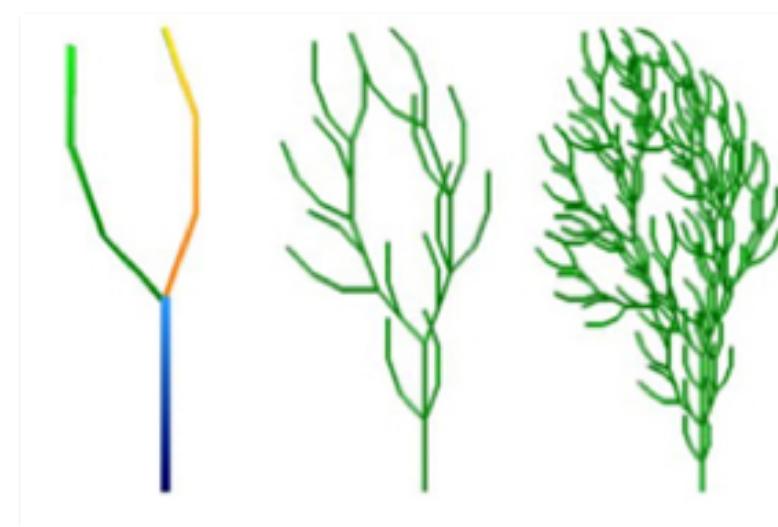
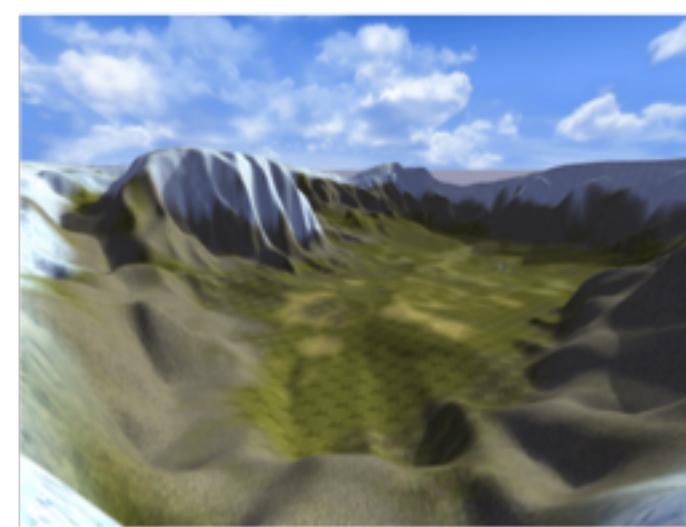
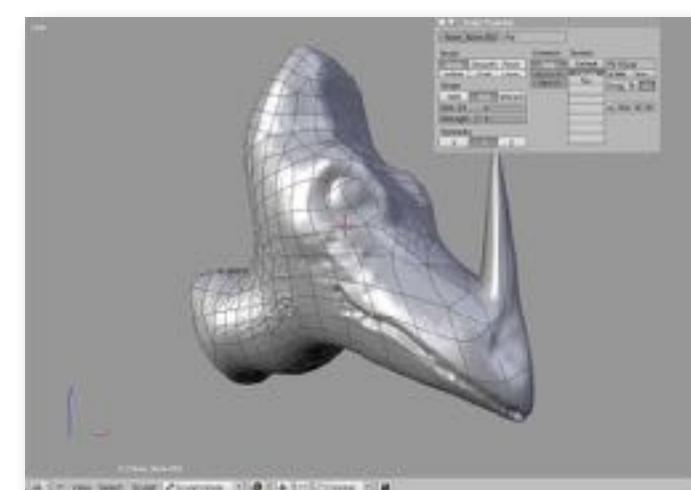
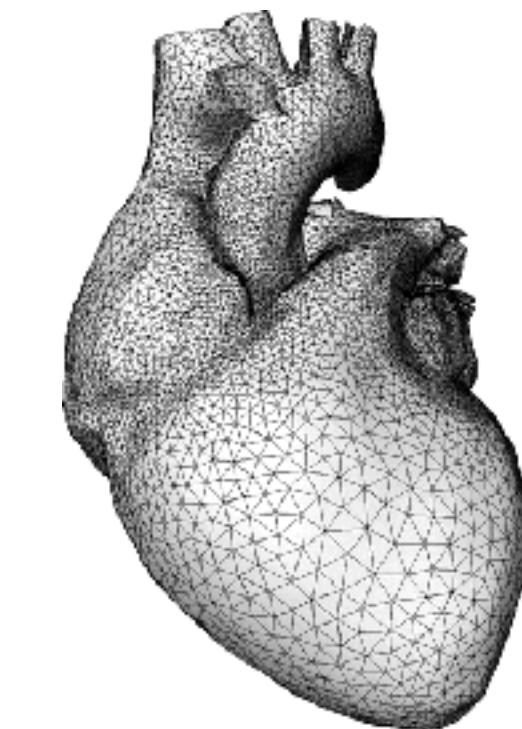


Shape Representation

Acknowledgements: *Daniele Panozzo*
CSC 486B/586B - Geometric Modeling - Teseo Schneider

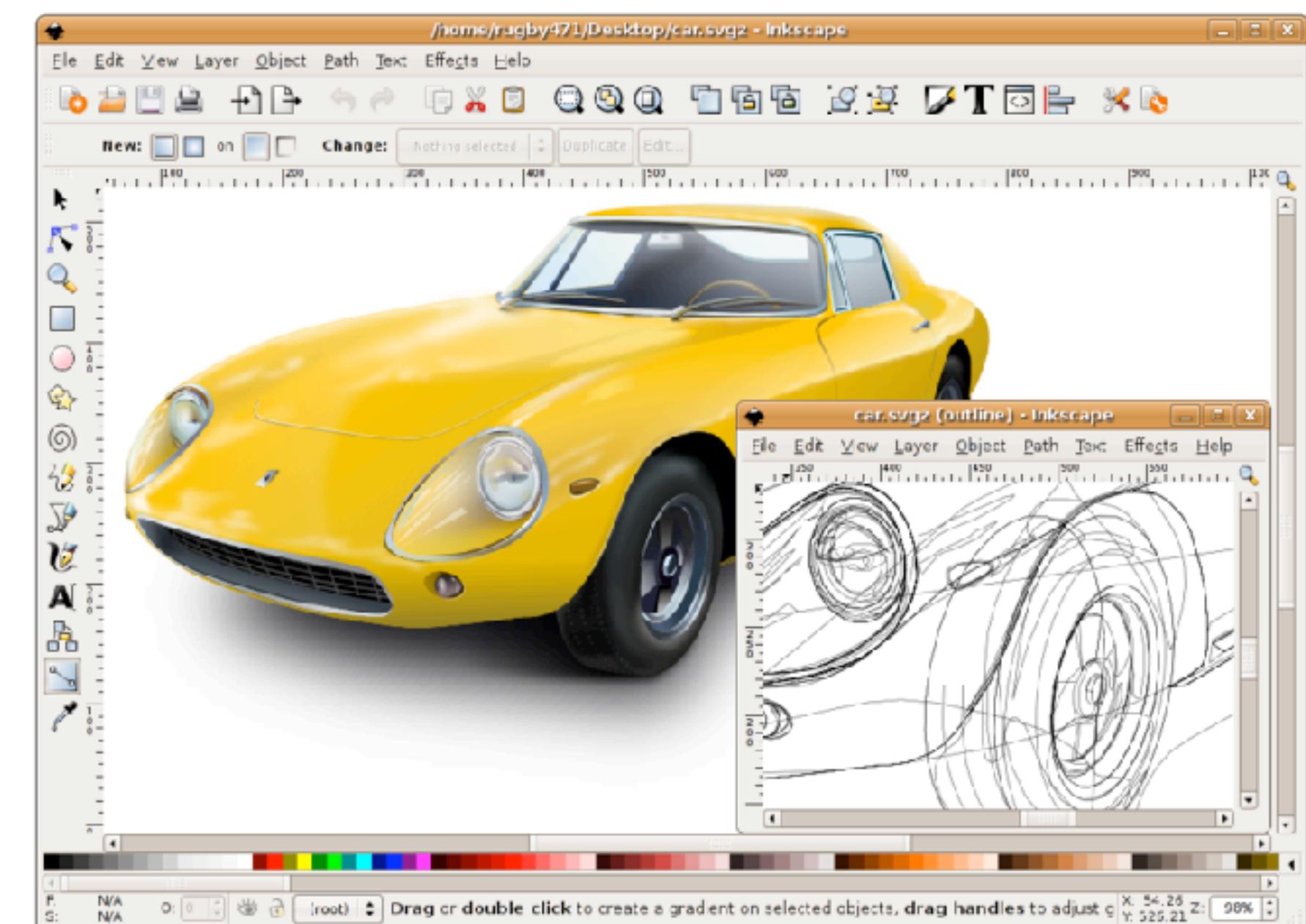
Shape Representation: Origin- and Application-Dependent

- Acquired real-world objects:
 - Discrete sampling
 - Points, meshes
- Modeling “by hand”:
 - Higher-level representations, amendable to modification, control
 - Parametric surfaces, subdivision surfaces, implicits
- Procedural modeling
 - Algorithms, grammars



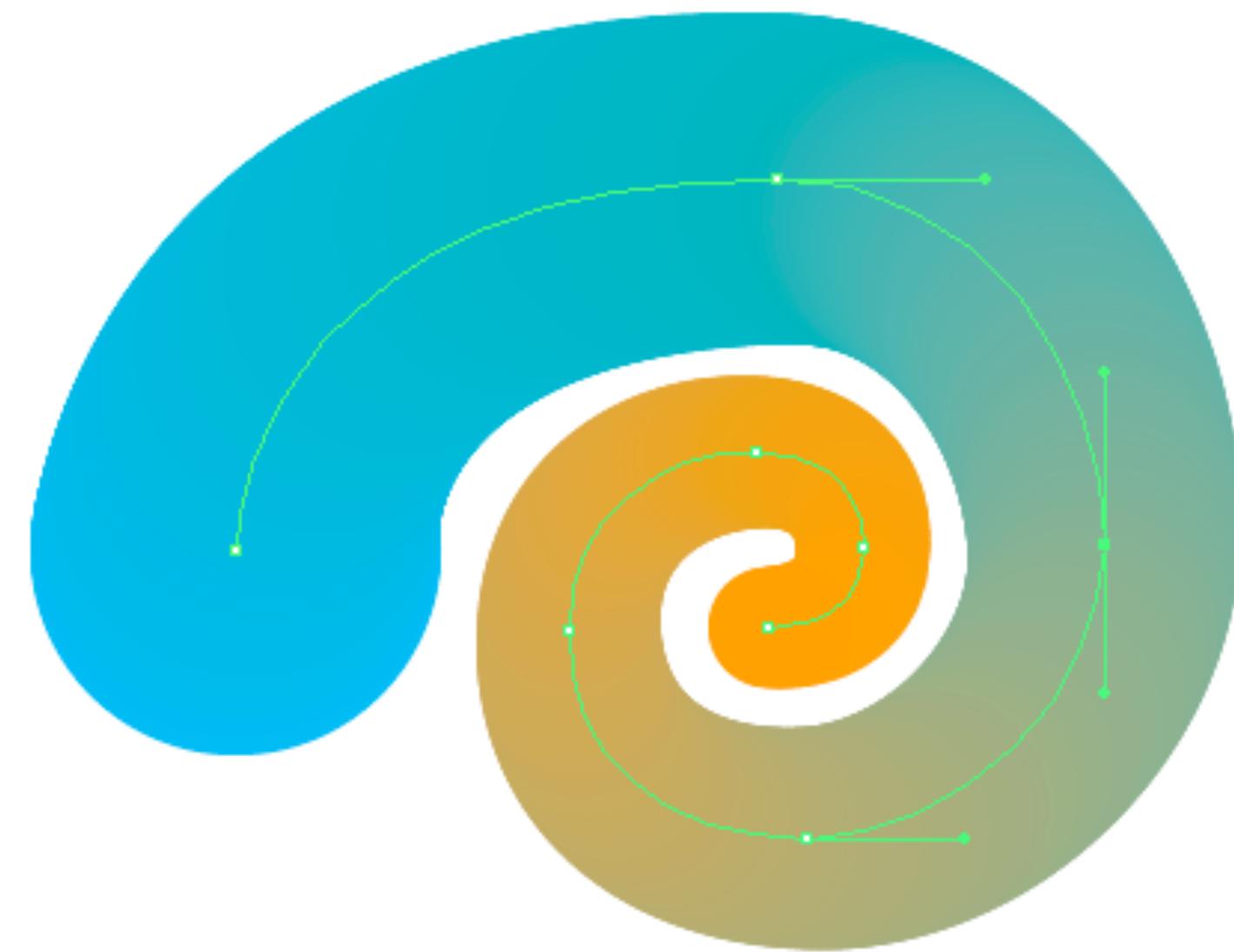
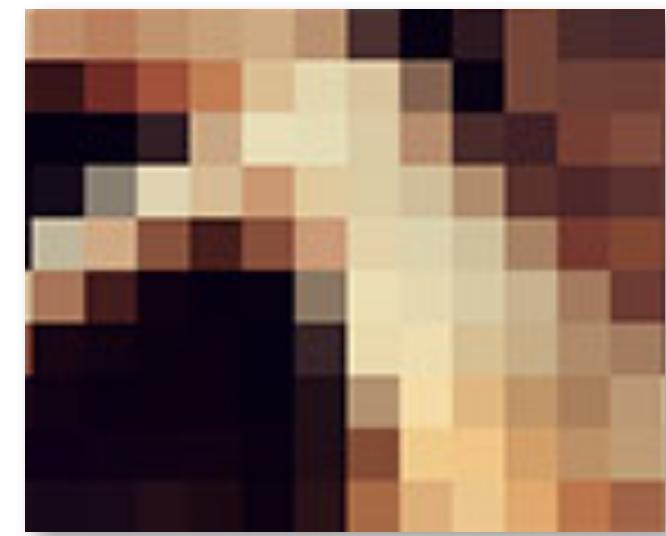
Similar to the 2D Image Domain

- Acquired digital images:
 - Discrete sampling
 - Pixels on a grid
- Painting “by hand”:
 - Strokes + color/shading
 - Vector graphics
 - Controls for editing



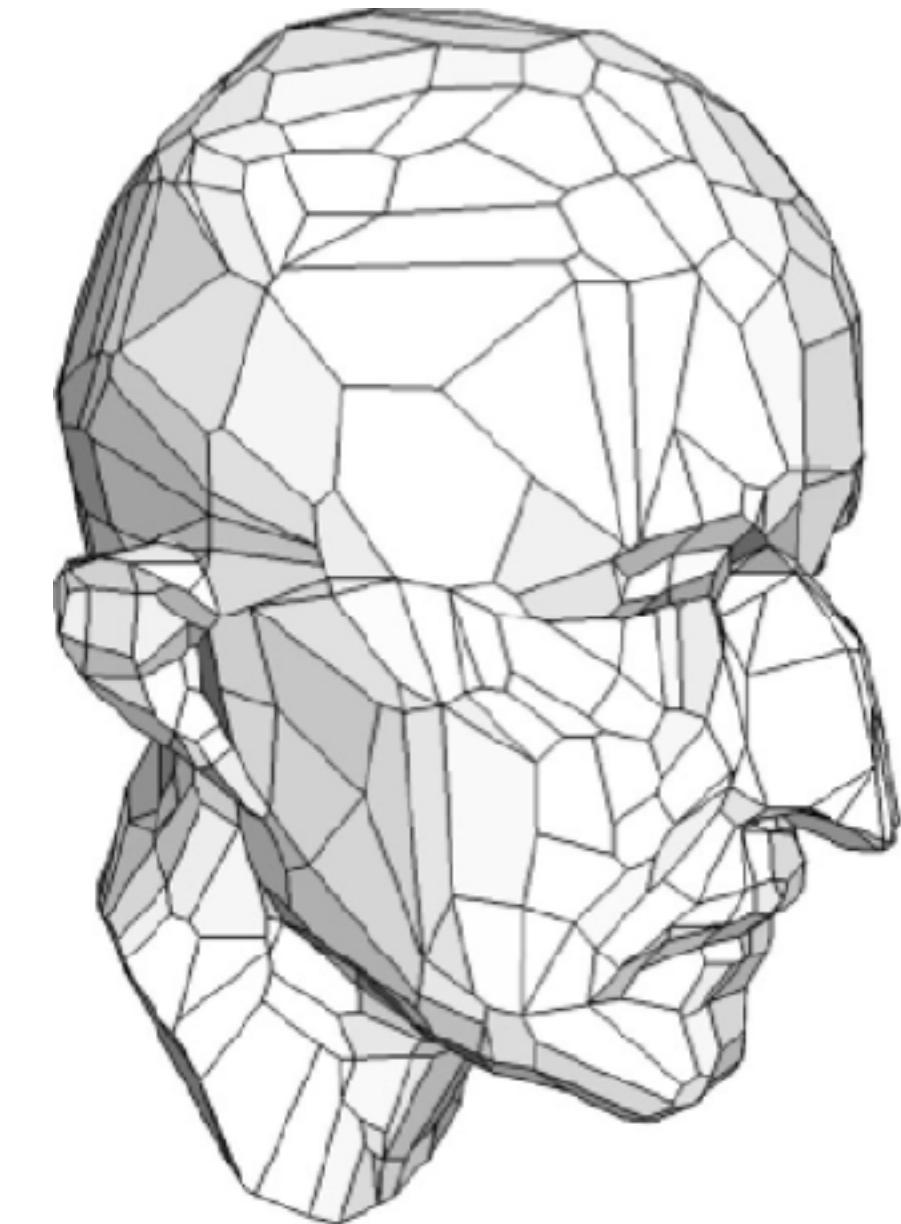
Similar to the 2D Image Domain

- Acquired digital images:
 - Discrete sampling
 - Pixels on a grid
- Painting “by hand”:
 - Strokes + color/shading
 - Vector graphics
 - Controls for editing



Representation Considerations

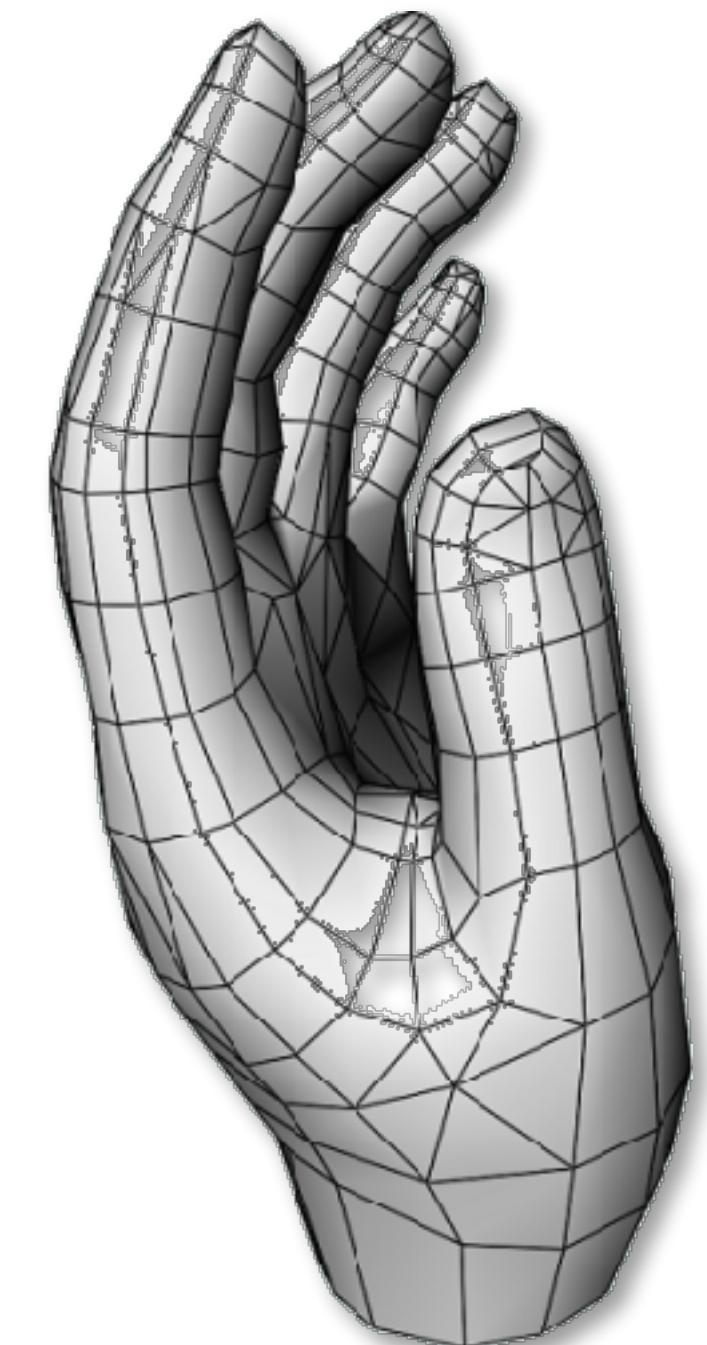
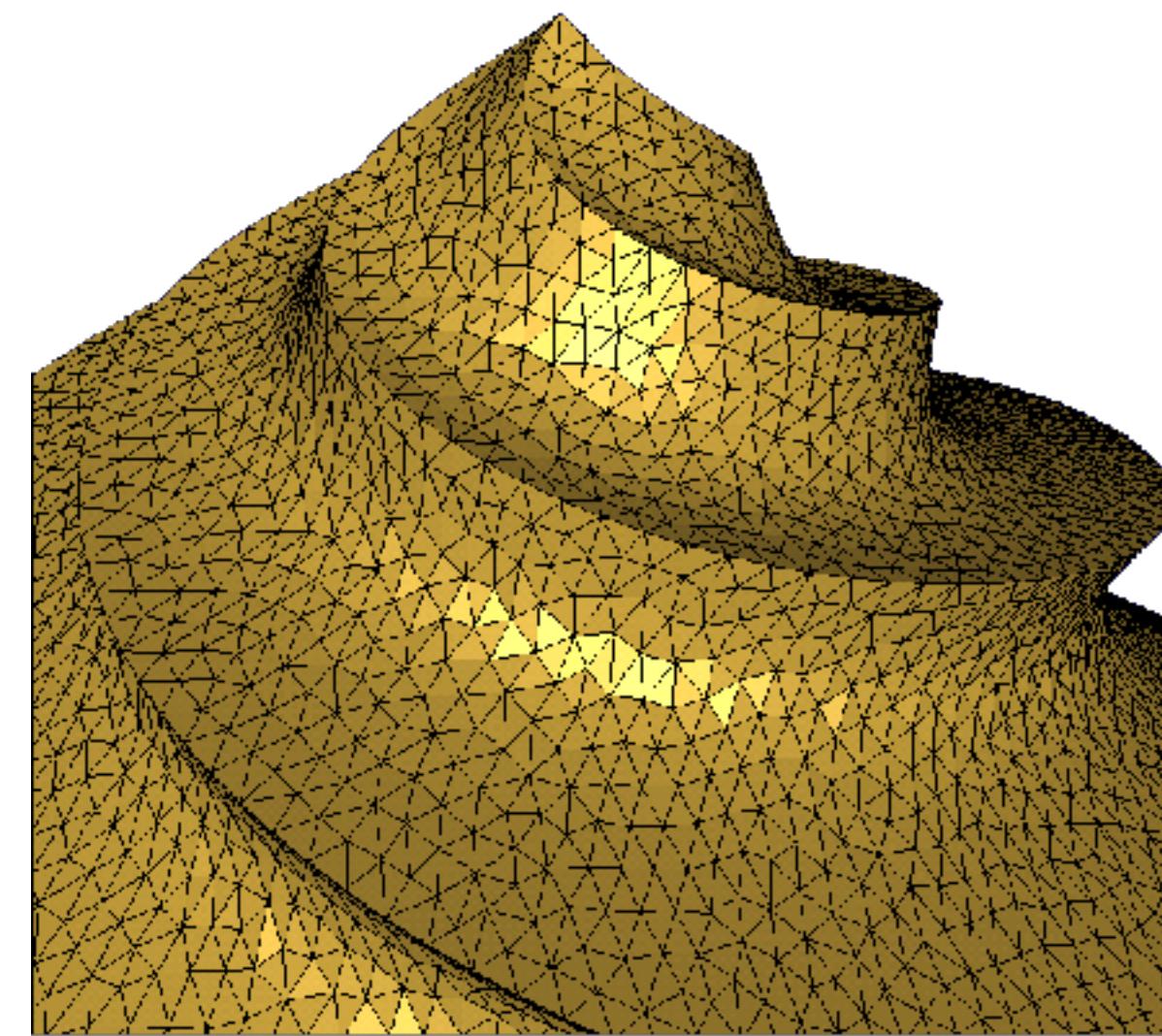
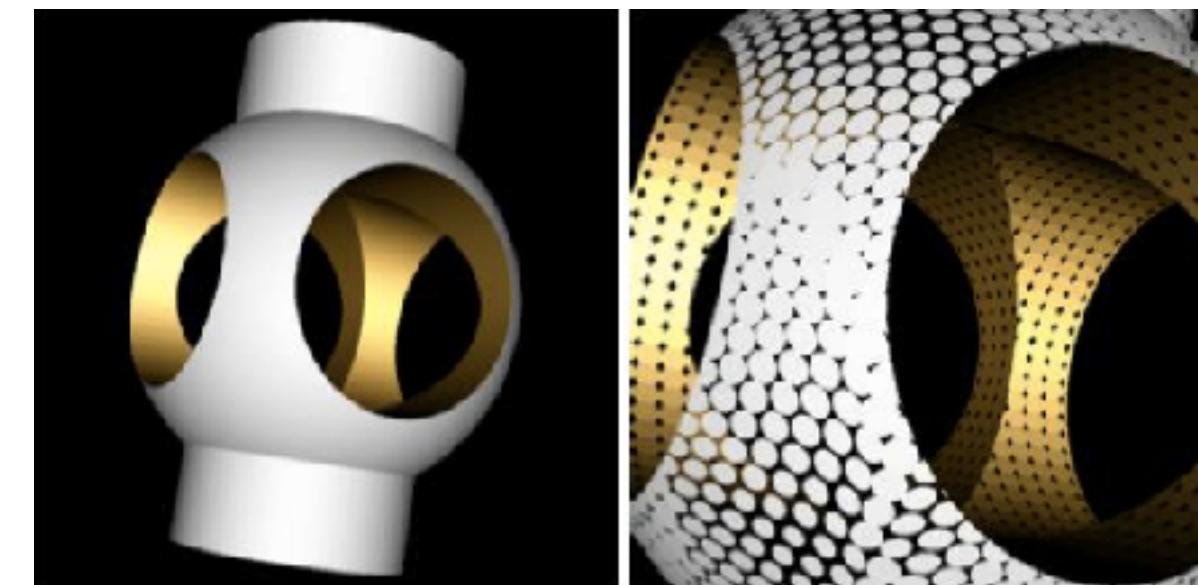
- How should we represent geometry?
 - Needs to be stored in the computer
 - Creation of new shapes
 - Input metaphors, interfaces...
 - What operations do we apply?
 - Editing, simplification, smoothing, filtering, repair...
 - How to render it?
 - Rasterization, raytracing...



Variational Shape Approximation

Shape Representations

- Points
- Polygonal meshes



Anisotropic Polygonal Remeshing

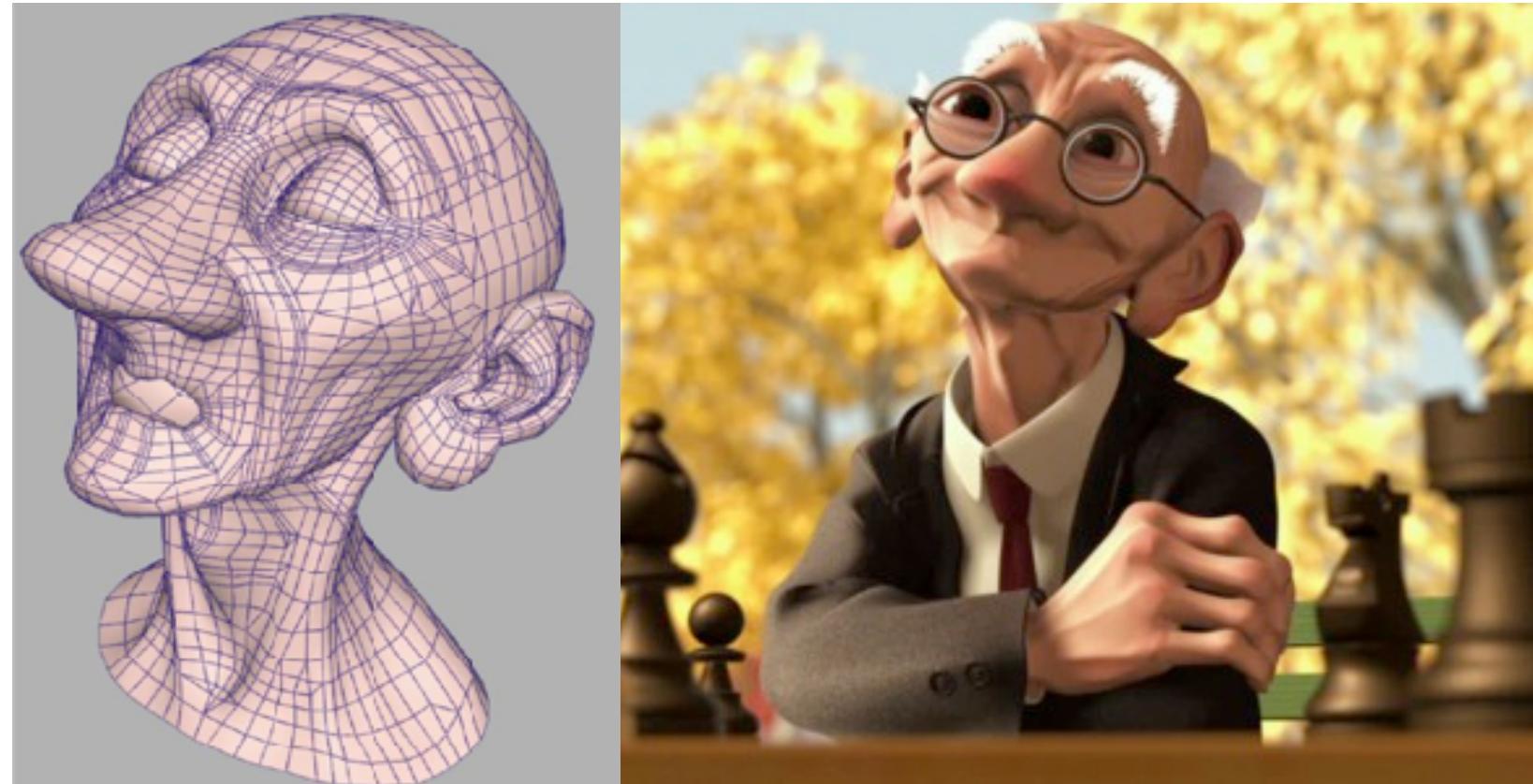
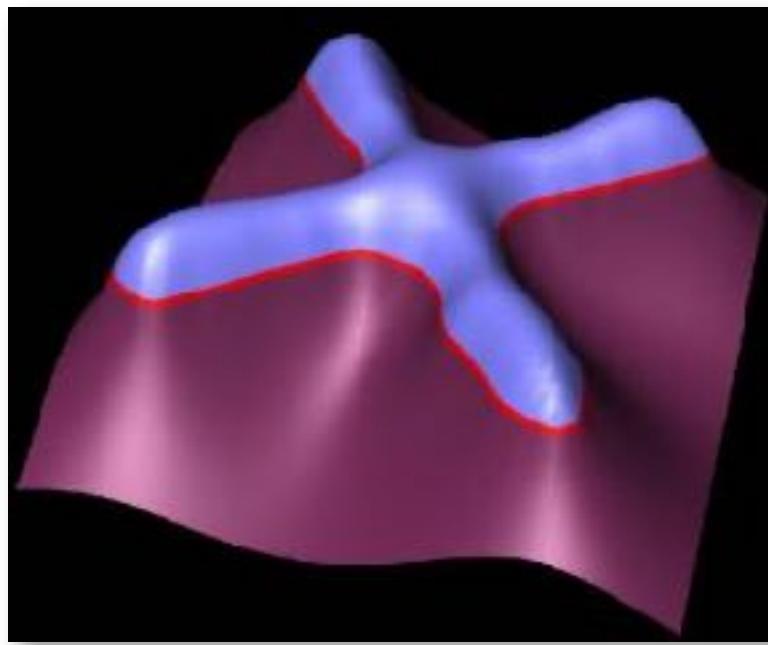


**University
of Victoria**

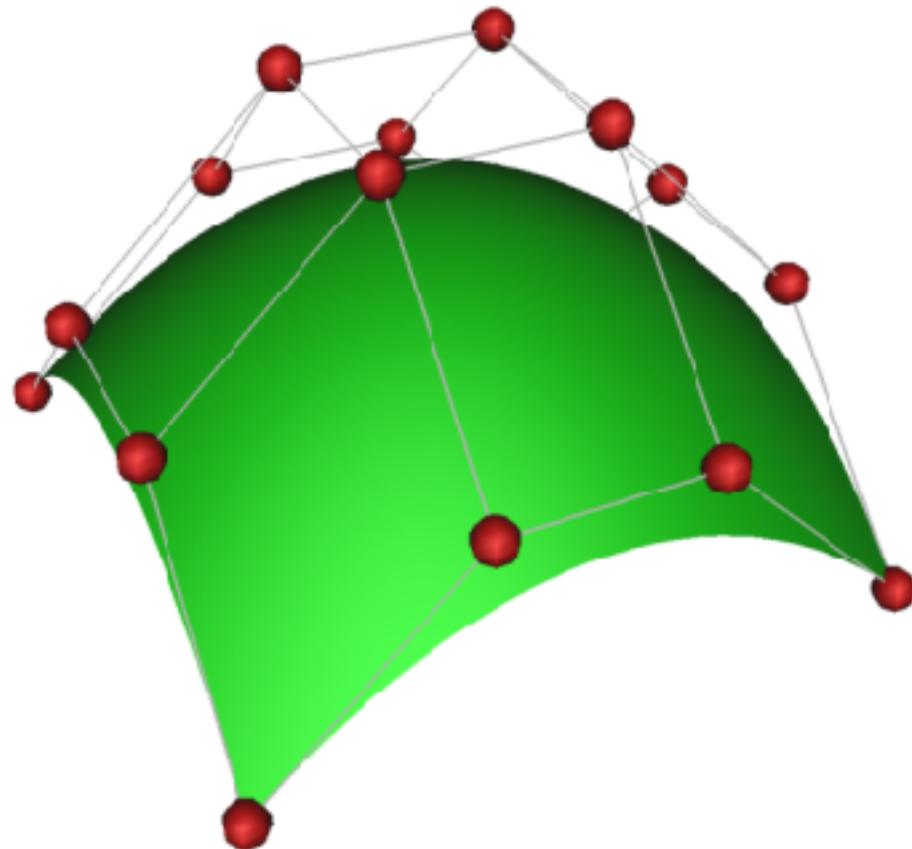
Computer Science

Shape Representations

- Parametric surfaces
- Subdivision surfaces
- Implicit functions



Pixar



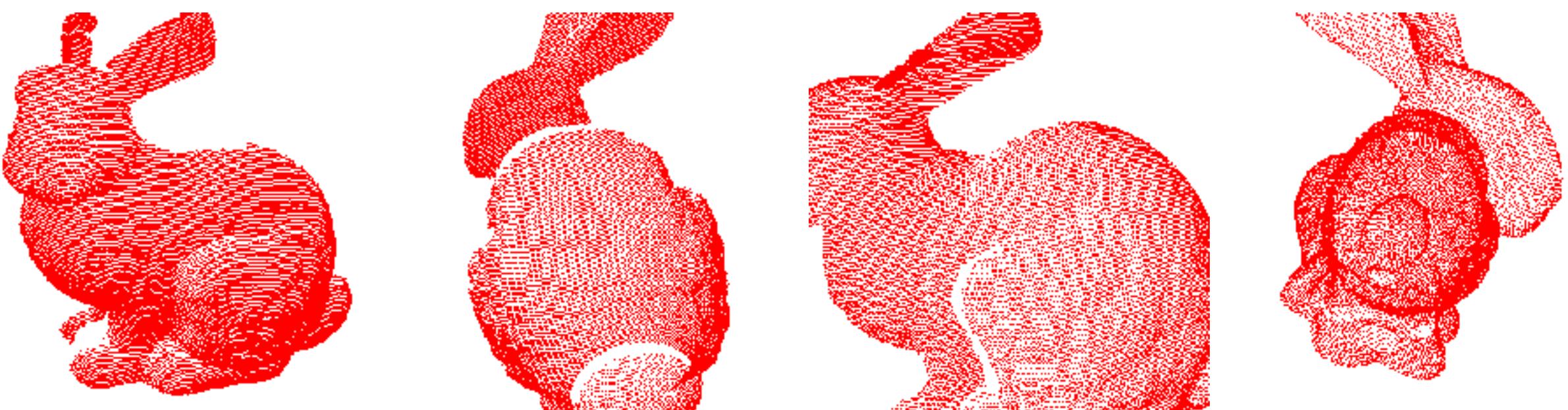
Points

Output of Acquisition



Points

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

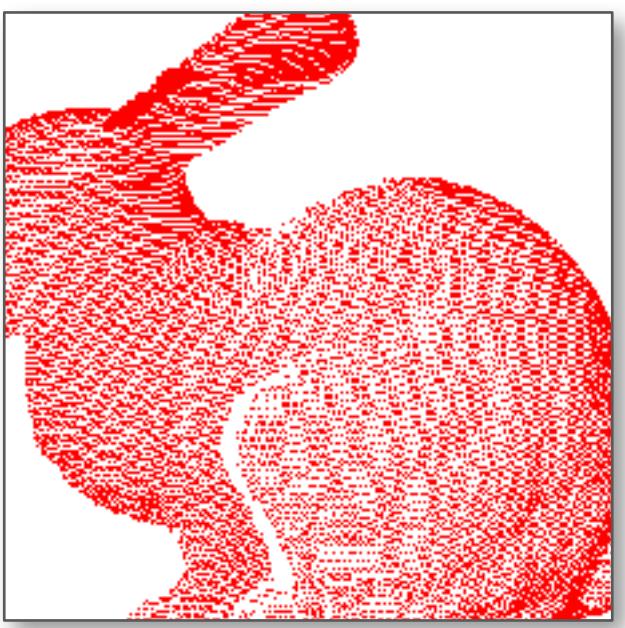


- Depth imaging
- Registration of multiple images



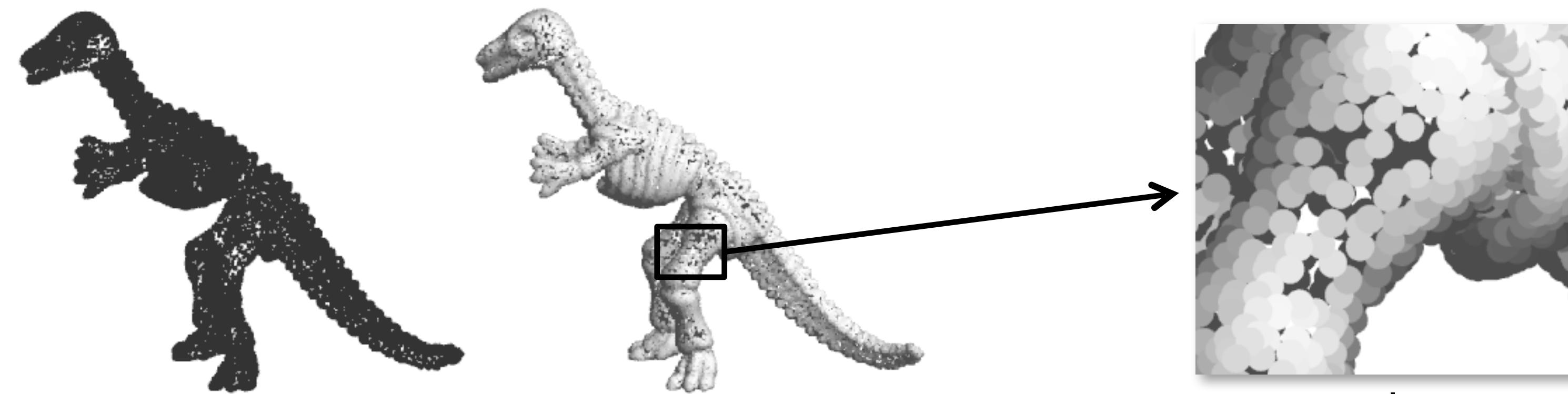
Points

- points = set of 3-tuples (often a collection of grids)
- 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?

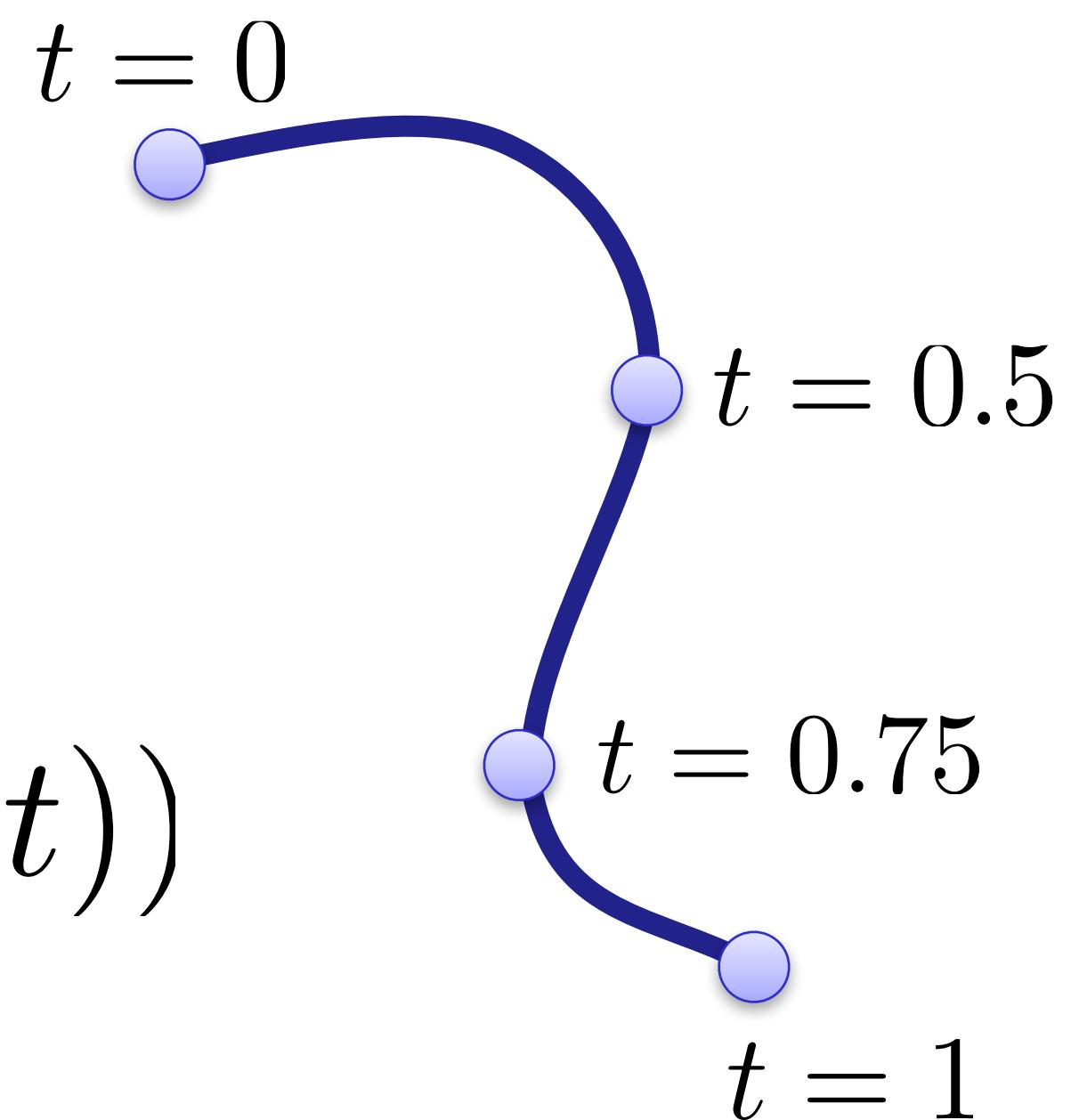


- Need sub-linear-time implementations of
 - k-nearest neighbors to point \mathbf{x}
 - In-radius search $\|\mathbf{p}_i - \mathbf{x}\| < \varepsilon$

Parametric Curves and Surfaces

Parametric Representation of Curves

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

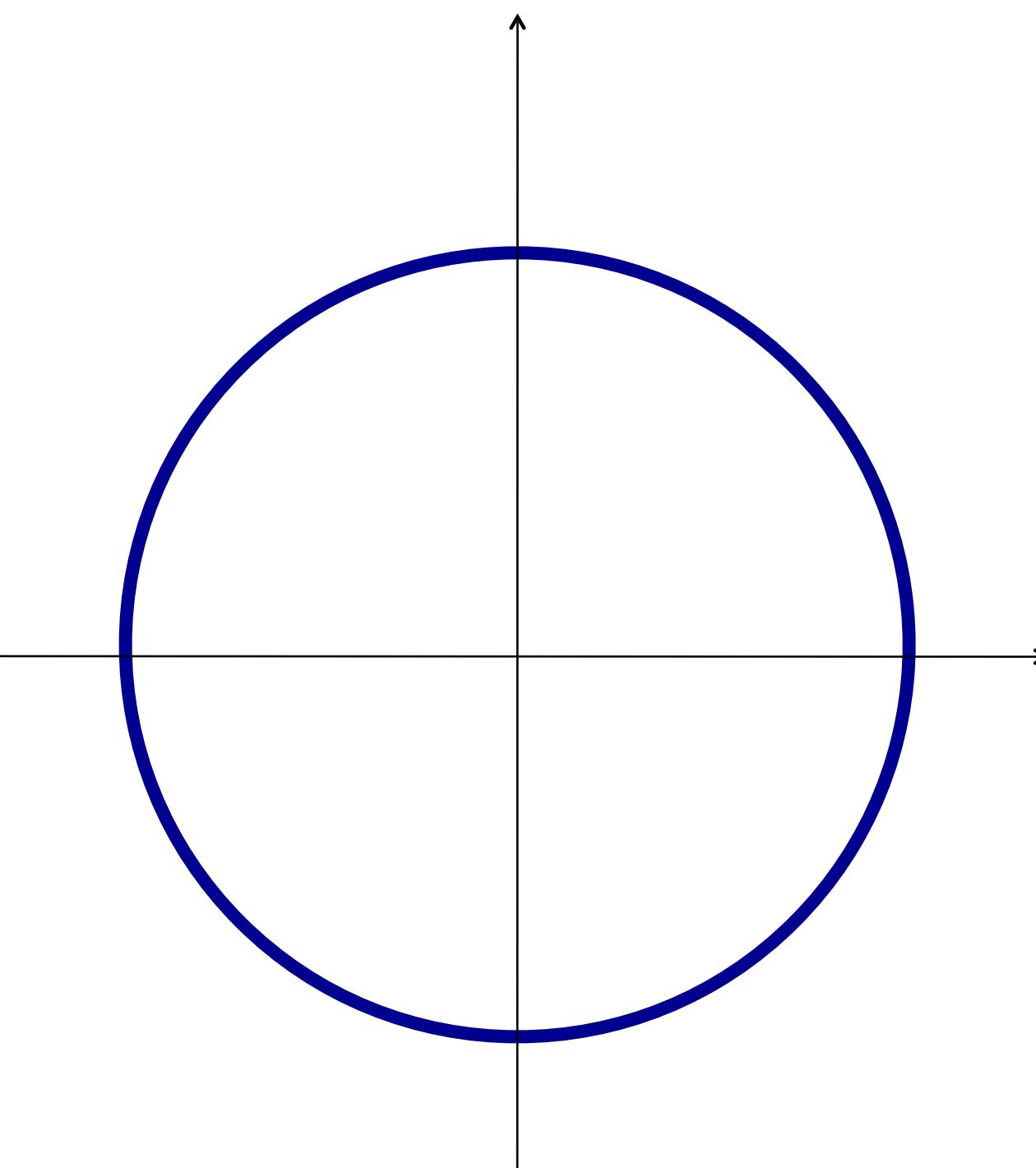


Parametric Curves

- Explicit curve/circle in 2D

$$\mathbf{p}(t) = r (\cos(t), \sin(t))$$

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

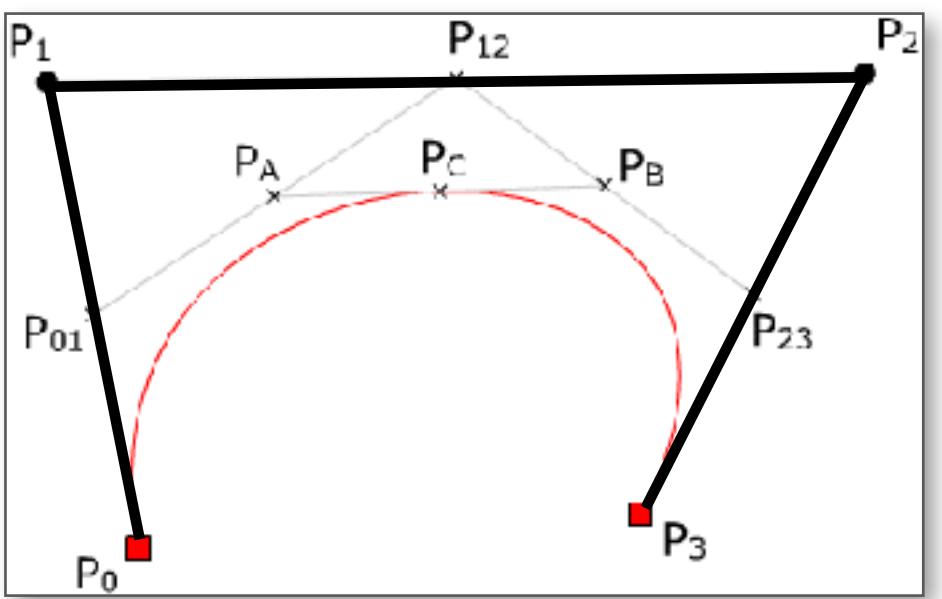


Parametric Curves

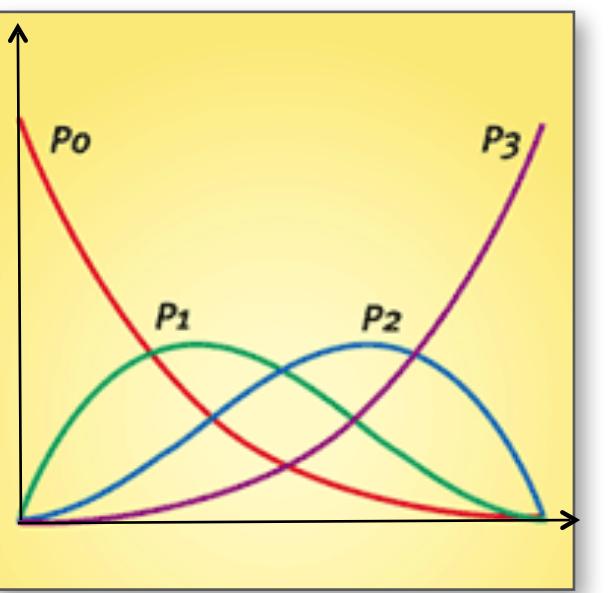
- Bezier curves, splines

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

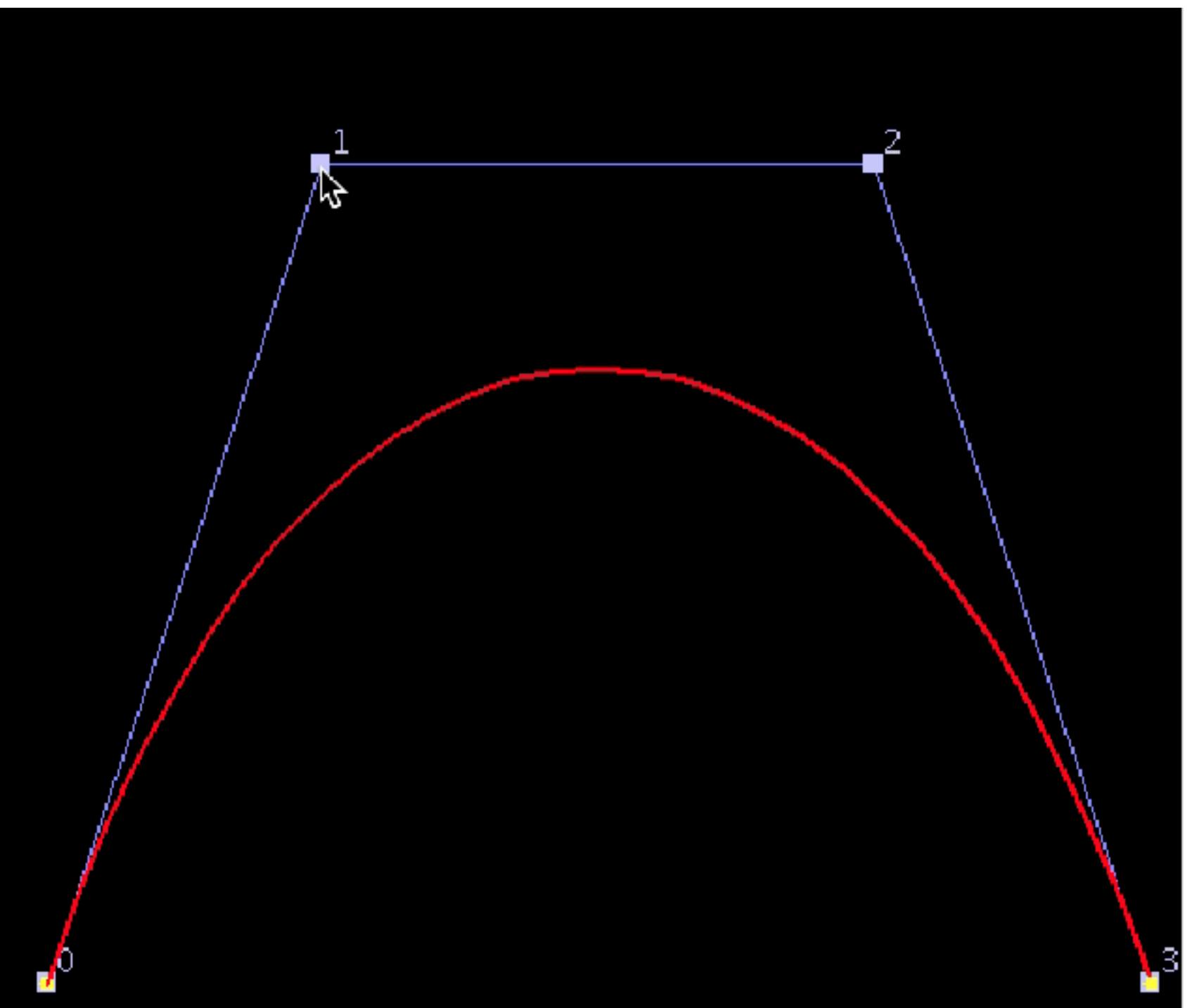
$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$



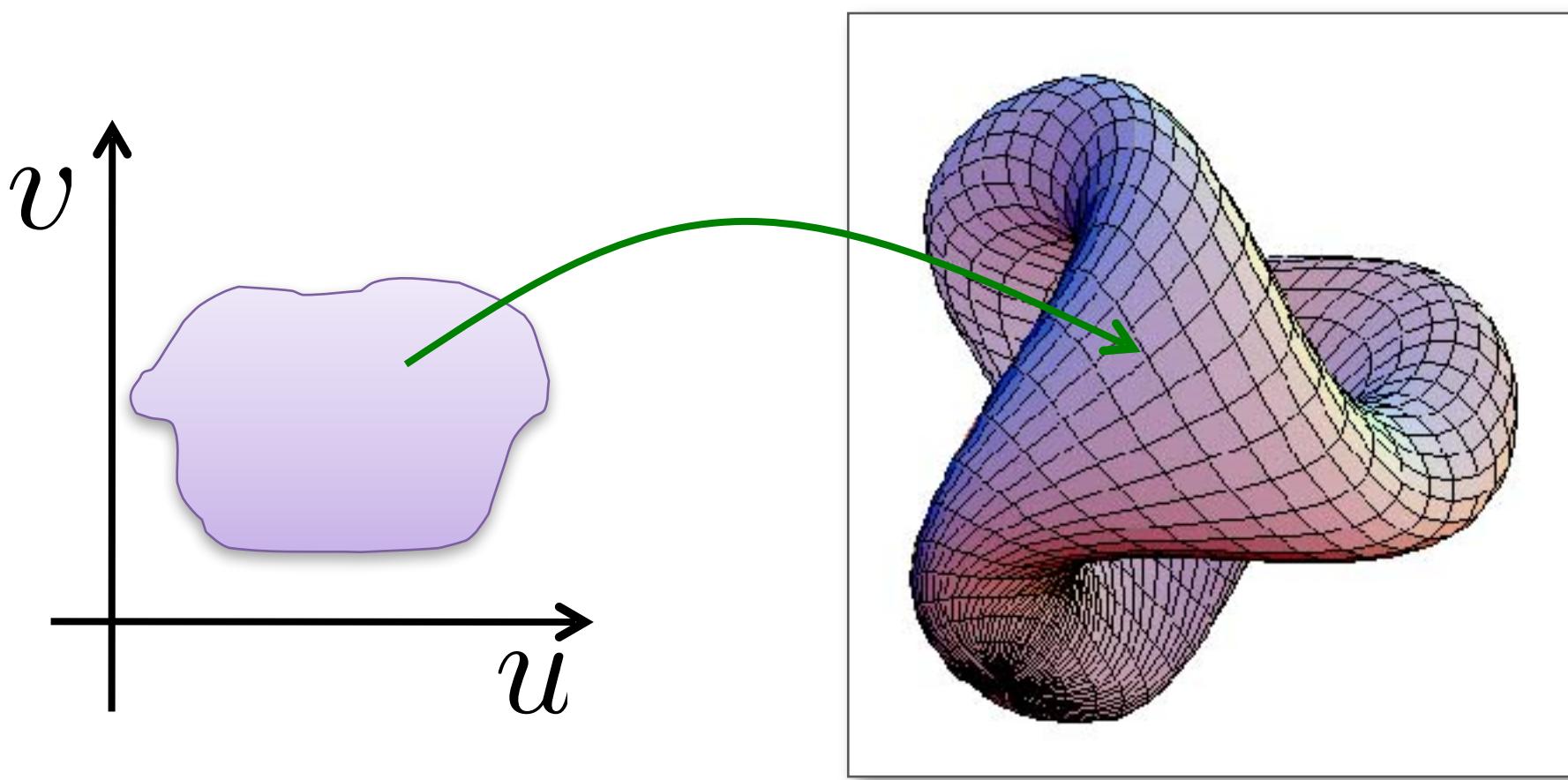
Curve and control polygon



Basis functions



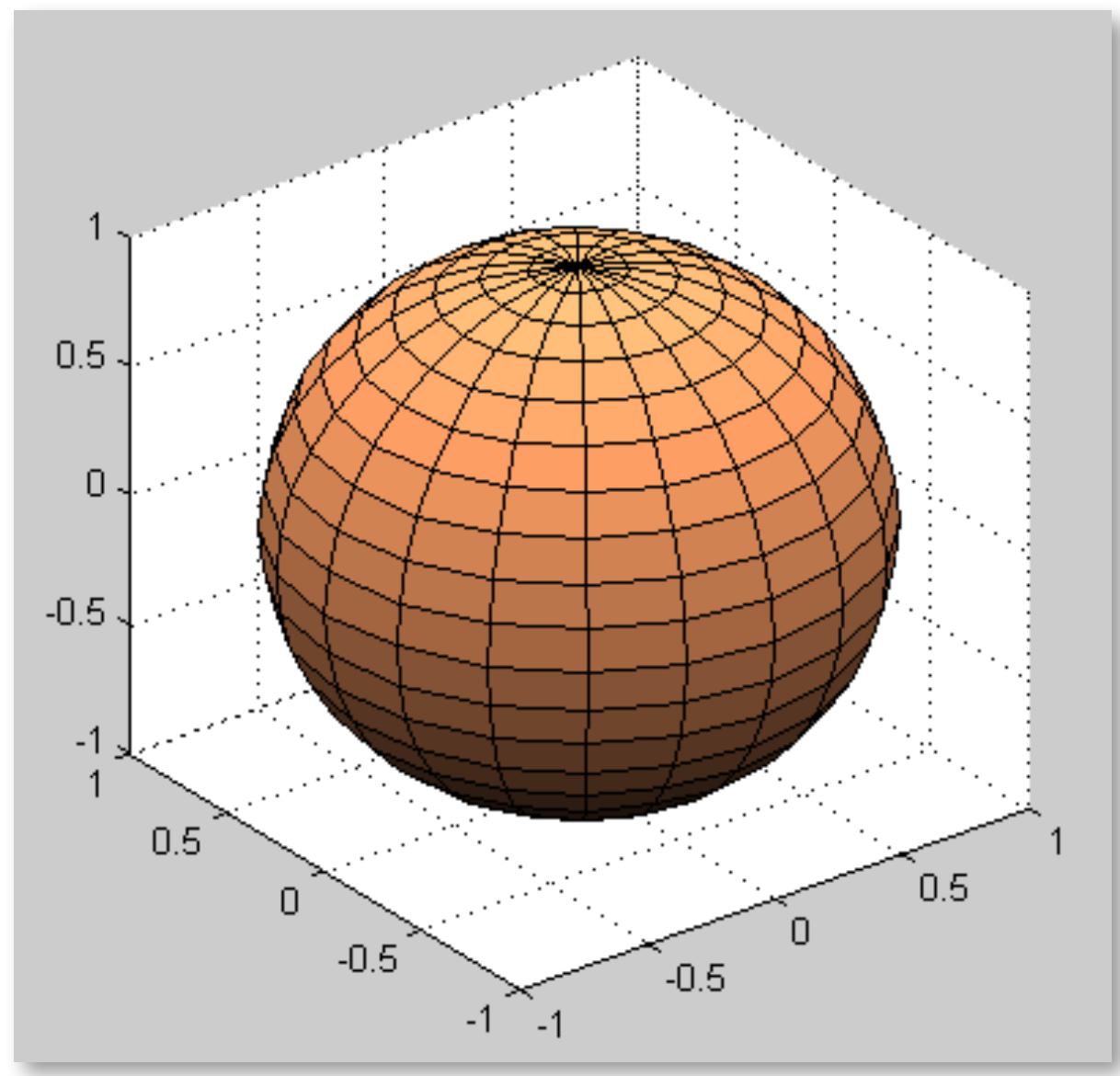
Parametric Representation of Surfaces



$$s(u, v) = (x(u, v), y(u, v), z(u, v))$$

Parametric Surfaces

- Sphere in 3D



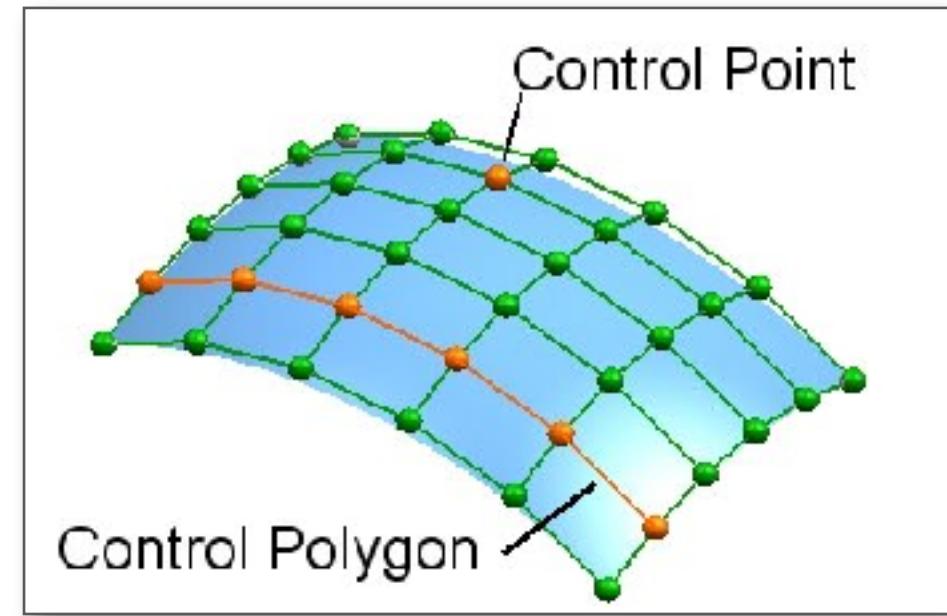
$$s(u, v) = r (\cos(u) \cos(v), \sin(u) \cos(v), \sin(v))$$

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

Parametric Surfaces

- Bezier surface:

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



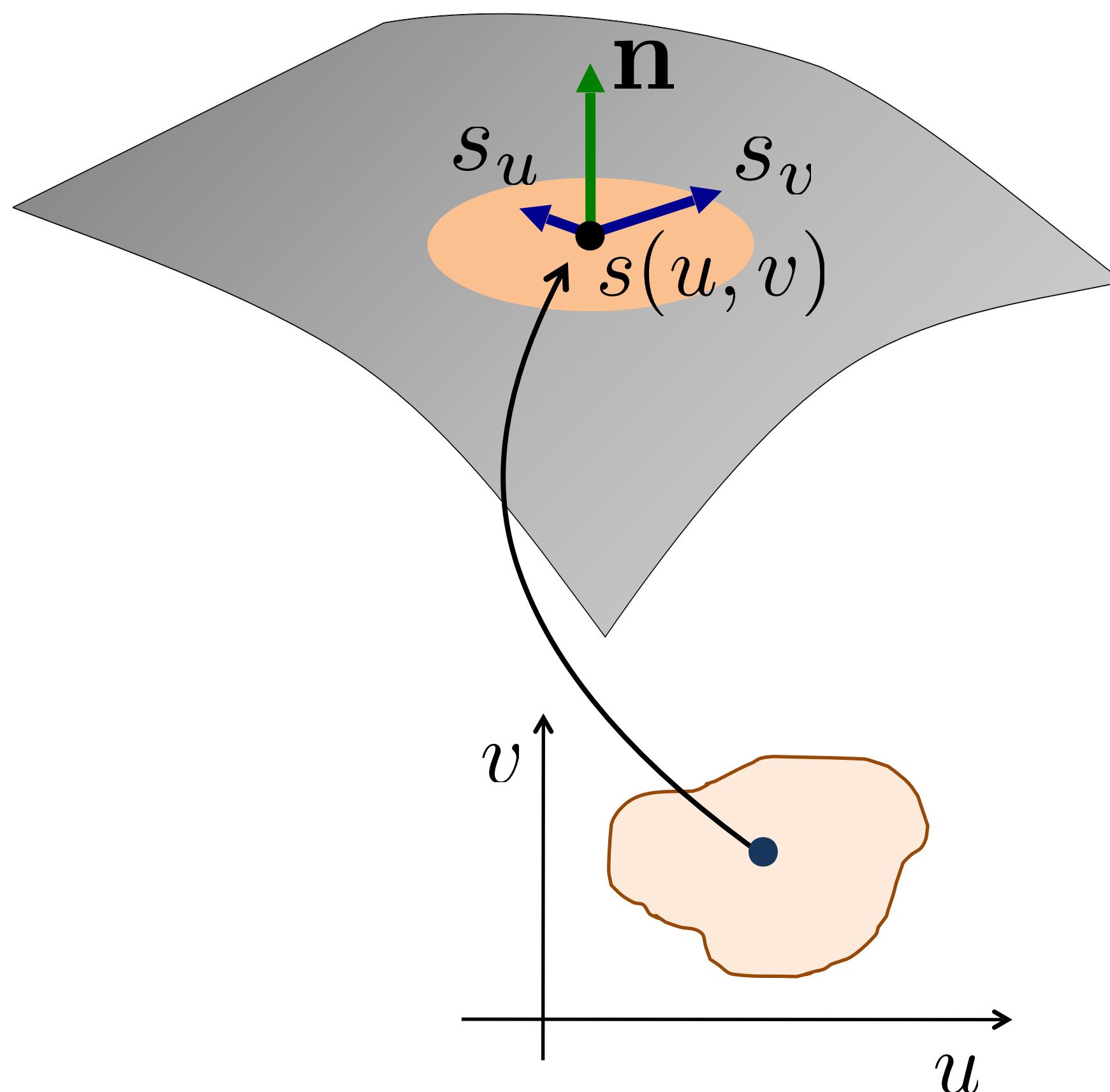
Tangents and Normal

$$s_u = \frac{\partial s(u, v)}{\partial u}$$

$$s_v = \frac{\partial s(u, v)}{\partial v}$$

$$\mathbf{n} = \frac{s_u \times s_v}{\|s_u \times s_v\|}$$

Tangent plane is normal to \mathbf{n}



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

- Kernel of a scalar function

- Curve in 2D:

$$S = \{x \in \mathbb{R}^2 | f(x) = 0\}$$

- Surface in 3D:

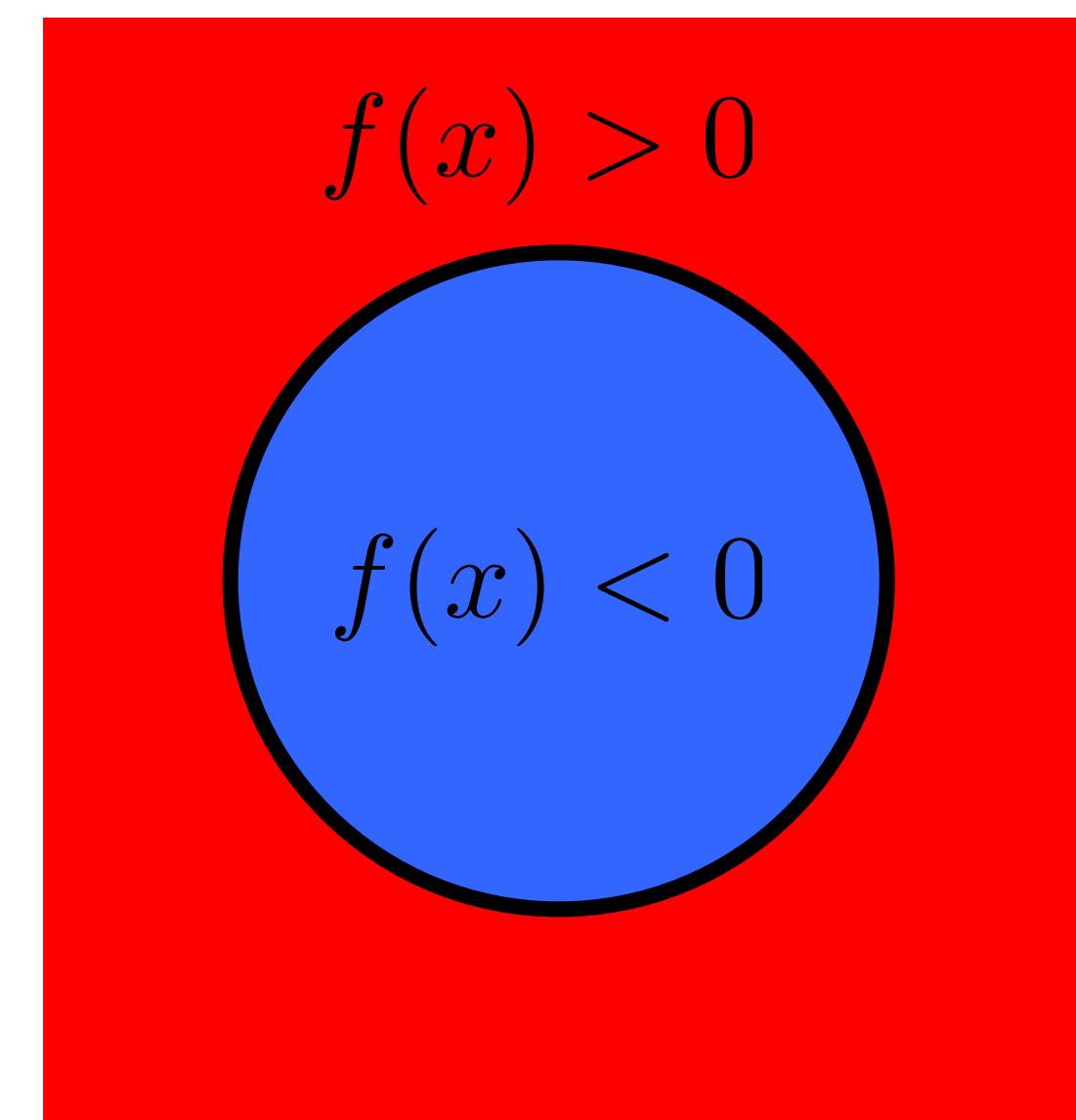
$$S = \{x \in \mathbb{R}^3 | f(x) = 0\}$$

- Space partitioning

$\{x \in \mathbb{R}^m | f(x) > 0\}$ Outside

$\{x \in \mathbb{R}^m | f(x) = 0\}$ Curve/Surface

$\{x \in \mathbb{R}^m | f(x) < 0\}$ Inside

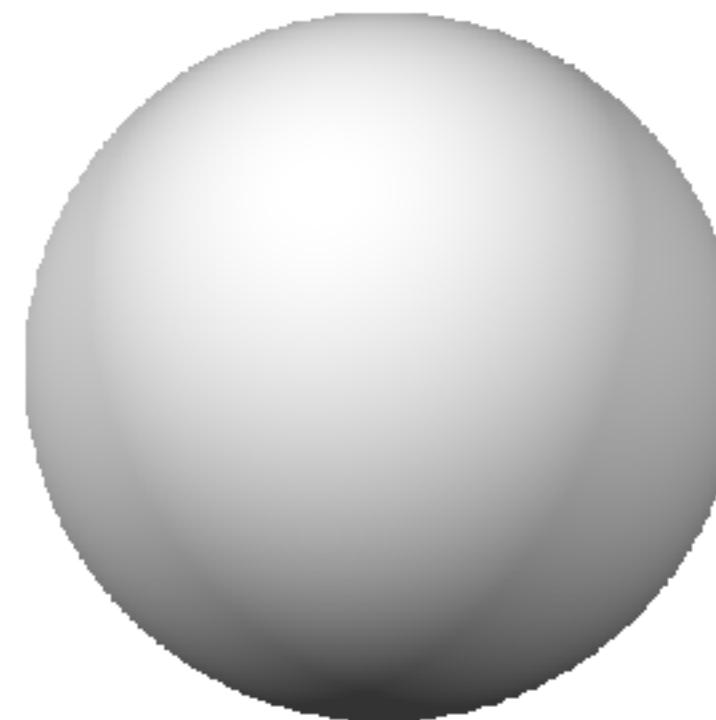
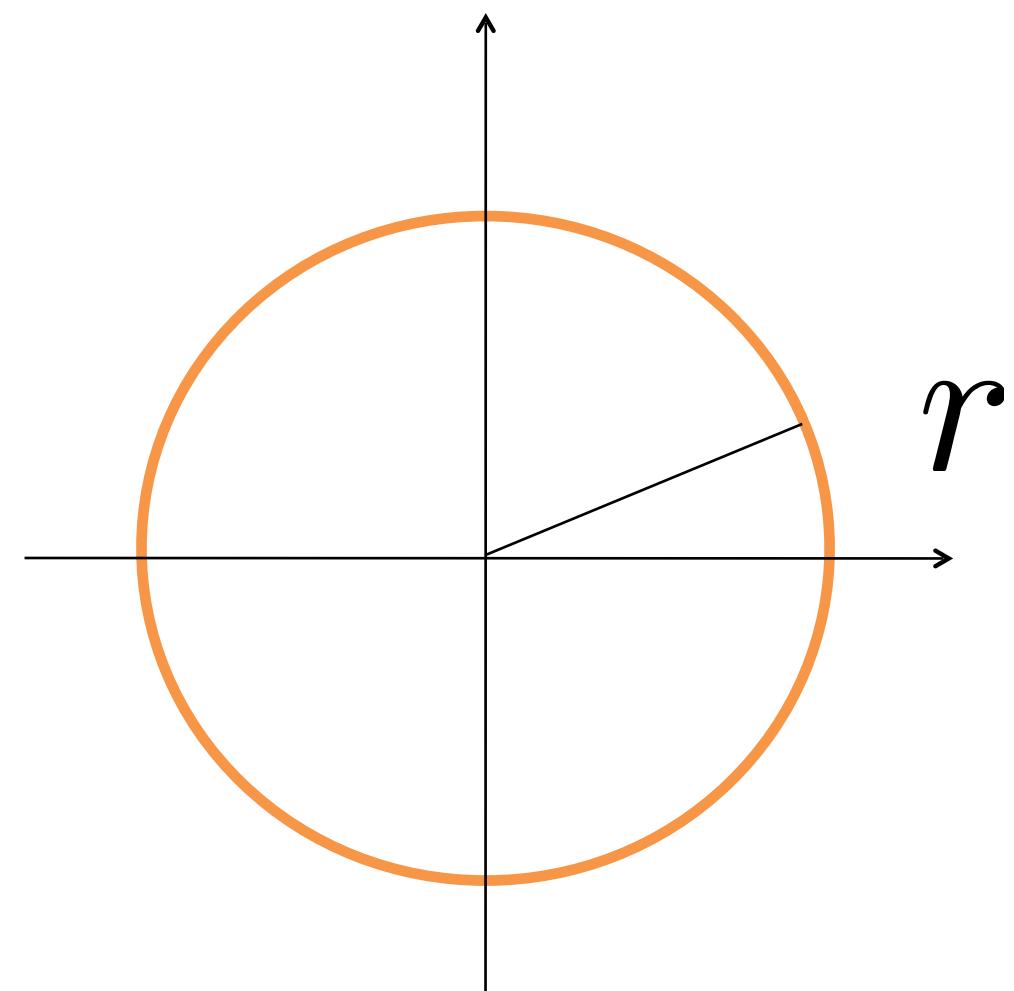


Implicit Curves and Surfaces

- Implicit circle and sphere

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

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



Implicit Curves and Surfaces

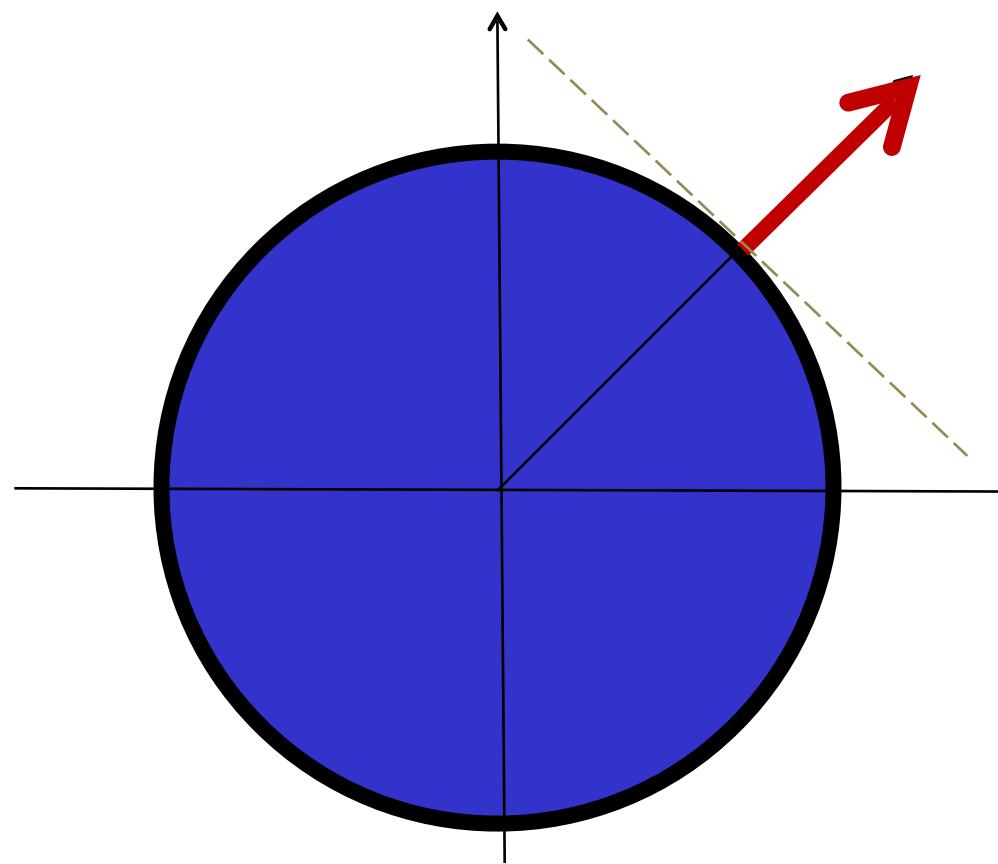
- The normal direction to the surface (curve) is given by the gradient of the implicit function

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

- Example

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

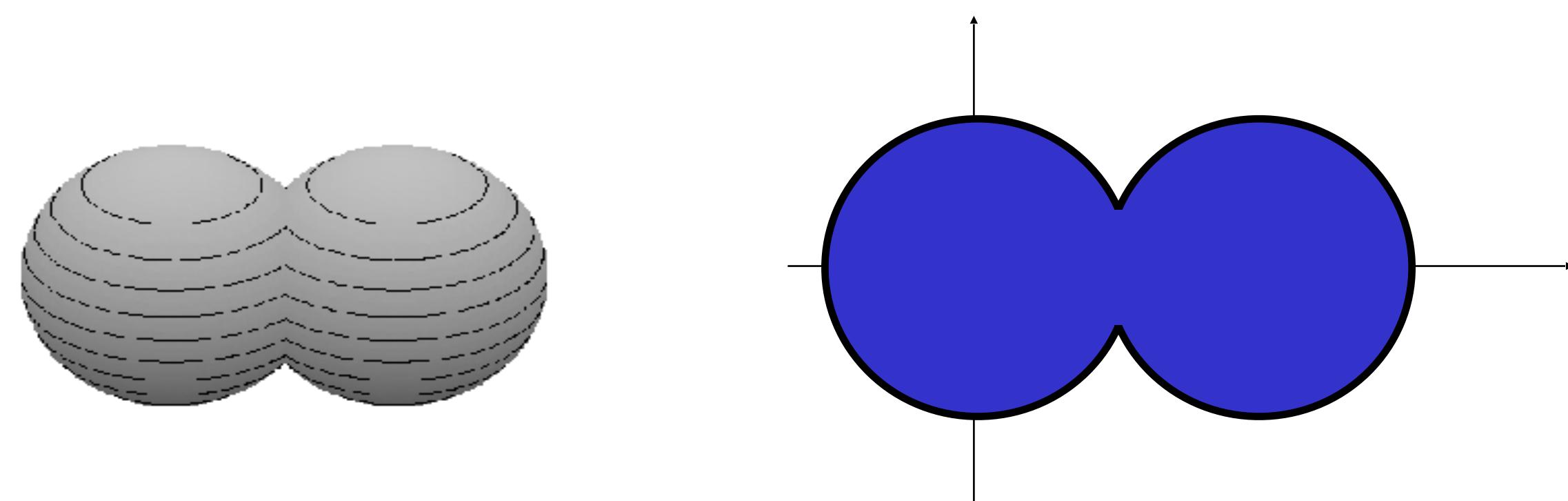
$$\nabla f(x, y, z) = (2x, 2y, 2z)^T$$



Boolean Set Operations

- Union:

$$\bigcup_i f_i(x) = \min f_i(x)$$



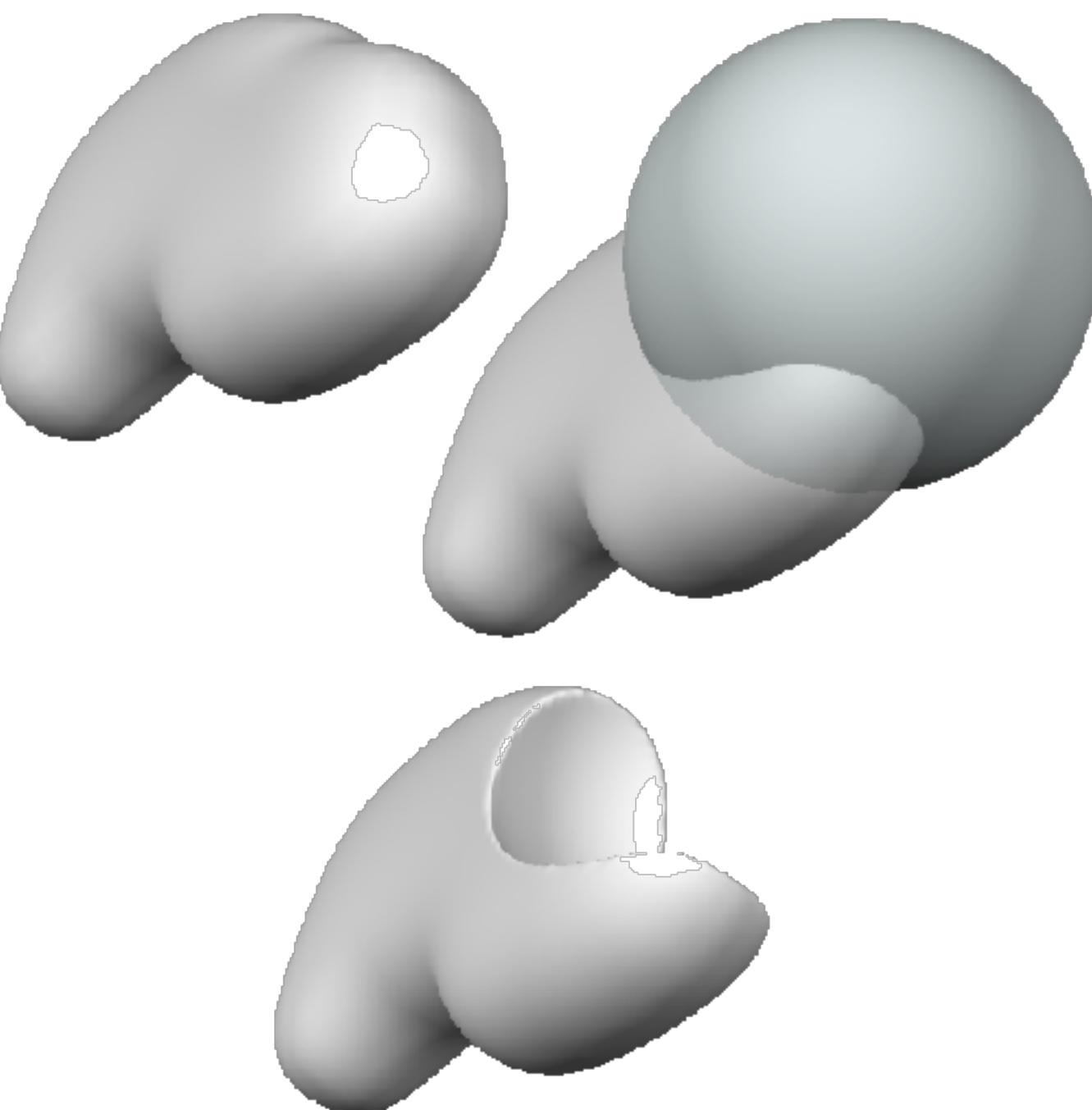
- Intersection:

$$\bigcap_i f_i(x) = \max f_i(x)$$

Boolean Set Operations

- Positive = outside, negative = inside
- Boolean subtraction: $h = \max(f, -g)$

		$f > 0$	$f < 0$
$g > 0$	$h > 0$	$h < 0$	
$g < 0$	$h > 0$	$h > 0$	



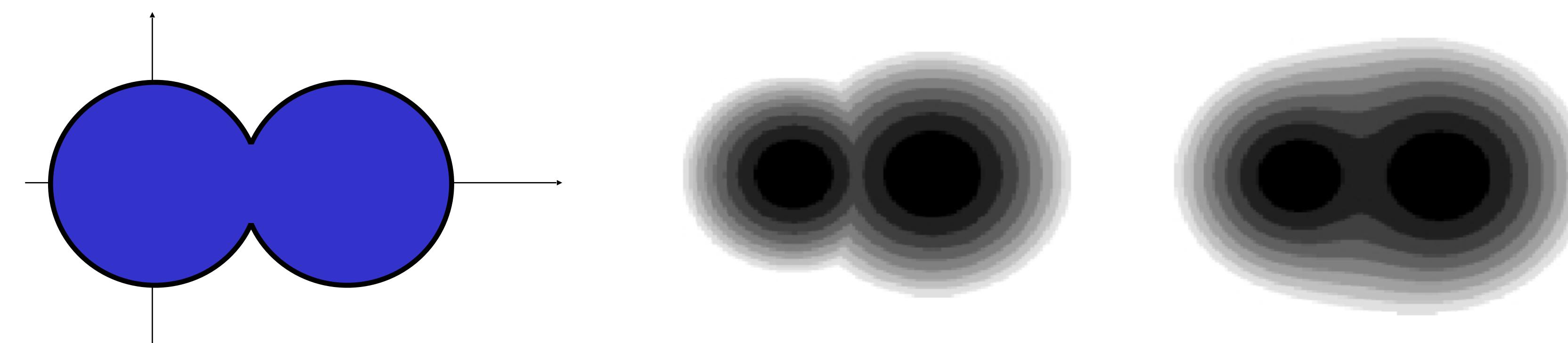
- Much easier than for parametric surfaces!

Smooth Set Operations

- In many cases, smooth blending is desired
 - Pasko and Savchenko, Blending operations for the functionally based constructive geometry [1994]

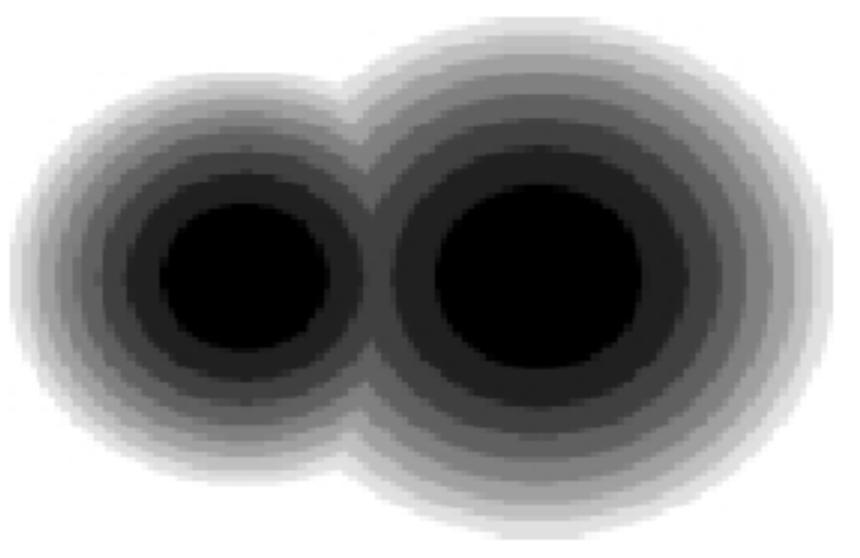
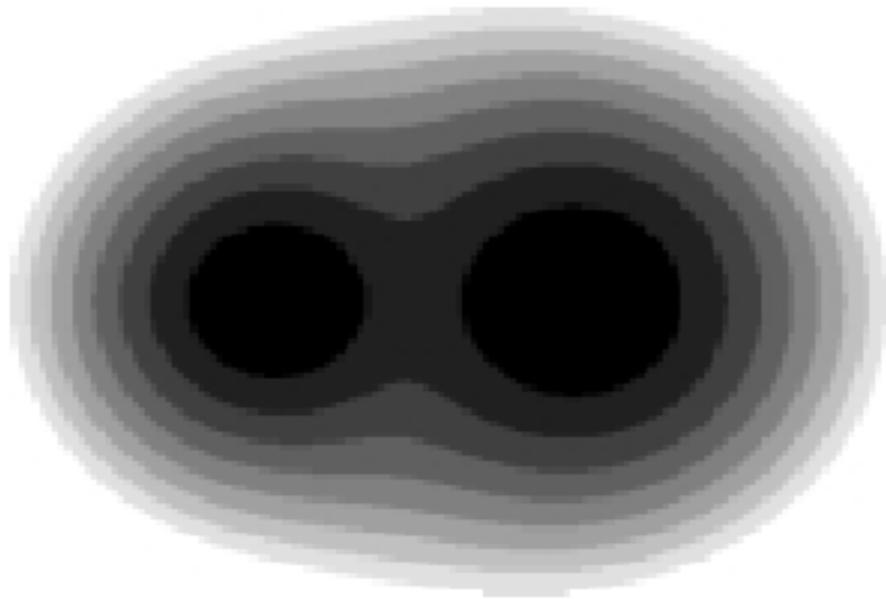
$$f \cup g = \frac{1}{1+\alpha} \left(f + g - \sqrt{f^2 + g^2 - 2\alpha fg} \right)$$

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



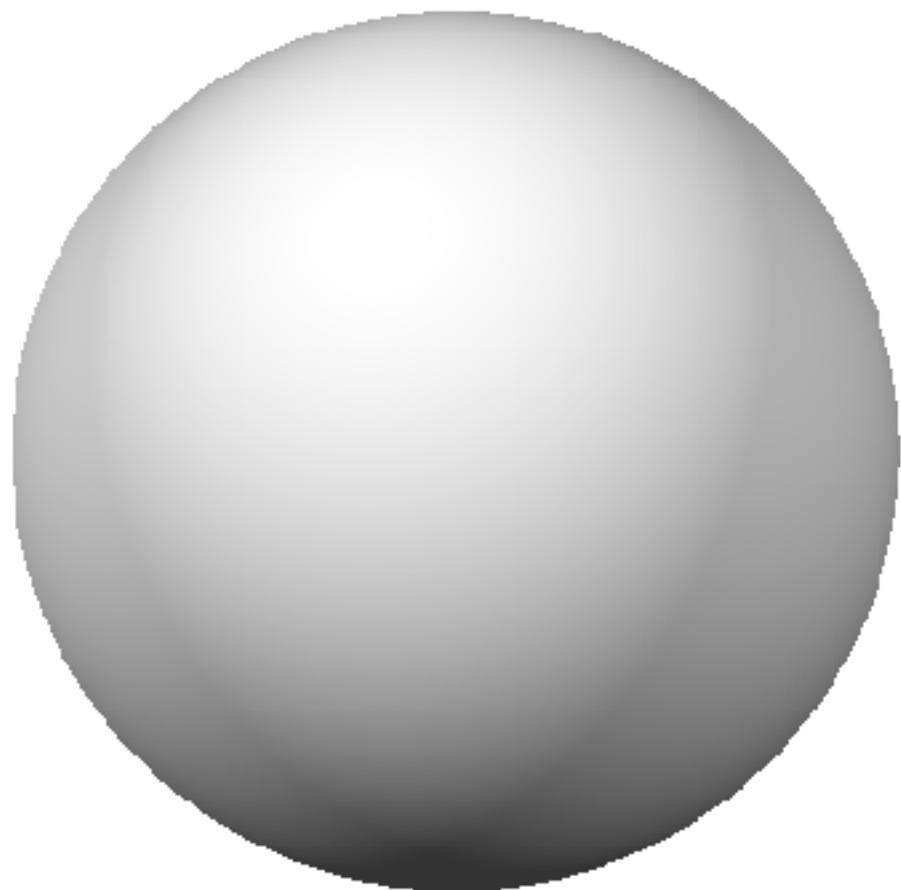
Smooth Set Operations

- Examples



Designing with Implicit Surfaces

- Zero set (or level set) of a function:



$$f(\mathbf{p}) = \|\mathbf{p}\|^2 - r^2$$

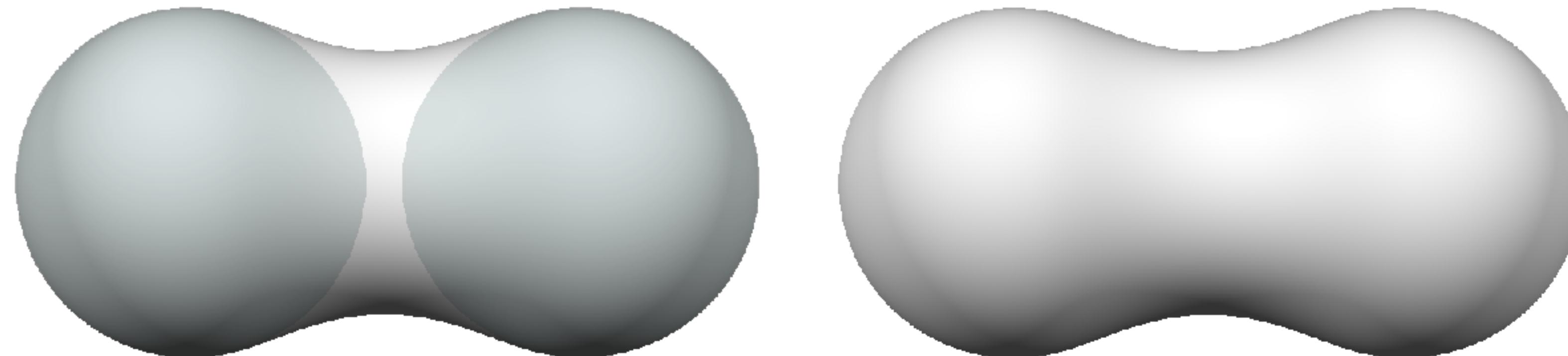
- But also a level set at value e^{-1} of this function:

$$f(\mathbf{p}) = e^{-\|\mathbf{p}\|^2/r^2} \text{ at } e^{-1}$$

Designing with Implicit Surfaces

- With smooth falloff functions, adding implicit functions generates a blend:

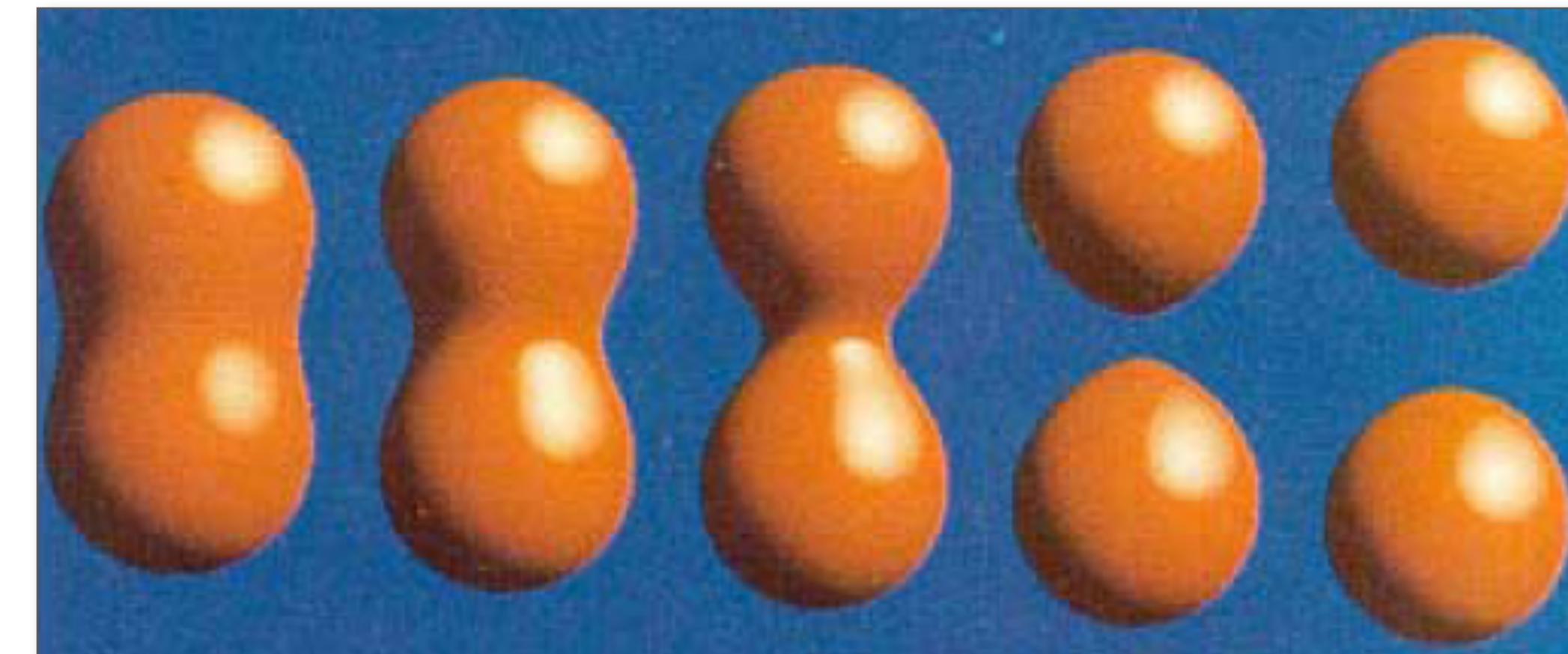
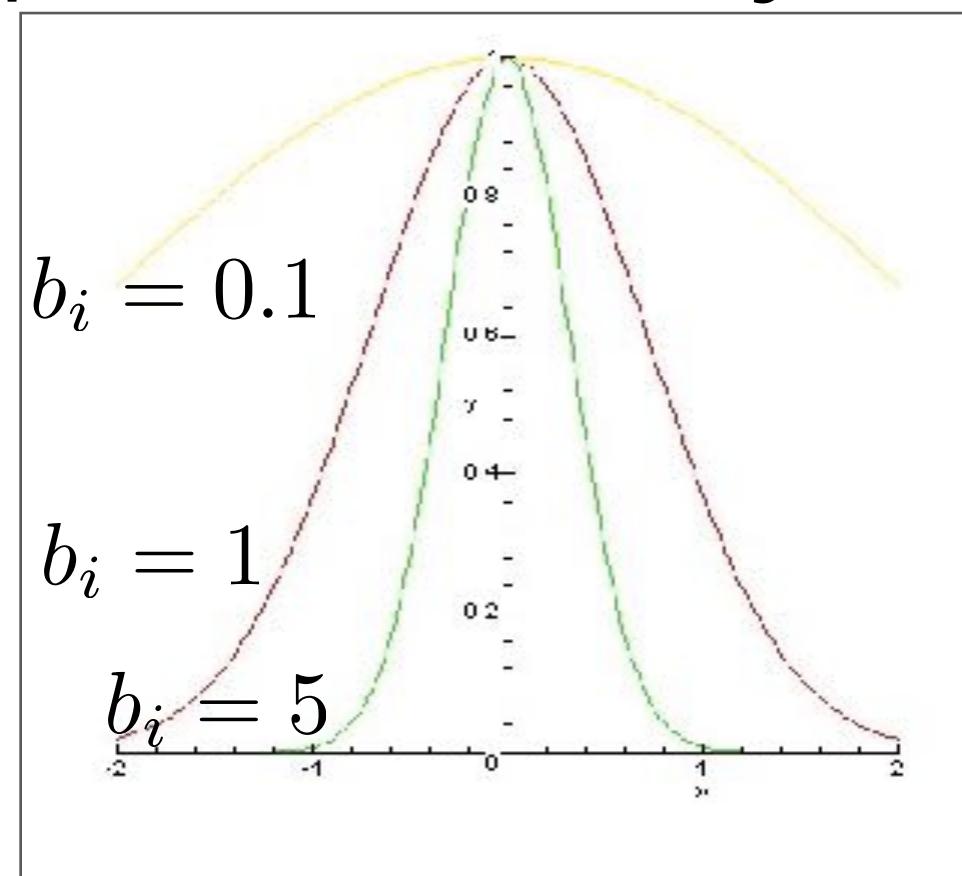
$$f(\mathbf{p}) = e^{-\|\mathbf{p}-\mathbf{p}_1\|^2} + e^{-\|\mathbf{p}-\mathbf{p}_2\|^2}$$



- Called “Metaballs” or “Blobs”

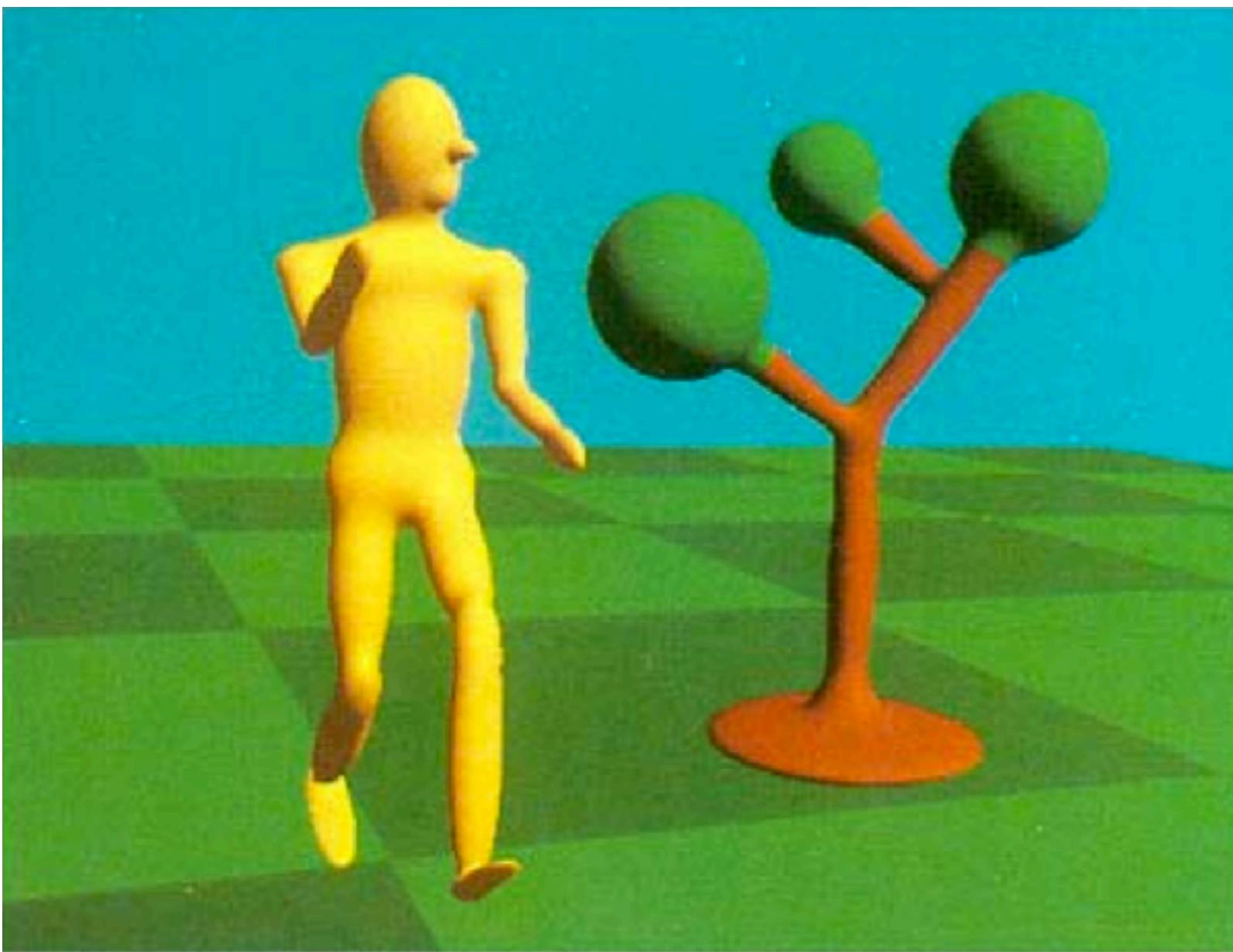
Blobs

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

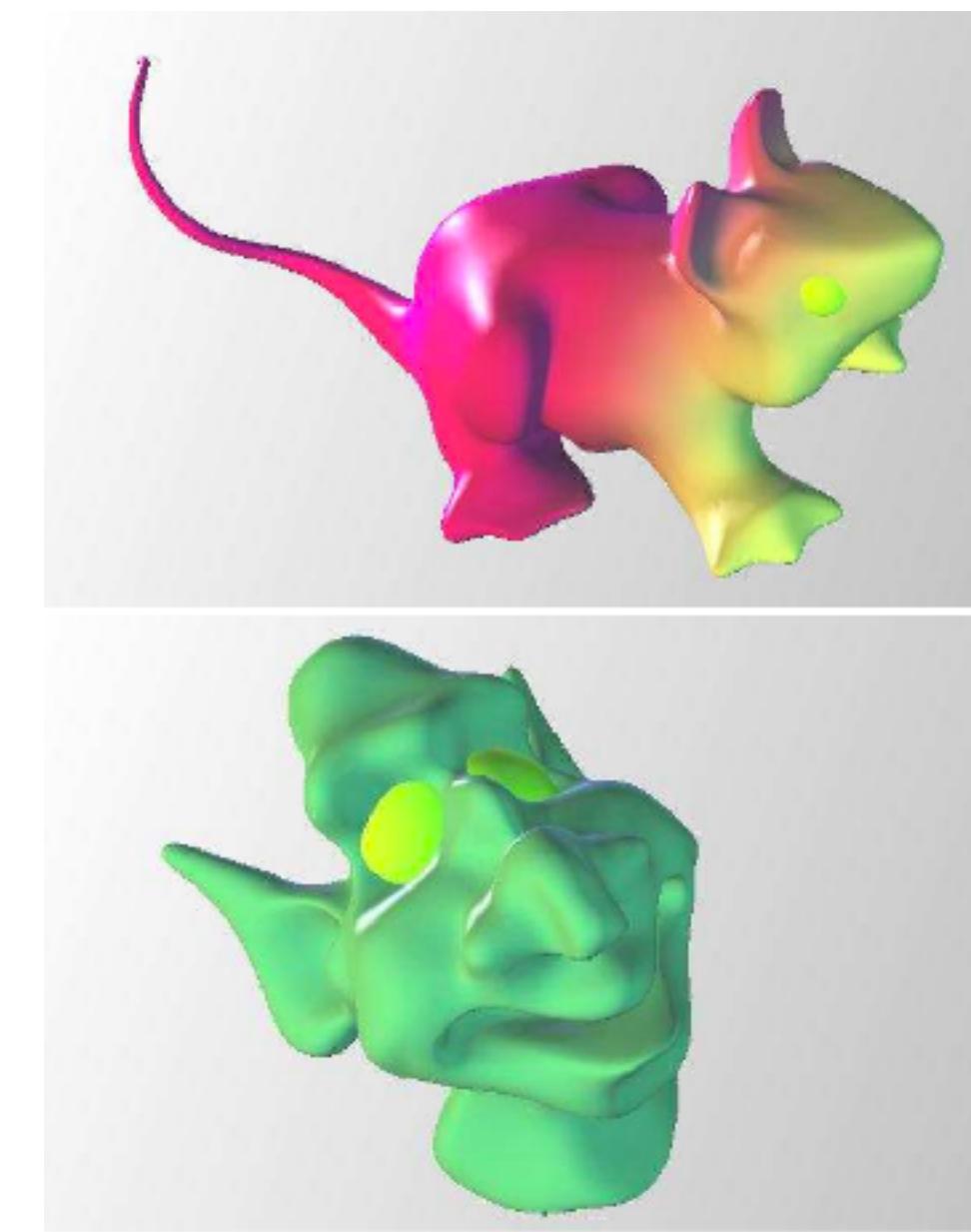


J. Blinn, "A Generalization of Algebraic Surface Drawing", ACM Transactions on Graphics, Vol. 1, No. 3, pp. 235-256, July, 1982.

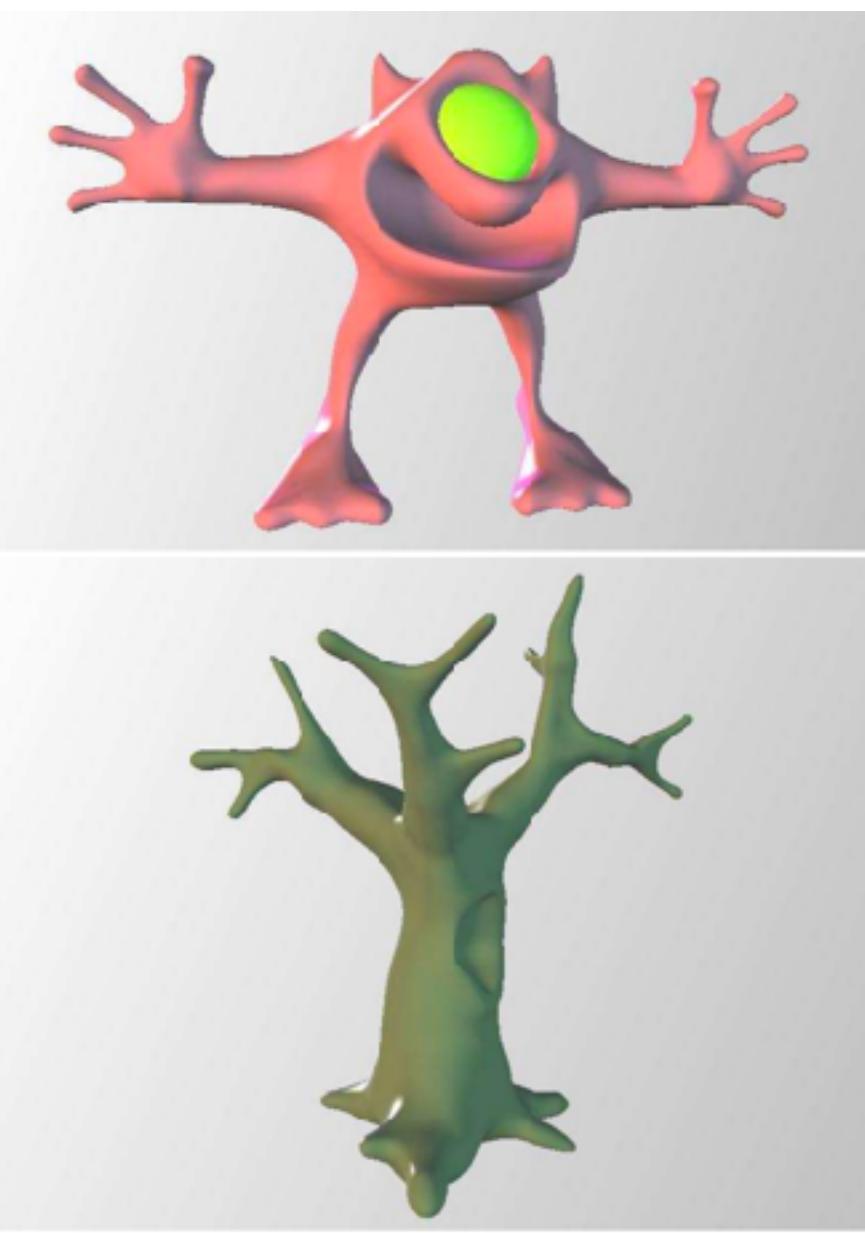
Blobs



J. Blinn, "A Generalization of Algebraic Surface Drawing", ACM Transactions on Graphics, Vol. 1, No. 3, pp. 235-256, July, 1982.



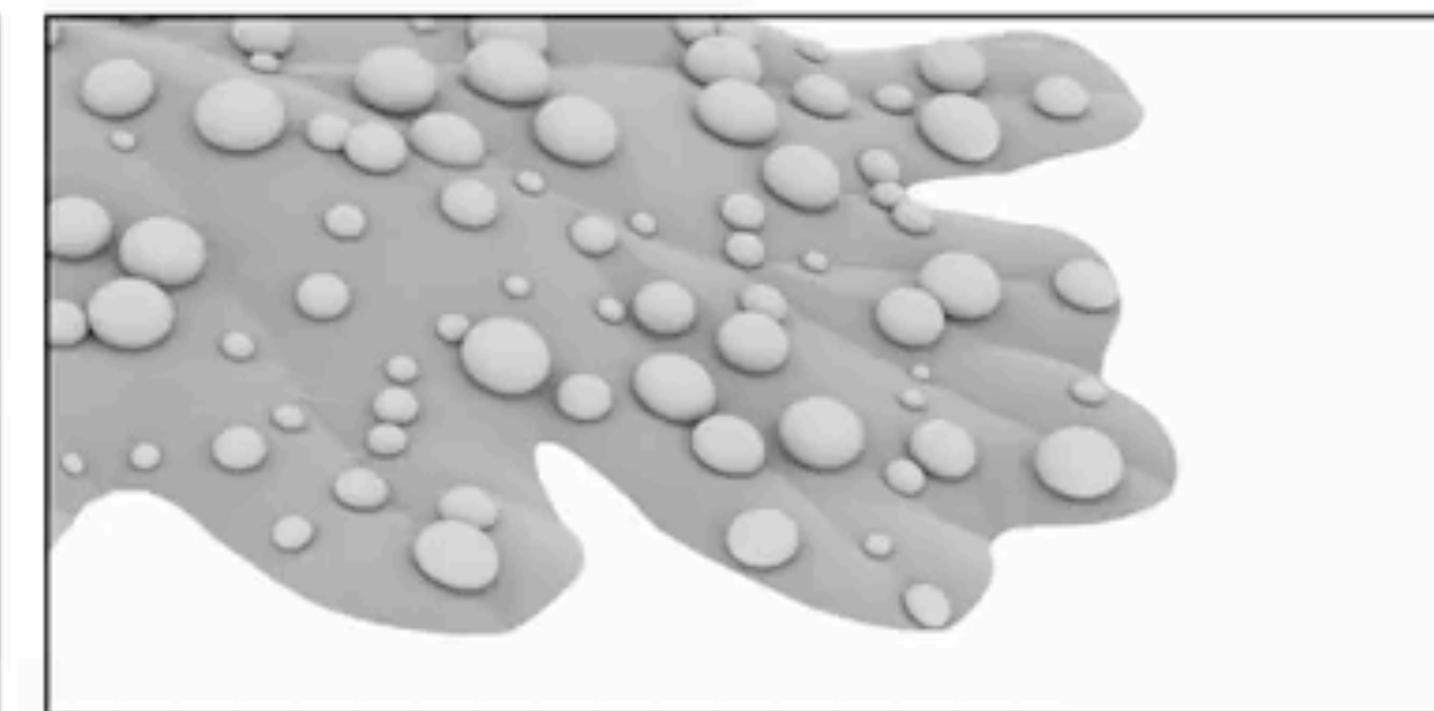
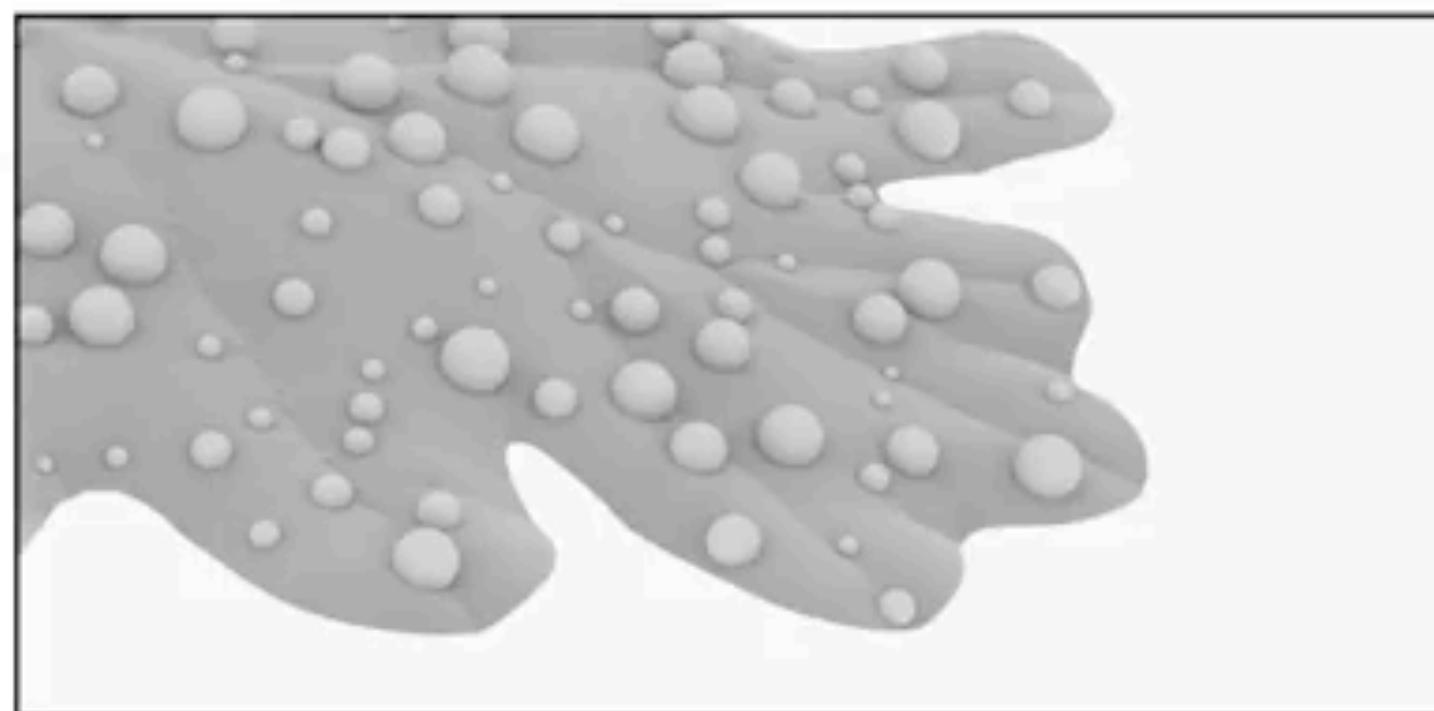
Angelidis et al., "Swirling-Sweepers: Constant-Volume Modeling", Pacific Graphics 2004





Sketch-Based Implicit Blending

Baptiste Angles^{1,2}, Marco Tarini³, Brian Wyvill¹, Loïc Barthe², Andrea Tagliasacchi¹



1. University of Victoria



2. Université de Toulouse, IRIT/CNRS



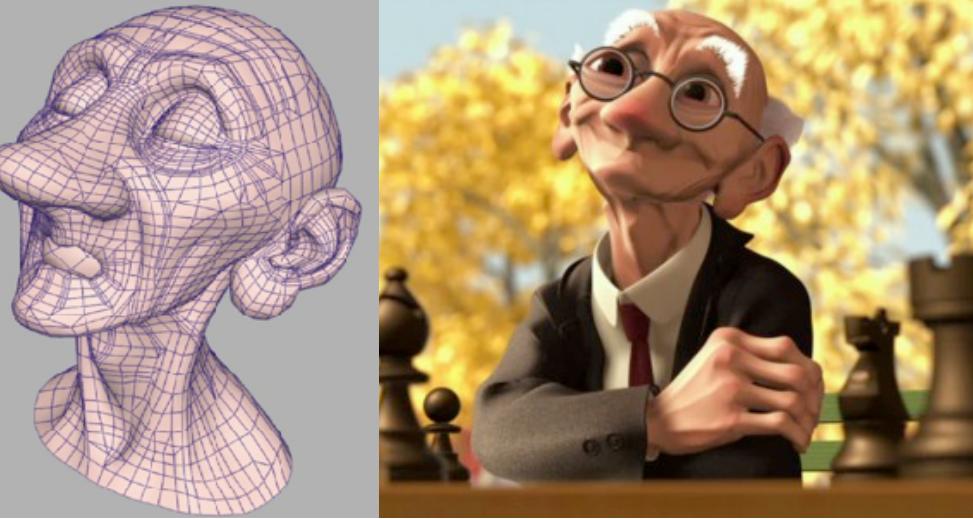
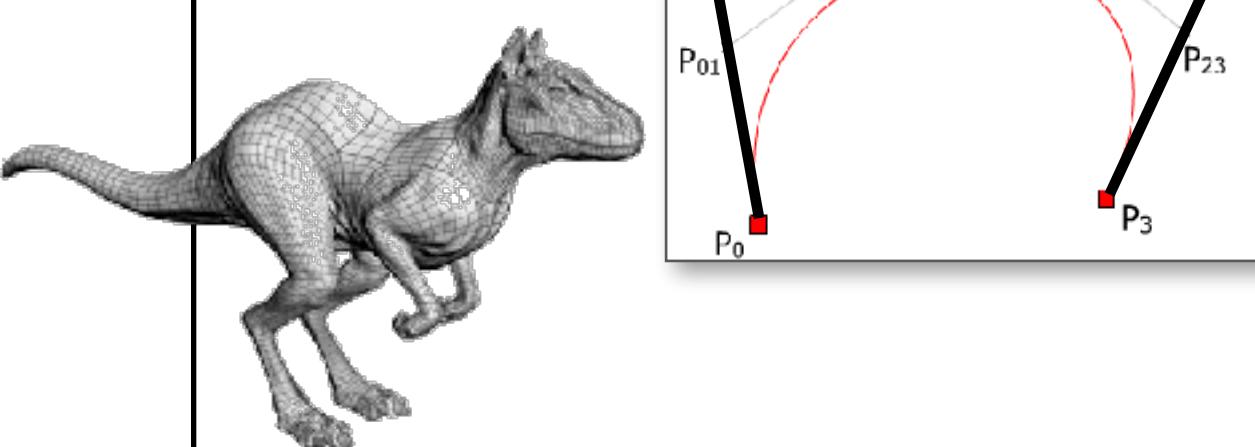
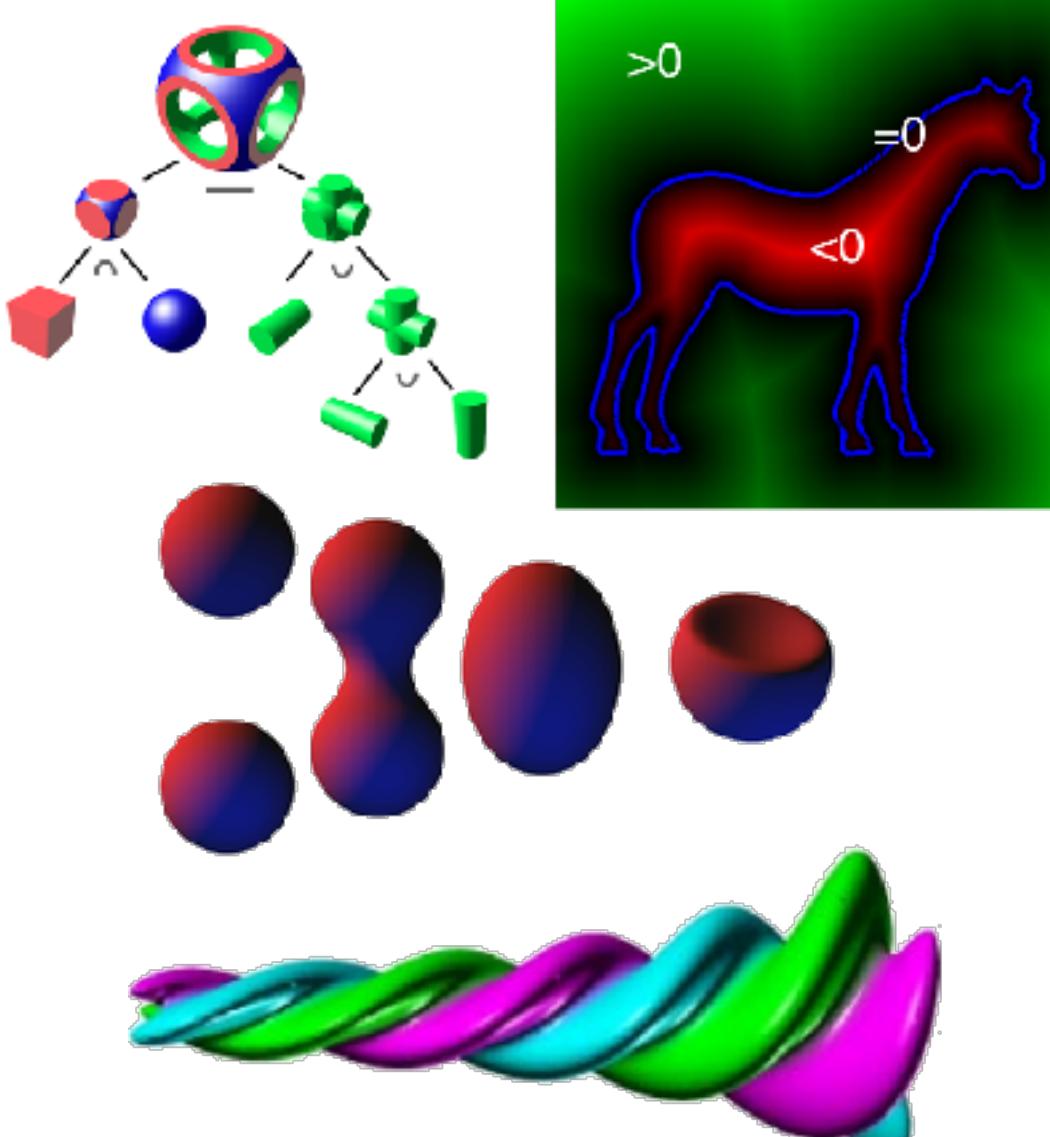
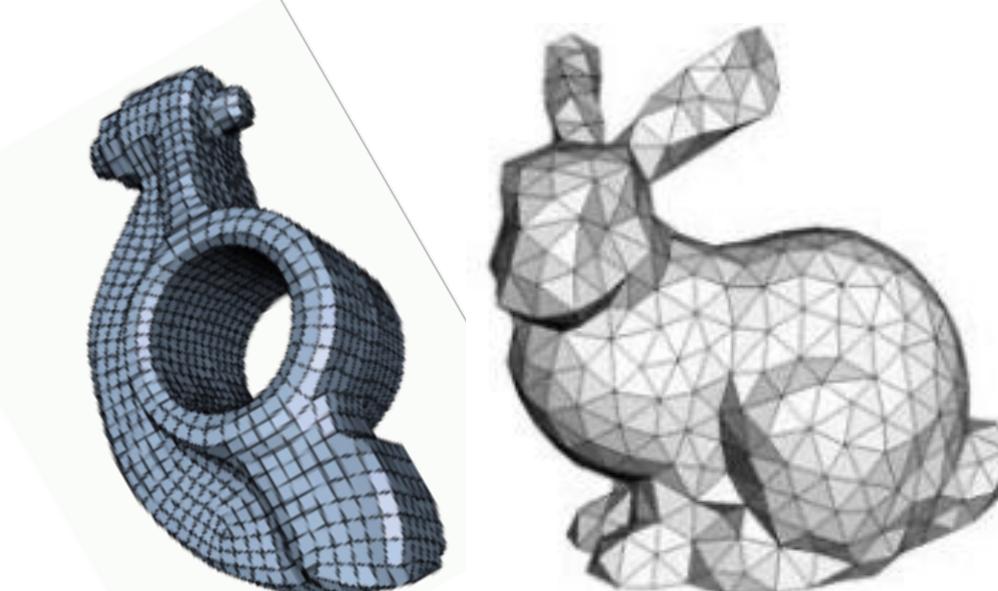
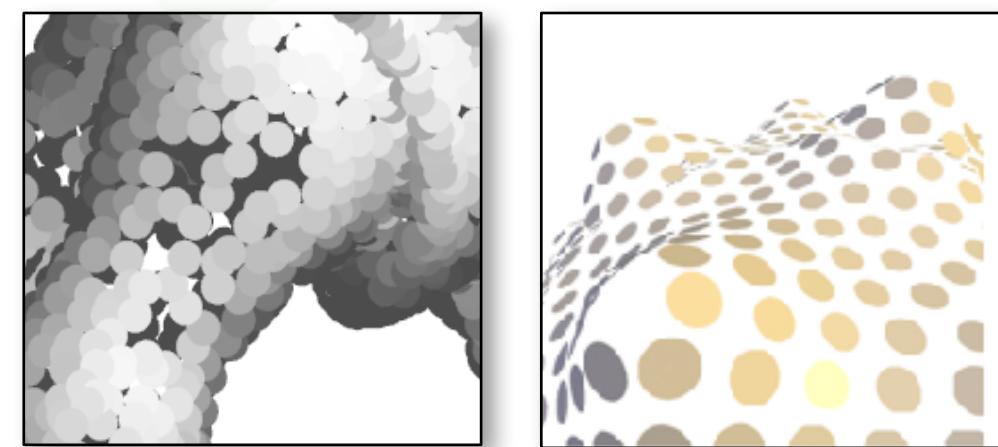
3. Università dell'Insubria, ISTI / CNR



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

Summary

Parametric	Implicit	Discrete/Sampled
  <ul style="list-style-type: none">• Splines, tensor-product surfaces• Subdivision surfaces	 <ul style="list-style-type: none">• Metaballs/blobs• Distance fields	  <ul style="list-style-type: none">• Meshes• Point set surfaces

References

- Polygon Mesh Processing, Chapter 1,2