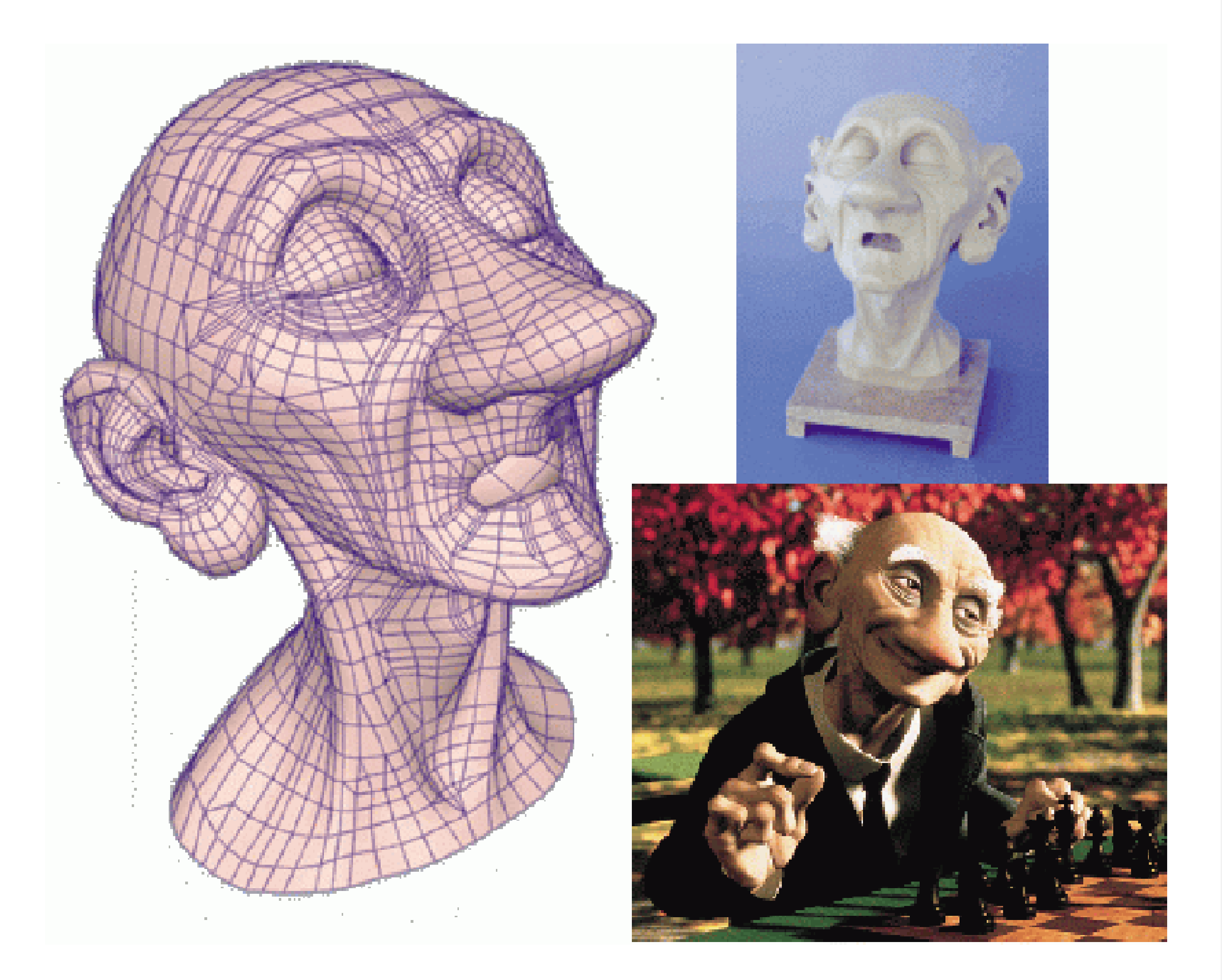
Object representation

Geri’s Game

- Pixar 1997
- Subdivision surfaces
- Polhemus 3d scan
- Over 700 controls



Quick test #1

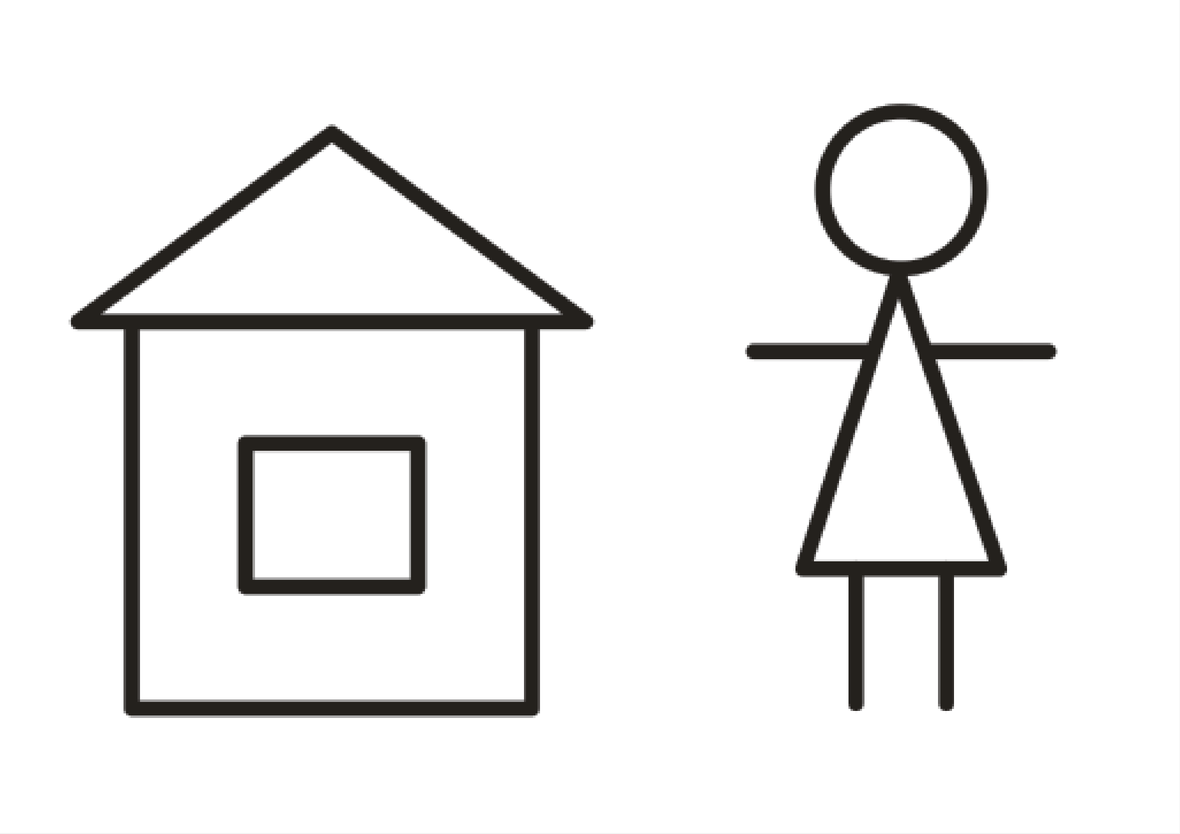
- Describe the picture

Quick test #1

- Describe the picture

Quick test #2

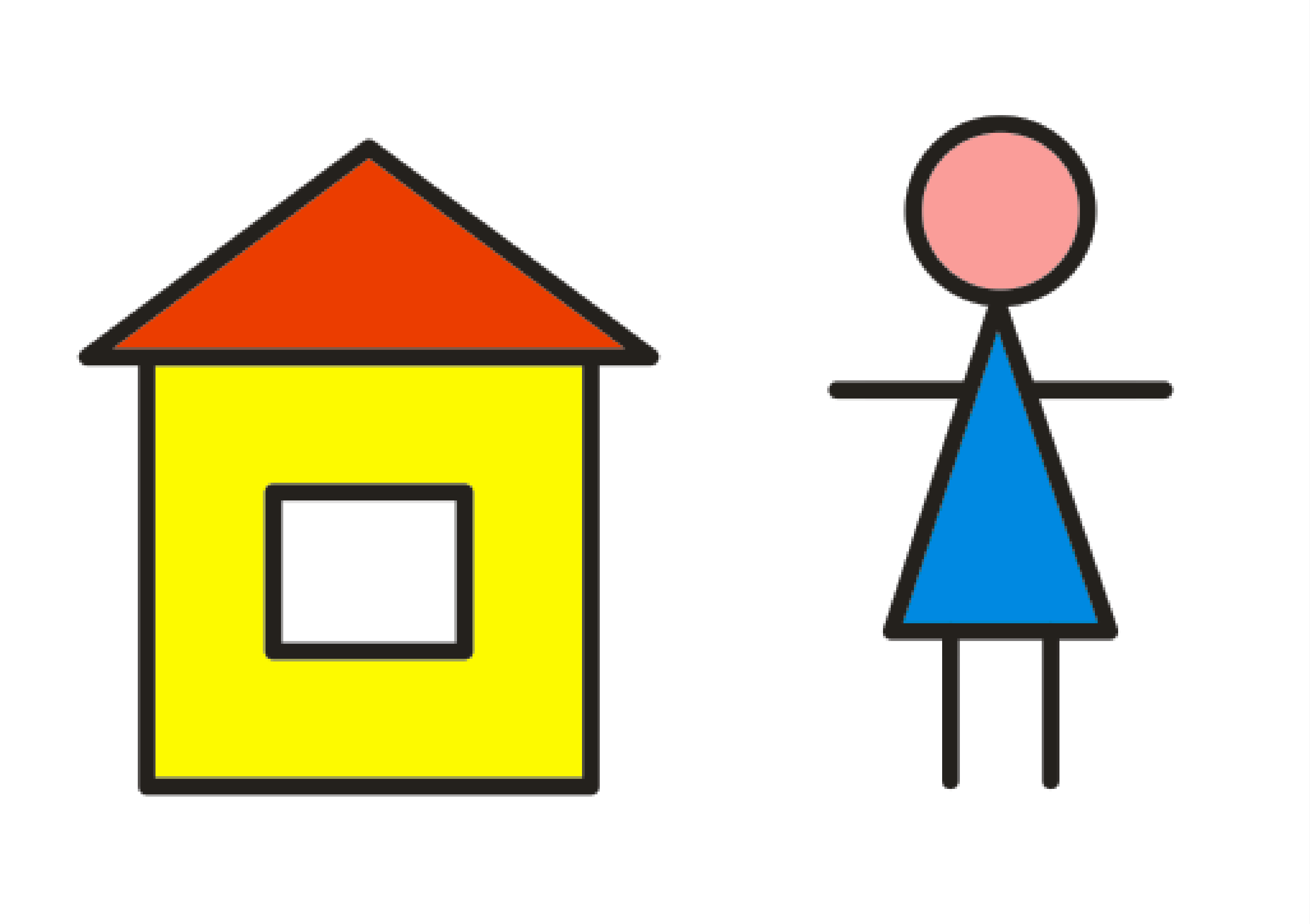
- Volunteers: Describe the image to others
- Others: Reproduce the image



Semantic vs. numeric

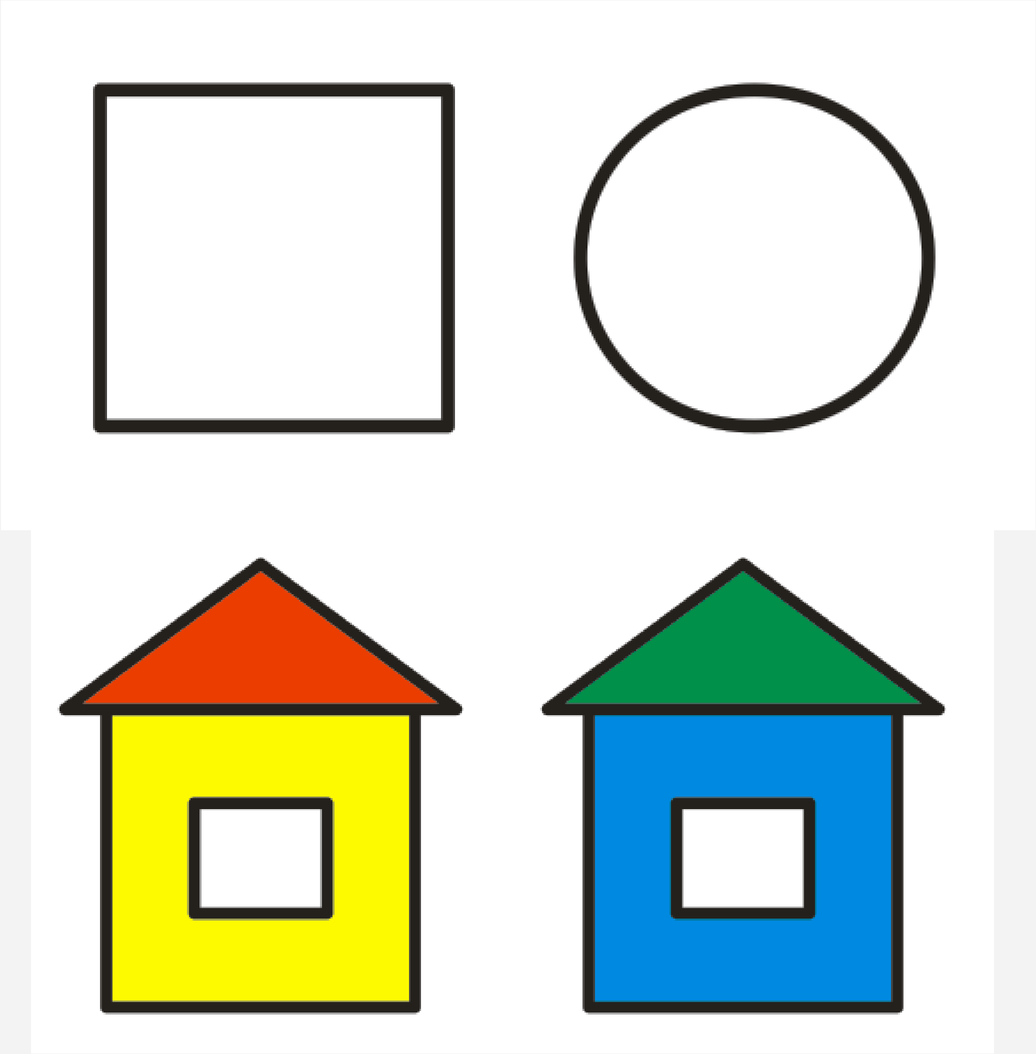
- Humans – semantic representation
  - – concepts, notions, meanings, emotions...
  - – imprecise, ambiguous
- Computers – numeric representation
  - – exact, mathematical, straightforward

Detailed representation



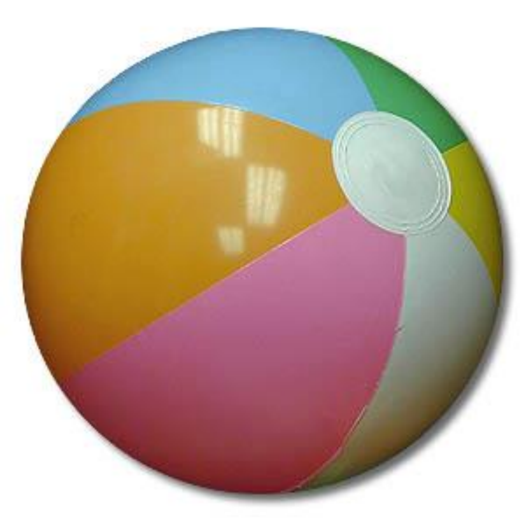
Object properties – basic

- What object properties there are?



Object properties - advanced

- Physical object properties
  - mass, stiffness, elasticity



- Material properties
  - shininess, roughness
  - light behavior
  - friction
  - etc.

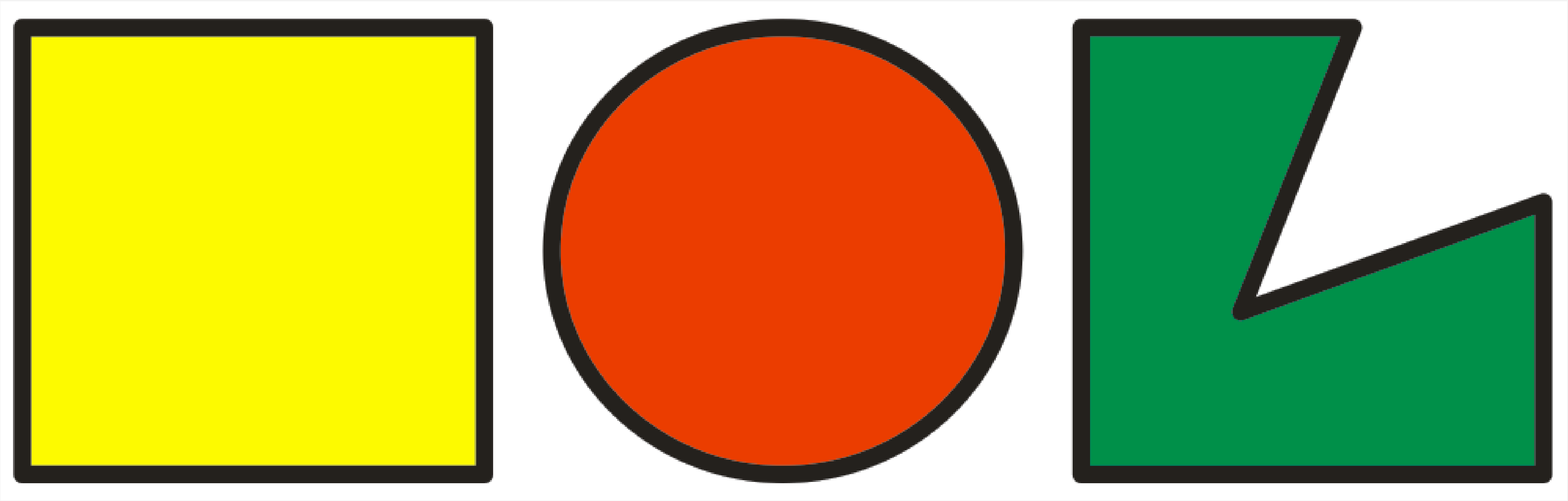
We'll deal with materials later

Object definition

- Geometry
  - plus model transformation (local → global)
- Material
  - color, shininess, index of refraction
- Body properties
  - weight, elasticity...

2D objects

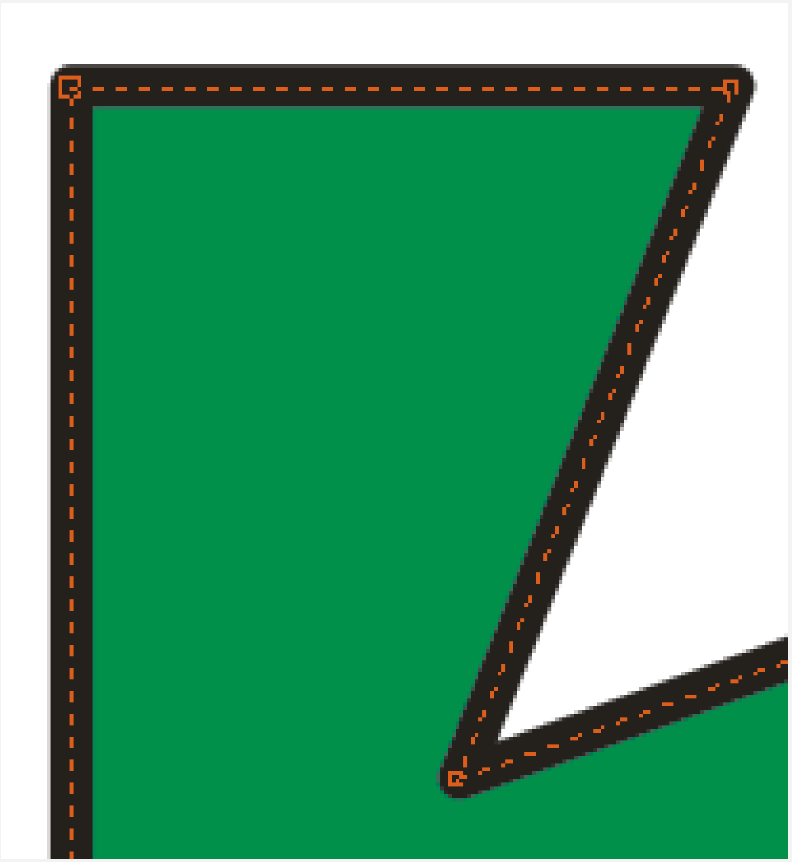
- Let’s define these objects



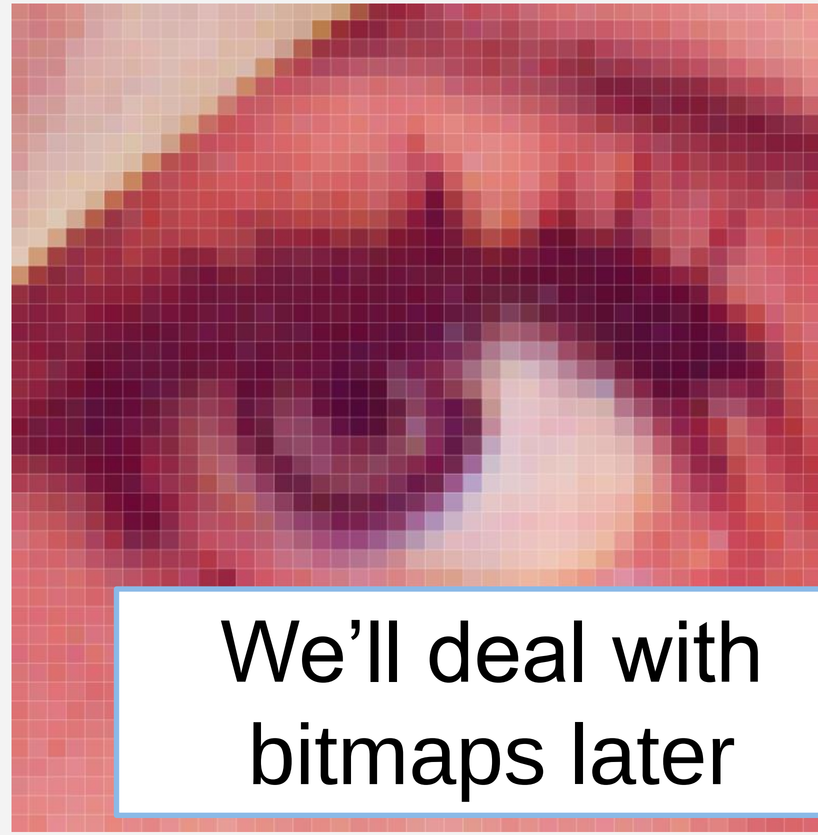
2D objects

- Let’s define these objects

Shapes vs. bitmaps



Shapes, vectors, curves, parametric, implicit



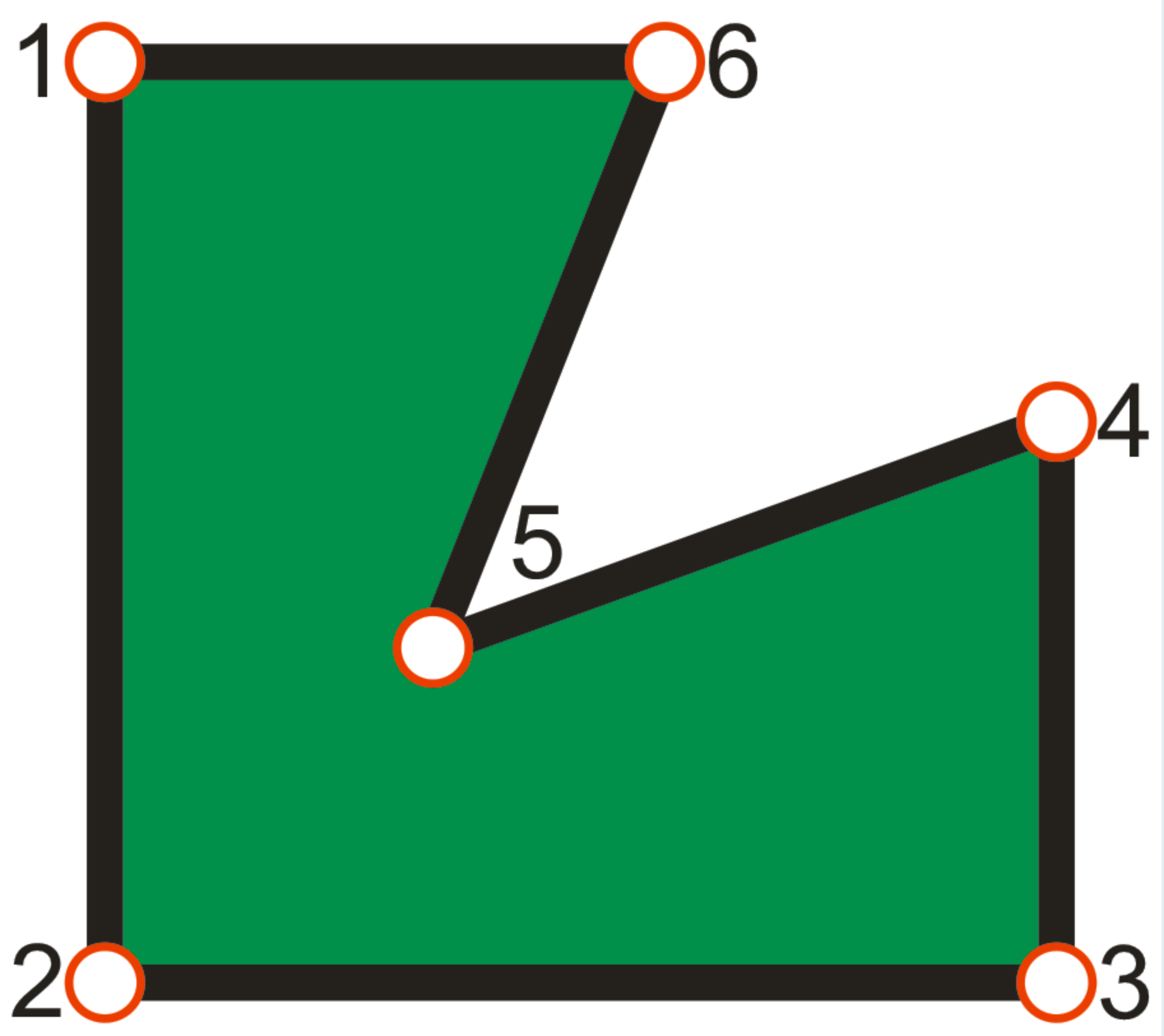
We'll deal with bitmaps later

Bitmaps, raster, pixels, explicit

2D geometry

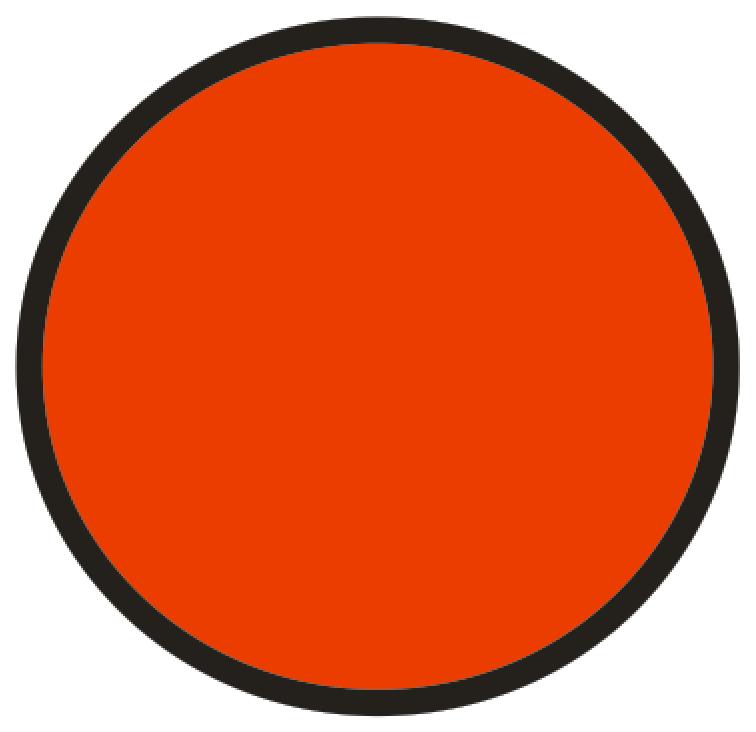
Polygons

- Control vertices
  - x,y coordinates
  - in order
  - CCW
- Edges
  - width
  - shape
  - style (solid, dotted etc.)



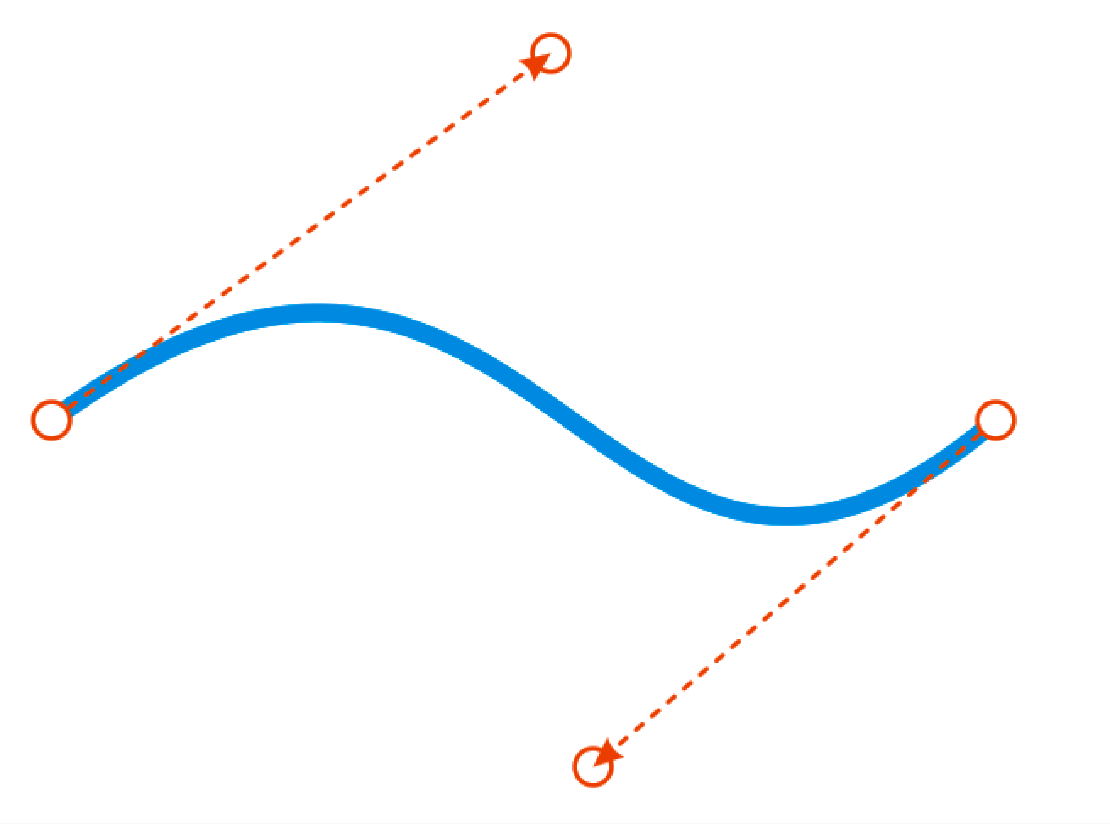
Curves

- Primitives – circle, ellipse
- General – parametric curves



Parametric curves

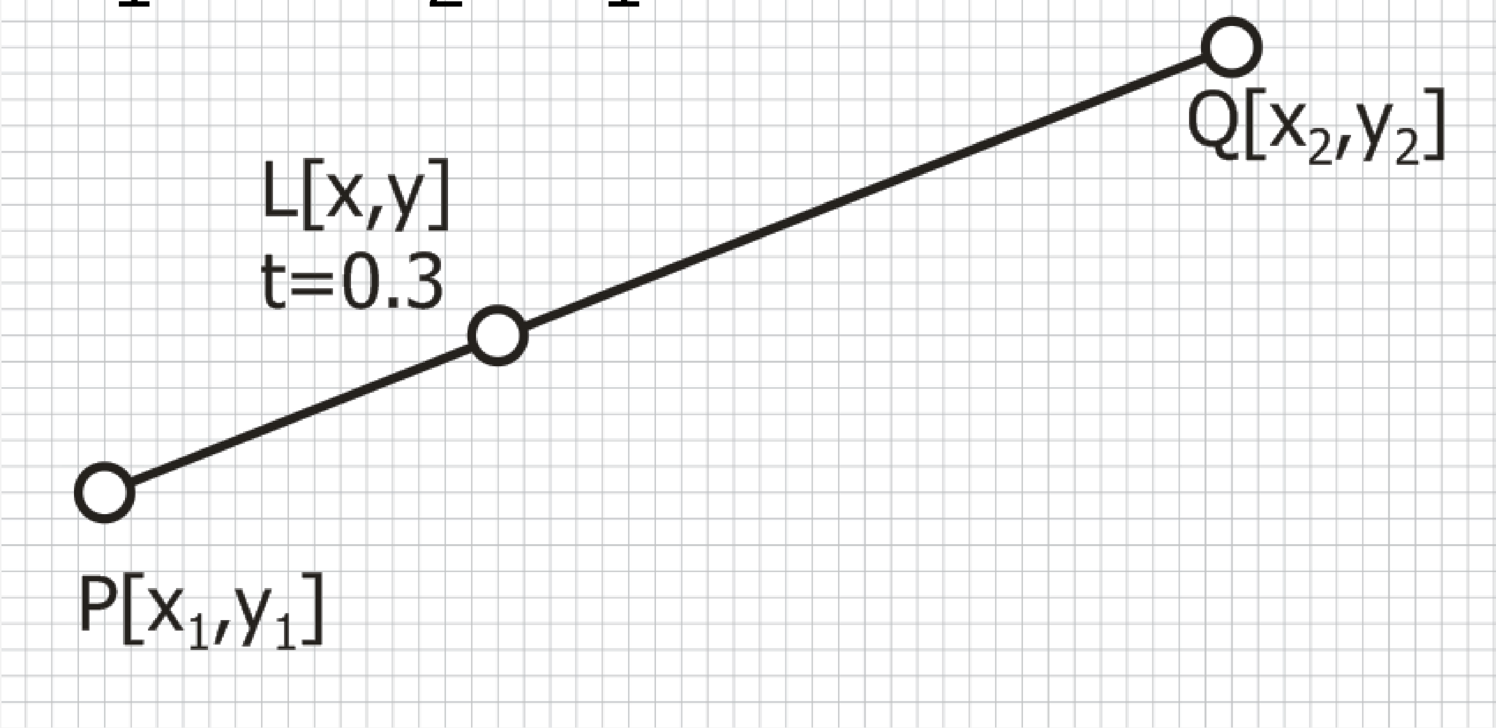
- Bezier curves
- Spline curves



Remember: line equation

- Line **P-Q** where $P = [x_1, y_1]$, $Q = [x_2, y_2]$
- $x = x_1 + t * (x_2 - x_1)$
- $y = y_1 + t * (y_2 - y_1)$

} **L** = **P** + t*(**Q**-**P**)

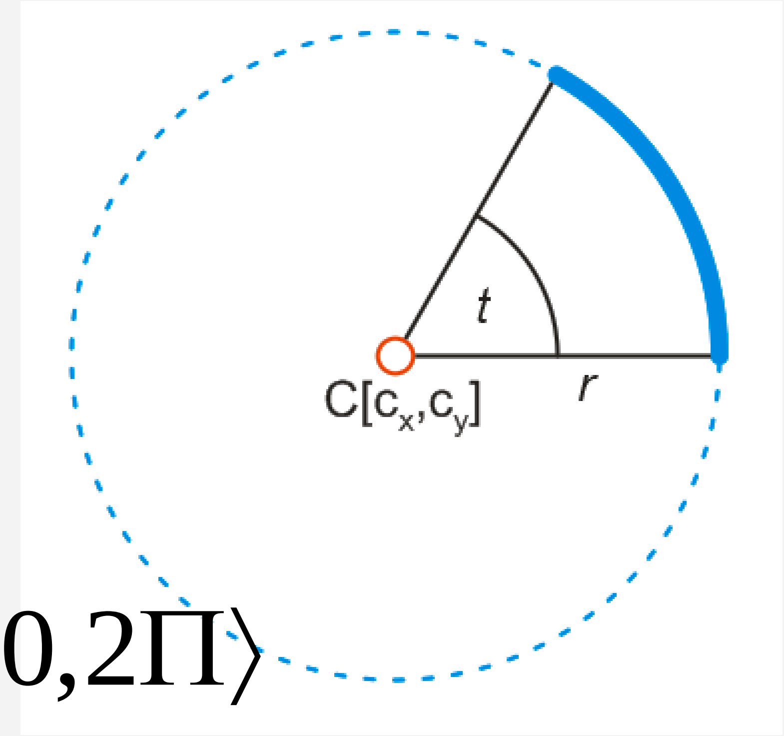


Parametric curve

- Generalization
- $x = f_1(t)$
- $y = f_2(t)$

$$\left.\begin{aligned} x &= f_1(t) \\ y &= f_2(t) \end{aligned}\right\} C(x,y) = f(t)$$

- Example – Circle

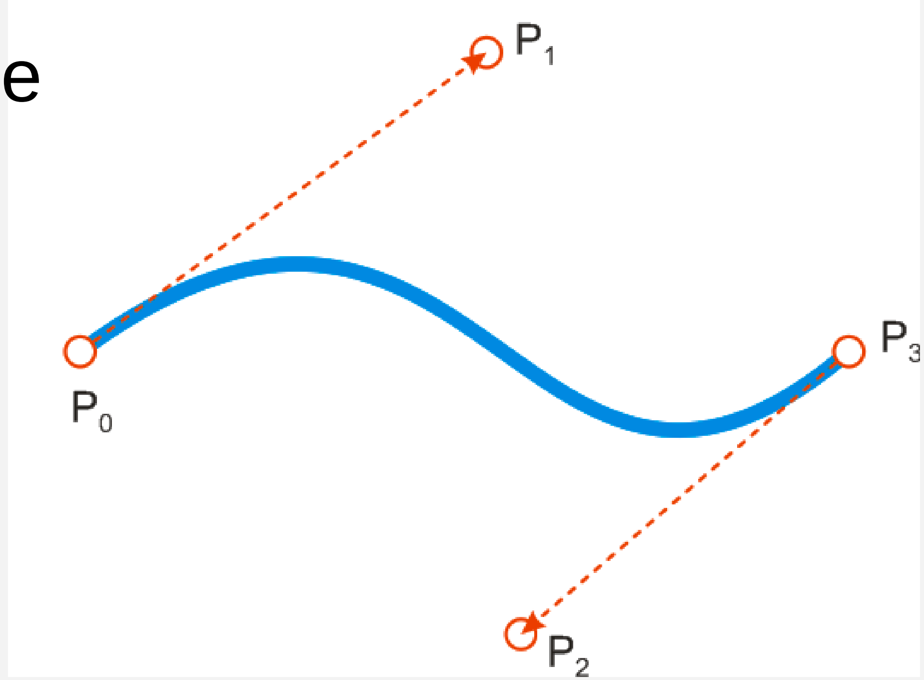


$$x = c_x + r.\cos(t)$$

$$y = c_y + r.\sin(t), \quad t \in \langle 0, 2\Pi \rangle$$

Cubic Bezier curves

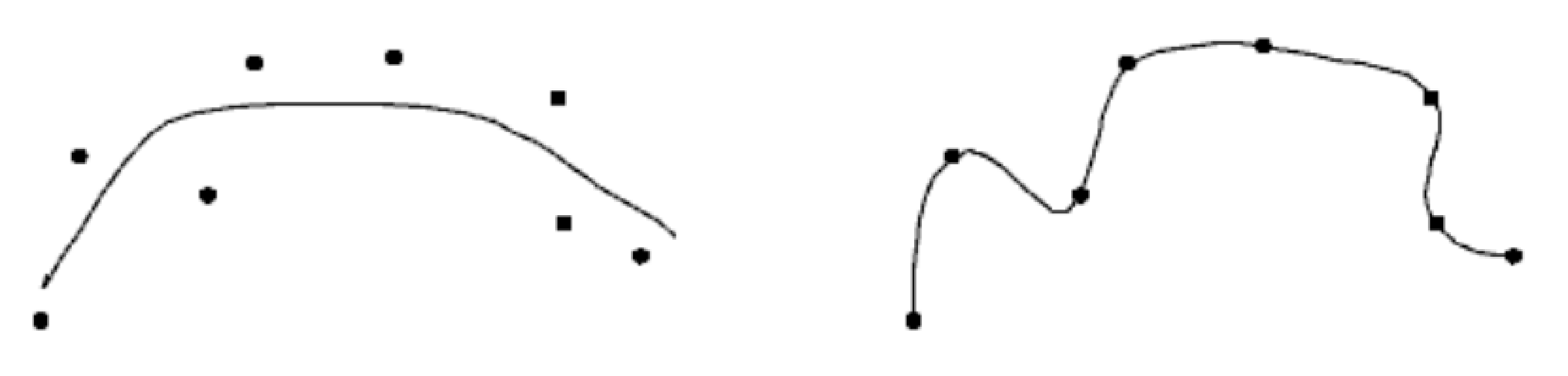
- 4 control points C_0, C_1 , C_2 , C_3
  - C_0 and C_3 endpoints
  - C_1 and C_2 define shape



- Can be expressed in matrix form

$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , \ t \in [0,1].$$

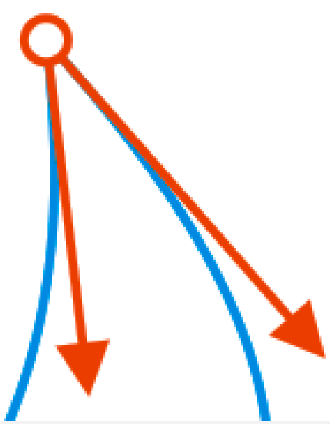
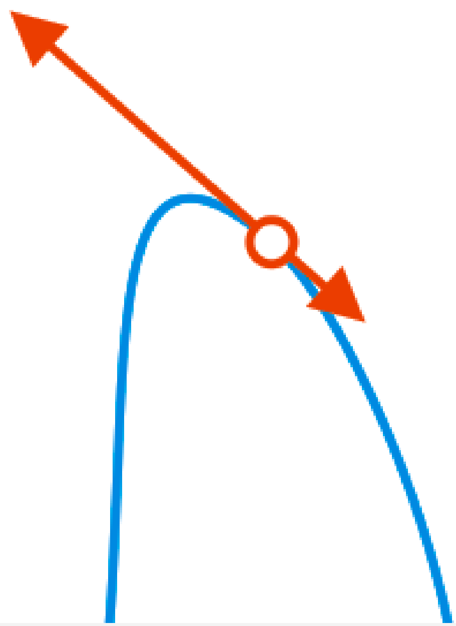
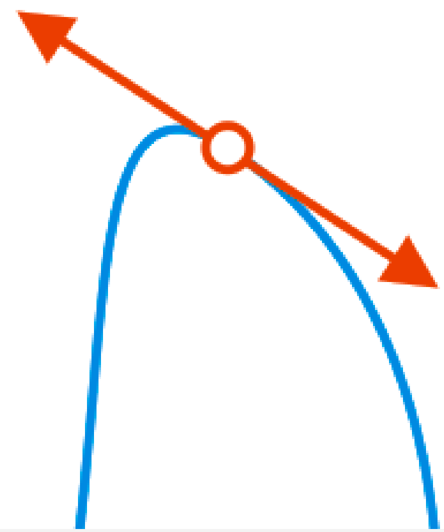
Approximation / interpolation



- http://lubovo.misto.cz/curves/

Interpolation curves

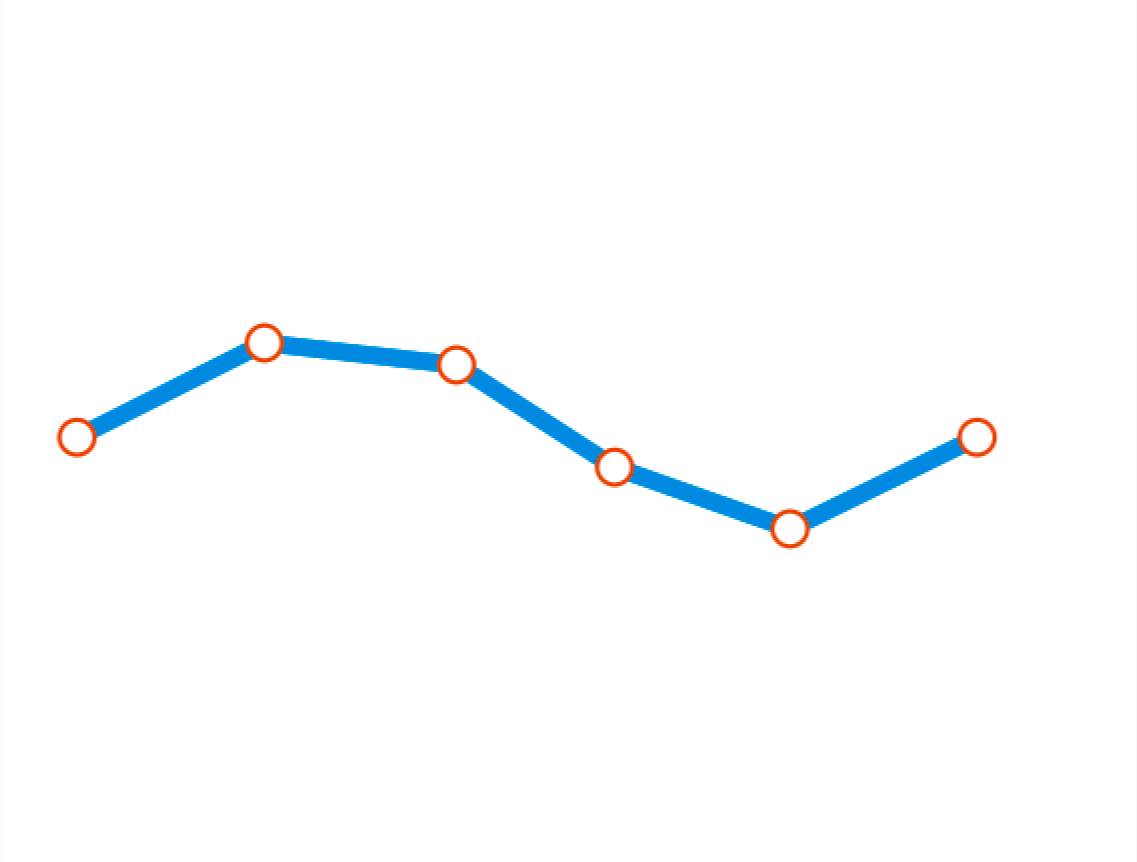
- Control vertices + tangent vectors
  - corner (cusp)
  - smooth
  - symmetrical

- Usually cubic splines – good manipulation

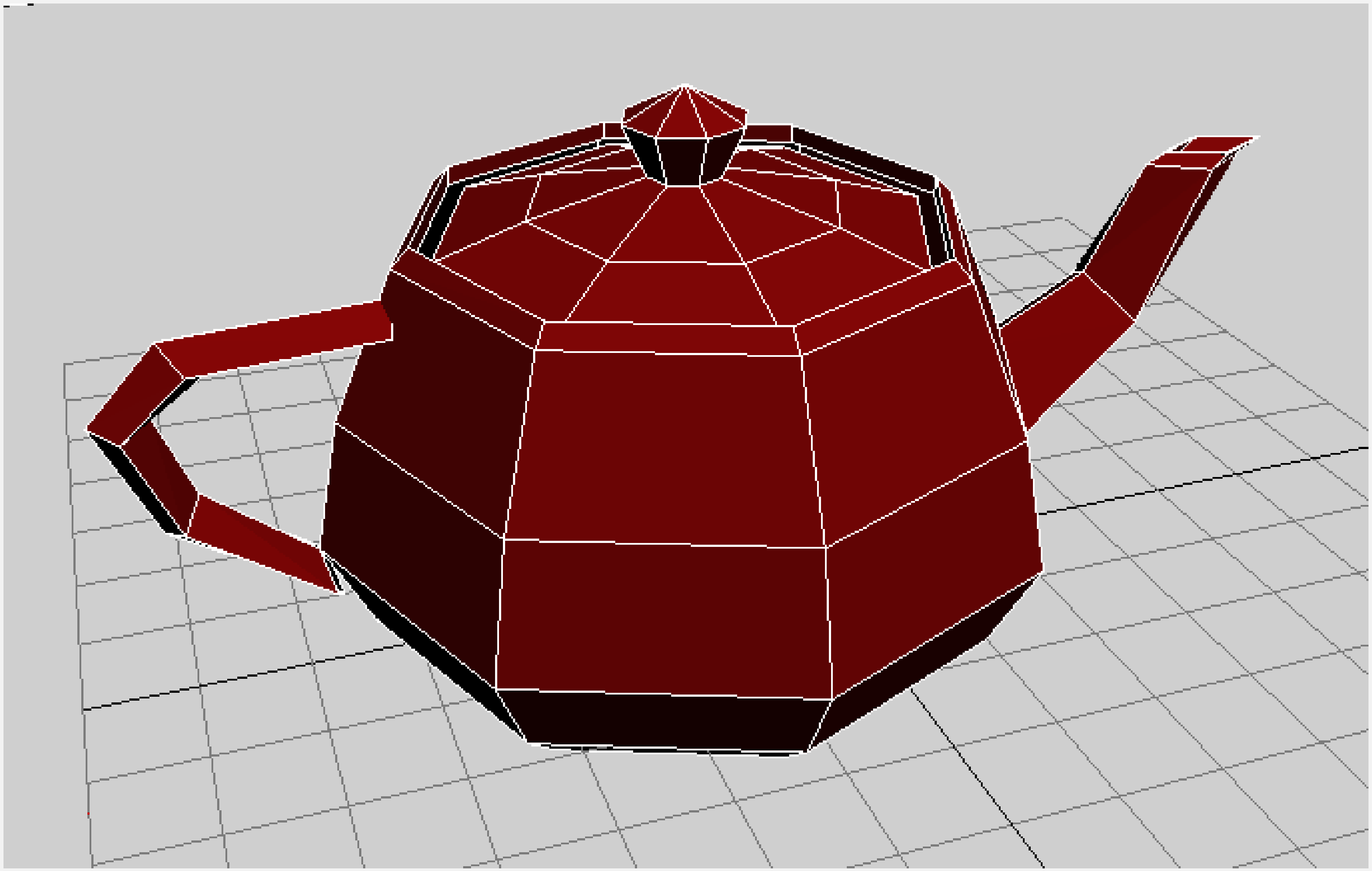
Parametric → polygonal

- C(x,y) = f(t)
- t = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0

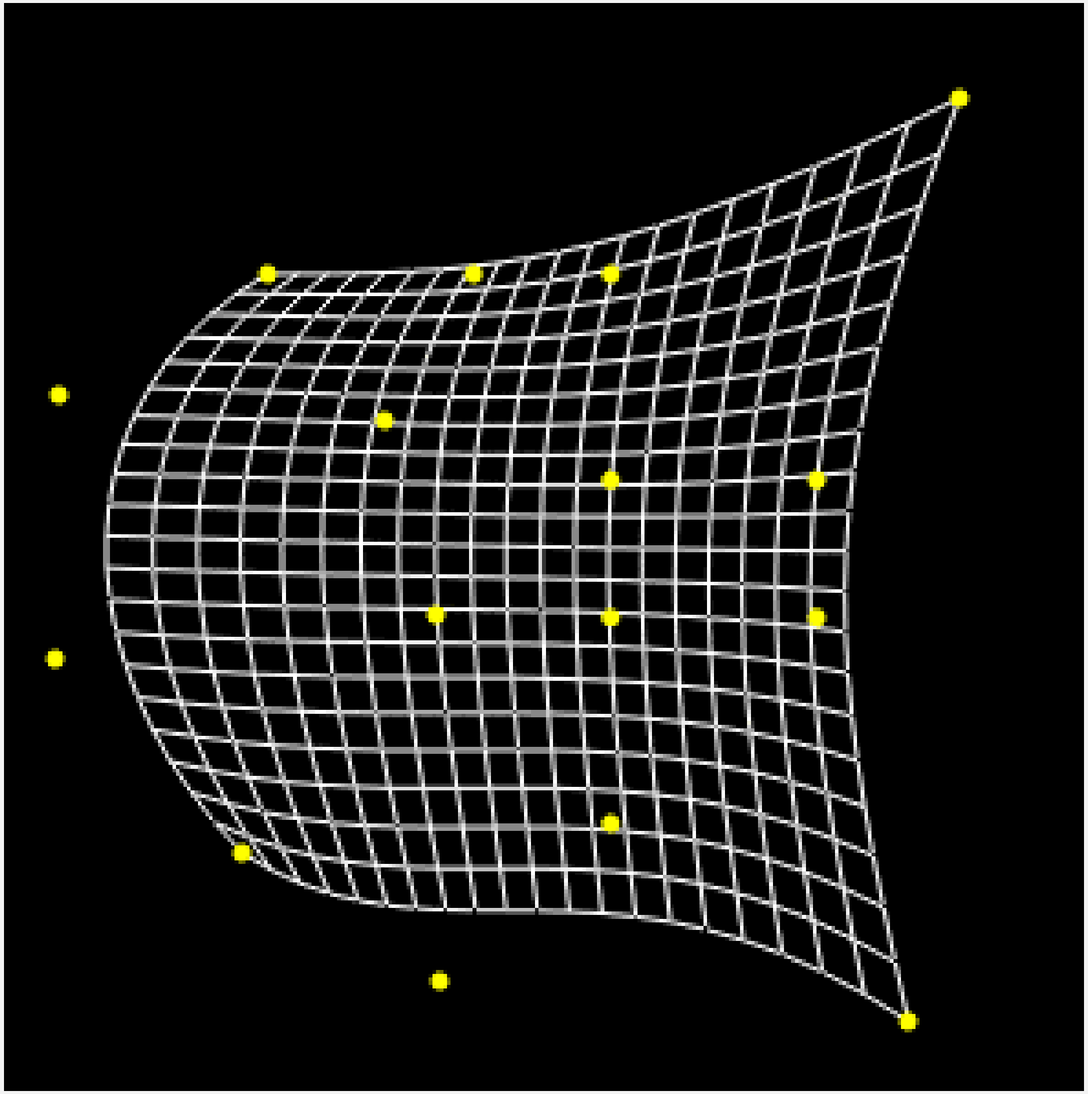


3D geometry

From polygons to polyhedra



From curves to surfaces



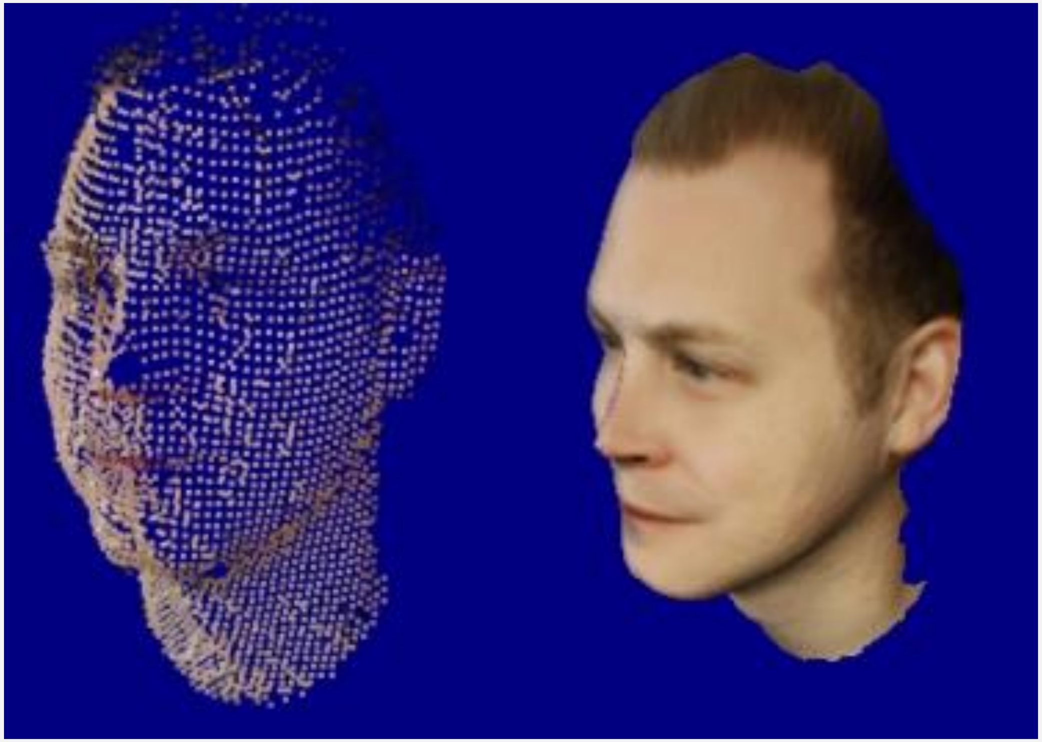
Boundary representation

- Only the surface of the object is described
- No information about the inside

- Point cloud
- Wireframe
- Polygonal mesh
- Parametric surfaces
- Subdivision surfaces
- Implicit surfaces

Point cloud

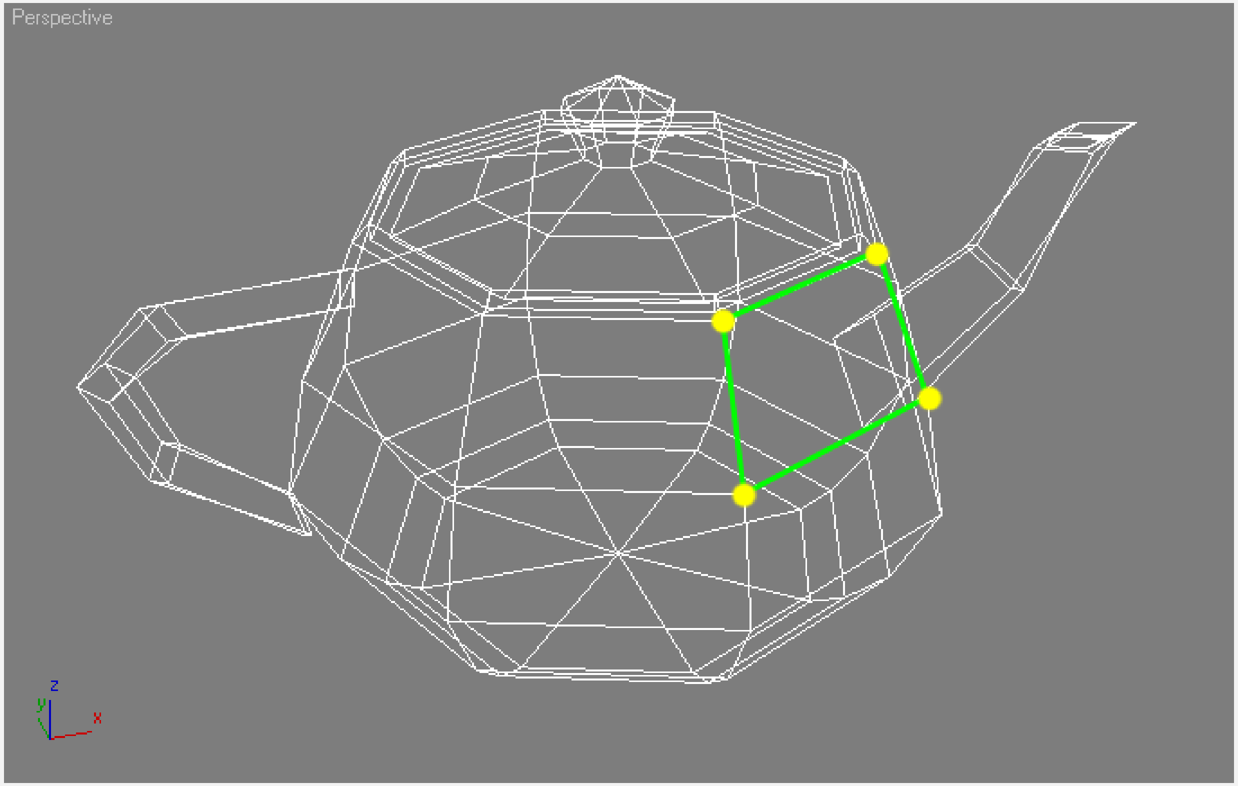
- Set of points located on object's surface
- Usually obtained by 3D scanning
- Connectivity information?



http://www.photomodeler.com

Wireframe

- Set of vertices $V(x,y,z)$
- Edges $E(V_i,V_j)$

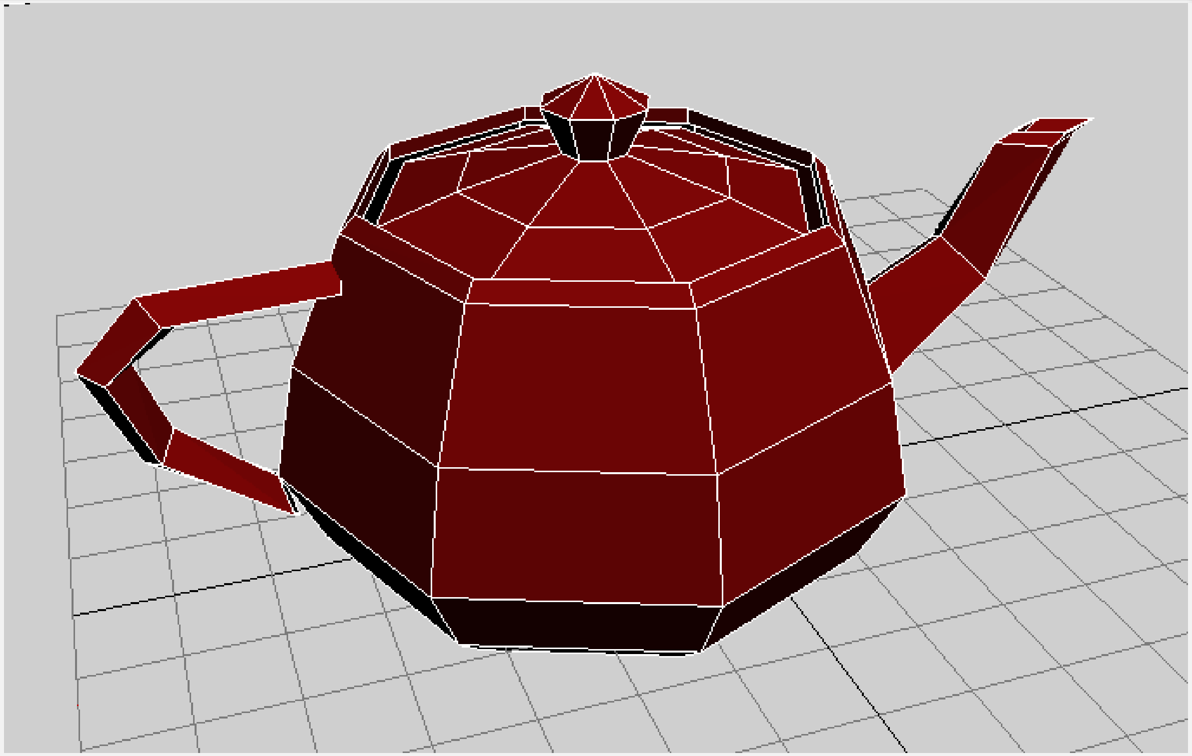


“Real wireframe”

- Benedict Radcliffe’s Toyota Corolla

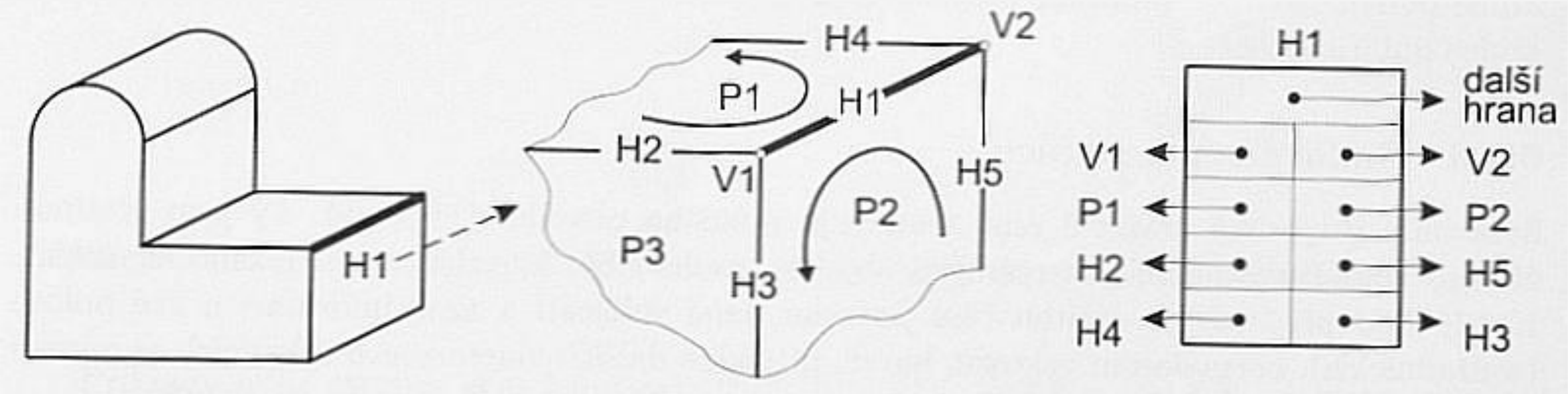
Polygonal representation

- Polygonal mesh
- Vertices
  - V(x,y,z)
- Faces
  - $F(V_1, V_2, \ldots V_n)$
- (Edges)
  - $E(V_i, V_j)$



Example – winged edge

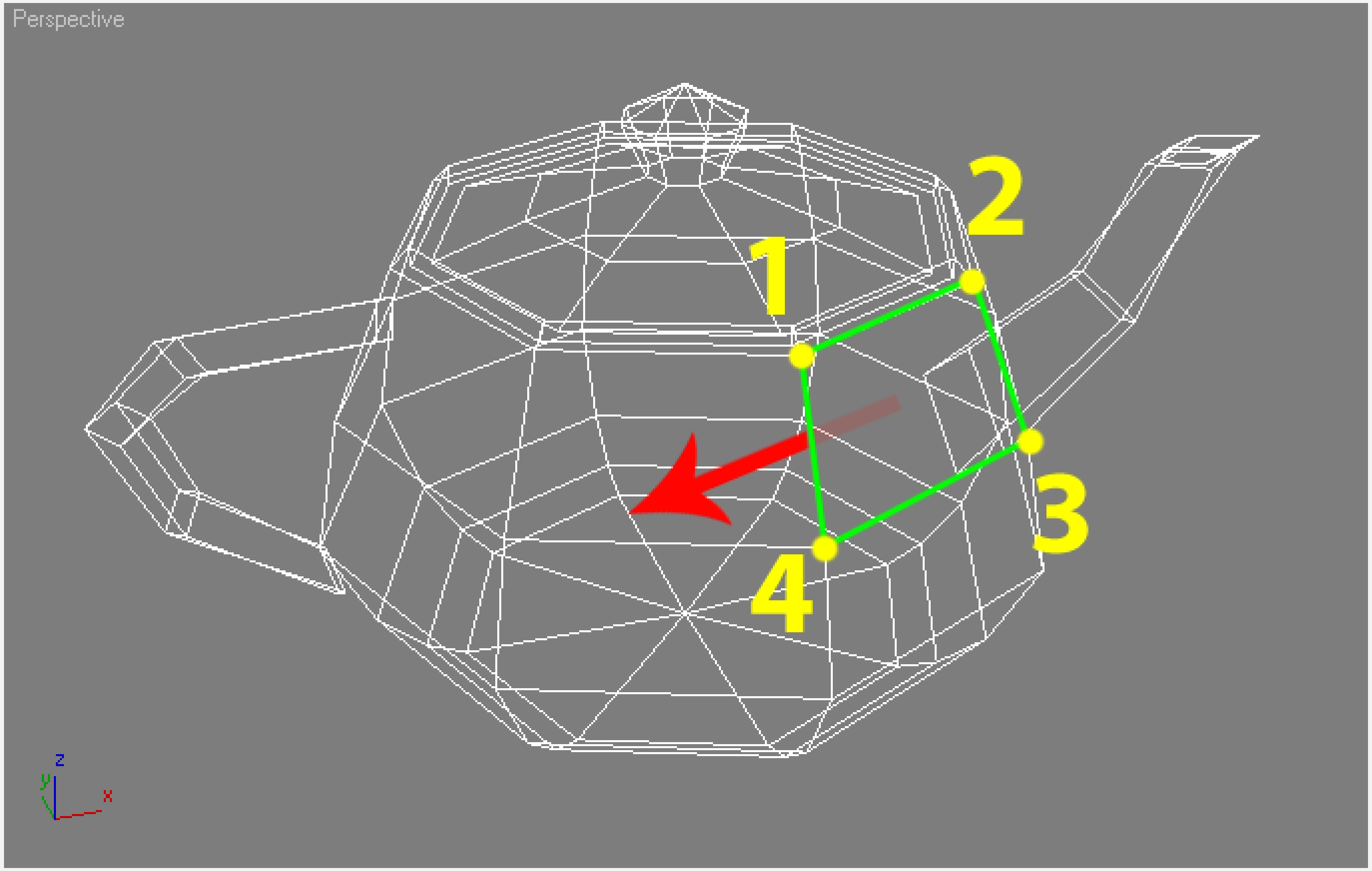
- Data structure for mesh representation



- Fast mesh traversal
- Splitting / merging operations

Polygon orientation → normal

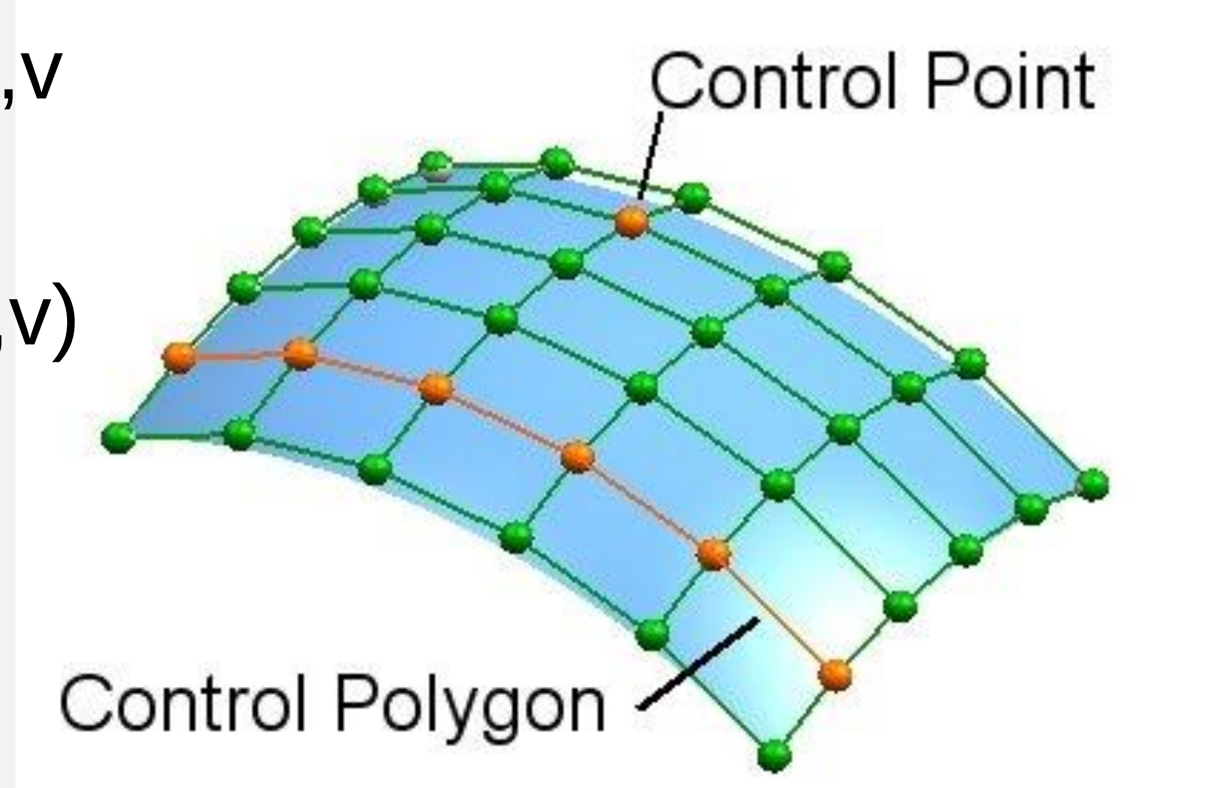
- Remember right-hand coordinate system
- Surface normal



Parametric surfaces

- 3D generalization of parametric curves
- m × n control points
- parameters u,v

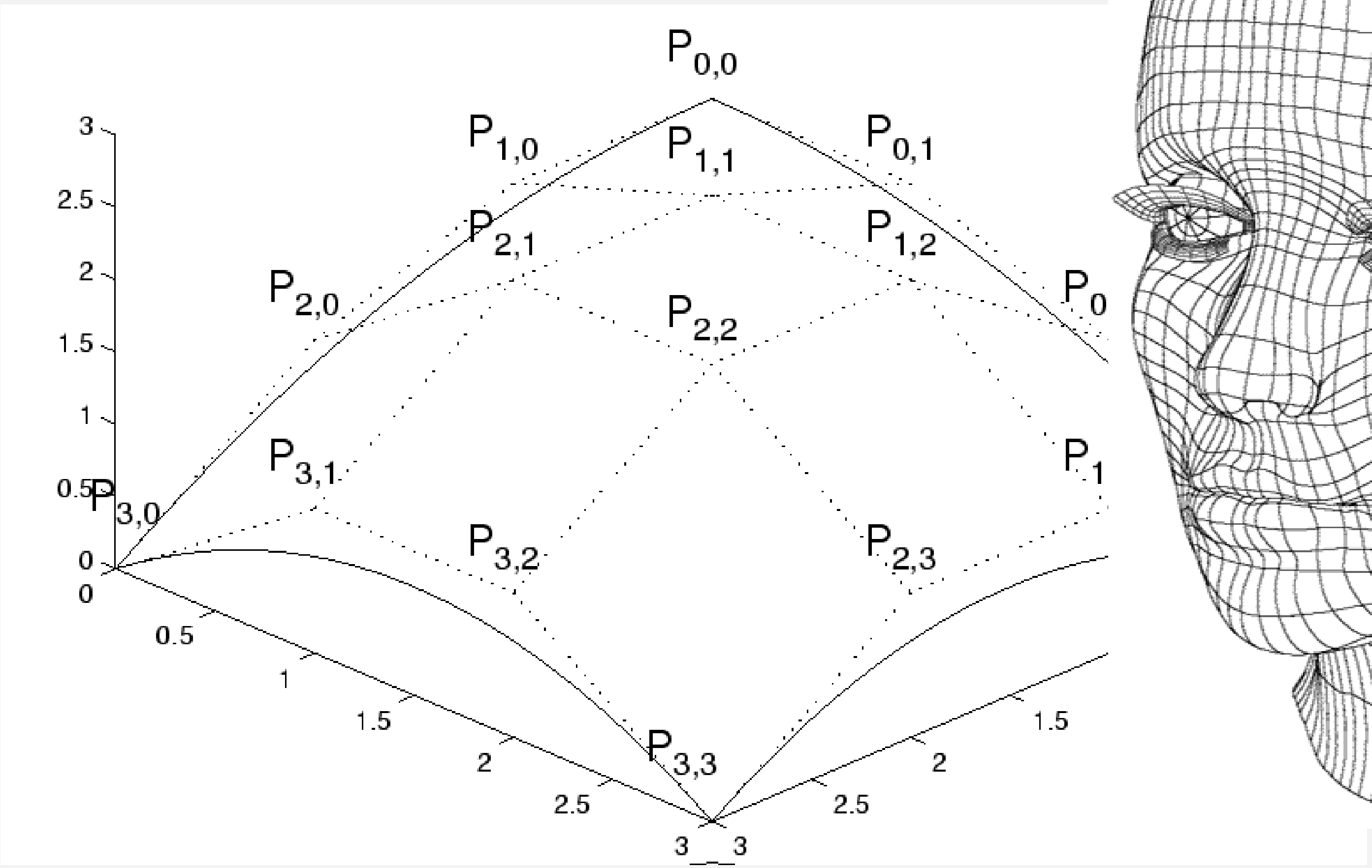
- C(x,y,z) = f(u,v)

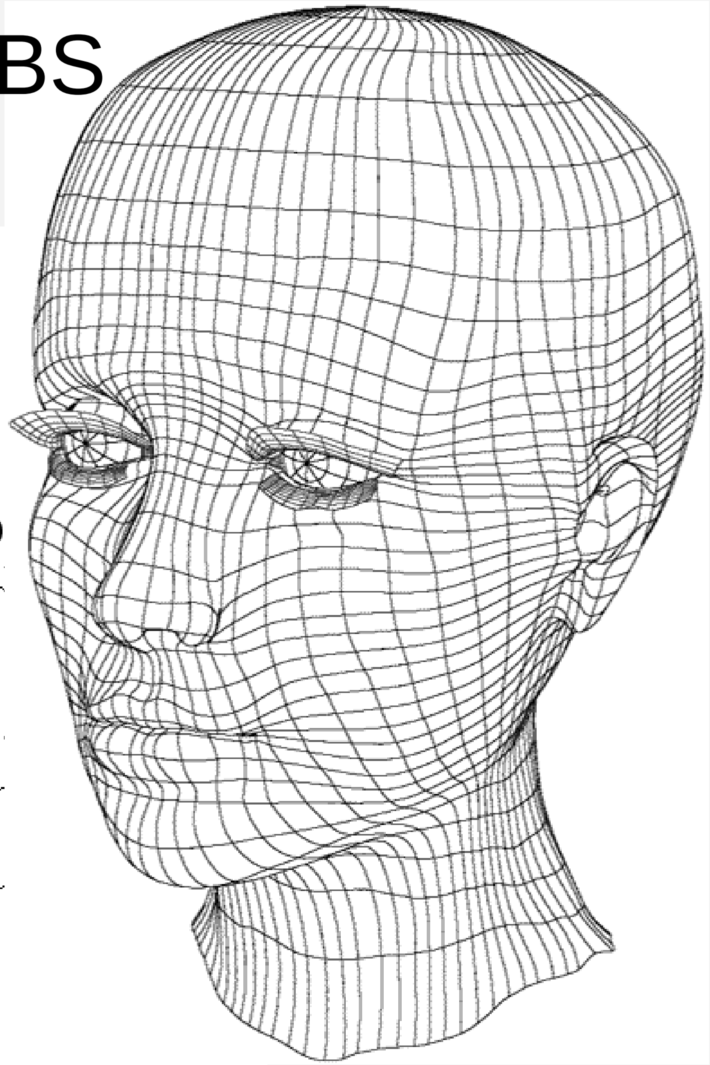


http://cadauno.sourceforge.net/

Parametric surfaces – e.g.

- Cubic Bezier surface, NURBS



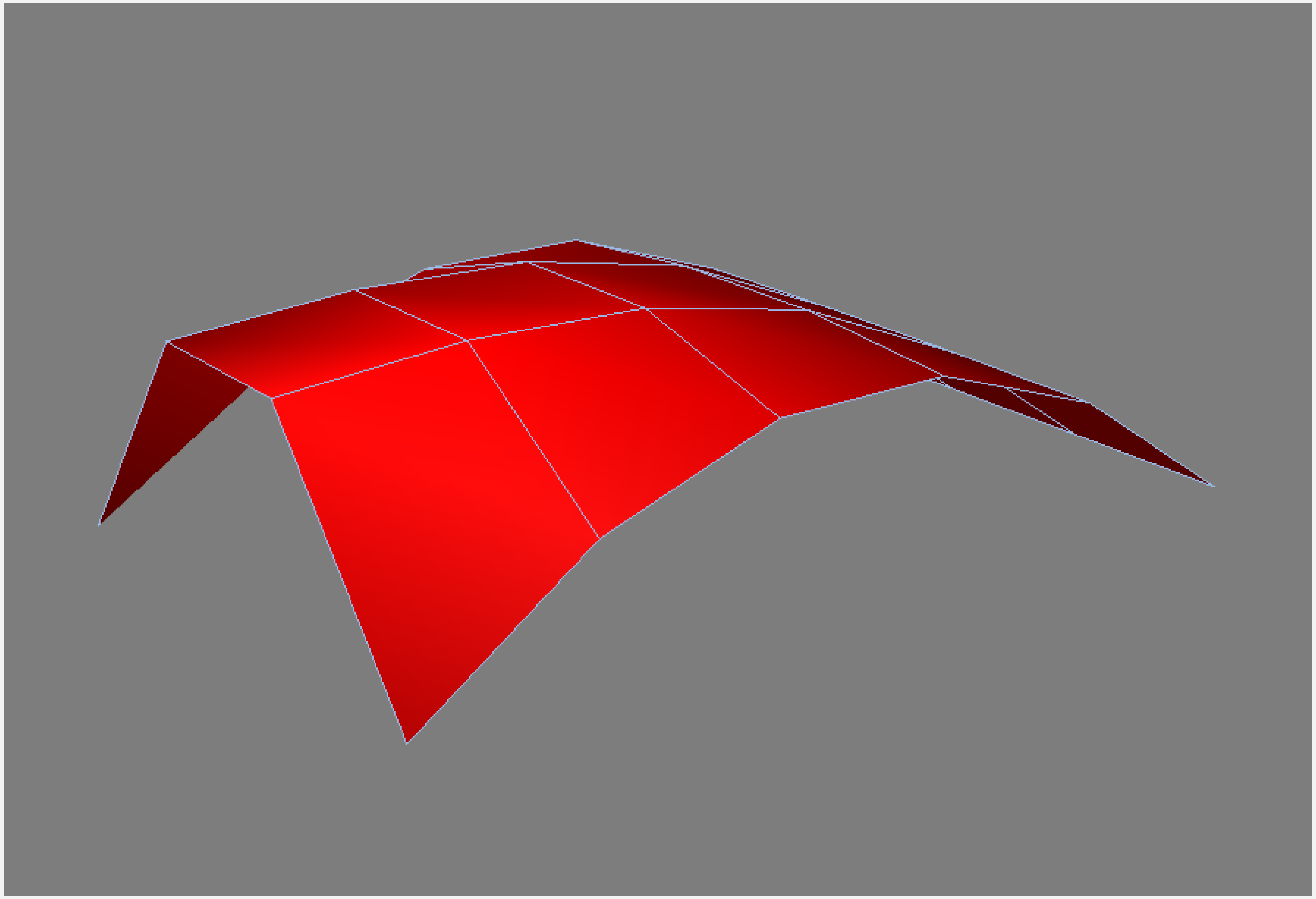


Parametric vs. polygonal

- Parametric
  - smooth, reparametrizable
  - harder rendering
  - precise rendering
- Polygonal
  - discrete, hard to reparametrize
  - faster rendering or rasterization
  - approximative rendering

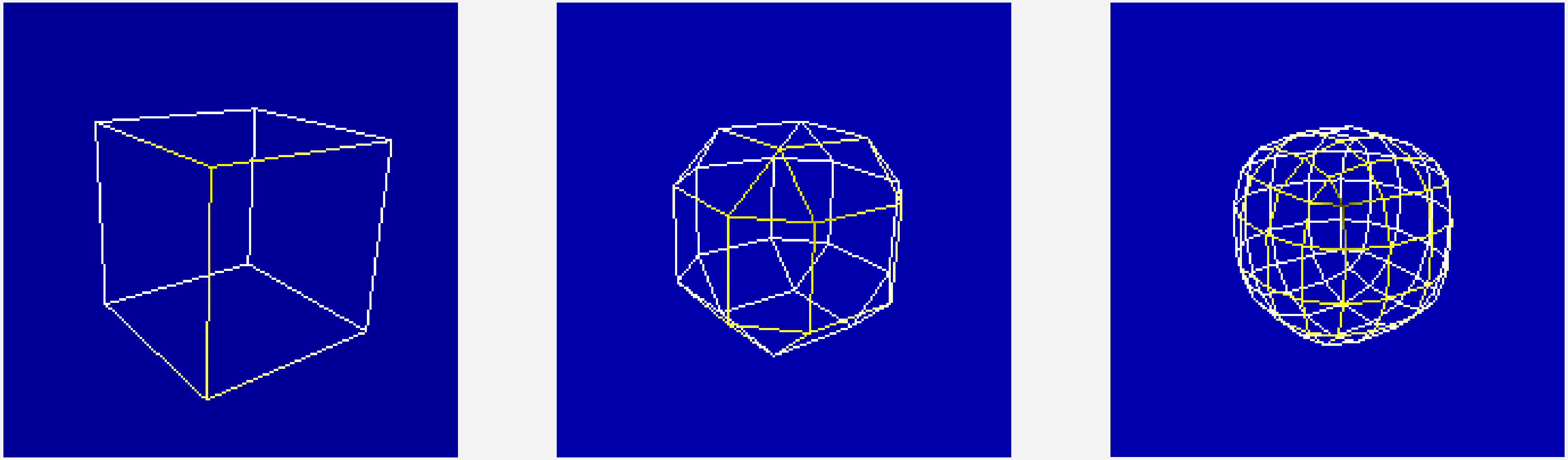
Parametric → polygonal

- $C(x,y,z) = f(u,v)$
- u = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, v = 0.0, 0.33, 0.66, 1.0



Subdivision surfaces

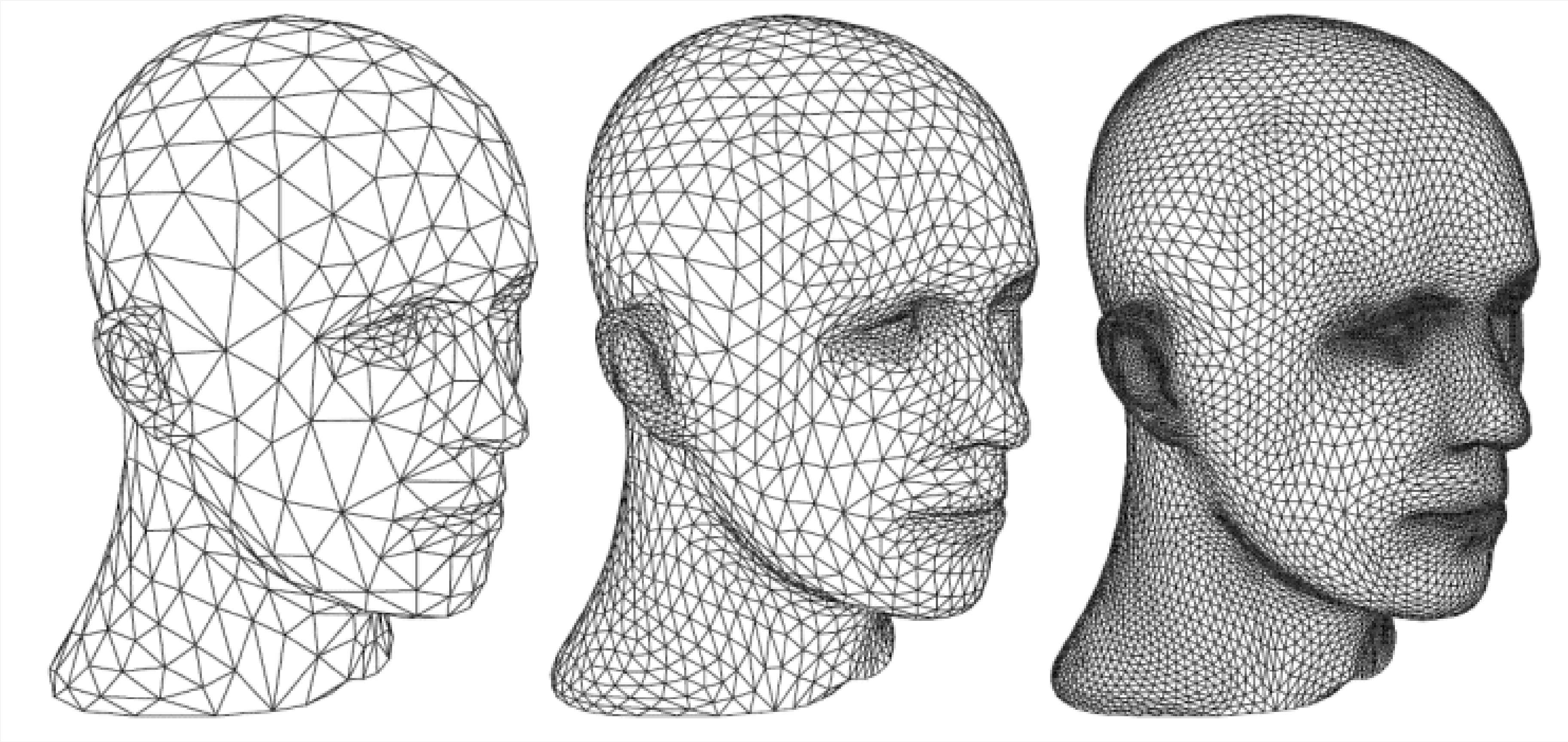
- Recursive subdivision of a polygonal model
- Limit surface - continuous



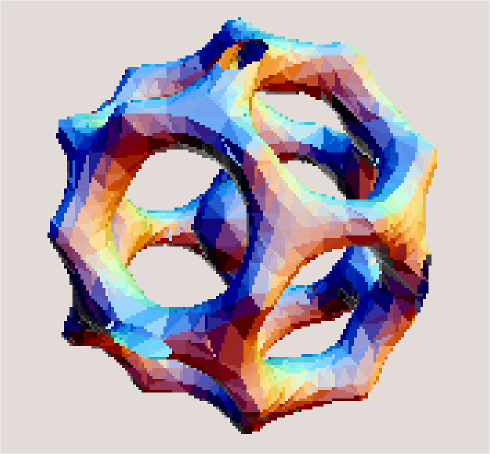
- Easy modeling, small data size

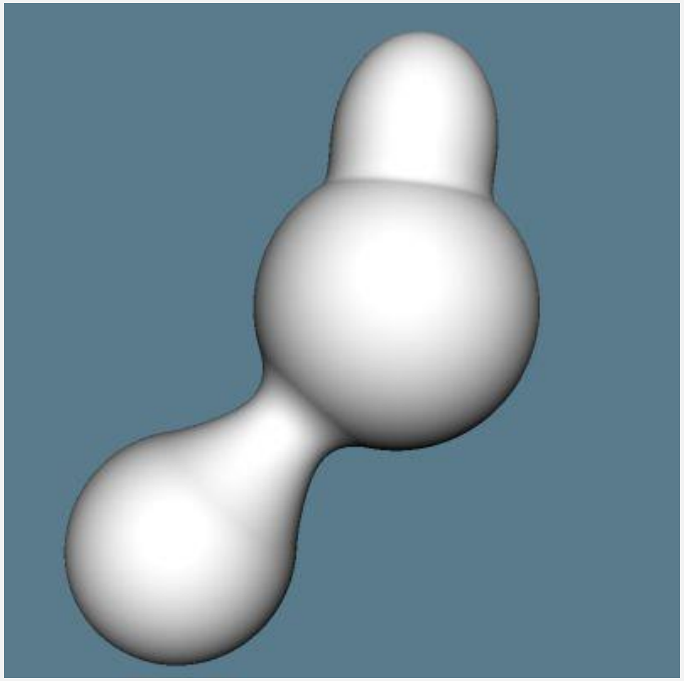
http://www.holmes3d.net/graphics/subdivision/

Subdivision example



Implicit surfaces





- F(x,y,z) = 0
- sphere:

$$x^2 + y^2 + z^2 - r^2 = 0$$

- metaballs:

$$\sum_m R/((x - x_m)^2 + (y - y_m)^2 + (z - z_m)^2) - c = 0$$

- Examples: http://iat.ubalt.edu/summers/math/platsol.htm

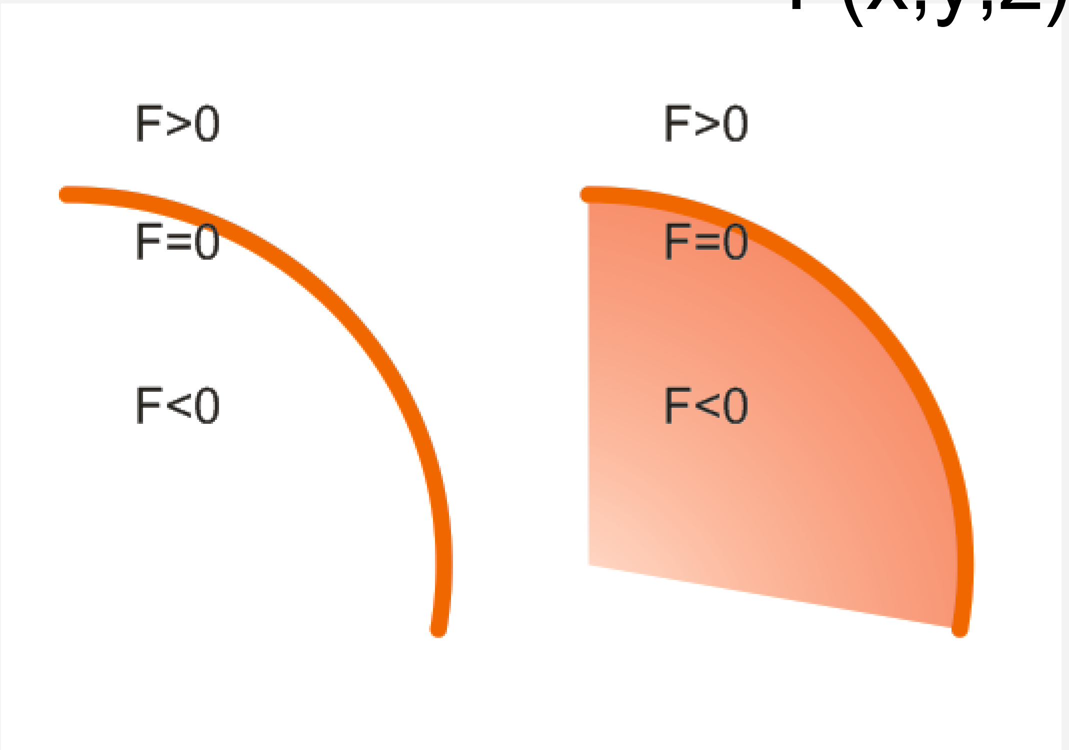
Video

- http://www.youtube.com/watch?v=k27ZVOp1PW4

Question

What happens if we turn $F(x,y,z) = 0$

into $F(x,y,z) \leq 0$?

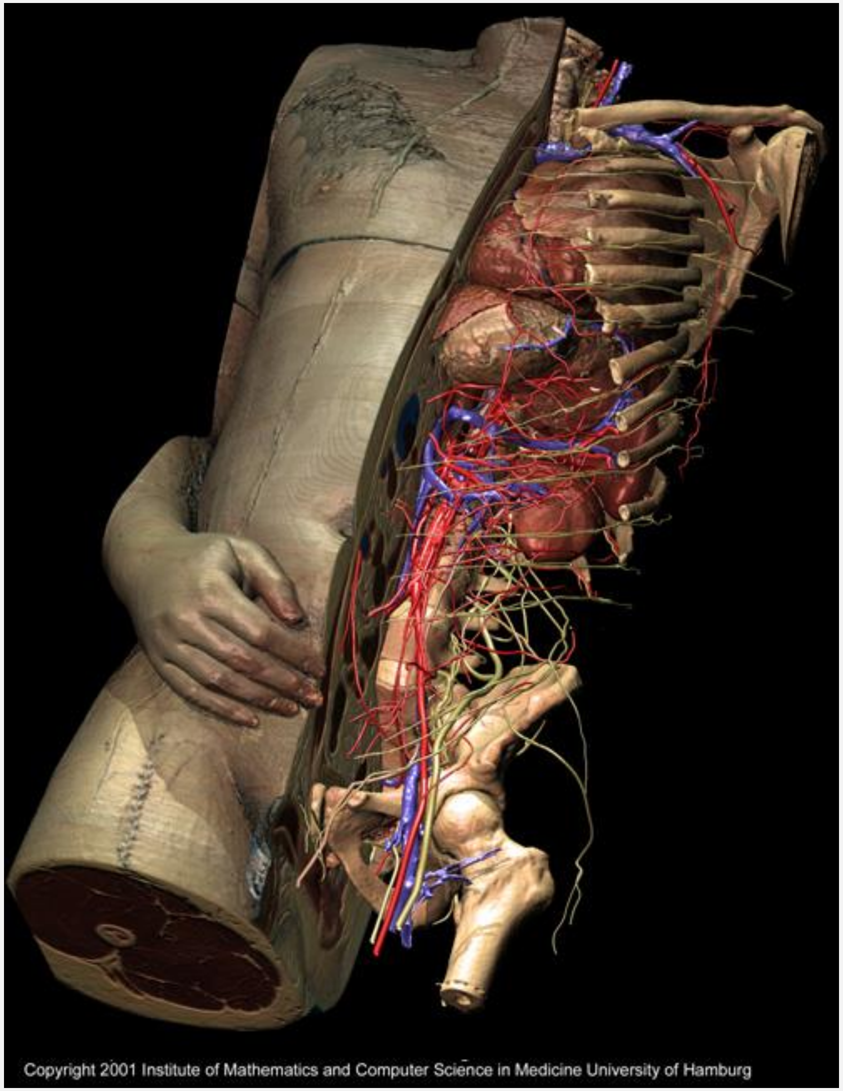


surface → volume

Volumetric representation

- not only boundary but also the insides of the object

- Medicine
- Physics
- Simulations
- Animation



Voxels

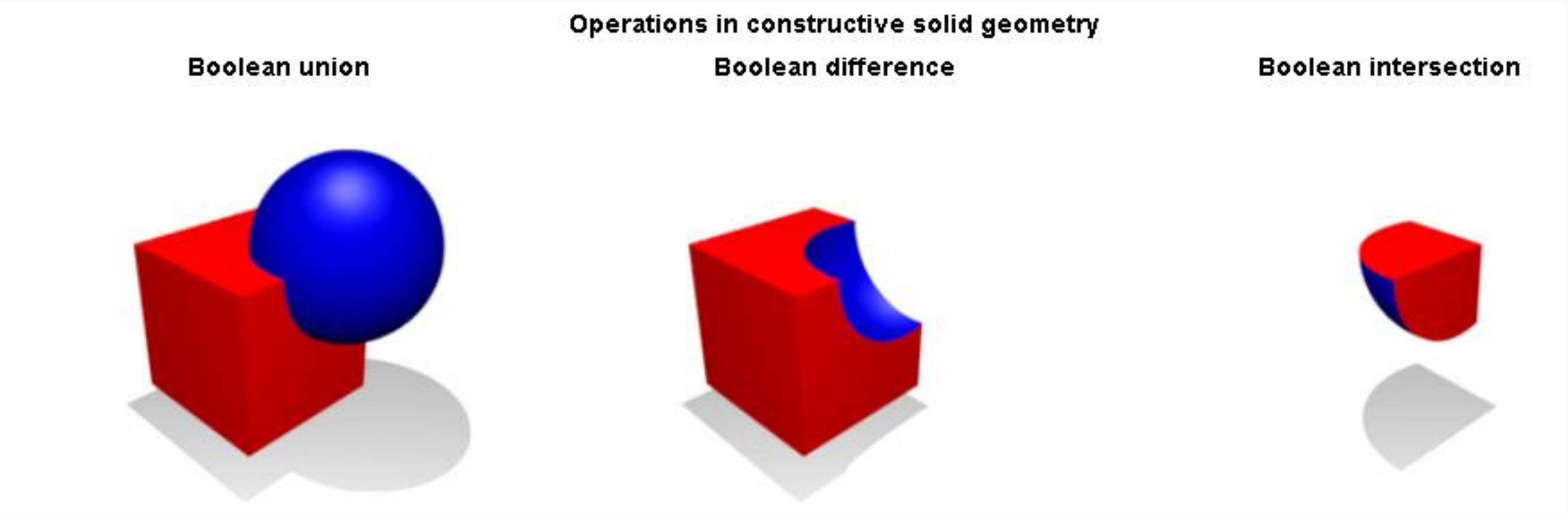
- Volume elements, “3D pixels”
- Discrete
- Binary values
- Float values

Video

- Prof. Miloš Šrámek
- http://www.viskom.oeaw.ac.at/~milos/
- Mpeg animation of 615 images, rendered from a segmented MRI data set.

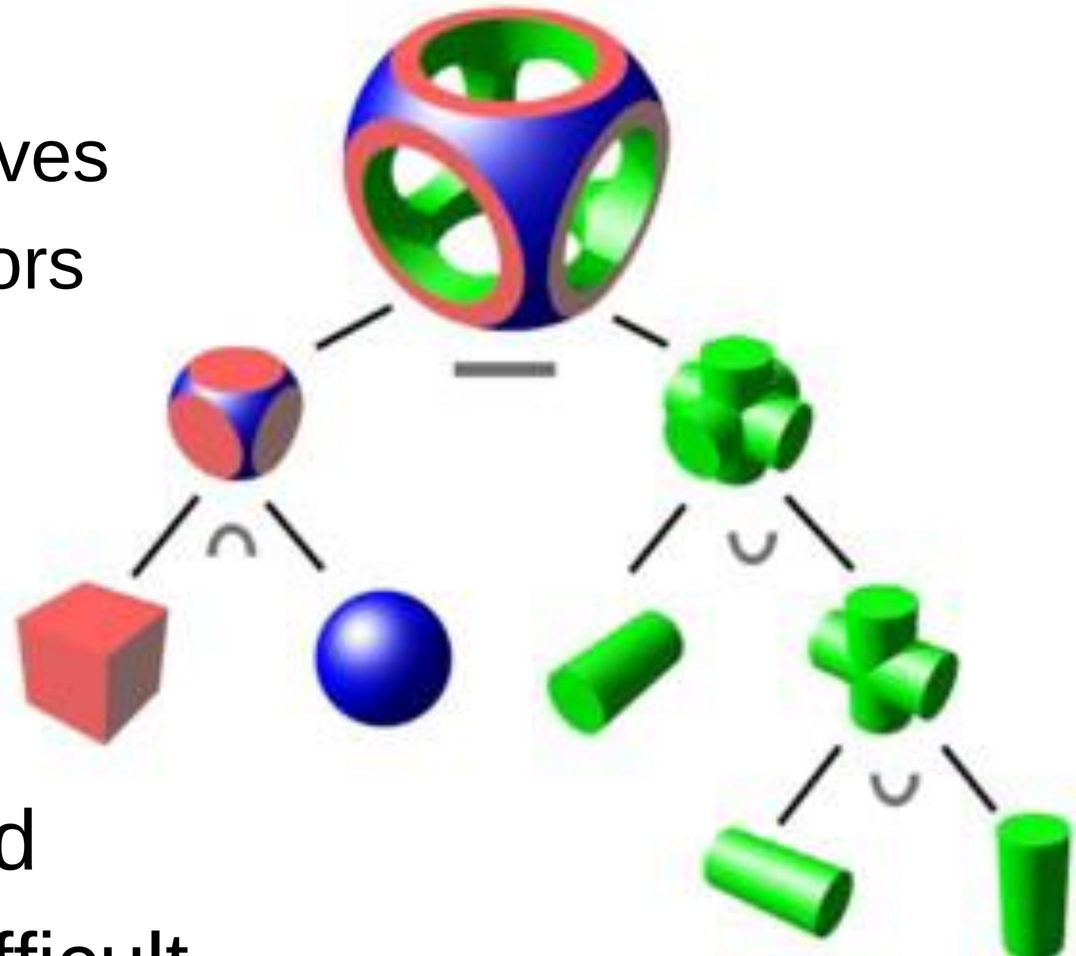
Constructive solid geometry

- Primitives + Boolean operators on sets
  - AND, OR, NOT
- singularities
- manifolds



CSG continued

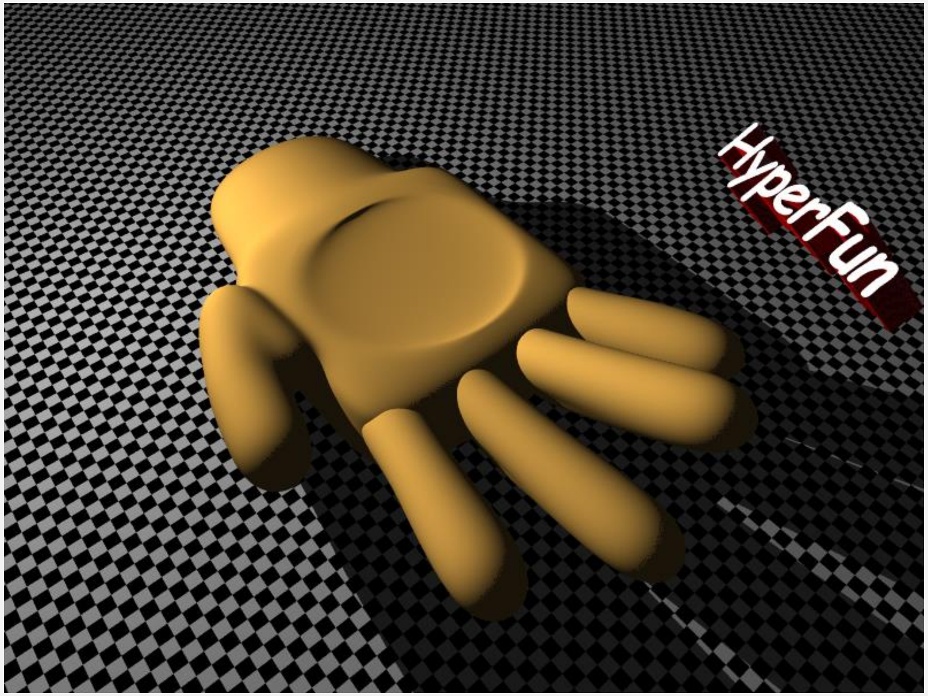
- Hierarchy
  - Leaves = primitives
  - Nodes = operators



- Volume-rep good
- Boundary-rep difficult

Functional representation

- F-rep ~ generalization of CSG
- More node functions – operators
  - e.g. object blending



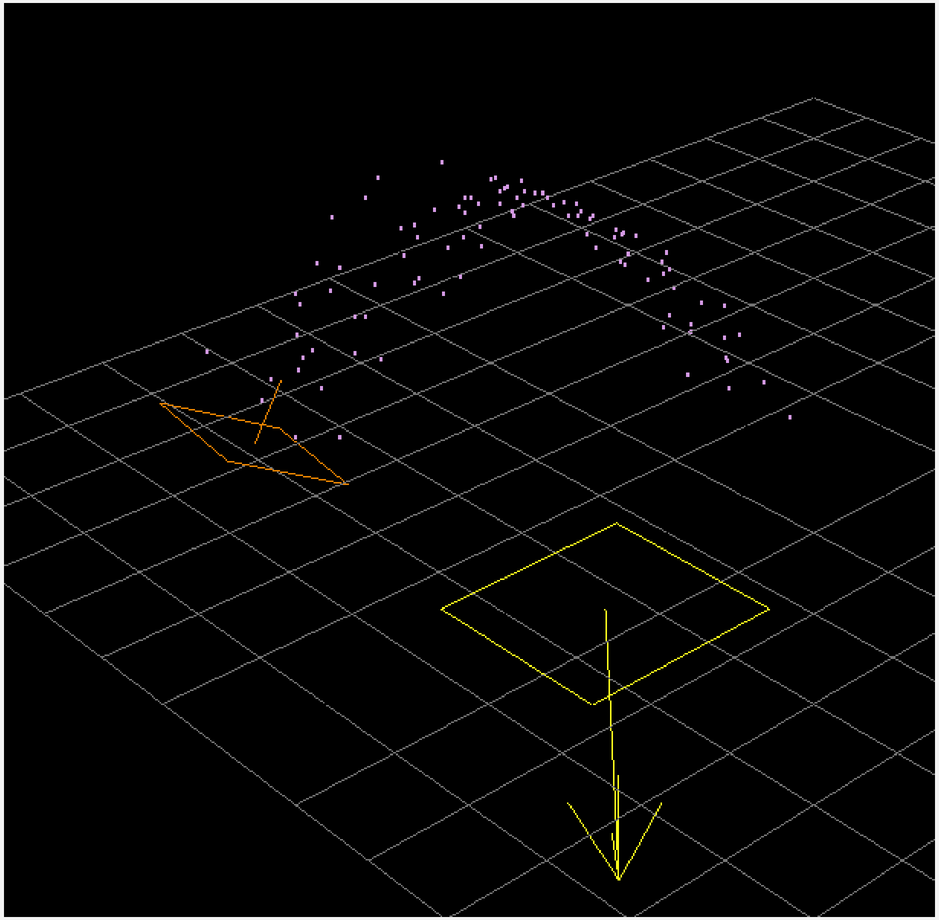

```
center = [0, 0.5, 0];
se = hfSuperell(x, center, 8, 2.5, 8, 0.3, 0.3);

center = [0, -0.5, 0];
el_cly = hfEllCylZ(x, center, 4, 2);

wrist = el_cly & (8-x[3]) & (x[3]+20);

center = [0, 3.5, 0];
el1 = hfEllipsoid(x, center, 8, 1, 8);

center = [-2, 3.5, 0];
el2 = hfEllipsoid(x, center, 8, 1, 8);

center = [2, 3.5, 0];
el3 = hfEllipsoid(x, center, 8, 1, 8);

center = [-0.5, 3.5, -2];
el4 = hfEllipsoid(x, center, 8, 1, 8);

el = el1 | el2 | el3 | el4;

palm = hfBlendUni(se, wrist, 5, 2, 2) \ el;
```

Special objects

Particle systems

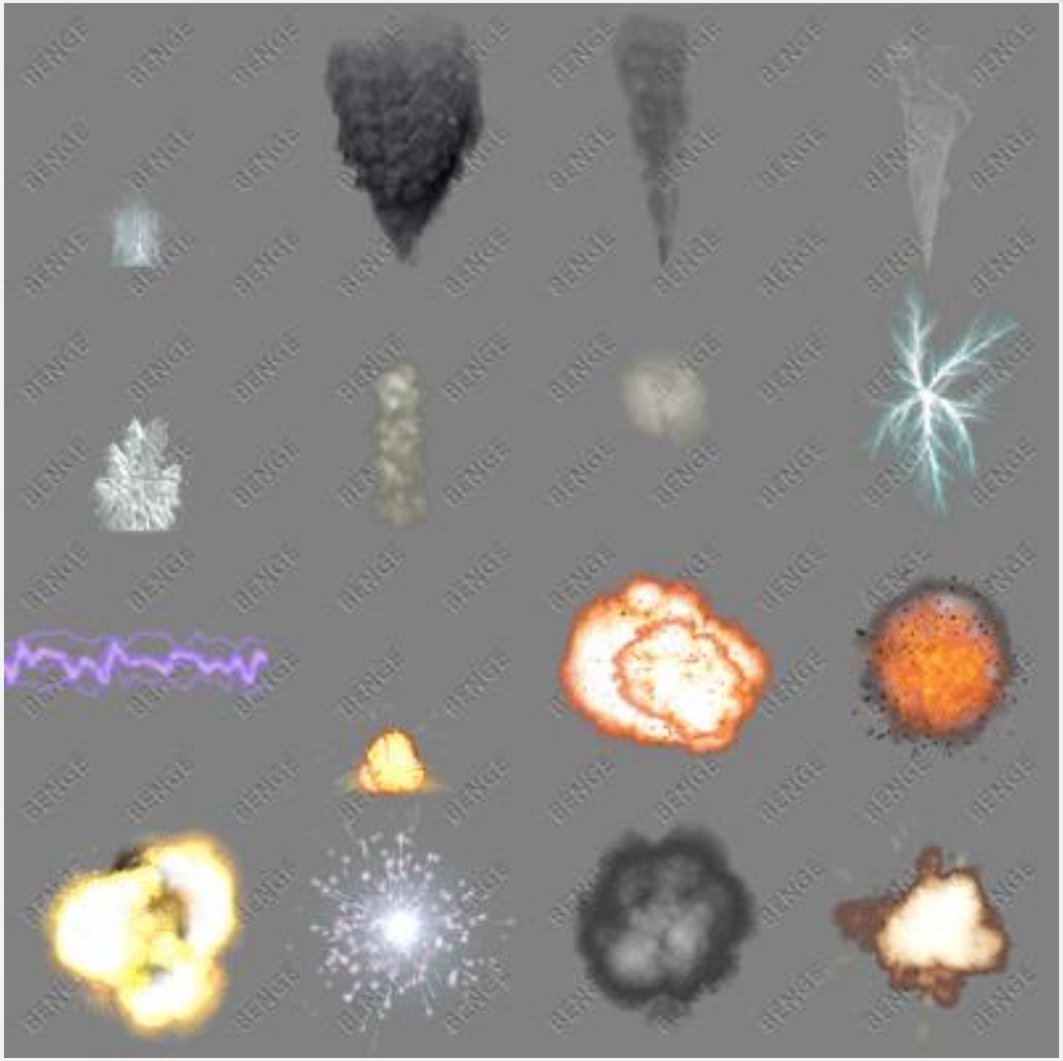
- Emitter + physics + elementary particle

- Water, snow, rain
- Smoke, fog, clouds
- Swarms, crowds
- Multi-agent systems



Billboards

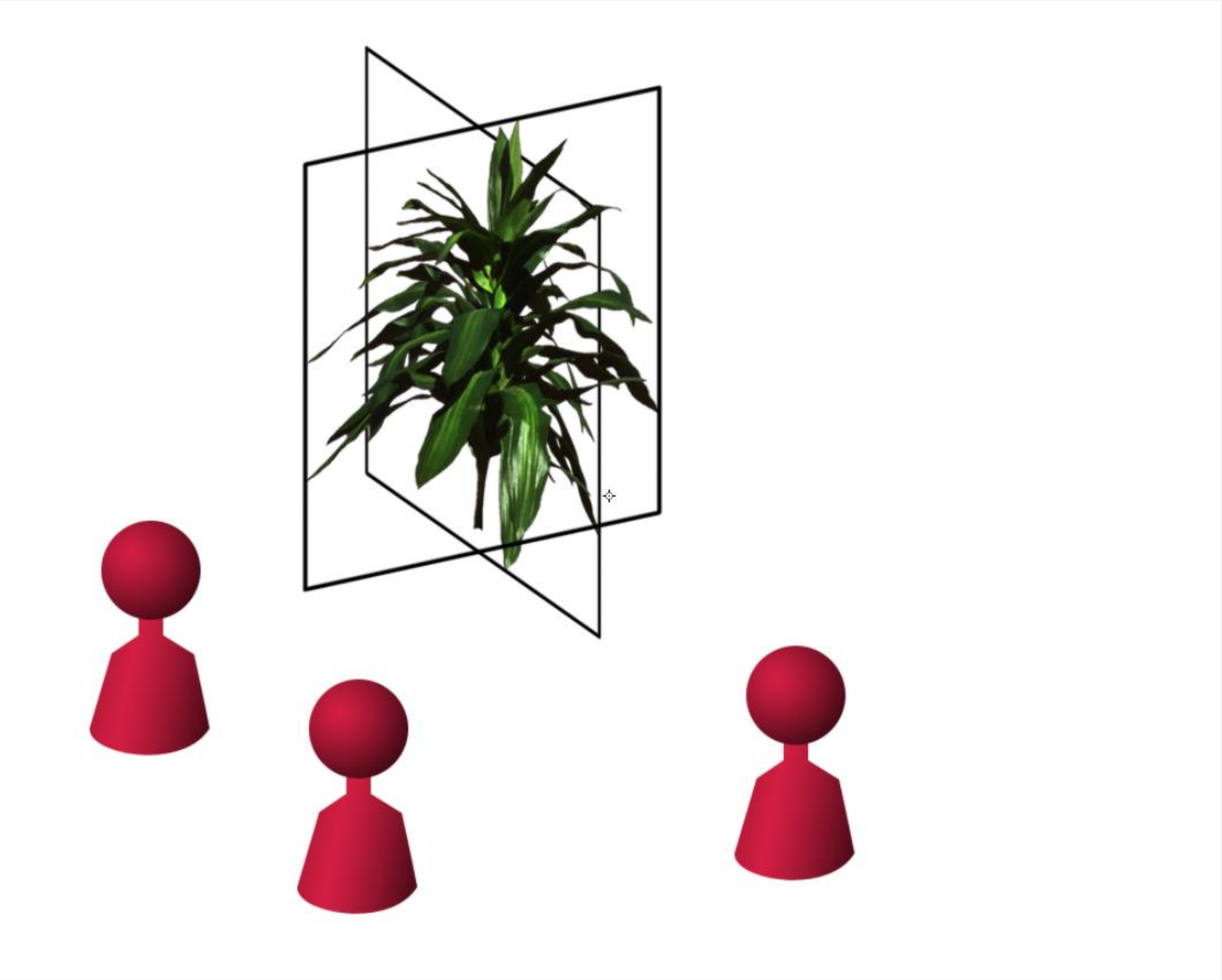
- Complex objects faked by a picture
- Trees, grass
- Fire, smoke
- Light effects
- Distant objects
- Can be animated



Animated Sprite Pack
http://www.thegamecreators.com/?m=view_product&id=2154

Billboard Tree

Multiple sprites



Summary

- Boundary vs. Volume representation
- Parametric vs. Polygonal representation
  – Parametric → Polygonal
- Sprites / Billboards