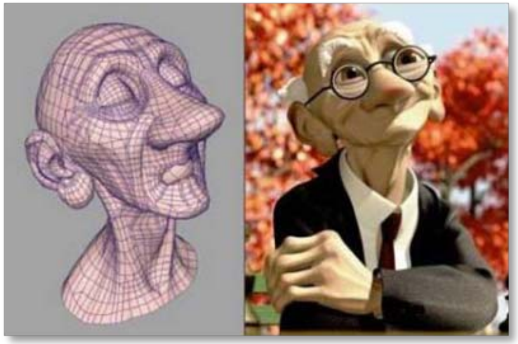
G22.3033-008, Spring 2010

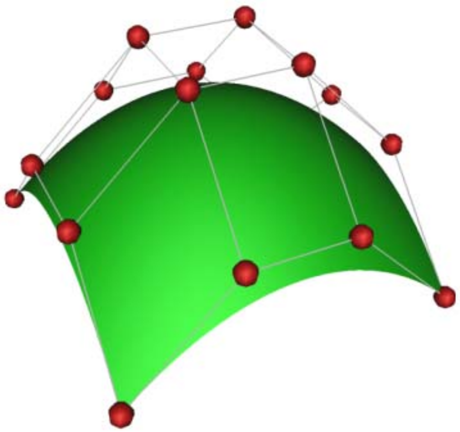
Geometric Modeling

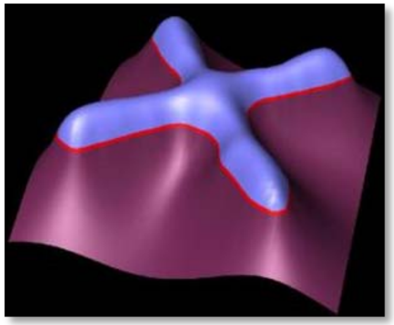
Shape Representations

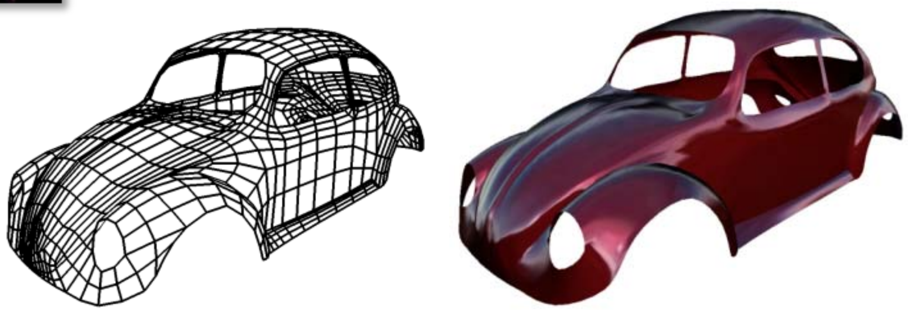
Course topics

- Shape representation
  - Points
  - Parametric surfaces
  - Implicits



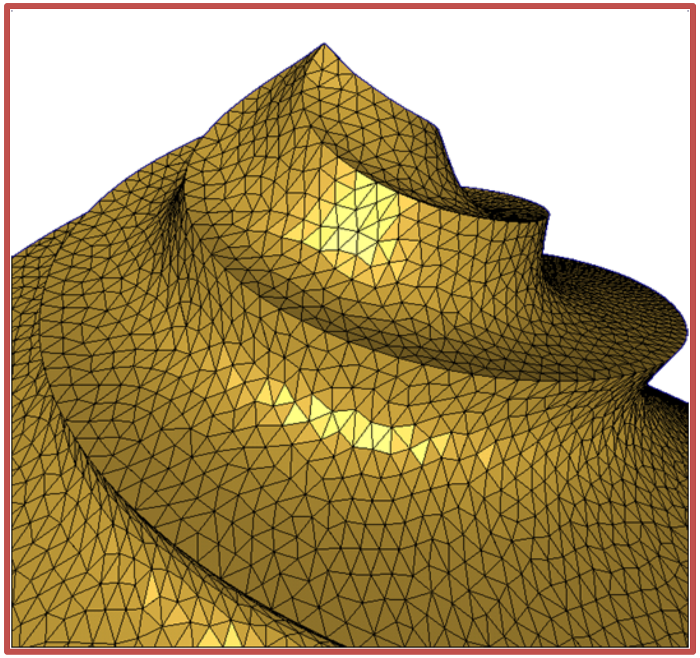


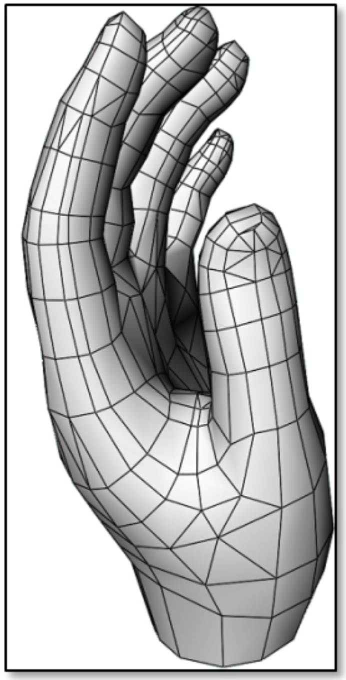


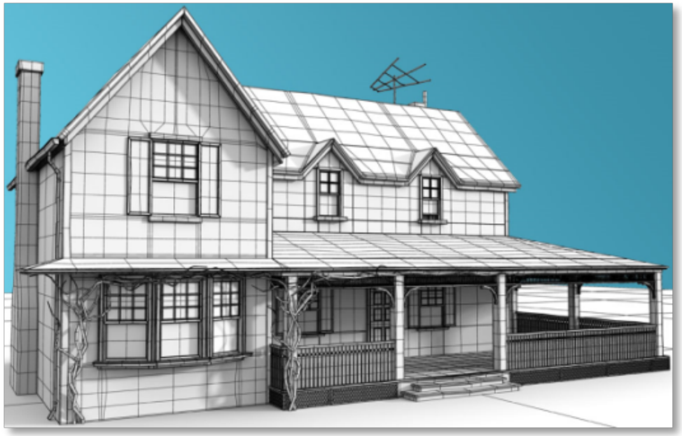


Course topics

- Shape representation
  - Polygonal meshes
  - Subdivision surfaces

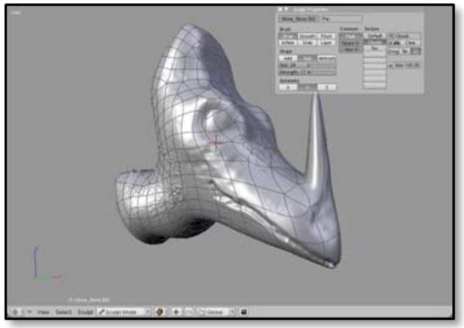


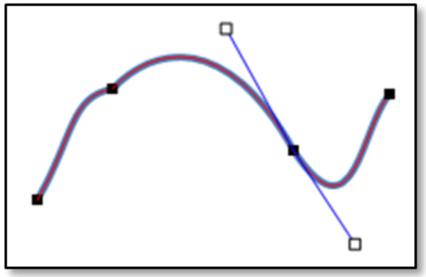


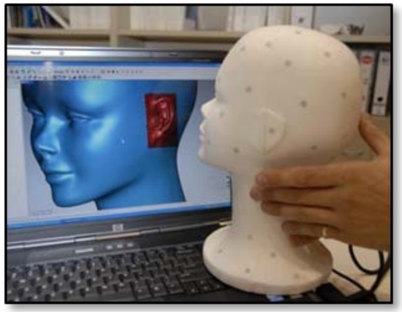


Shape representation

- Where does the shape come from?
- Modeling "by hand":
  - Higher-level representations, amendable to modification, control
  - Parametric surfaces, subdivision surfaces, implicits
- Acquired real-world objects:
  - Discrete sampling
  - Points, meshes







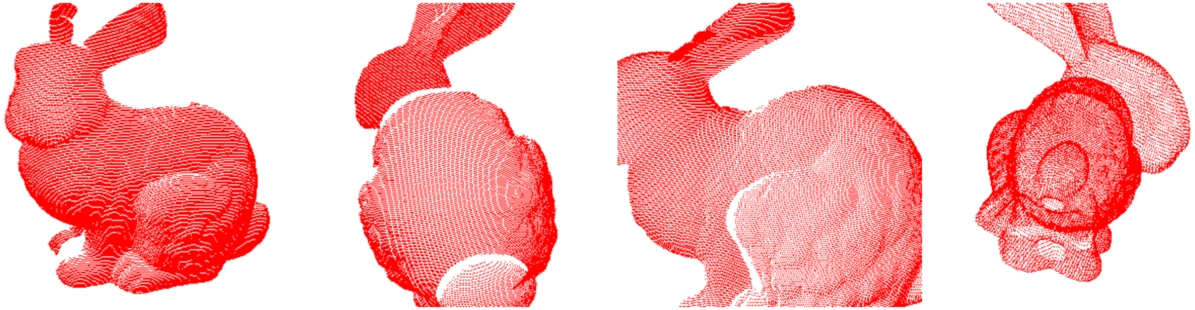
Points

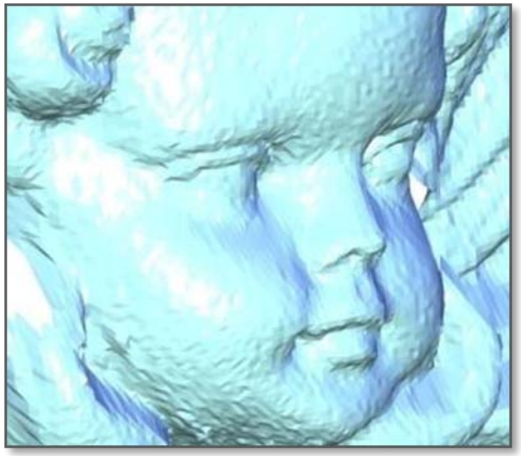
Shape acquisition

Sampling of real world objects

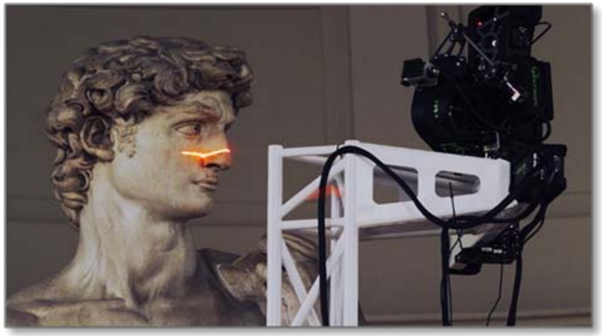
Points

- Standard 3D data from a variety of sources
  - Often results from scanners
  - Potentially noisy

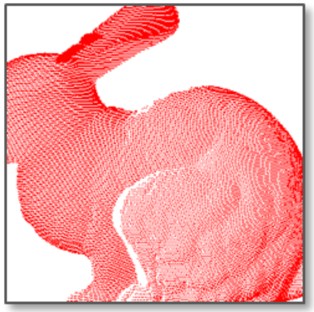




  - Depth imaging (e.g. by triangulation
  - Registration of multiple images



Points

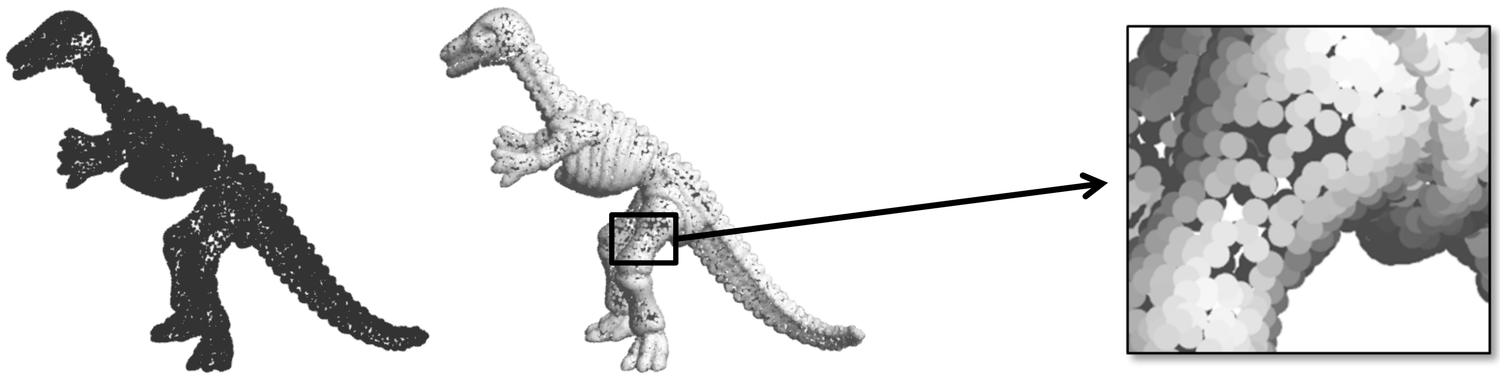


- points = unordered set of 3-tuples
- Often converted to other reps
  - Meshes, implicits, parametric surfaces
  - Easier to process, edit and/or render
- Efficient point processing and modeling requires a spatial partitioning data structure
  - To figure out neighborhoods

Points

Neighborhood information

- Why do we need neighbors?



need normals (for shading)

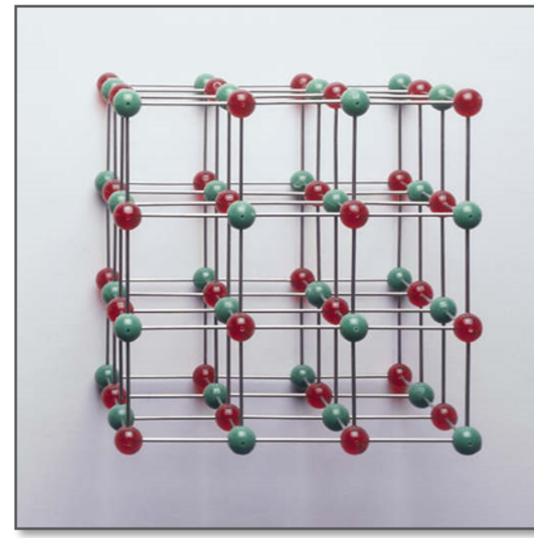
upsampling – need to count density

- Need sub-linear implementations of
  - k-nearest neighbors to point $\mathbf{x}$
  - In radius search $\|\mathbf{p}_i - \mathbf{x}\| < \varepsilon$

Spatial Data Structures

Commonly used for point processing

- Regular uniform 3D lattice
  - Simple point insertion by coordinate discretization
  - Simple proximity queries by searching neighboring cells

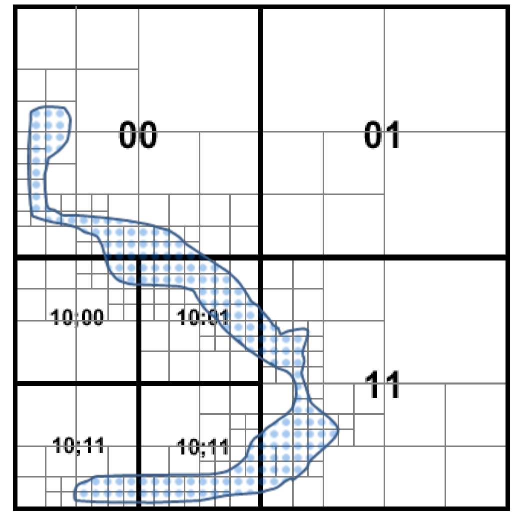


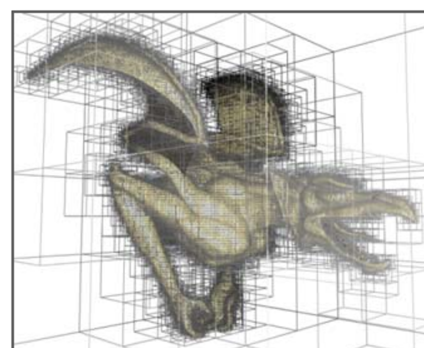
  - Determining lattice parameters (i.e. cell dimensions) is nontrivial
  - Generally unbalanced, i.e. many empty cells

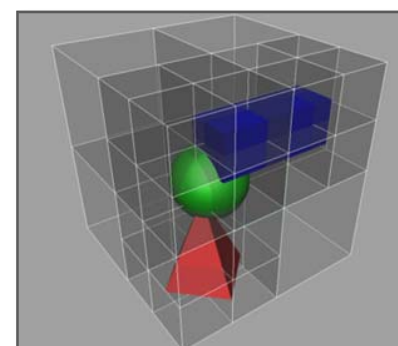
Spatial Data Structures

Commonly used for point processing

- Octree
  - Splits each cell into 8 equal cells
  - Adaptive, i.e. only splits when too many points in cell
  - Proximity search by (recursive) tree traversal and distance to neighboring cells
  - Tree might not be balanced



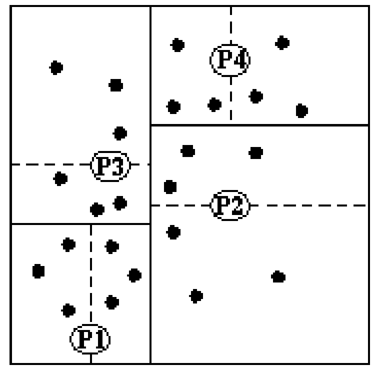


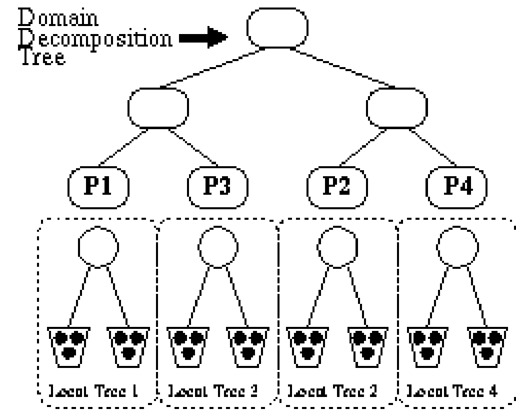


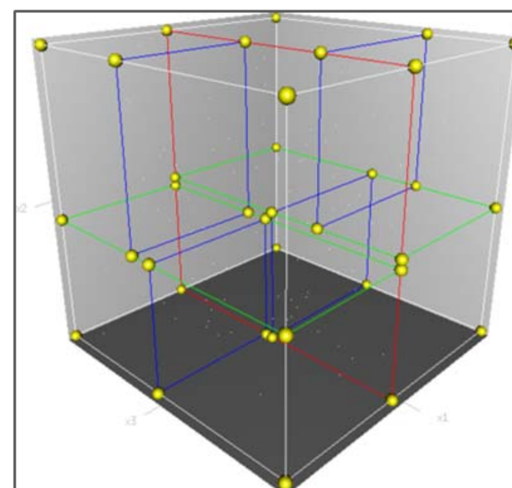
Spatial Data Structures

Commonly used for point processing

- Kd-Tree
  - Each cell is individually split along the median into two cells
  - Same amount of points in cells
  - Perfectly balanced tree
  - Proximity search similar to the recursive search in an Octree
  - More data storage required for inhomogeneous cell dimensions

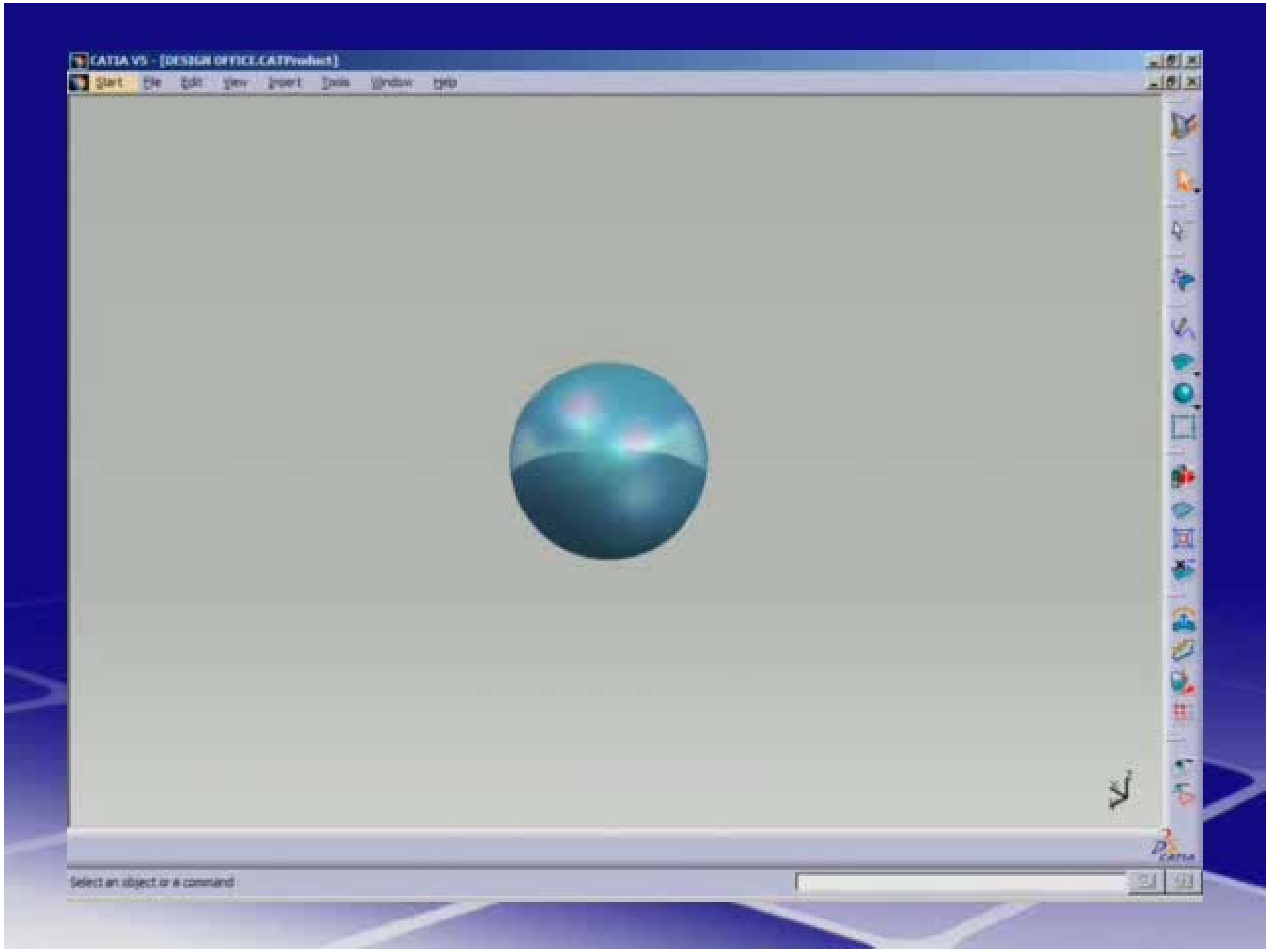






Parametric Curves and Surfaces

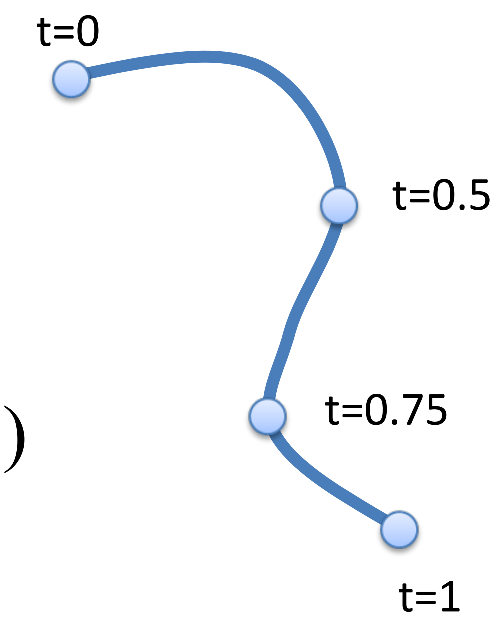
Smooth Shape Representation



Parametric Curves and Surfaces

- Curves are 1-dimensional objects

$$S(t) = \mathbf{x}_t$$

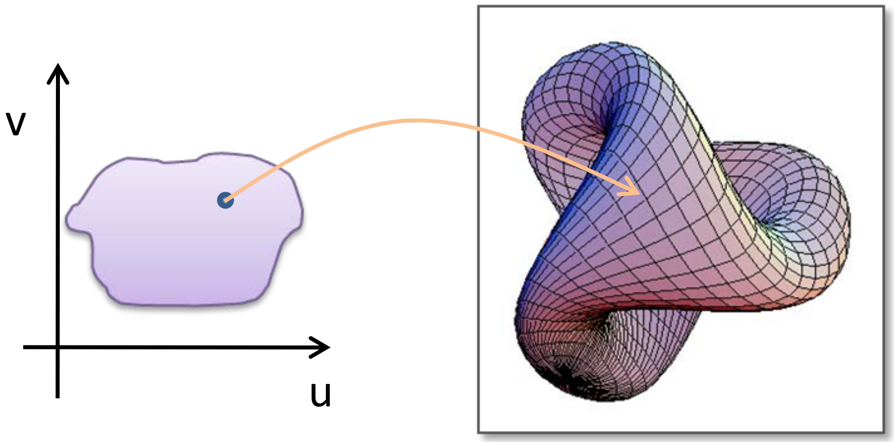


  - Planar curve: $S(t) = (x(t), y(t))$
  - Space curve: $S(t) = (x(t), y(t), z(t))$

Parametric Curves and Surfaces

- Surfaces are 2-dimensional parameterizations

$$S(u,v) = \mathbf{x}_t$$



$$S(u,v) = (x(u,v),\ y(u,v),\ z(u,v))$$

Parametric Curves

Examples

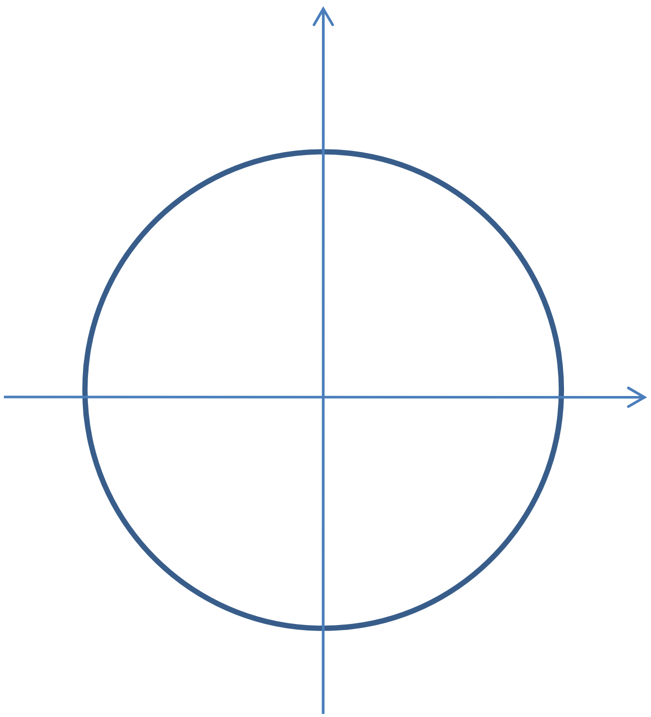
- Explicit curve/circle in 2D

$$\mathbf{p} : \mathrm{R} \rightarrow \mathrm{R}^2$$

$$t \mapsto \mathbf{p}(\mathrm{t}) = \left(x(t), \mathrm{y}(t)\right)$$

$$\mathbf{p}(t) = \mathrm{r} \cdot \left(\cos(t), \sin(t)\right)$$

$$t \in [0, 2\pi)$$



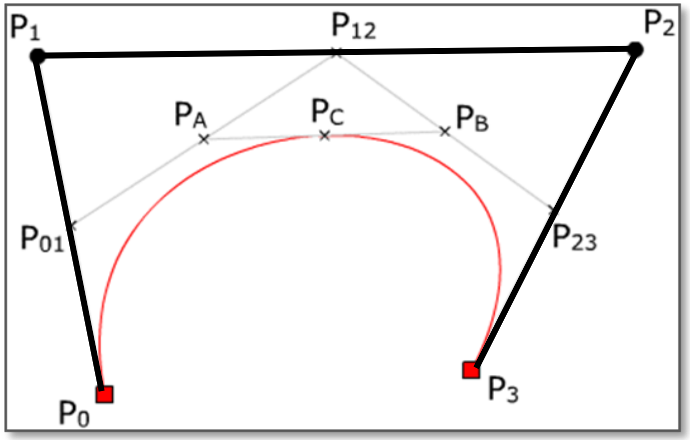
Parametric Curves and Surfaces

Examples

- Bezier curves

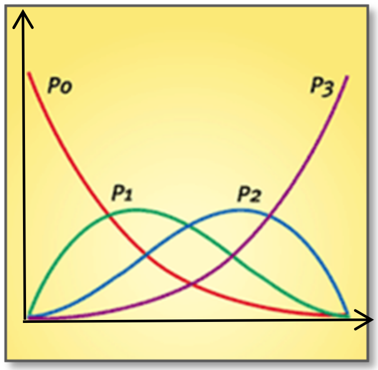
$$S(t) = \sum_{i=0}^{n} \mathbf{p}_i \, B_i^n(t)$$

Curve and control polygon



$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

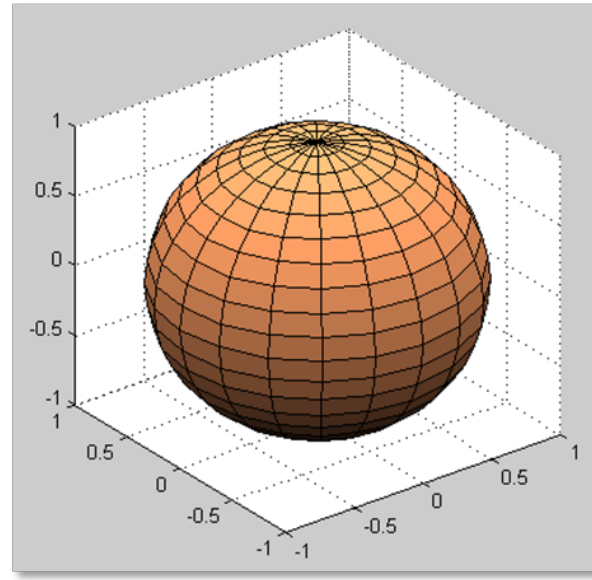
Basis functions



Parametric Surfaces

Examples

- Sphere in 3D



$$S : R^2 \rightarrow R^d, d = 1,2,3,\ldots$$

$$S(u,v) = r \cdot \big(\cos(u)\cos(v),\ \sin(u)\cos(v),\ \sin(v)\big)$$

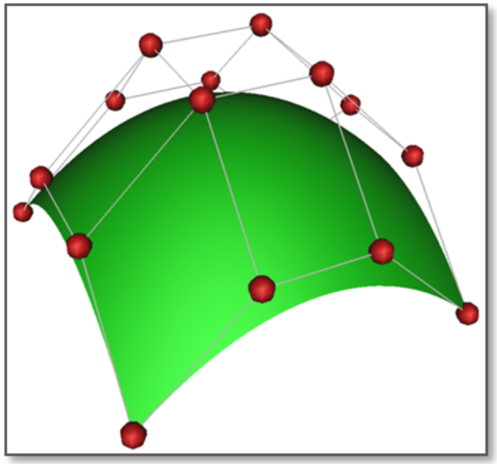
$$(u,v) \in [0, 2\pi) \times [-\pi/2, \pi/2]$$

Parametric Surfaces

Tensor product surfaces

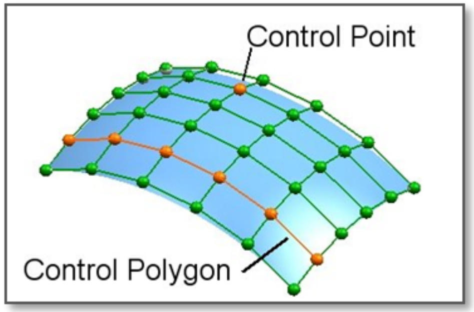
- Curve swept by another curve

$$S(u,v) = \sum_{i,j} \mathbf{p}_{ij} B_i(u) B_j(v)$$



- Bezier surface:

$$S(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{ij}\, B_i^m(u) B_j^n(v)$$



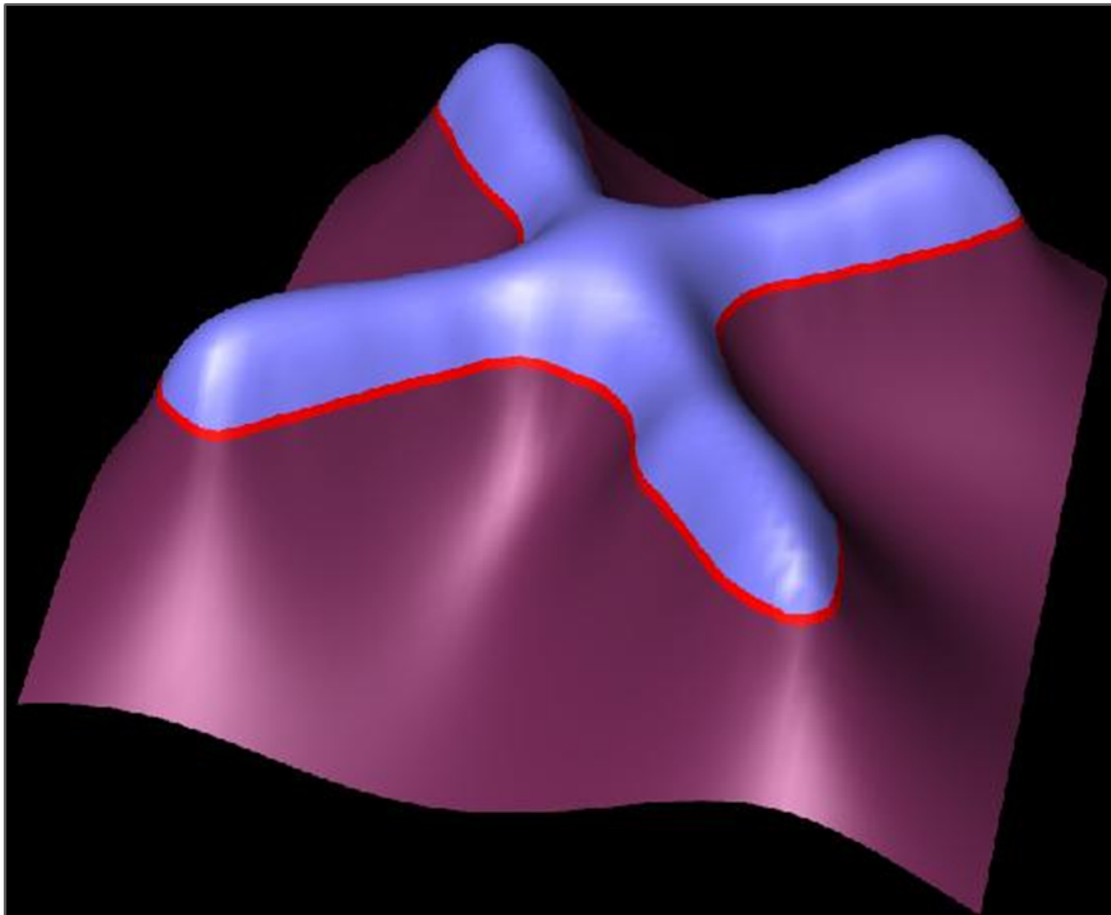
Parametric Curves and Surfaces

- Advantages
  - Easy to generate points on the curve/surface
  - Separates x/y/z components
- Disadvantages
  - Hard to determine inside/outside
  - Hard to determine if a point is on the curve/surface

Implicit Curves and Surfaces

Implicit Curves and Surfaces

Illustration



Implicit Curves and Surfaces

Definition

- Definition $g : \mathbb{R}^3 \rightarrow \mathbb{R}$

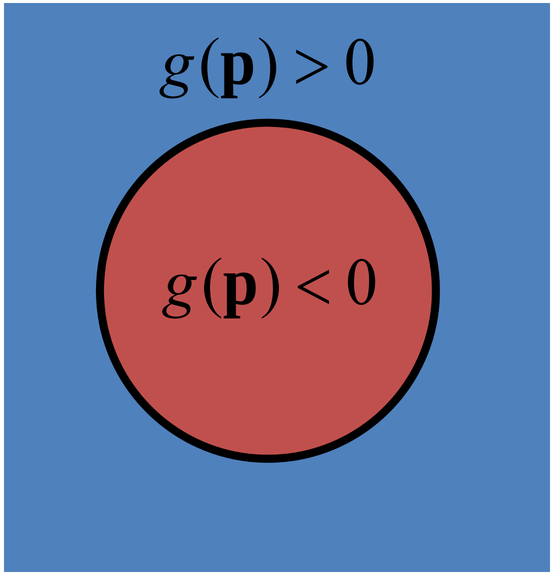
$$K = g^{-1}(0) = \{\mathbf{p} \in \mathbb{R}^3 : g(\mathbf{p}) = 0\}$$

- Space partitioning

$\{\mathbf{p} \in \mathbb{R}^3 : g(\mathbf{p}) < 0\}$ Inside

$\{\mathbf{p} \in \mathbb{R}^3 : g(\mathbf{p}) = 0\}$ Curve/Surface

$\{\mathbf{p} \in \mathbb{R}^3 : g(\mathbf{p}) > 0\}$ Outside



Implicit Curves and Surfaces

Examples

- Implicit circle and sphere

$$f : R^2 \to R$$

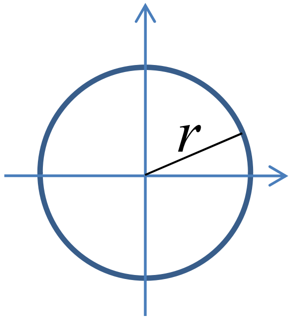
$$K = \{\mathbf{p} \in R^2 : f(\mathbf{p}) = 0\}$$

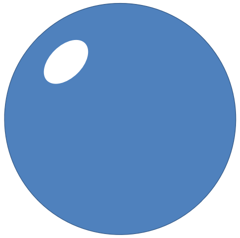
$$g : R^3 \to R$$

$$K = \{\mathbf{p} \in R^3 : g(\mathbf{p}) = 0\}$$

$$f(x, y) = x^2 + y^2 - r^2$$

$$g(x, y, z) = x^2 + y^2 + z^2 - r^2$$





Implicit Curves and Surfaces

Gradient

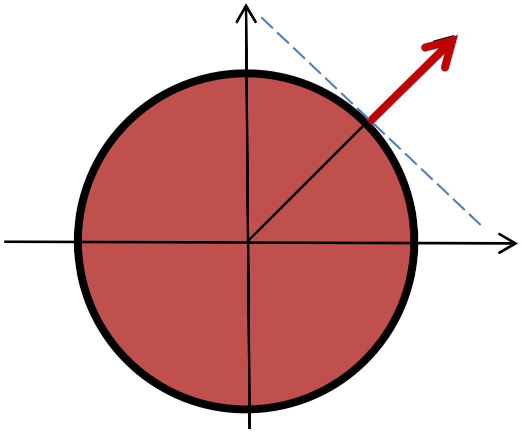
- The normal vector to the surface (curve) is given by the gradient of the (scalar) implicit function

$$\nabla g(x, y, z) = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}, \frac{\partial g}{\partial z} \right)^{\mathrm{T}}$$

- Example

$$g(x, y, z) = x^2 + y^2 + z^2 - r^2$$

$$\nabla g(x, y, z) = (2x, 2y, 2z)^{\mathrm{T}}$$

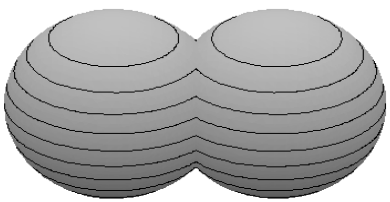


$$\nabla g(x, y, z) = (2, 2, 0)^{\mathrm{T}}$$

Implicit Curves and Surfaces

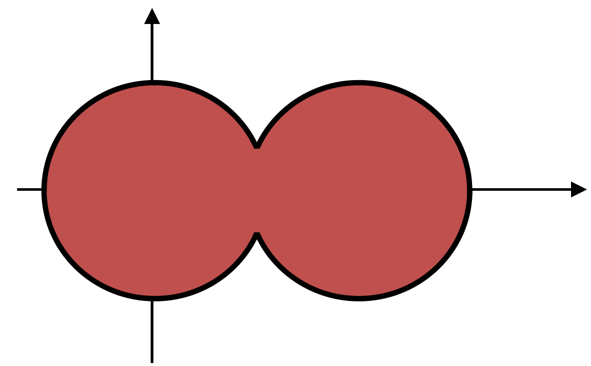
Smooth set operations

- Standard operations: union and intersection



$$\bigcup_i g_i(\mathbf{p}) = \min\, g_i(\mathbf{p})$$

$$\bigcap_i g_i(\mathbf{p}) = \max\, g_i(\mathbf{p})$$



- In many cases, smooth blending is desired
  - Pasko and Savchenko [1994]

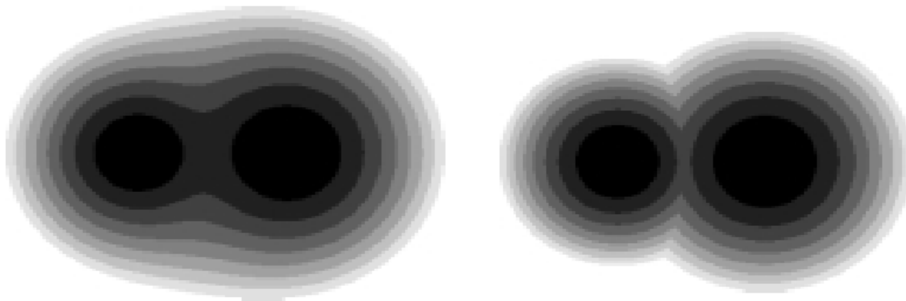
$$g \cup f = \tfrac{1}{1+\alpha}\left(g + f - \sqrt{g^2 + f^2 - 2\alpha g f}\right)$$

$$g \cap f = \tfrac{1}{1+\alpha}\left(g + f + \sqrt{g^2 + f^2 - 2\alpha g f}\right)$$

Implicit Curves and Surfaces

Smooth set operations

- Examples



$\alpha = 0$ $\alpha = 1$

- For $\alpha = 1$, this is equivalent to min and max

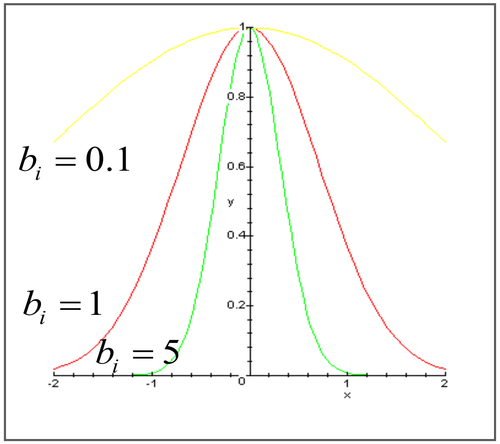
$$\lim_{\alpha \to 1} g \cup f = \tfrac{1}{2}\left(g + f - \sqrt{(g-f)^2}\right) = \frac{g+f}{2} - \frac{|g-f|}{2} = \min(g, f)$$

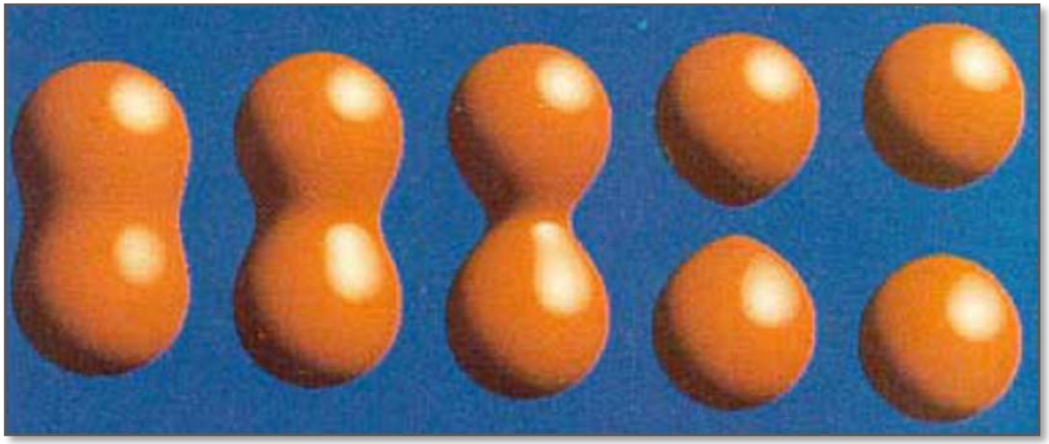
$$\lim_{\alpha \to 1} g \cap f = \tfrac{1}{2}\left(g + f + \sqrt{(g-f)^2}\right) = \frac{g+f}{2} + \frac{|g-f|}{2} = \max(g, f)$$

Implicit Curves and Surfaces

Blobs

- Suggested by Blinn [1982]
  - Defined implicitly by a potential function around a point $\mathbf{p}_i$: $g_i(\mathbf{p}) = a_i e^{-b_i \|\mathbf{p}-\mathbf{p}_i\|^2}$
  - Set operations by simple addition/subtraction





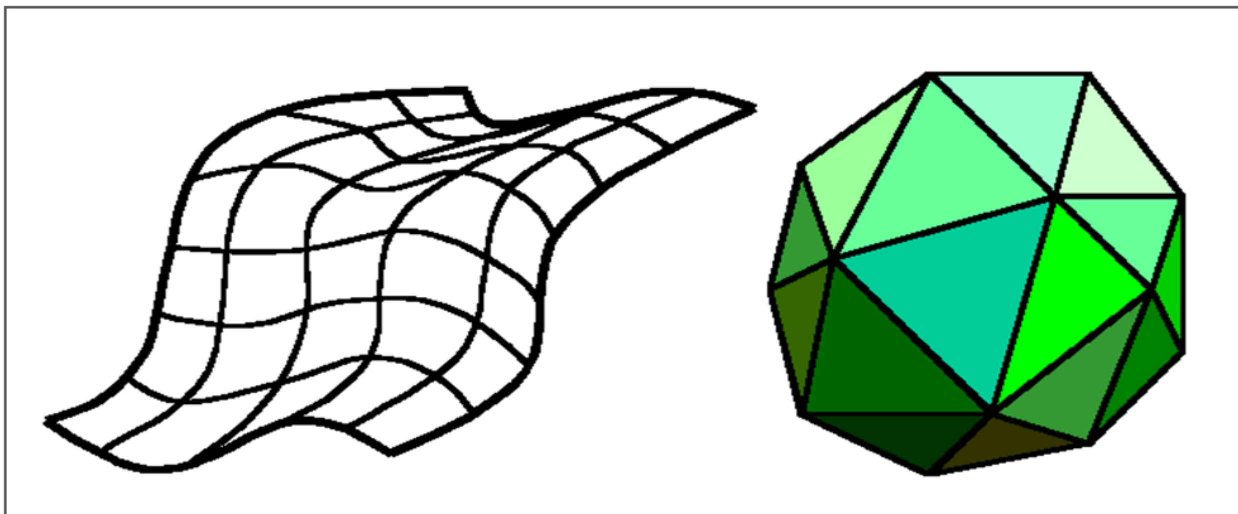
Implicit Curves and Surfaces

- Advantages
  - Easy to determine inside/outside
  - Easy to determine if a point is on the curve/surface
- Disadvantages
  - Hard to generate points on the curve/surface
  - Does not lend itself to (real-time) rendering

Polygonal Meshes

Polygonal Meshes

- Boundary representations of objects
  - Surfaces, polyhedrons

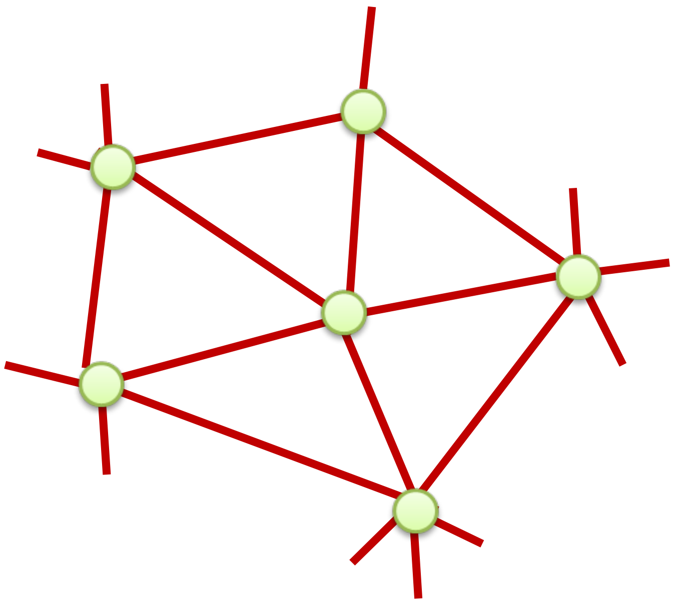


  - How are these objects stored?

Definitions

Geometric graph

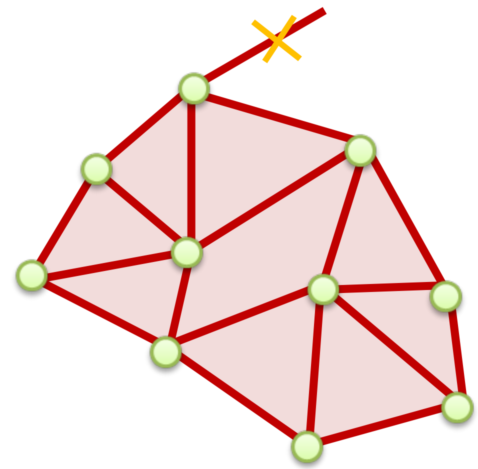
- A graph is a pair $G=(V, E)$
  - V is a set of n distinct vertices $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$
  - E is a set of edges $(\mathbf{v}_i, \mathbf{v}_j)$
- If $V \subset R^d$ with $d \geq 2$, then $G=(V, E)$ is a *geometric graph*
- The *degree* or *valence* of a vertex describes the number of edges incident to this vertex



Definitions

Polygonal mesh

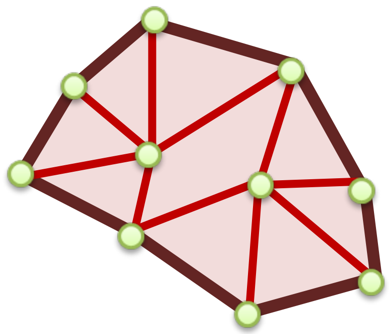
- A finite set M of closed, simple polygons Q_i is a **polygonal mesh** if:
  - The intersection of enclosed regions of any two polygons in M is empty
  - The intersection of two polygons in M is either empty, a vertex $\mathbf{v} \in V$ or an edge $\mathbf{e} \in E$
  - Every edge belongs to at least one polygon



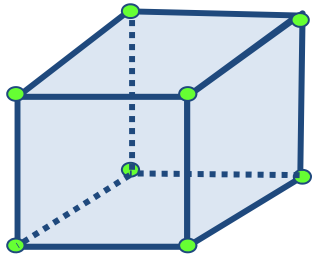
Definitions

Polygonal mesh

- The set of all edges that belong to only one polygon is termed the ***boundary*** of the polygonal mesh, and is either empty or forms closed loops



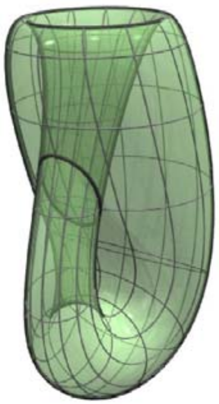
- If the set of edges that belong to only one polygon is empty, then the polygonal mesh is *closed*
- The set of all vertices and edges in a polygonal mesh form a graph



Definitions

Orientability

- A polygonal mesh is orientable, if the incident faces to every edge can be equally oriented
  - If the faces are equally oriented for every edge, the mesh is *oriented*
- Notes
  - Every **non-orientable closed** mesh embedded in R^3 intersects itself
  - The surface of a polyhedron is always orientable



Klein bottle

Möbius strip

Data structures for meshes

Space requirements

- Coordinates/attributes

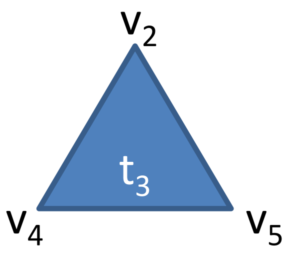
3 x 16 + k bits/vertex

vertex 1	x	y	z	c
vertex 2	x	y	z	c
vertex 3	x	y	z	c

- Connectivity

3 x $\log_2 V$ bits/triangle

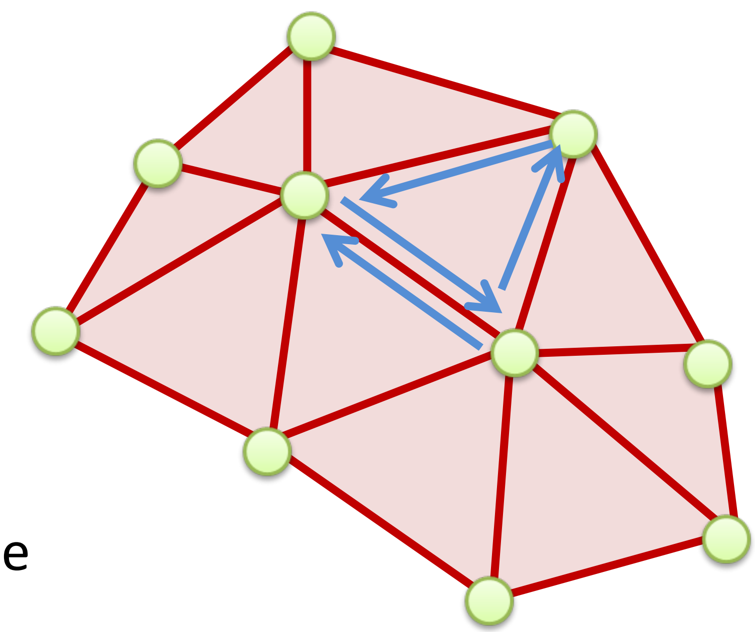
triangle 1	1	2	3
triangle 2	3	2	4
triangle 3	4	2	5
triangle 4	7	5	6
triangle 5	6	5	8



Data structures for meshes

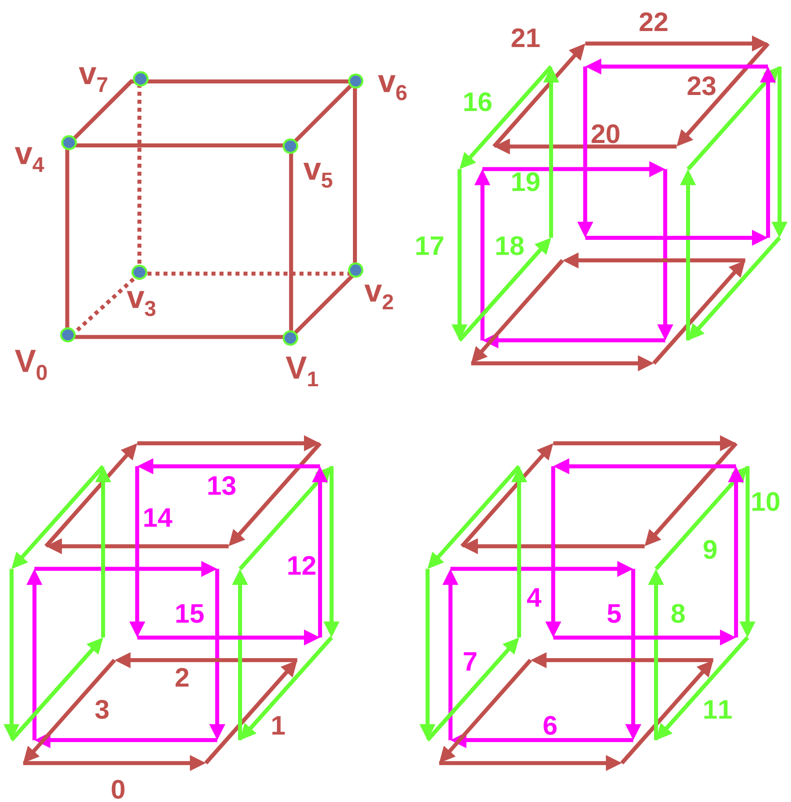
Half-edge data structure

- Half-edge has:
  - Pointer to twin h-e
  - Pointer to origin vertex
  - Pointer to next h-e
  - Pointer to previous h-e
  - Pointer to incident face
- Vertex has:
  - Pointer to one emanating h-e
- Face has:
  - Pointer to one of its enclosing h-e



Data structures for meshes

Half-edge data structure



Vertexlist

v	coord			he
0	0.0	0.0	0.0	0
1	1.0	0.0	0.0	1
2	1.0	1.0	0.0	2
3	0.0	1.0	0.0	3
4	0.0	0.0	1.0	4
5	1.0	0.0	1.0	9
6	1.0	1.0	1.0	13
7	0.0	1.0	1.0	16

Face

f	e
0	e0
1	e8
2	e4
3	e16
4	e12
5	e20

Half-Edgelist

he	vstart	next	prev	opp	he	vstart	next	prev	opp
0	0	1	3	6	12	2	13	15	10
1	1	2	0	11	13	6	14	12	22
2	2	3	1	15	14	7	15	13	19
3	3	0	2	18	15	3	12	14	2
4	4	5	7	20	16	7	17	19	21
5	5	6	4	8	17	4	18	16	7
6	1	7	5	0	18	0	19	17	3
7	0	4	6	17	19	3	16	18	14
8	1	9	11	5	20	5	21	23	4
9	5	10	8	23	21	4	22	20	16
10	6	11	9	12	22	7	23	21	13
11	2	8	10	1	23	6	20	22	9

Next week

- More about shape representations
- Parametric representations
  - Bezier curves
  - Splines
- First Homework Assignment (warmup)

Thanks

(and please register if you want to take the course)