3D Modeling

0368-3236, Spring 2019

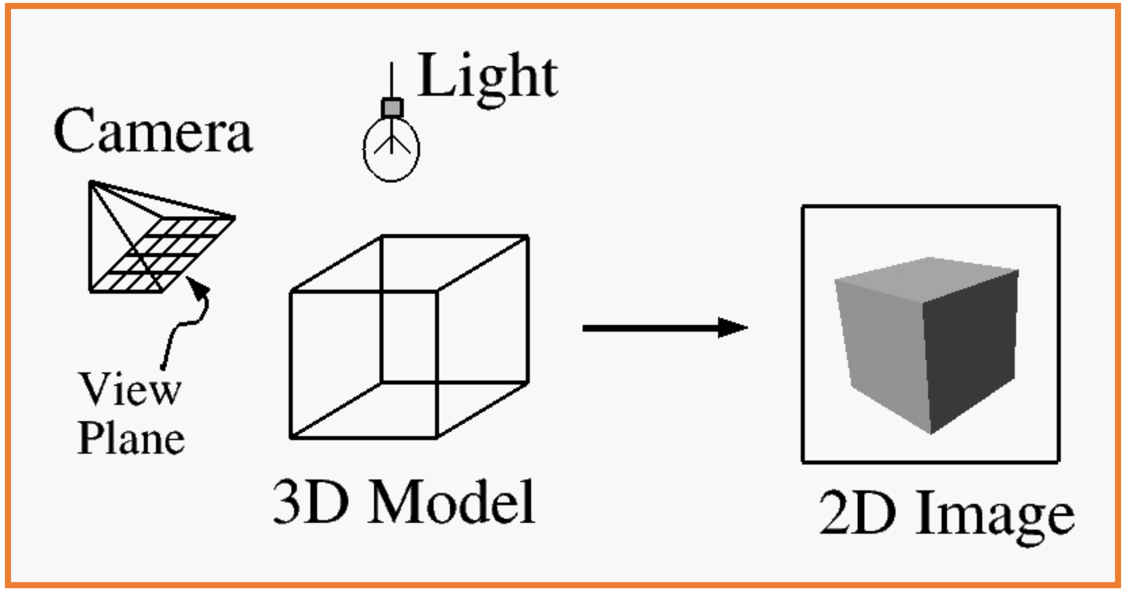
Tel-Aviv University

Amit Bermano

What is 3D Modeling?

Topics in computer graphics

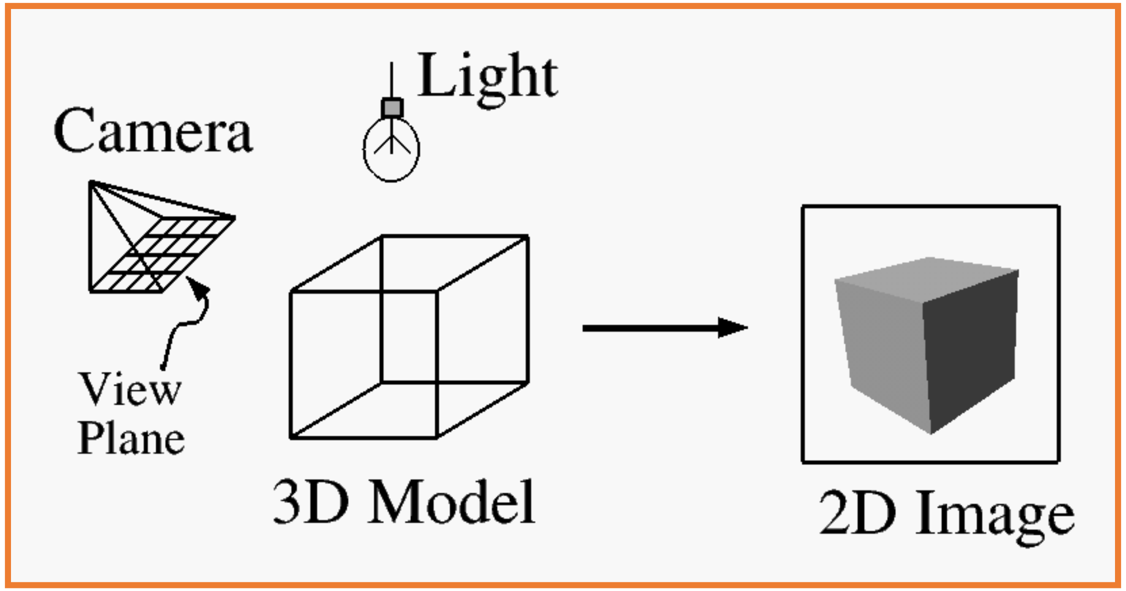
- Imaging = *representing & manipulating 2D images*
- Rendering = *constructing 2D images from 3D models*
- Modeling = *representing & manipulating 3D objects*
- Animation = *simulating changes over time*



What is 3D Modeling?

Topics in computer graphics

- Imaging = *representing & manipulating 2D images*
- Rendering = *constructing 2D images from 3D models*
- Modeling = *representing & manipulating 3D objects*
- Animation = *simulating changes over time*



Modeling

Blender demoreel 2018/2019

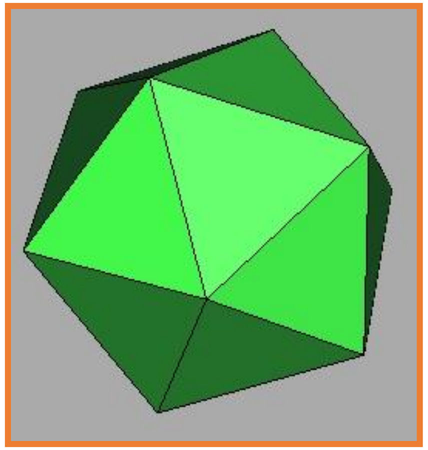
https://www.youtube.com/watch?v=HBwtw3J4mhA

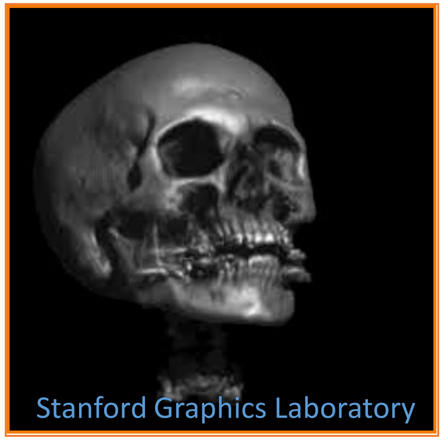
Blender

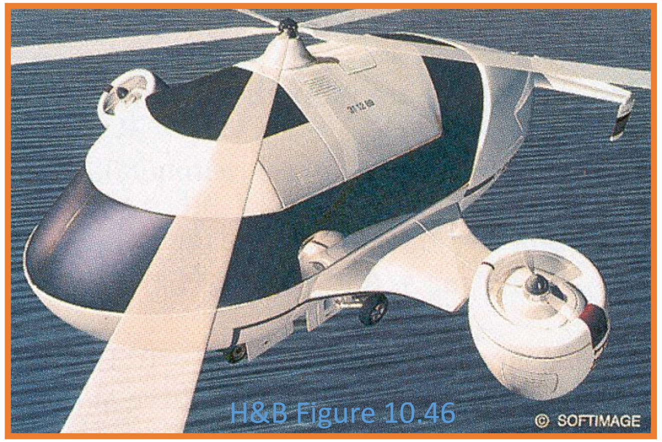
Modeling

How do we ...

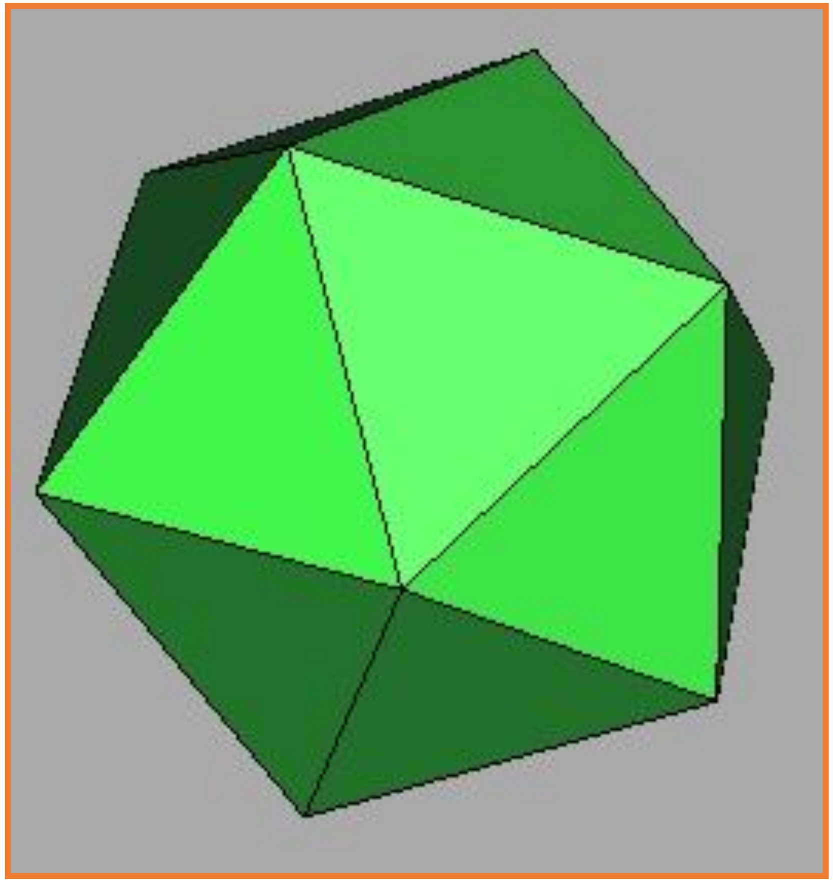
- Represent 3D objects in a computer?
- Acquire computer representations of 3D objects?
- Manipulate computer representations of 3D objects?





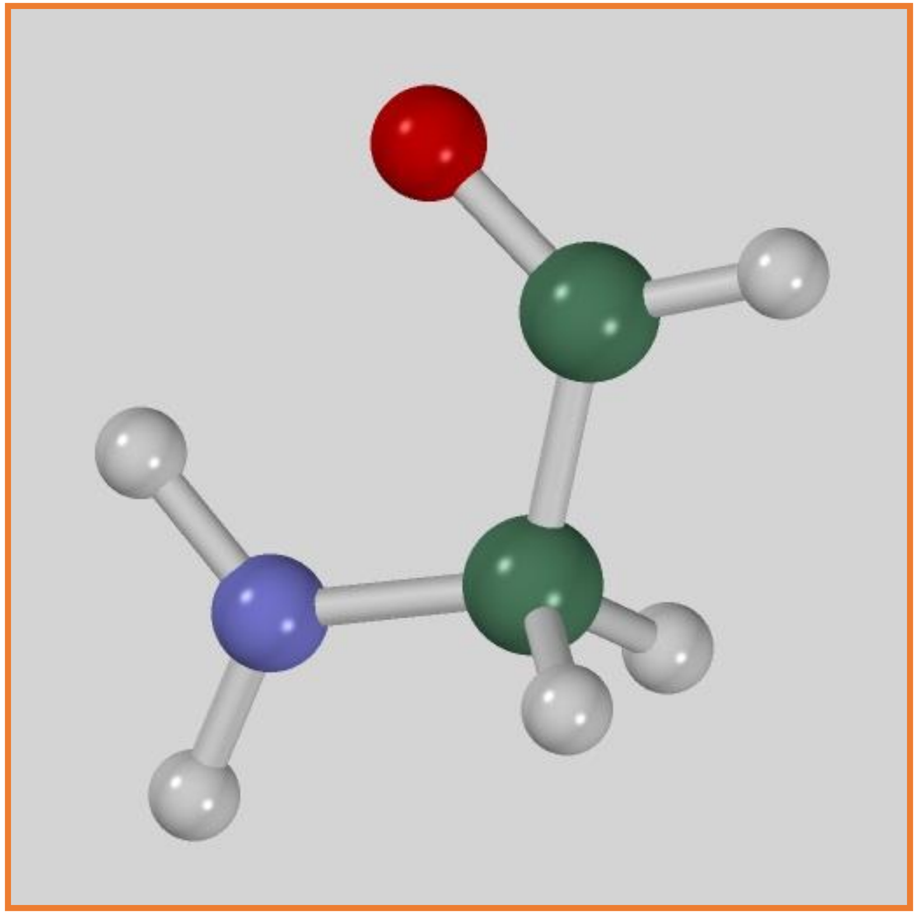


3D Object Representations



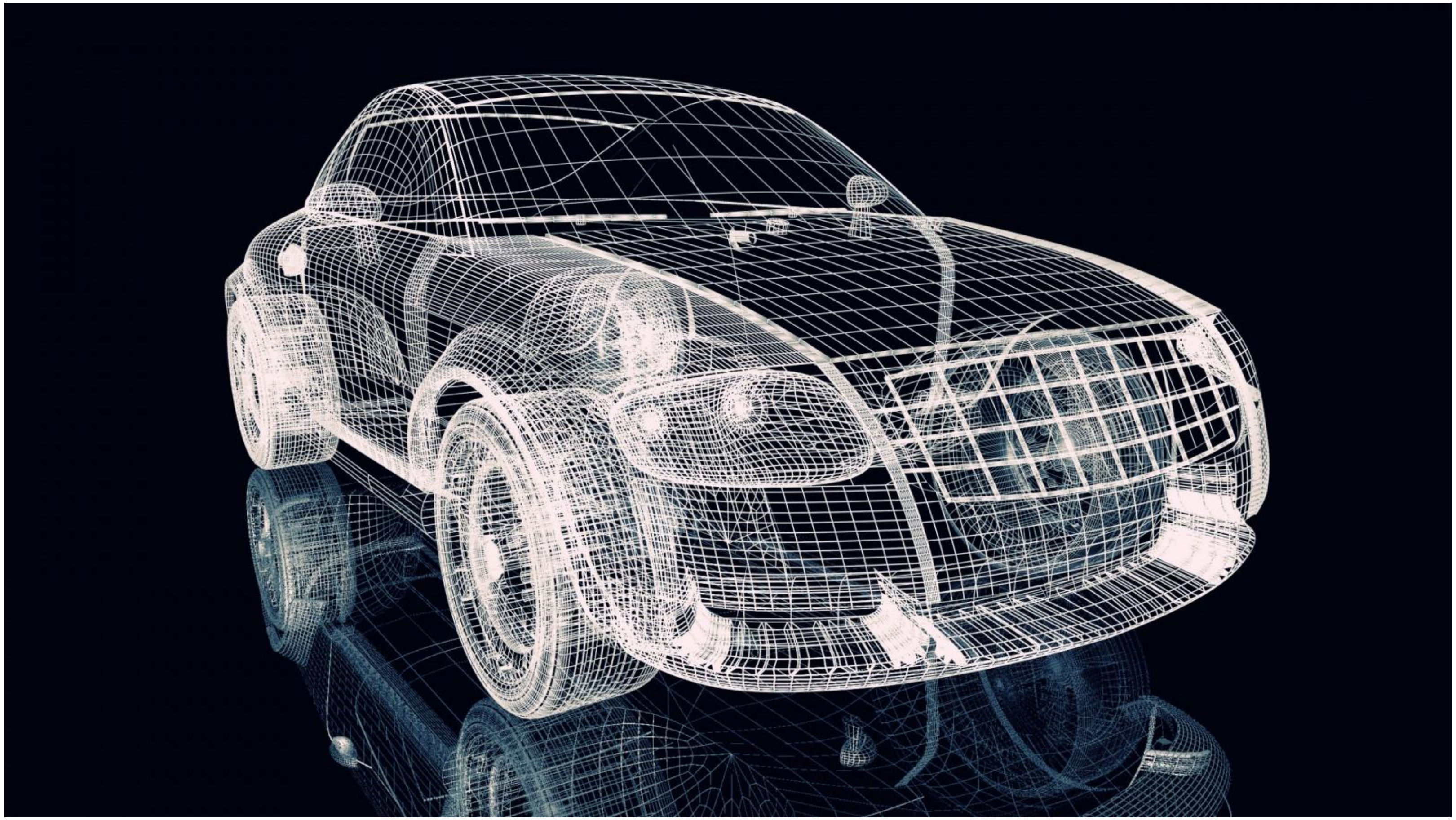
How can this object be represented in a computer?

3D Object Representations



How about this one?

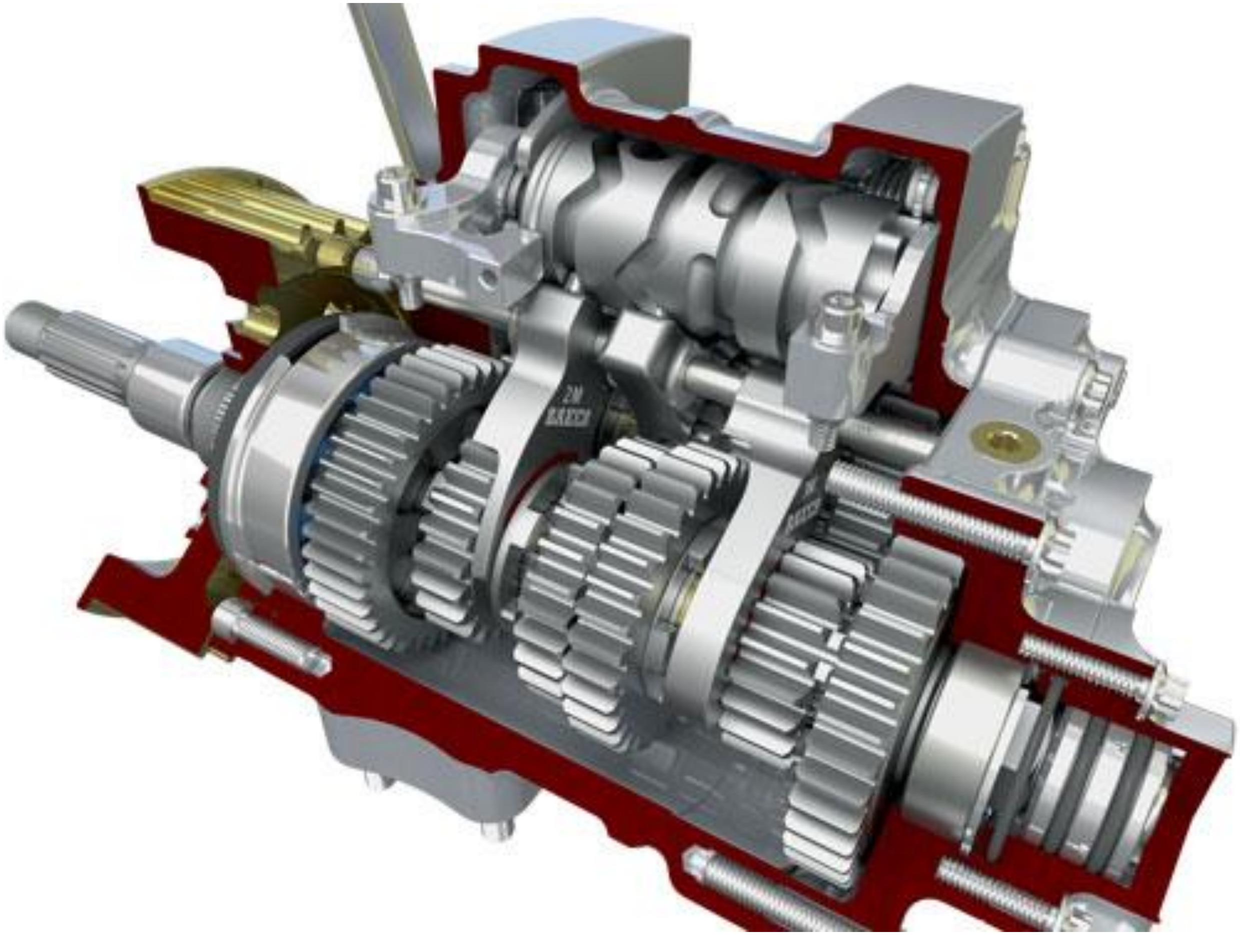
3D Object Representations



Wallpapersonly.net

And this one?

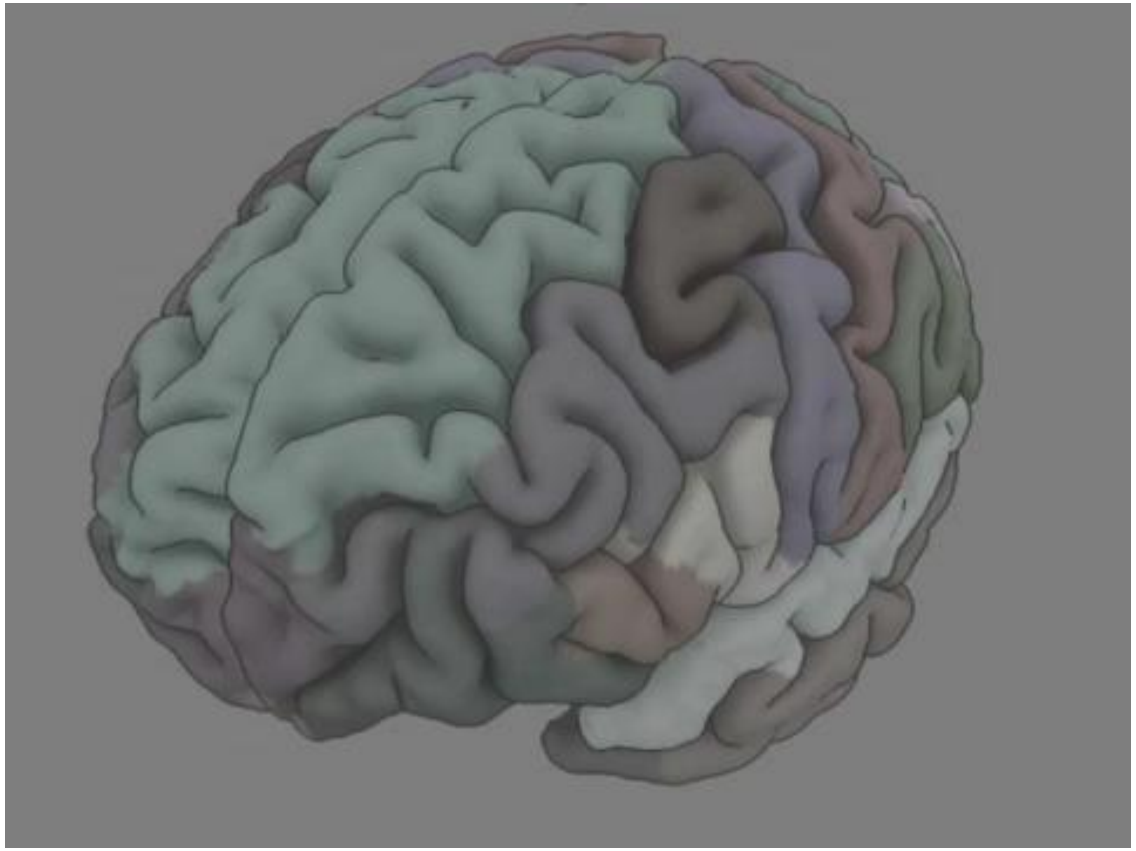
3D Object Representations



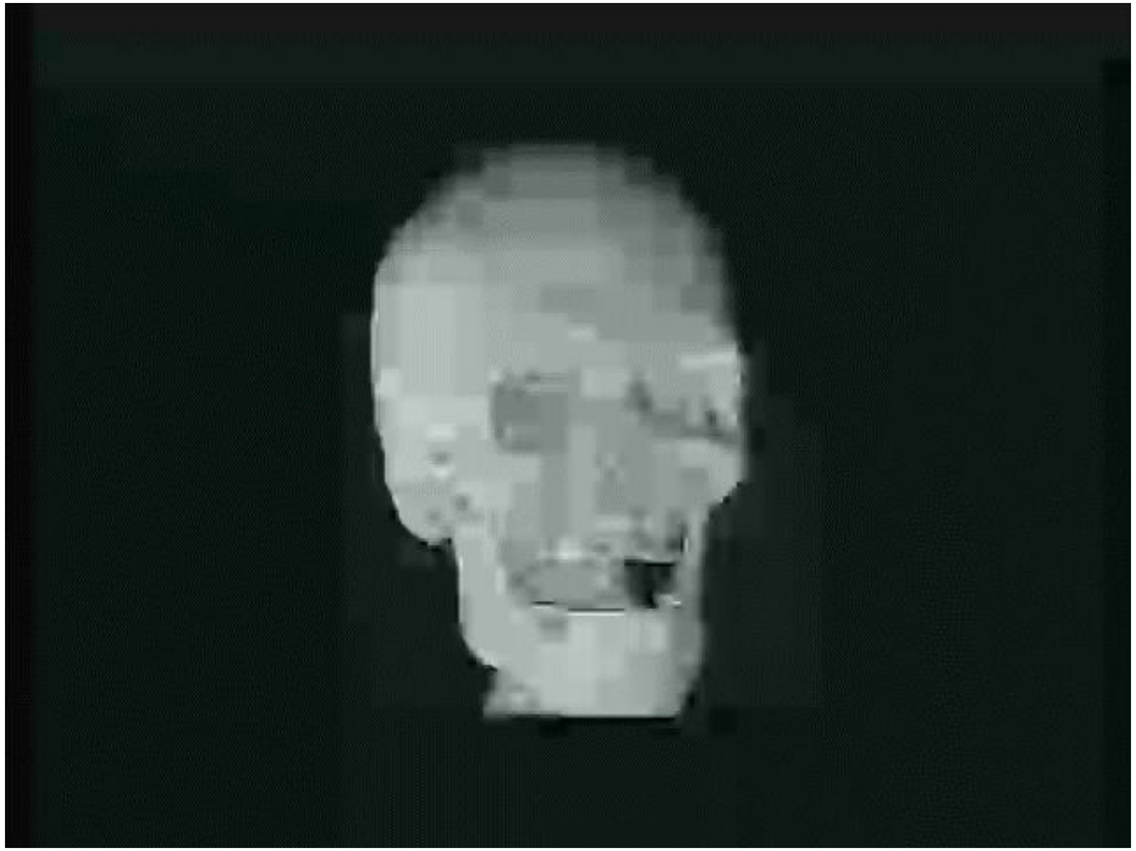
Solidworks

This one?

3D Object Representations



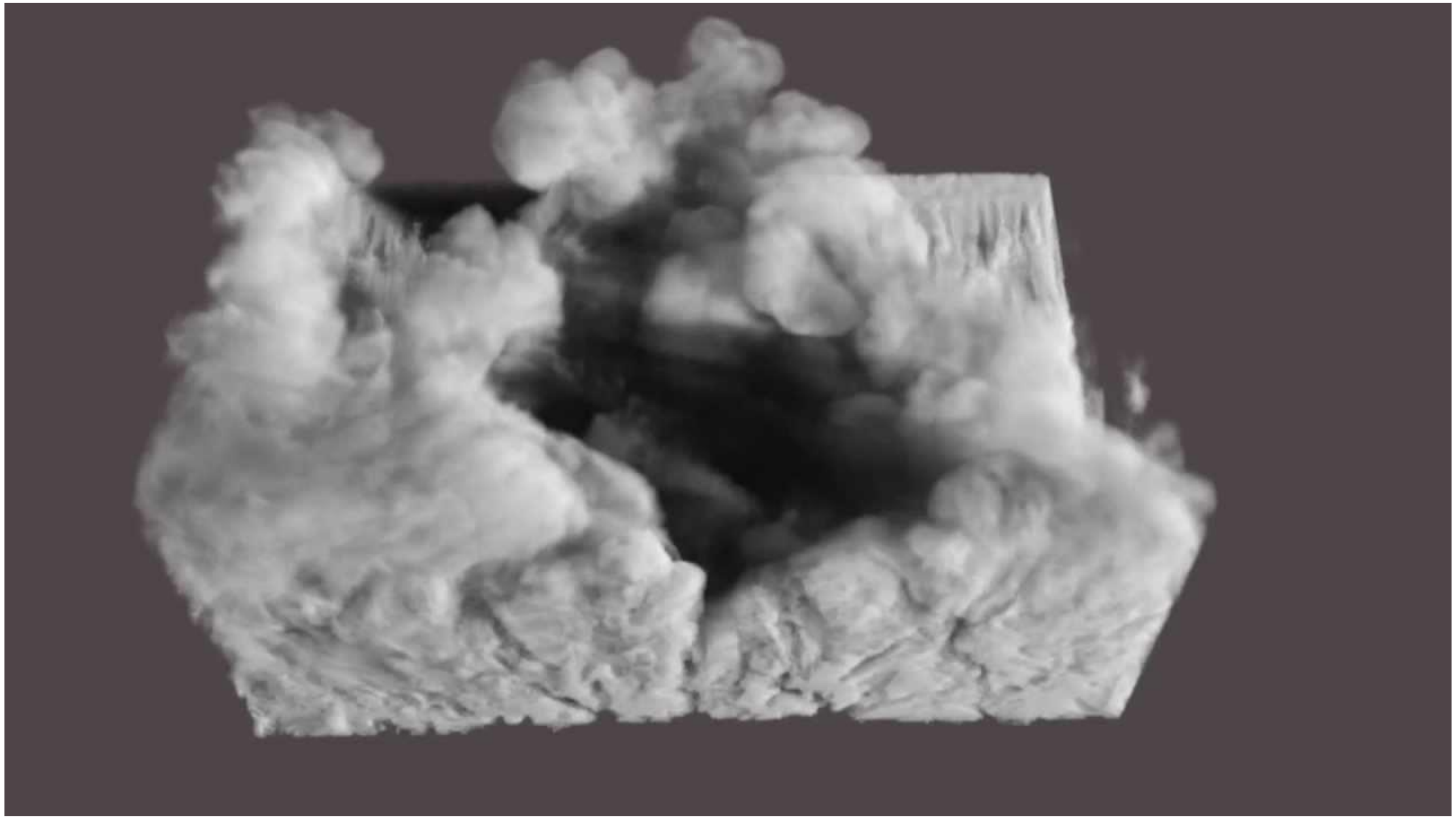
https://www.youtube.com/watch?v=0gmJYugUJU8



The visible human

This one?

3D Object Representations



FumeFx

This one?

3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

Equivalence of Representations

Thesis:

- Each representation has enough expressive power to model the shape of any geometric object
- It is possible to perform all geometric operations with any fundamental representation

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation

Data structures determine algorithms

Why Different Representations?

Efficiency for different tasks

- Acquisition
  - Computer Vision
- Rendering
- Analysis
- Manipulation
- Animation

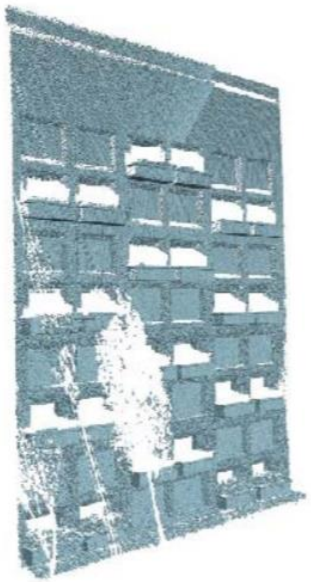
USC

Indiana University

Why Different Representations?

Efficiency for different tasks

- Acquisition
  - Range Scanning
- Rendering
- Analysis
- Manipulation
- Animation

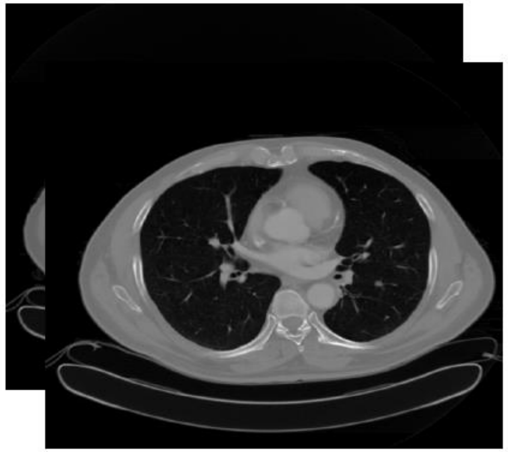


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
  - Tomography
- Rendering
- Analysis
- Manipulation
- Animation



DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

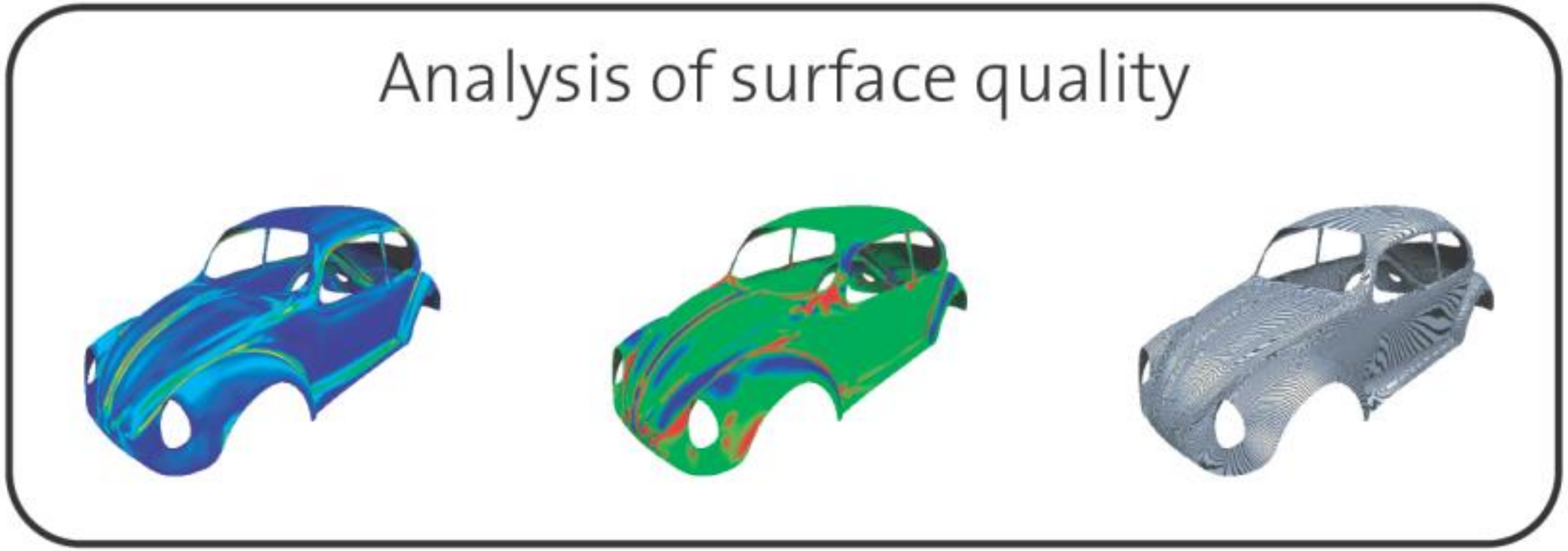
- Acquisition
- Rendering
  - Intersection
- Analysis
- Manipulation
- Animation

Autodesk

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Curvature, smoothness
- Manipulation
- Animation

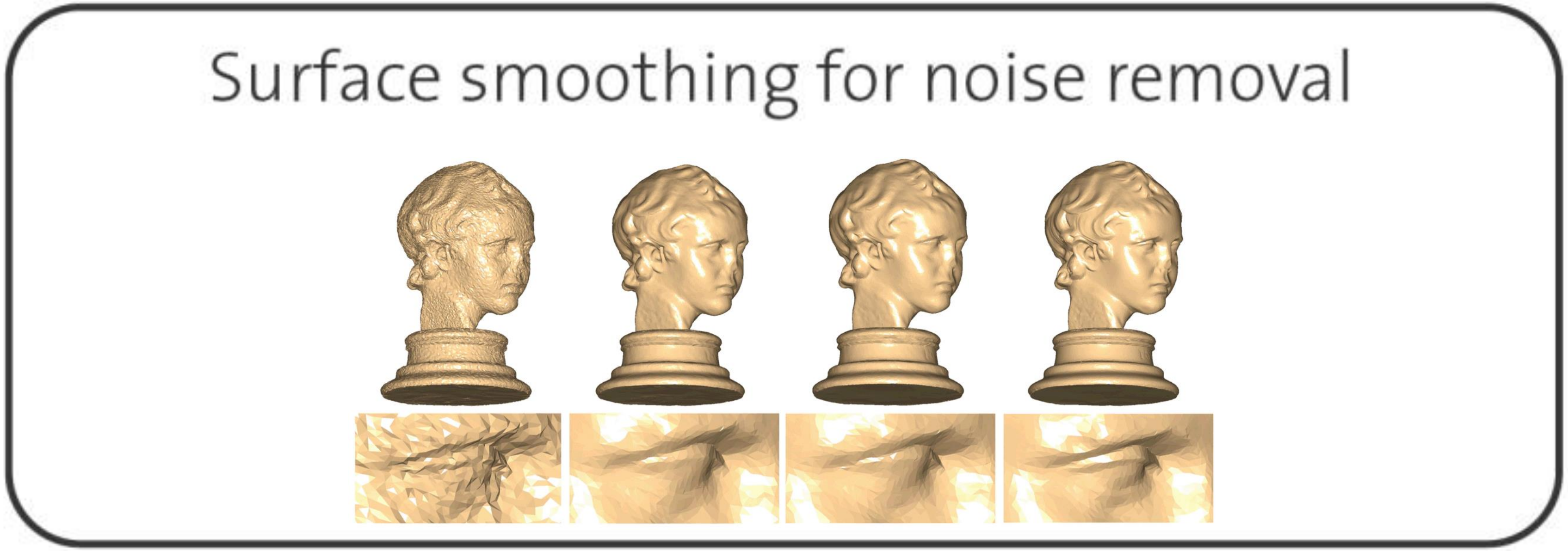


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Fairing
- Manipulation
- Animation

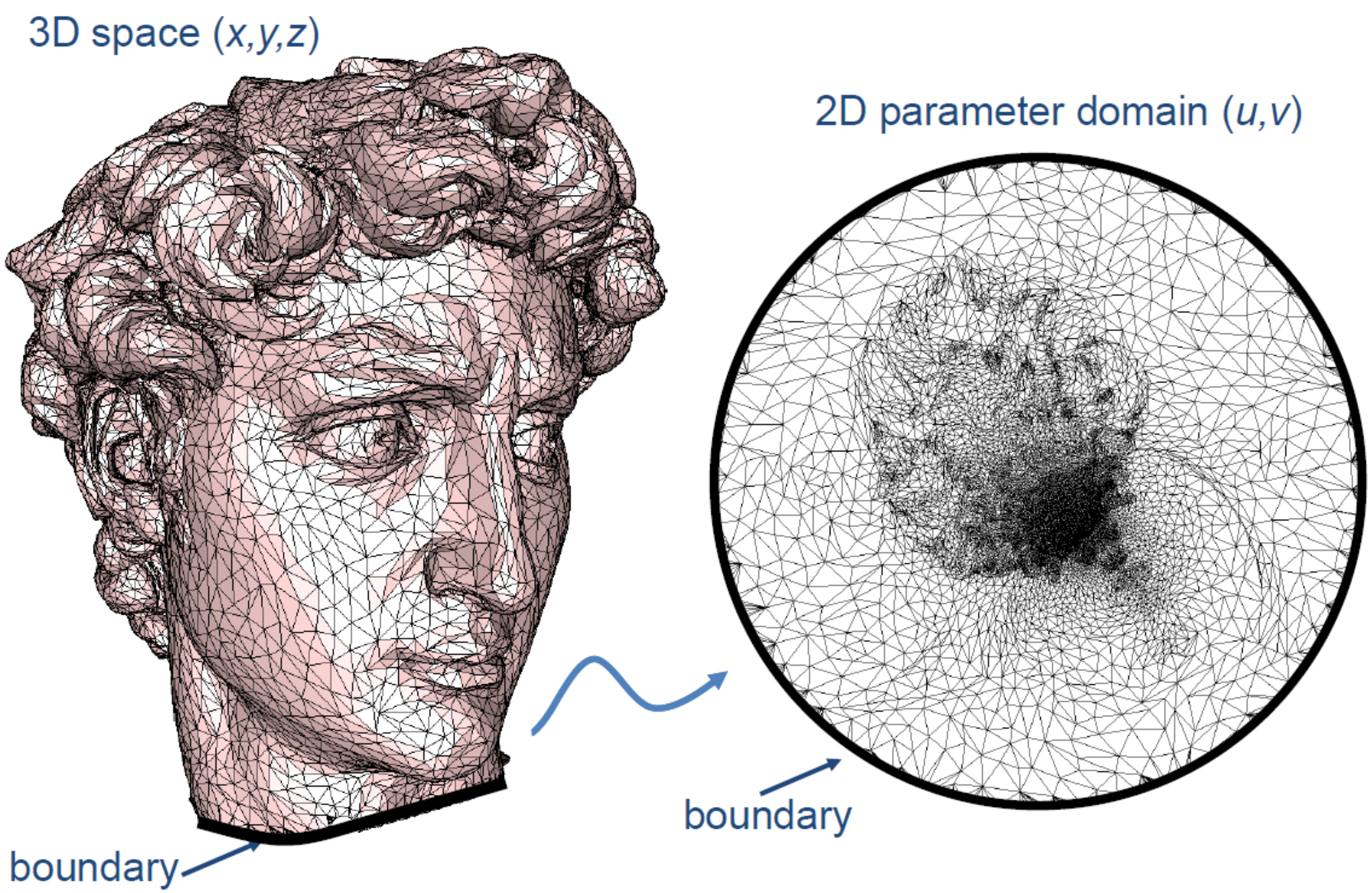


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Parametrization
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

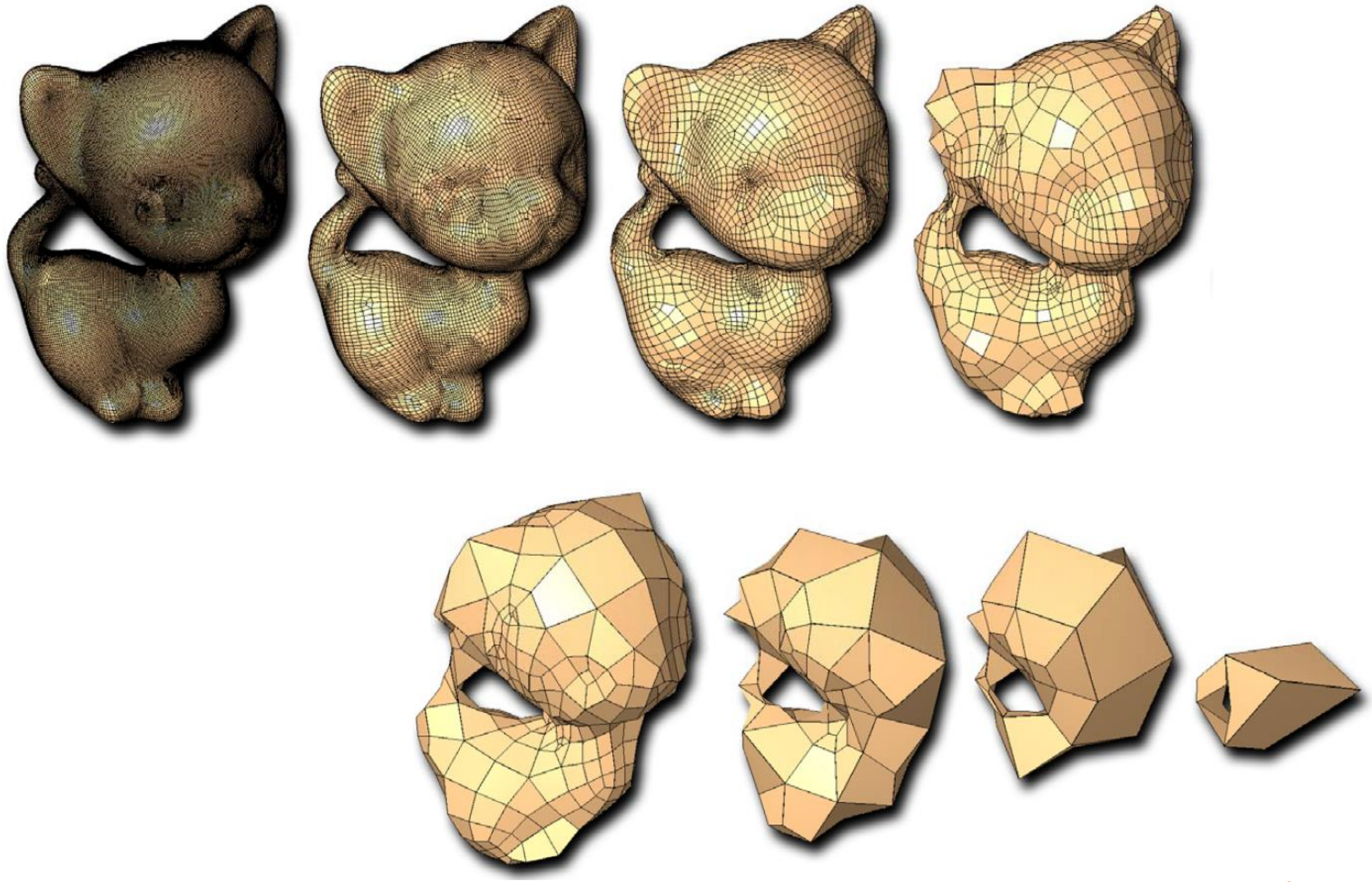
- Acquisition
- Rendering
- Analysis
  - Texture mapping
- Manipulation
- Animation

DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

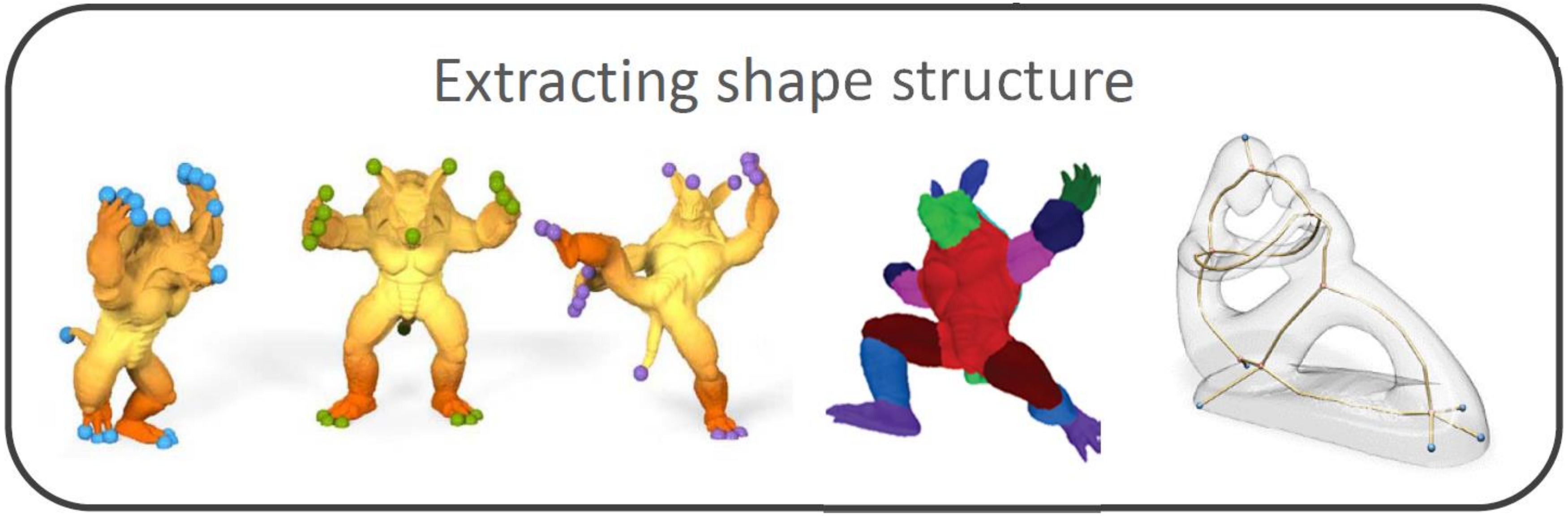
- Acquisition
- Rendering
- Analysis
  - Reduction
- Manipulation
- Animation



Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Structure
- Manipulation
- Animation



DGP course notes, Technion

Why Different Representations?

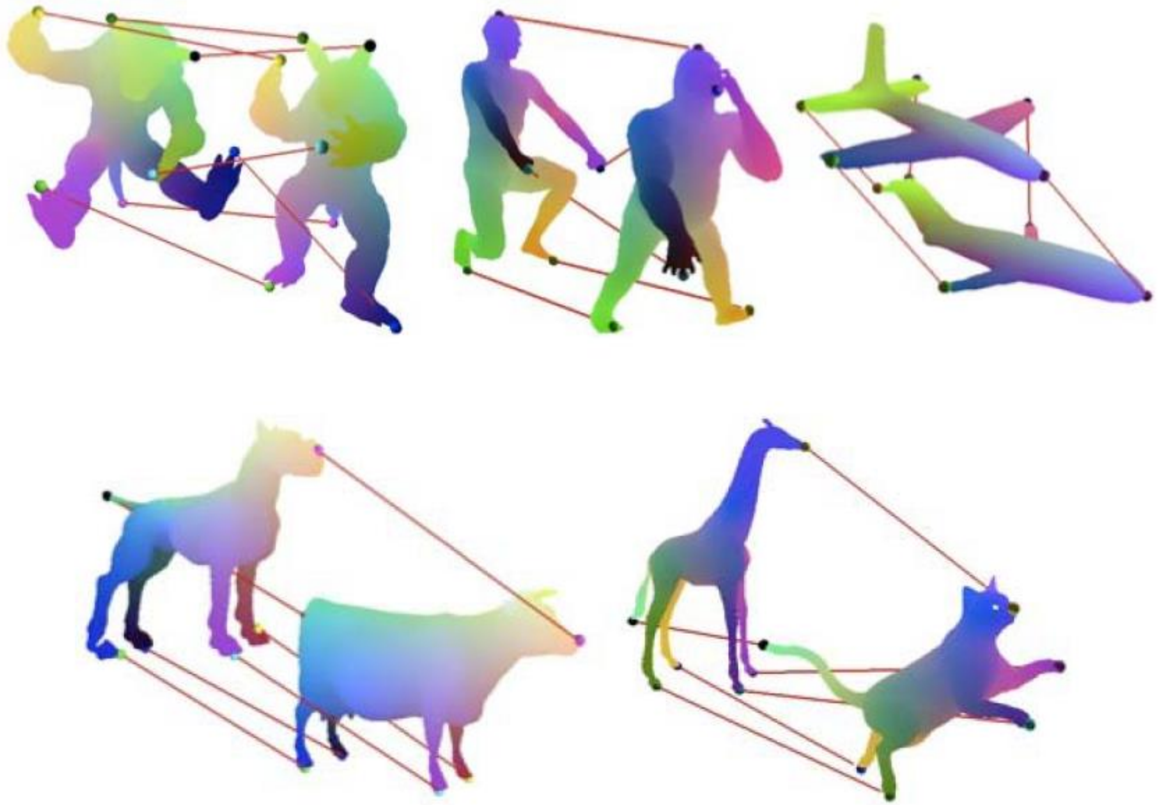
Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Symmetry detection
- Manipulation
- Animation

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Correspondence
- Manipulation
- Animation

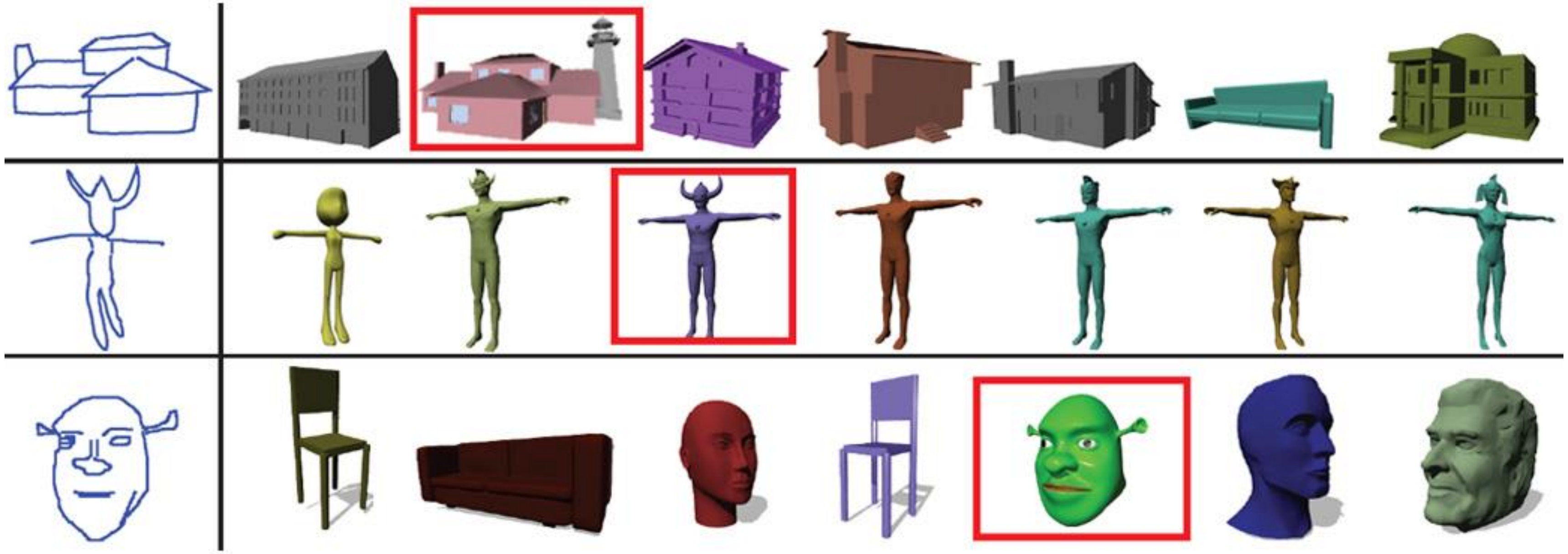


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Shape retrieval
- Manipulation
- Animation

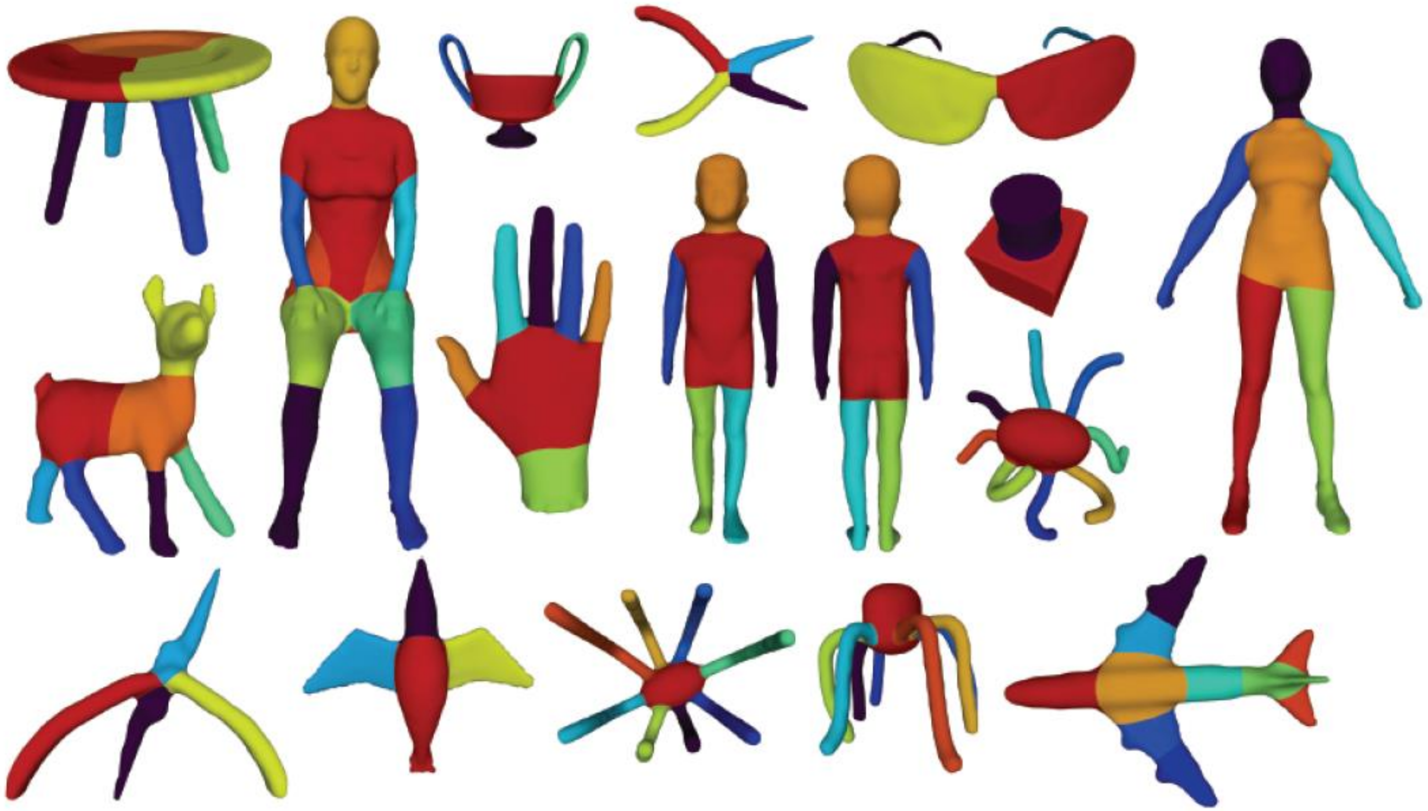


Shao et al. 2011

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Segmentation
- Manipulation
- Animation

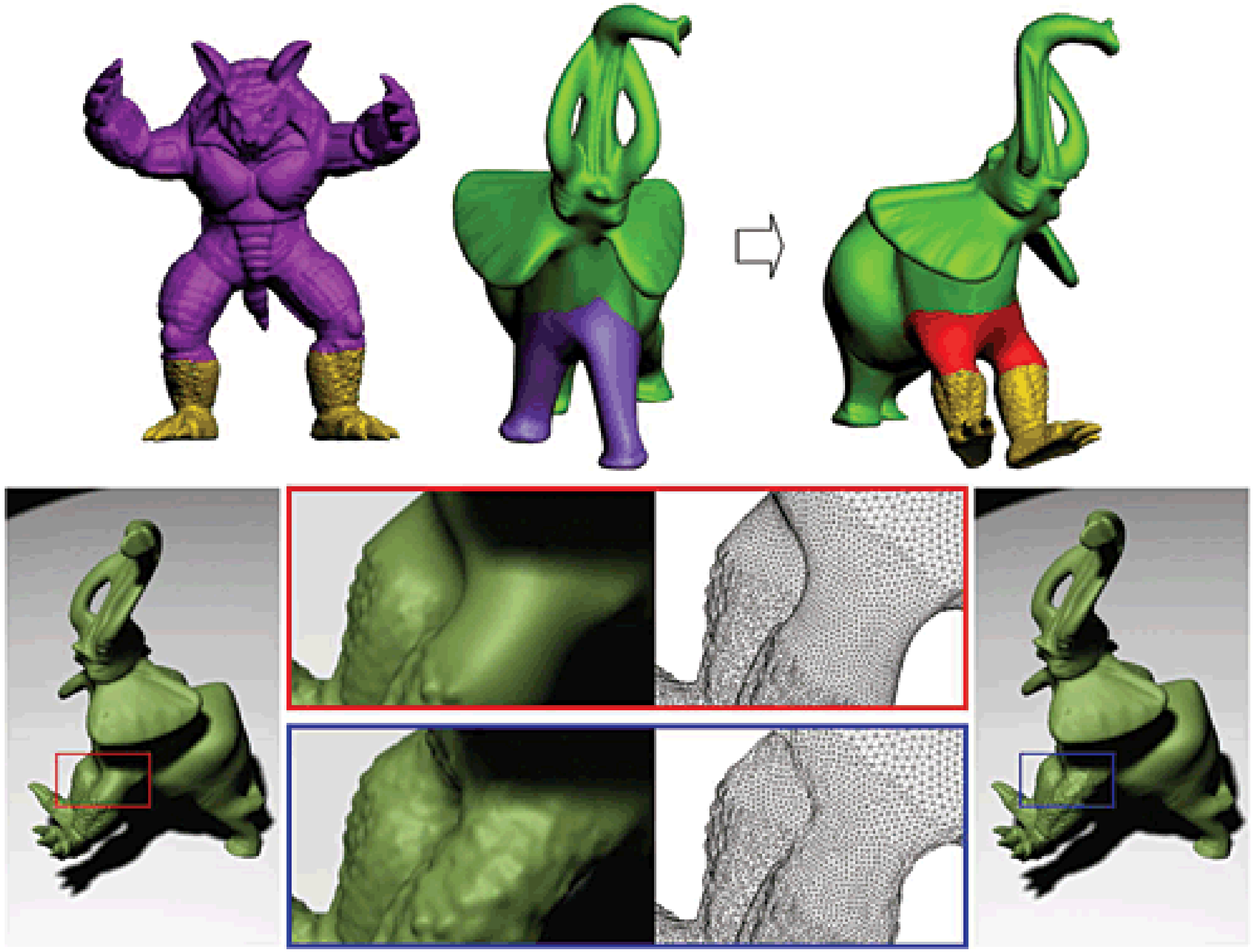


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Composition
- Manipulation
- Animation

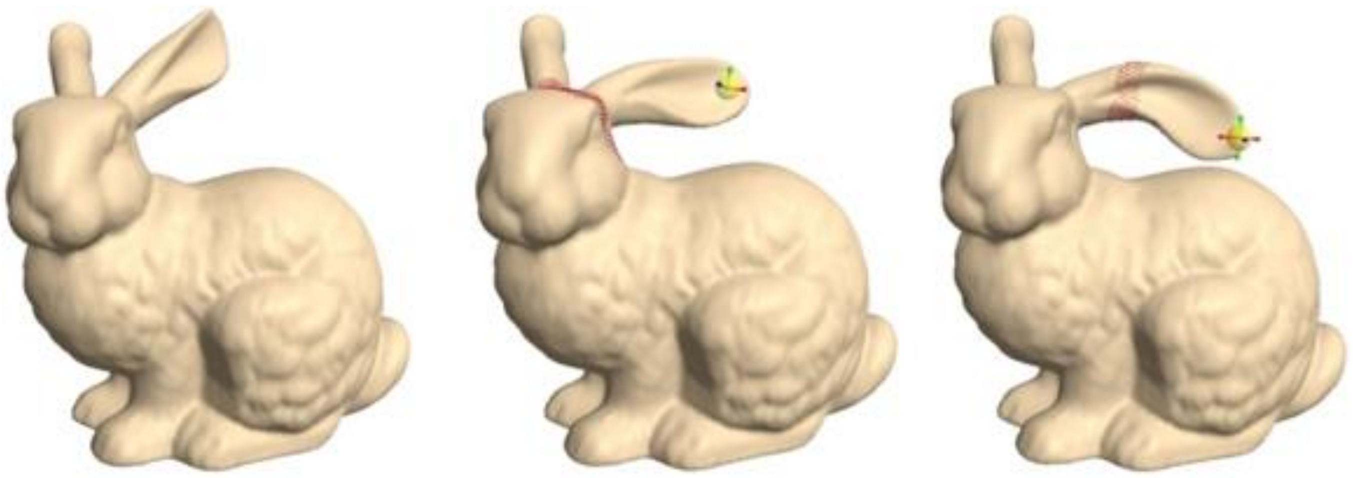


Lin et al. 2008

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
  - Deformation
- Animation

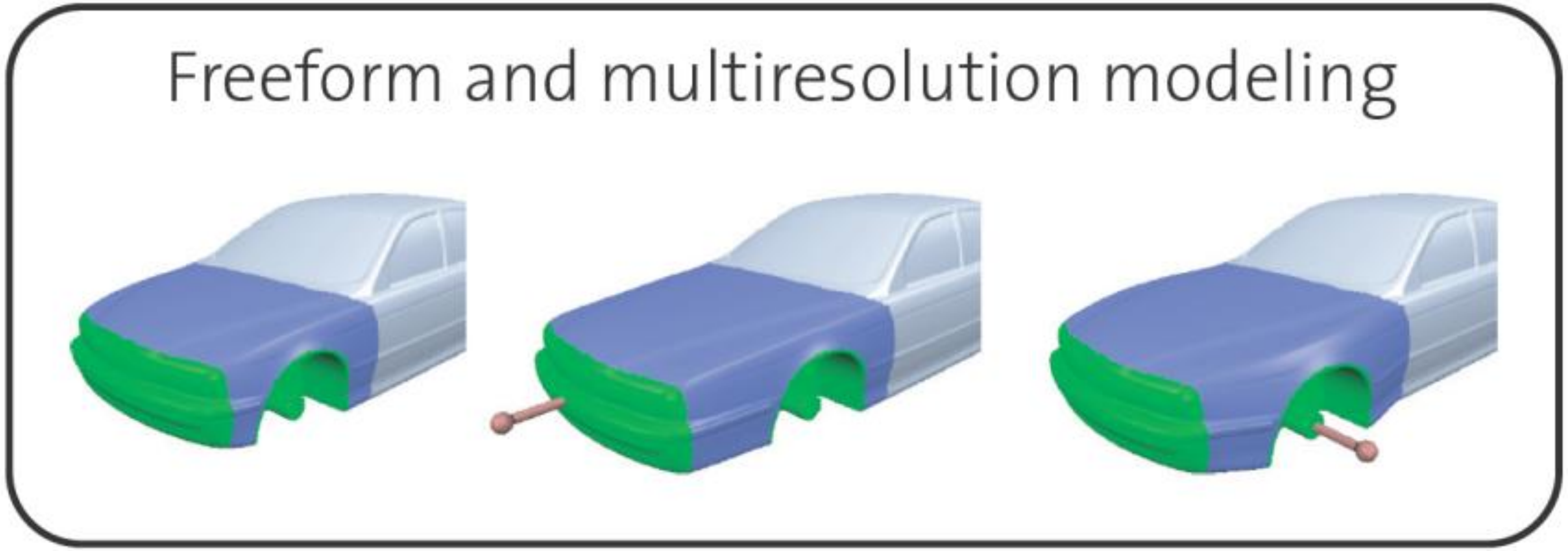


IGL

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
    - Deformation
- Animation

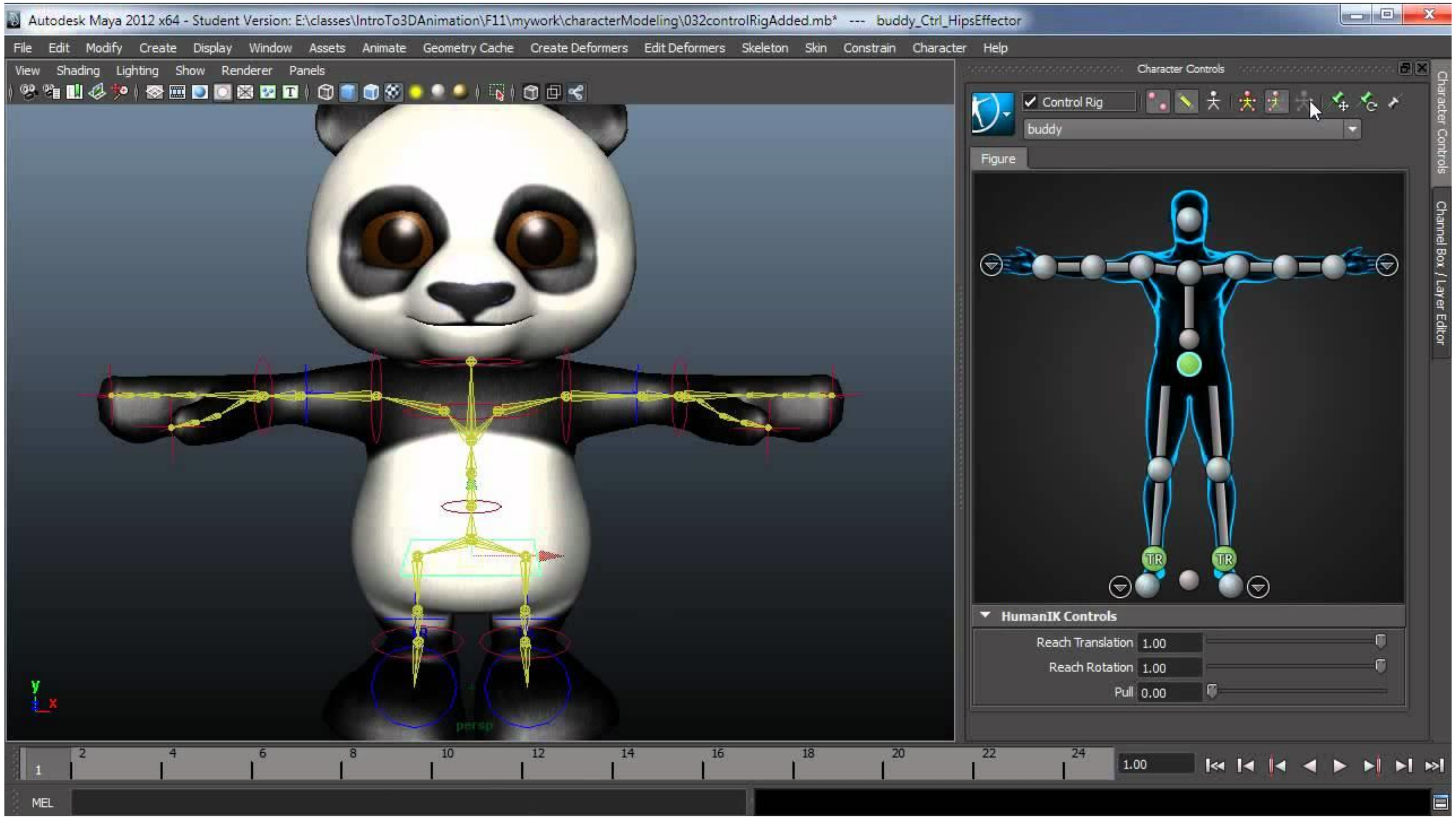


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

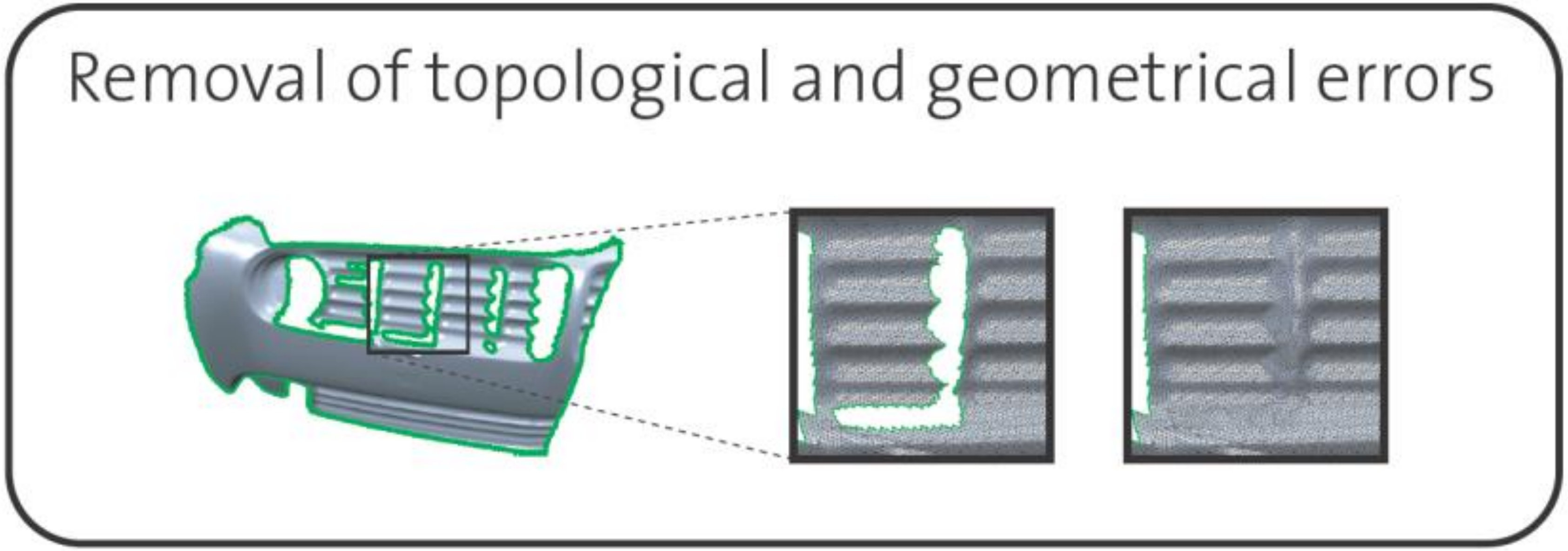
- Acquisition
- Rendering
- Analysis
- Manipulation
  - Control
- Animation



Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
  - Healing
- Animation

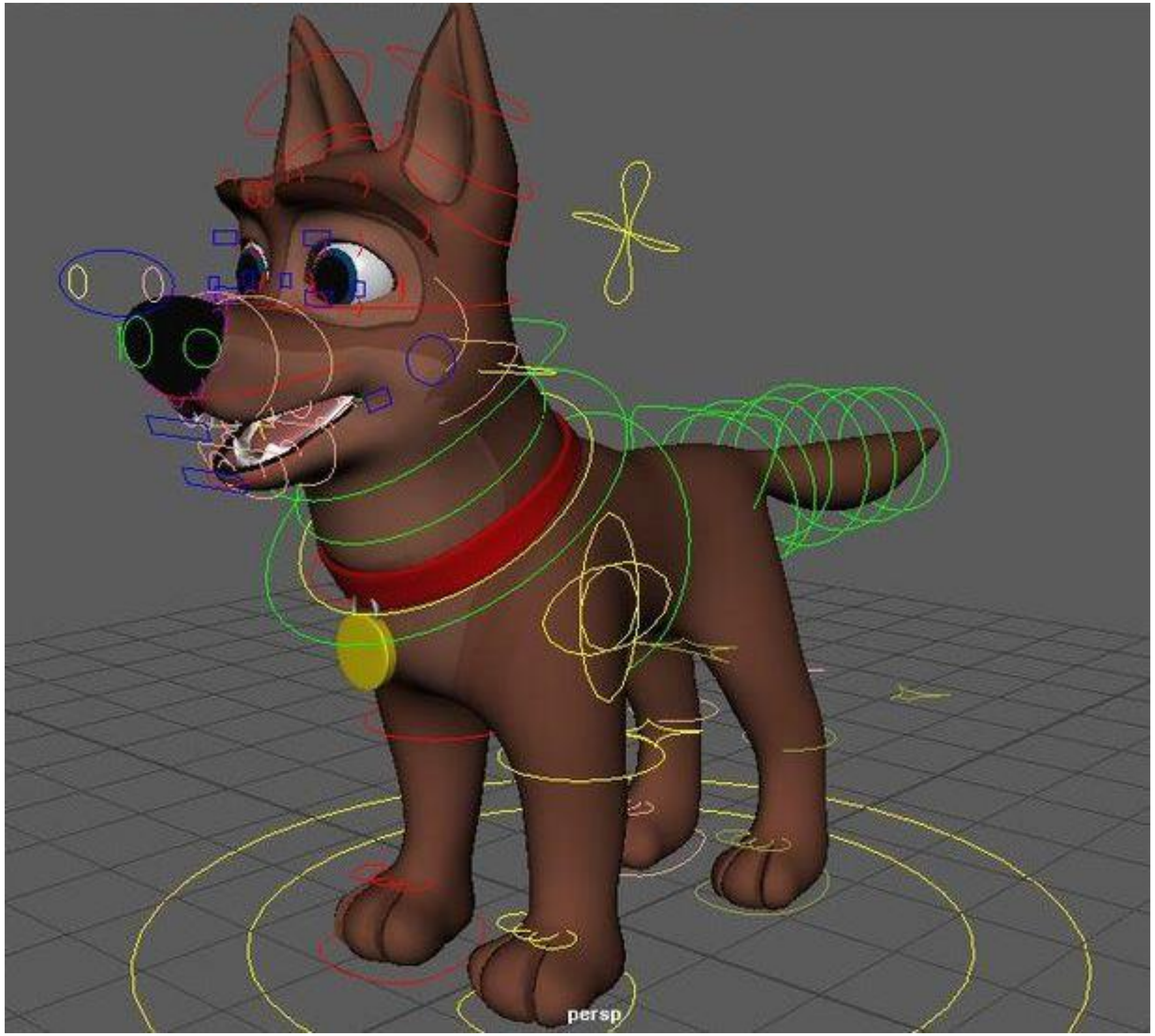


DGP course notes, Technion

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation
  - Rigging

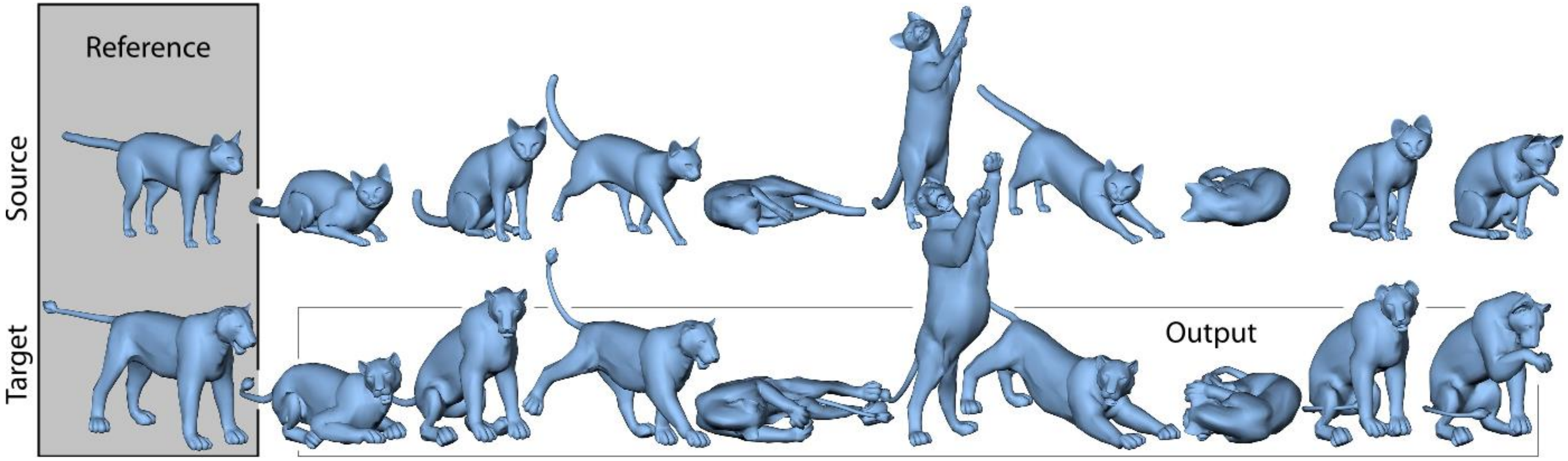


Animation Buffet

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation
    - Deformation transfer

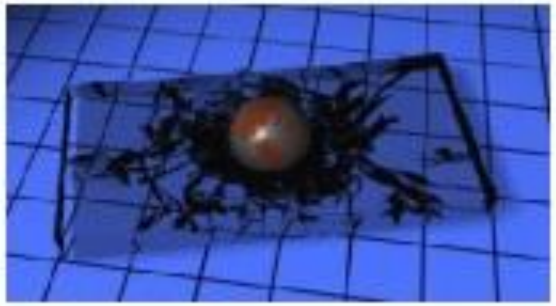
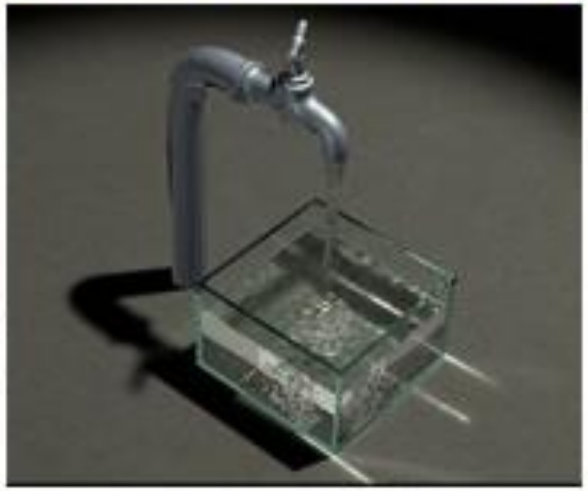


Sumner et al. 2004

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation
    - Simulation

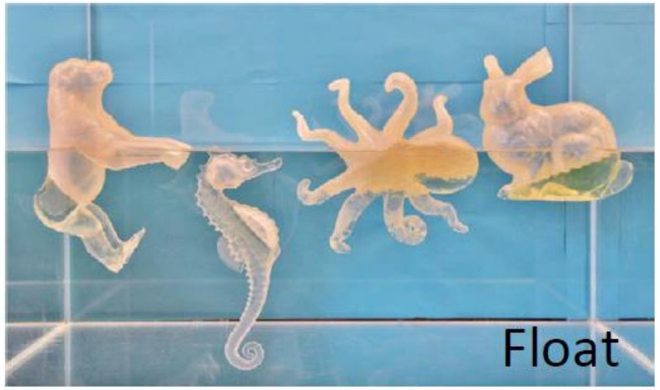




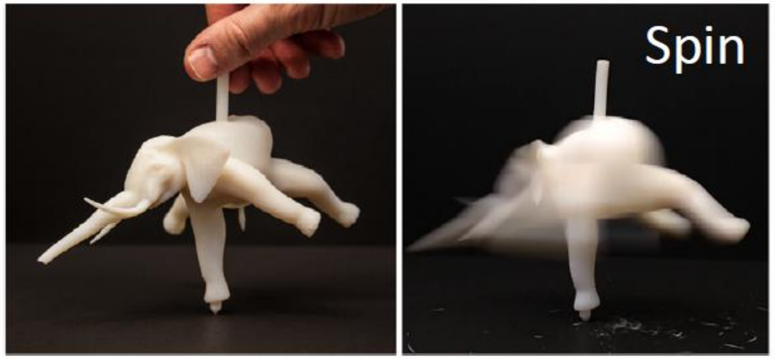
Physically Based Modelling course notes, USC

Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation
    - Fabrication





3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

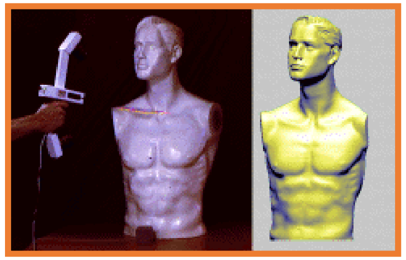
3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

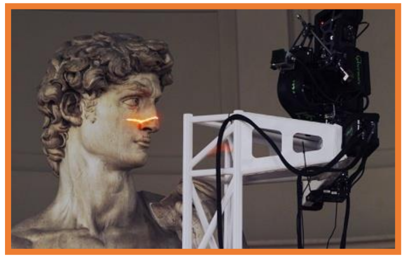
Range Image

Set of 3D points mapping to pixels of depth image

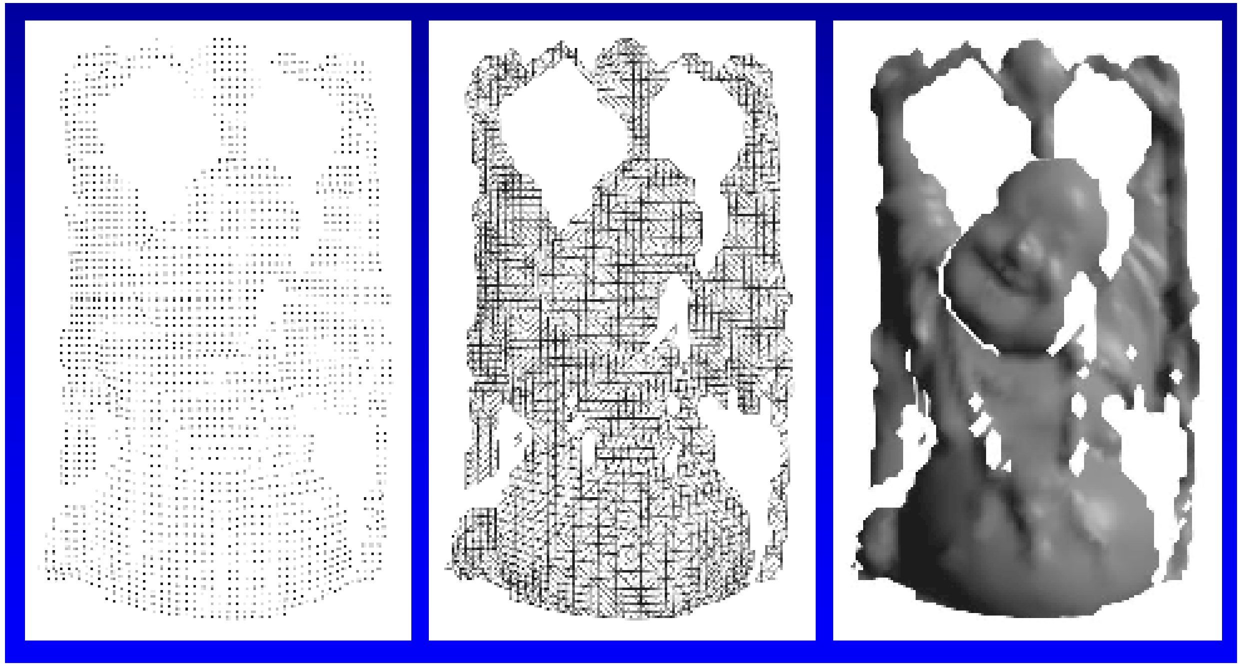
- Can be acquired from range scanner



Cyberware



Stanford



Range Image　　Tesselation　　Range Surface

Brian Curless
SIGGRAPH 99
Course #4 Notes

Point Cloud

Unstructured set of 3D point samples

- Acquired from range finder, computer vision, etc

Polhemus

Microscribe-3D

Hoppe

Hoppe

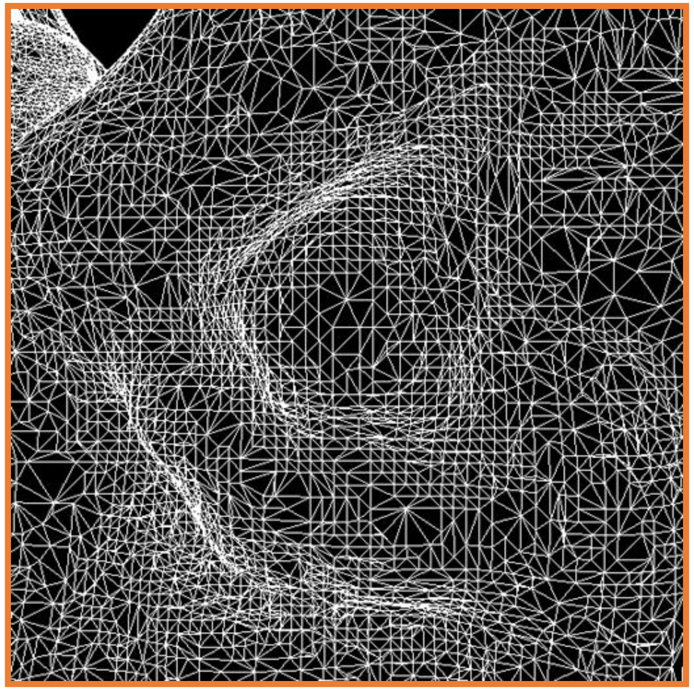
Meshlab demo

3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

Polygonal Mesh

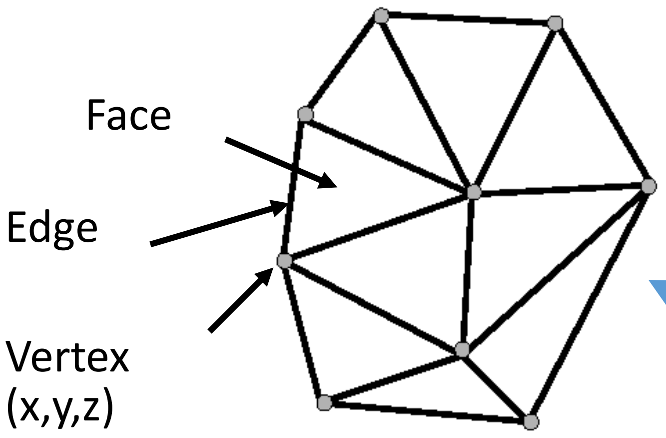
Connected set of polygons (often triangles)

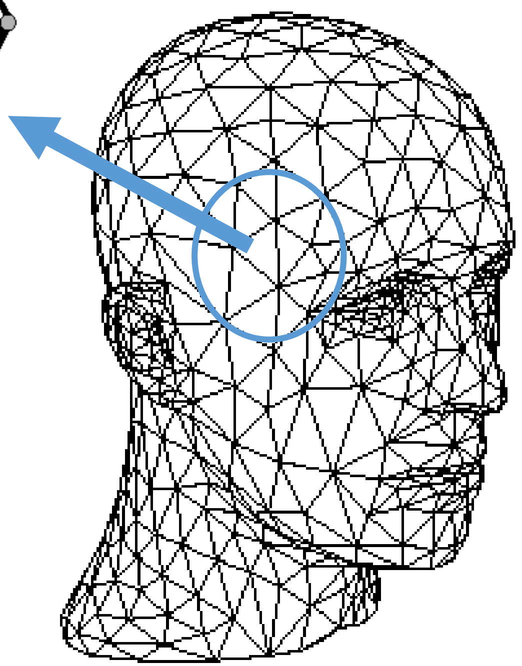


Stanford Graphics Laboratory

3D Polygonal Mesh

Set of polygons representing a 2D surface embedded in 3D





Meshlab demo

Zorin & Schroeder

Parametric Surfaces

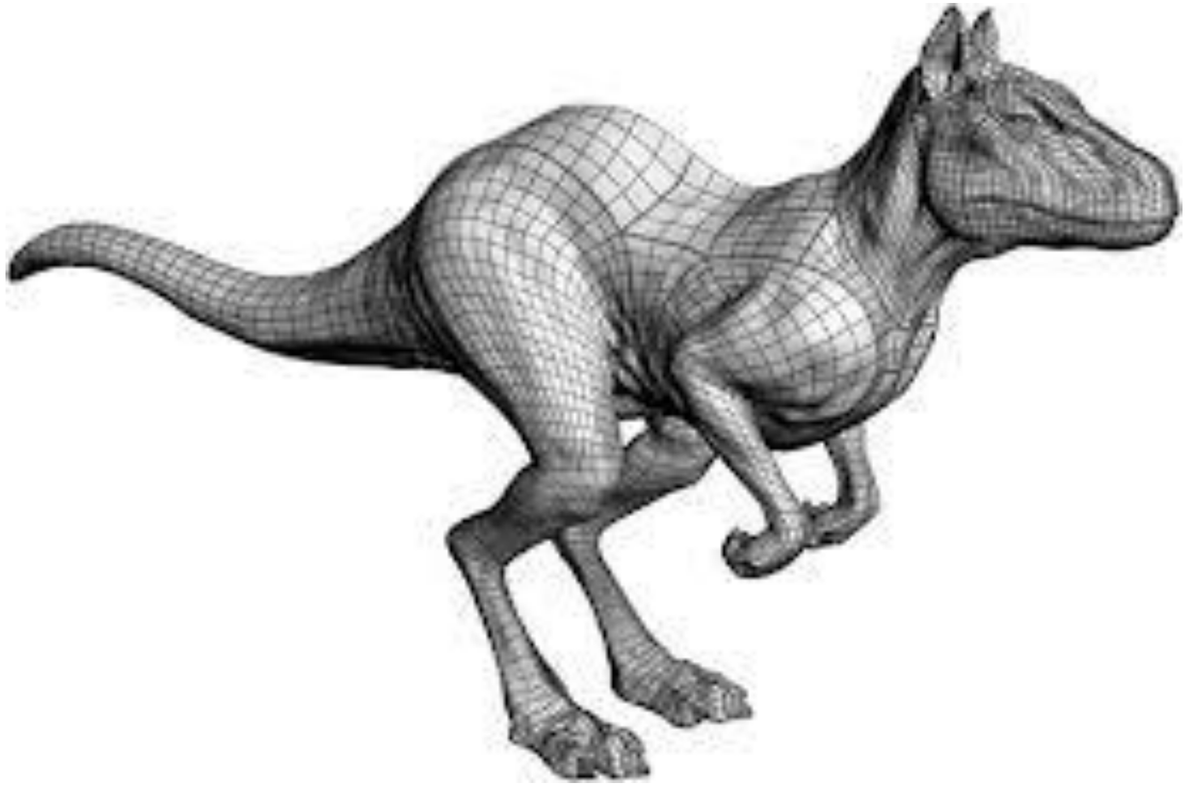
Applications

- Design of smooth surfaces in cars, ships, etc.

Parametric Surfaces

Applications

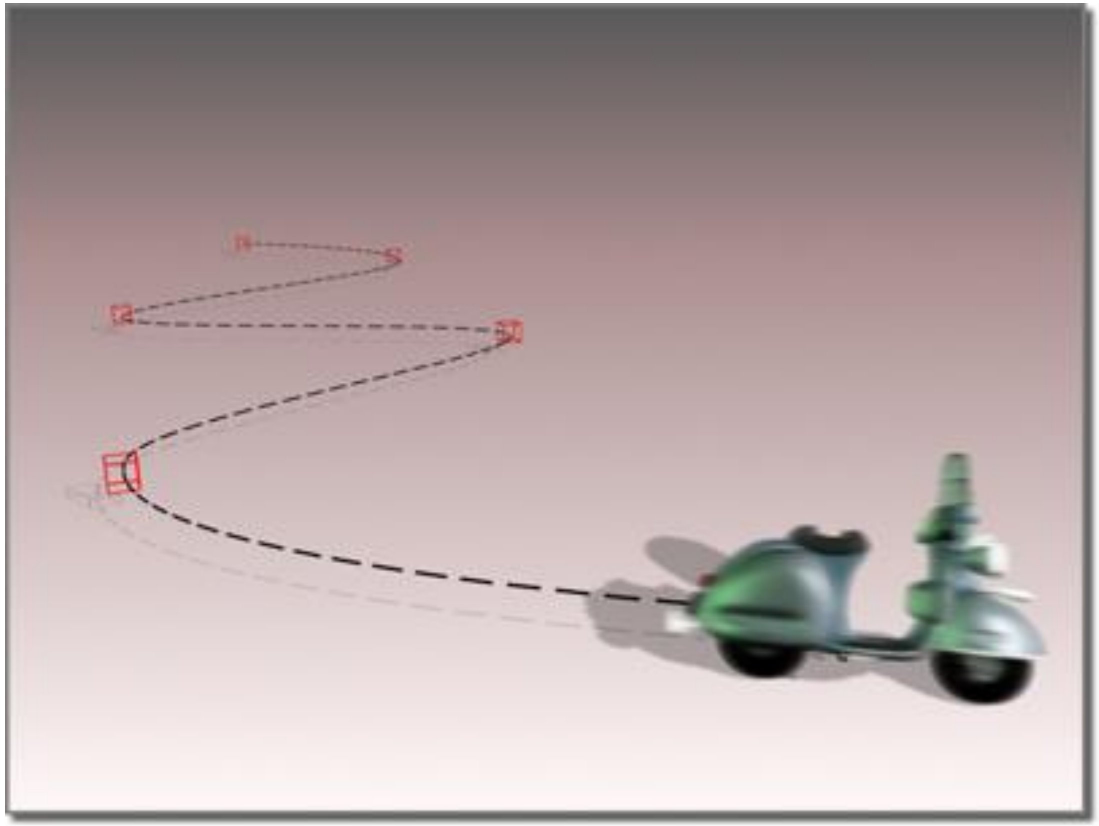
- Design of smooth surfaces in cars, ships, etc.
- Creating characters or scenes for movies



Parametric Curves

Applications

- Defining motion trajectories for objects or cameras



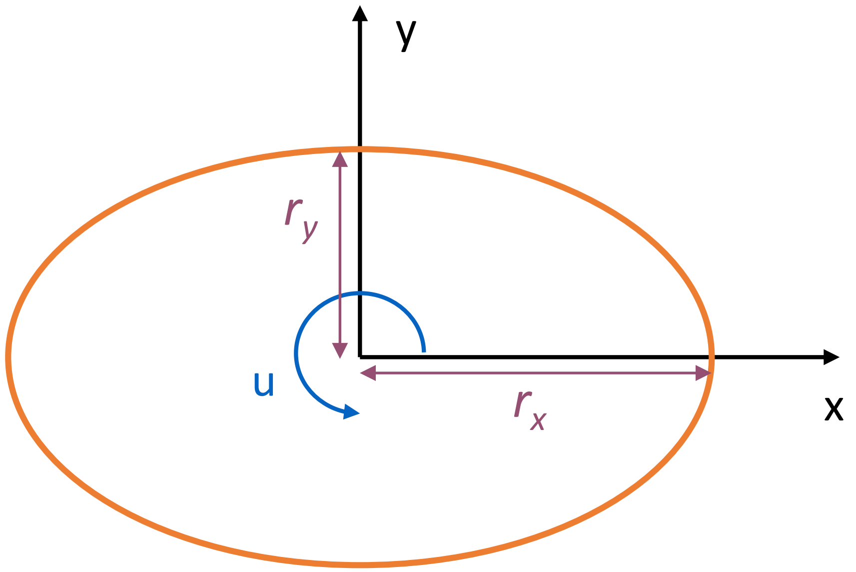
AutoDesk

Parametric Curves

- Defined by parametric functions:
  - $x = f_x(u)$
  - $y = f_y(u)$

- Example: ellipse

$$f_x(u) = r_x \cos(2\pi u)$$
$$f_y(u) = r_y \sin(2\pi u)$$
$$u \in [0..1]$$

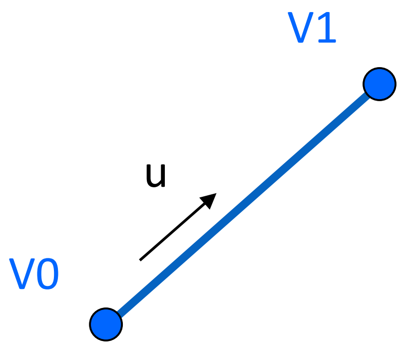


H&B Figure 10.10

Parametric curves

How to easily define arbitrary curves?

$$x = f_x(u)$$
$$y = f_y(u)$$



Use functions that “blend” control points

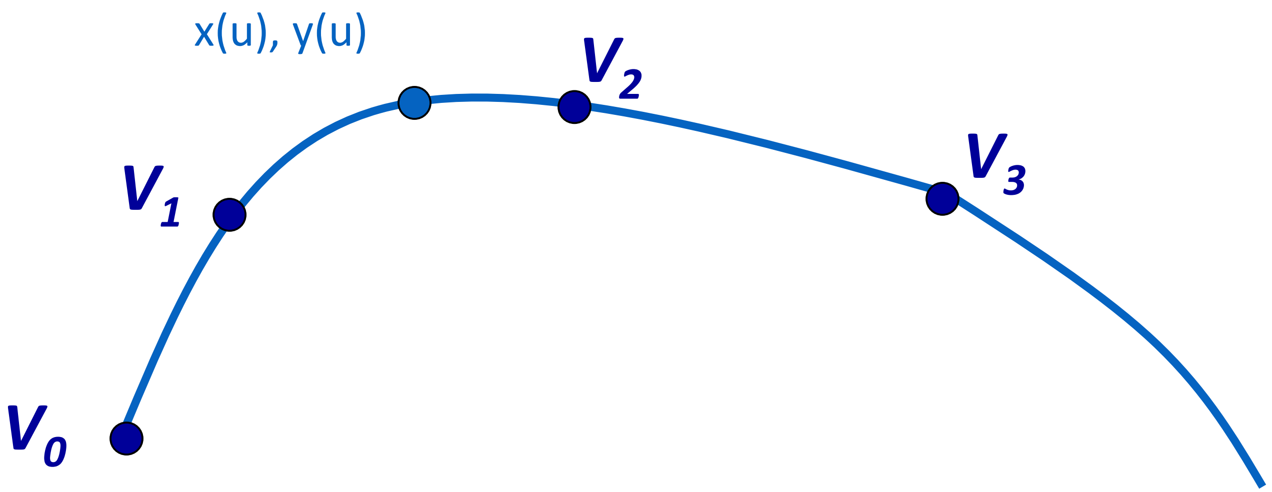
$$x = f_x(u) = V0_x*(1 - u) + V1_x*u$$
$$y = f_y(u) = V0_y*(1 - u) + V1_y*u$$

Parametric curves

More generally:

$$x(u) = \sum_{i=0}^{n} B_i(u) * Vi_x$$

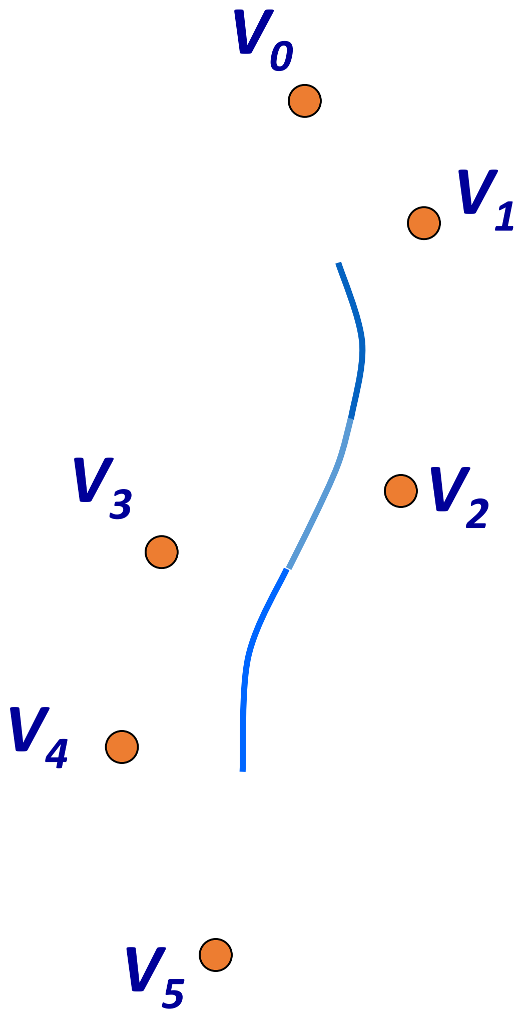
$$y(u) = \sum_{i=0}^{n} B_i(u) * Vi_y$$



Cubic B-Spline Blending Functions

- Four cubic polynomials for four vertices
  - 16 variables (degrees of freedom)
  - Variables are a_i, b_i, c_i, d_i for four blending functions

$$b_{-0}(u) = a_0 u^3 + b_0 u^2 + c_0 u^1 + d_0$$
$$b_{-1}(u) = a_1 u^3 + b_1 u^2 + c_1 u^1 + d_1$$
$$b_{-2}(u) = a_2 u^3 + b_2 u^2 + c_2 u^1 + d_2$$
$$b_{-3}(u) = a_3 u^3 + b_3 u^2 + c_3 u^1 + d_3$$



Cubic B-Spline Blending Functions

Fifteen continuity constraints:

$$0 = b_{-0}(0)$$
$$b_{-0}(1) = b_{-1}(0)$$
$$b_{-1}(1) = b_{-2}(0)$$
$$b_{-2}(1) = b_{-3}(0)$$
$$b_{-3}(1) = 0$$

$$0 = b_{-0}'(0)$$
$$b_{-0}'(1) = b_{-1}'(0)$$
$$b_{-1}'(1) = b_{-2}'(0)$$
$$b_{-2}'(1) = b_{-3}'(0)$$
$$b_{-3}'(1) = 0$$

$$0 = b_{-0}''(0)$$
$$b_{-0}''(1) = b_{-1}''(0)$$
$$b_{-1}''(1) = b_{-2}''(0)$$
$$b_{-2}''(1) = b_{-3}''(0)$$
$$b_{-3}''(1) = 0$$

One more convenient constraint:

$$b_{-0}(0) + b_{-1}(0) + b_{-2}(0) + b_{-3}(0) = 1$$

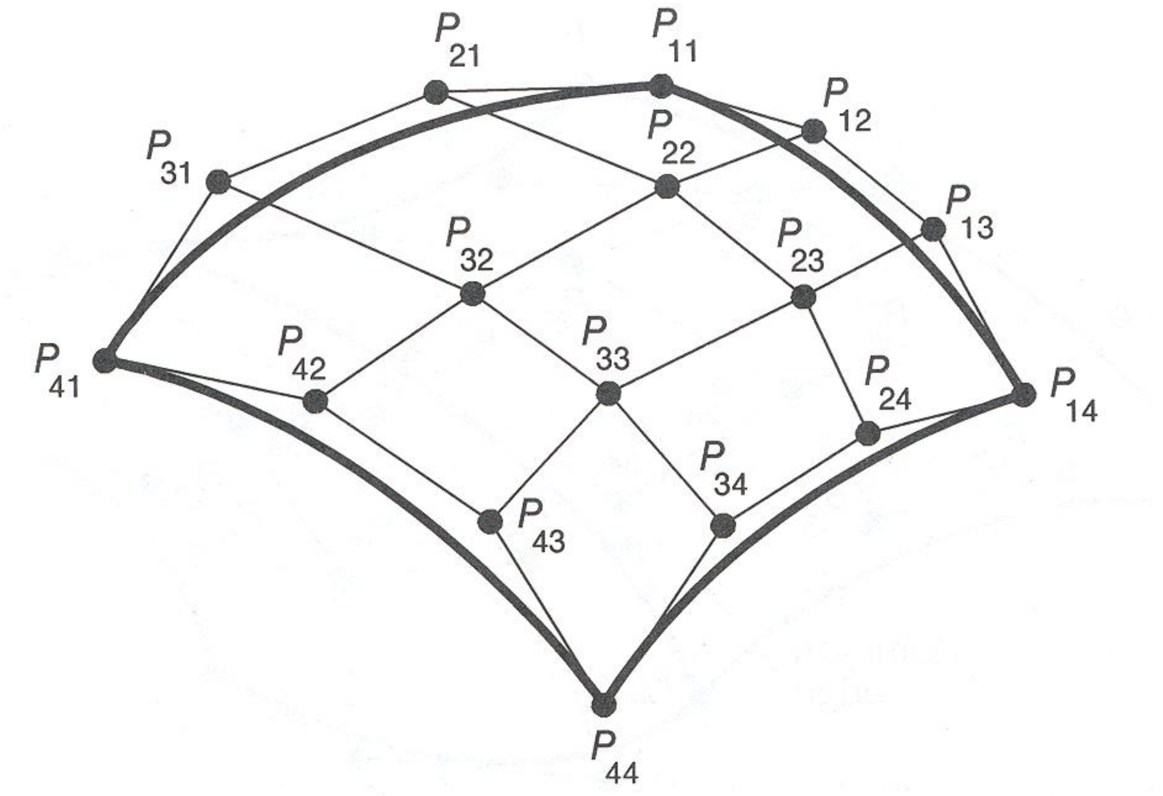
Cubic B-Spline Blending Functions

Solving the system of equations yields:

$$b_{-3}(u) = -\tfrac{1}{6}u^3 + \tfrac{1}{2}u^2 - \tfrac{1}{2}u + \tfrac{1}{6}$$
$$b_{-2}(u) = \tfrac{1}{2}u^3 - u^2 + \tfrac{2}{3}$$
$$b_{-1}(u) = -\tfrac{1}{2}u^3 + \tfrac{1}{2}u^2 + \tfrac{1}{2}u + \tfrac{1}{6}$$
$$b_{-0}(u) = \tfrac{1}{6}u^3$$

Parametric Patches

- Each patch is defined by blending control points

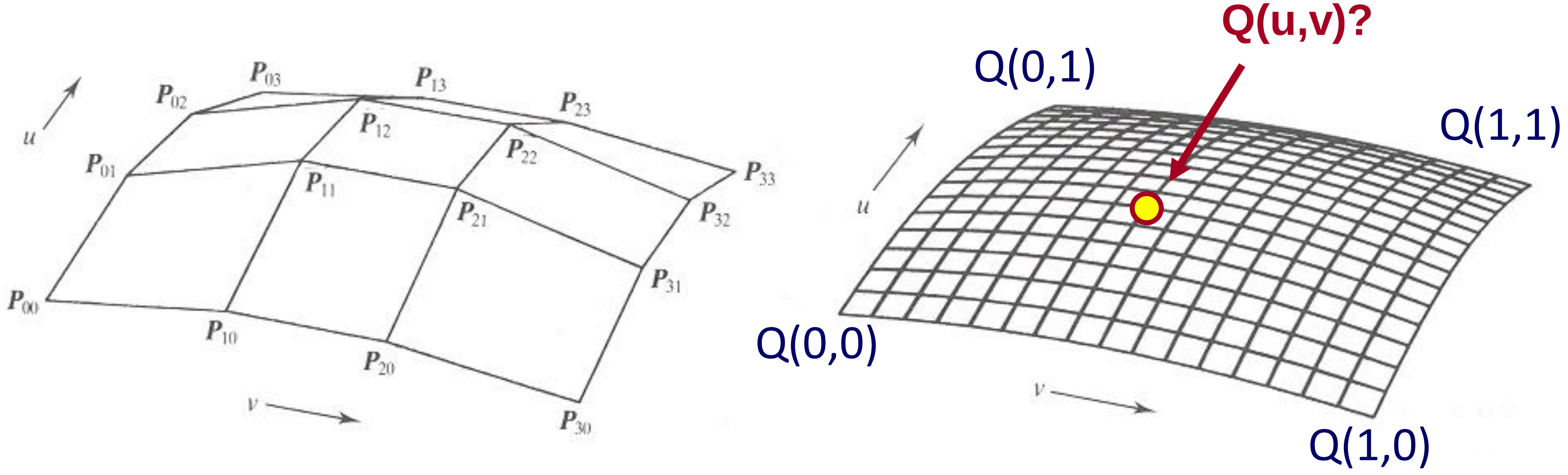


Same ideas as parametric curves!

FvDFH Figure 11.44

Parametric Patches

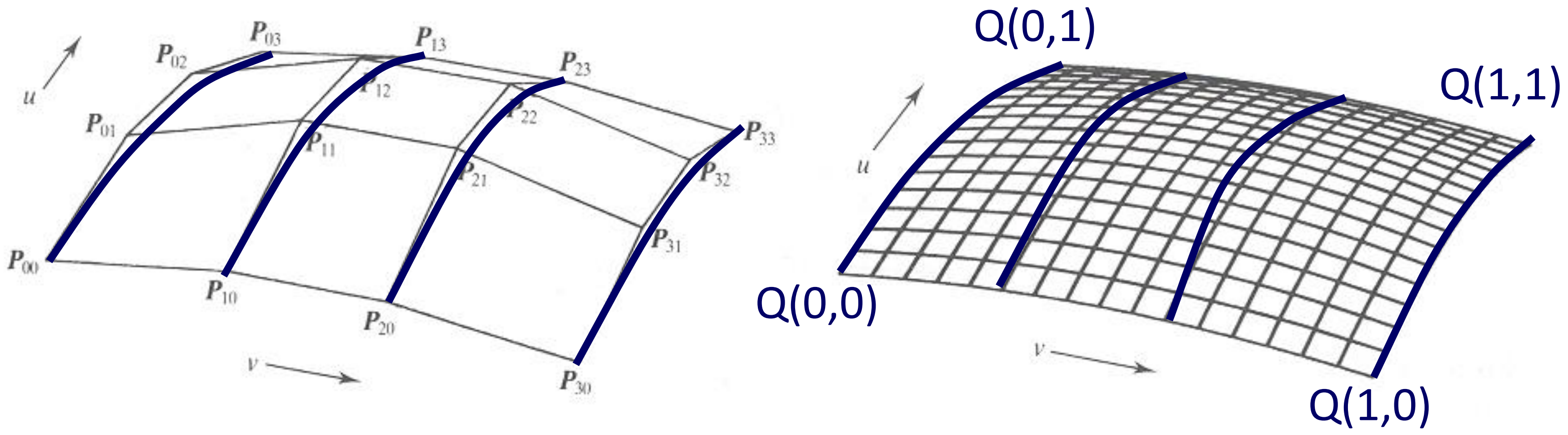
- Point Q(u,v) on the patch is the tensor product of parametric curves defined by the control points



Watt Figure 6.21

Parametric Patches

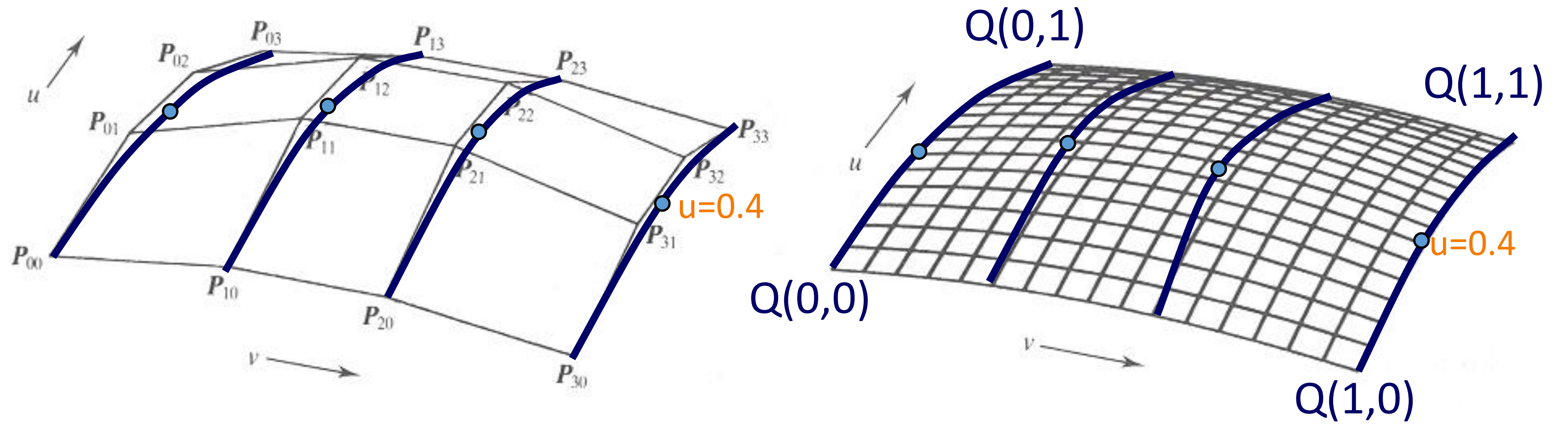
- Point Q(u,v) on the patch is the tensor product of parametric curves defined by the control points



Watt Figure 6.21

Parametric Patches

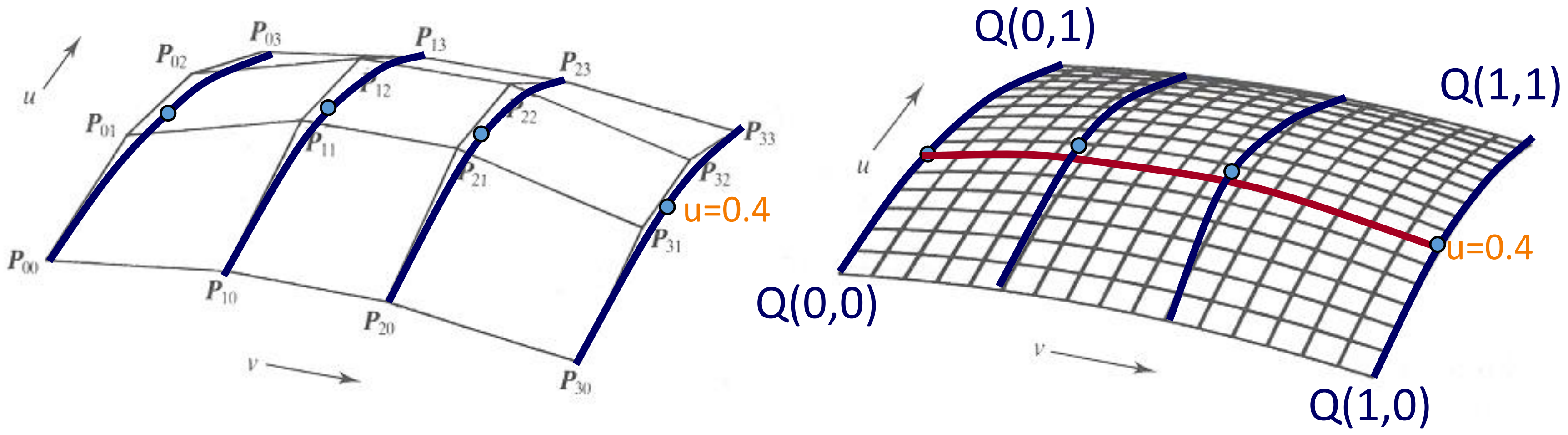
- Point Q(u,v) on the patch is the tensor product of parametric curves defined by the control points



Watt Figure 6.21

Parametric Patches

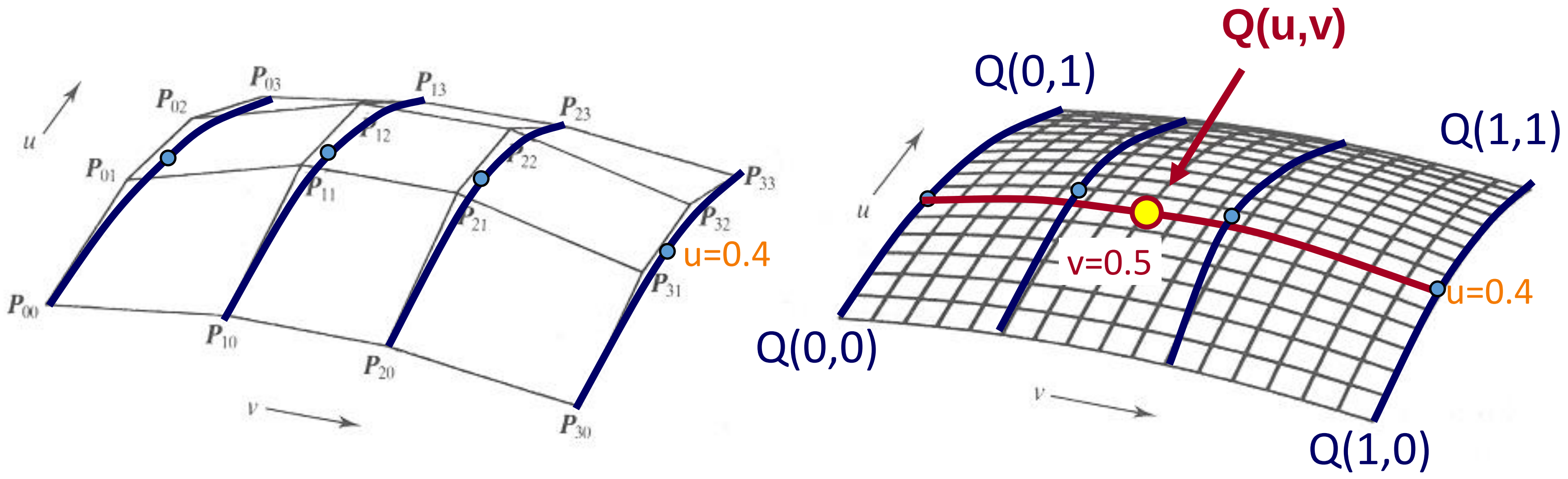
- Point Q(u,v) on the patch is the tensor product of parametric curves defined by the control points



Watt Figure 6.21

Parametric Patches

- Point Q(u,v) on the patch is the tensor product of parametric curves defined by the control points



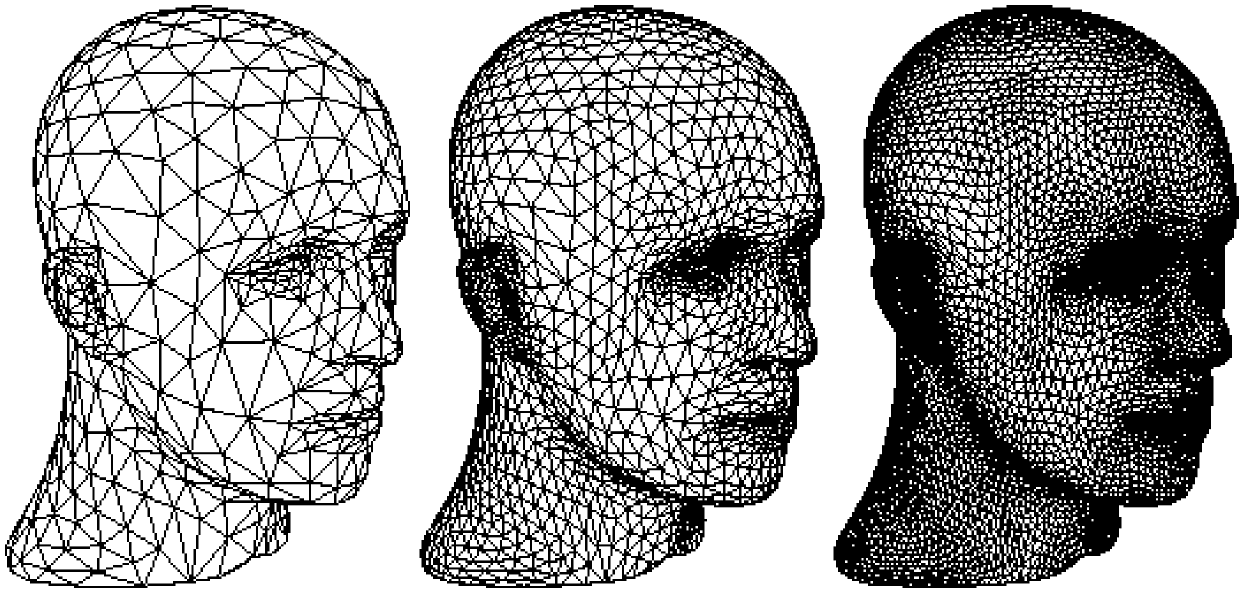
Watt Figure 6.21

NURBS Surfaces

Subdivision Surface

Coarse mesh & subdivision rule

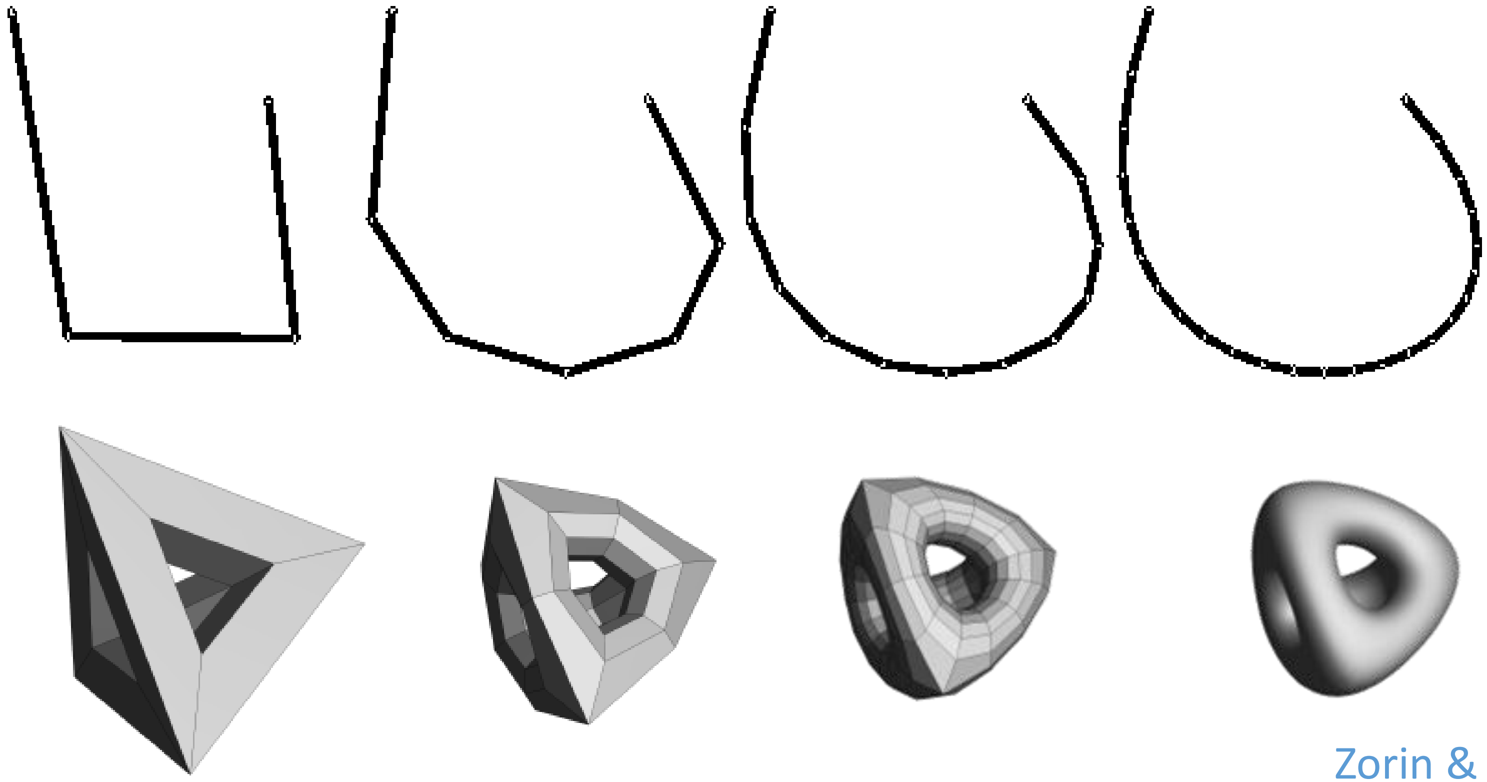
- Smooth surface is limit of sequence of refinements



Zorin & Schroeder
SIGGRAPH 99
Course Notes

Subdivision

How do you make a surface with guaranteed continuity?

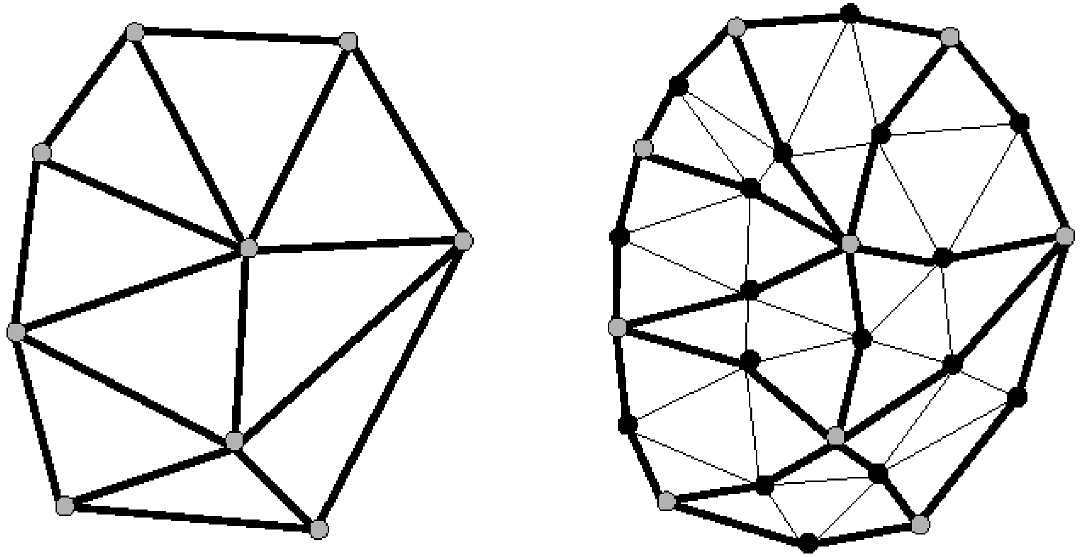


Zorin & Schroeder
SIGGRAPH 99
Course Notes

Subdivision Surfaces

Repeated application of

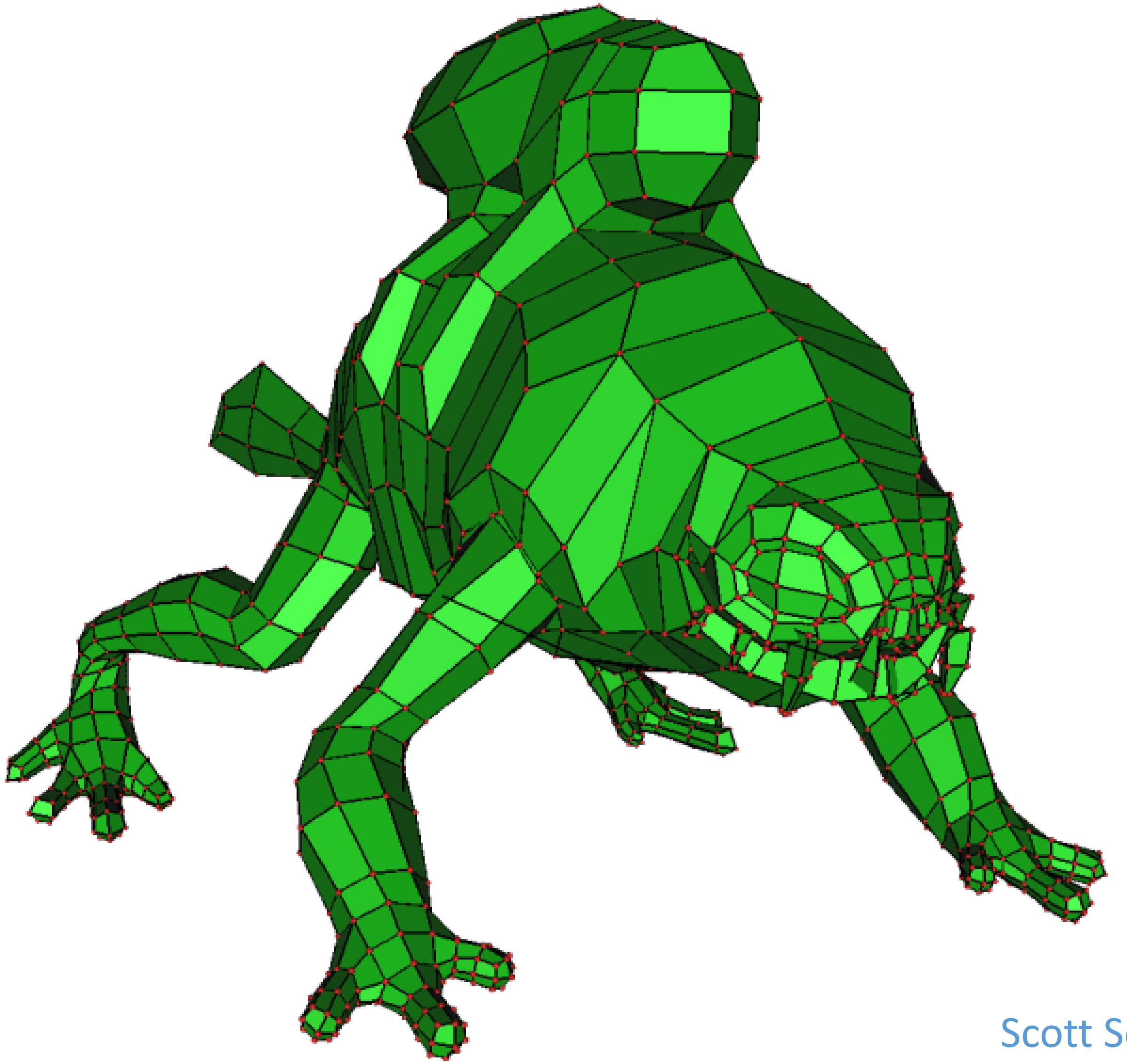
- Topology refinement (splitting faces)
- Geometry refinement (weighted averaging)



Zorin & Schroeder
SIGGRAPH 99
Course Notes

Subdivision Surfaces – Examples

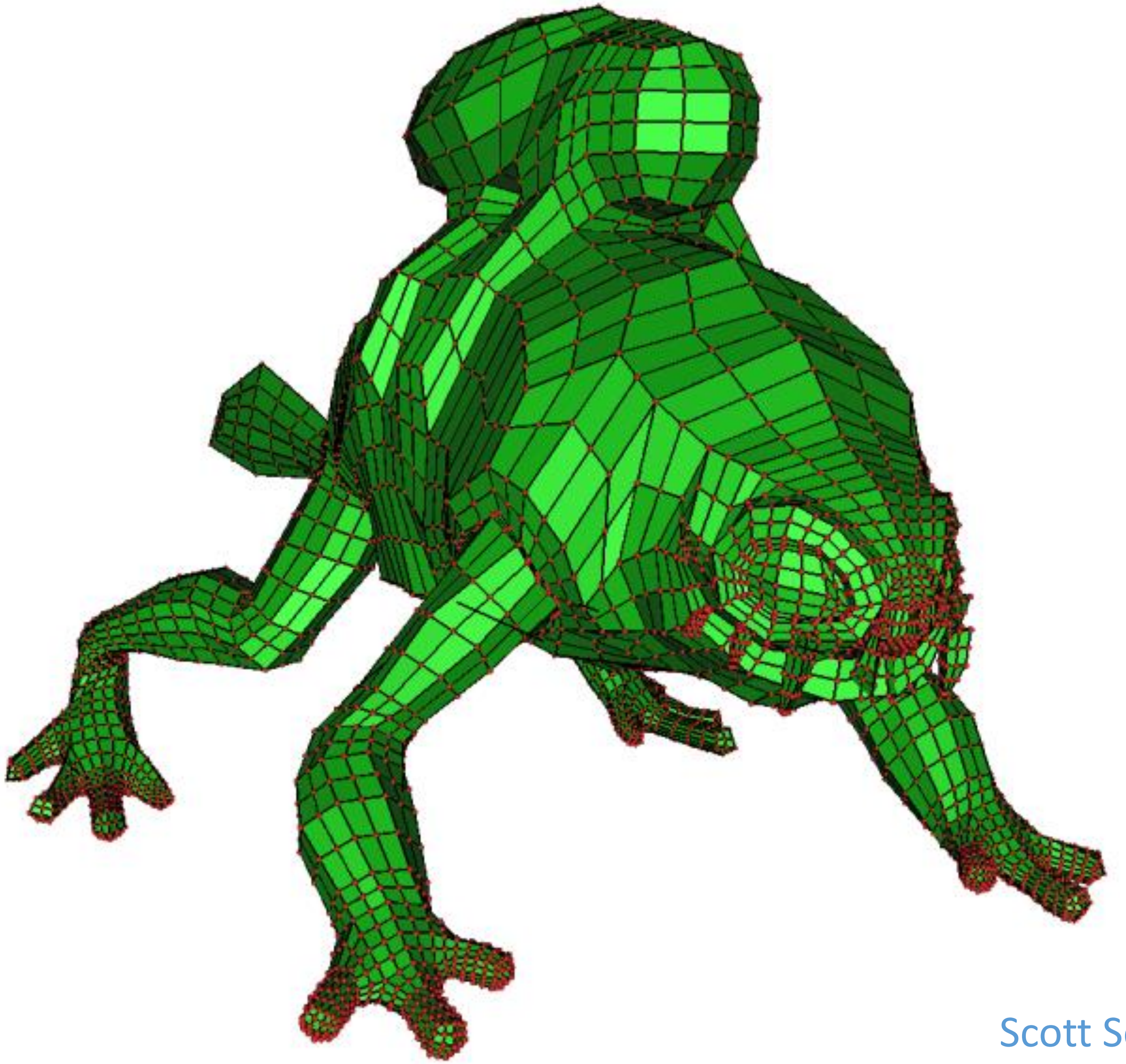
Base mesh



Scott Schaefer

Subdivision Surfaces – Examples

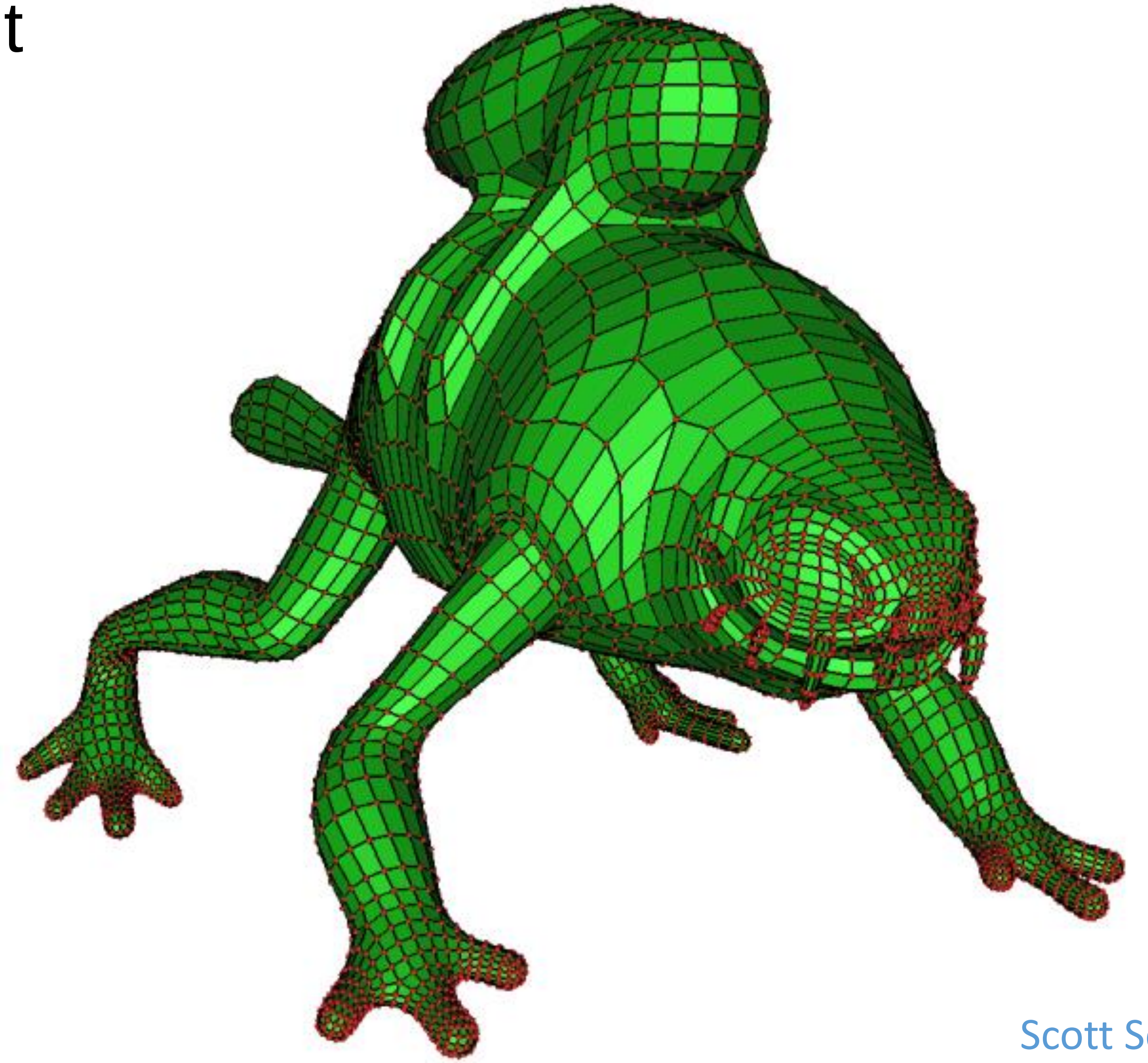
Topology refinement



Scott Schaefer

Subdivision Surfaces – Examples

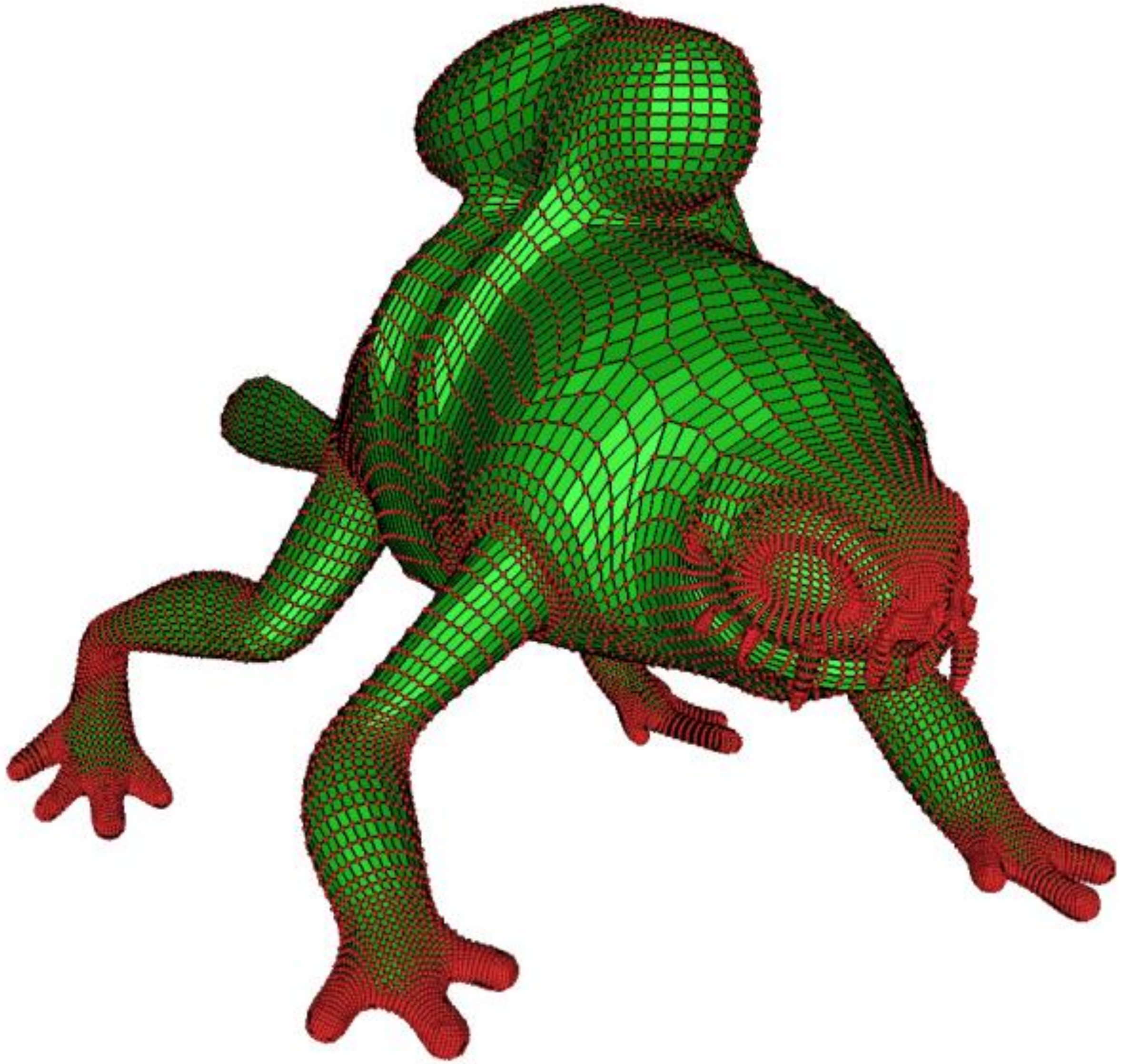
Geometry refinement



Scott Schaefer

Subdivision Surfaces – Examples

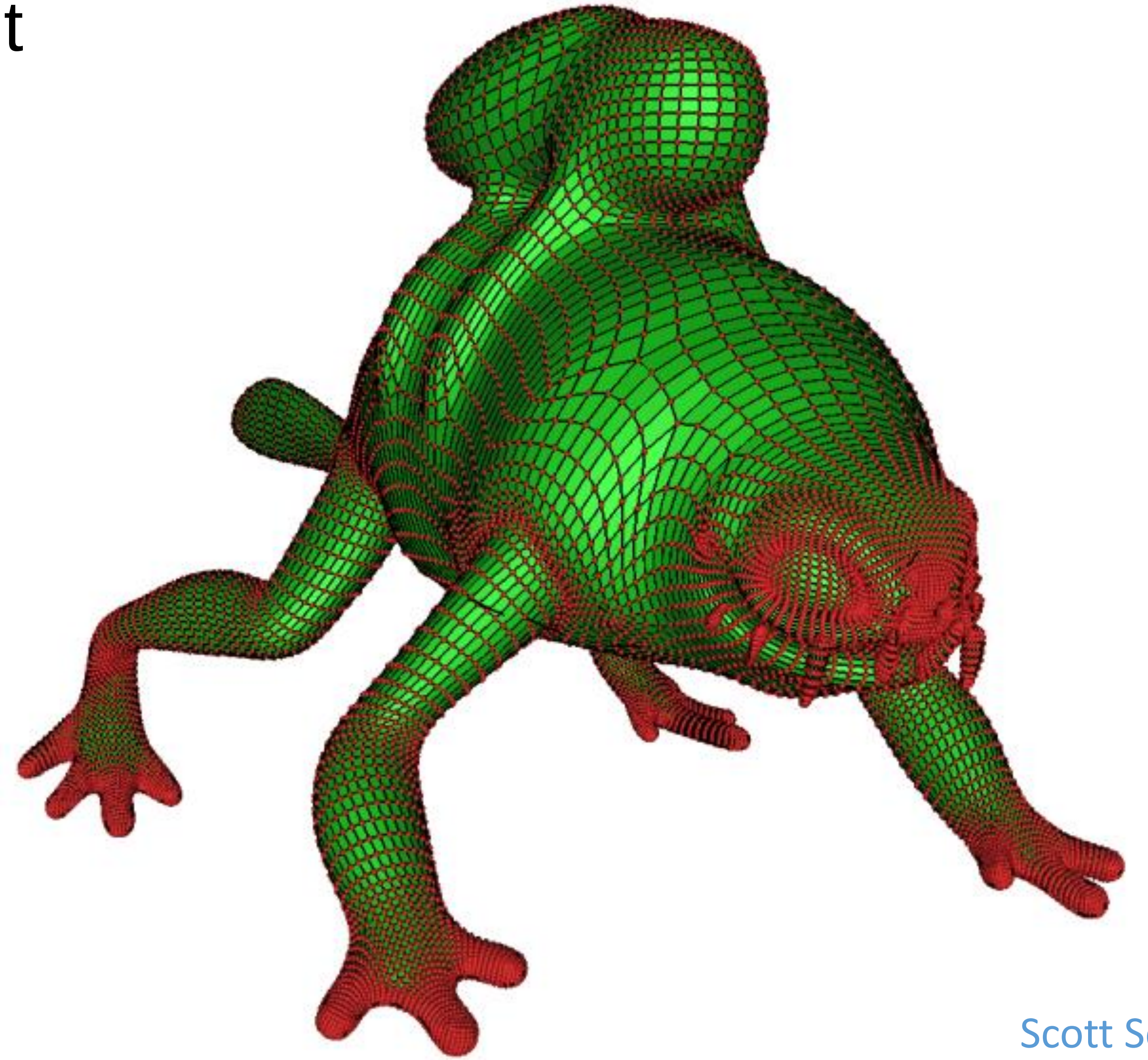
Topology refinement



chaefer

Subdivision Surfaces – Examples

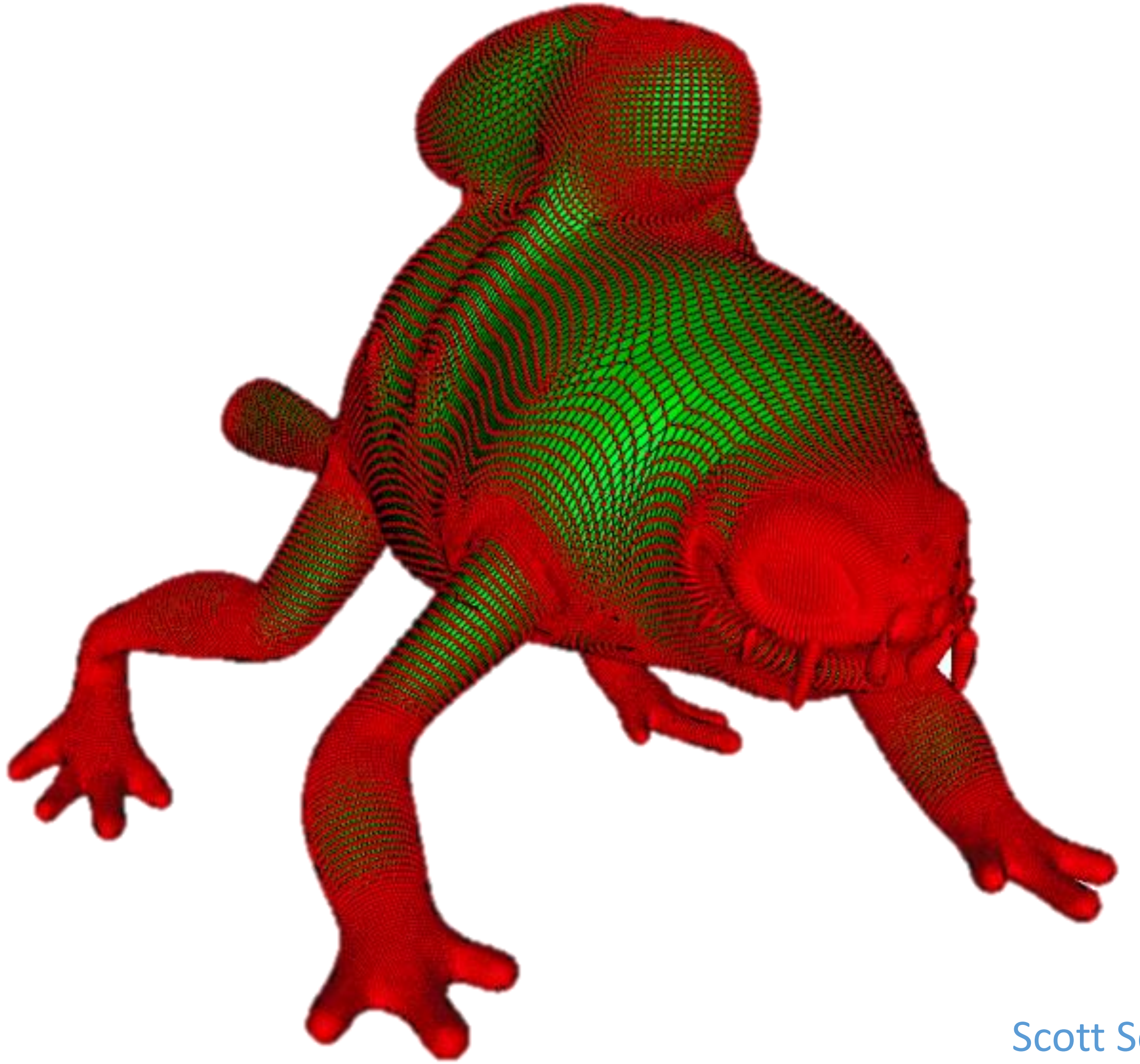
Geometry refinement



Scott Schaefer

Subdivision Surfaces – Examples

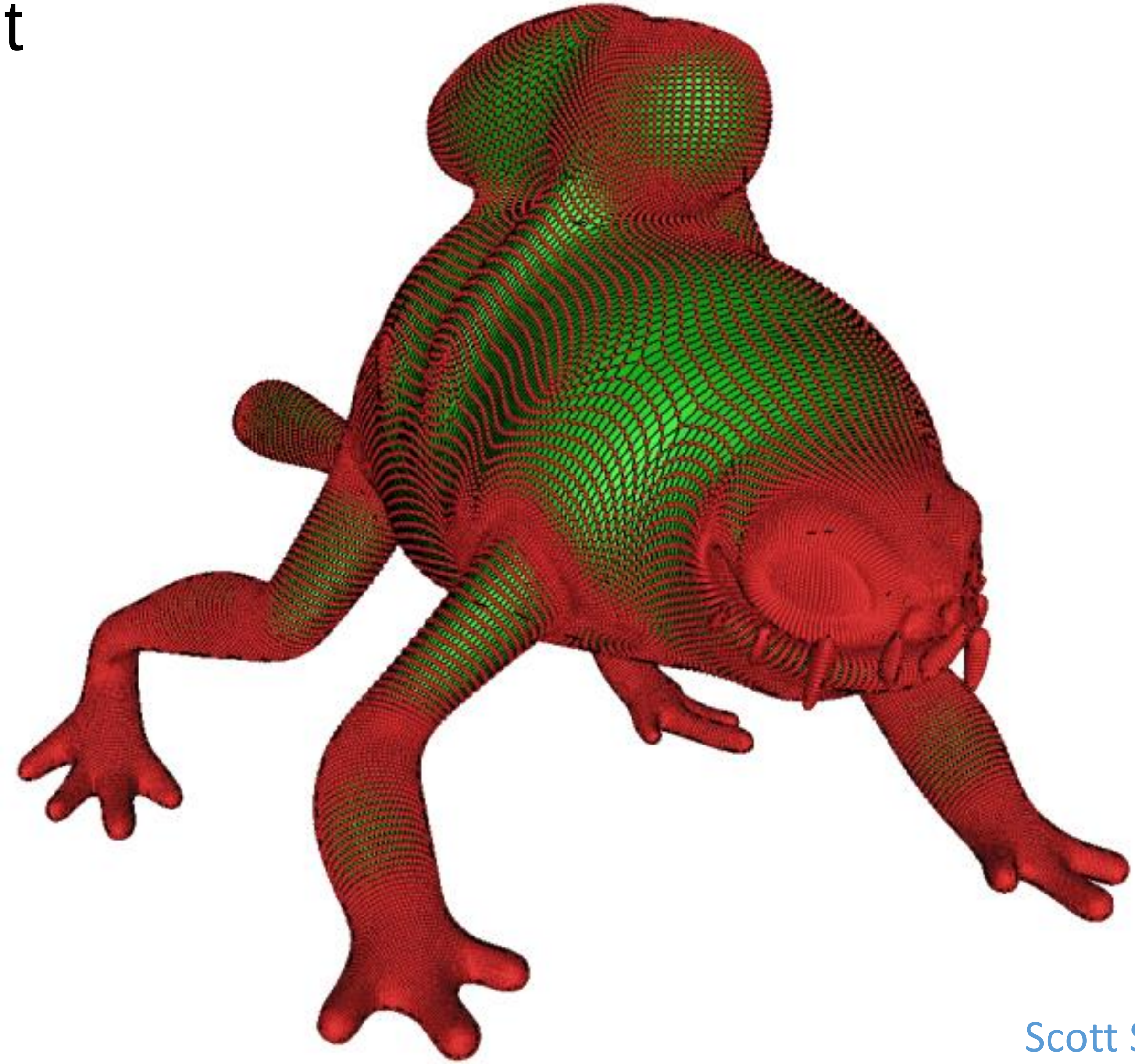
Topology refinement



Scott Schaefer

Subdivision Surfaces – Examples

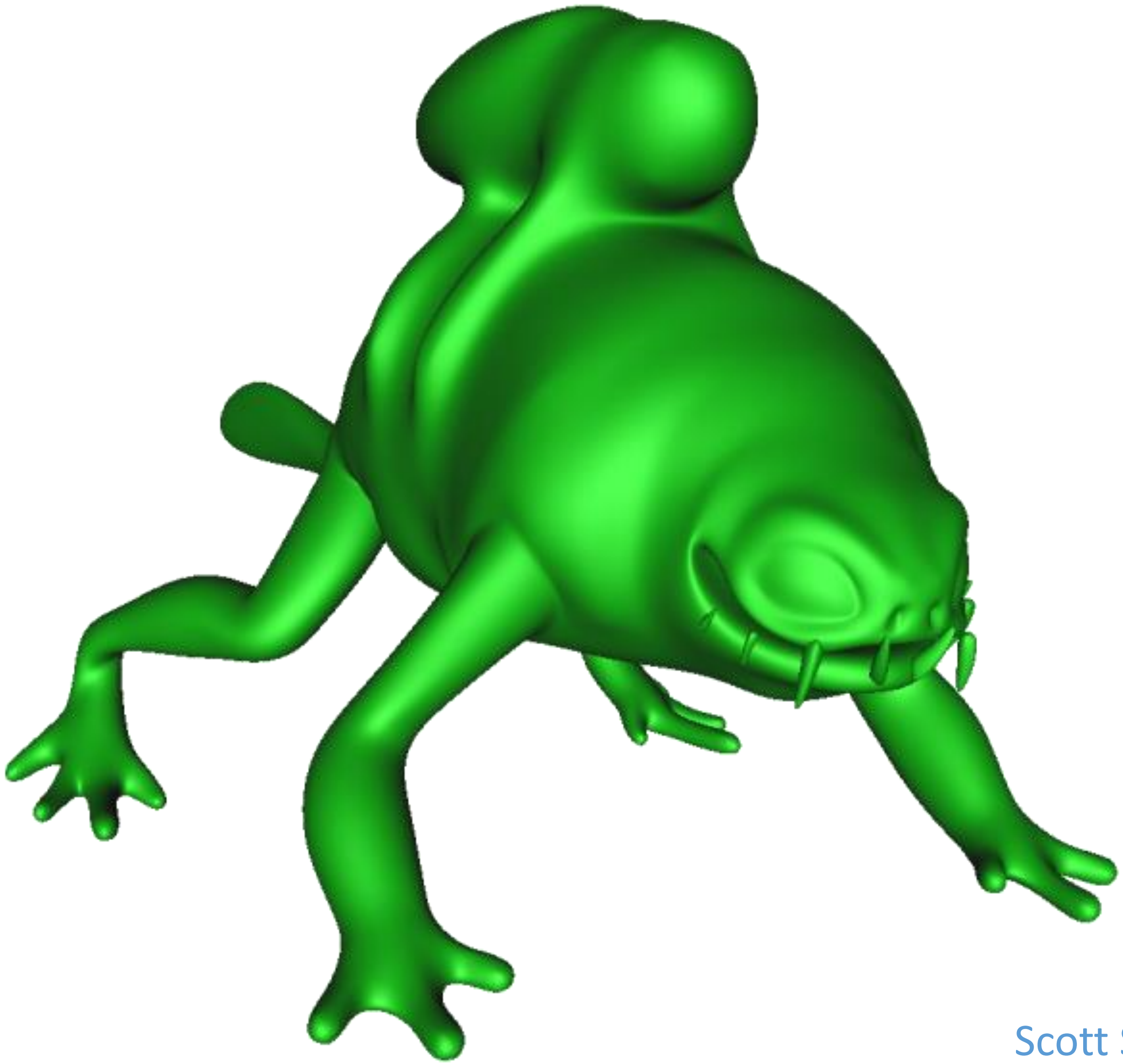
Geometry refinement



Scott Schaefer

Subdivision Surfaces – Examples

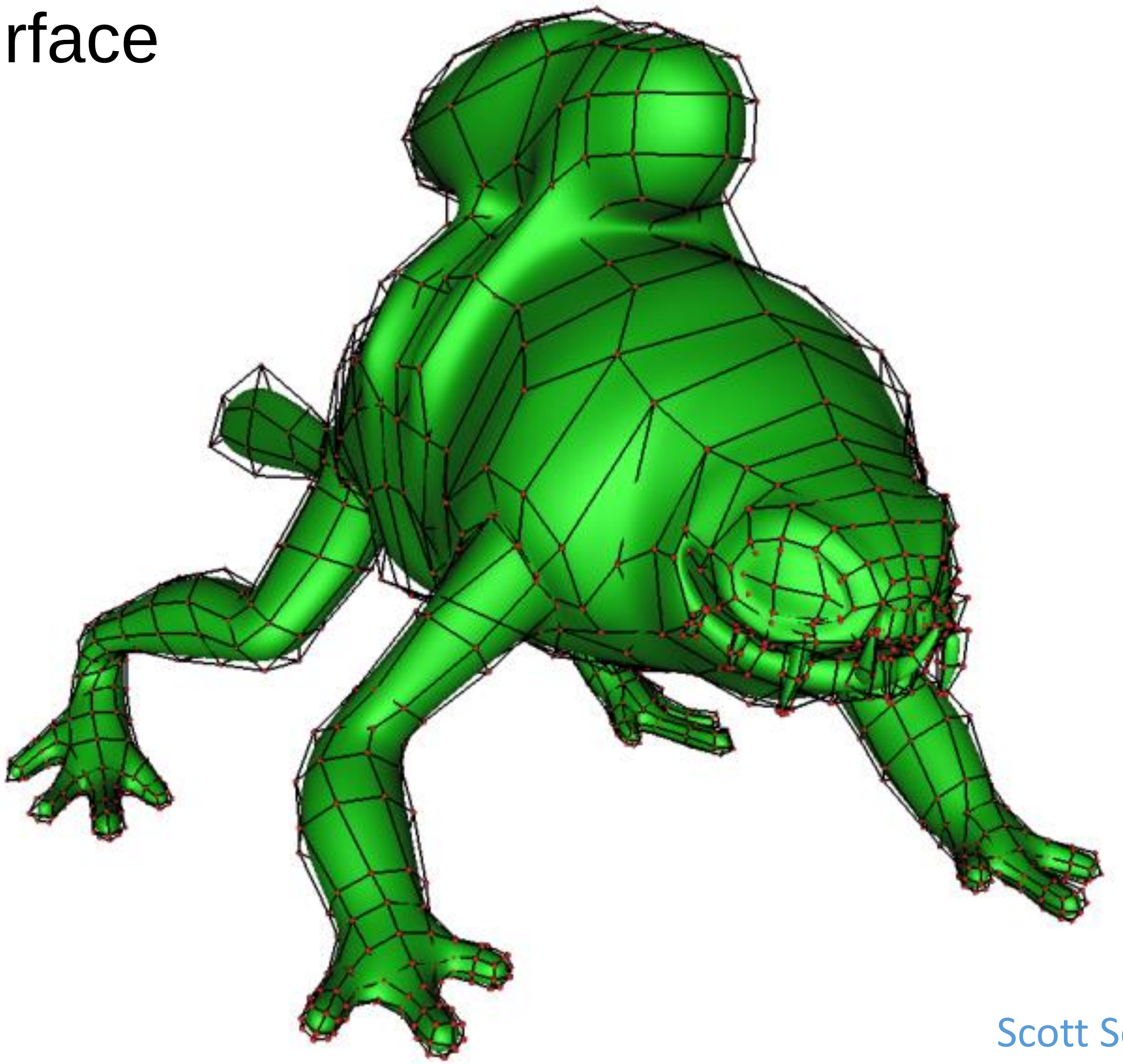
Limit surface



Scott Schaefer

Subdivision Surfaces – Examples

Base mesh + limit surface



Scott Schaefer

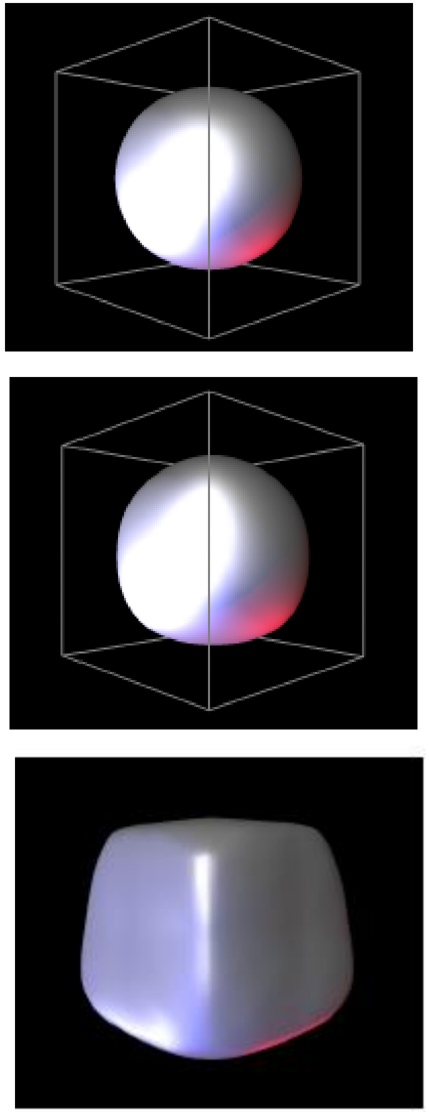
Meshlab demo

Subdivision Schemes

- Common subdivision schemes
  - Catmull-Clark
  - Loop
  - Many others
- Differ in ...
  - Input topology
  - How refine topology
  - How refine geometry

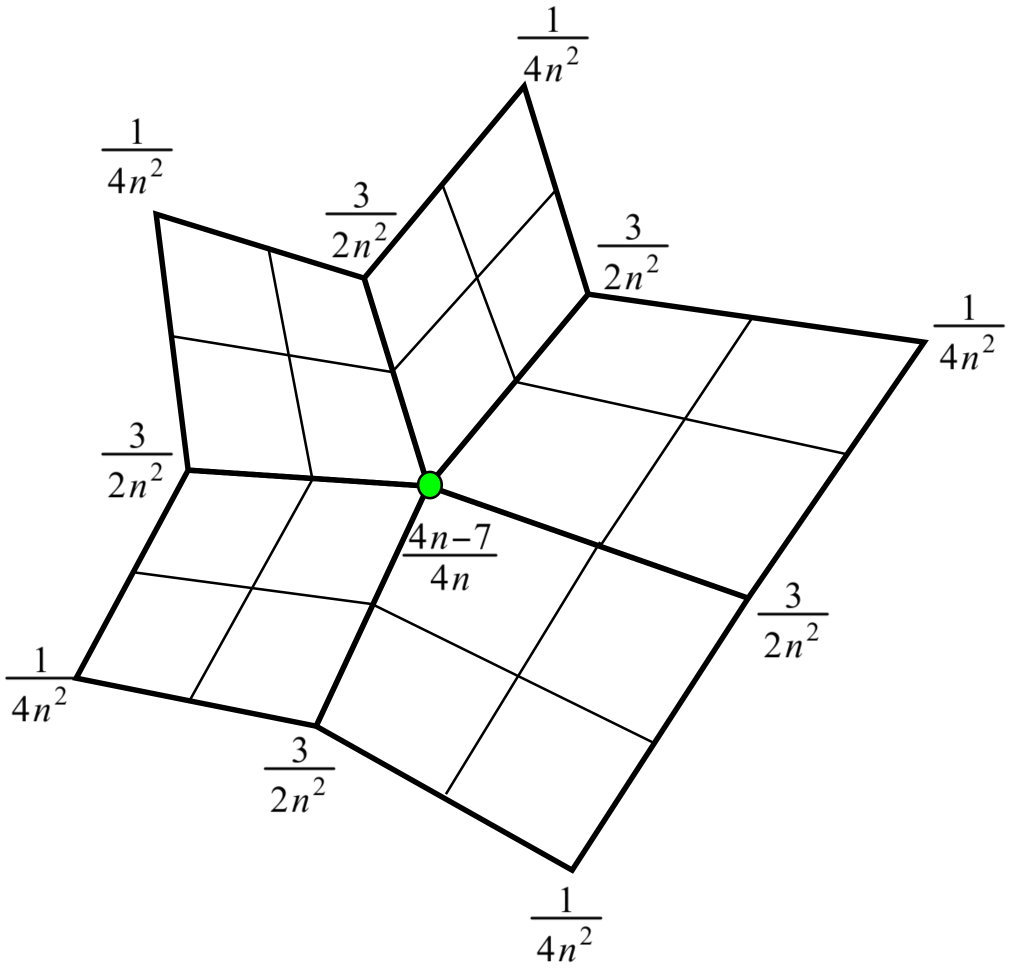
… which makes differences in …

  - Provable properties



/

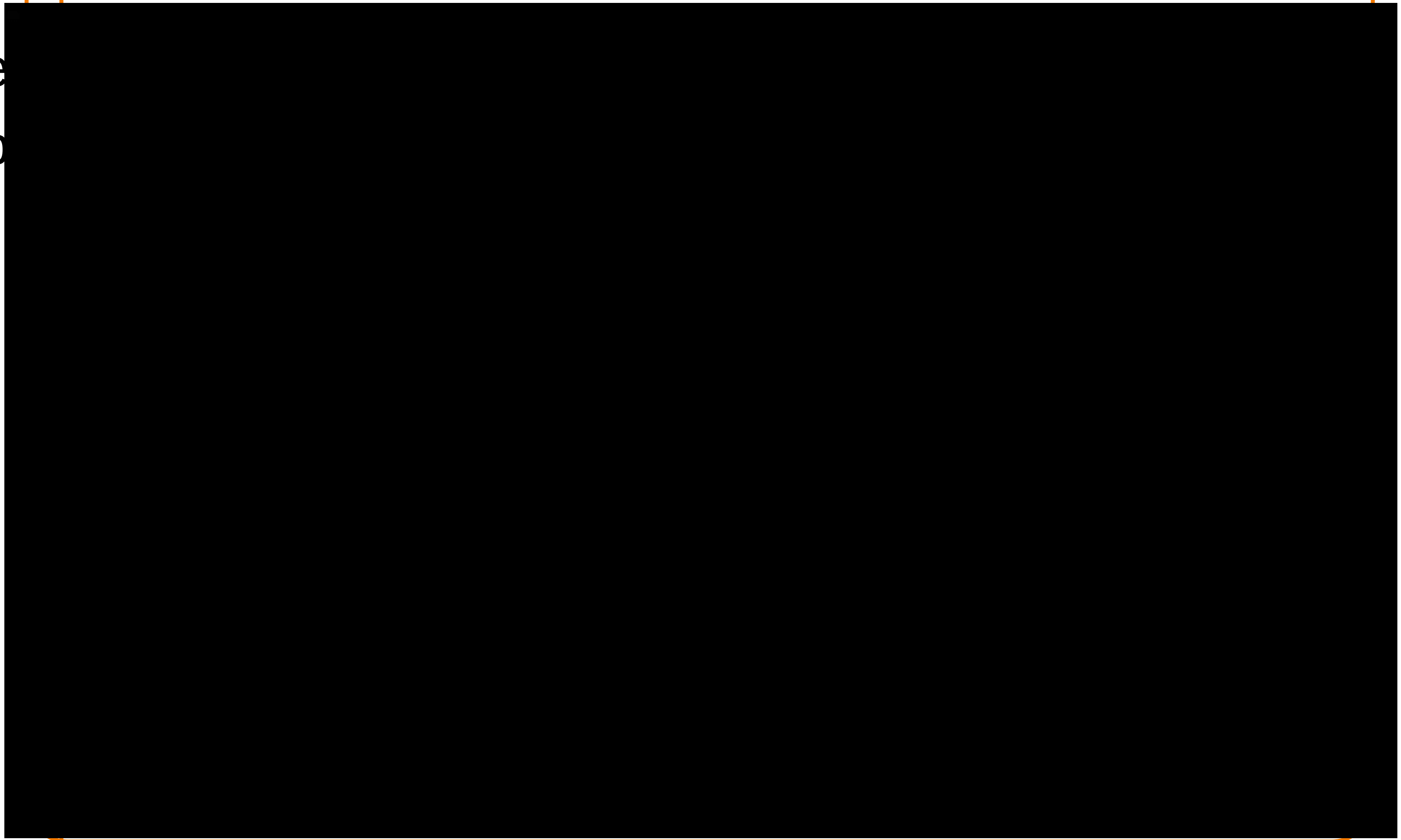
Catmull-Clark Subdivision



Scott Schaefer

Subdivision Surfaces

- Use [redacted]
- Sup [redacted]



Geri’s Game © Pixar Animation Studios

Geri's Game

- "served as a demonstration of a new animation tool called subdivision surfaces" (Wikipedia)
- Subdivision used for head, hands & some clothing
- Academy Award winner

Geri's Game

- Guest performance in Toy Story 2

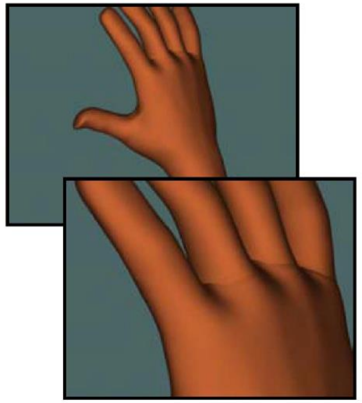
Toy Story 2 © Pixar Animation Studios

Subdivision Surfaces

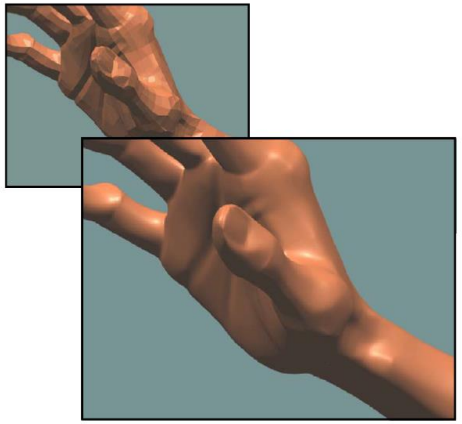
- An alternative to NURBS, overcoming:
  - Many patches
  - Difficult to mark sharp features

  - Irregularities after deformation

Woody's hand (NURBS)



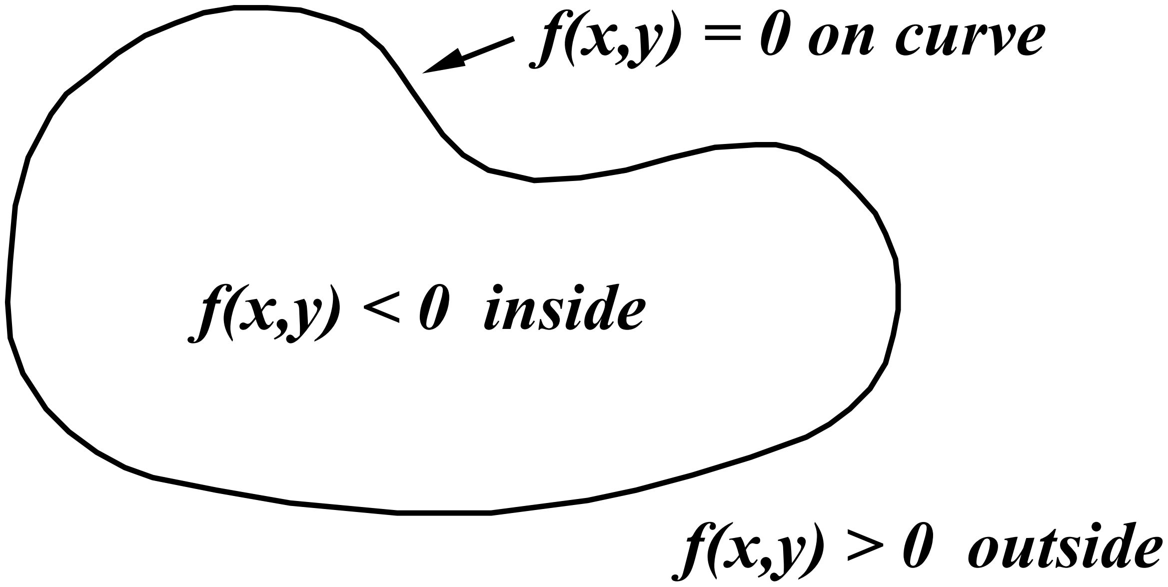
Geri's hand (subdivision)



Stanford Graphics course notes

Implicit Surfaces

- Surface defined implicitly by function:
  - $f\ (x, y, z) = 0$ (on surface)
  - $f\ (x, y, z) < 0$ (inside)
  - $f\ (x, y, z) > 0$ (outside)



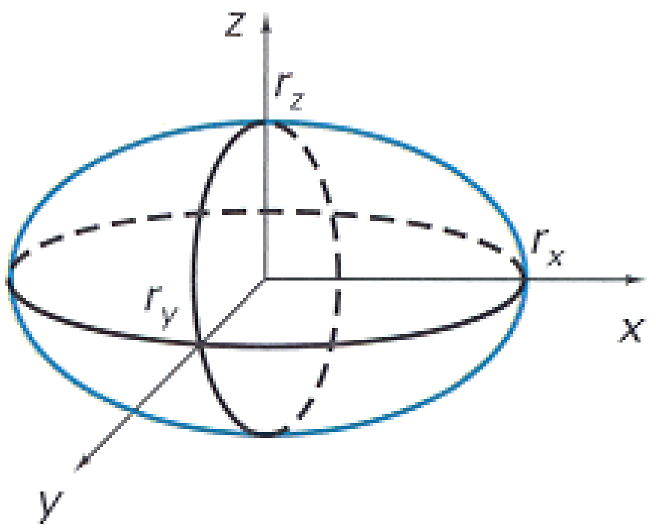
Turk

Implicit Surface Properties

Efficient check for whether point is inside

- Evaluate f(x,y,z) to see if point is inside/outside/on
- Example: ellipsoid

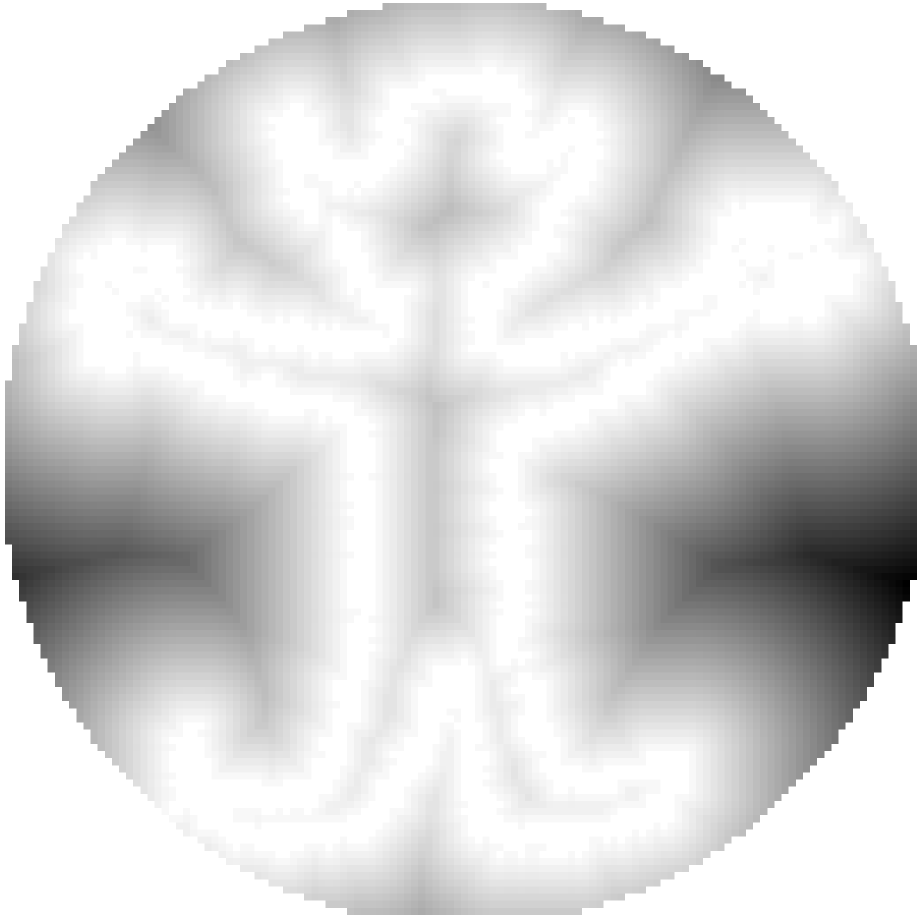
$$f(x,y,z) = \left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 - 1$$



H&B Figure 10.10

Implicit Surfaces

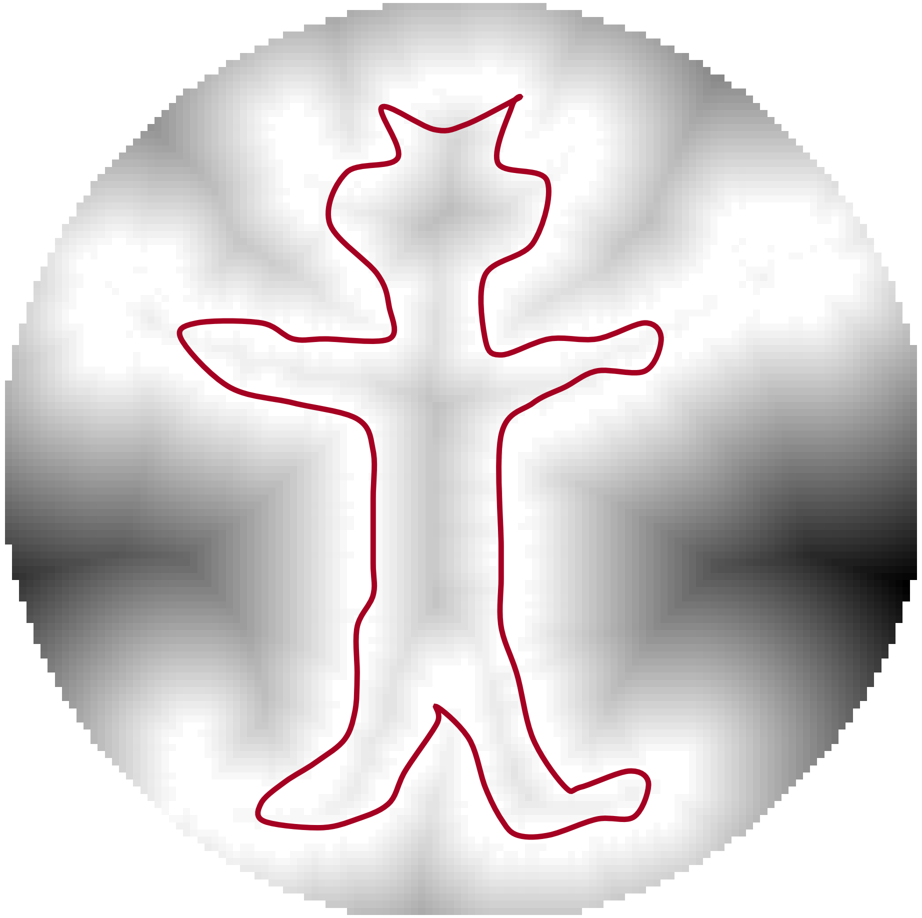
- Represent surface with function over all space



Kazhdan

Implicit Surfaces

- Surface defined implicitly by function

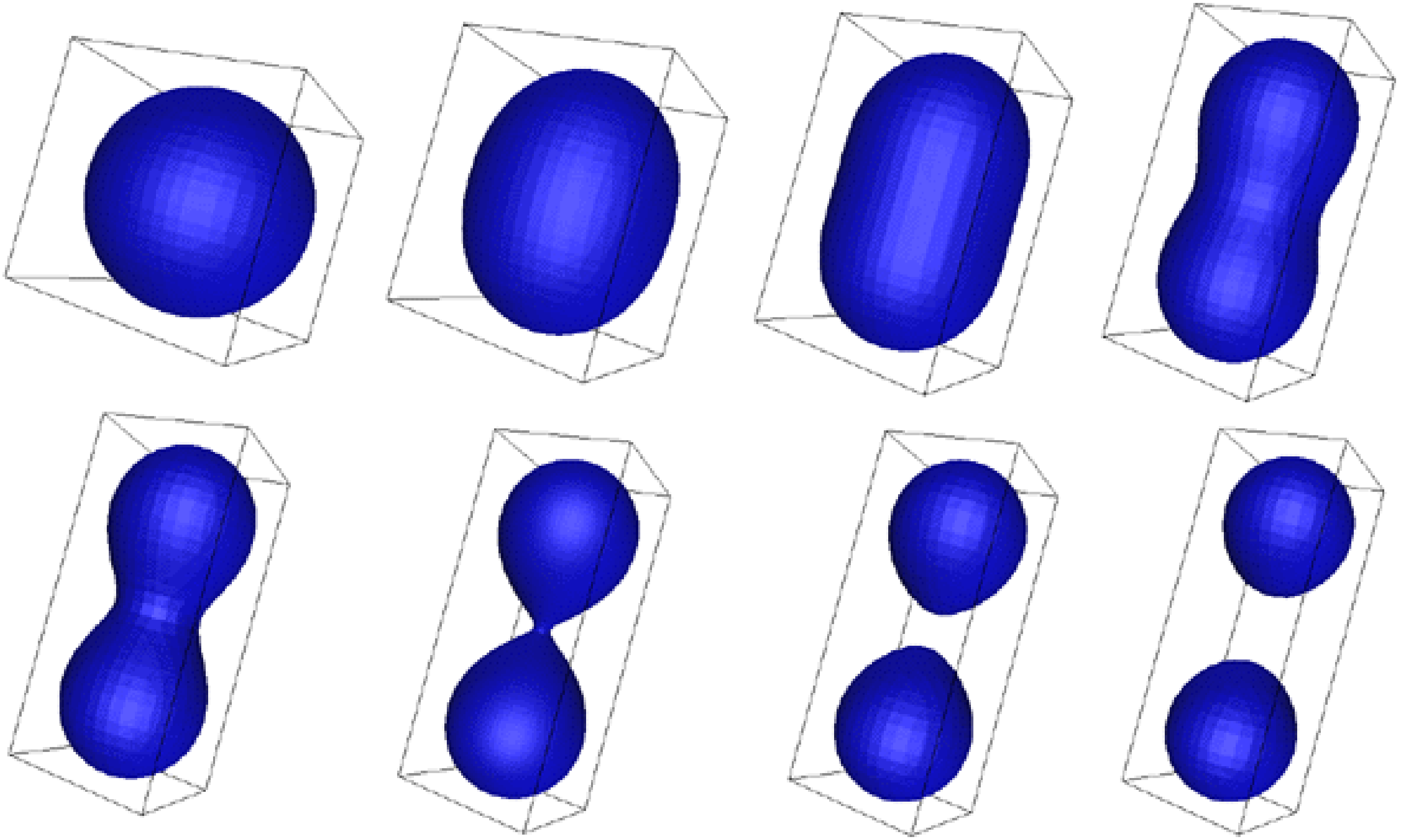


Kazhdan

Implicit Surface Properties

Efficient topology changes

- Surface is not represented explicitly!

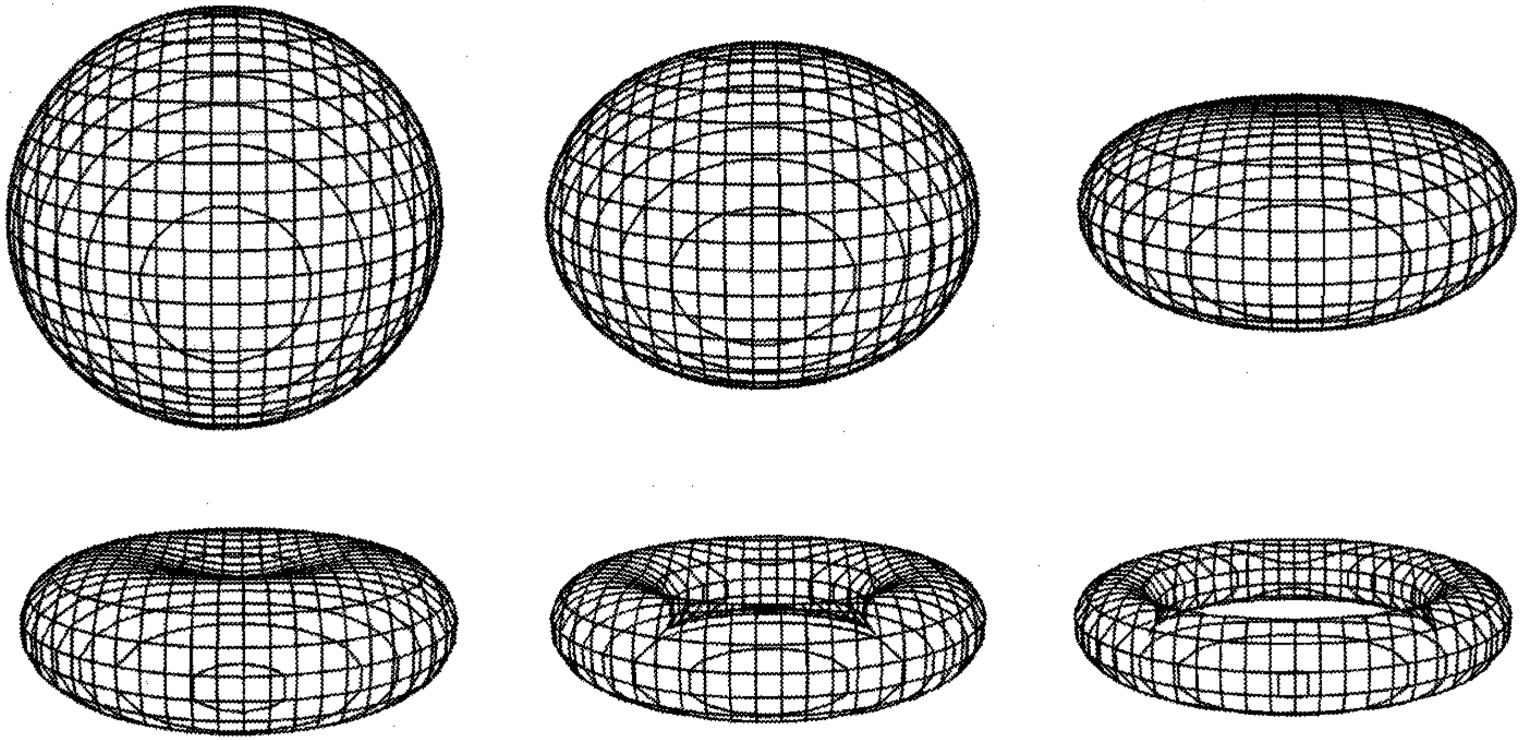


Bourke

Implicit Surface Properties

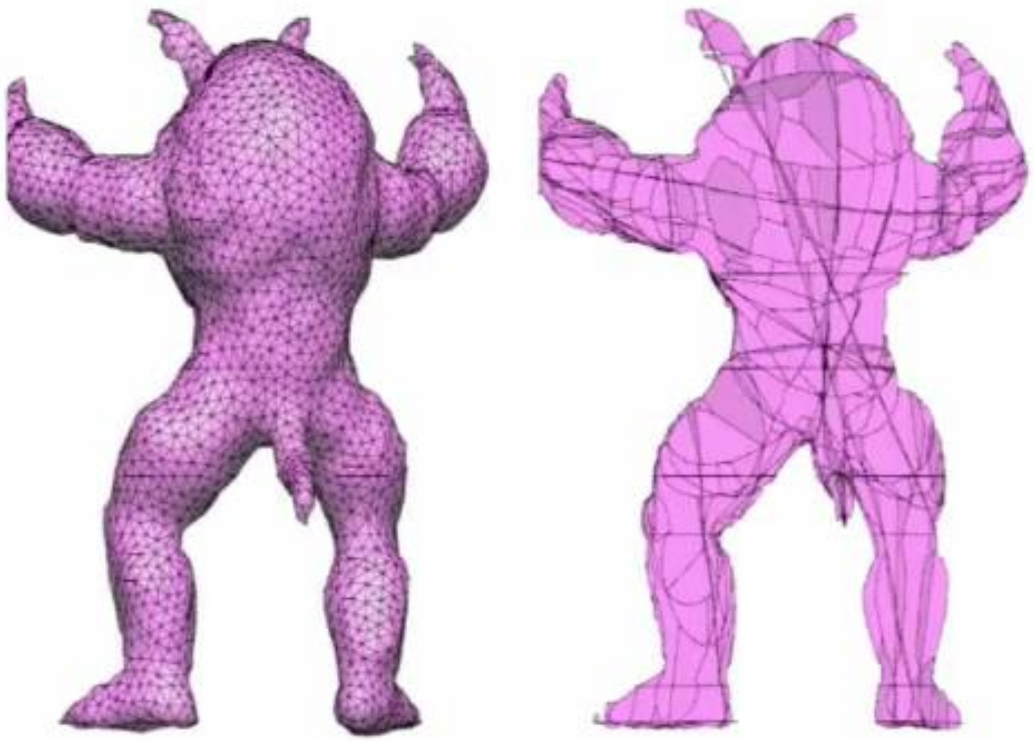
Efficient topology changes

- Surface is not represented explicitly!



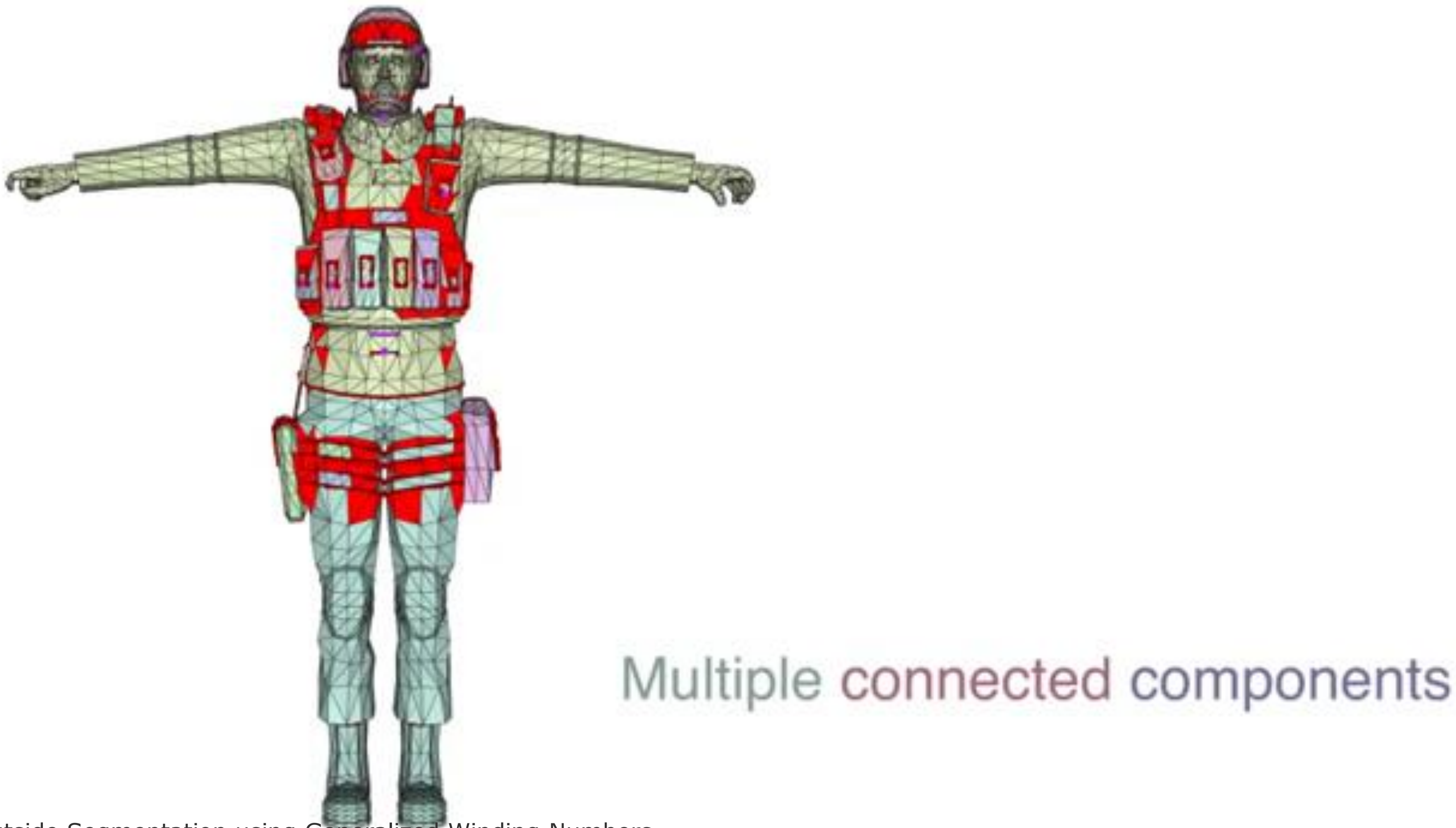
Bloomenthal

Applications



Online Reconstruction of 3D Objects from Arbitrary Cross-Sections
[Bermano et al. 2011]

Applications



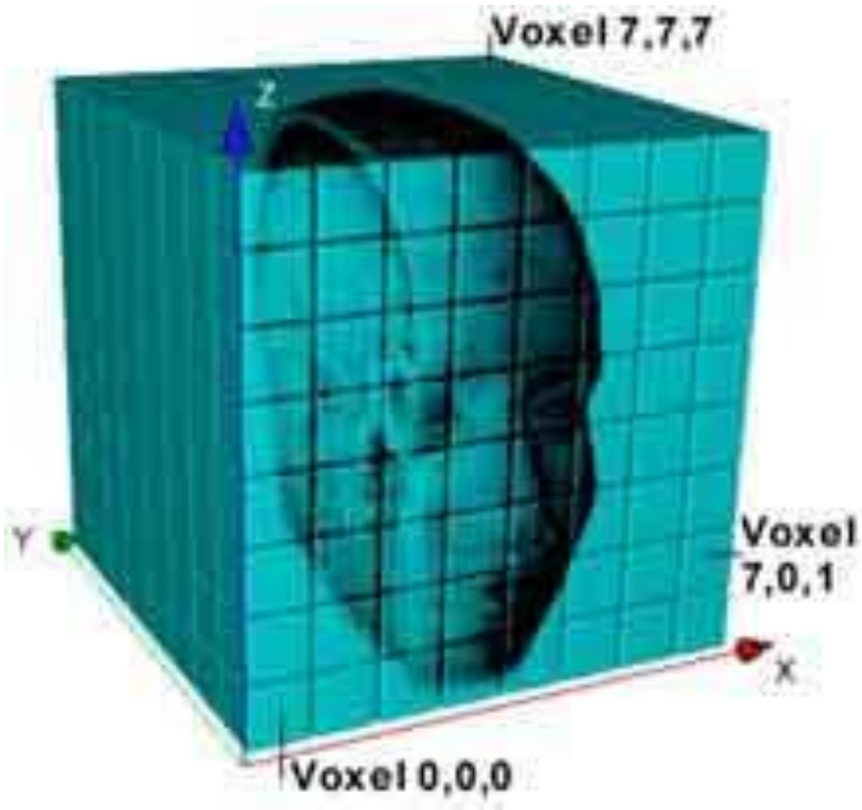
Robust Inside-Outside Segmentation using Generalized Winding Numbers
[Jacobson et al. 2013]

3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

Voxels

- Regular array of 3D samples (like image)
  - Samples are called *voxels* (“**vo**lume pi**xels**”)



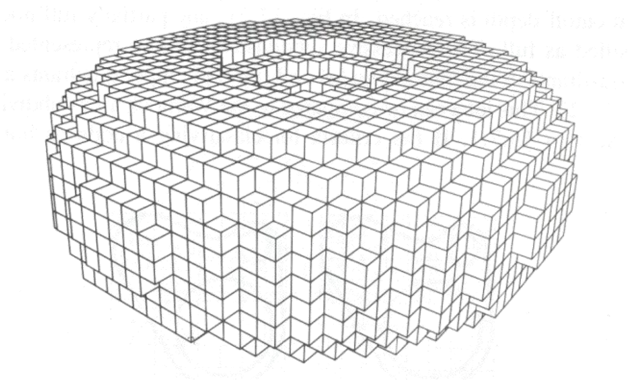
www.volumegraphics.com

Voxel grid

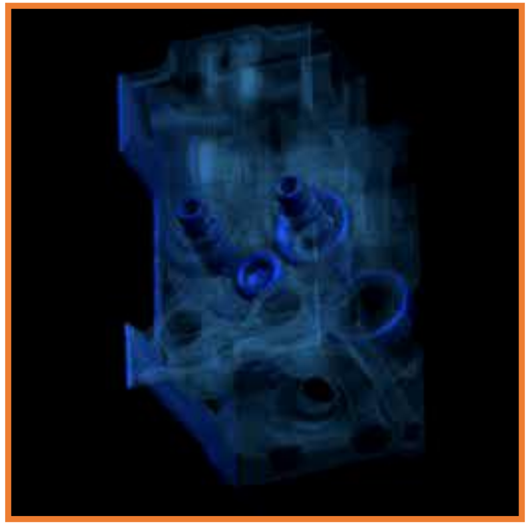
Uniform volumetric grid of samples:

- Occupancy
  (object vs. empty space)
- Density
- Color
- Other function
  (speed, temperature, etc.)

- Often acquired via
  simulation or from
  CAT, MRI, etc.



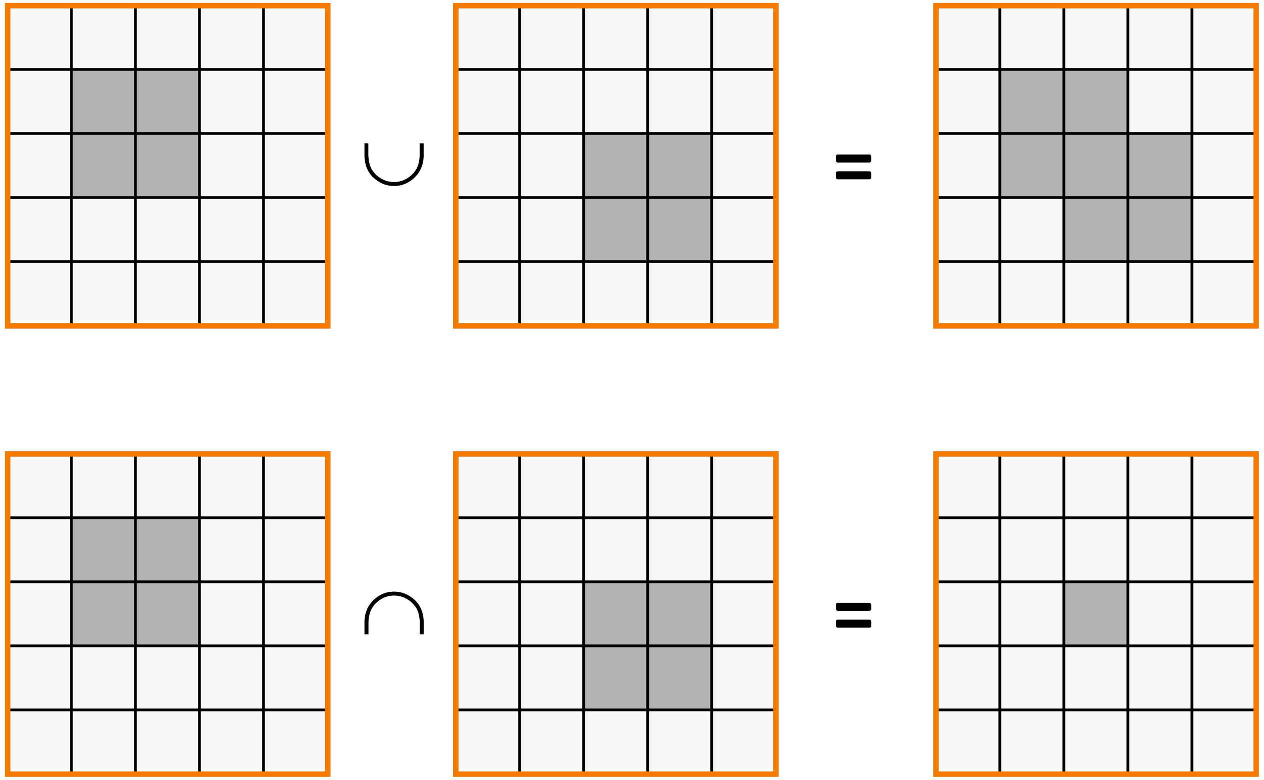
FvDFH Figure 12.20



Stanford Graphics Laboratory

Voxel Boolean Operations

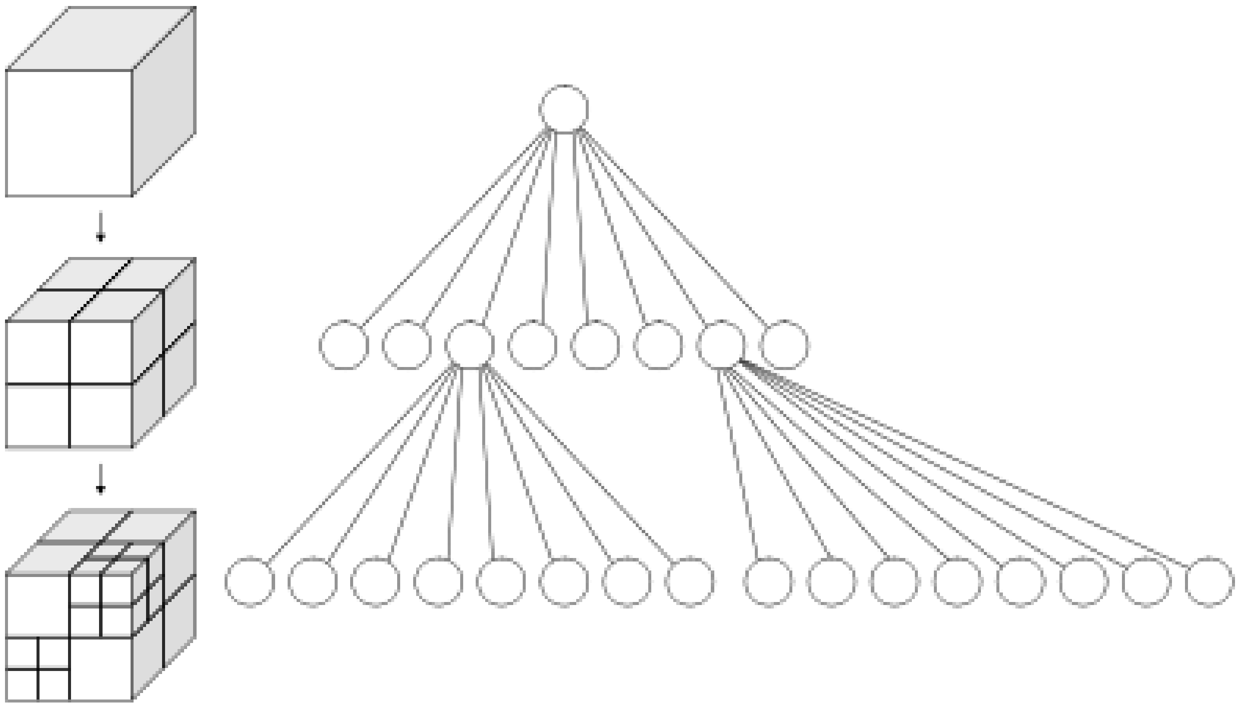
- Compare objects voxel by voxel
  - Trivial

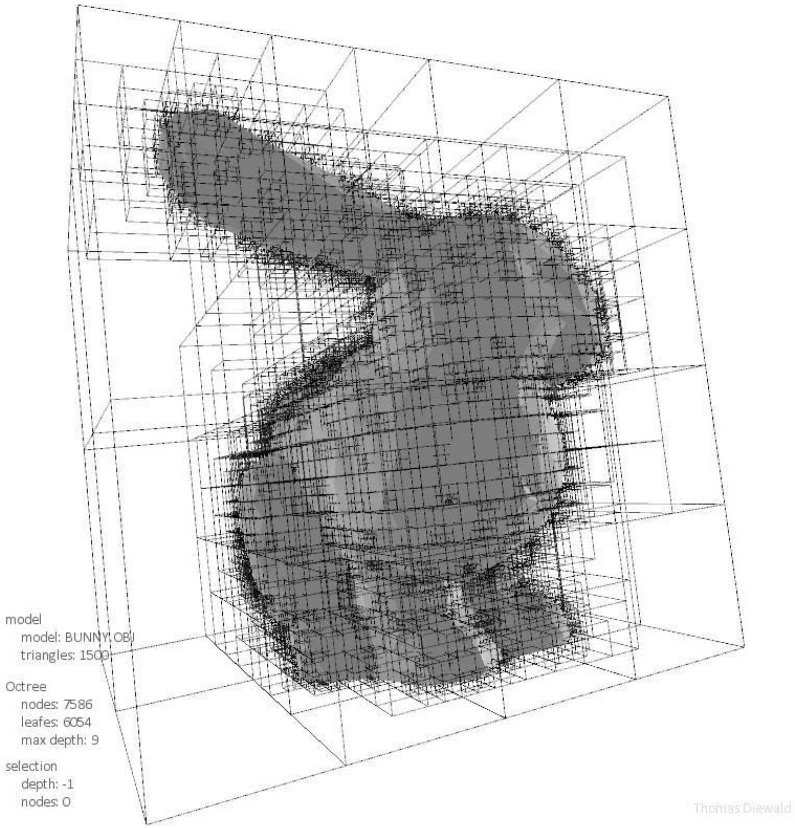


Octree

The adaptive version of the voxel grid

- Significantly more space efficient
- Makes operations more cumbersome

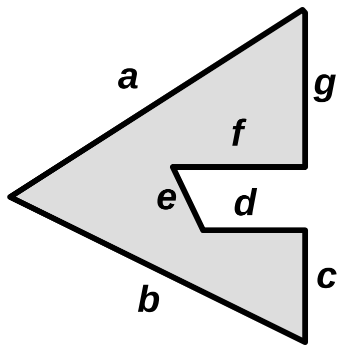




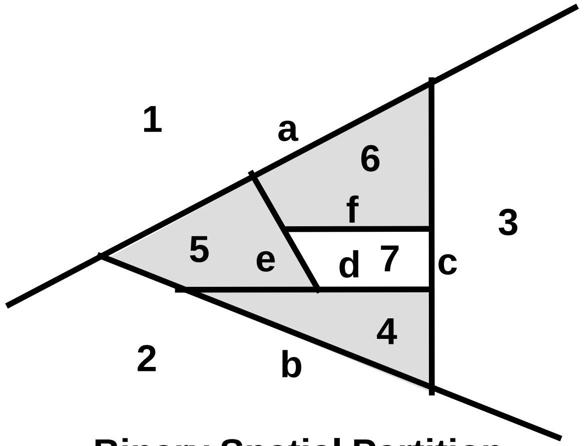
BSP Tree

Hierarchical **B**inary **S**pace **P**artition with solid/empty cells labeled

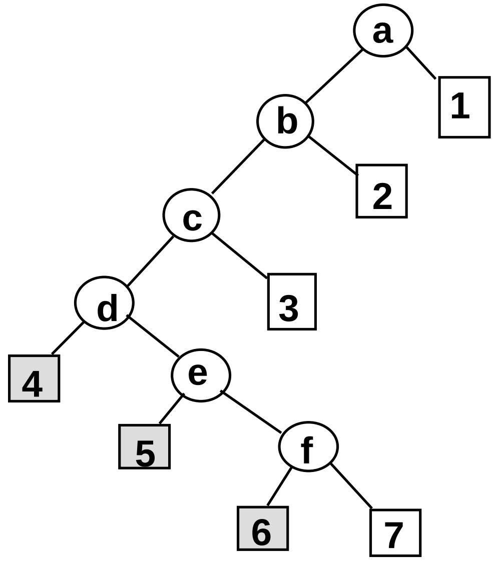
- Constructed from polygonal representations



Object



Binary Spatial Partition

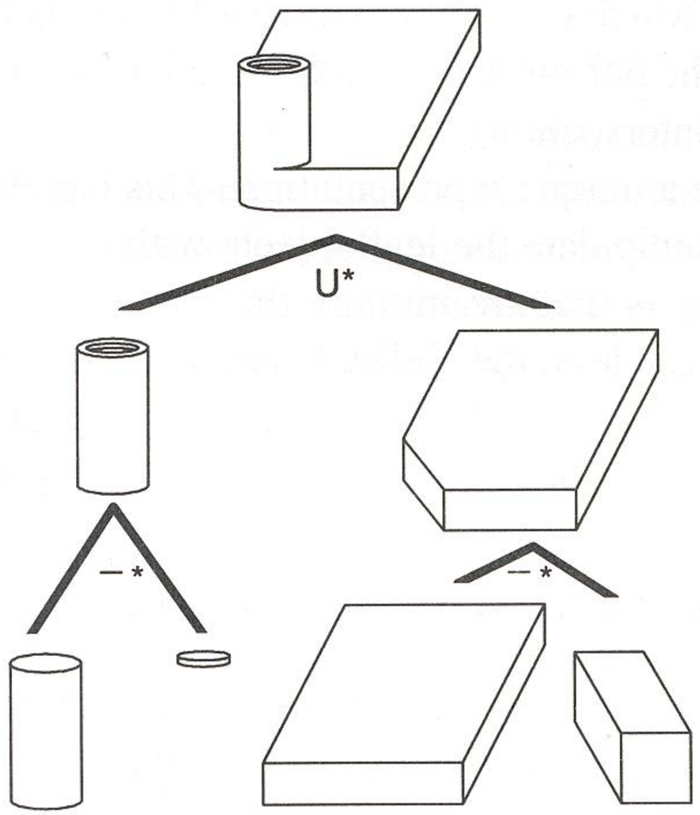


Binary Tree

Naylor

Constructive Solid Geometry (CSG)

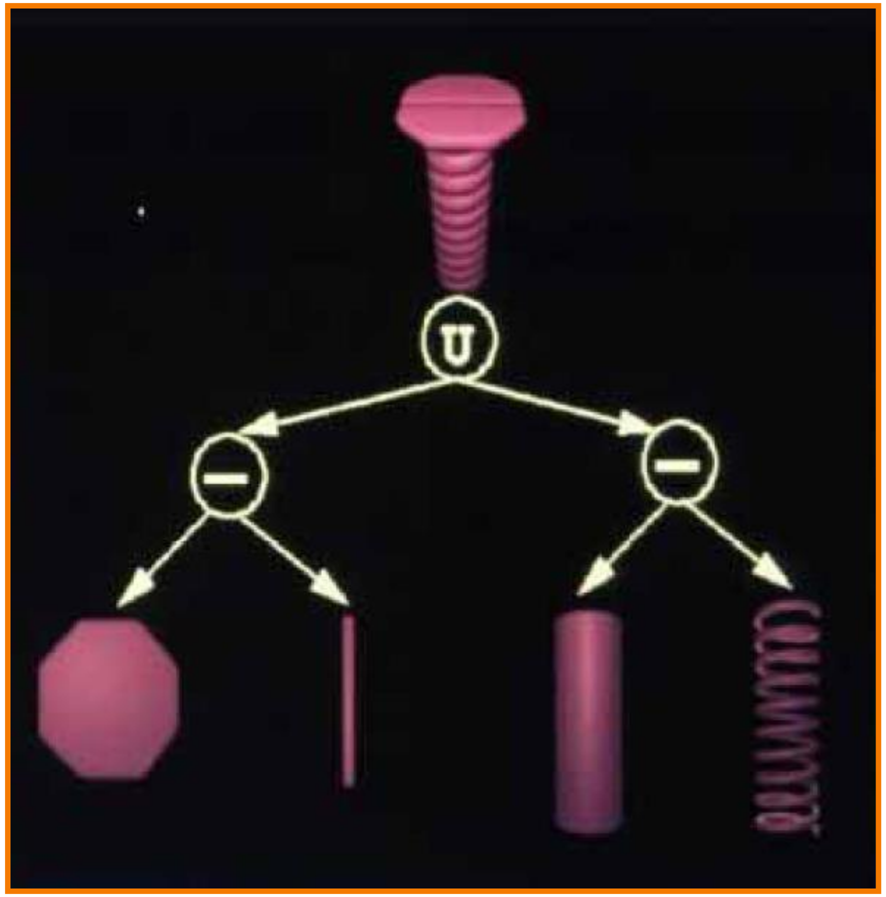
- Represent solid object as hierarchy of boolean operations
  - Union
  - Intersection
  - Difference



FvDFH Figure 12.27

CSG

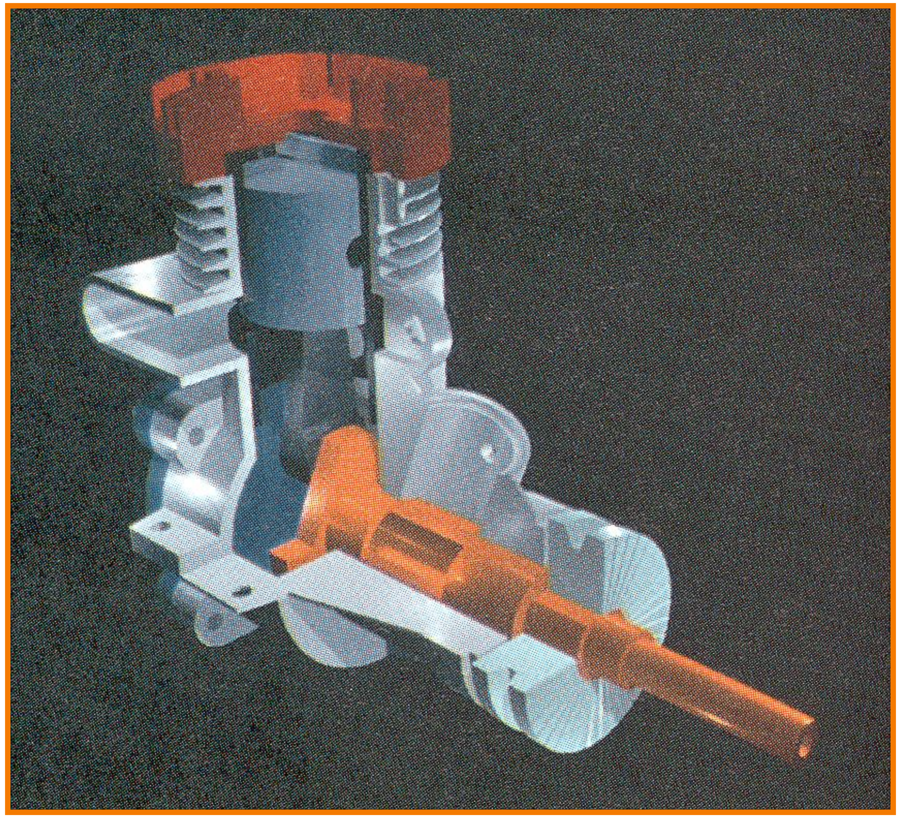
- Interactive modeling programs
  - Intuitive way to design objects



SUNY Stoney Brook

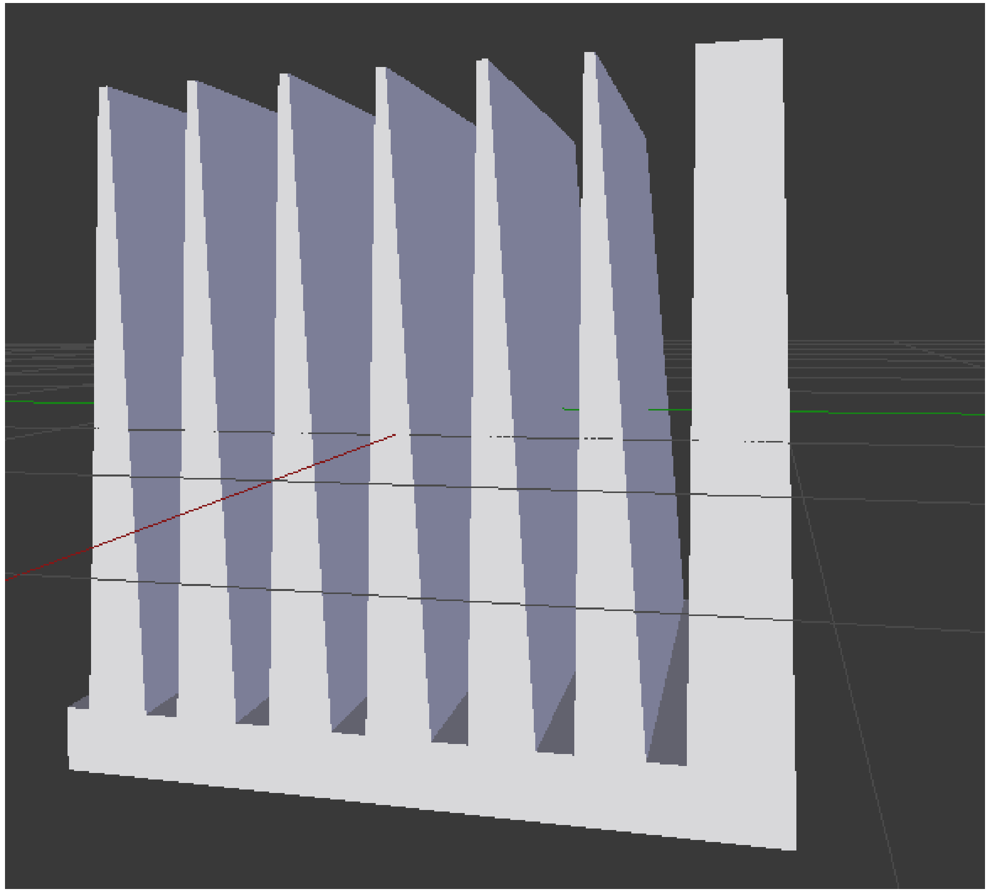
CSG

- Interactive modeling programs
  - Intuitive way to design objects



H&B Figure 9.9

CSG



3D Object Representations

- Points
  - Range image
  - Point cloud
- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit
- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep
- High-level structures
  - Scene graph
  - Application specific

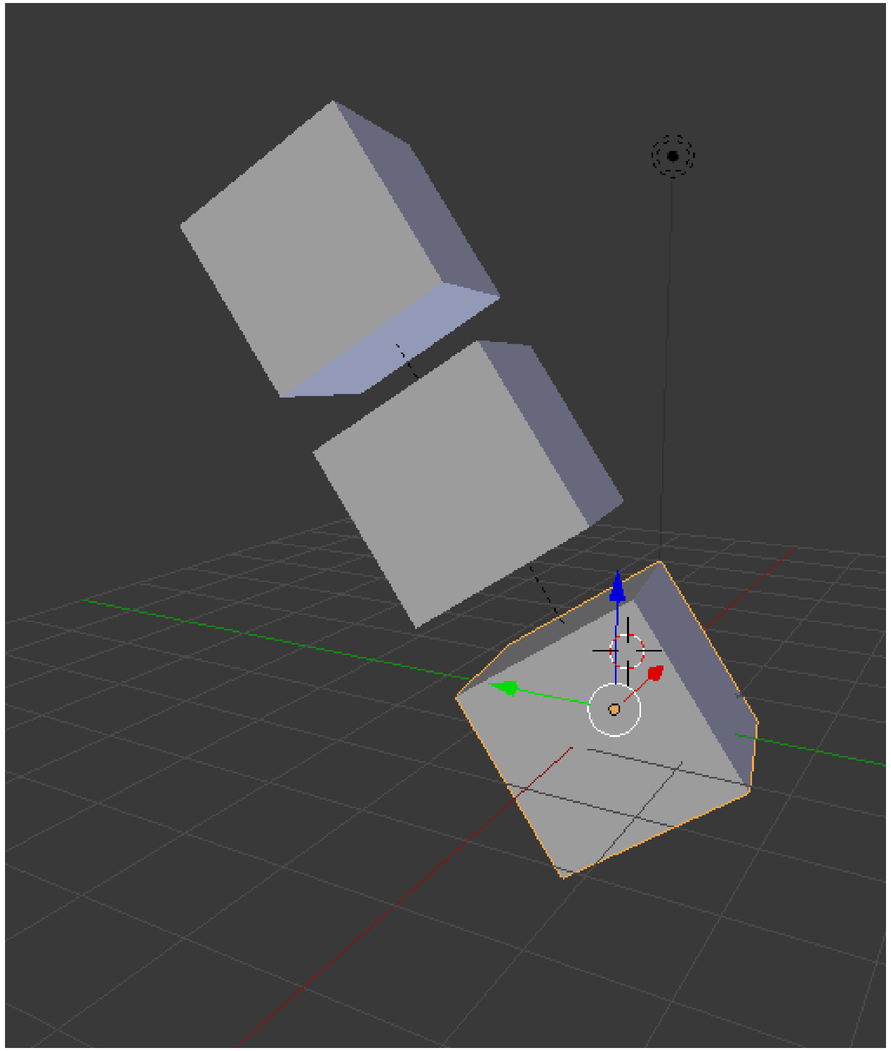
Scene Graph

Union of objects at leaf nodes

Bell Laboratories

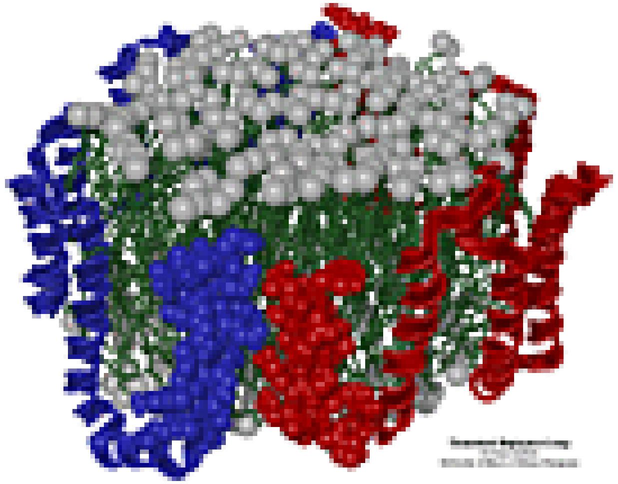
avalon.viewpoint.com

Scene Graph



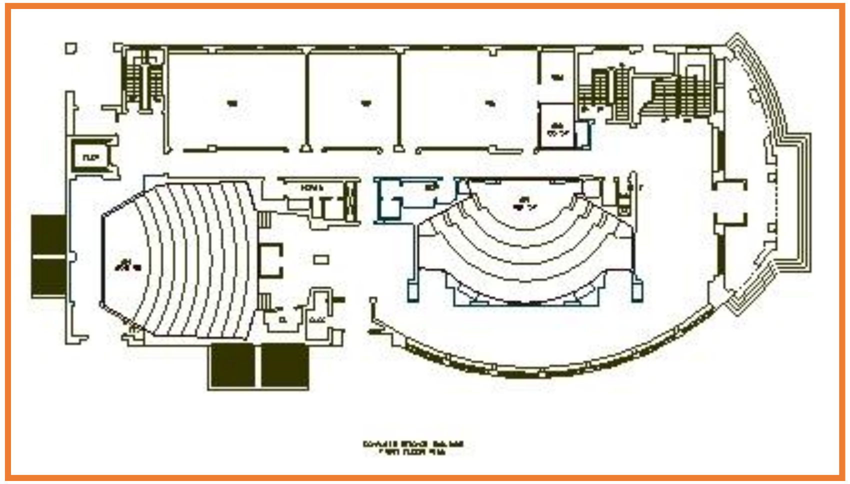
blender

Application Specific



Apo A-1

(Theoretical Biophysics Group,
University of Illinois at Urbana-Champaign)



Architectural Floorplan

(CS Building, Princeton University)

Computational Differences

- Efficiency
  - Representational complexity (e.g. surface vs. volume)
  - Computational complexity (e.g. $O(n^2)$ vs $O(n^3)$)
  - Space/time trade-offs (e.g. tree data structures)
  - Numerical accuracy/stability (e.g. degree of polynomial)
- Simplicity
  - Ease of acquisition
  - Hardware acceleration
  - Software creation and maintenance
- Usability
  - Designer interface vs. computational engine