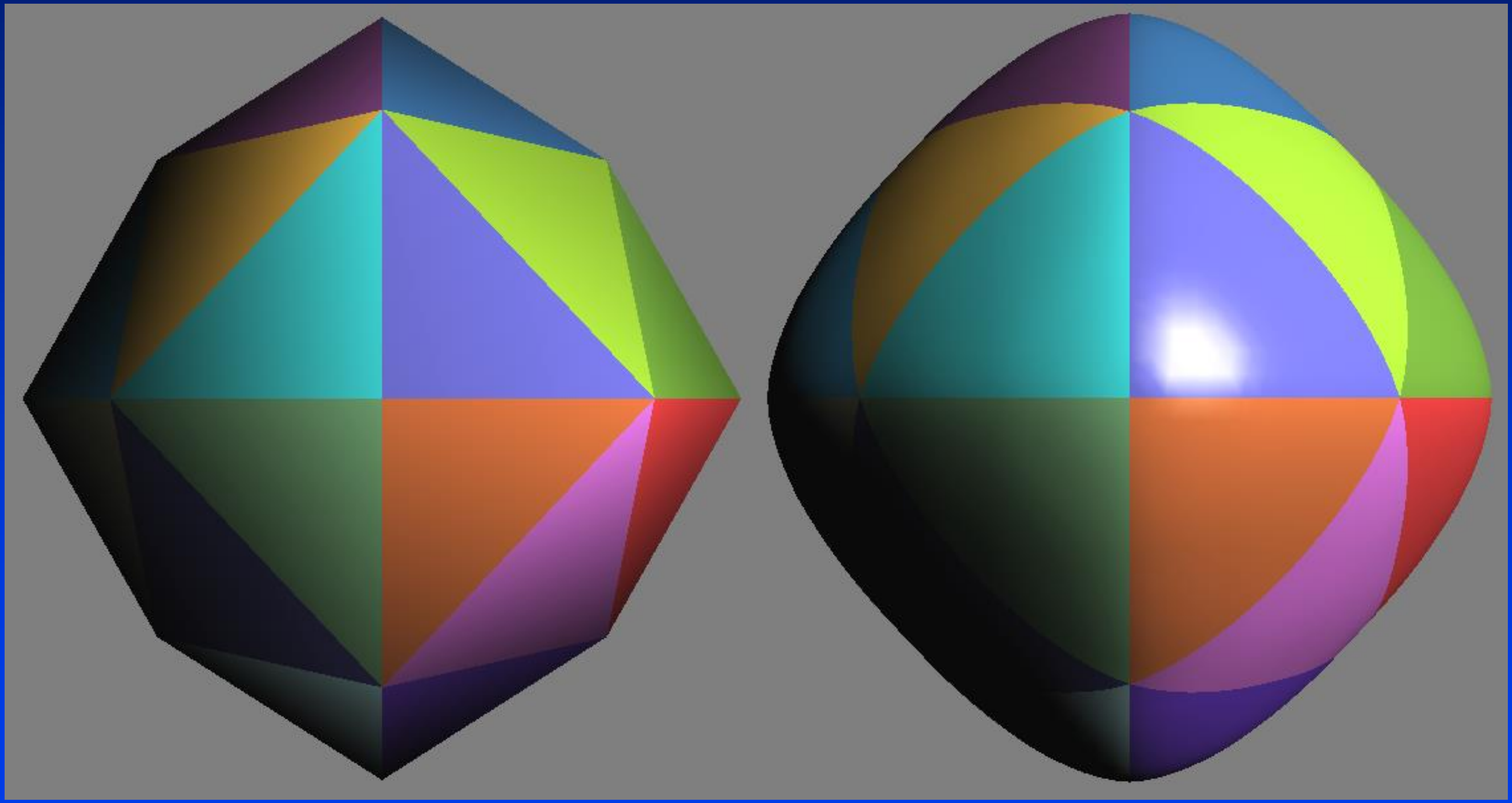
Control Vertices for Butterfly Surface



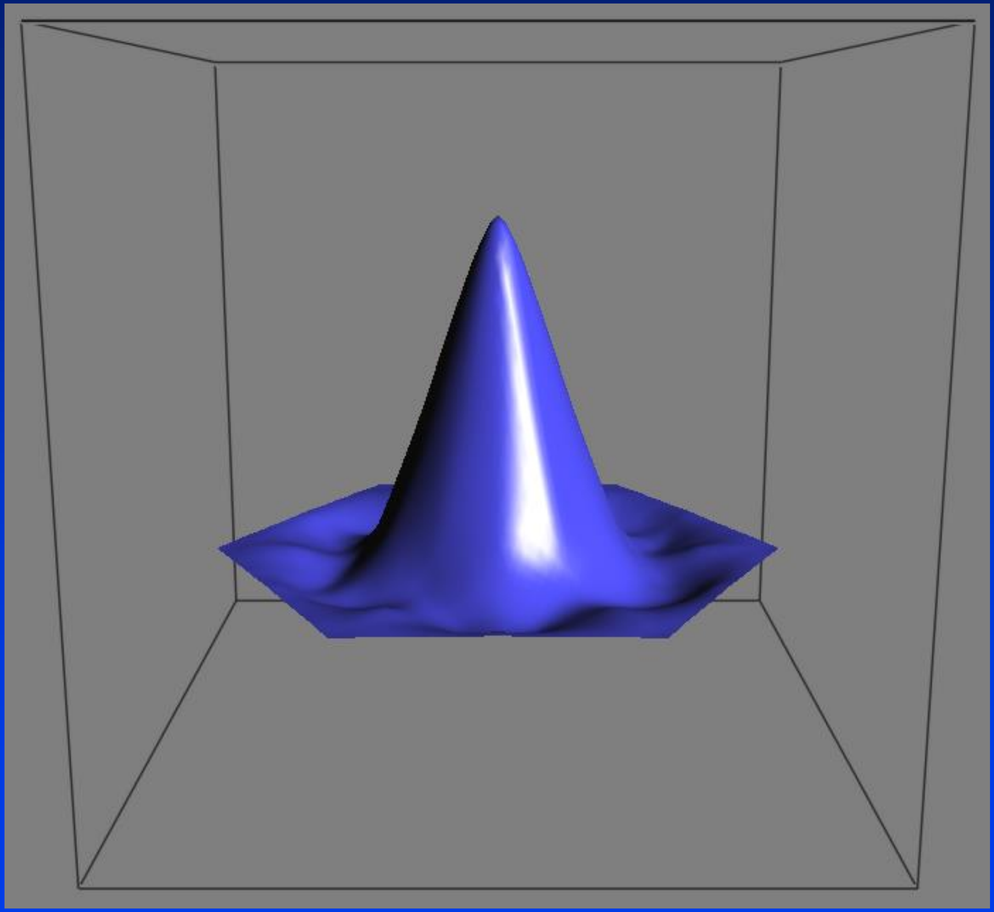
Control Vertices for Surface Patches



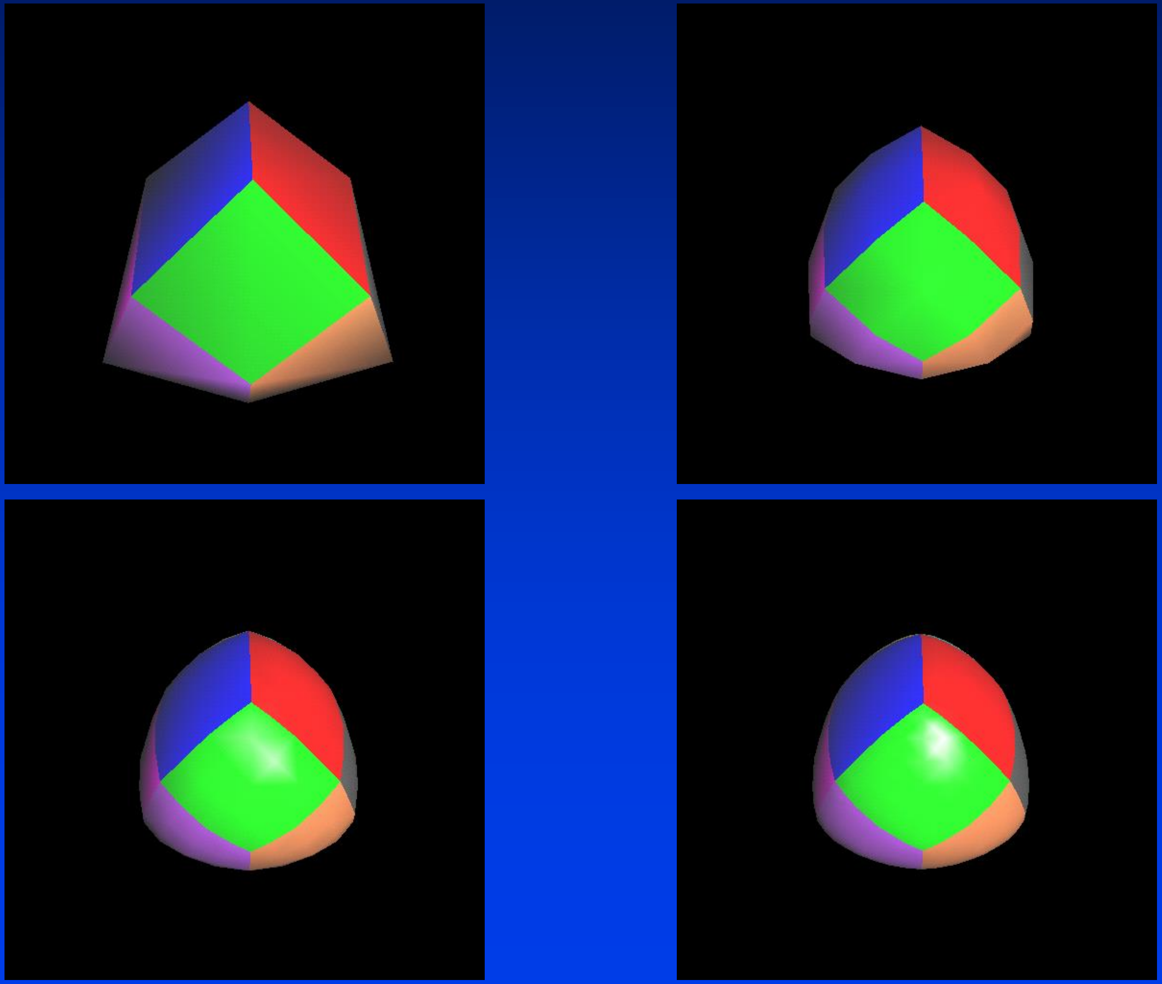
Butterfly Patches



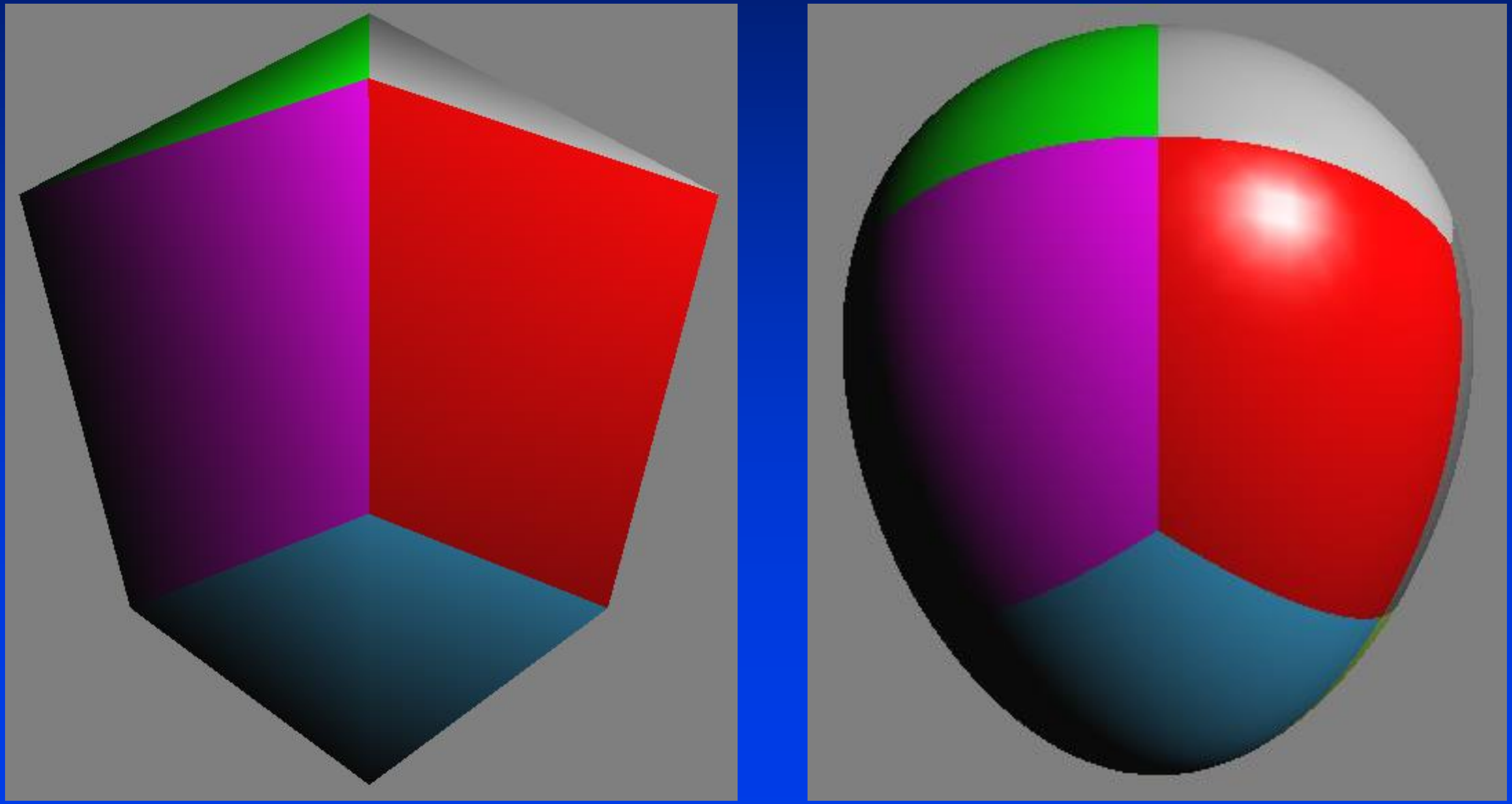
Butterfly Basis Function



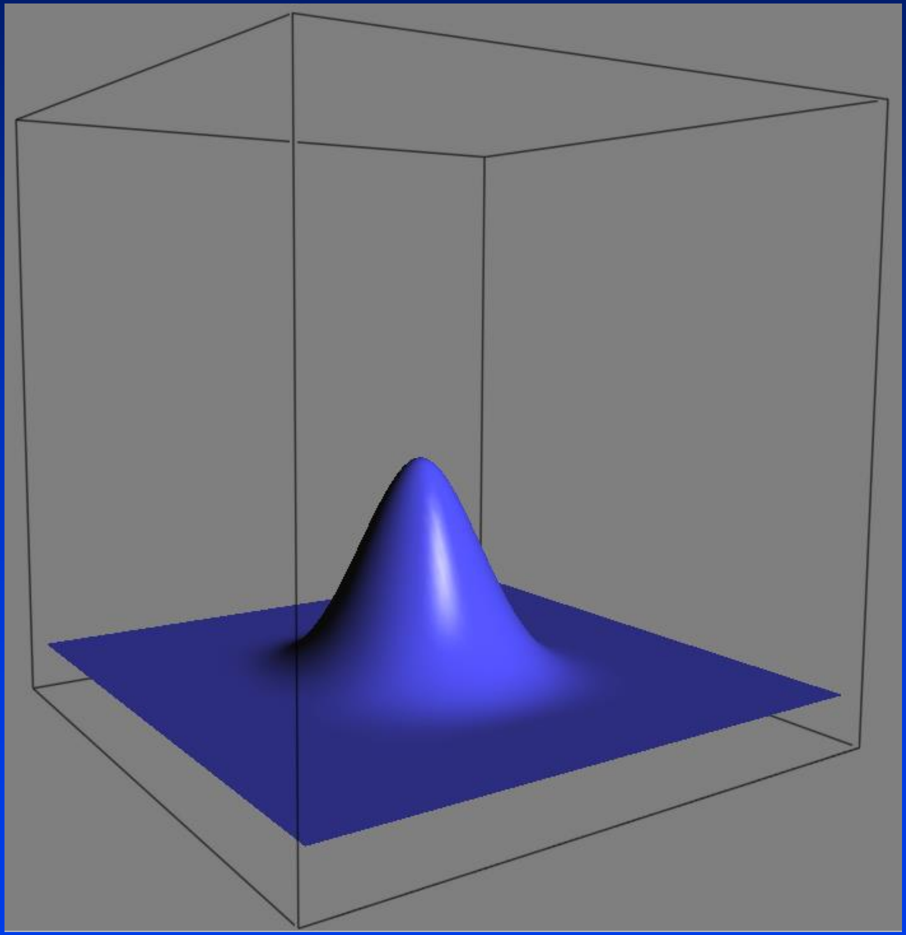
Catmull-Clark Surface Example



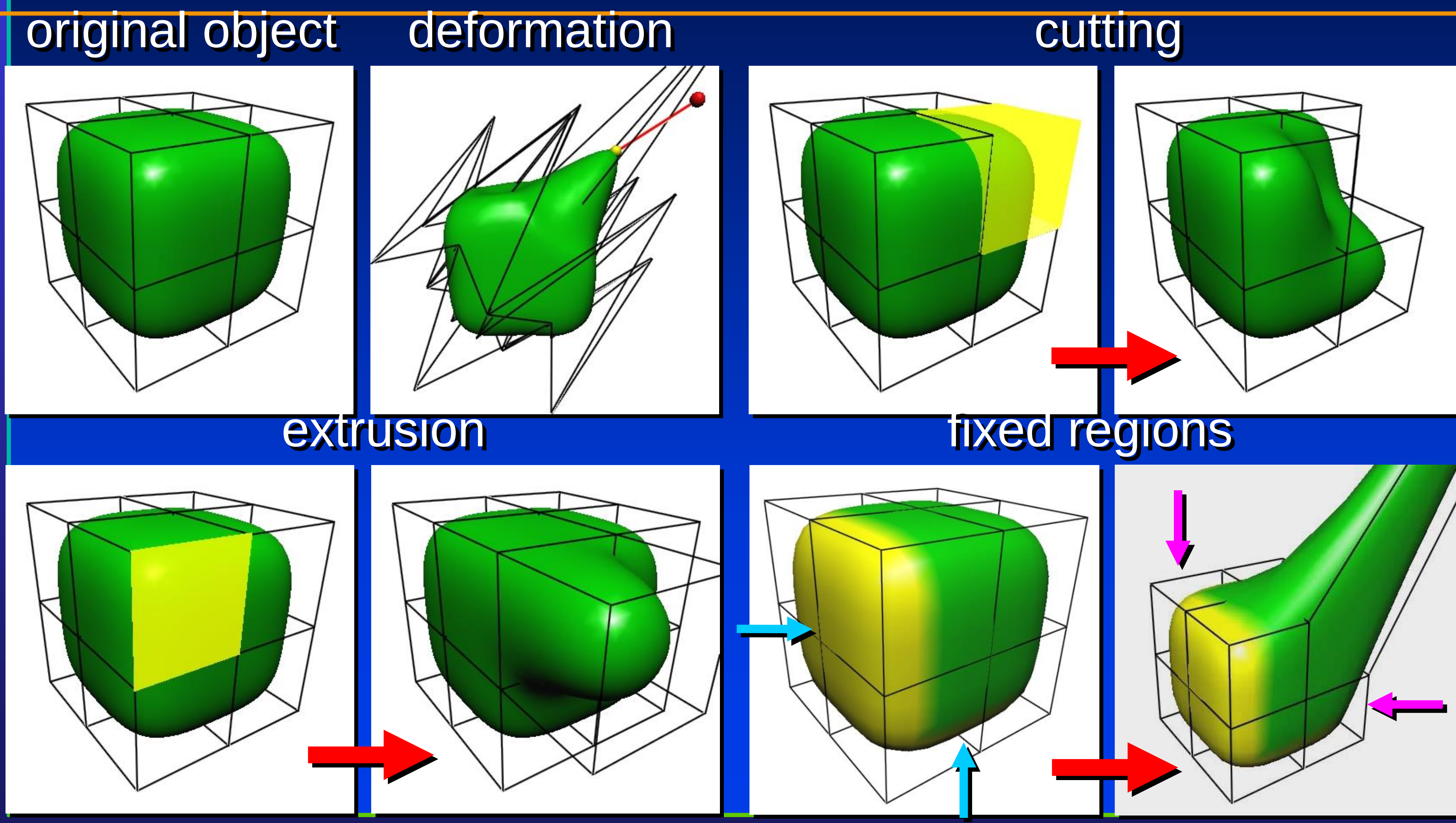
Catmull-Clark Patches



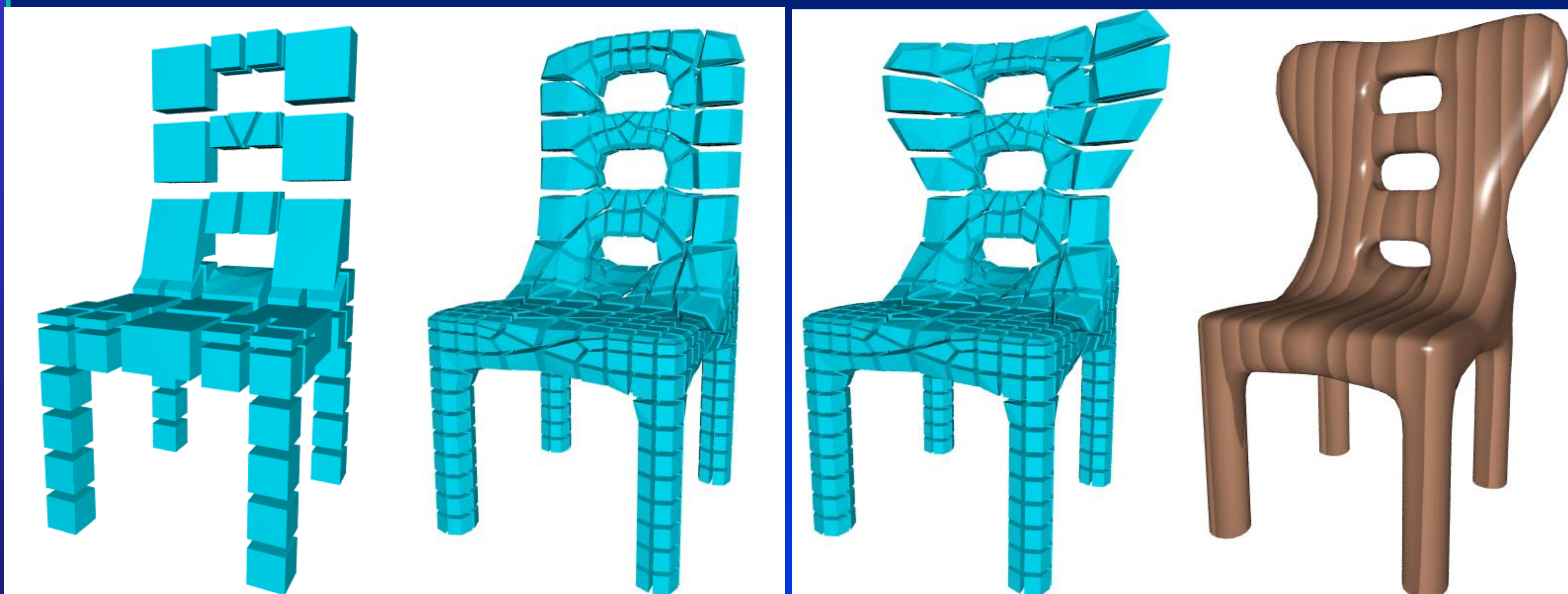
Catmull-Clark Basis Function



Simple Sculpting Examples



Chair Example --- Finite Element Simulation



Initial control lattice

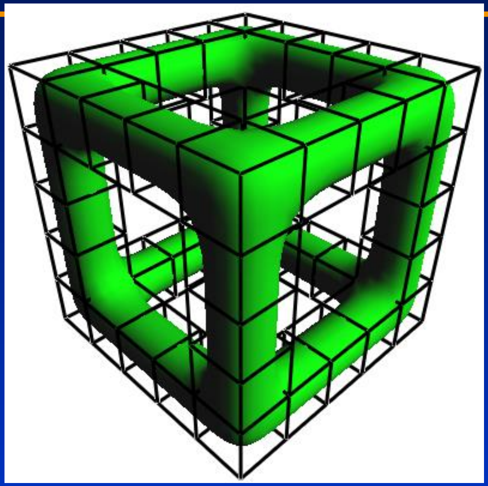
Finite element structure after a few subdivisions

Deformed object

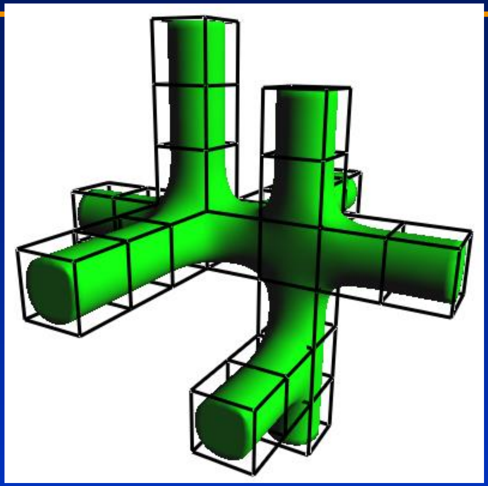
Photo-realistic rendering

Sculpting Tools

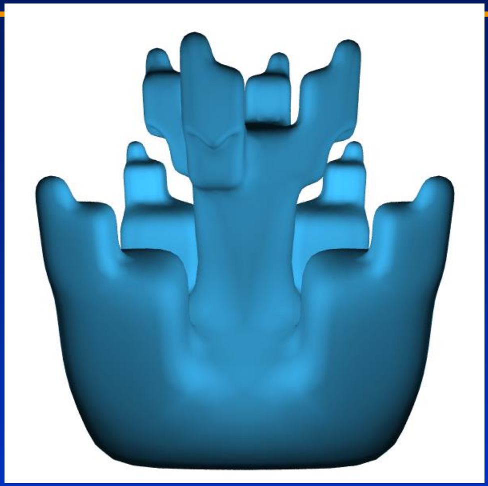
carving



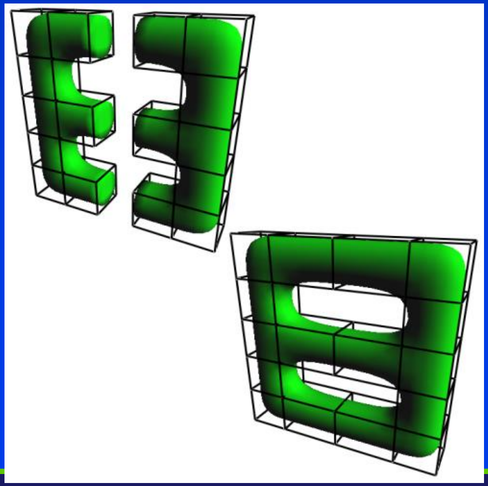
extrusion



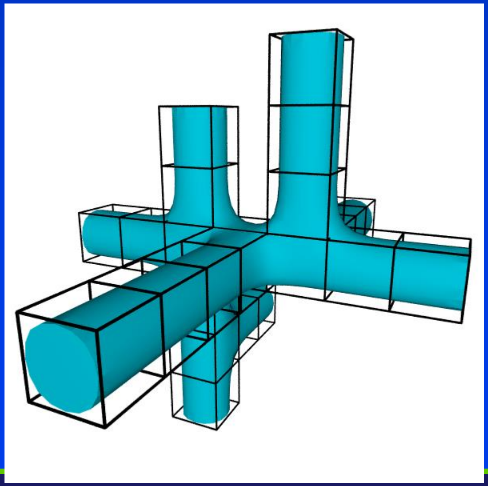
detail editing



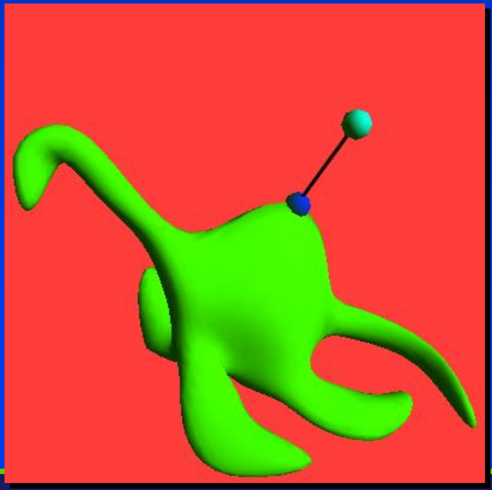
joining



sharp features

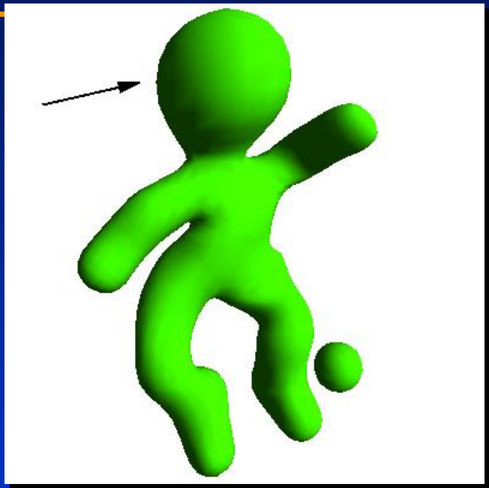


deformation

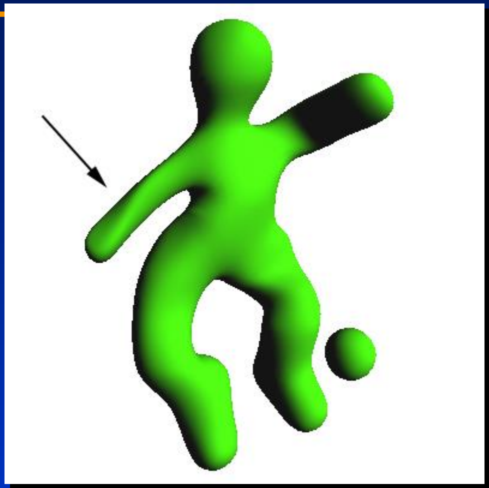


Sculpting Tools

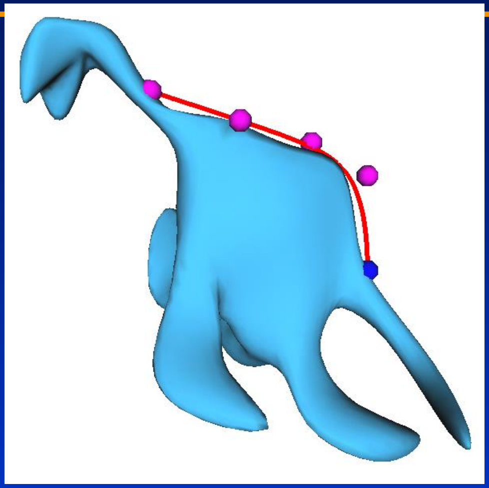
inflation



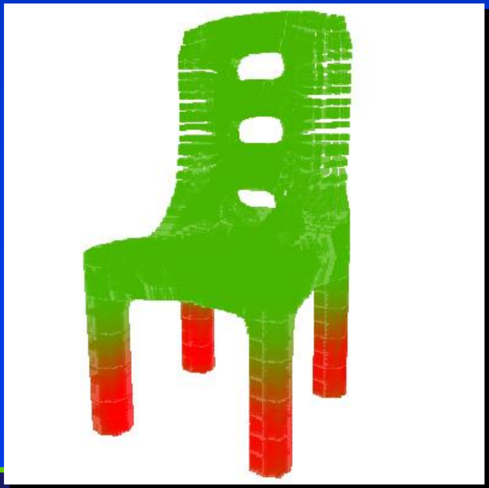
deflation



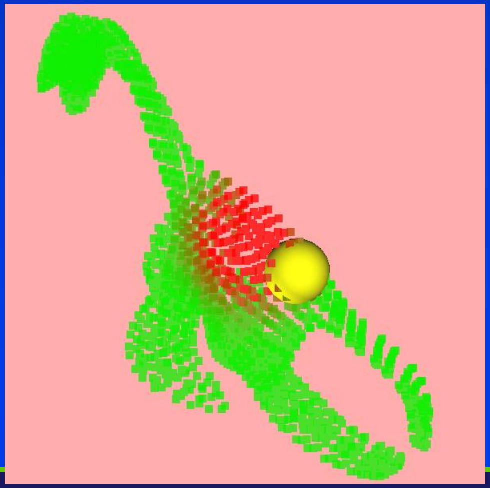
curve-based design



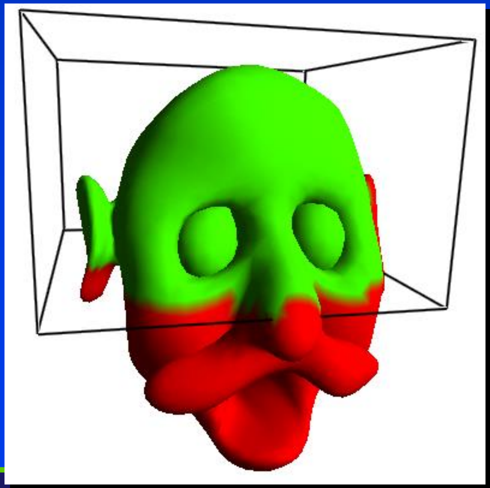
material mapping



material probing

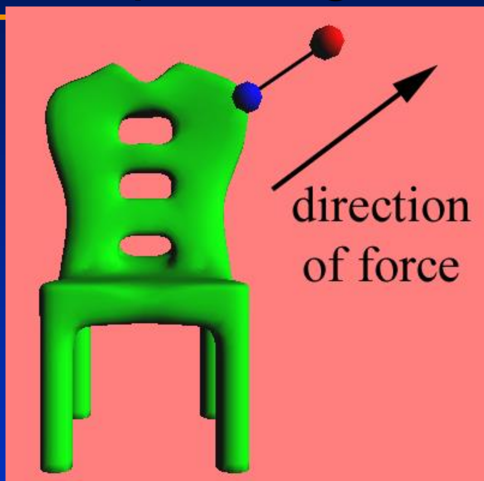


physical window

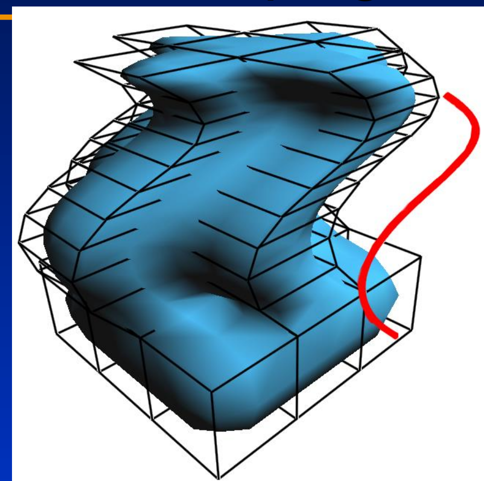


Sculpting Tools

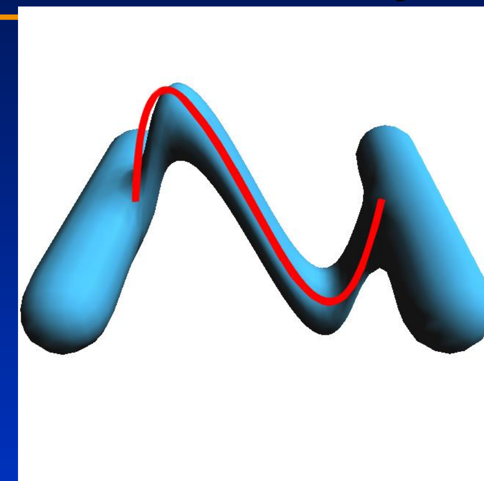
pushing



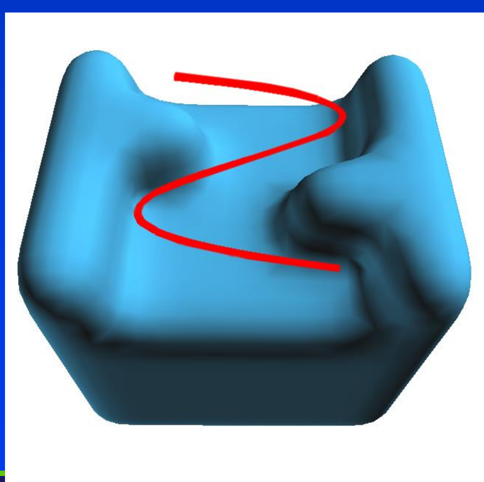
sweeping



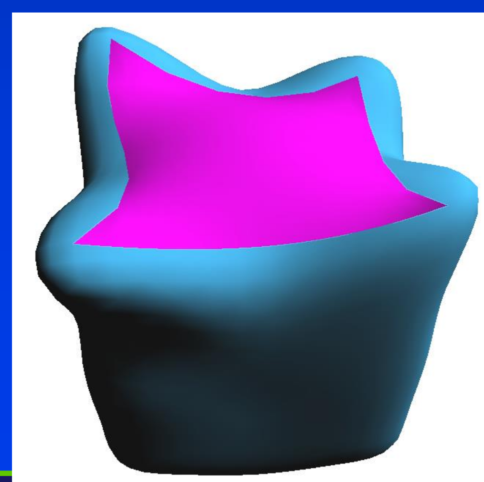
curve-based join



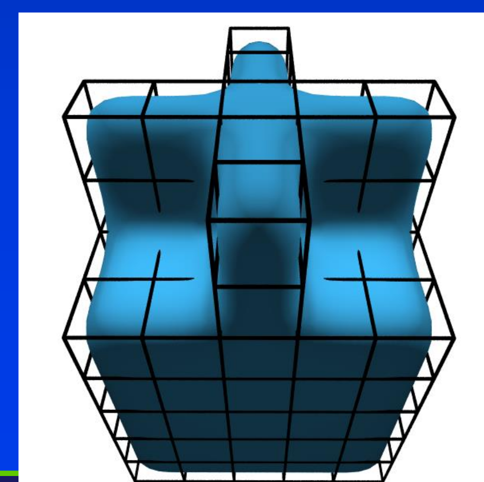
curve-based cutting



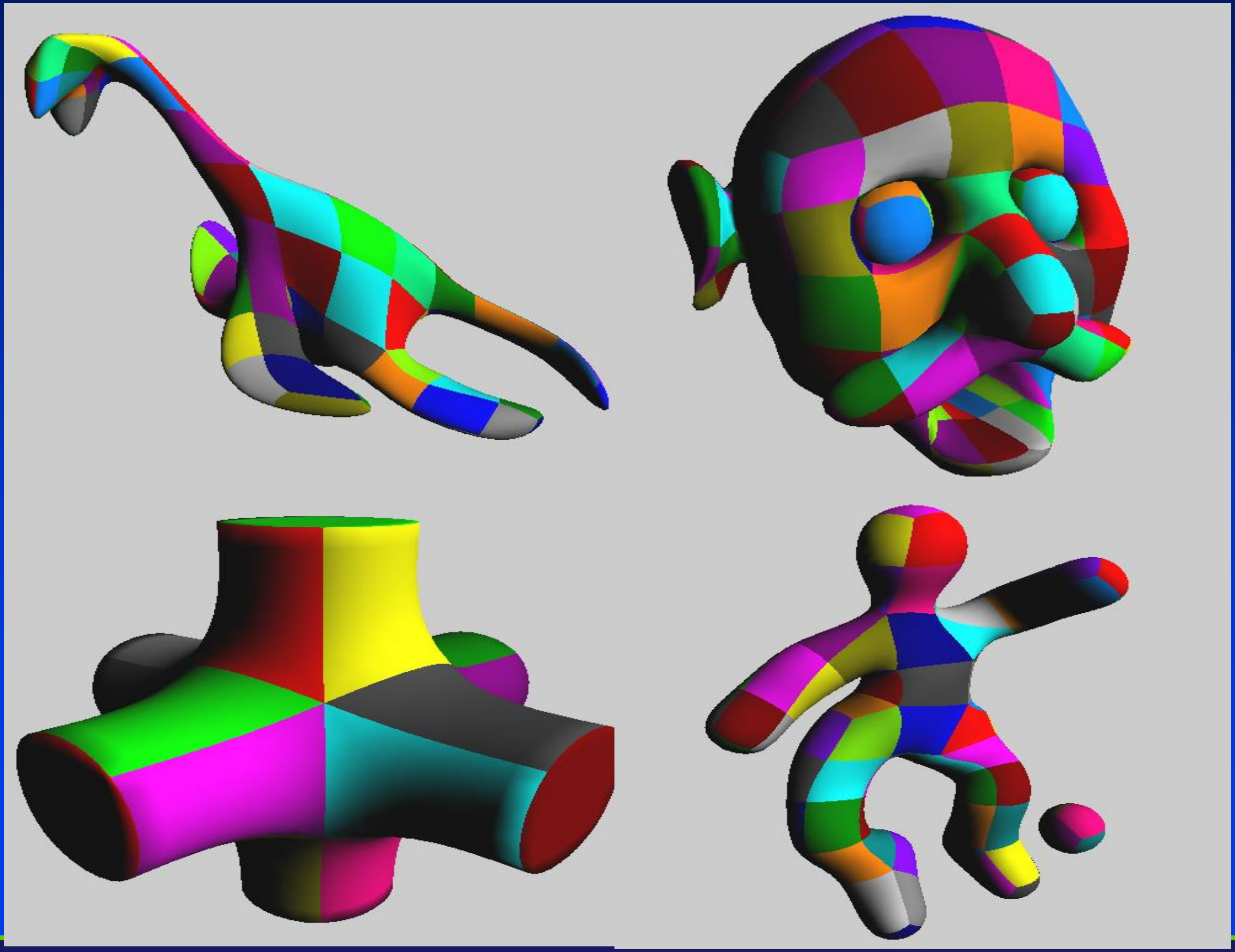
feature deformation



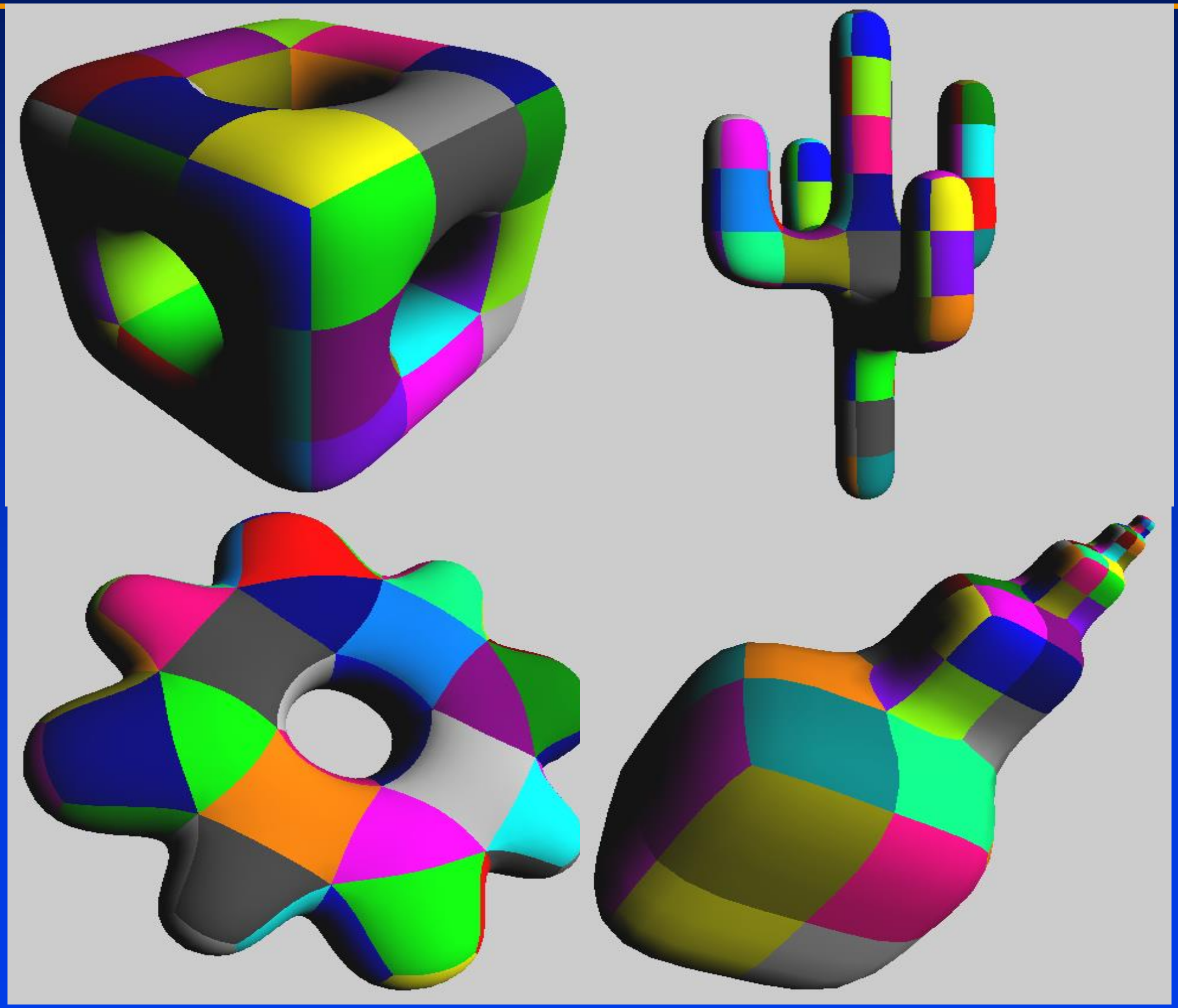
multi-face extrusion



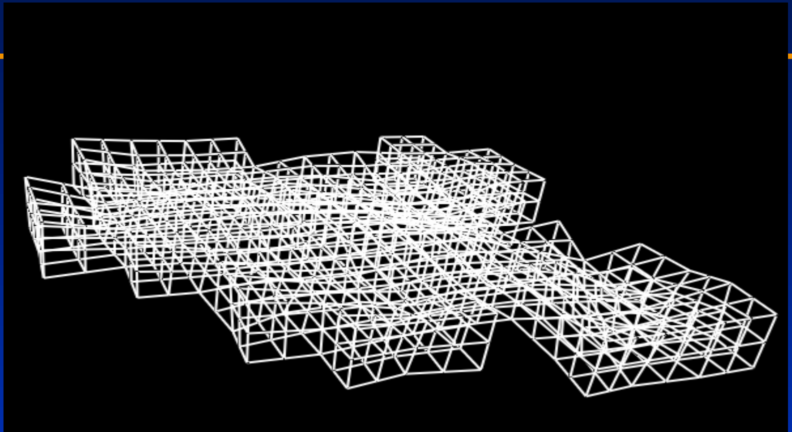
Interactive Sculpting



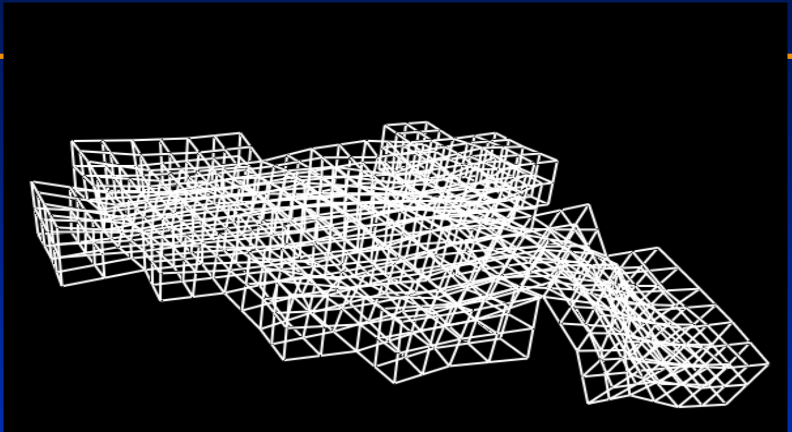
More Examples



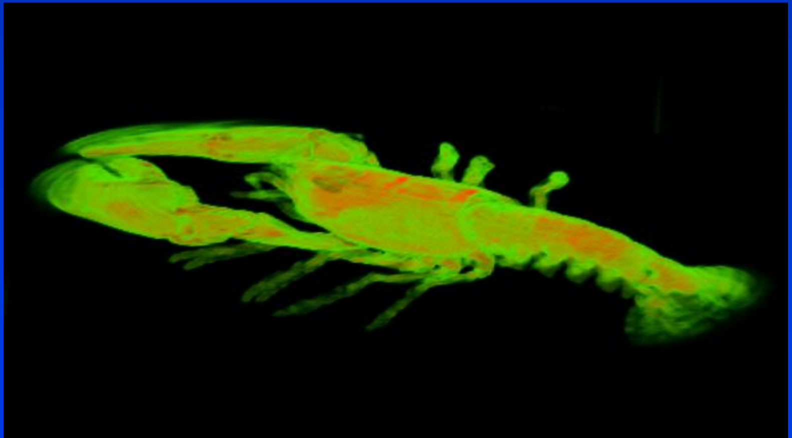
Volume Editing and Visualization



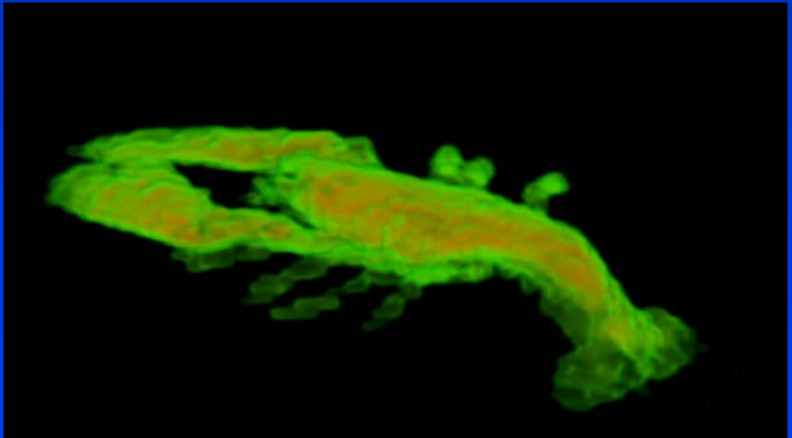
original lattice



deformed lattice

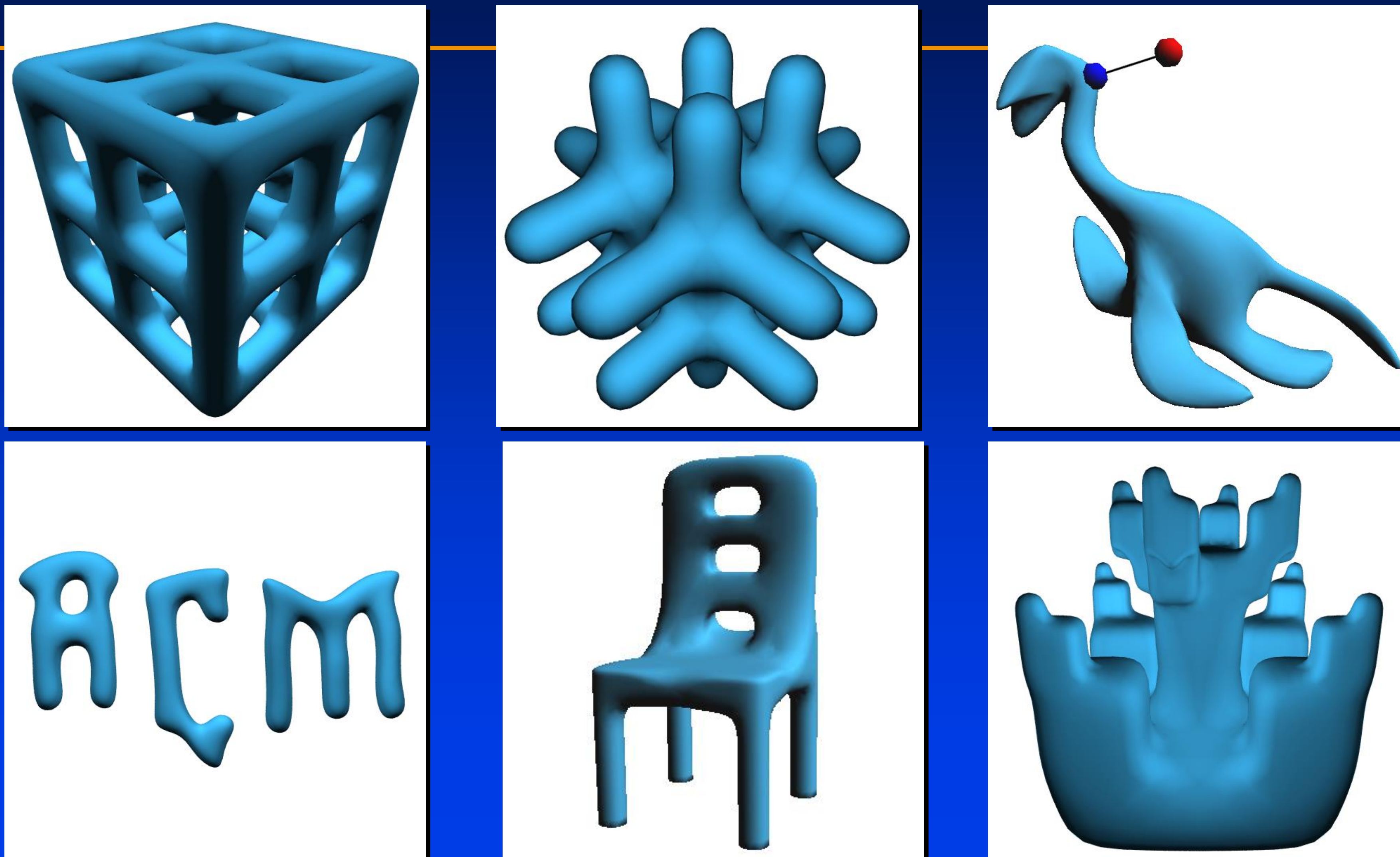


original volume



deformed volume

Sculpted CAD Models



Subdivision Solids

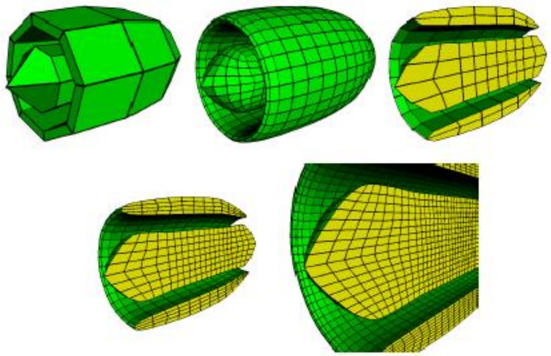
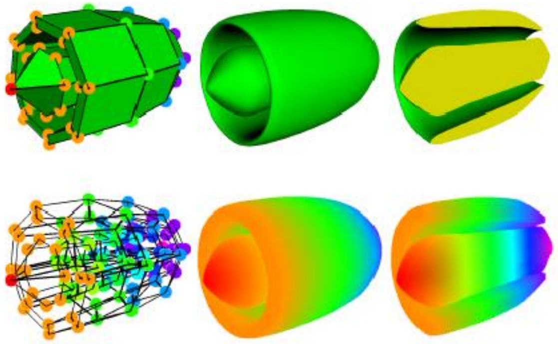


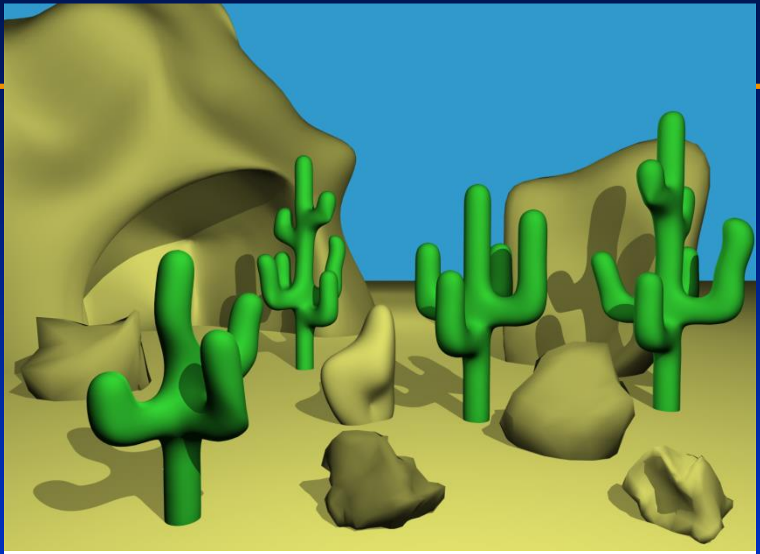
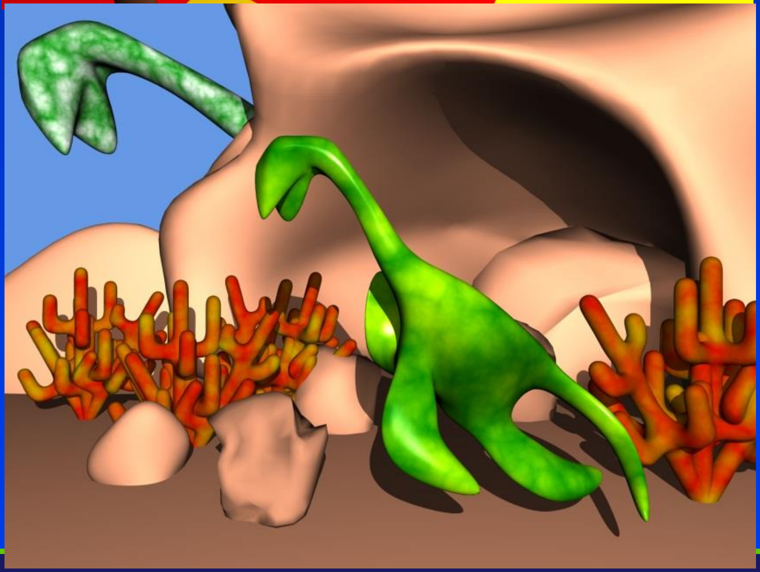
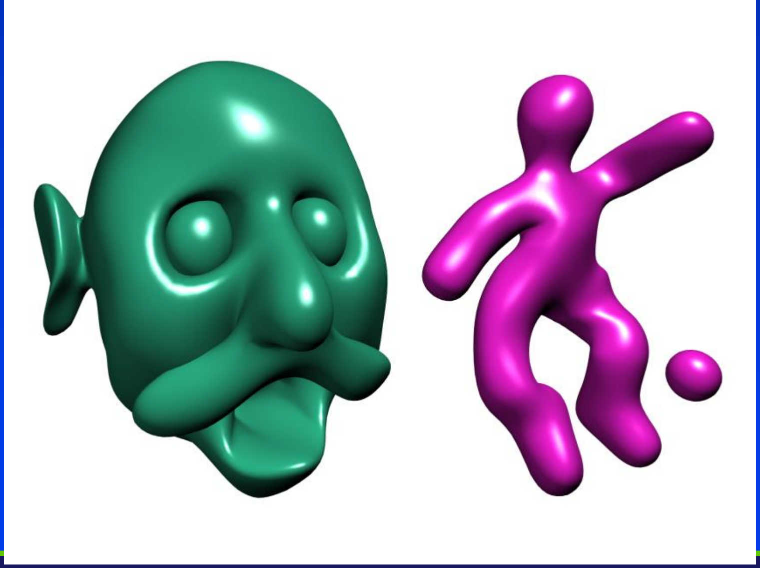
Fig. 22. Jet engine model comprised of two disconnected parts.



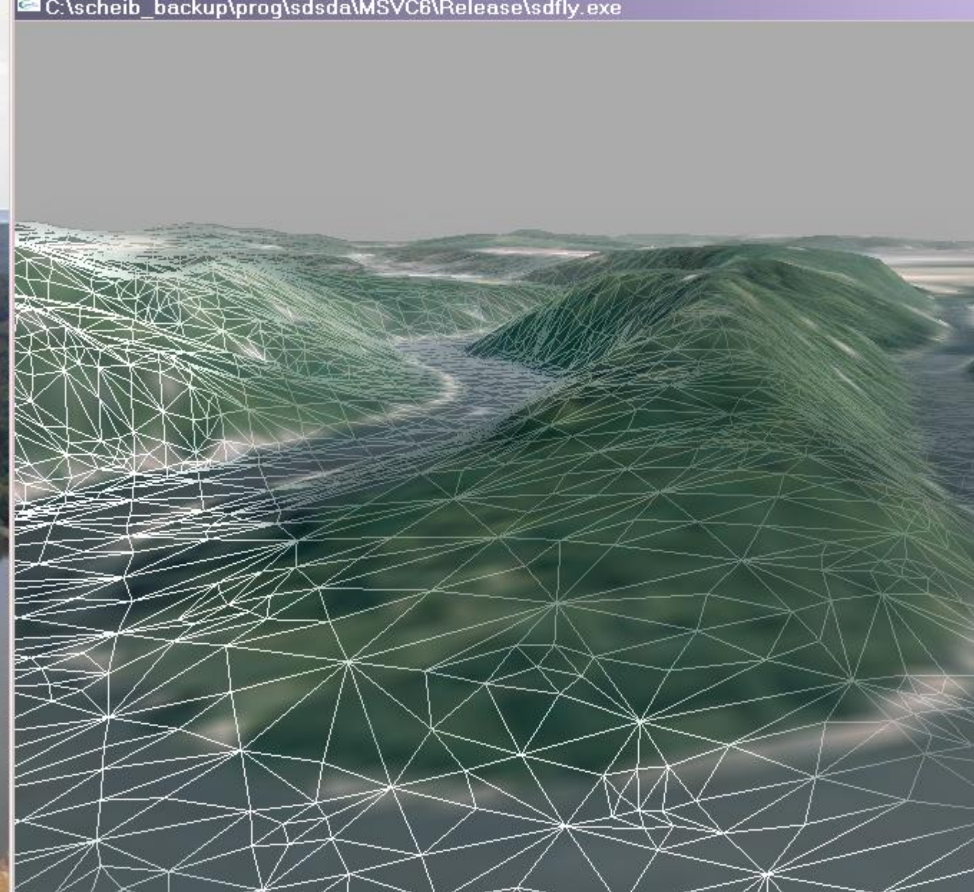
(a) (b) (c)

Fig. 23. Material properties can be interpolated smoothly throughout the entire volumetric domain. (a) Control mesh with color. (b-c) Model after three levels of subdivision.

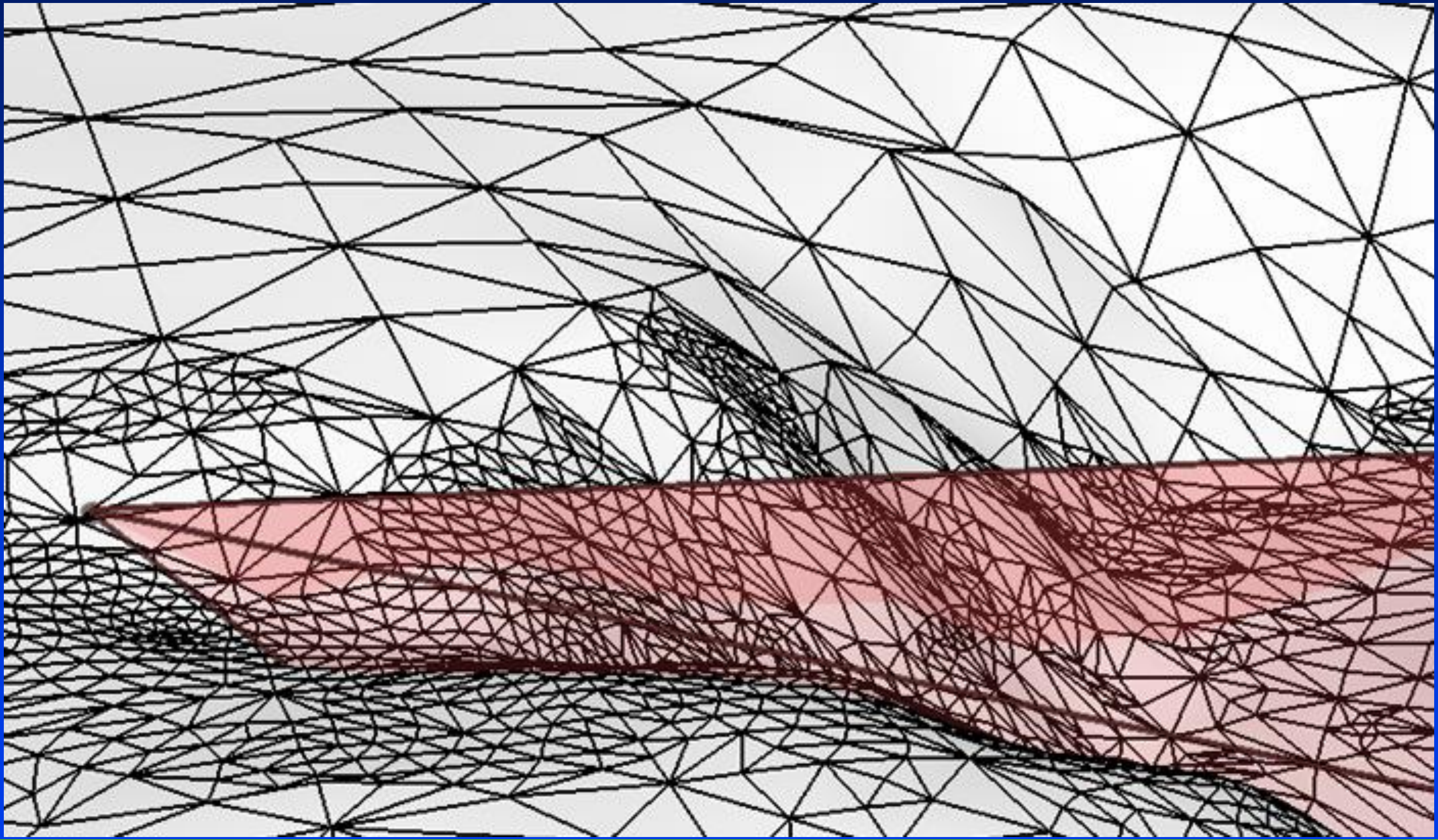
Scenes and Sculptures

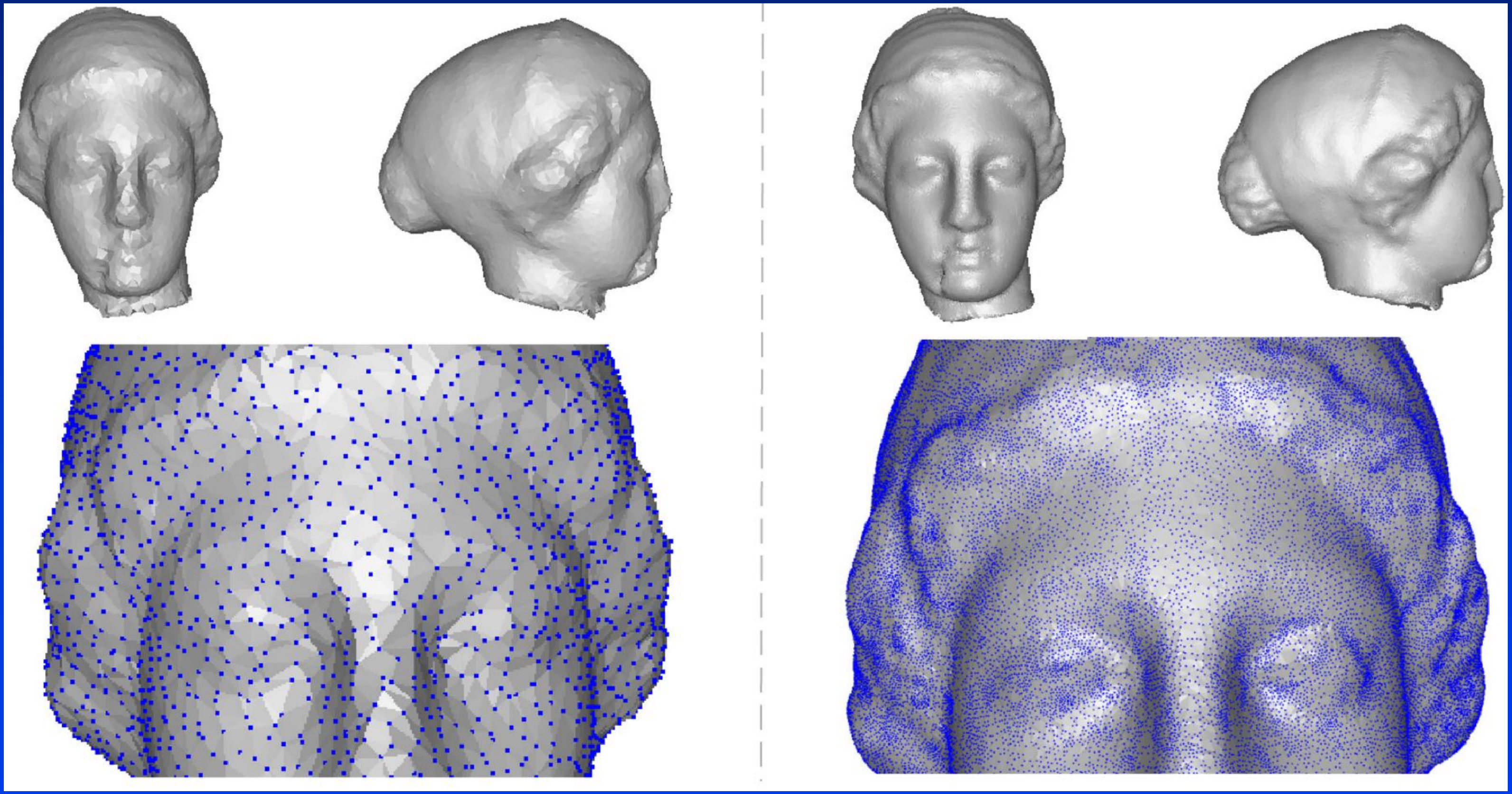
Other Applications


C:\scheib_backup\prog\sdsda\MSVC6\Release\sdfly.exe

Rendering – Adaptive Tessellation

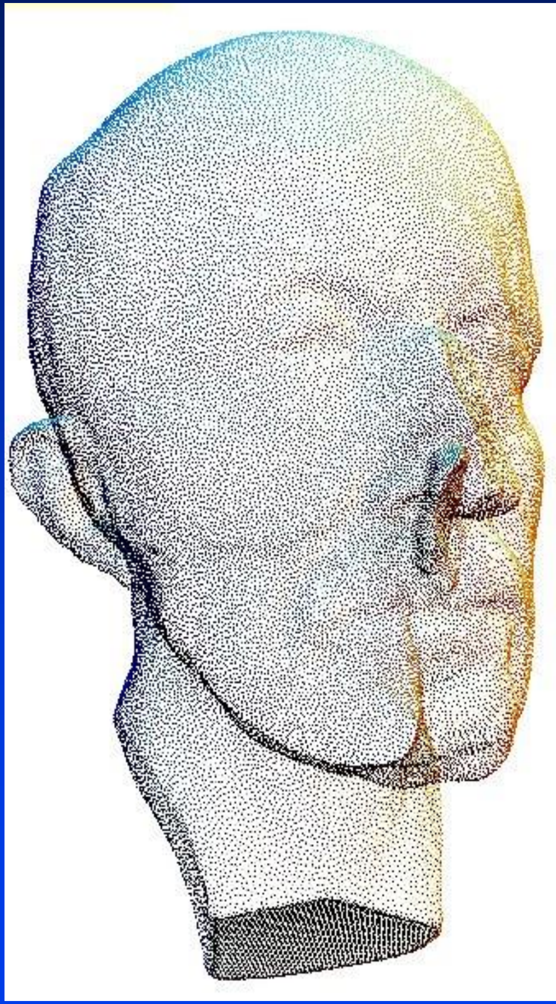


Meshless Geometric Subdivision



Point-based Graphics

- Core : unstructured point cloud
- Points with attributes :
  - color, normal, etc.
- Advantages :
  - acquisition
  - multiresolution
  - storage
- Drawback : **meshing + visualization**



Modeling + Visualization enabled by Subdivision Surface Fitting