252-0538-00L, Spring 2025

Shape Modeling and Geometry Processing

Introduction and Overview

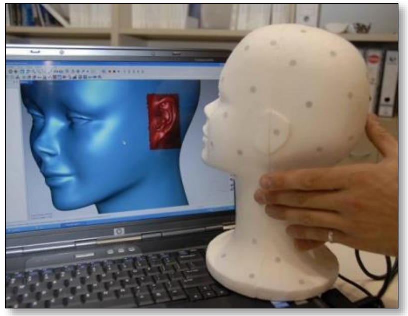
INTERACTIVE GEOMETRY LAB

February 19, 2025

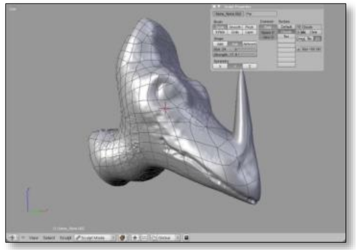
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Shape Representation: Origin- and Application-Dependent

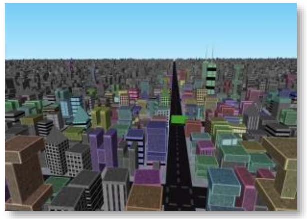
- Acquired real-world objects:
  - Discrete sampling
  - Points, meshes

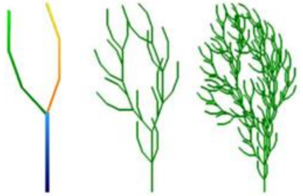



- Modeling “by hand”:
  - Higher-level representations, amendable to modification, control
  - Parametric surfaces, subdivision surfaces, implicits

- Procedural modeling
  - Algorithms, grammars

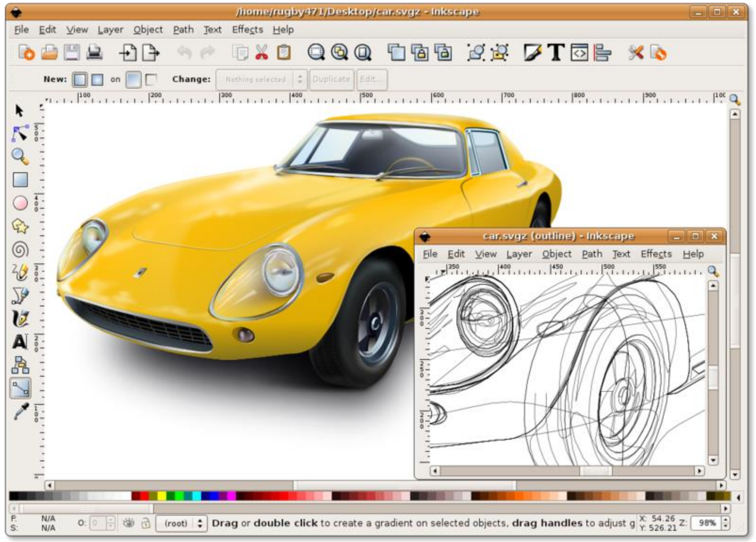


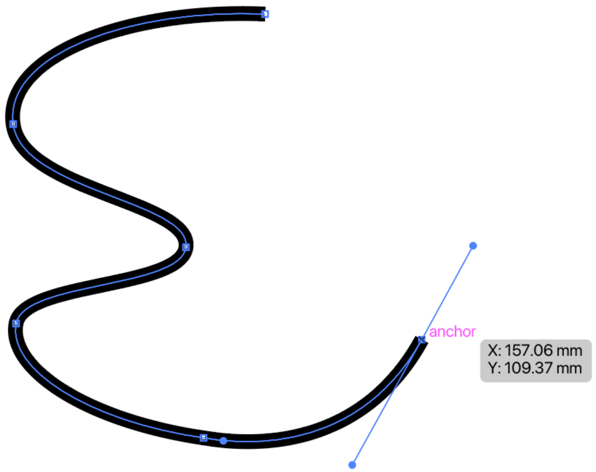


Similar to the 2D Image Domain

- Acquired digital images:
  - Discrete sampling
  - Pixels on a grid

- Painting “by hand”:
  - Strokes + color/shading
  - Vector graphics
  - Controls for editing

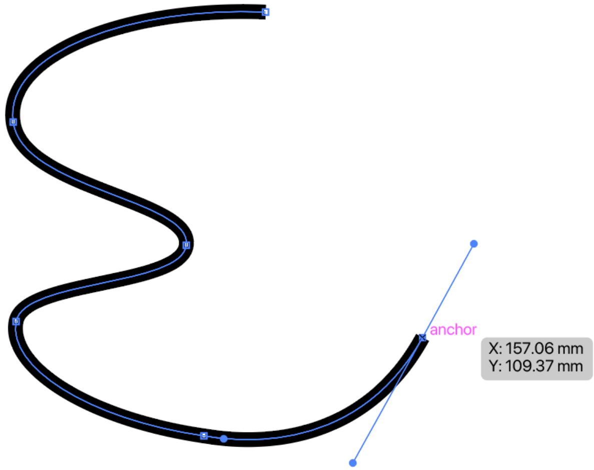


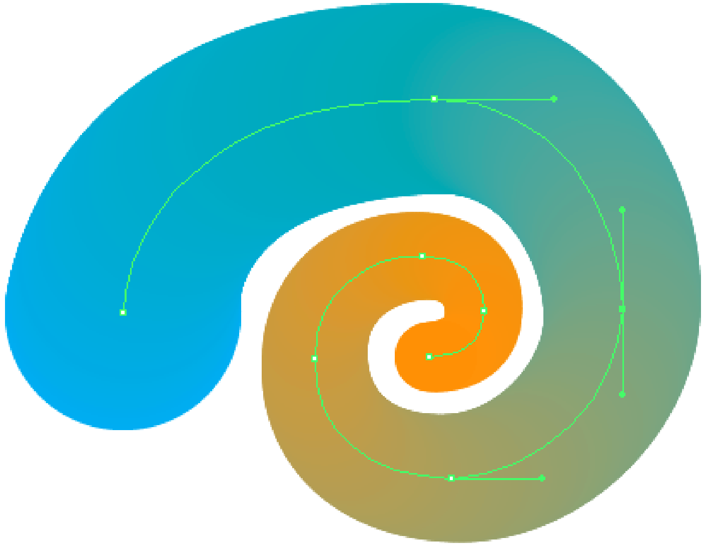


Similar to the 2D Image Domain

- Acquired digital images:
  - Discrete sampling
  - Pixels on a grid

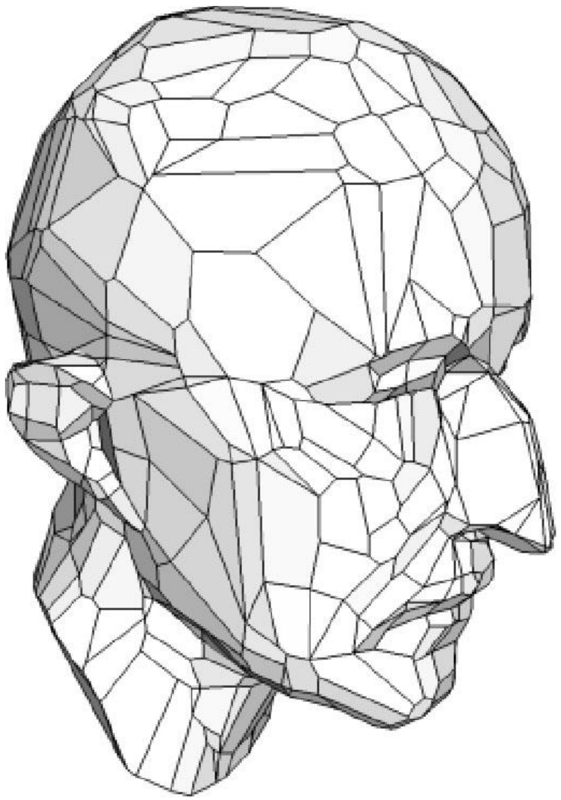
- Painting “by hand”:
  - Strokes + color/shading
  - Vector graphics
  - Controls for editing





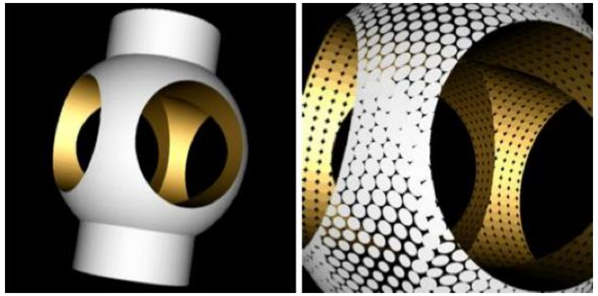
Representation Considerations

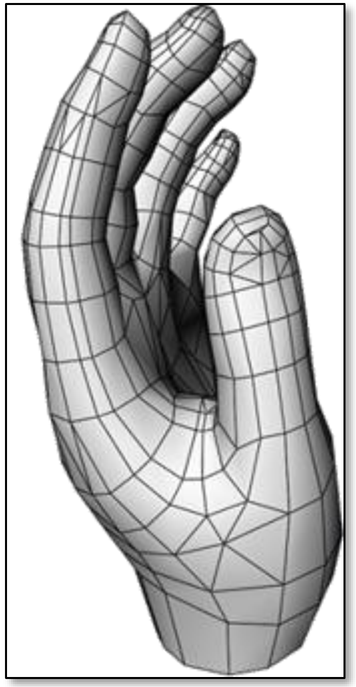
- How should we represent geometry?
  - Needs to be stored in the computer
  - Creation of new shapes
    - Input metaphors, interfaces, generative AI …
  - What operations do we apply?
    - Editing, simplification, smoothing, filtering, repair…
  - How to render it?
    - Rasterization, raytracing…

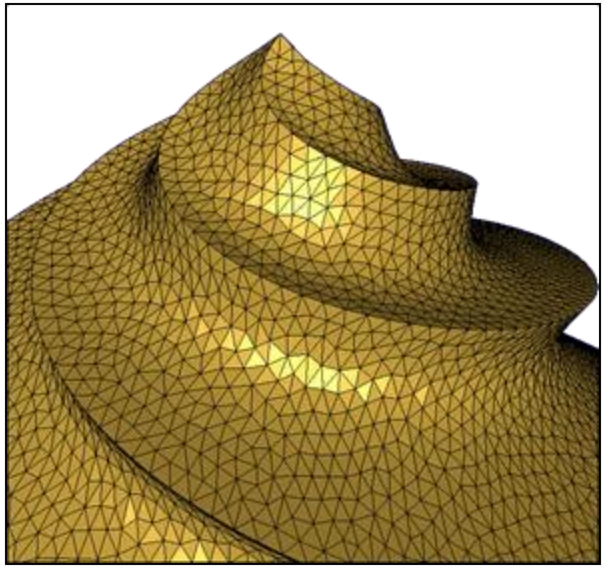


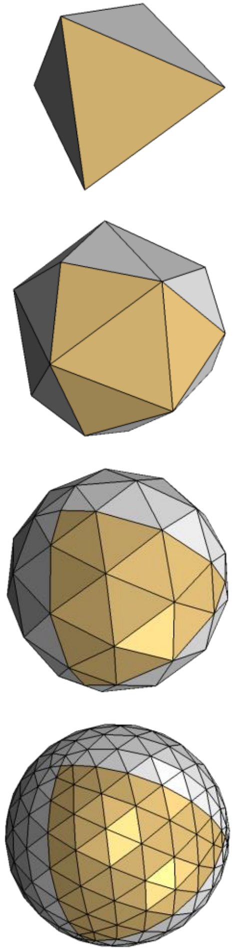
Shape Representations

- Points
- Polygonal meshes



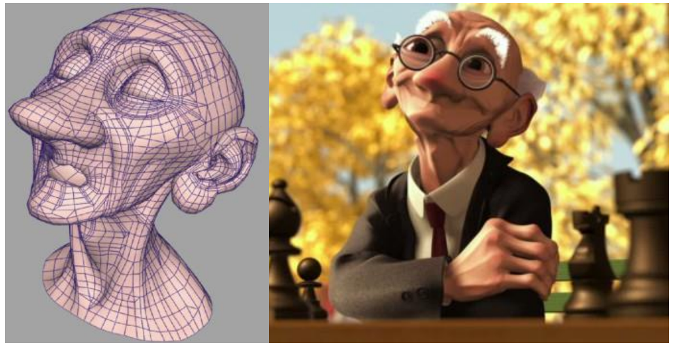


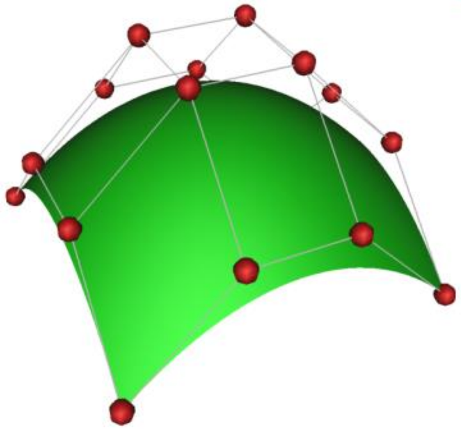


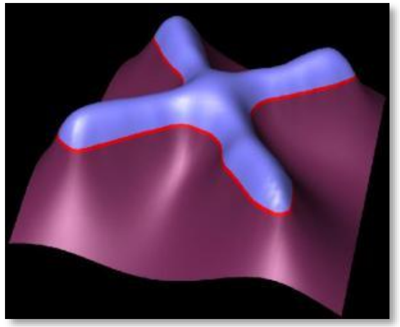


Shape Representations

- Parametric surfaces
- Implicit functions
- Subdivision surfaces







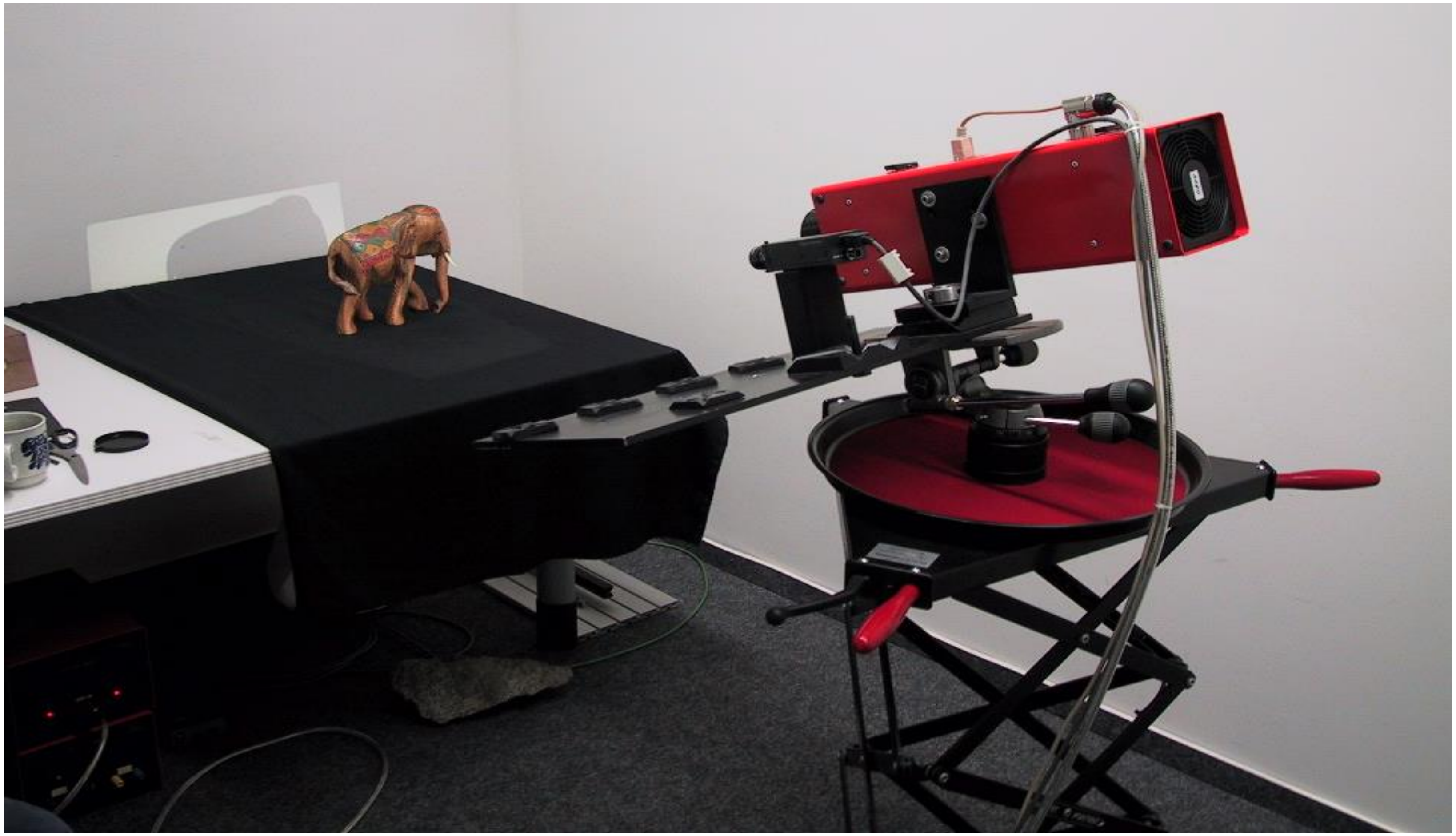
Points

INTERACTIVE GEOMETRY LAB

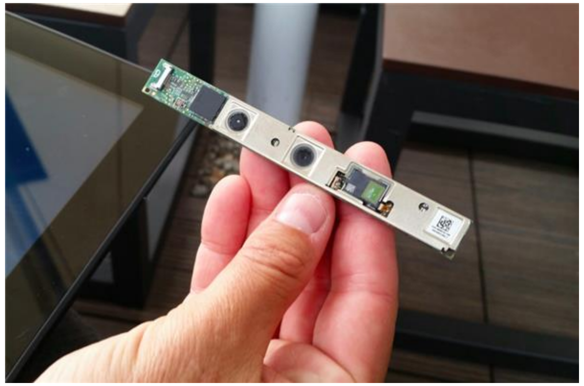
February 19, 2025

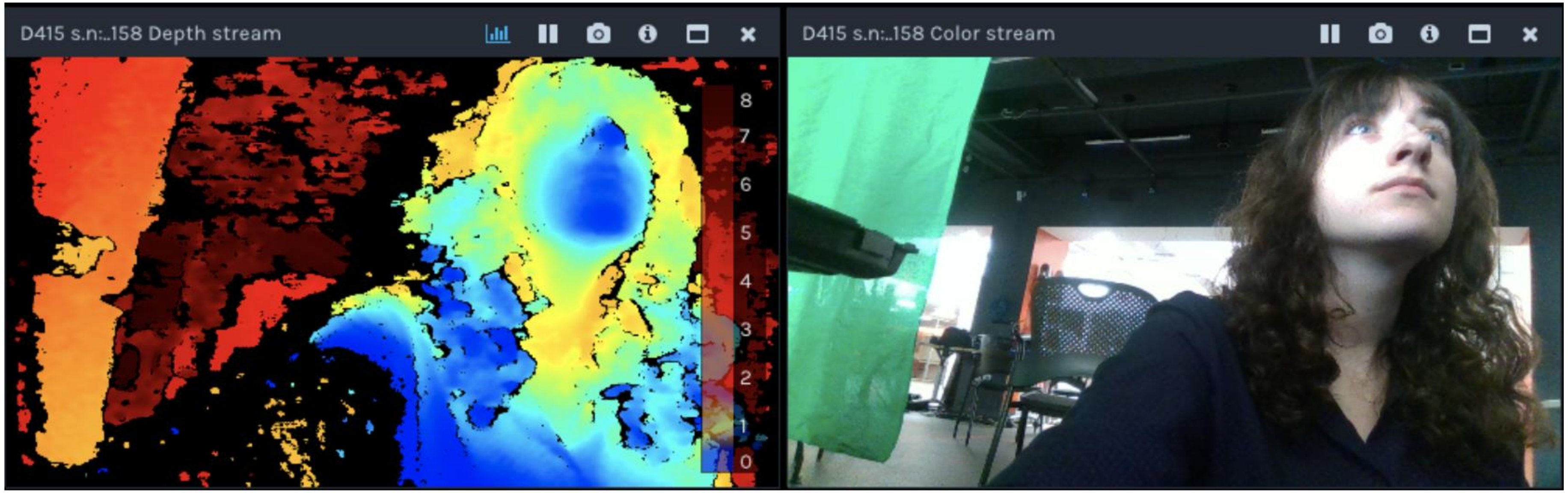
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Output of Acquisition



Output of Acquisition

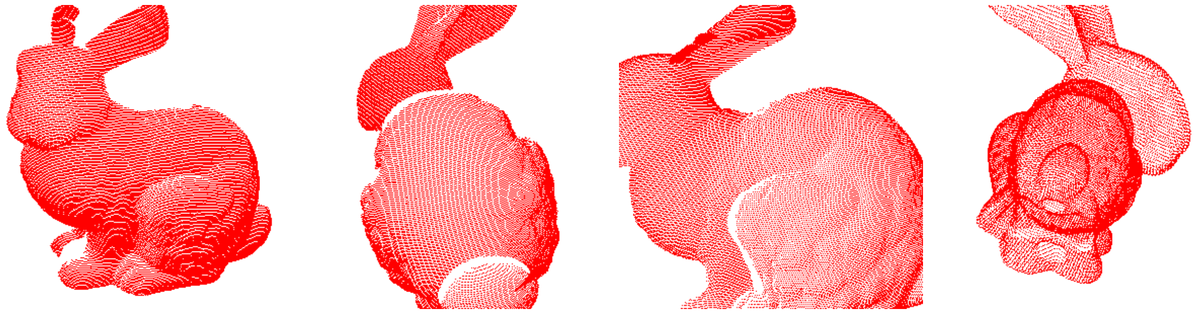


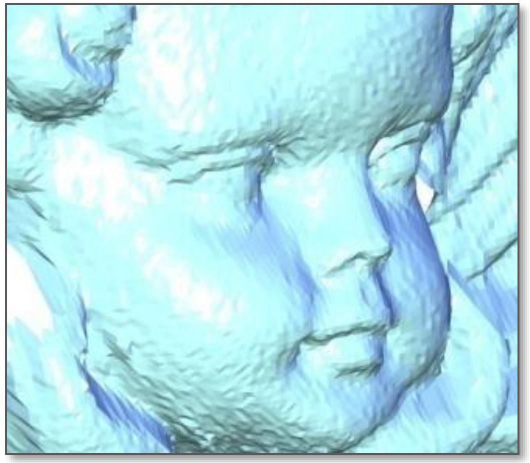


https://frl.nyu.edu/volumetric-capture-with-realsense-depth-cameras/

Points

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

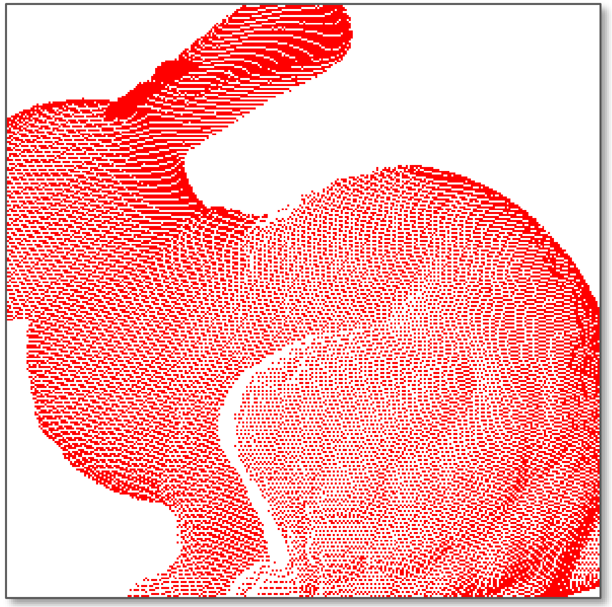




  - Depth imaging (e.g. by triangulation)
  - Registration of multiple images

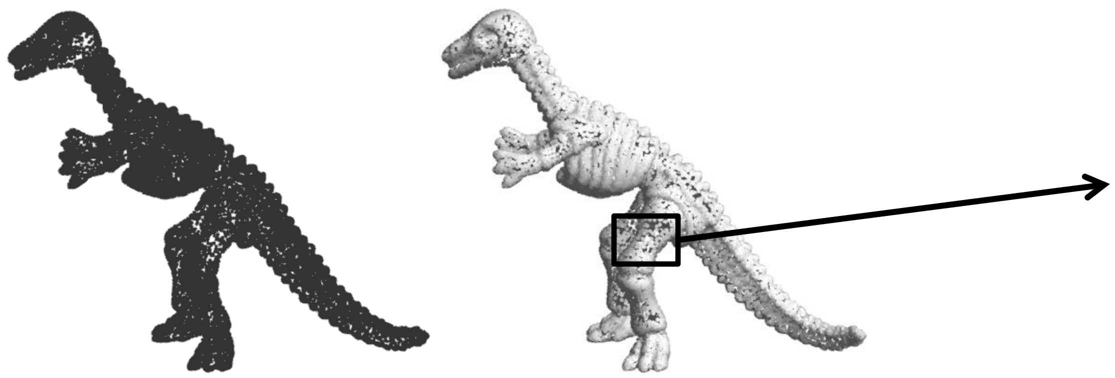
Points

- points = unordered set of 3-tuples
- Often converted to other reps
  - Meshes, implicits, parametric surfaces
  - Those are easier to process, edit and/or render
- Spatial partitioning data structure required
  - For efficient point processing
  - To figure out neighborhoods

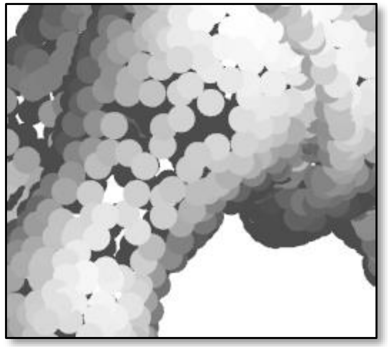


Points: Neighborhood information

- Why do we need neighbors?



need normals (for shading)

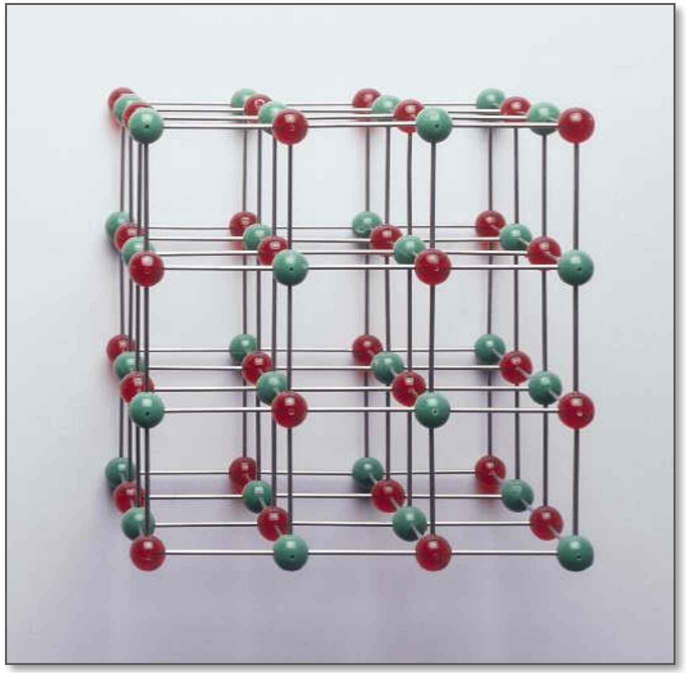


upsampling – need to count density

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

Spatial Data Structures

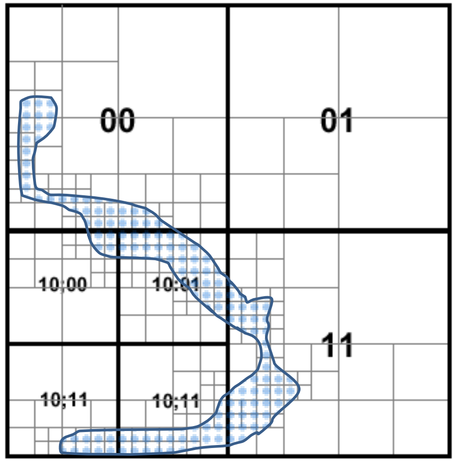
- Regular uniform 3D lattice
  - Simple point insertion by coordinate discretization
  - Simple proximity queries by searching neighboring cells

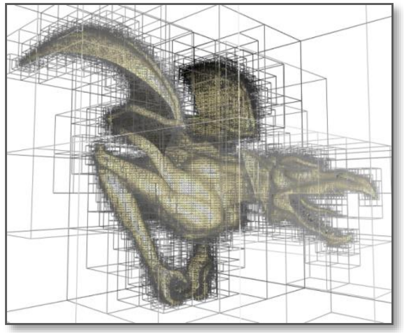


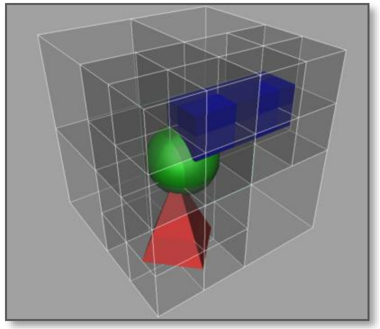
  - Determining lattice parameters (i.e. cell dimensions) is nontrivial
  - Generally unbalanced, i.e. many empty cells

Spatial Data Structures

- 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 be unbalanced

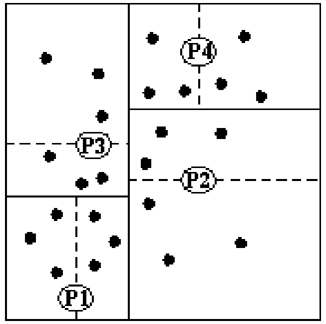


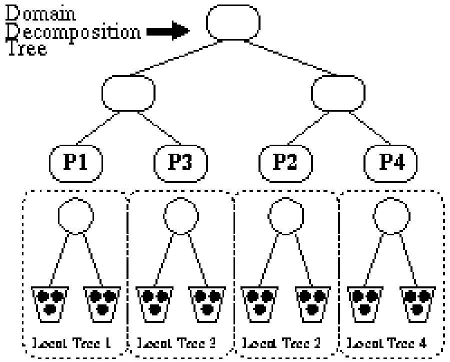


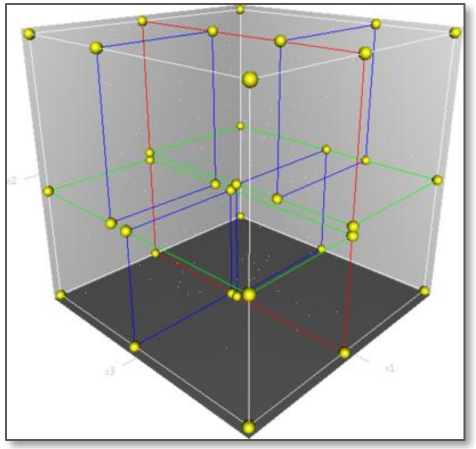


Spatial Data Structures

- 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

INTERACTIVE GEOMETRY LAB

February 19, 2025

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Parametric Representation

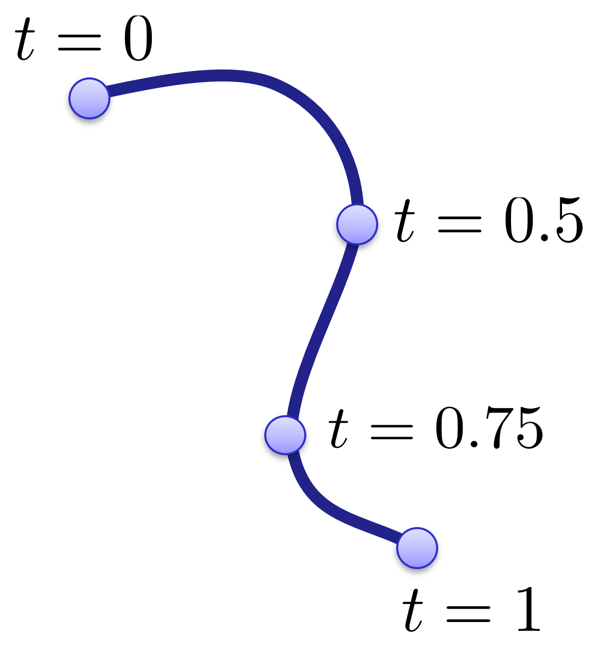
- **Range** of a function $f : X \to Y, \quad X \subseteq \mathbb{R}^m, \; Y \subseteq \mathbb{R}^n$

  - Planar curve: $m = 1, n = 2$

    $$f(t) = (x(t), y(t))$$

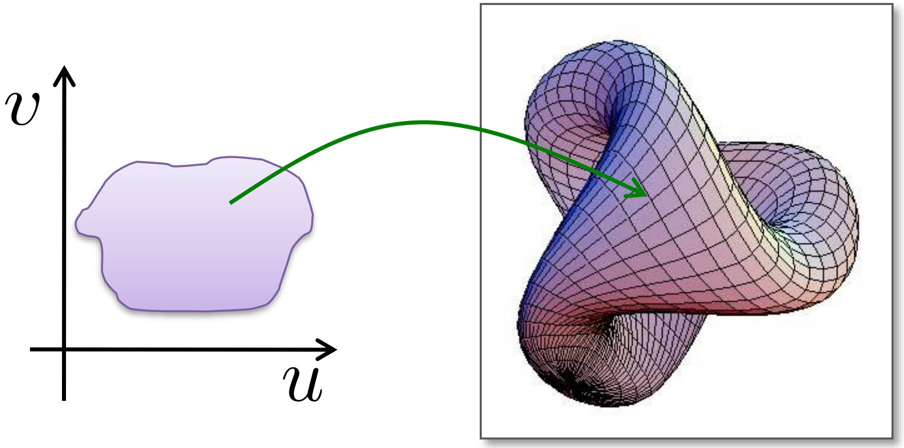
  - Space curve: $m = 1, n = 3$

    $$f(t) = (x(t), y(t), z(t))$$



Parametric Representation

- **Range** of a function $f : X \to Y, \quad X \subseteq \mathbb{R}^m, \ Y \subseteq \mathbb{R}^n$
  - Surface in 3D: $m = 2, n = 3$



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

Parametric Curves

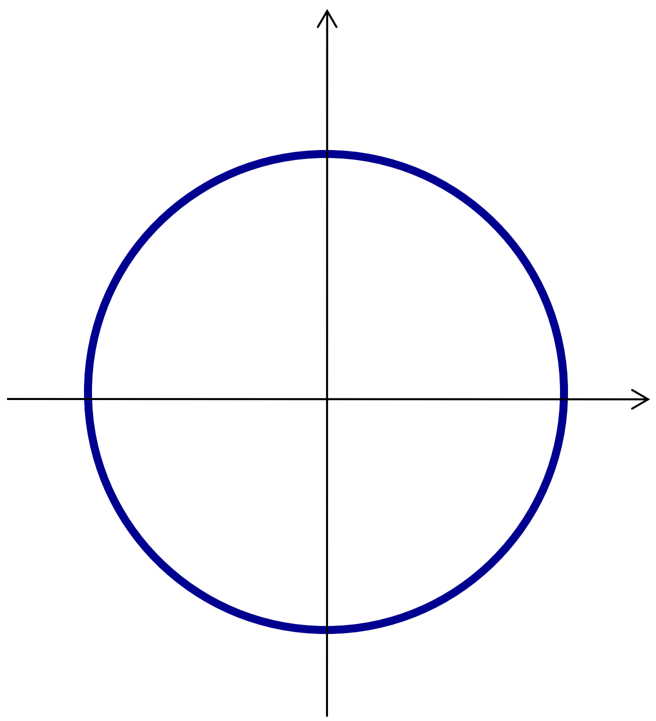
- Parametric circle in 2D

$$\mathbf{p}: \mathbb{R} \to \mathbb{R}^2$$

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

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

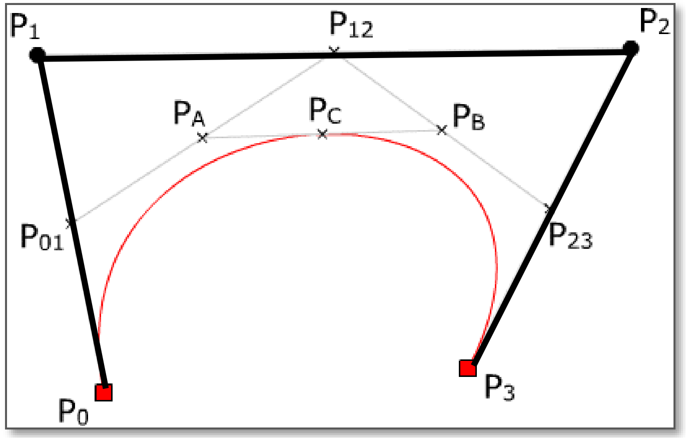
$$t \in [0, 2\pi)$$



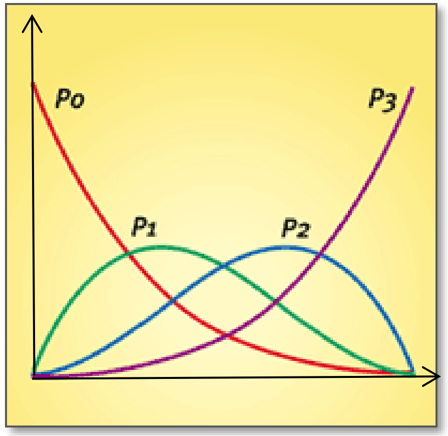
Parametric Curves

- Bézier curves, splines

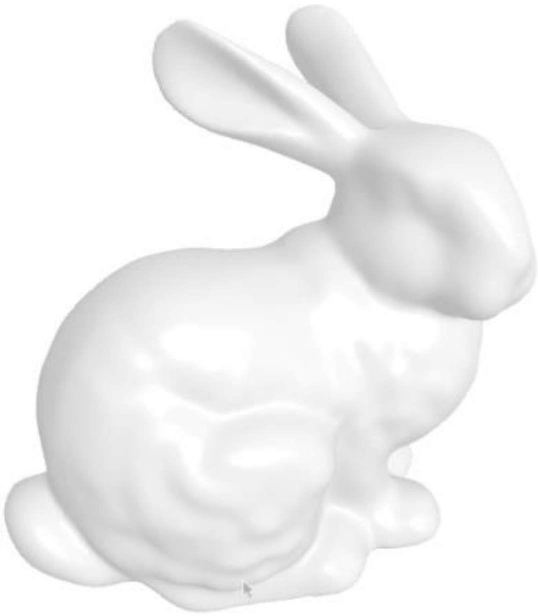
$$\mathbf{p}(t) = \sum_{i=0}^{n} \mathbf{p}_i B_i^n(t) \qquad B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$



Curve and control polygon



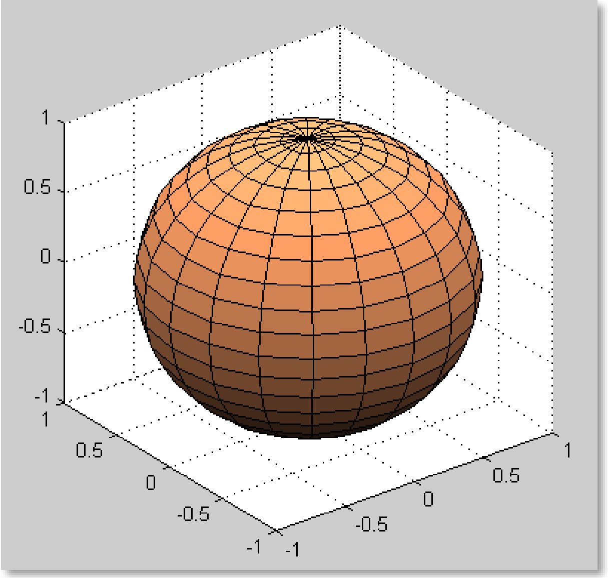
Basis functions



Parametric Surfaces

- Sphere in 3D

$$s : \mathbb{R}^2 \to \mathbb{R}^3$$



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

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

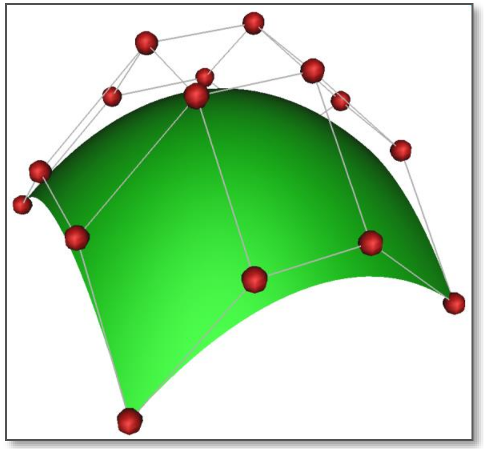
Parametric Surfaces

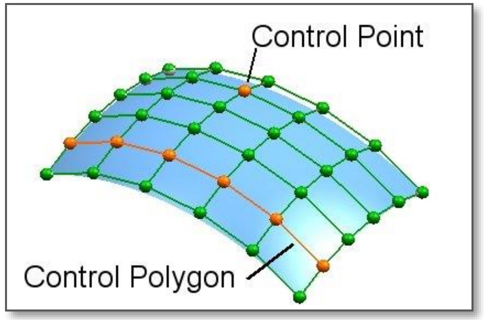
- Curve swept by another curve

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

- Bézier surface:

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



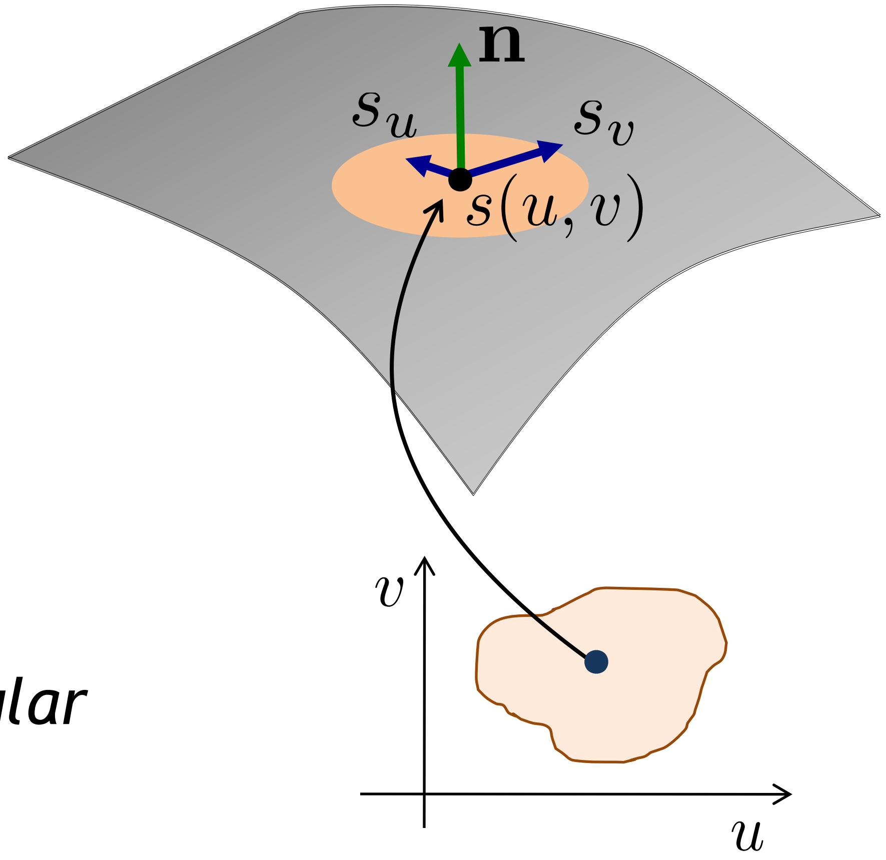


Tangents and Normal

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

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

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



$\mathbf{n}$ exists if parameterization is *regular*

Tangent plane is normal to $\mathbf{n}$

Parametric Curves and Surfaces

- Advantages
  - Easy to generate points on the curve/surface
  - Easy to compute tangents, normal, etc.

- Disadvantages
  - Hard to determine inside/outside
  - Hard to determine if a point is **on** the curve/surface

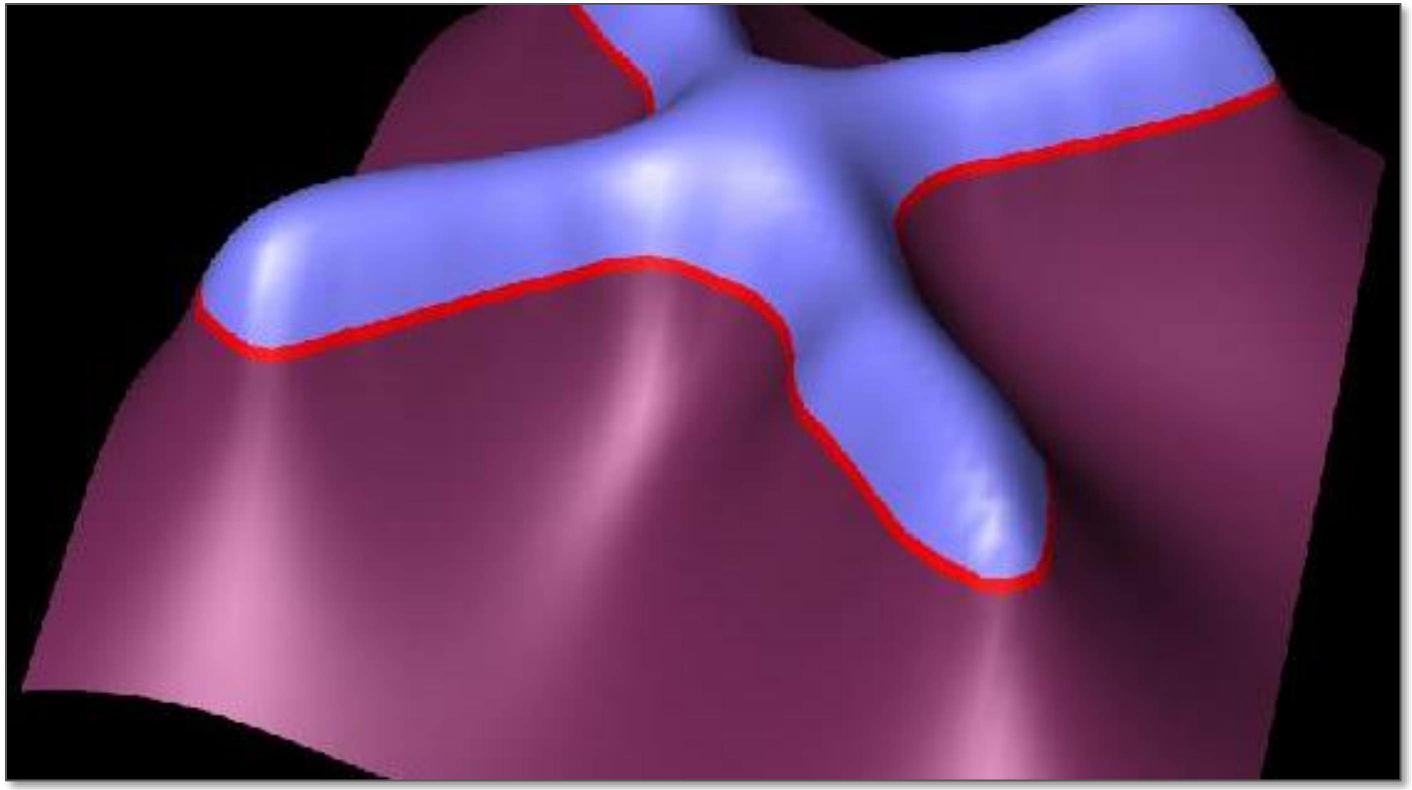
Implicit Curves and Surfaces

igl
INTERACTIVE GEOMETRY LAB

February 19, 2025

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Implicit Curves and Surfaces



Implicit Curves and Surfaces

(usually the zero level set)

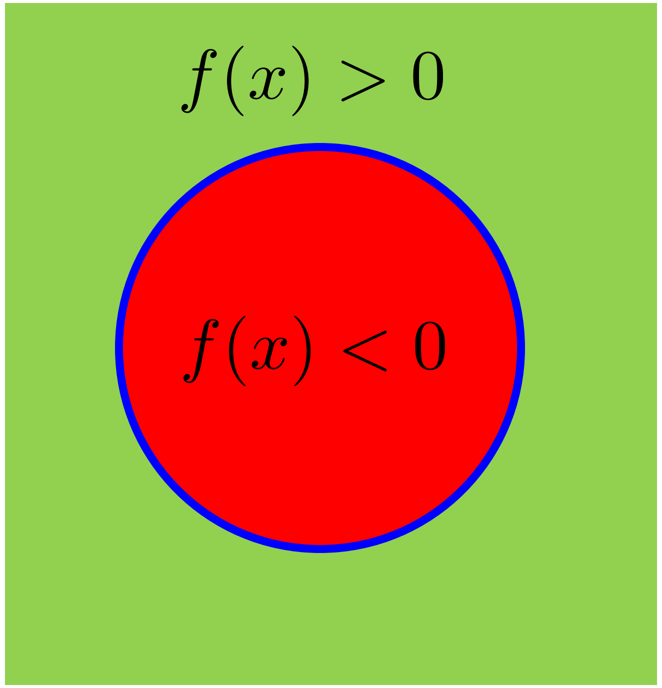
- **Level set of a scalar function** $f : \mathbb{R}^m \to \mathbb{R}$
  - Curve in 2D: $S = \{x \in \mathbb{R}^2 | f(x) = 0\}$
  - Surface in 3D: $S = \{x \in \mathbb{R}^3 | f(x) = 0\}$

- 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

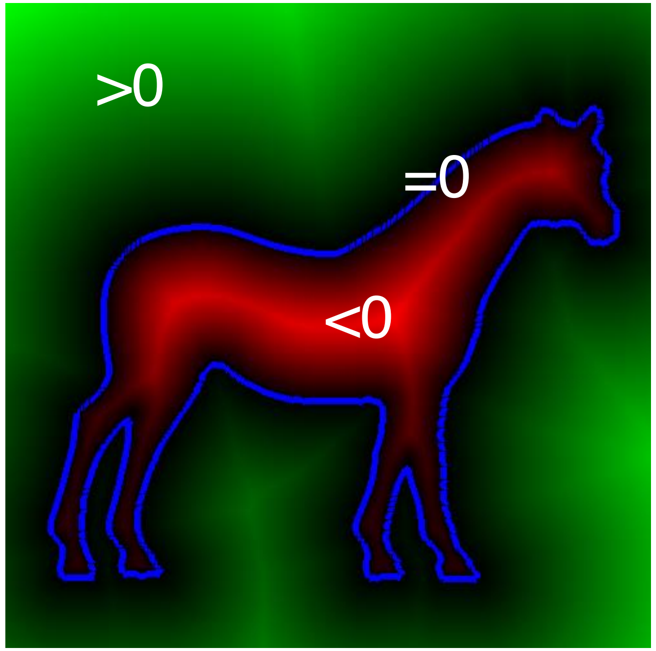


Implicit Curves and Surfaces

(usually the zero level set)

- **Level set** of a scalar function $f : \mathbb{R}^m \to \mathbb{R}$
  - Curve in 2D: $S = \{x \in \mathbb{R}^2 | f(x) = 0\}$
  - Surface in 3D: $S = \{x \in \mathbb{R}^3 | f(x) = 0\}$

- Popular choice: zero level set of the **signed distance function**

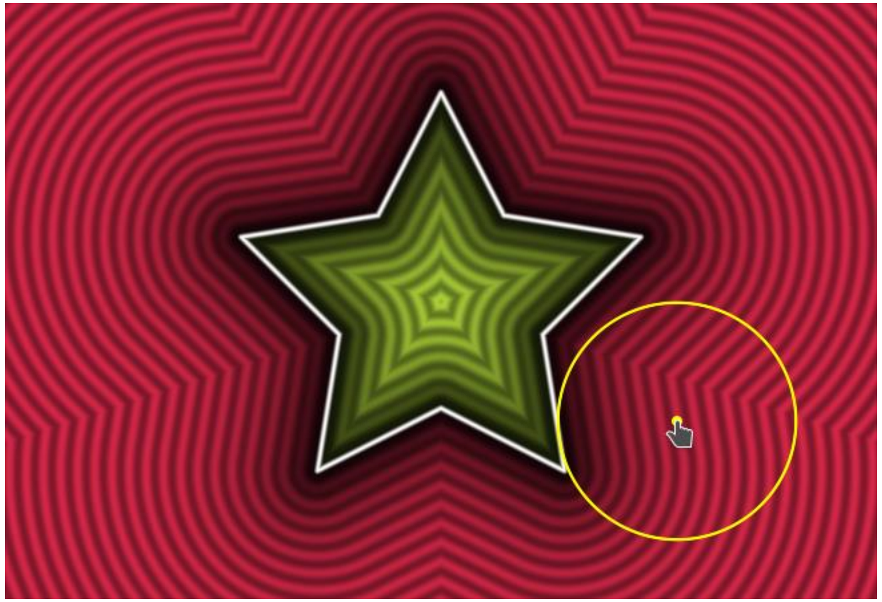


Implicit Curves and Surfaces

(usually the zero level set)

- **Level set** of a scalar function $f : \mathbb{R}^m \to \mathbb{R}$
  - Curve in 2D: $S = \{x \in \mathbb{R}^2 | f(x) = 0\}$
  - Surface in 3D: $S = \{x \in \mathbb{R}^3 | f(x) = 0\}$

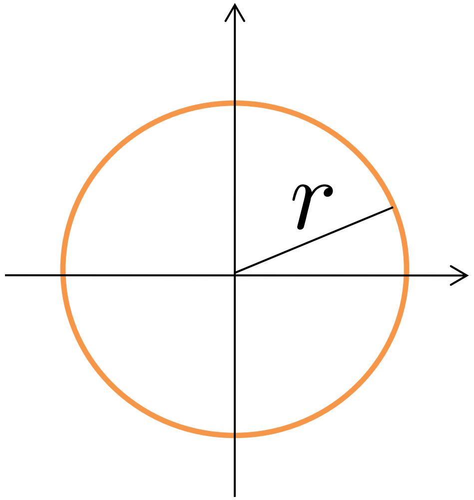
- Popular choice: zero level set of the **signed distance function**



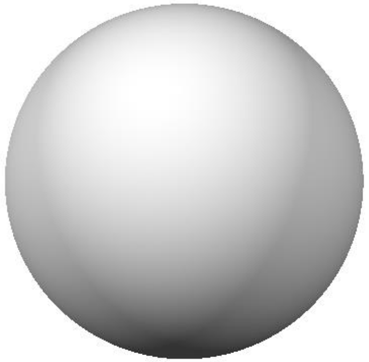
Implicit Curves and Surfaces

- Implicit circle and sphere

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



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



… can convert to true signed distance using sqrt

Implicit Curves and Surfaces

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

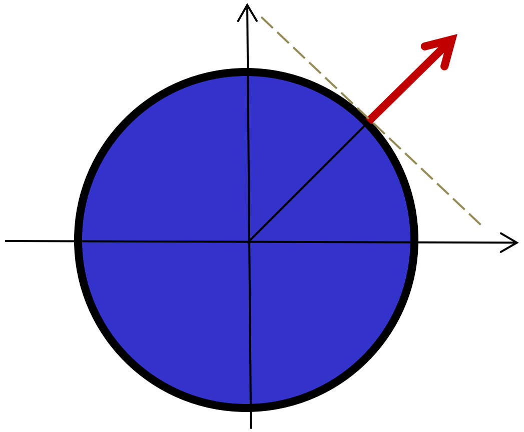
$$\nabla f(x,y,z) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)^{\mathsf{T}}$$

Why?

- Example

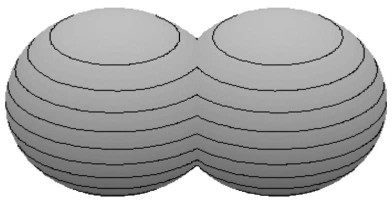
$$f(x,y,z) = x^2 + y^2 + z^2 - r^2$$

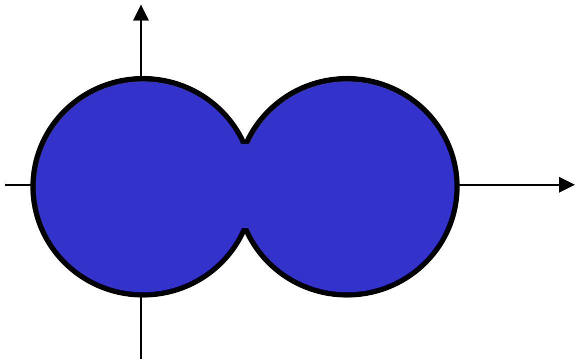
$$\nabla f(x,y,z) = (2x, 2y, 2z)^{\mathsf{T}}$$



Boolean Set Operations

- Union: $$\bigcup_i f_i(x) = \min f_i(x)$$



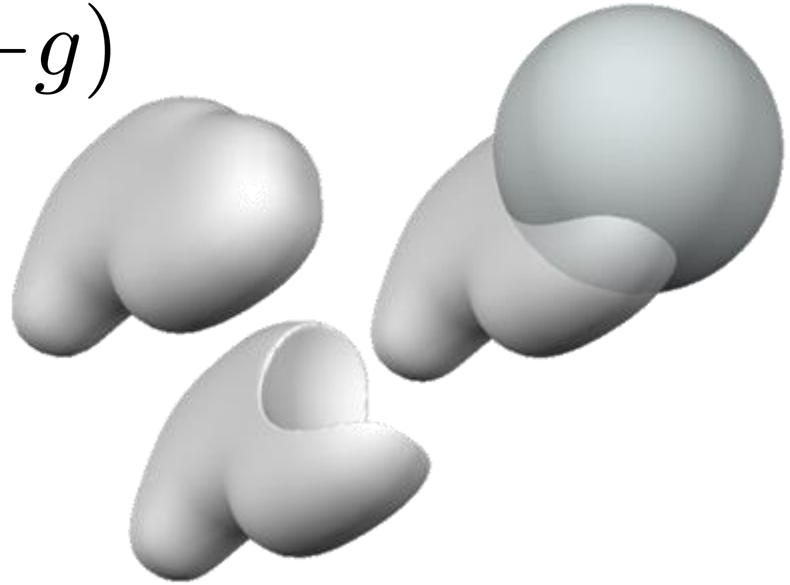


- Intersection: $$\bigcap_i f_i(x) = \max f_i(x)$$

Boolean Set Operations

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

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



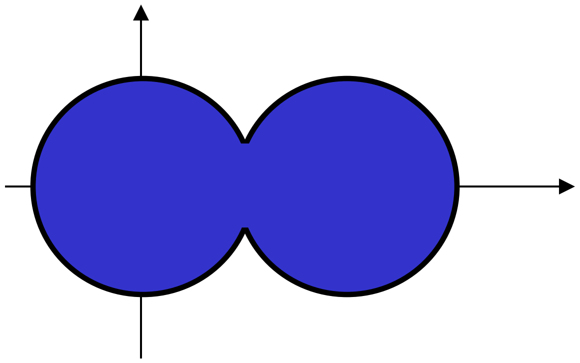
- Much easier than for parametric surfaces!

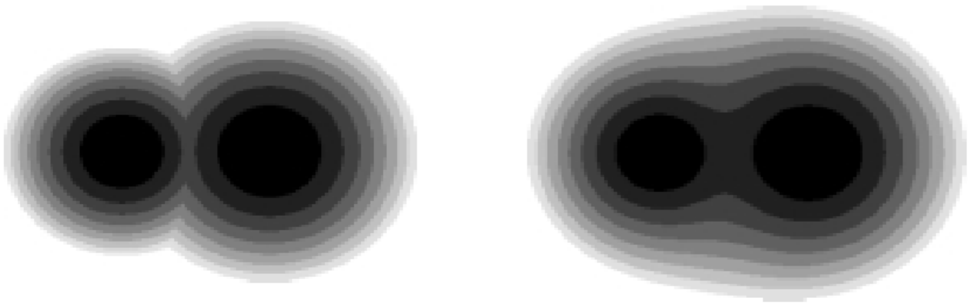
Smooth Set Operations

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

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

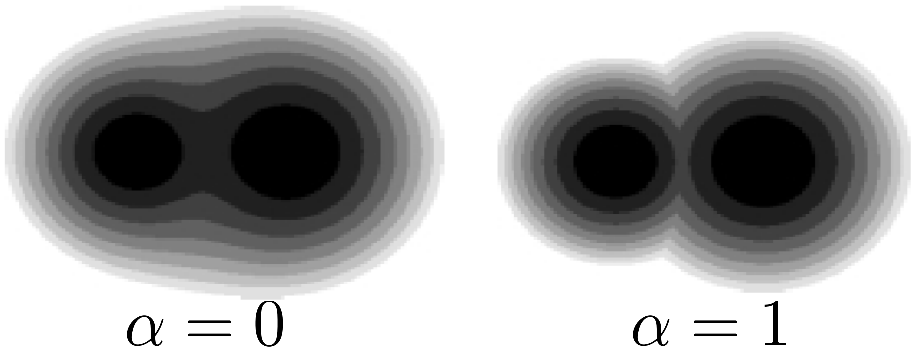
$$f \cap g = \frac{1}{1+\alpha}\left(f + g + \sqrt{f^2 + g^2 - 2\alpha f g}\right)$$





Smooth Set Operations

- Examples



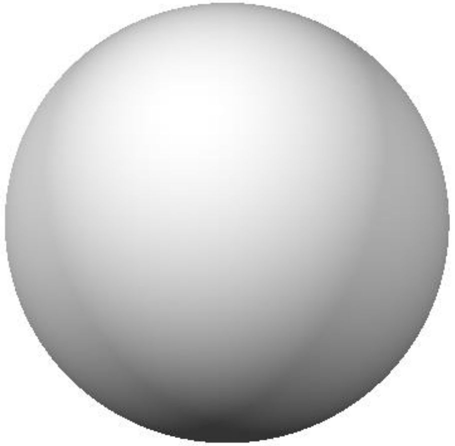
- For $\alpha = 1$, this is equivalent to $\min$ and $\max$

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

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

Designing with Implicit Surfaces

- Sphere: zero level set of this function:



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

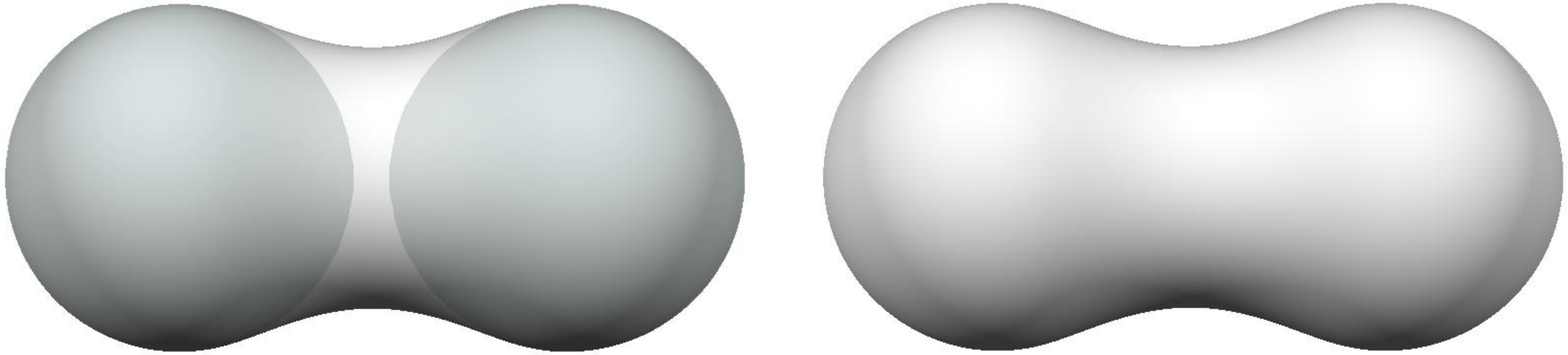
- But also the level set **at value e^{-1}** of this function:

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

Designing with Implicit Surfaces

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

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



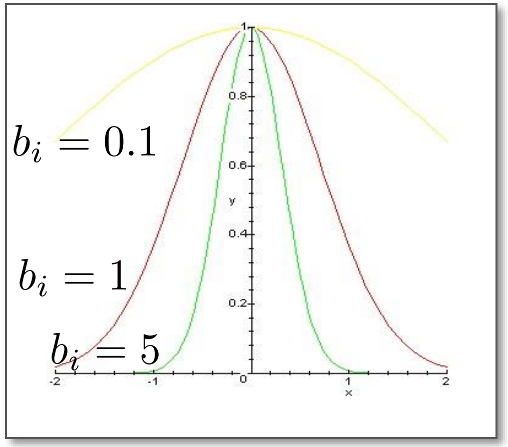
- Called “Metaballs” or “Blobs”

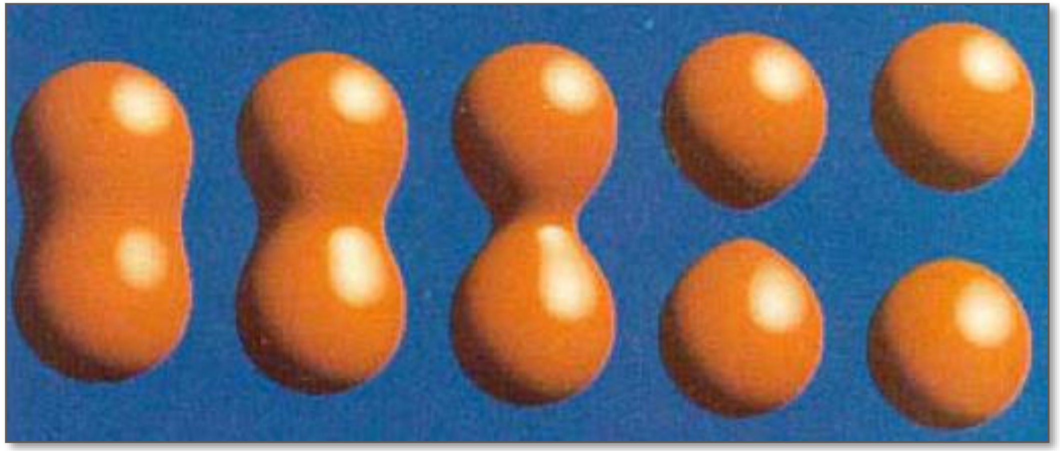
Blobs

- Suggested by Blinn [1982]
  - Defined implicitly by a potential function around a point $\mathbf{p}_i$:

$$f(\mathbf{p}) = a_i e^{-b_i \|\mathbf{p}-\mathbf{p}_i\|^2}$$

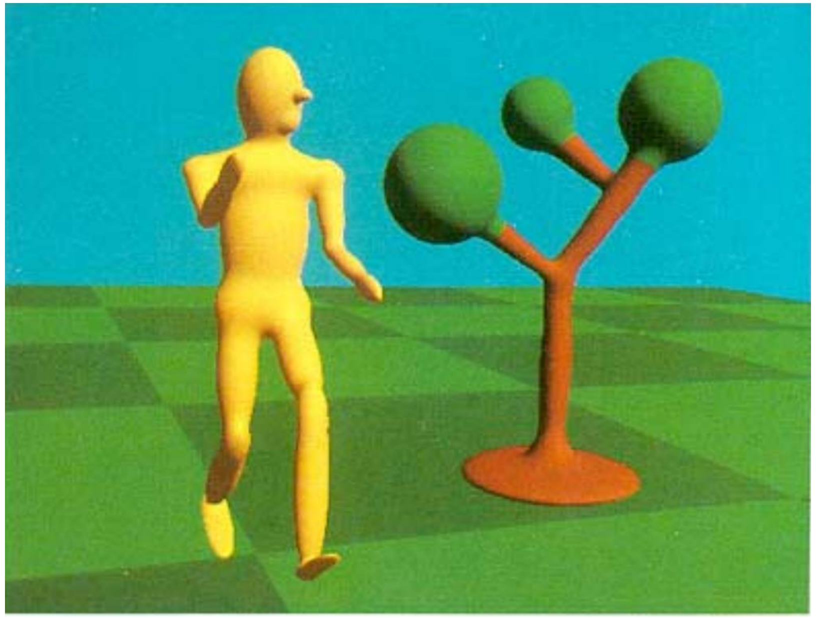
  - Set operations by simple addition/subtraction



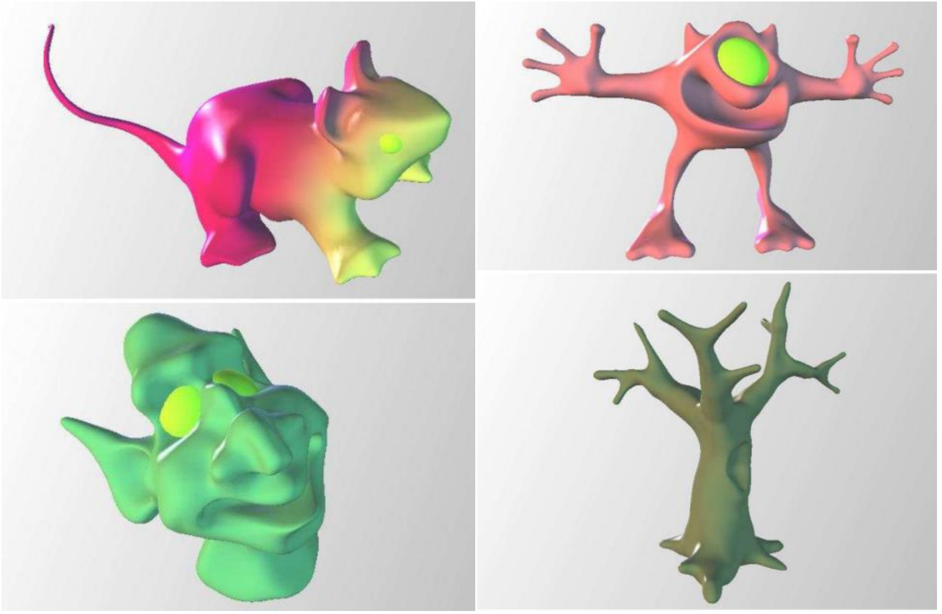


J. Blinn, "A Generalization of Algebraic Surface Drawing", ACM Transactions on Graphics, Vol. 1, No. 3, pp. 235-256, July, 1982.

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

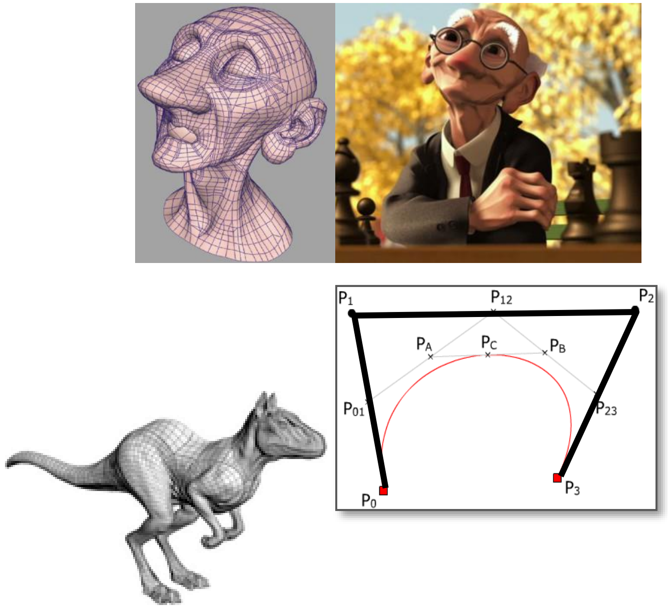
Implicit Curves and Surfaces

- Advantages
  - Easy to determine inside/outside
  - Easy to determine if a point is **on** the curve/surface
  - *Regular* sampling of the entire space, like a grid (good e.g. for neural networks)

- Disadvantages
  - Hard to generate points on the curve/surface
  - Does not easily lend itself to (real-time) rendering

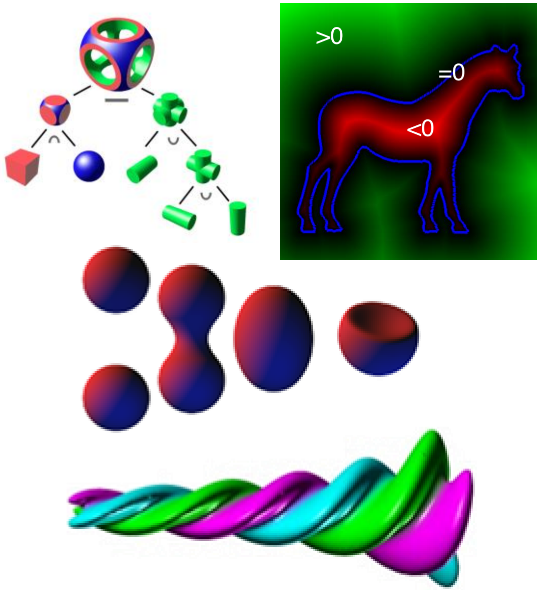
Summary

Parametric



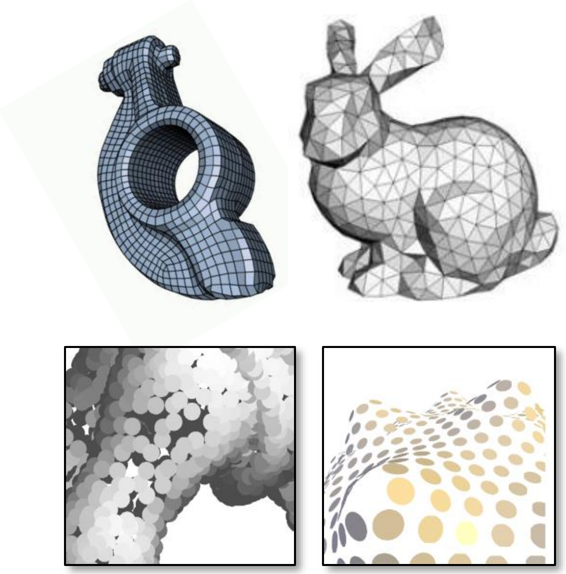
- Splines, tensor-product surfaces
- Subdivision surfaces

Implicit



- Metaballs/blobs
- Distance fields

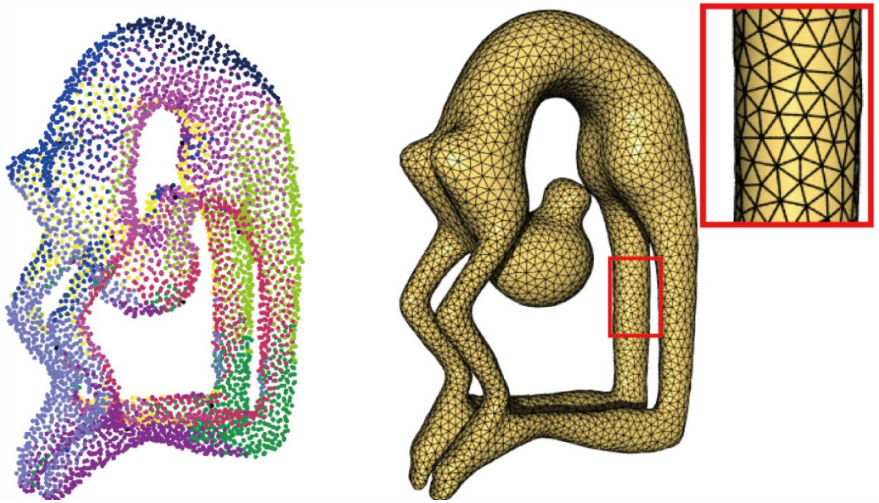
Discrete/Sampled



- Meshes
- Point set surfaces

In the Next Lectures

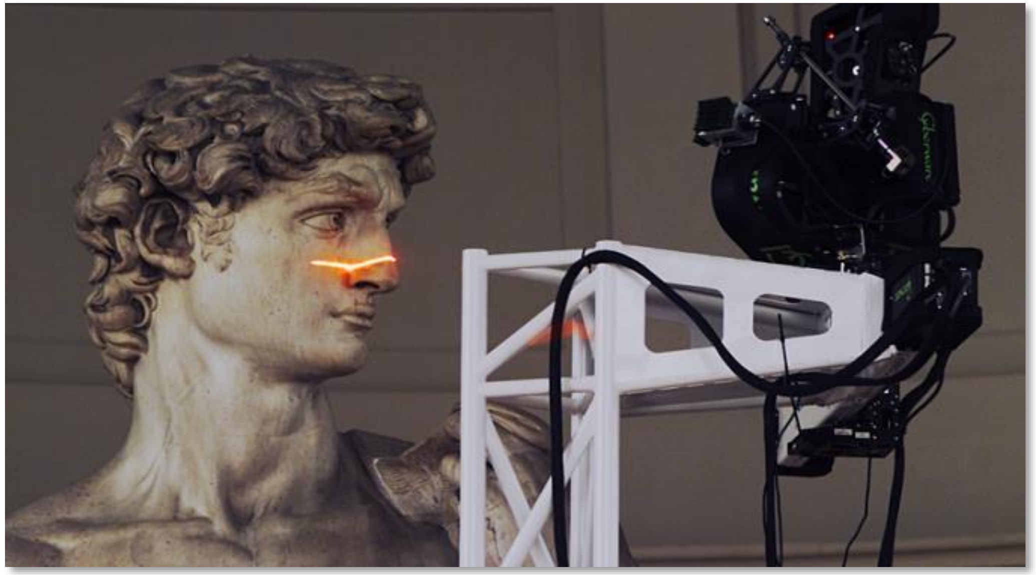
- How to get a clean, watertight surface mesh from a sampled point set

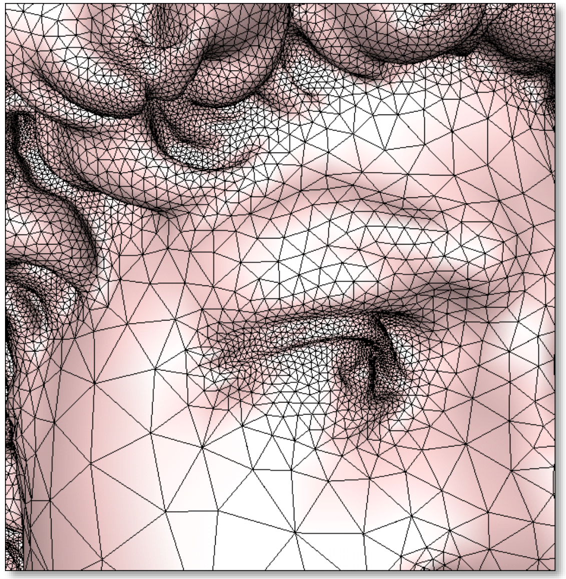


- The most popular way:
points ➐implicit function ➐surface mesh

Next Week

- A bit about geometry acquisition
- All About Meshes





Thank you