

CS 231N Lecture 16: 3D Vision





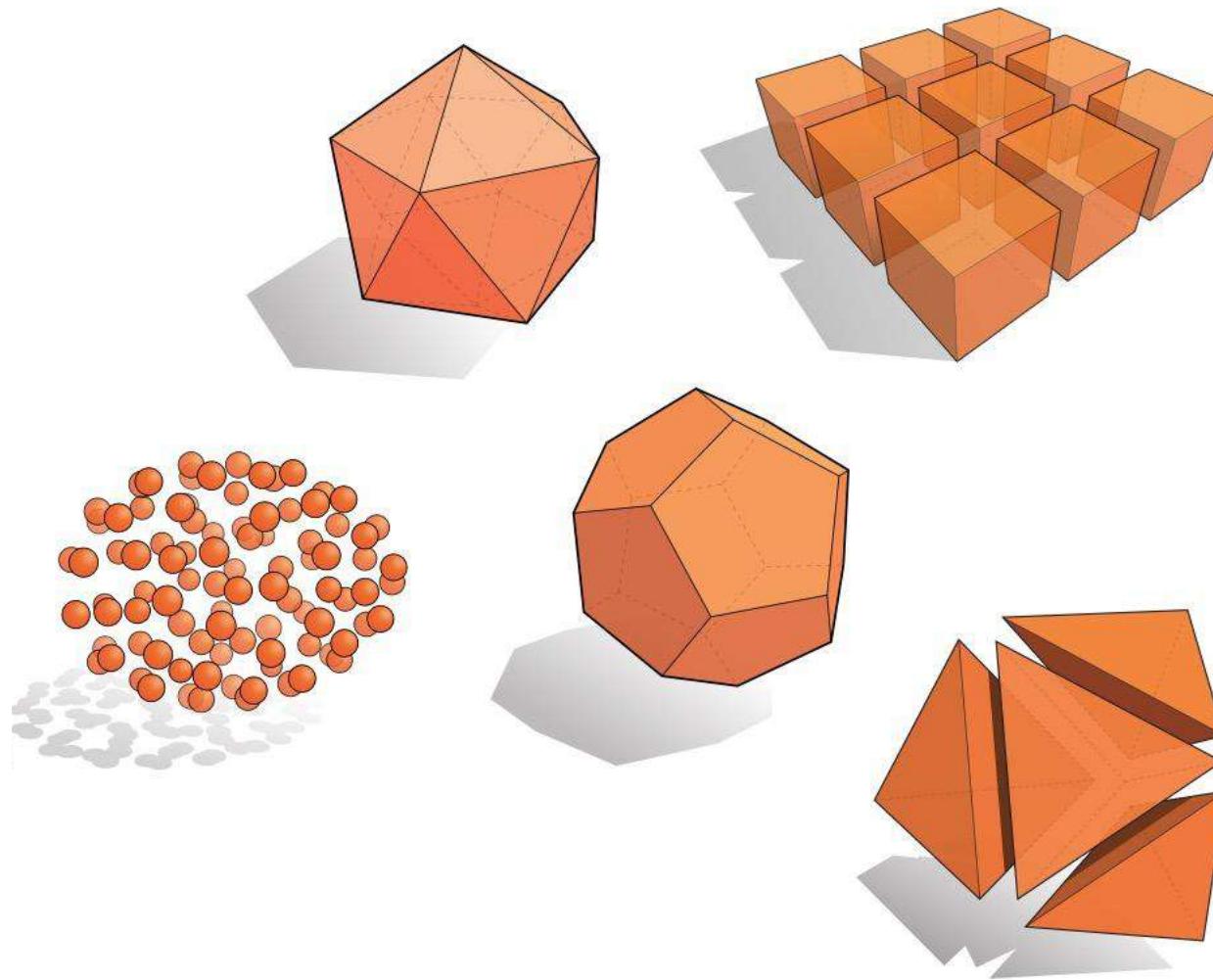


NATIONAL
GEOGRAPHIC

Photograph by Adriana Franco, Your Shot

©2013 National Geographic Society. All rights reserved.

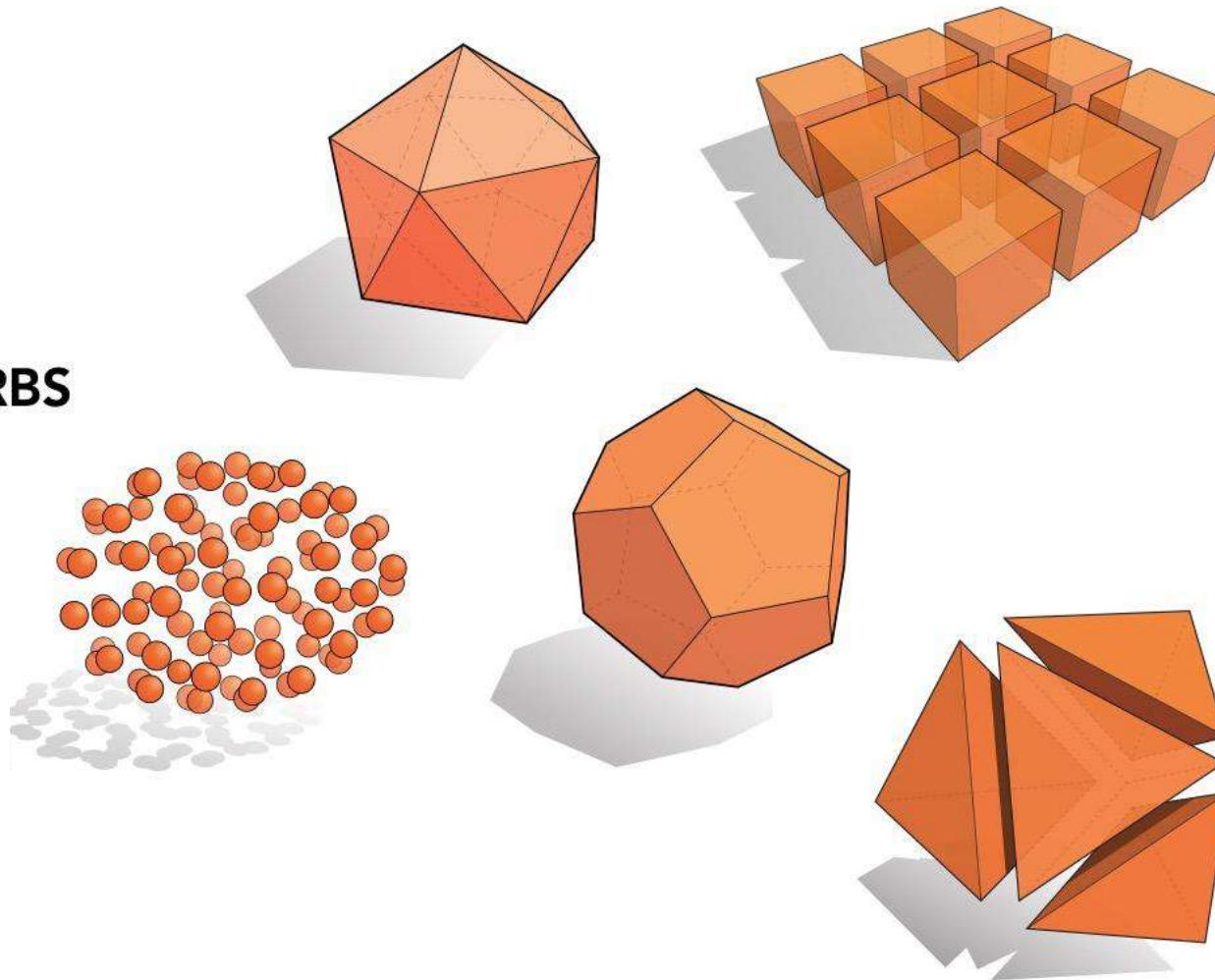
Many Ways to Represent Geometry



Many Ways to Represent Geometry

Explicit

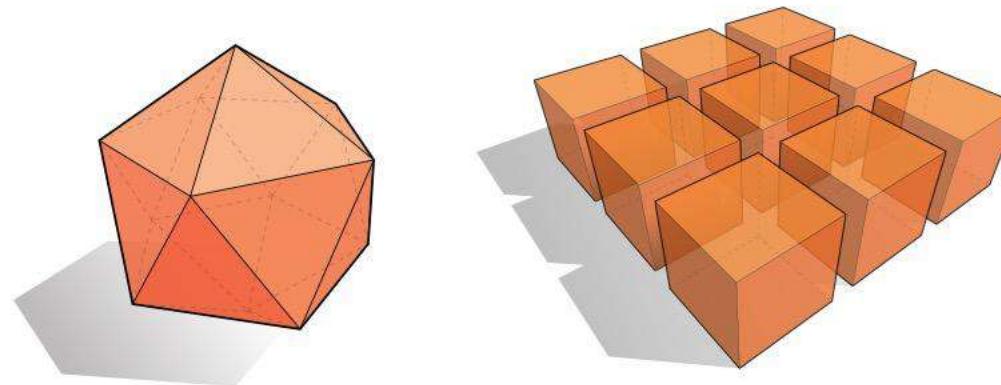
- point cloud
- polygon mesh
- subdivision, NURBS
- ...



Many Ways to Represent Geometry

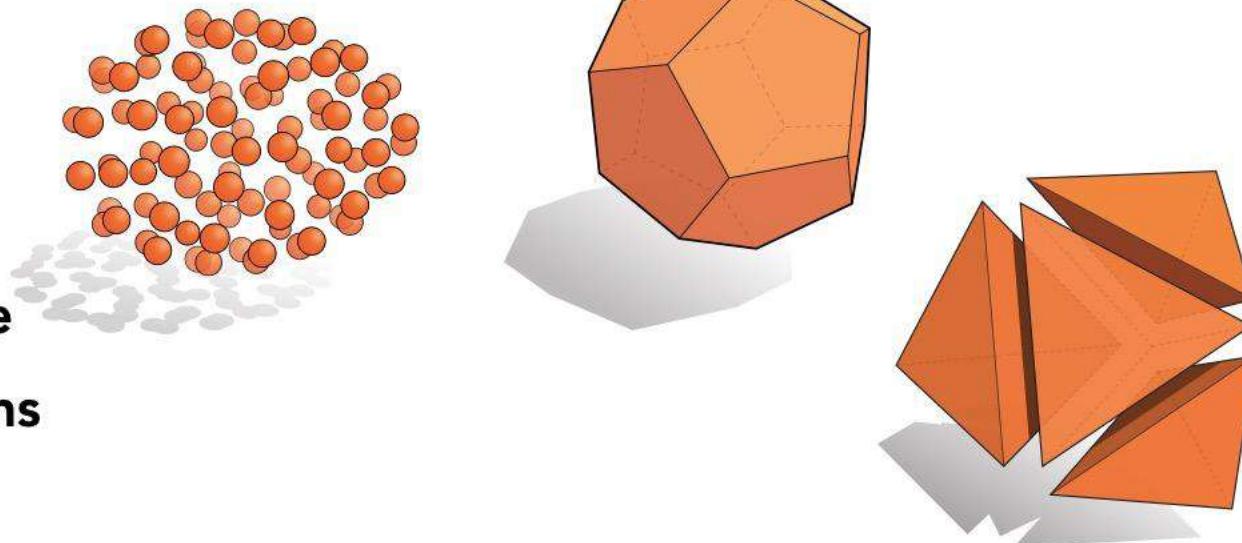
Explicit

- point cloud
- polygon mesh
- subdivision, NURBS
- ...



Implicit

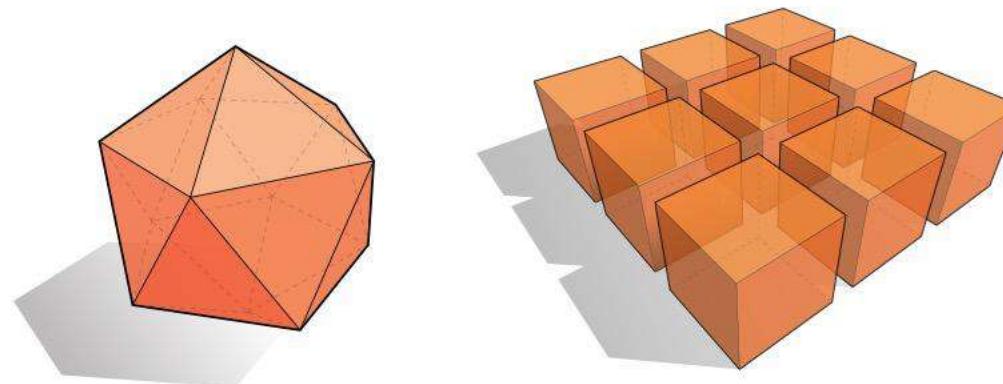
- level sets
- algebraic surface
- distance functions
- ...



Many Ways to Represent Geometry

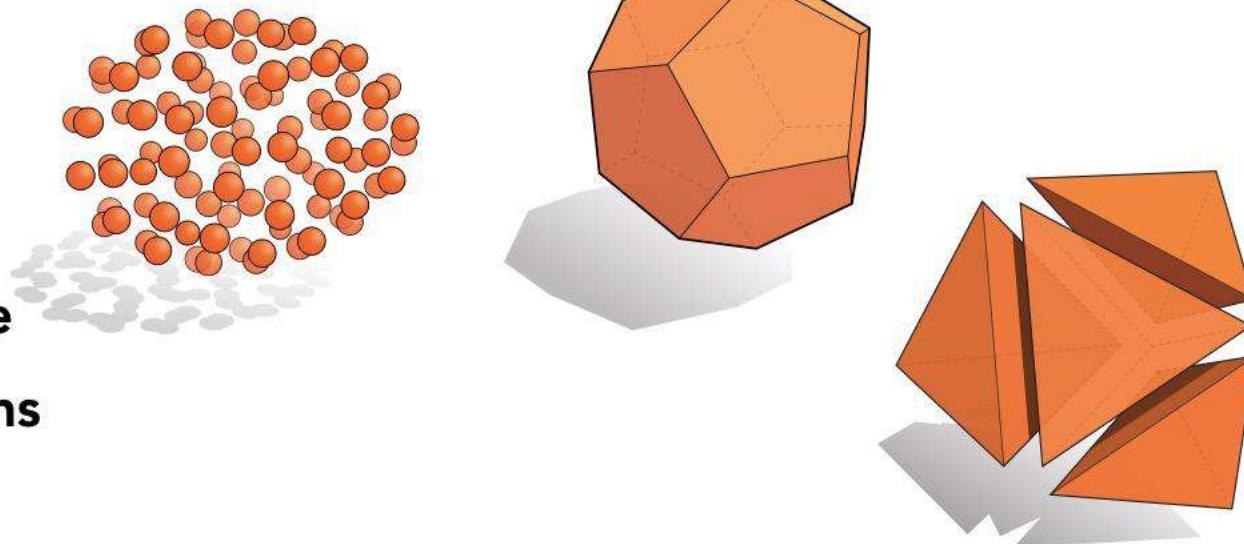
Explicit

- point cloud
- polygon mesh
- subdivision, NURBS
- ...



Implicit

- level sets
- algebraic surface
- distance functions
- ...

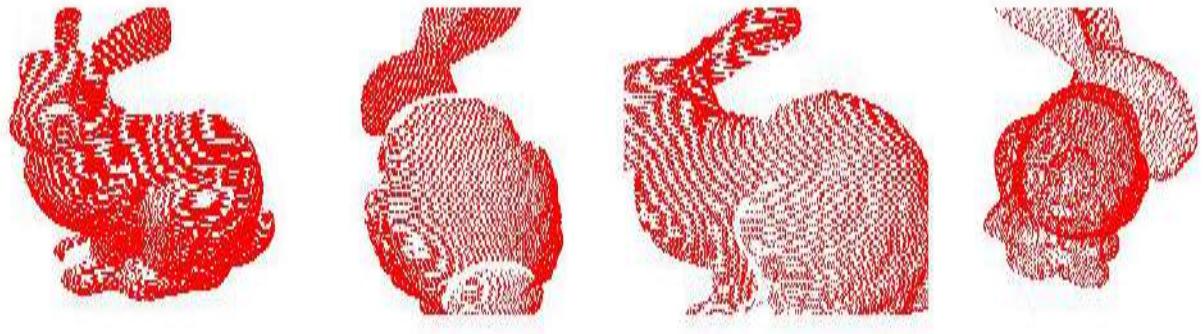


Each choice best suited to a different task/type of geometry

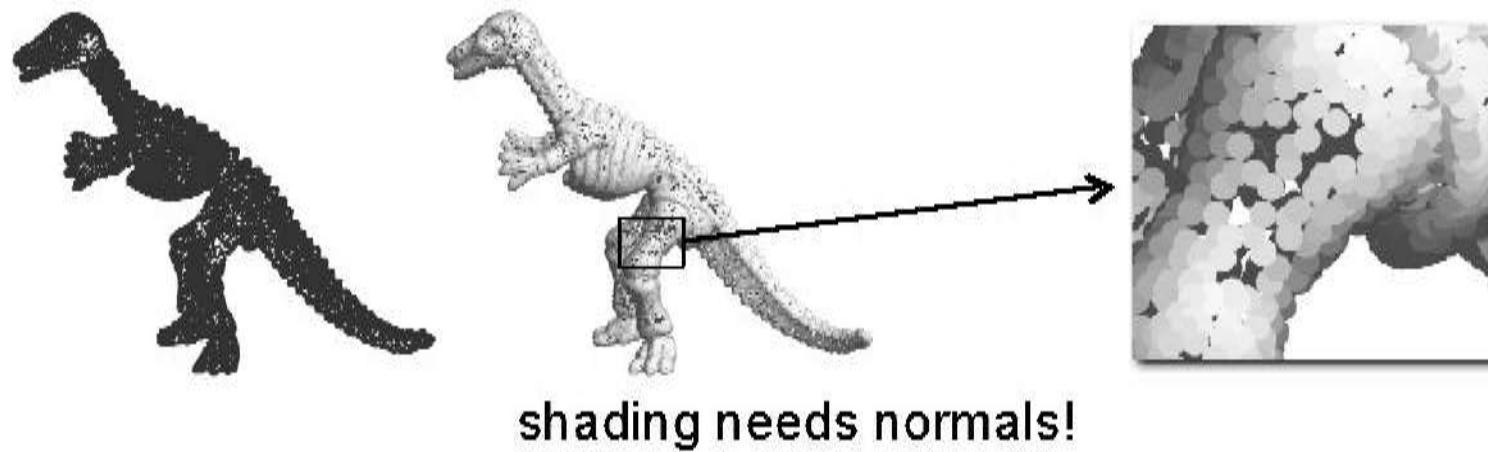
Representation Considerations

- Needs to be stored in the computer
- Creation of new shapes
 - Input metaphors, interfaces...
- Operations
 - Editing, simplification, smoothing, filtering, repairing...
- Rendering
 - Rasterization, ray tracing...
- Animation

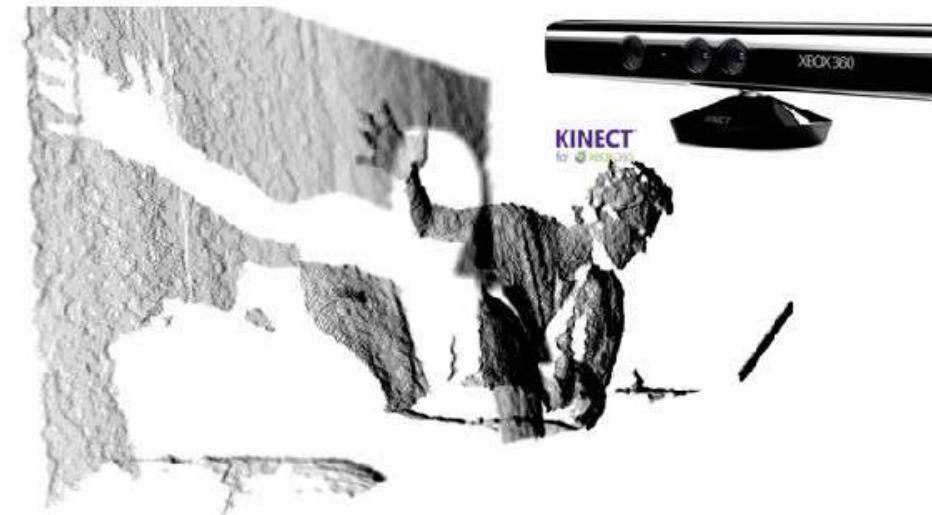
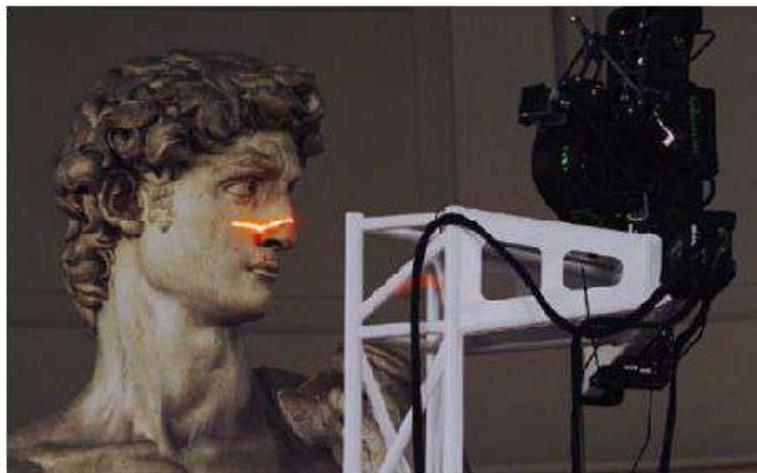
Point Clouds



- Simplest representation: **only points**, no connectivity
- Collection of (x,y,z) coordinates, possibly with normal
- Points with orientation are called **surfels**

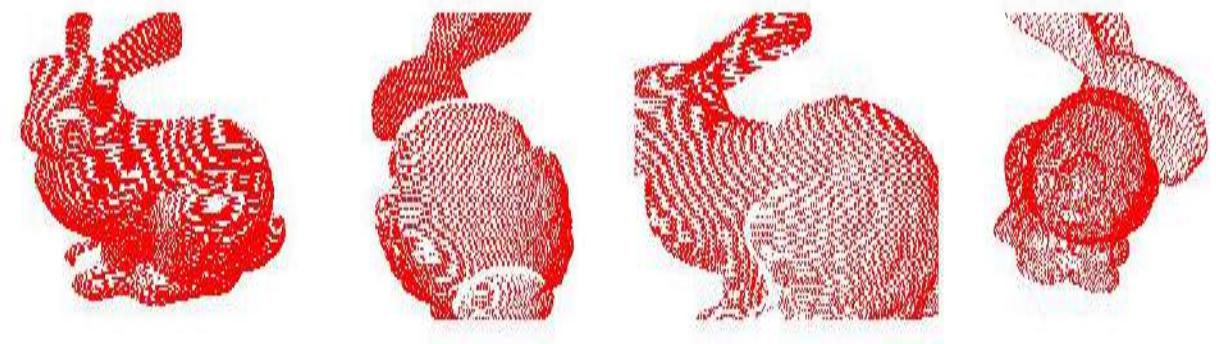


Output of Acquisition

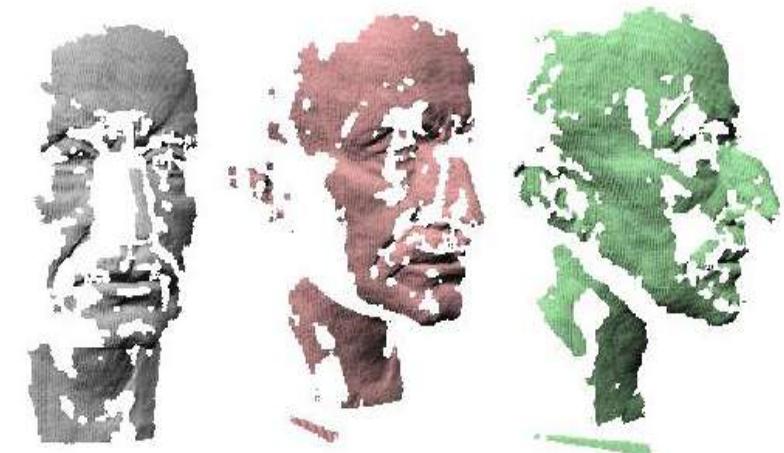


Slide credit: Hao Su

Point Clouds

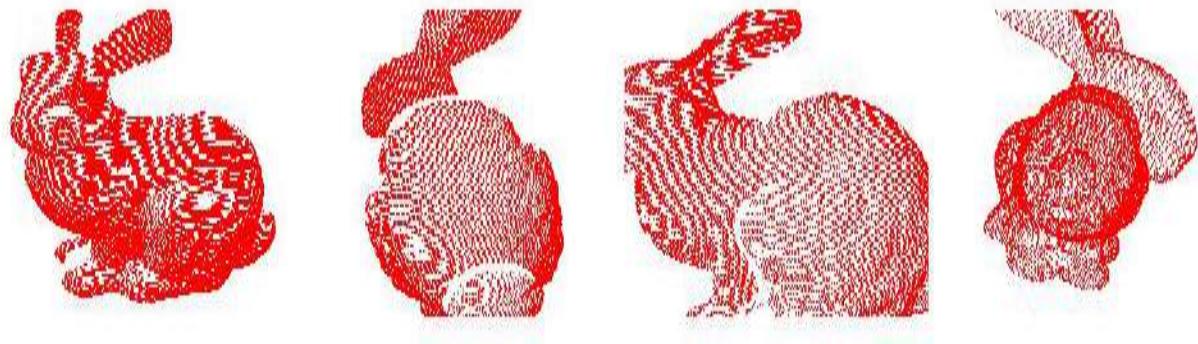


- Simplest representation: **only points**, no connectivity
- Collection of (x,y,z) coordinates, possibly with normal
- Points with orientation are called **surfels**
- Often results from scanners
- Potentially noisy
- Registration of multiple images



set of raw scans

Point Clouds

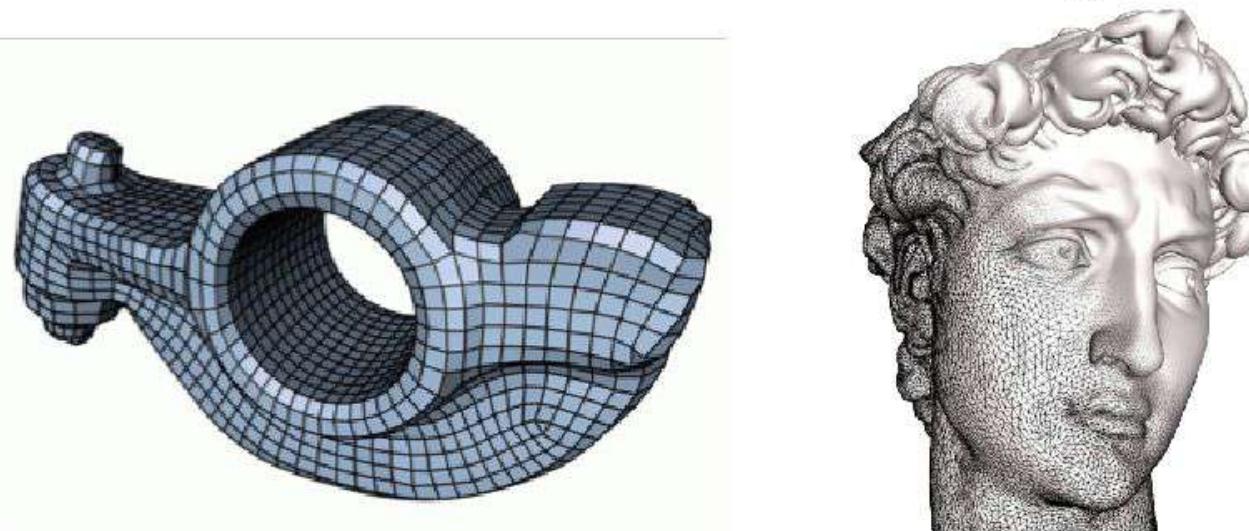
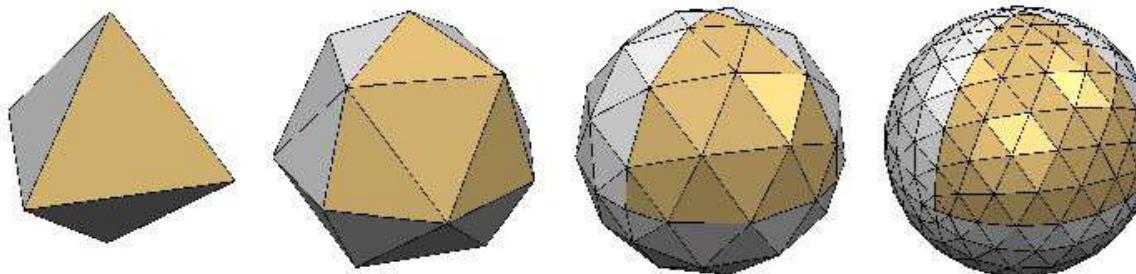


- Easily represent any kind of geometry
- Useful for large datasets
- Difficult to draw in undersampled regions
- Other limitations:
 - No simplification or subdivision
 - No direction smooth rendering
 - No topological information



Polygonal Meshes

- Boundary representations of objects



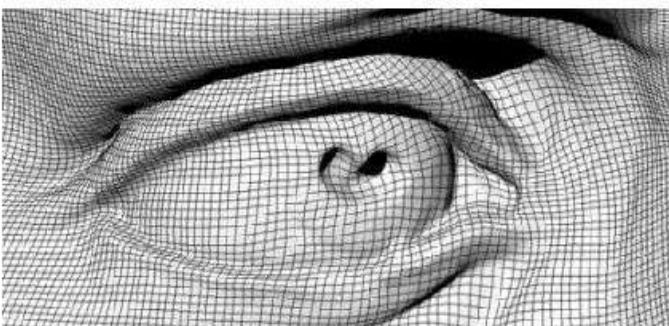
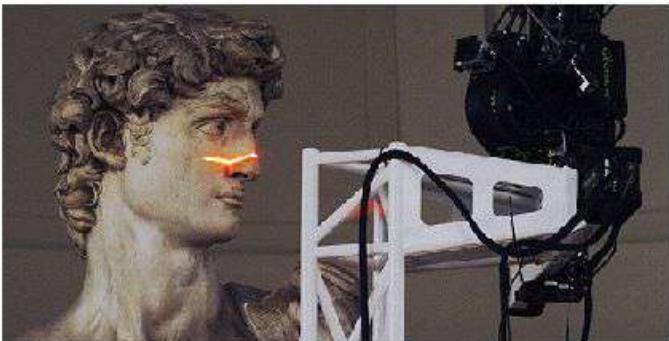
A Large Triangle Mesh

David

Digital Michelangelo Project

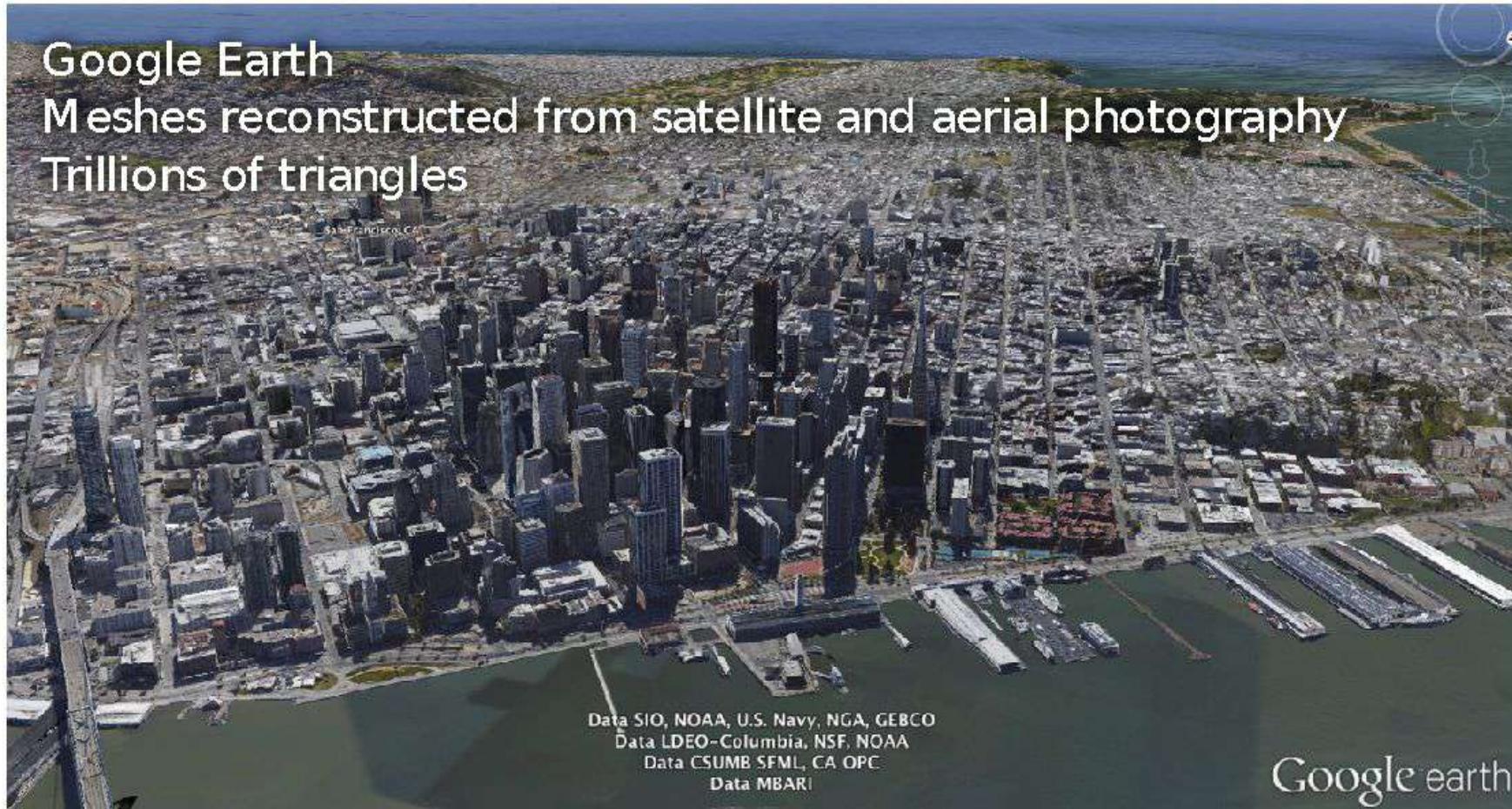
28,184,526 vertices

56,230,343 triangles



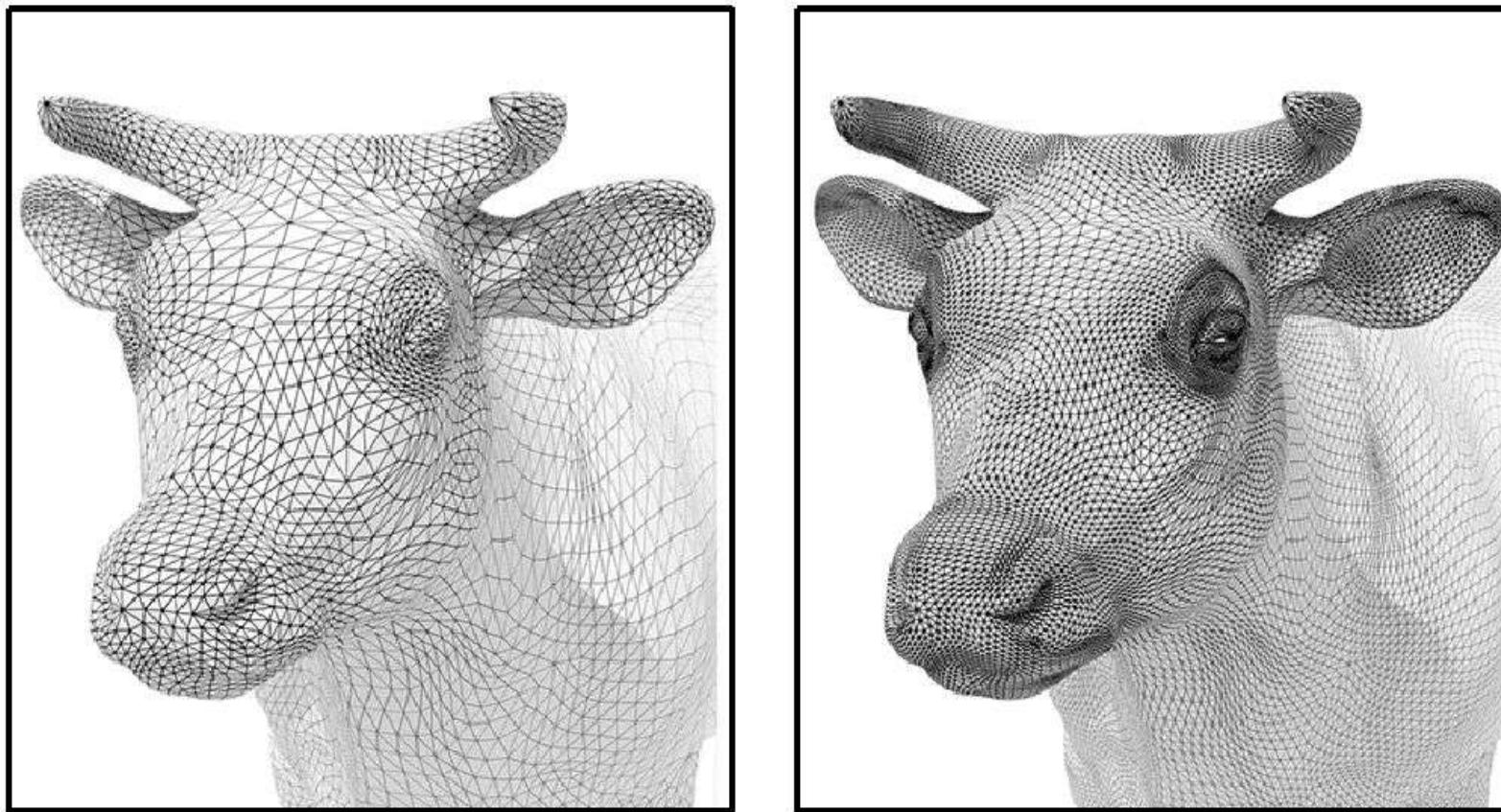
Slide credit: Ren Ng

A Very Large Triangle Mesh



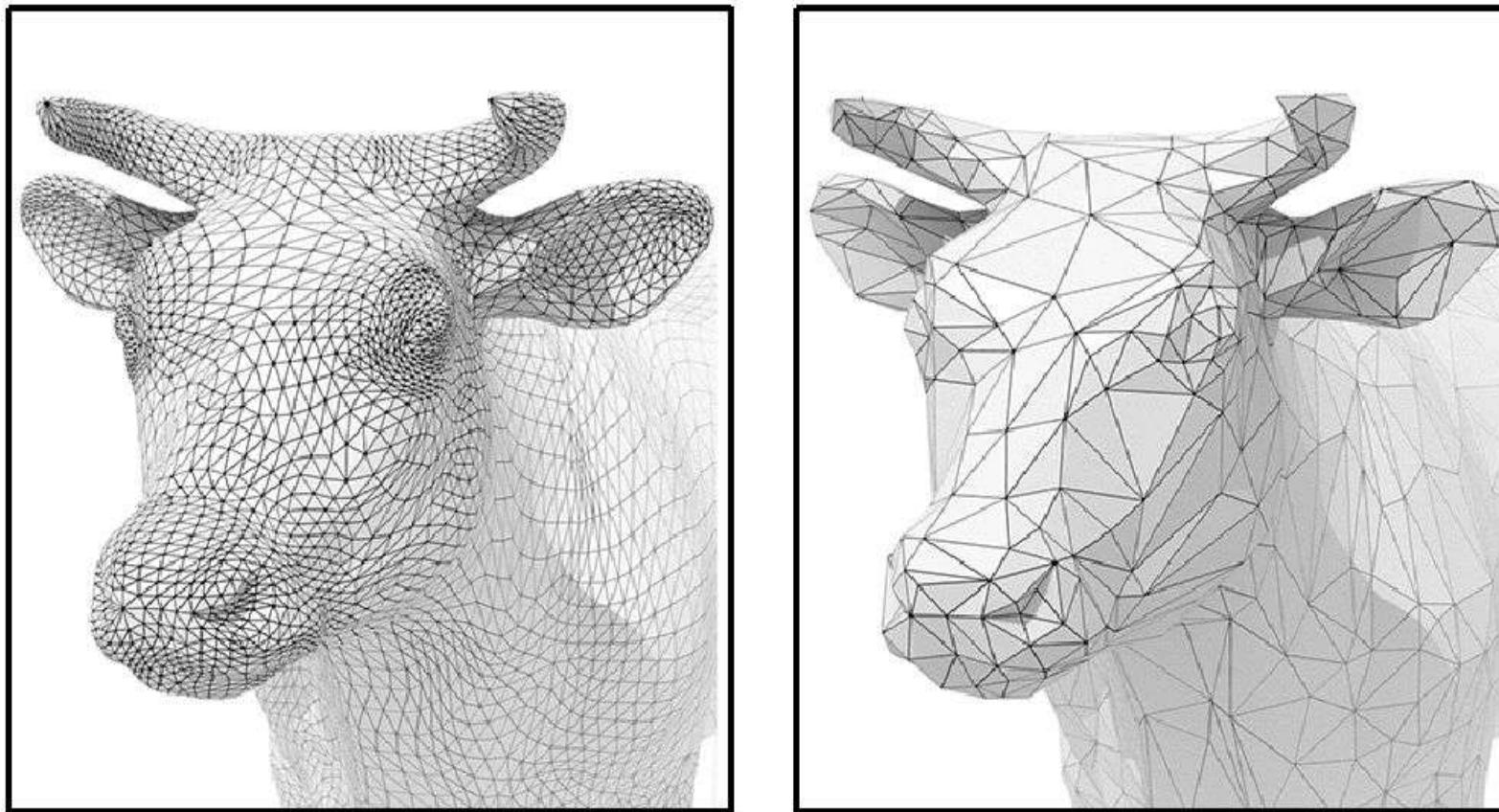
Slide credit: Ren Ng

Mesh Upsampling – Subdivision



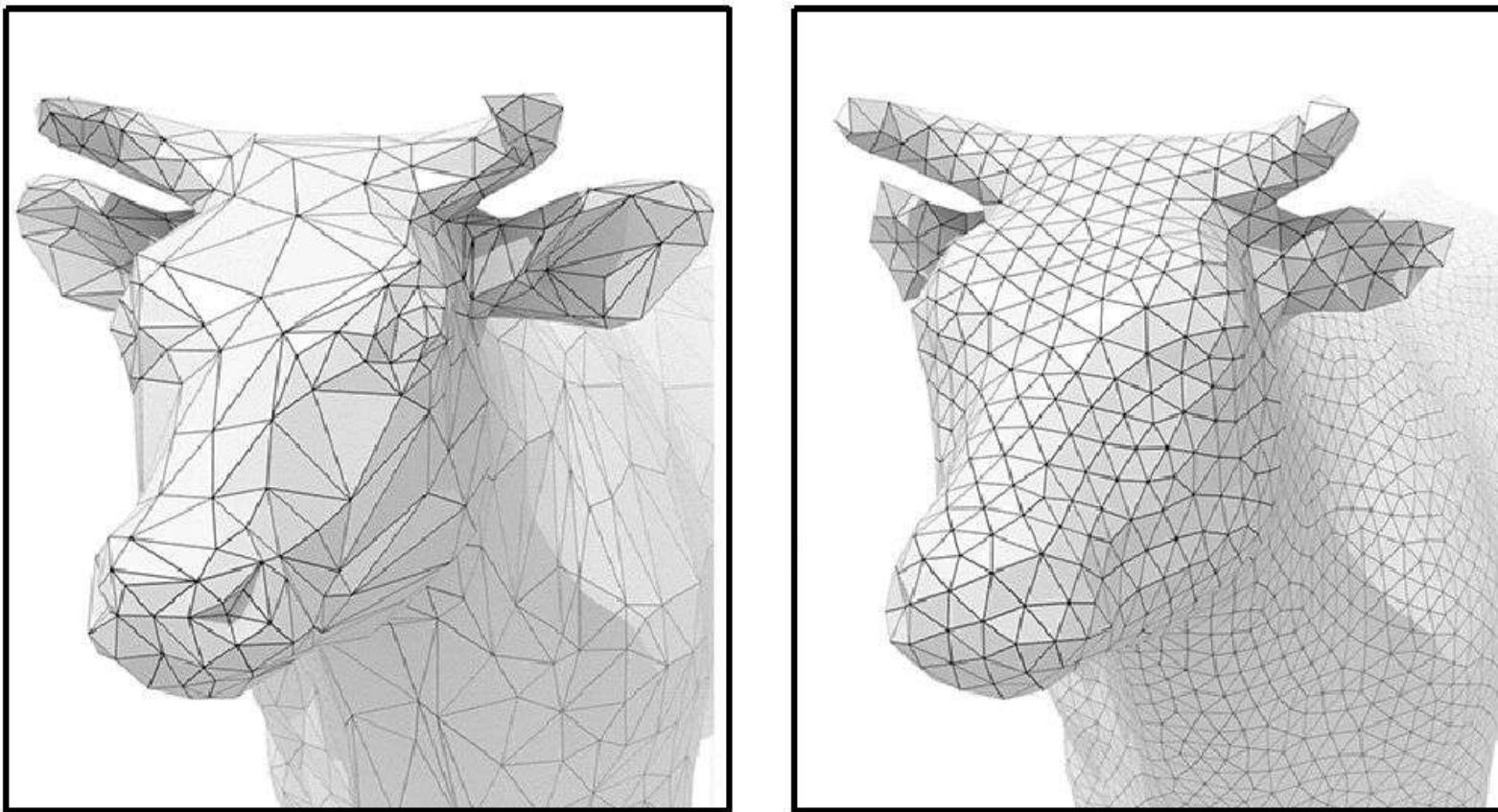
Increase resolution via interpolation

Mesh Downsampling – Simplification



Decrease resolution; try to preserve shape/appearance

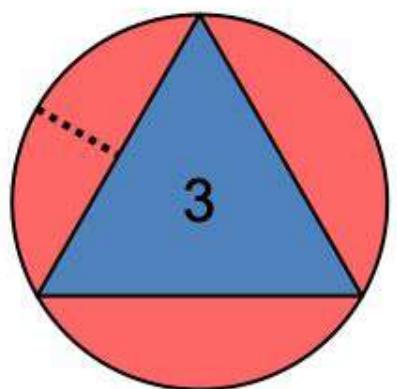
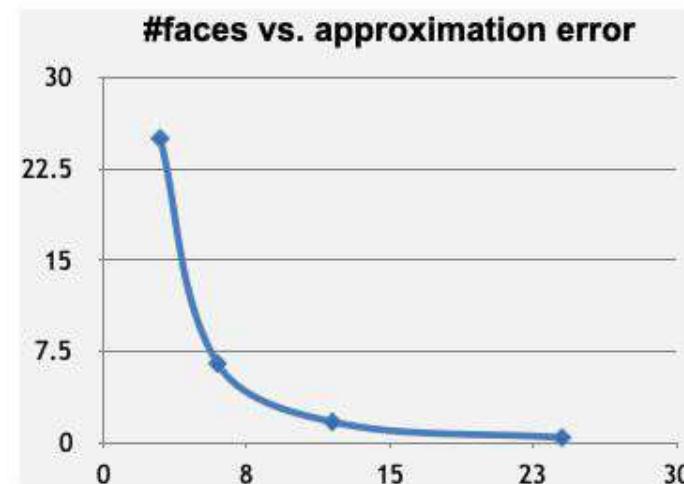
Mesh Regularization



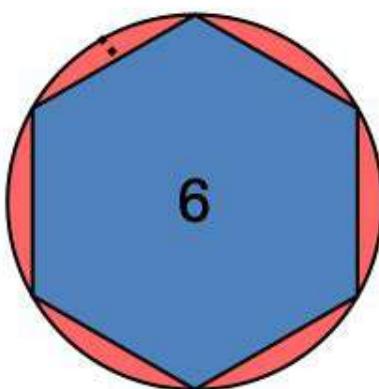
Modify sample distribution to improve quality

Meshes as Approximations of Smooth Surfaces

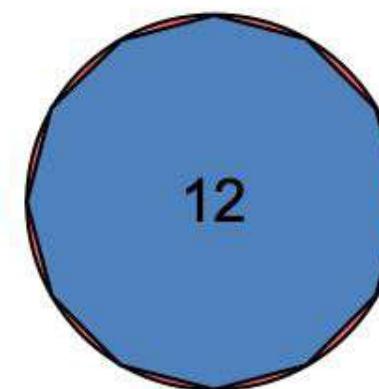
- Piecewise linear approximation
 - Error is $O(h^2)$



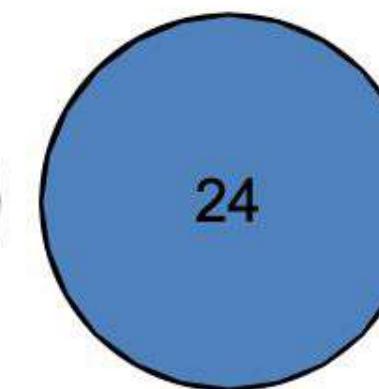
25%



6.5%



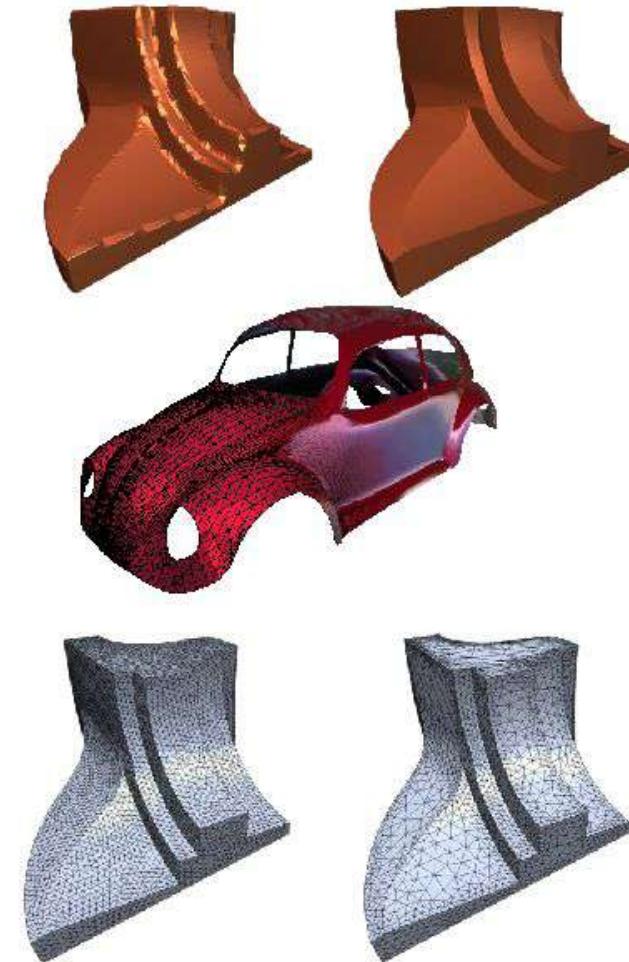
1.7%



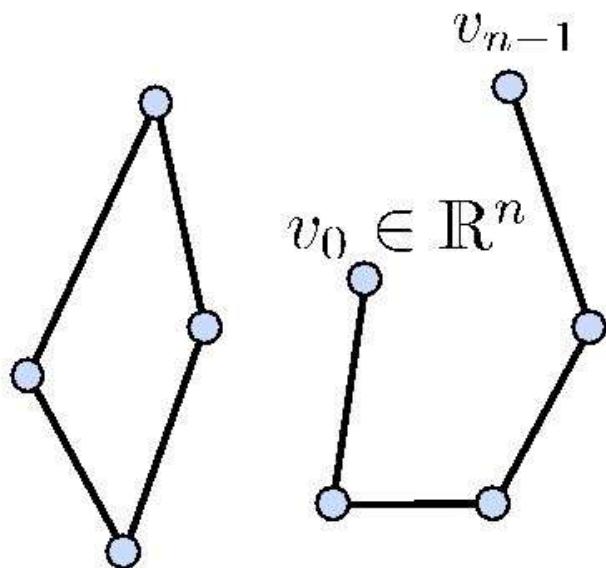
0.4%

Polygonal Meshes

- Polygonal meshes are a good representation
 - approximation $O(h^2)$
 - arbitrary topology
 - adaptive refinement
 - efficient rendering

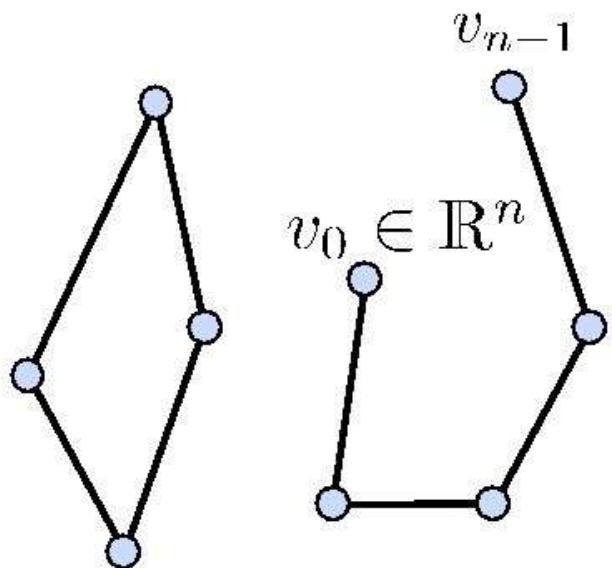


Polygon



- **Vertices:** v_0, v_1, \dots, v_{n-1}
- **Edges:** $\{(v_0, v_1), \dots, (v_{n-2}, v_{n-1})\}$

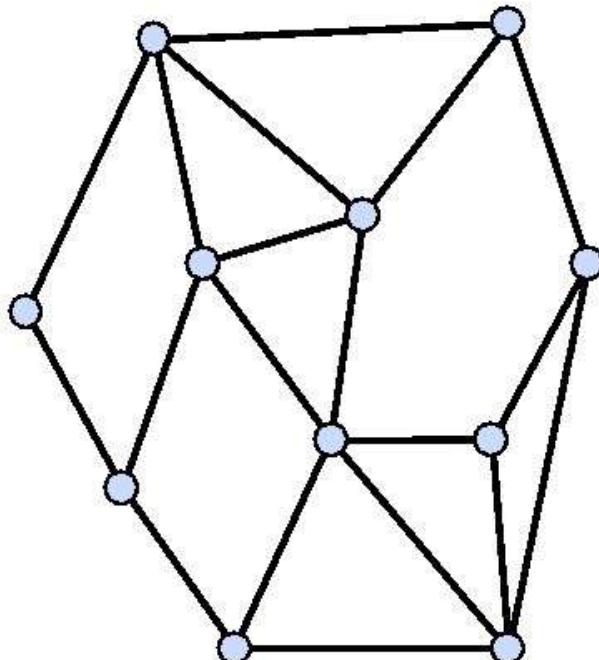
Polygon



- **Vertices:** v_0, v_1, \dots, v_{n-1}
- **Edges:** $\{(v_0, v_1), \dots, (v_{n-2}, v_{n-1})\}$
- **Closed:** $v_0 = v_{n-1}$
- **Planar:** all vertices on a plane
- **Simple:** not self-intersecting

Polygonal Mesh

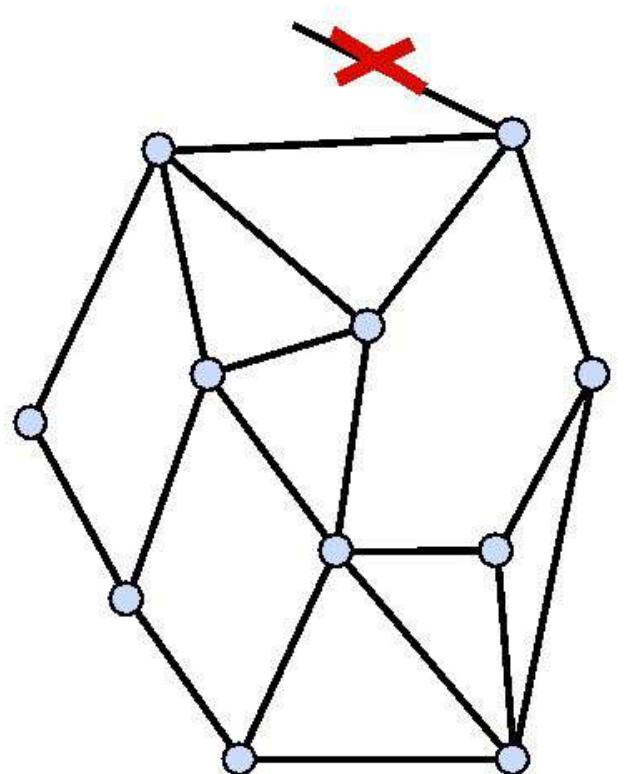
- A finite set M of closed, simple polygons Q_i is a polygonal mesh
- The intersection of two polygons in M is either empty, a vertex, or an edge



$$M = \langle V, E, F \rangle$$

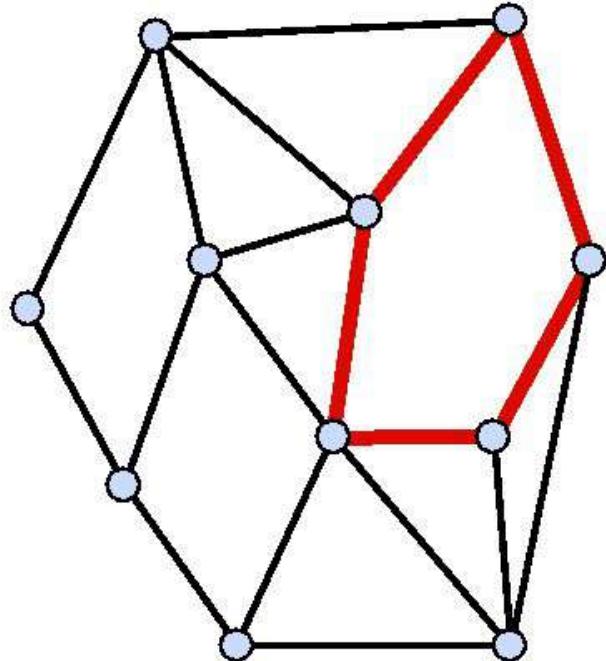
vertices edges faces

Polygonal Mesh



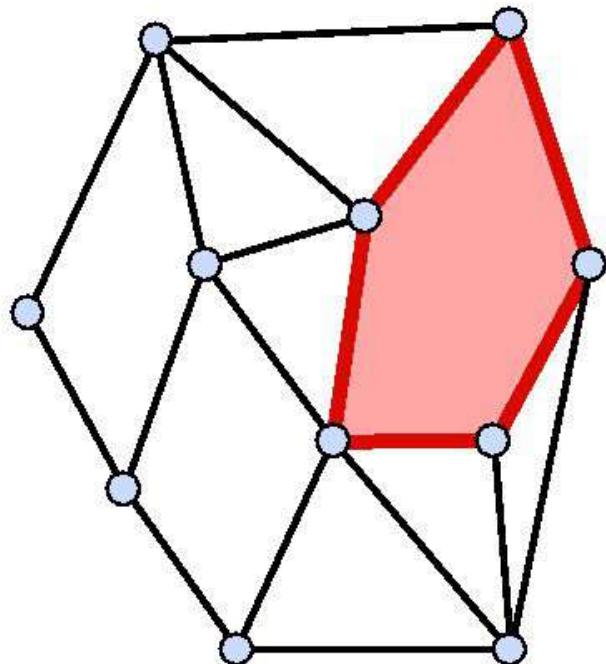
- A finite set M of closed, simple polygons Q_i is a polygonal mesh
- The intersection of two polygons in M is either empty, a vertex, or an edge
- Every edge belongs to at least one polygon

Polygonal Mesh



- A finite set M of closed, simple polygons Q_i is a polygonal mesh
- The intersection of two polygons in M is either empty, a vertex, or an edge
- Every edge belongs to at least one polygon
- Each Q_i defines a **face** of the polygonal mesh

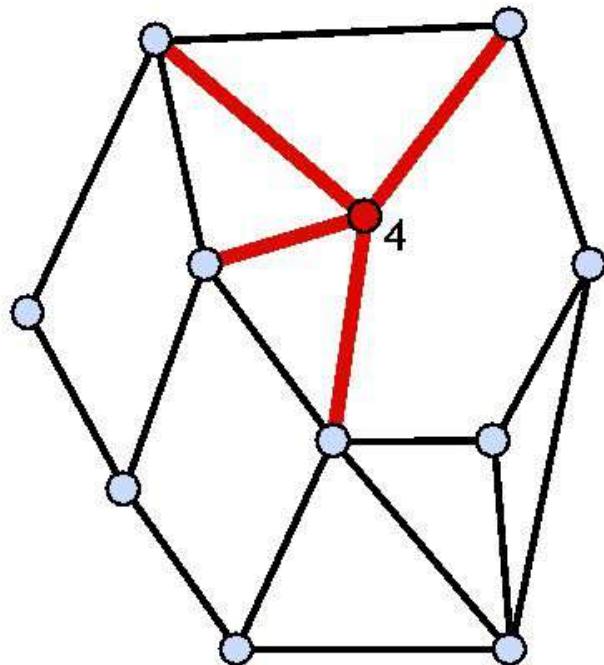
Polygonal Mesh



- A finite set M of closed, simple polygons Q_i is a polygonal mesh
- The intersection of two polygons in M is either empty, a vertex, or an edge
- Every edge belongs to at least one polygon
- Each Q_i defines a face of the polygonal mesh

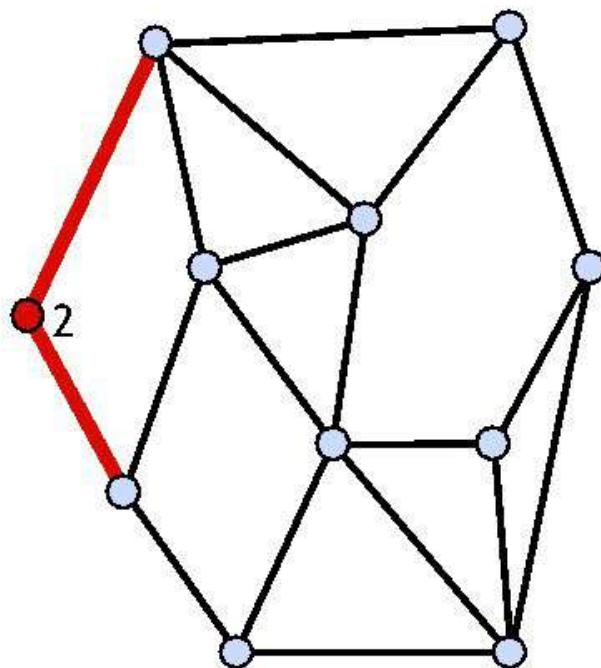
Polygonal Mesh

- Vertex **degree** or **valence** = number of incident edges



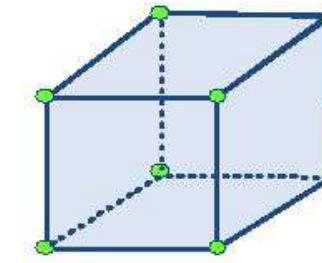
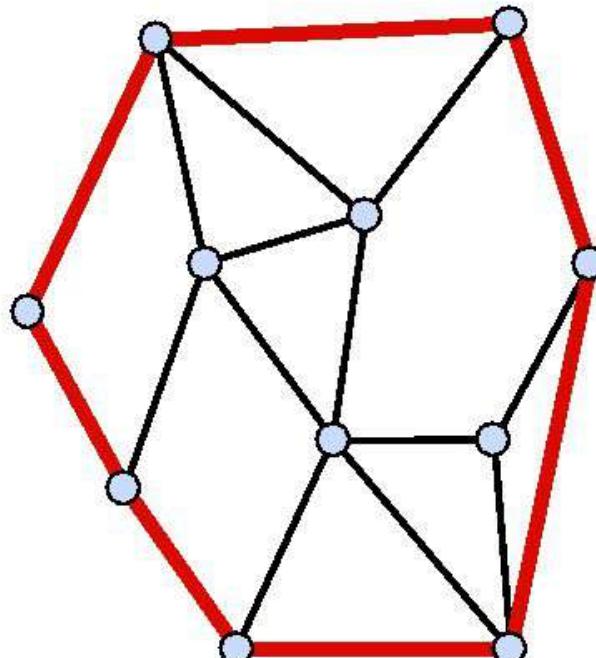
Polygonal Mesh

- Vertex degree or valence = number of incident edges



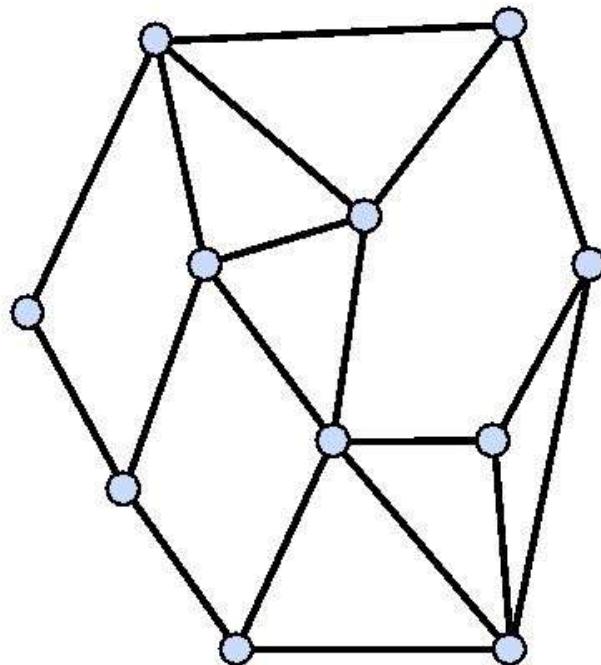
Polygonal Mesh

- Boundary: the set of all edges that belong to only one polygon
 - Either empty or forms closed loops
 - If empty, then the polygonal mesh is closed



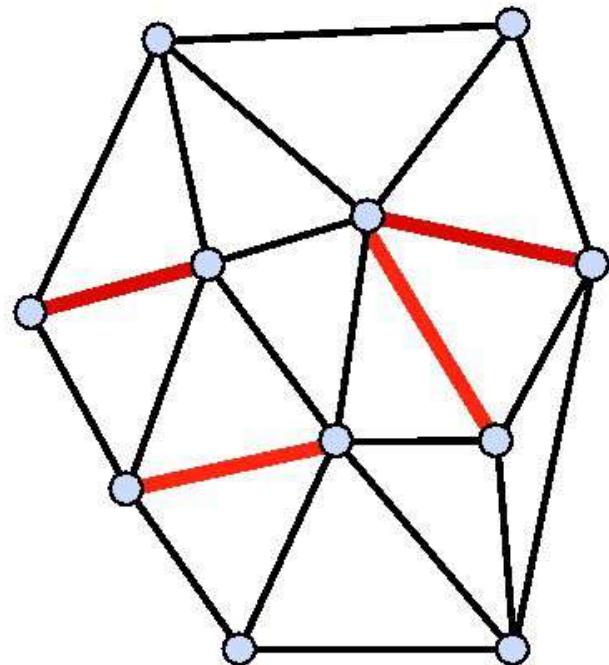
Triangulation

- Polygonal mesh where every face is a triangle



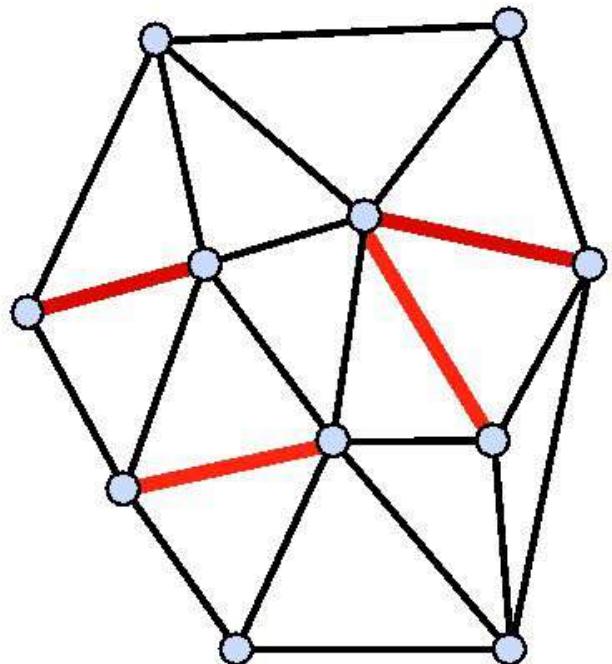
Triangulation

- Polygonal mesh where every face is a triangle



Triangulation

- Polygonal mesh where every face is a triangle



- Simplifies data structures
- Simplifies rendering
- Simplifies algorithms
- Each face planar and convex
- Any polygon can be triangulated

Data Structures

- What should be stored?
- Geometry: 3D coordinates



Data Structures



- What should be stored?
 - Geometry: 3D coordinates
 - Connectivity
 - Adjacency relationships

Data Structures



- What should be stored?
 - Geometry: 3D coordinates
 - Connectivity
 - Adjacency relationships
 - Attributes
 - Normal, color, texture coordinates
 - Per vertex, face, edge

Simple Data Structures: Indexed Face Set

- Used in formats
 - OBJ, OFF, WRL
- Storage
 - Vertex: position
 - Face: vertex indices
 - 12 bytes per vertex
 - 12 bytes per face
 - $36*v$ bytes for the mesh
- No explicit neighborhood info

Vertices			
v0	x0	y0	z0
v1	x1	x1	z1
v2	x2	y2	z2
v3	x3	y3	z3
v4	x4	y4	z4
v5	x5	y5	z5
v6	x6	y6	z6
...

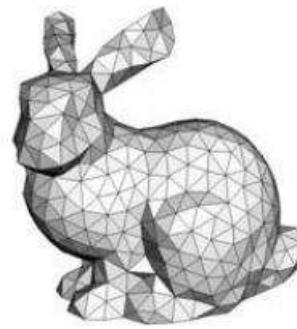
Triangles			
t0	v0	v1	v2
t1	v0	v1	v3
t2	v2	v4	v3
t3	v5	v2	v6
...

Shape Representations

Non-parametric



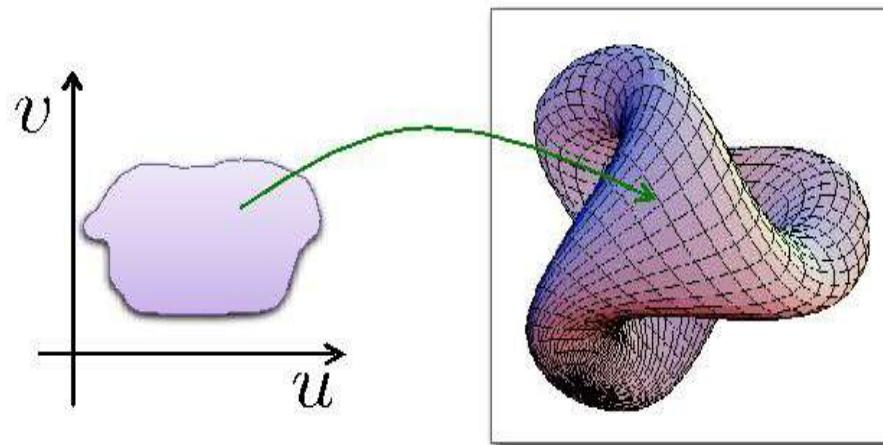
Points



Meshes

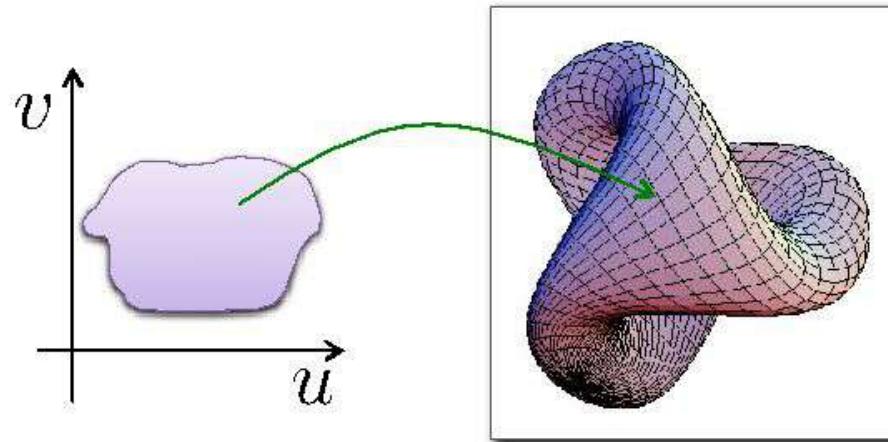
Parametric Representation

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



Parametric Representation

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



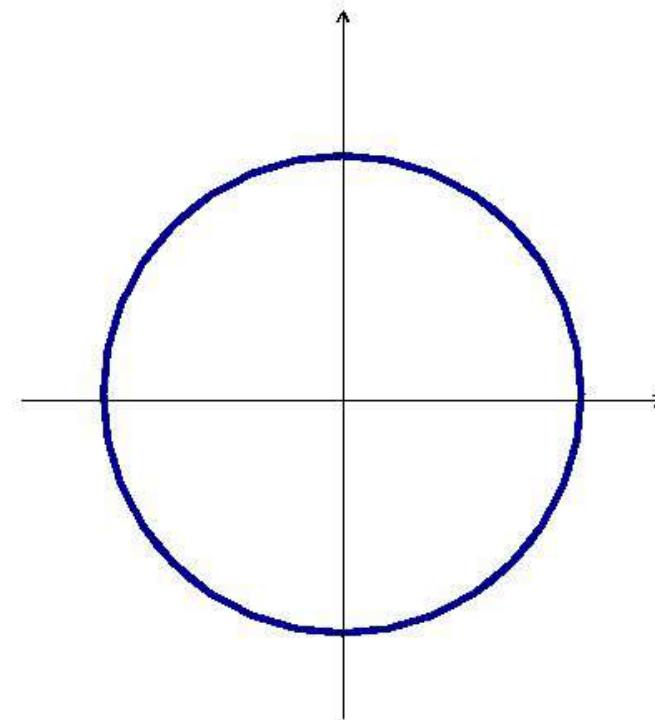
$$s(u, v) = (x(u, v), y(u, v), z(u, v))$$

Parametric Curves

- Example: Explicit curve/circle in 2D

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

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



Parametric Curves

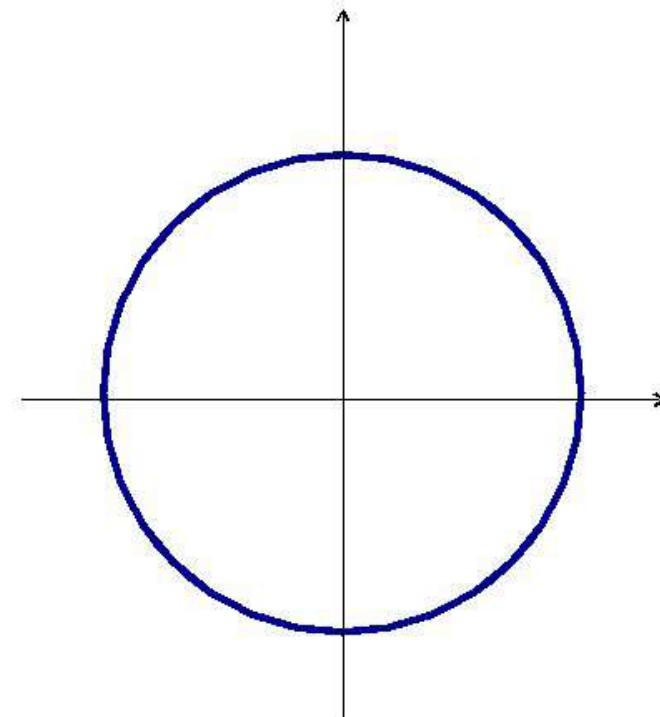
- Example: Explicit curve/circle in 2D

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

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

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

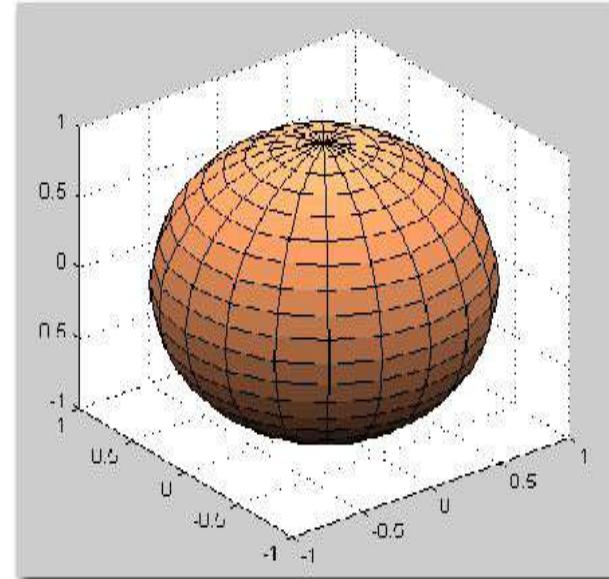
$$t \in [0, 2\pi)$$



Parametric Surfaces

- Sphere in 3D

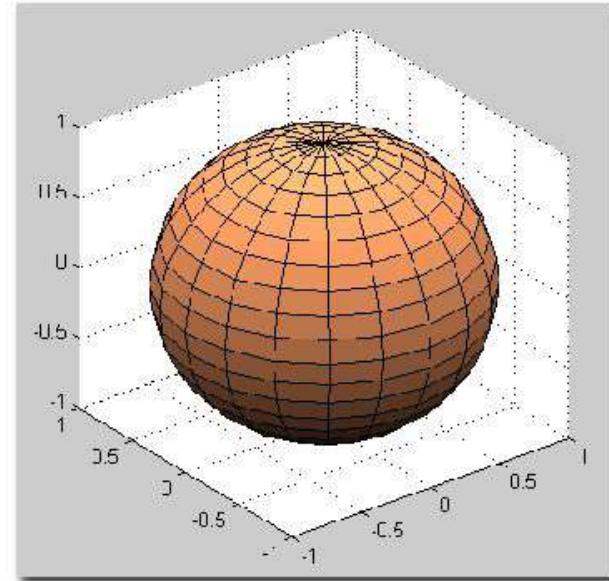
$$s : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



Parametric Surfaces

- Sphere in 3D

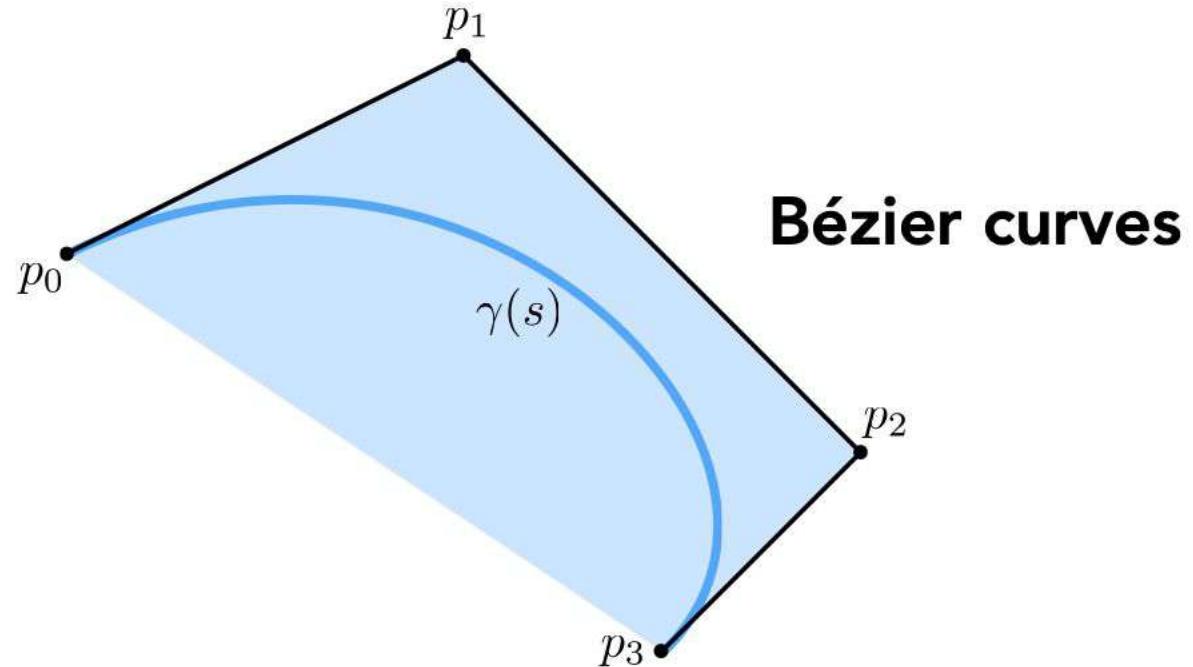
$$s : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



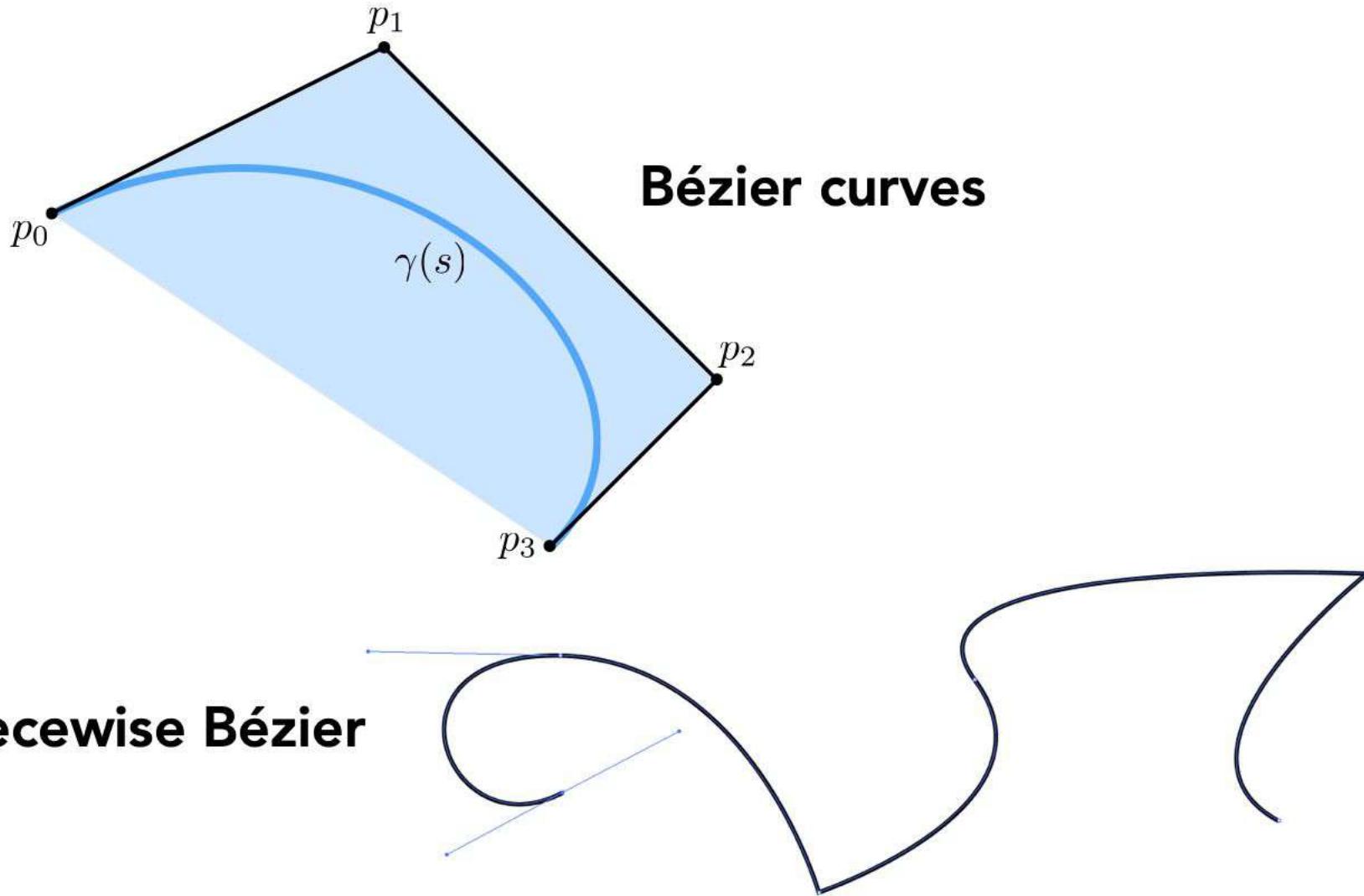
$$s(u, v) = r (\cos(u) \cos(v), \sin(u) \cos(v), \sin(v))$$

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

Bézier Curves

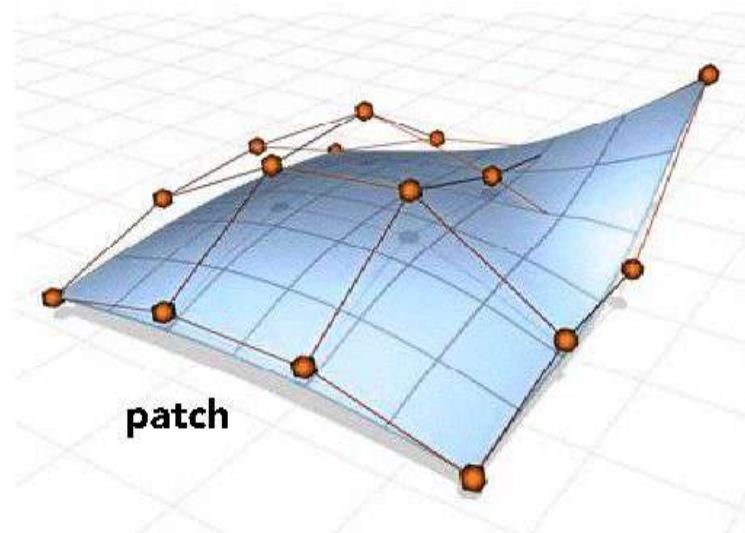


Bézier Curves



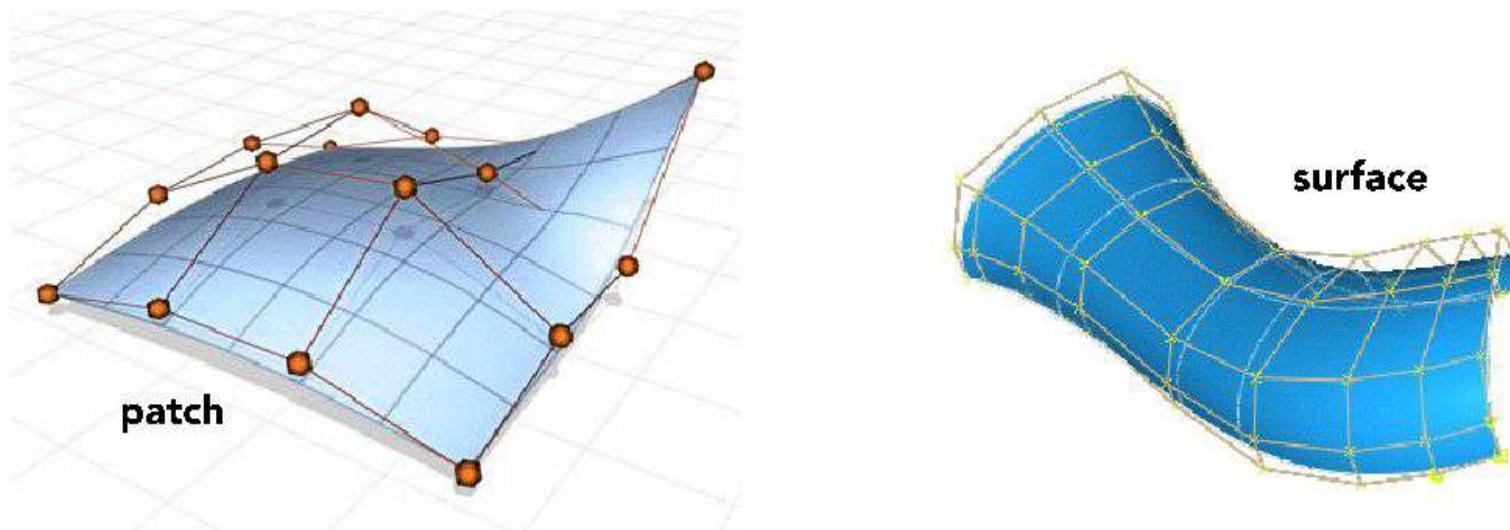
Bézier Surfaces

Use tensor product of Bézier curves to get a patch:



Bézier Surfaces

Use tensor product of Bézier curves to get a patch:

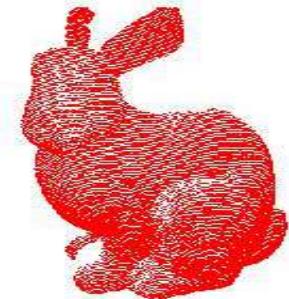


Multiple Bézier patches form a surface

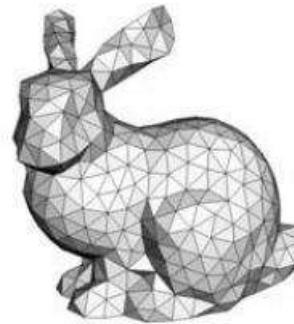
Shape Representations

Non-parametric

Explicit

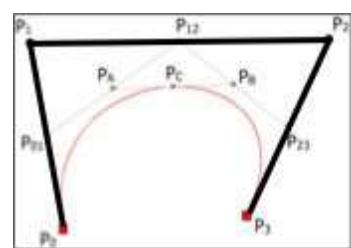


Points

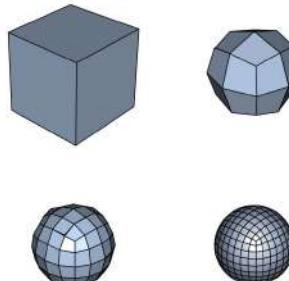


Meshes

Parametric



Splines



Subdivision
Surfaces

“Explicit” Representations of Geometry

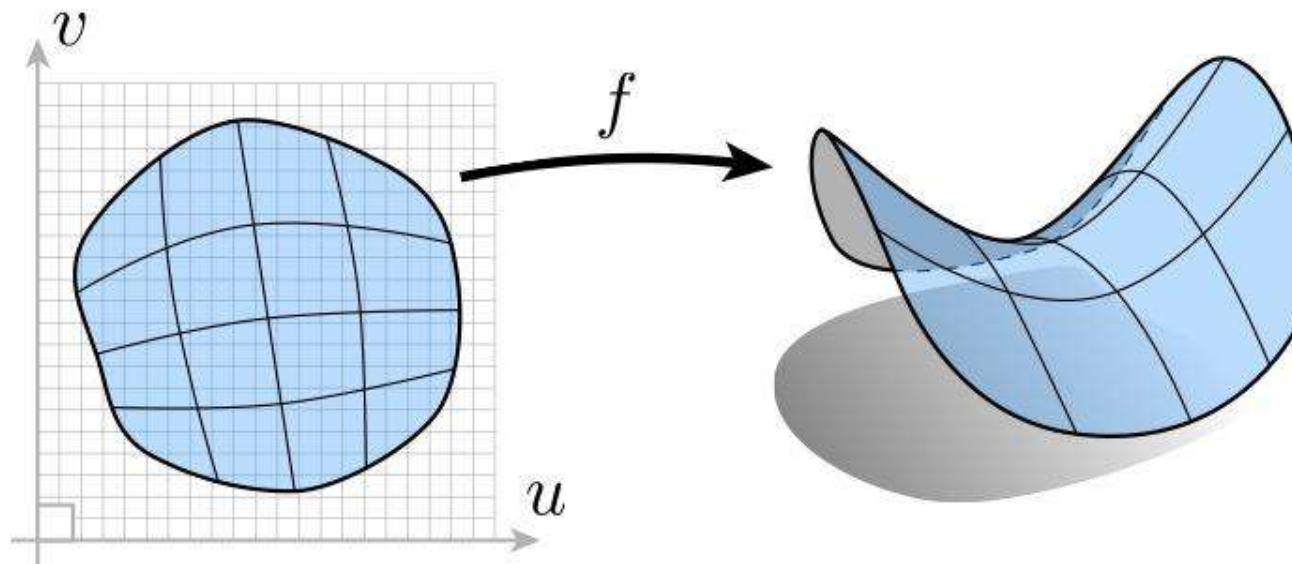
All points are given directly

"Explicit" Representations of Geometry

All points are given directly

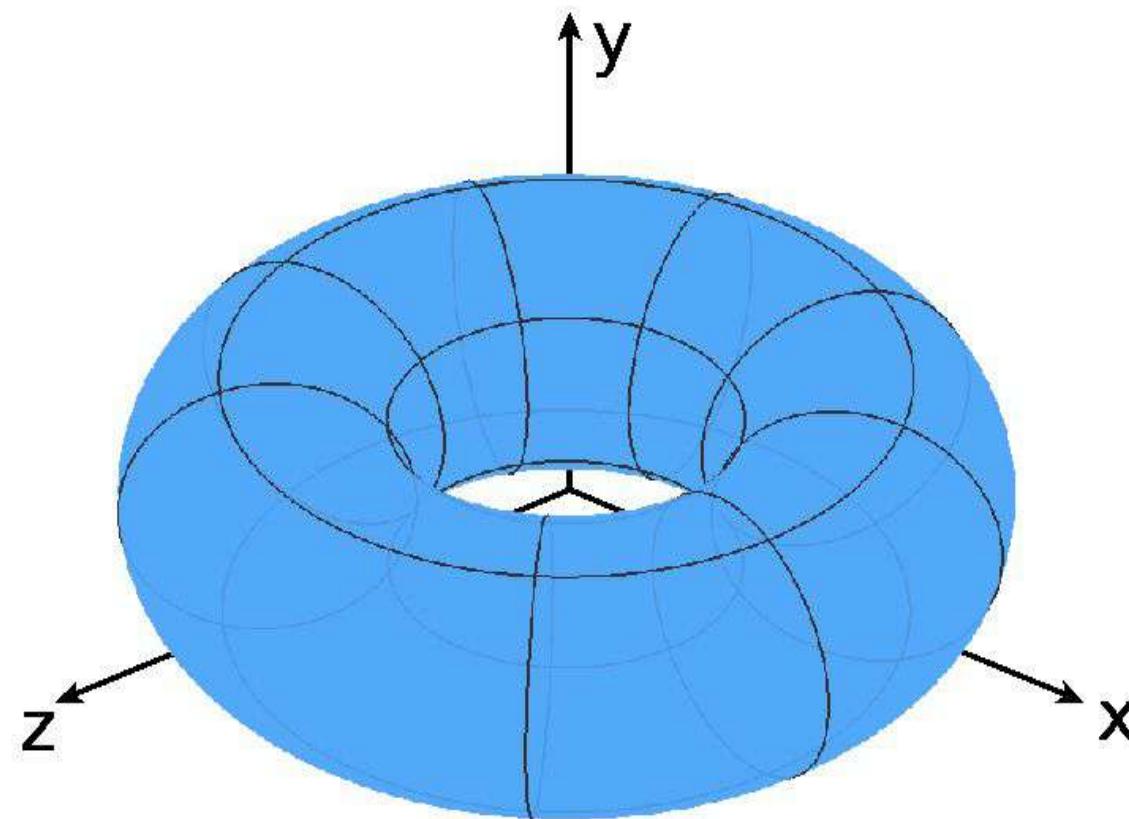
Generally:

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^3; (u, v) \mapsto (x, y, z)$$



Explicit Surface – Sampling Is Easy

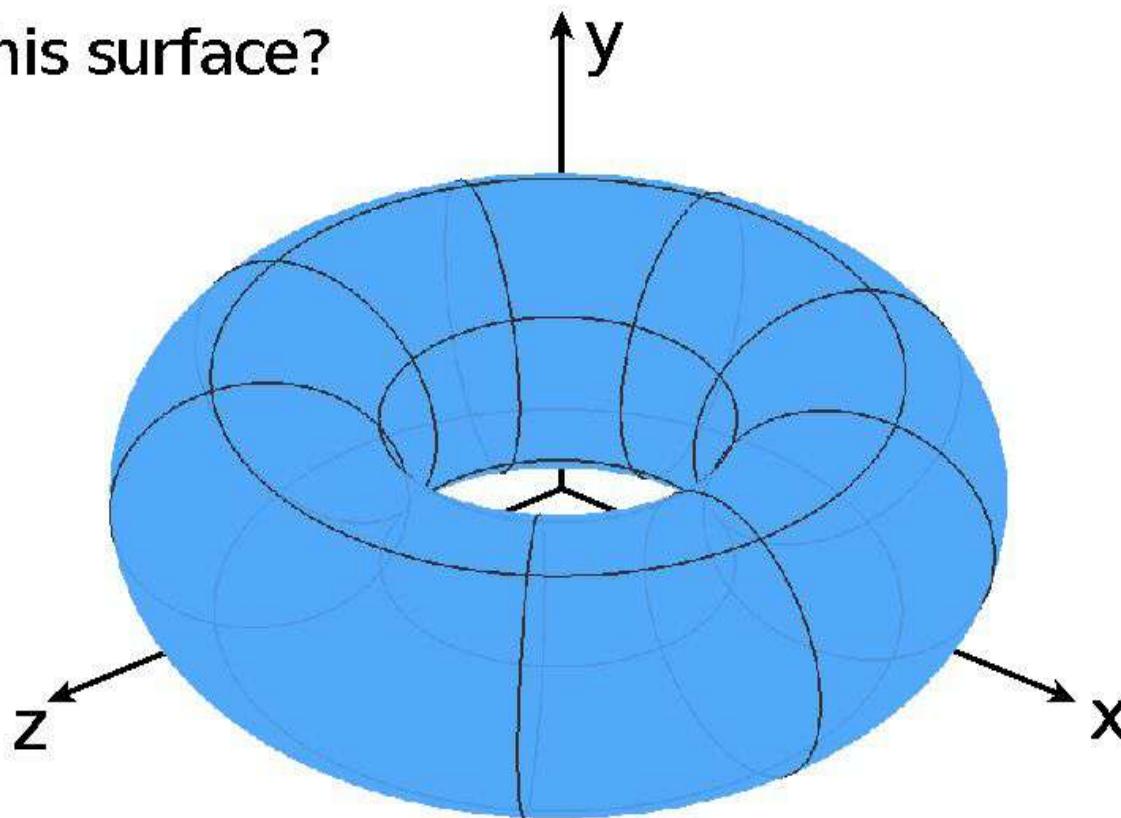
$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$



Explicit Surface – Sampling Is Easy

$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$

What points lie on this surface?

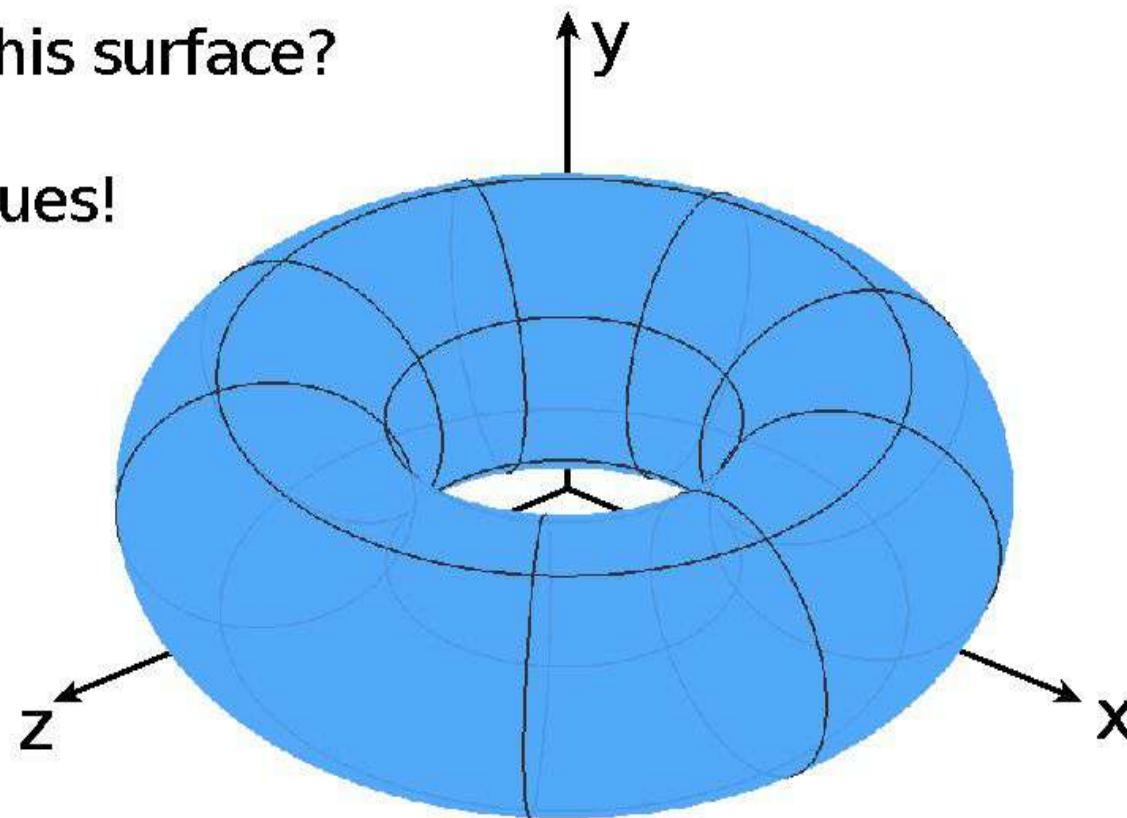


Explicit Surface – Sampling Is Easy

$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$

What points lie on this surface?

Just plug in (u, v) values!

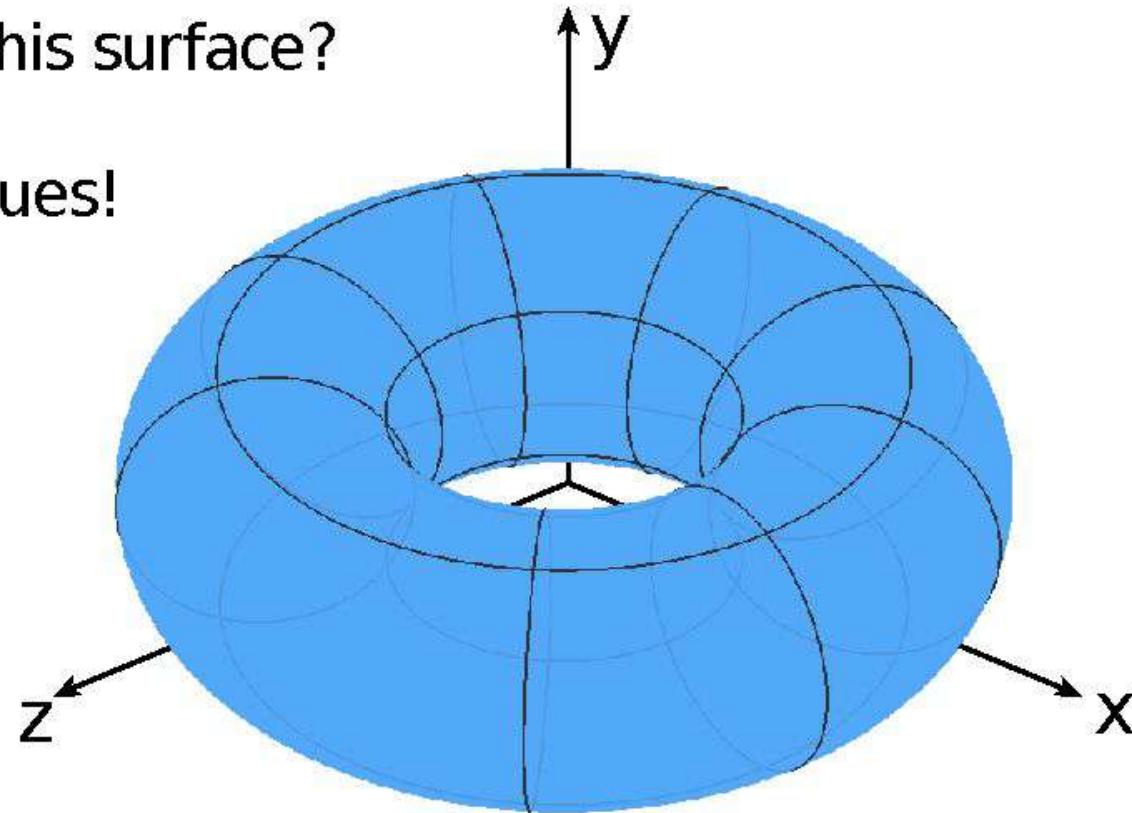


Explicit Surface – Sampling Is Easy

$$f(u, v) = ((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)$$

What points lie on this surface?

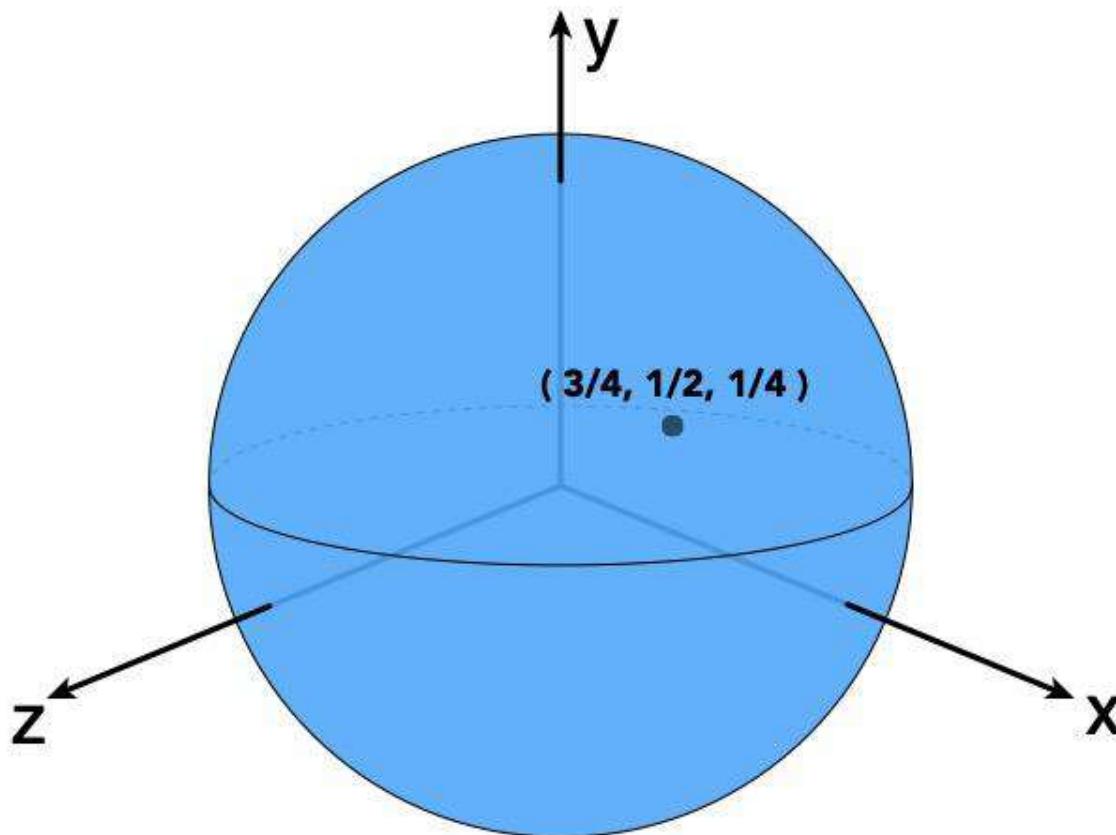
Just plug in (u, v) values!



Explicit representations make some tasks easy

Explicit Surface – Inside/Outside Test Hard

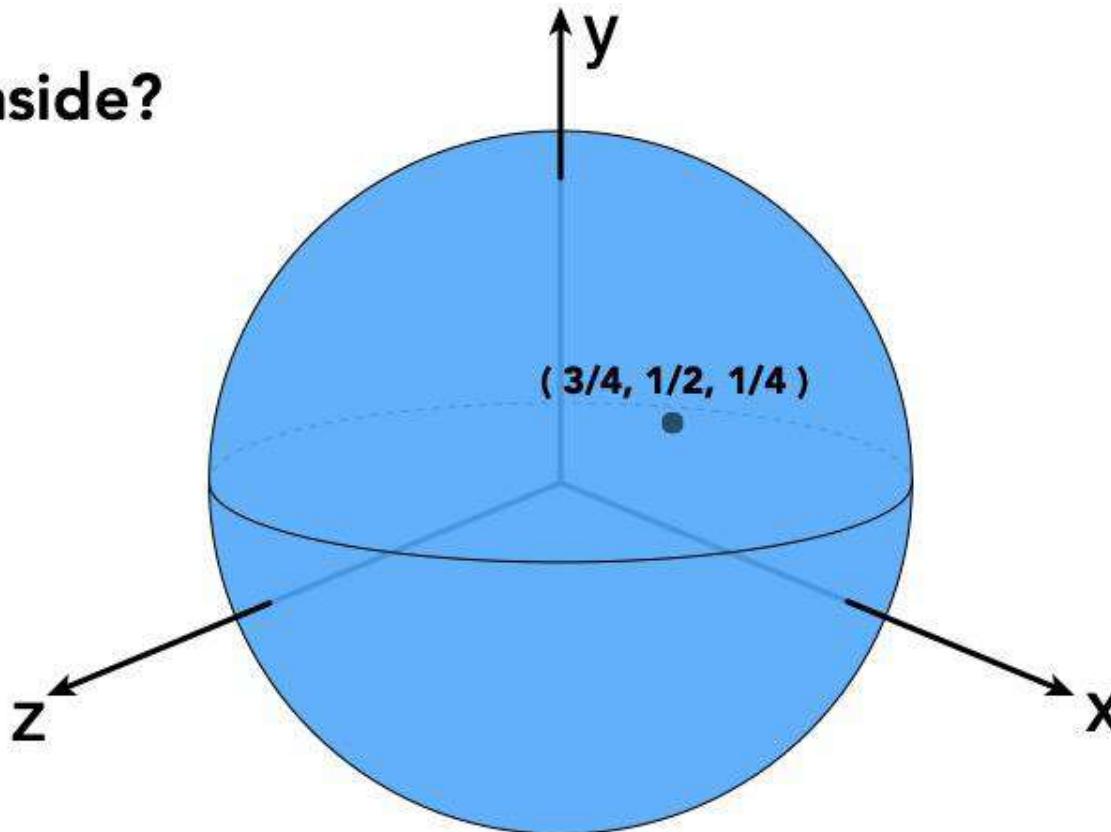
$$f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$$



Explicit Surface – Inside/Outside Test Hard

$$f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$$

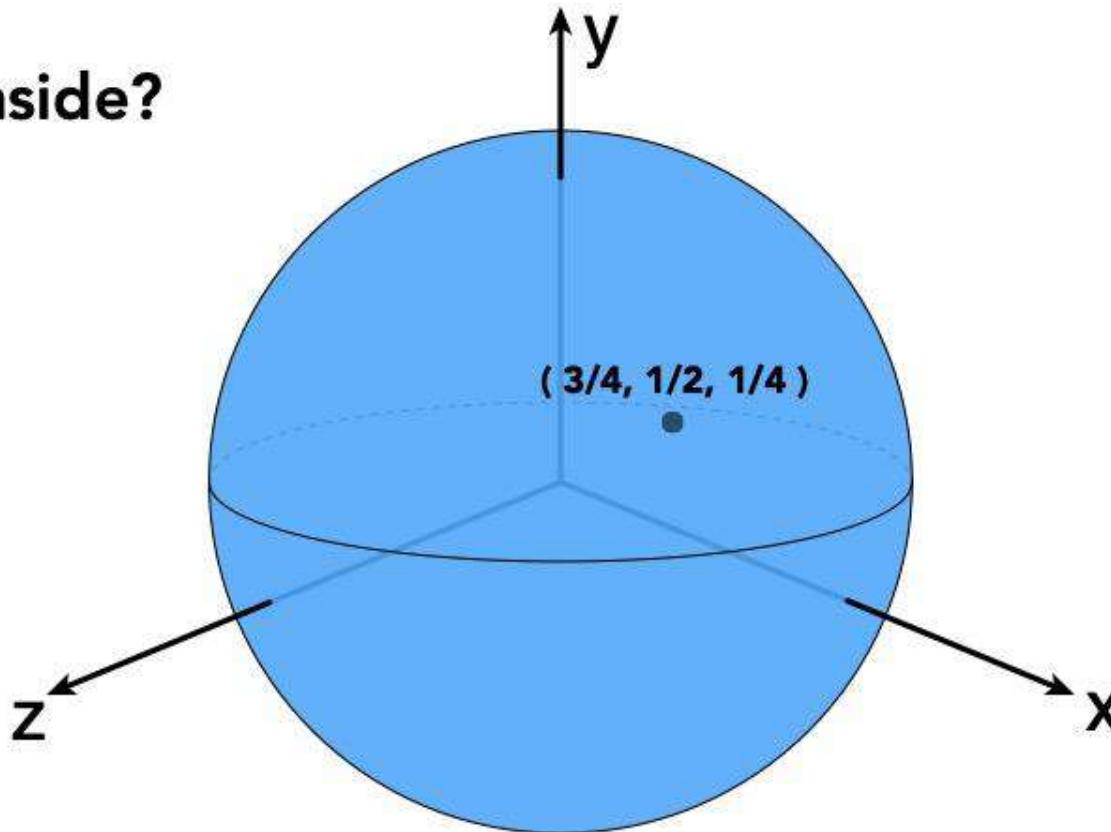
Is $(3/4, 1/2, 1/4)$ inside?



Explicit Surface – Inside/Outside Test Hard

$$f(u, v) = (\cos u \sin v, \sin u \sin v, \cos v)$$

Is $(3/4, 1/2, 1/4)$ inside?



Some tasks are hard with explicit representations

“Implicit” Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship

“Implicit” Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship

E.g. sphere: all points in 3D, where $x^2+y^2+z^2 = 1$

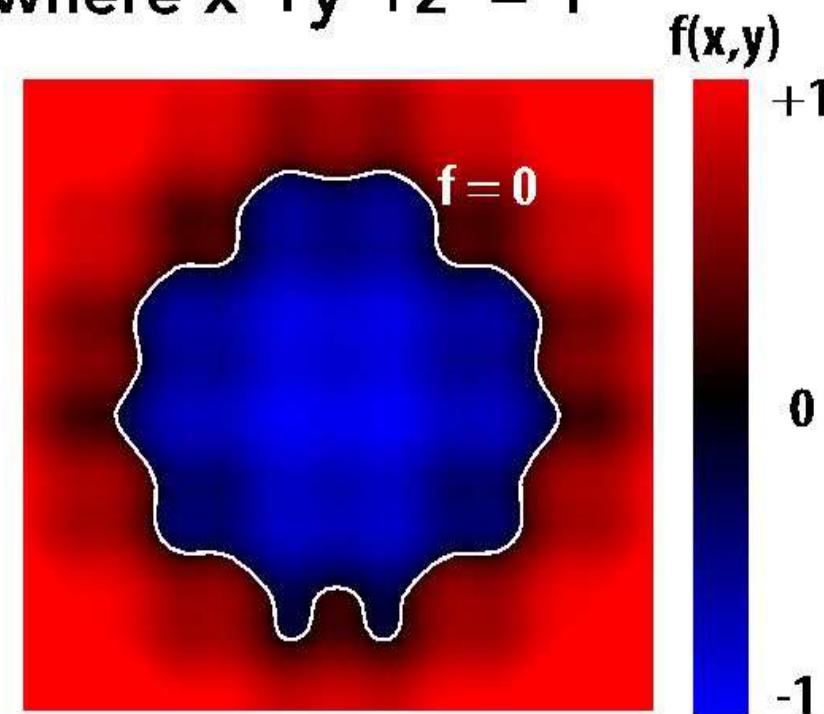
“Implicit” Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship

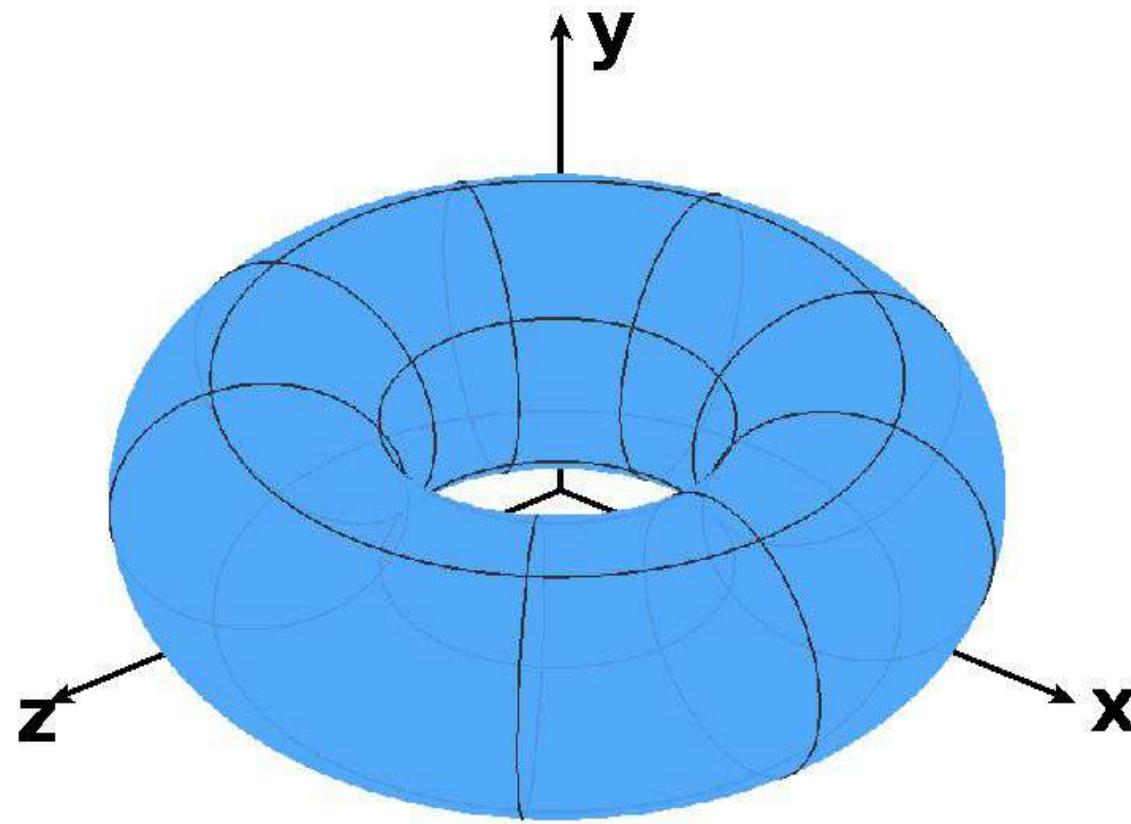
E.g. sphere: all points in 3D, where $x^2+y^2+z^2 = 1$

More generally, $f(x,y,z) = 0$



Implicit Surface – Sampling Can Be Hard

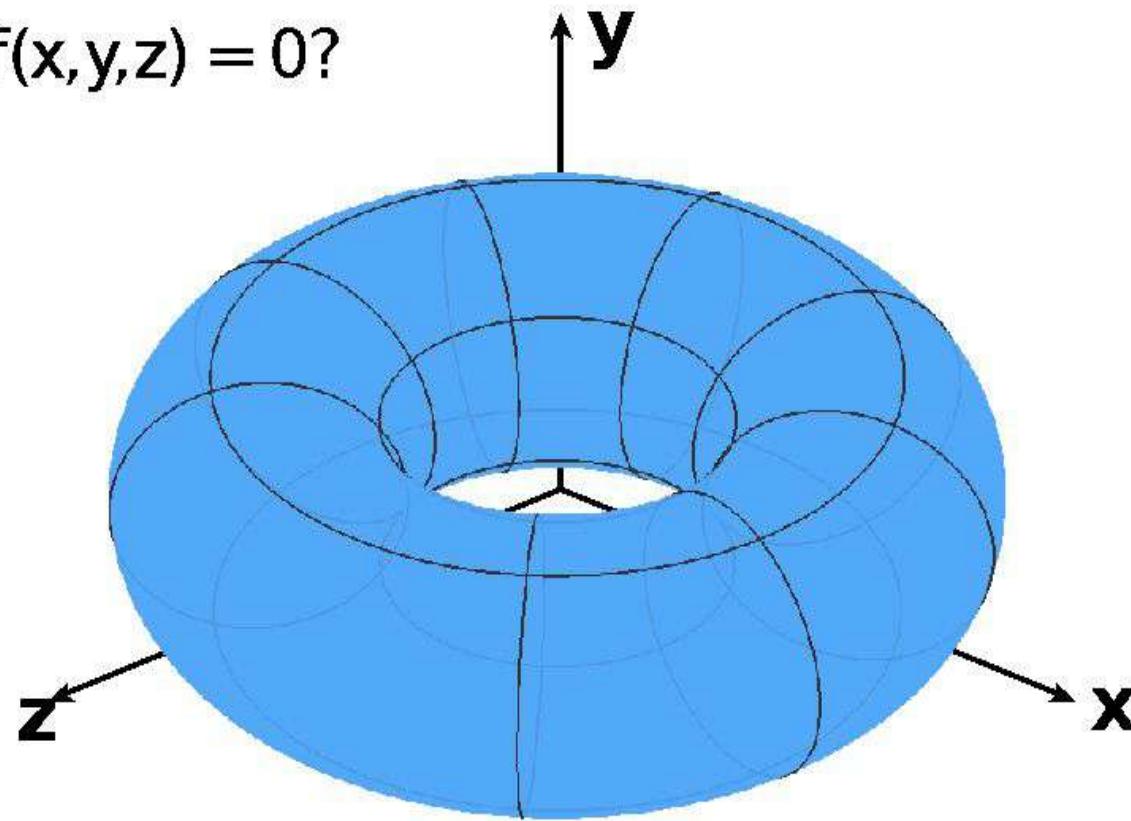
$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$



Implicit Surface – Sampling Can Be Hard

$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$

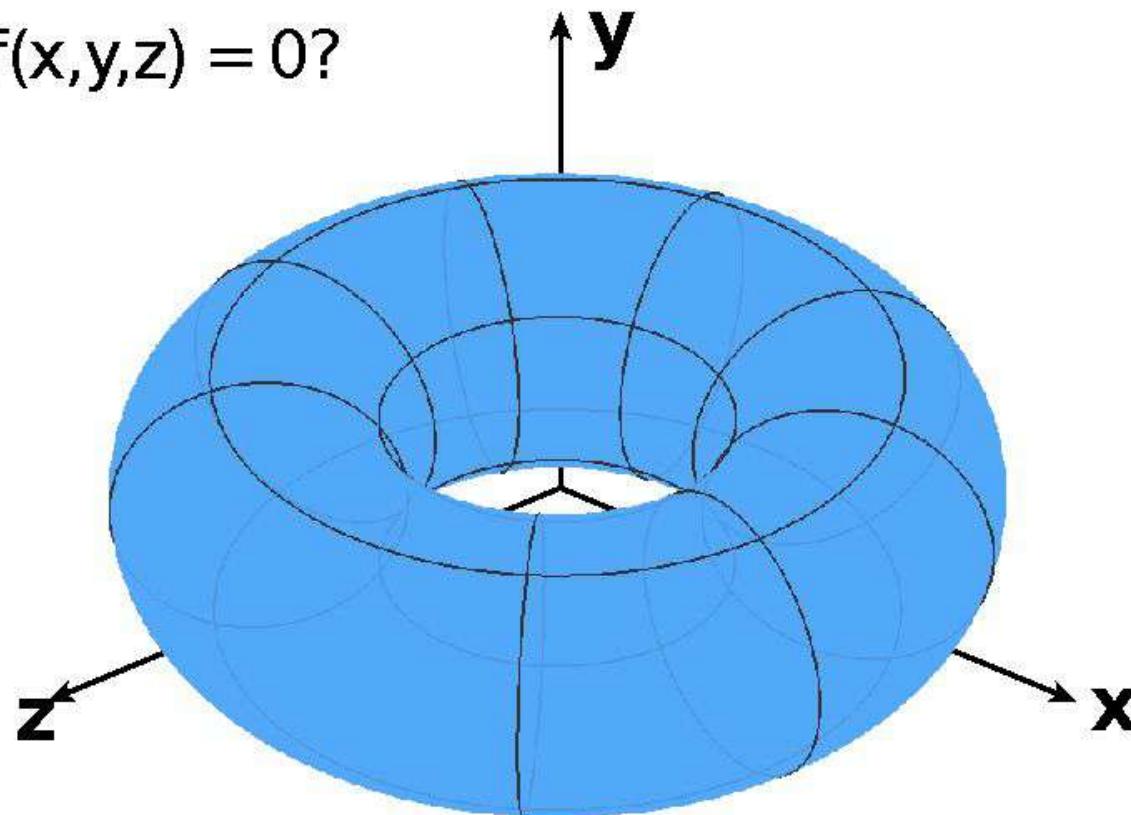
What points lie on $f(x, y, z) = 0$?



Implicit Surface – Sampling Can Be Hard

$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$

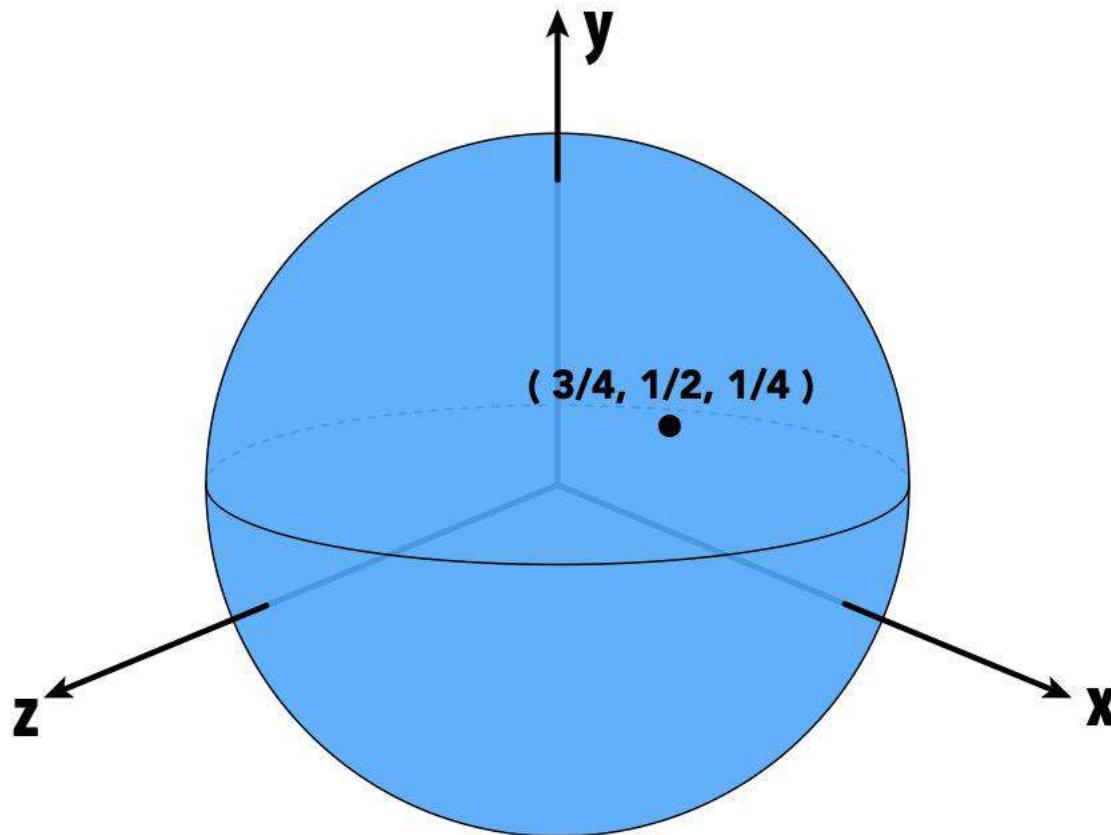
What points lie on $f(x, y, z) = 0$?



Some tasks are hard with implicit representations

Implicit Surface – Inside/Outside Tests Easy

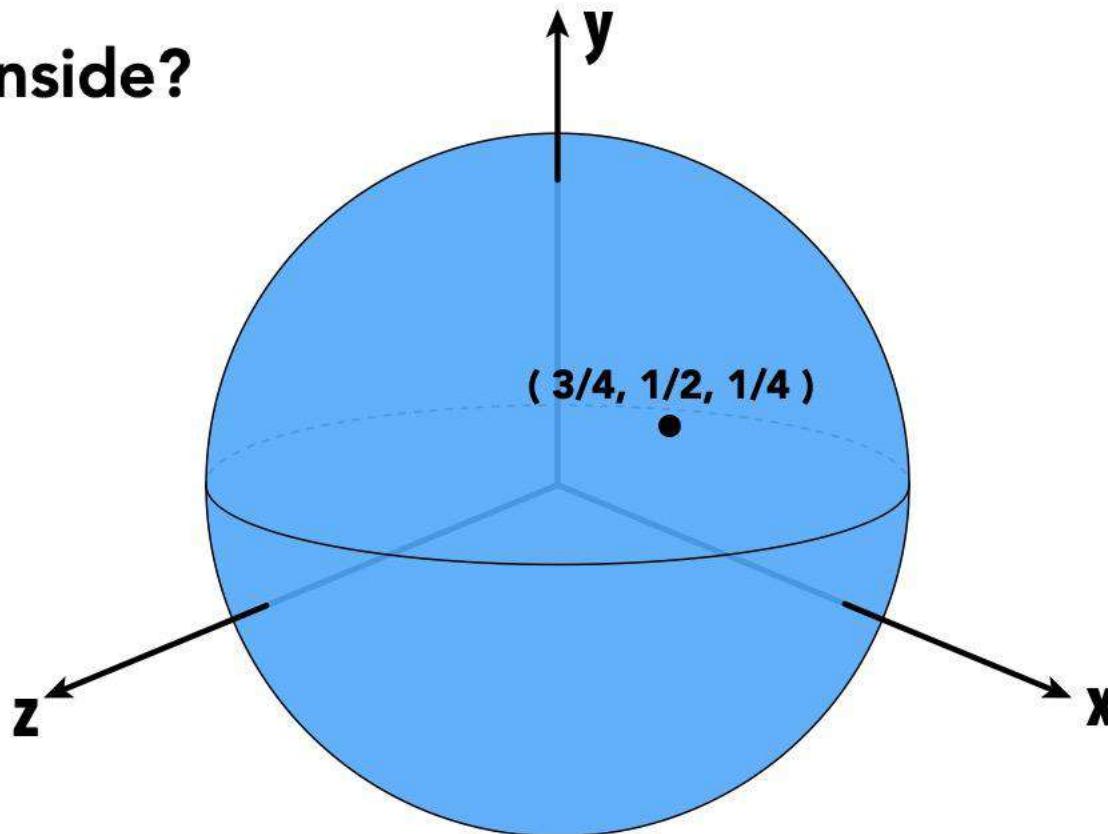
$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$



Implicit Surface – Inside/Outside Tests Easy

$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

Is $(3/4, 1/2, 1/4)$ inside?



Implicit Surface – Inside/Outside Tests Easy

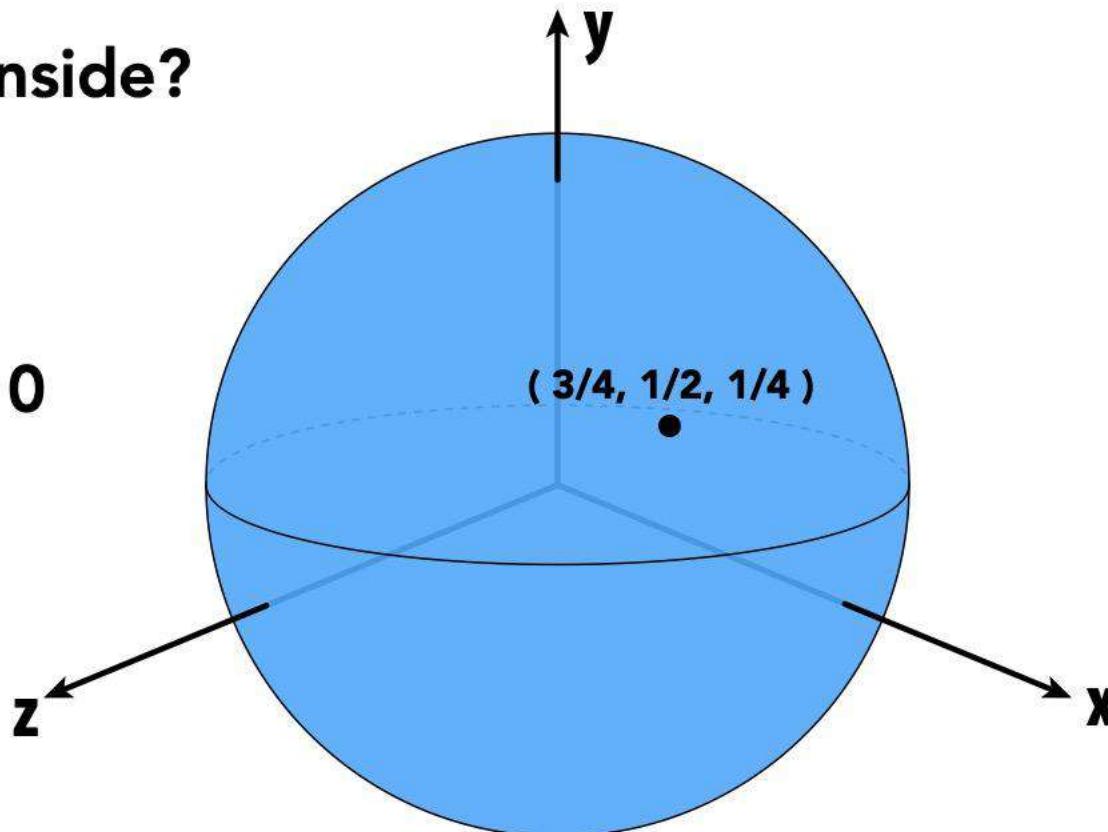
$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

Is $(3/4, 1/2, 1/4)$ inside?

Just plug it in:

$$f(x, y, z) = -1/8 < 0$$

Yes, inside.



Implicit Surface – Inside/Outside Tests Easy

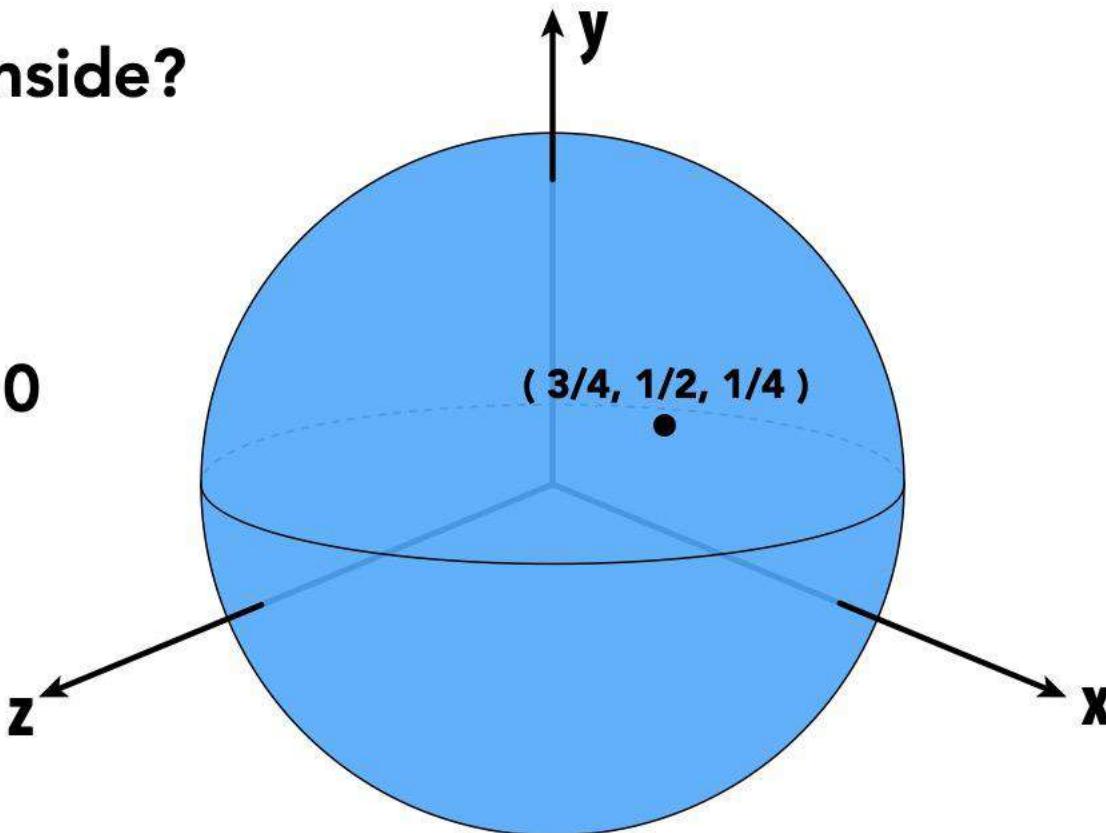
$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

Is $(3/4, 1/2, 1/4)$ inside?

Just plug it in:

$$f(x, y, z) = -1/8 < 0$$

Yes, inside.



Implicit representations make some tasks easy

Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in x, y, z



$$x^2 + y^2 + z^2 = 1$$

Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in x, y, z



$$x^2 + y^2 + z^2 = 1$$



$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$



$$(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 =$$

$$x^2 z^3 + \frac{9y^2 z^3}{80}$$

Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in x , y , z



$$x^2 + y^2 + z^2 = 1$$

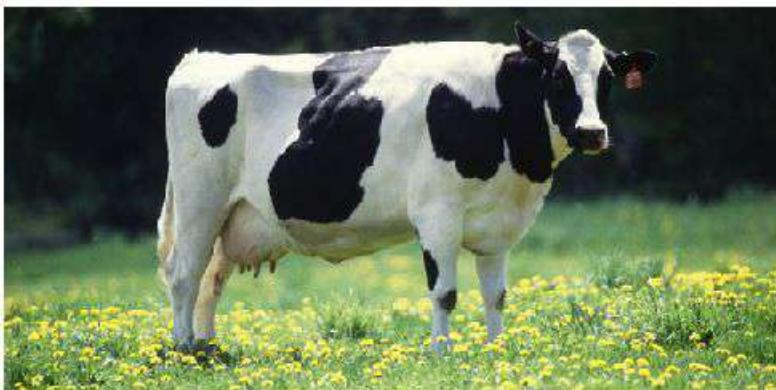


$$(R - \sqrt{x^2 + y^2})^2 + z^2 = r^2$$



$$(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 =$$

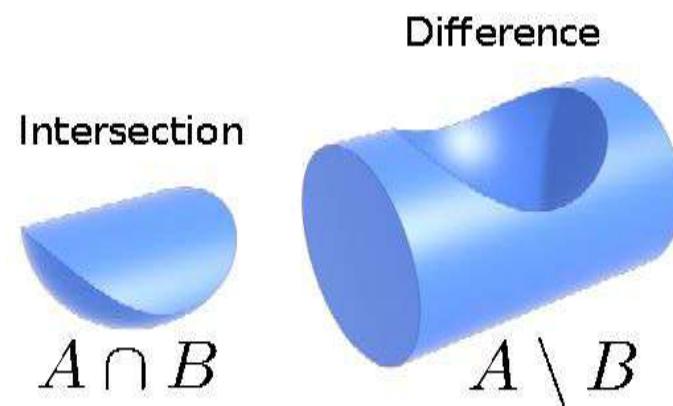
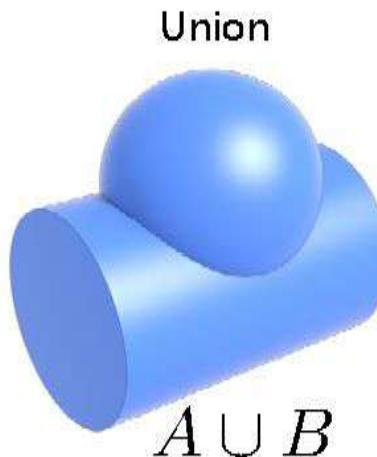
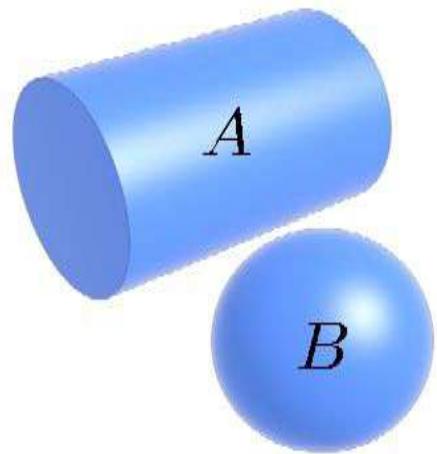
$$x^2 z^3 + \frac{9y^2 z^3}{80}$$



More complex shapes?

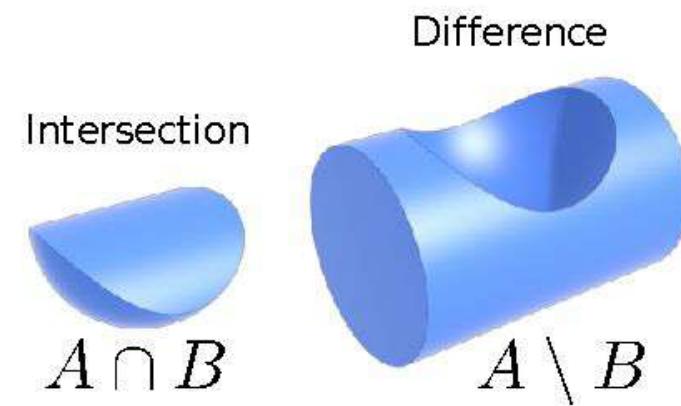
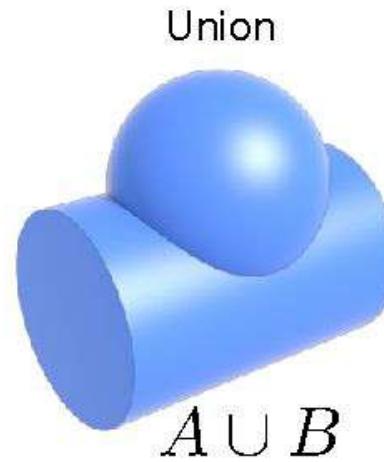
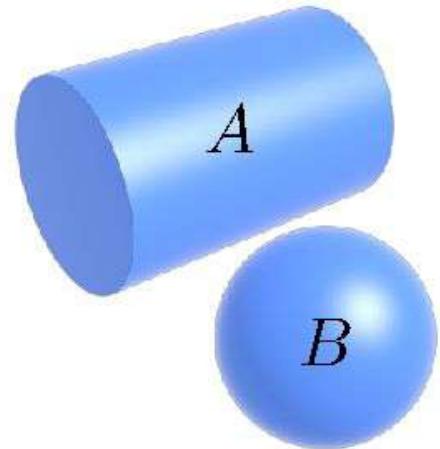
Constructive Solid Geometry (Implicit)

Combine implicit geometry via Boolean operations

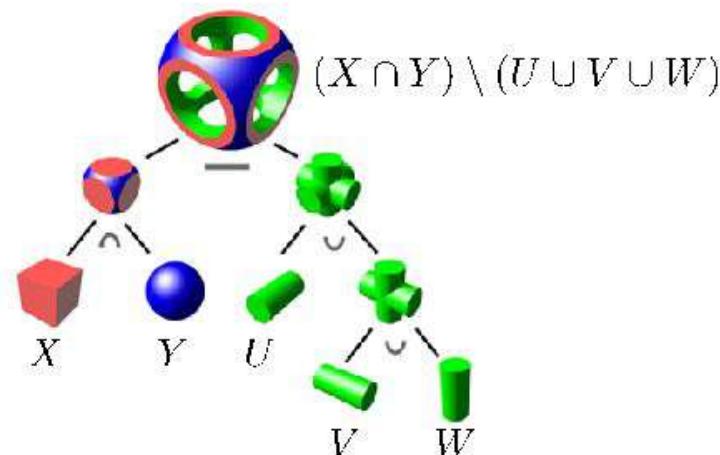


Constructive Solid Geometry (Implicit)

Combine implicit geometry via Boolean operations



Boolean expressions:

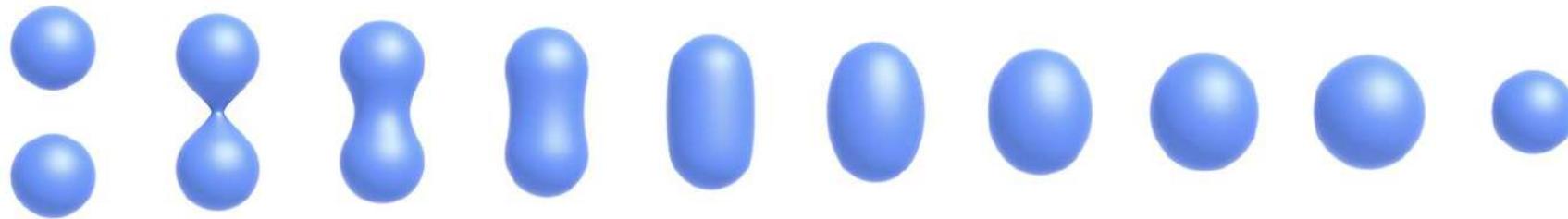


Distance Functions (Implicit)

Instead of Booleans, gradually blend surfaces together using

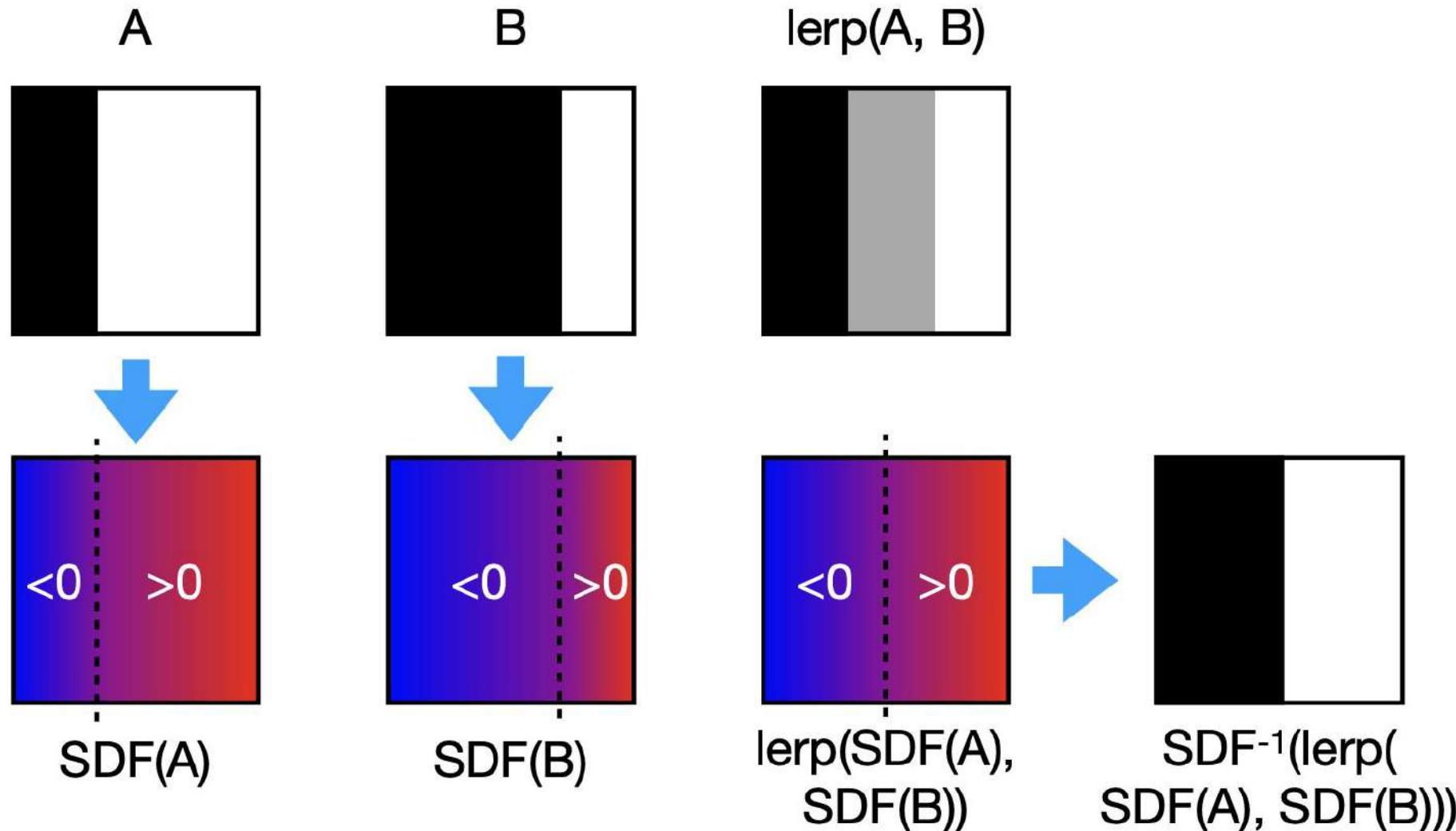
Distance functions:

giving minimum distance (could be **signed** distance)
from anywhere to object



Distance Functions (Implicit)

An Example: Blending (linear interp.) a moving boundary

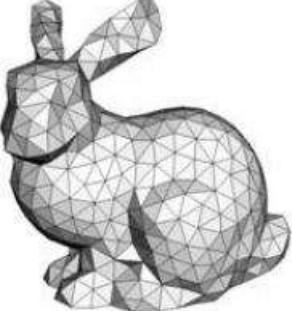
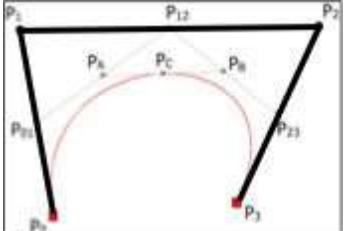
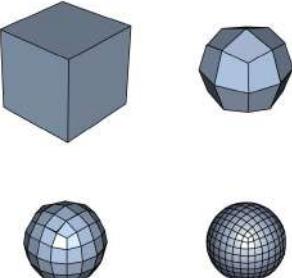
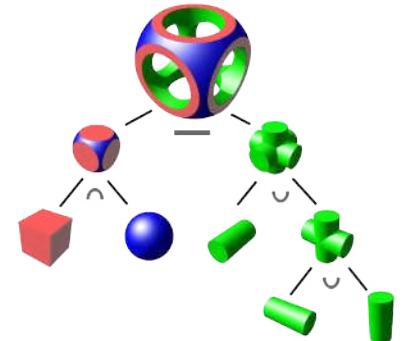


Scene of Pure Distance Functions (Not Easy!)



See <http://iquilezles.org/www/material/nvscene2008/nvscene2008.htm>

Shape Representations

	Explicit	Implicit
Non-parametric	 Points	 Meshes
Parametric	 Splines	 $x^2 + y^2 + z^2 = 1$ Algebraic Surfaces
	 Subdivision Surfaces	 Constructive Solid Geometry

Level Set Methods (Implicit)

Implicit surfaces have some nice features (e.g., merging/splitting)

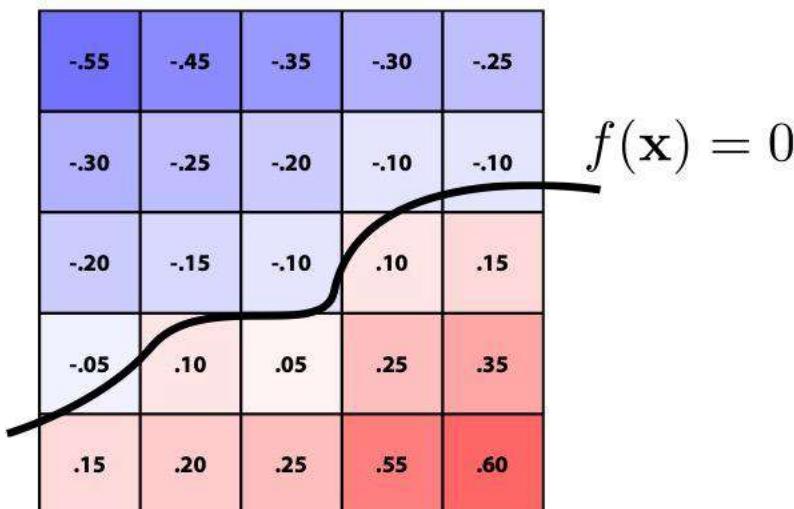
But, hard to describe complex shapes in closed form

Level Set Methods (Implicit)

Implicit surfaces have some nice features (e.g., merging/splitting)

But, hard to describe complex shapes in closed form

Alternative: store a grid of values approximating function

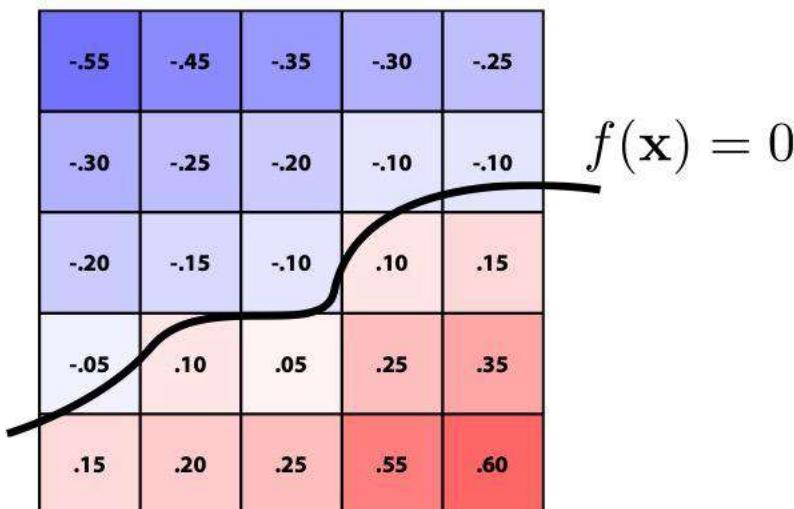


Level Set Methods (Implicit)

Implicit surfaces have some nice features (e.g., merging/splitting)

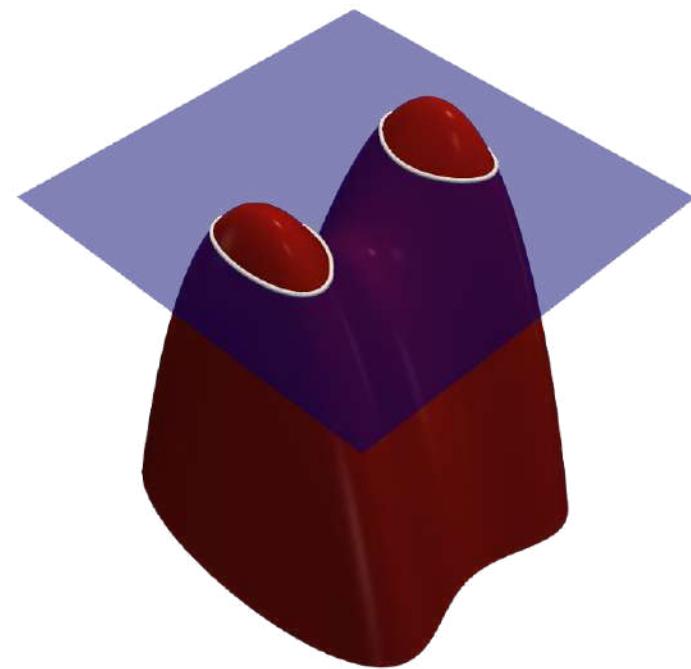
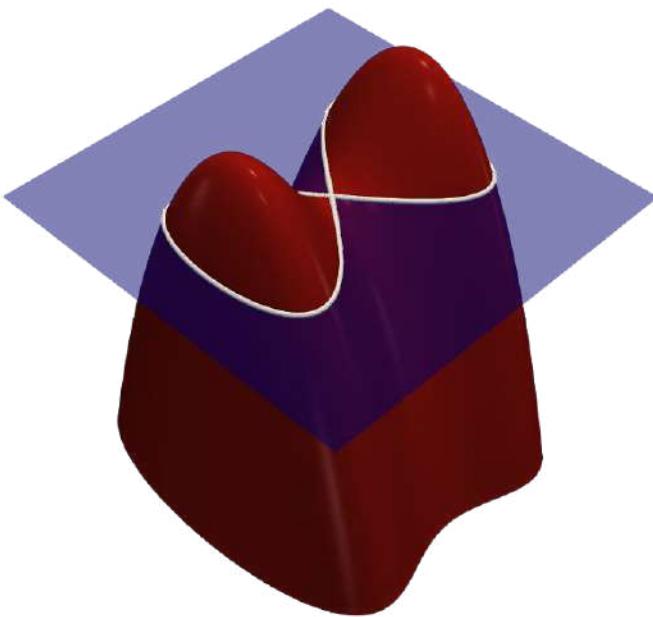
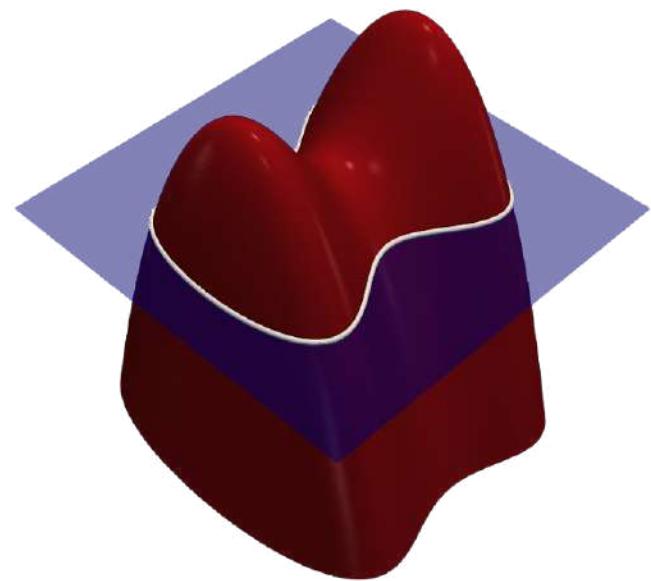
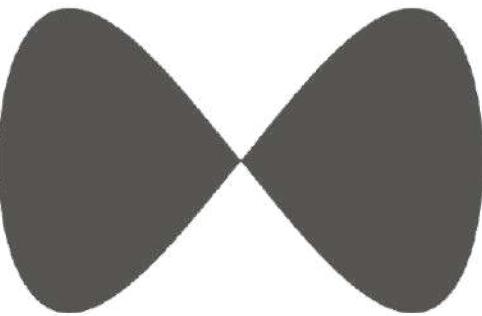
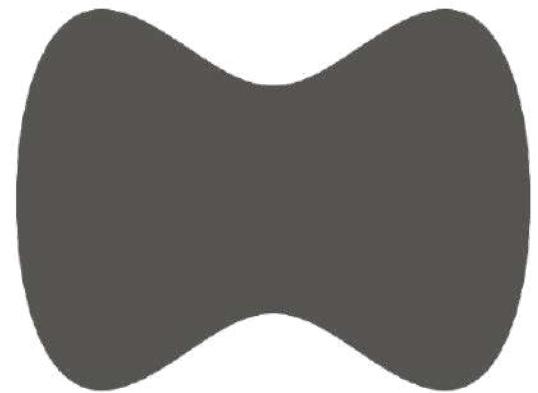
But, hard to describe complex shapes in closed form

Alternative: store a grid of values approximating function



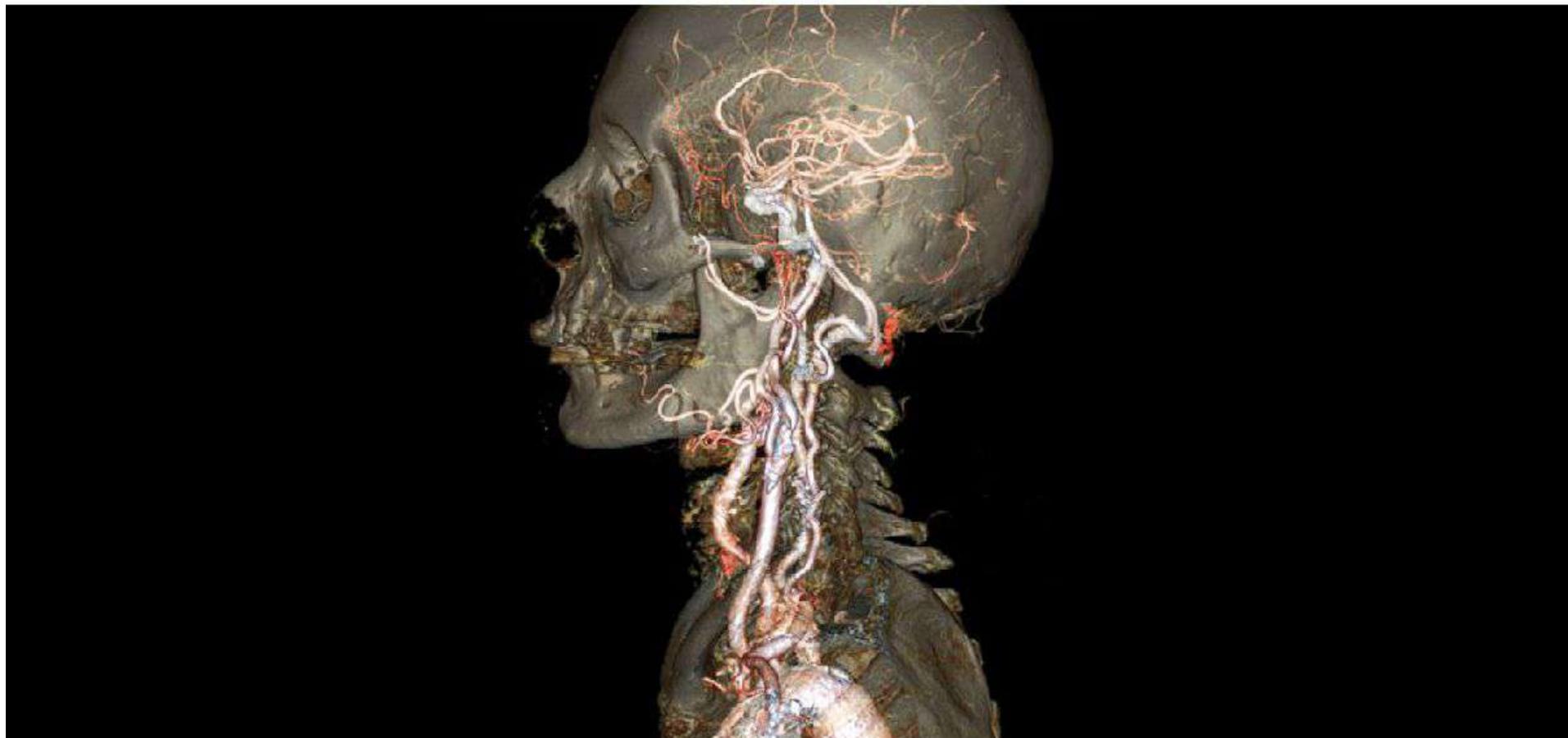
Surface is found where interpolated values equal zero

Provides much more explicit control over shape (like a texture)



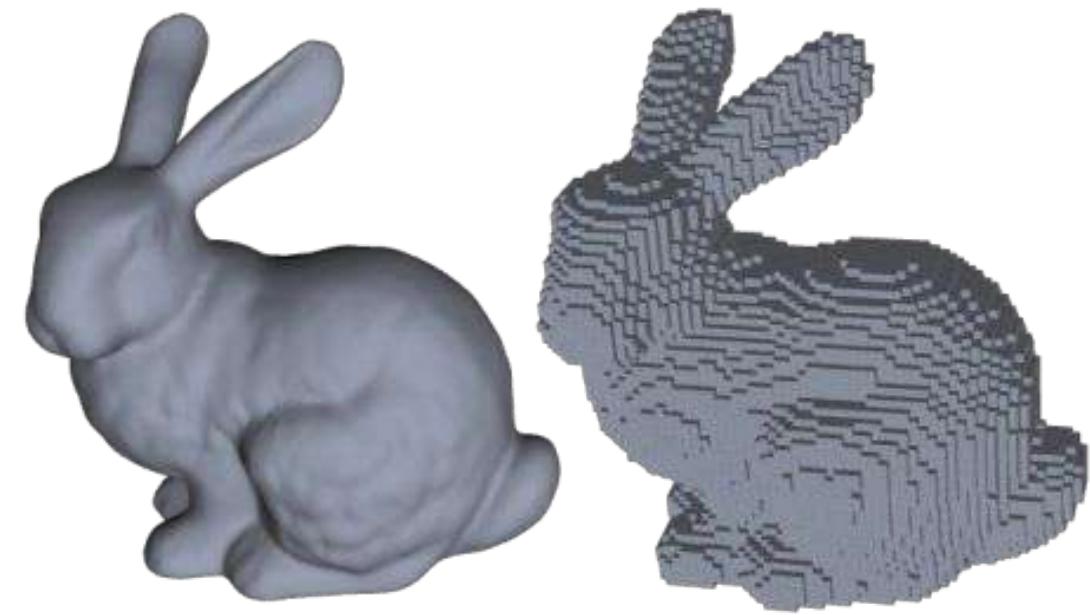
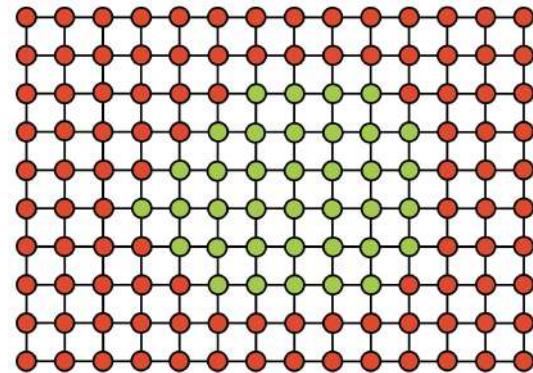
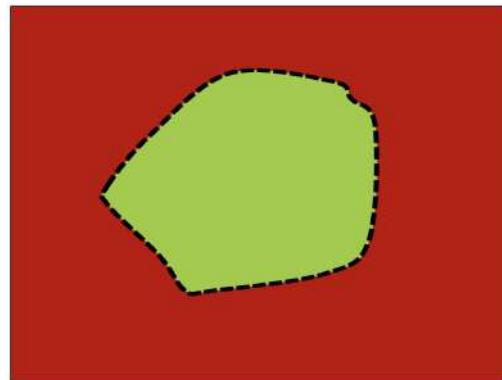
Level Sets from Medical Data (CT, MRI, etc.)

Level sets encode, e.g., constant tissue density



Related Representation: Voxels

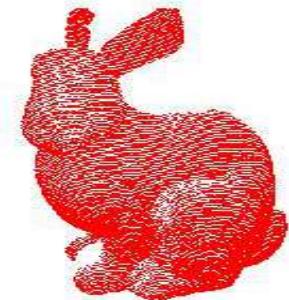
- Binary thresholding the volumetric grid



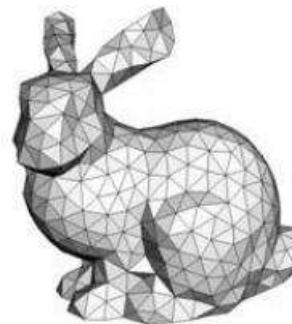
Shape Representations

Non-parametric

Explicit



Points

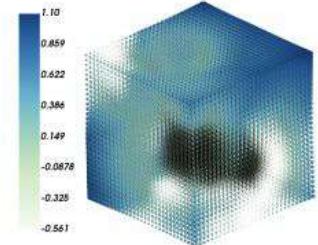


Meshes

Implicit (Eulerian)

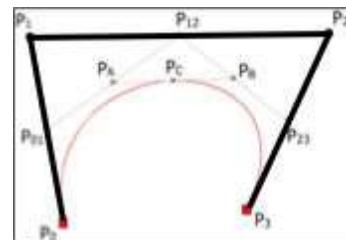


Voxels

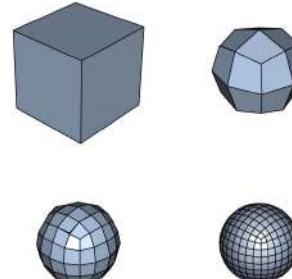


Level Sets

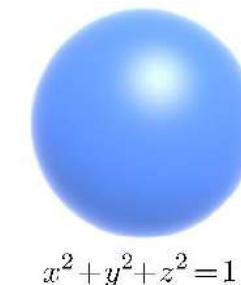
Parametric



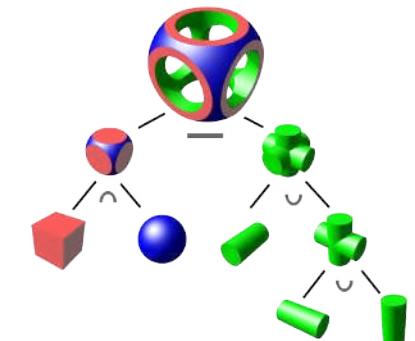
Splines



Subdivision
Surfaces



Algebraic
Surfaces



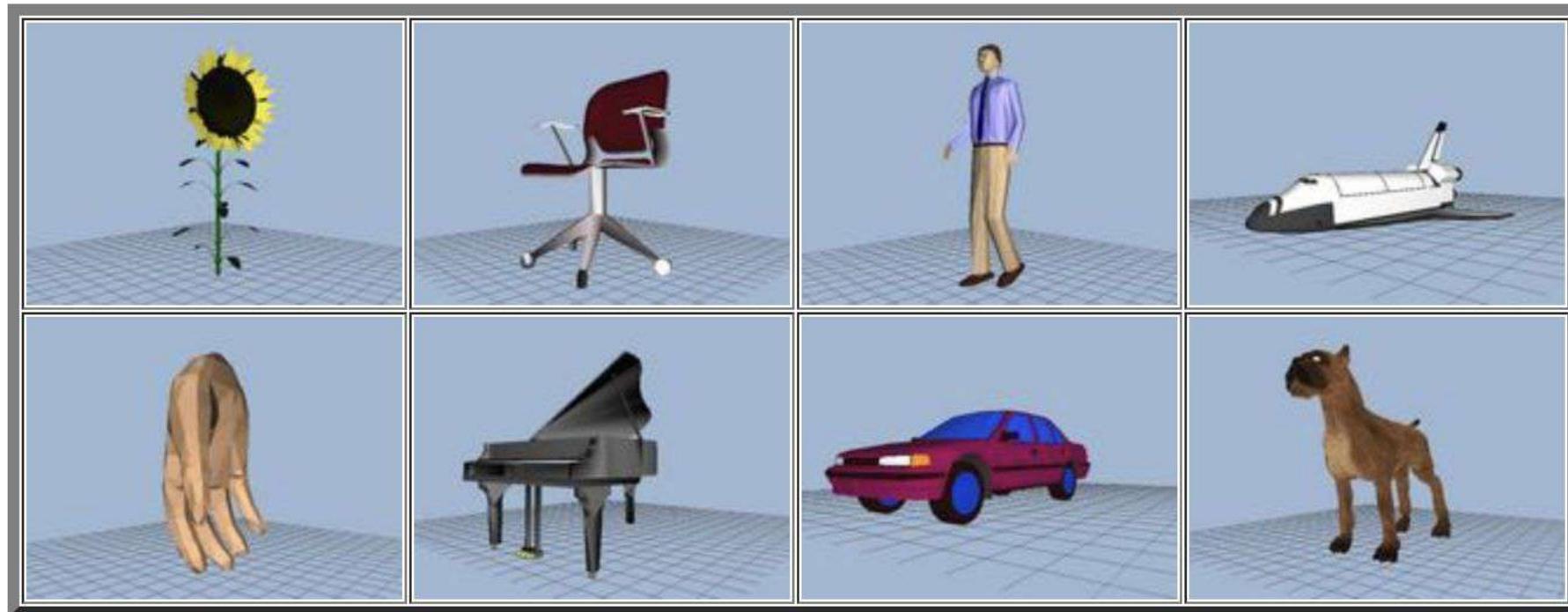
Constructive
Solid
Geometry

AI + Shapes

- In 3D ...

Princeton Shape Benchmark

- 1814 Models
- 182 Categories



Datasets Prior to 2014

Benchmarks	Types	# models	# classes	Avg # models per class
SHREC14LSGTB	Generic	8,987	171	53
PSB	Generic	907+907 (train+test)	90+92 (train+test)	10+10 (train+test)
SHREC12GTB	Generic	1200	60	20
TSB	Generic	10,000	352	28
CCCC	Generic	473	55	9
WMB	Watertight (articulated)	400	20	20
MSB	Articulated	457	19	24
BAB	Architecture	2257	183+180 (function+form)	12+13 (function+form)
ESB	CAD	867	45	19

Table 1. Source datasets from SHREC 2014: *Princeton Shape Benchmark (PSB)* [27], *SHREC 2012 generic Shape Benchmark (SHREC12GTB)* [16], *Toyohashi Shape Benchmark (TSB)* [29], *Konstanz 3D Model Benchmark (CCCC)* [32], *Watertight Model Benchmark (WMB)* [31], *McGill 3D Shape Benchmark (MSB)* [37], *Bonn Architecture Benchmark (BAB)* [33], *Purdue Engineering Shape Benchmark (ESB)* [9].

Datasets for 3D Objects

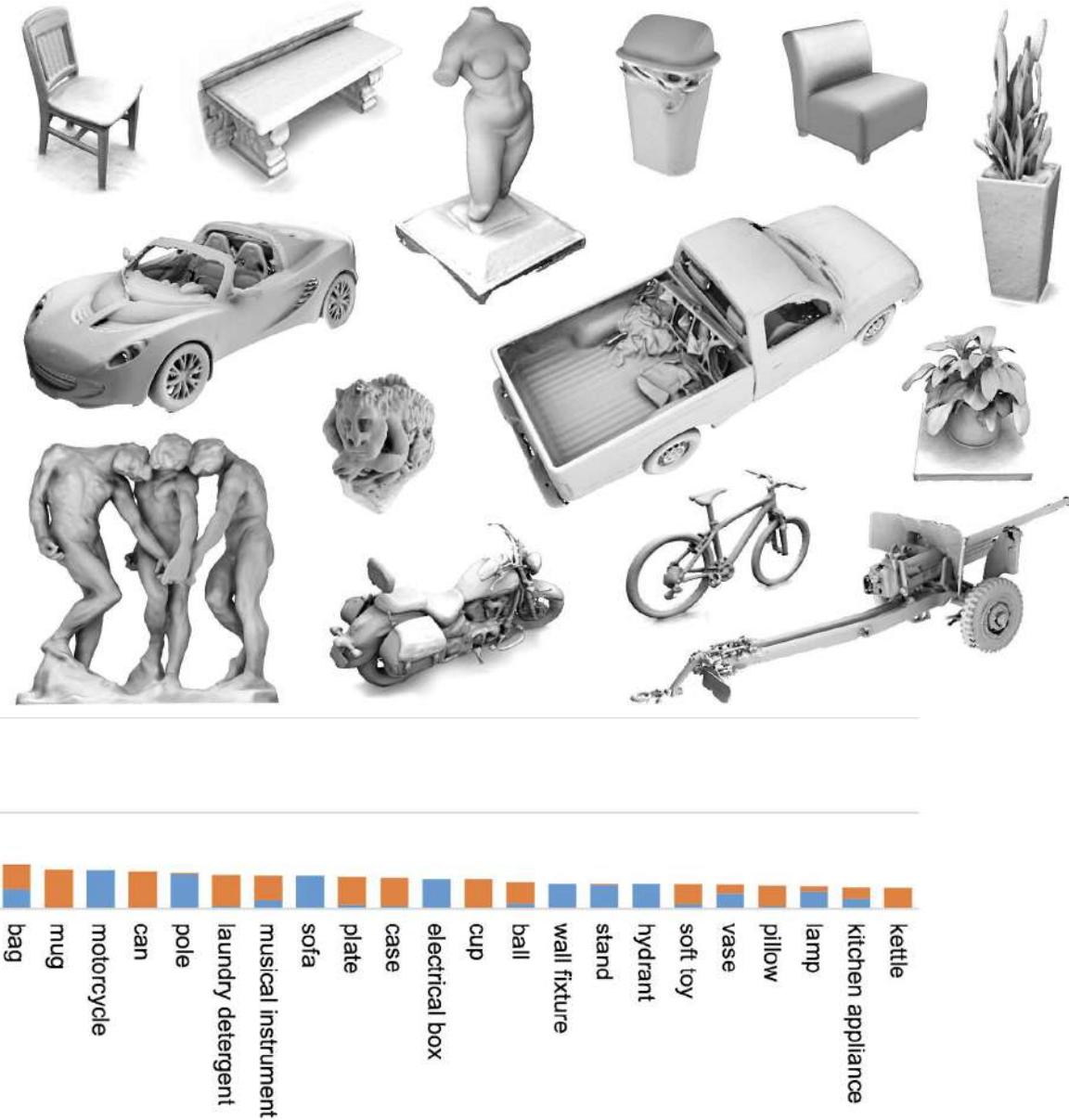
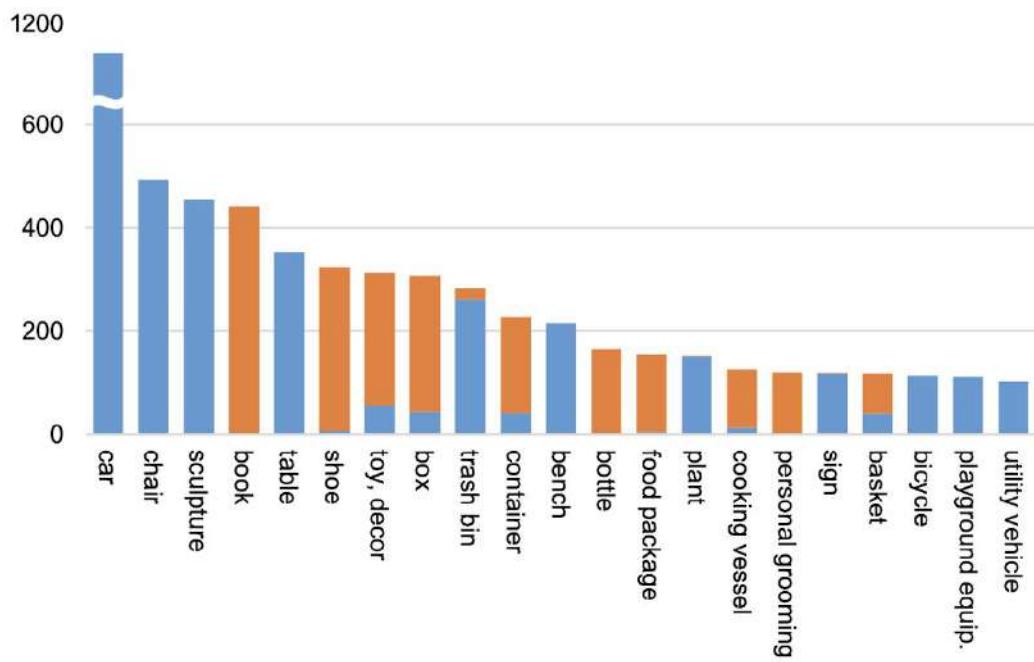
- Large-scale Synthetic Objects: ShapeNet, 3M models
- ModelNet: absorbed by ShapeNet
- ShapeNetCore: 51.3K models in 55 categories



Chang et al. ShapeNet. arXiv 2015
Wu et al. 3D ShapeNets. CVPR 2015

Object Scan

- 10,933 RGBD scans
 - 441 models



Pascal 3D+

- Retrieve a nearest-neighbor 3D model for objects in real images
- 8,505 PASCAL images (13,898 instances) + 22,394 ImageNet images
- 12 rigid categories, 3,000+ instances per category on average



Pix3D

- 10,069 images
- 395 shapes (IKEA furniture + 3D scan)



Link to WordNet Taxonomy Alignment+Symmetry



Swivel chair: a chair that swivels on its base

Hypernyms: chair > seat > furniture > ...

Part meronyms: backrest, seat, base

Sister terms: armchair, barber chair, ...

ImageNet



Swivel chair

Plane of Symmetry



Part Hierarchy

Backrest



Dim: 50 x 45 x 5 cm
Material: foam, fabric
Mass: 5 Kg
Function: support

Seat



Base



Leg



Wheel

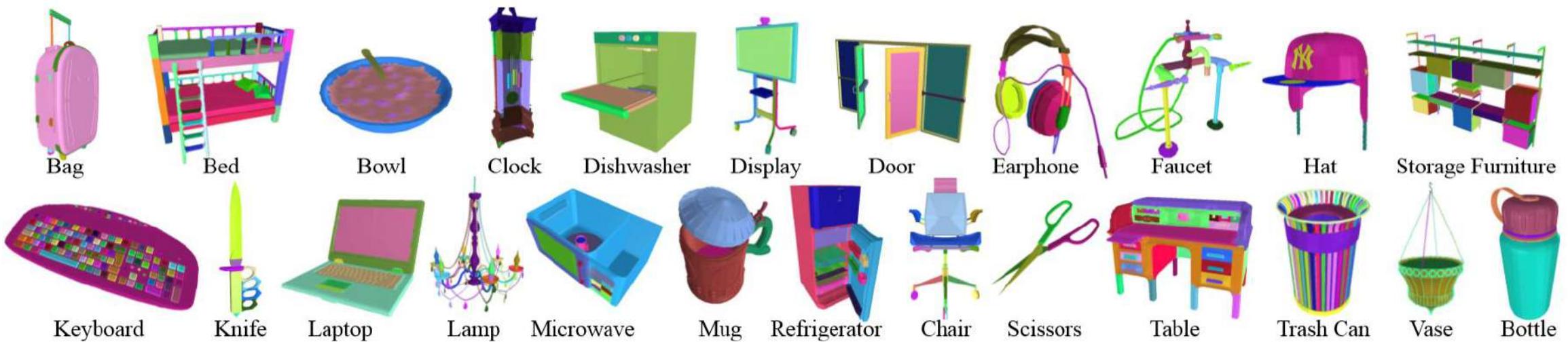


Figure from the ShapeNet paper, Chang et al. arXiv 2015

Datasets for 3D Object Parts

Fine-grained Parts: PartNet

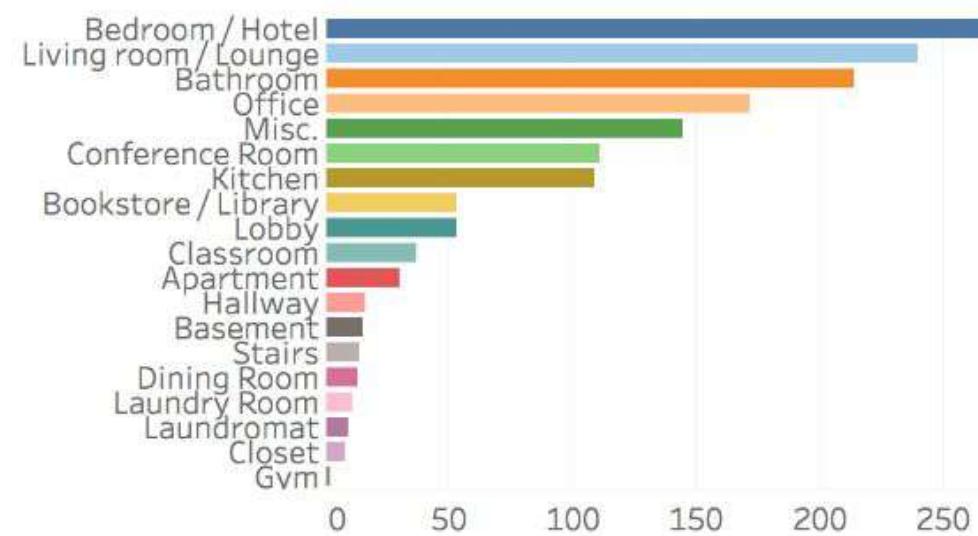
- Fine-grained (+mobility)
- Instance-level
- Hierarchical



Datasets for Indoor 3D Scenes

Large-scale Scanned Real Scenes: ScanNet

- 2.5M Views in 1,500 RGBD scans
- 3D camera poses
- Surface reconstructions
- Instance-level semantic segmentations



Physical Interaction with Articulated Objects

**300+ door
annotations**

**support
articulated
objects**

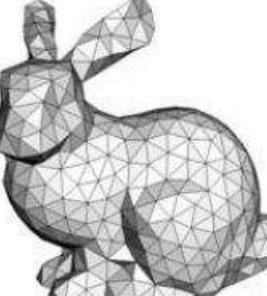
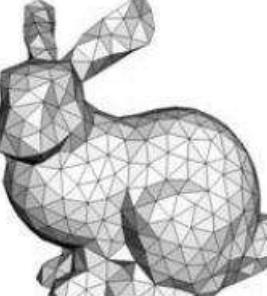
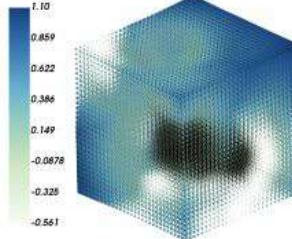
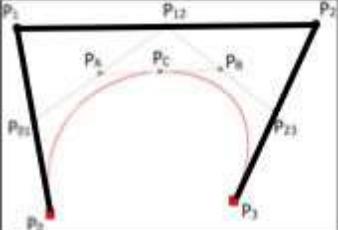
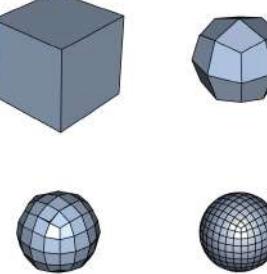
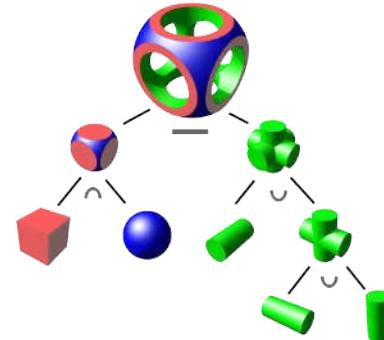
(cabinets, doors, fridge,
oven, window etc.)



AI + Geometry

- $P(S)$ or $P(S|c)$ --- Generative models
 - Learning (conditional) shape priors
 - Shape generation, completion, & geometry data processing
- $P(c|S)$ --- Discriminative models
 - Learning shape descriptors
 - Shape classification, segmentation, view estimation, etc.
- Joint modeling of 3D and 2D data
 - Large-scale 2D datasets & very good pretrained models
 - Differentiable projection/back-projection & differentiable/neural rendering

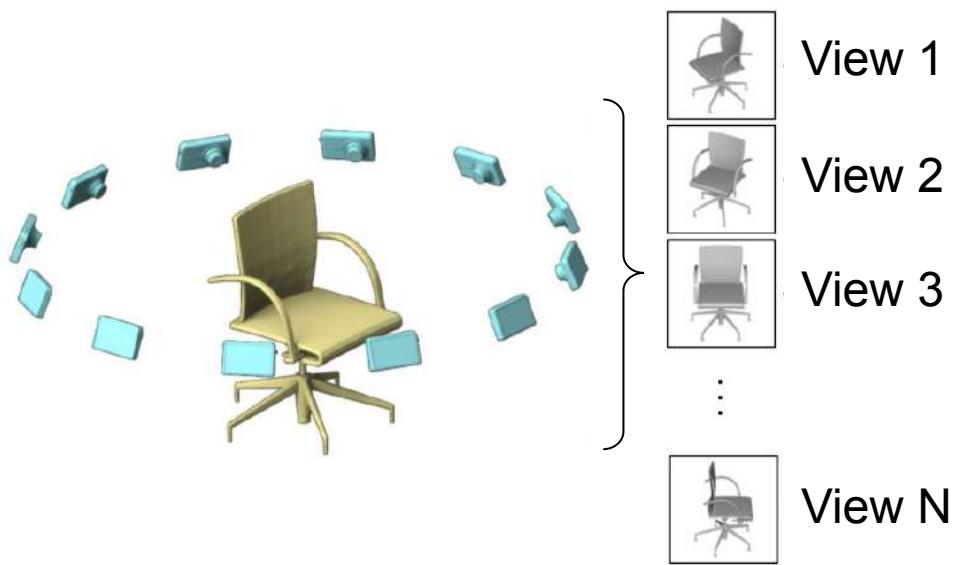
Which Shape Representation?

		Explicit	Implicit (Eulerian)
Non-parametric	Points		
	Meshes		
Parametric	Splines		 $x^2 + y^2 + z^2 = 1$
	Subdivision Surfaces		

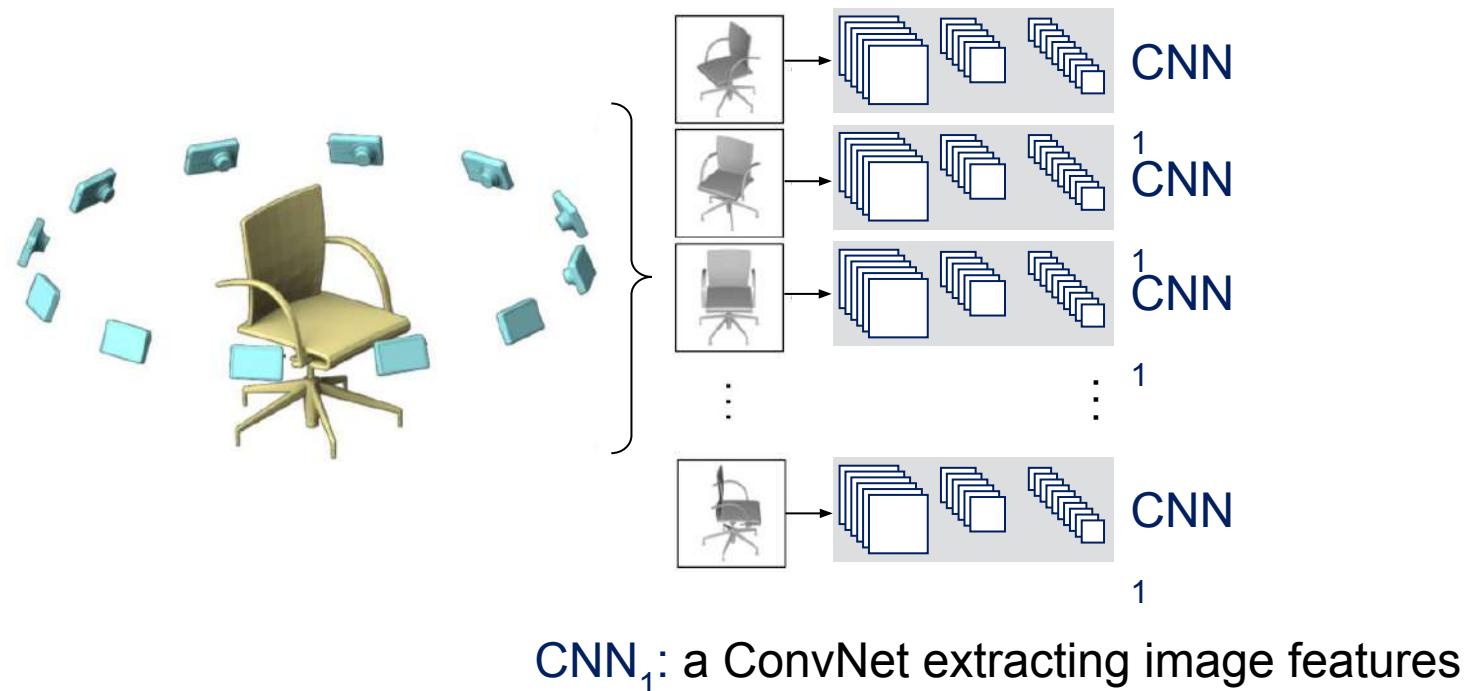
Multi-View CNN



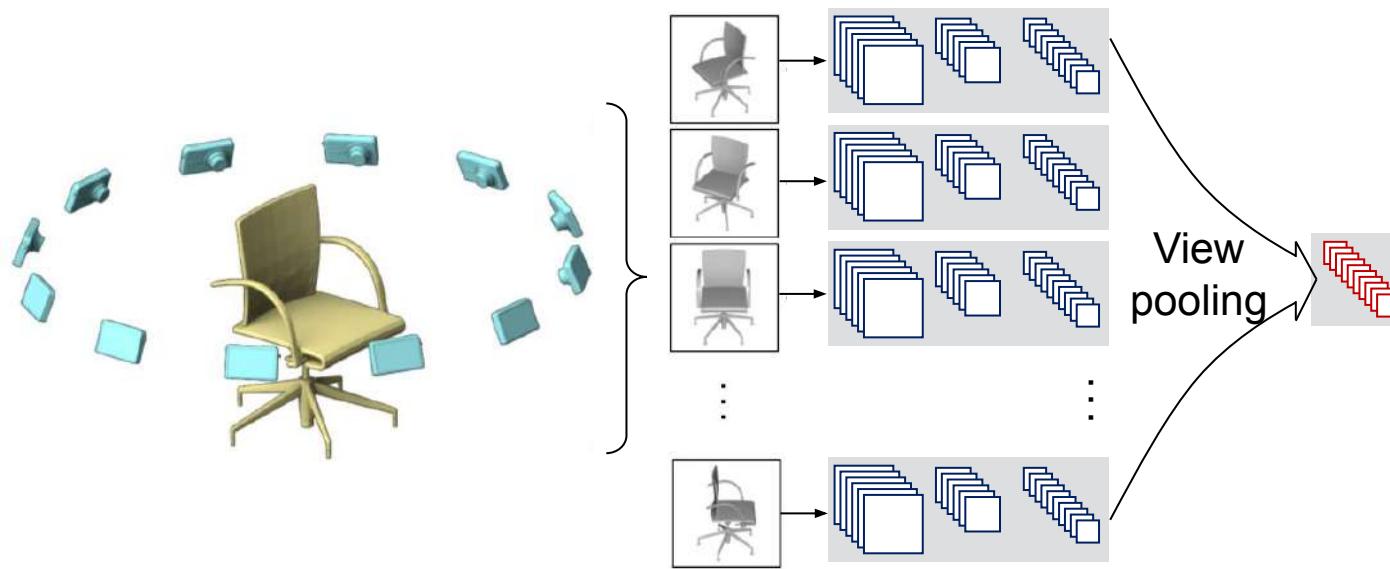
Multi-View CNN



Multi-View CNN

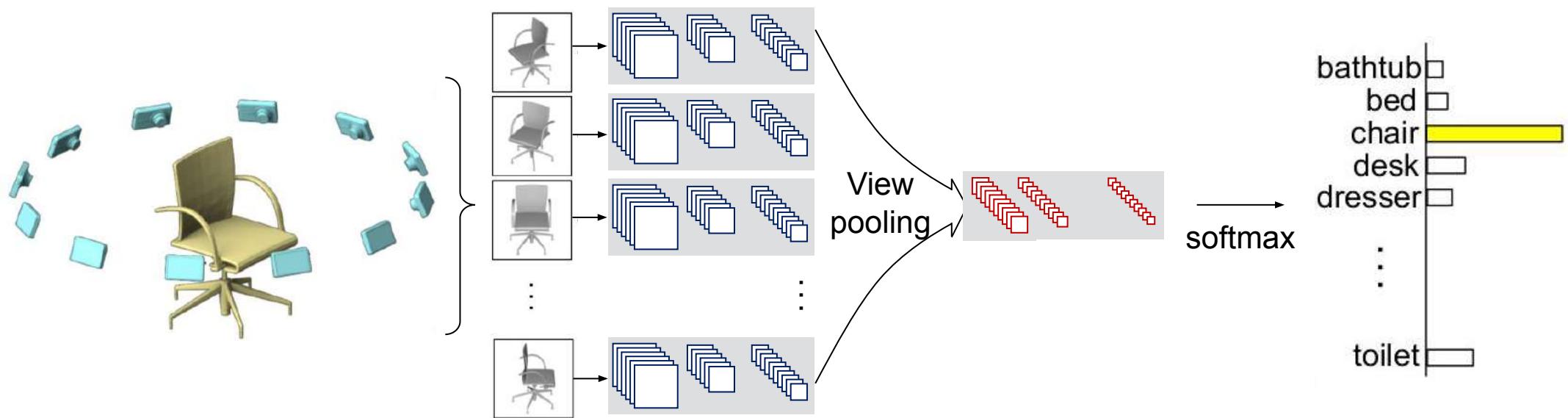


Multi-View CNN



View pooling: element-wise
max-pooling across all views

Multi-View CNN



CNN_2 : a second ConvNet
producing shape descriptors

Experiments – Classification & Retrieval

Non-dee
p {

Method	Classification	Retrieval
	(Accuracy)	(mAP)
SPH	68.2%	33.3%
LFD	75.5%	40.9%
3D ShapeNets	77.3%	49.2%
FV, 12 views	84.8%	43.9%
CNN, 12 views	88.6%	62.8%
MVCNN, 12 views	89.9%	70.1%
MVCNN+metric, 12 views	89.5%	80.2%
MVCNN, 80 views	90.1%	70.4%
MVCNN+metric, 80 views	90.1%	79.5%

On ModelNet

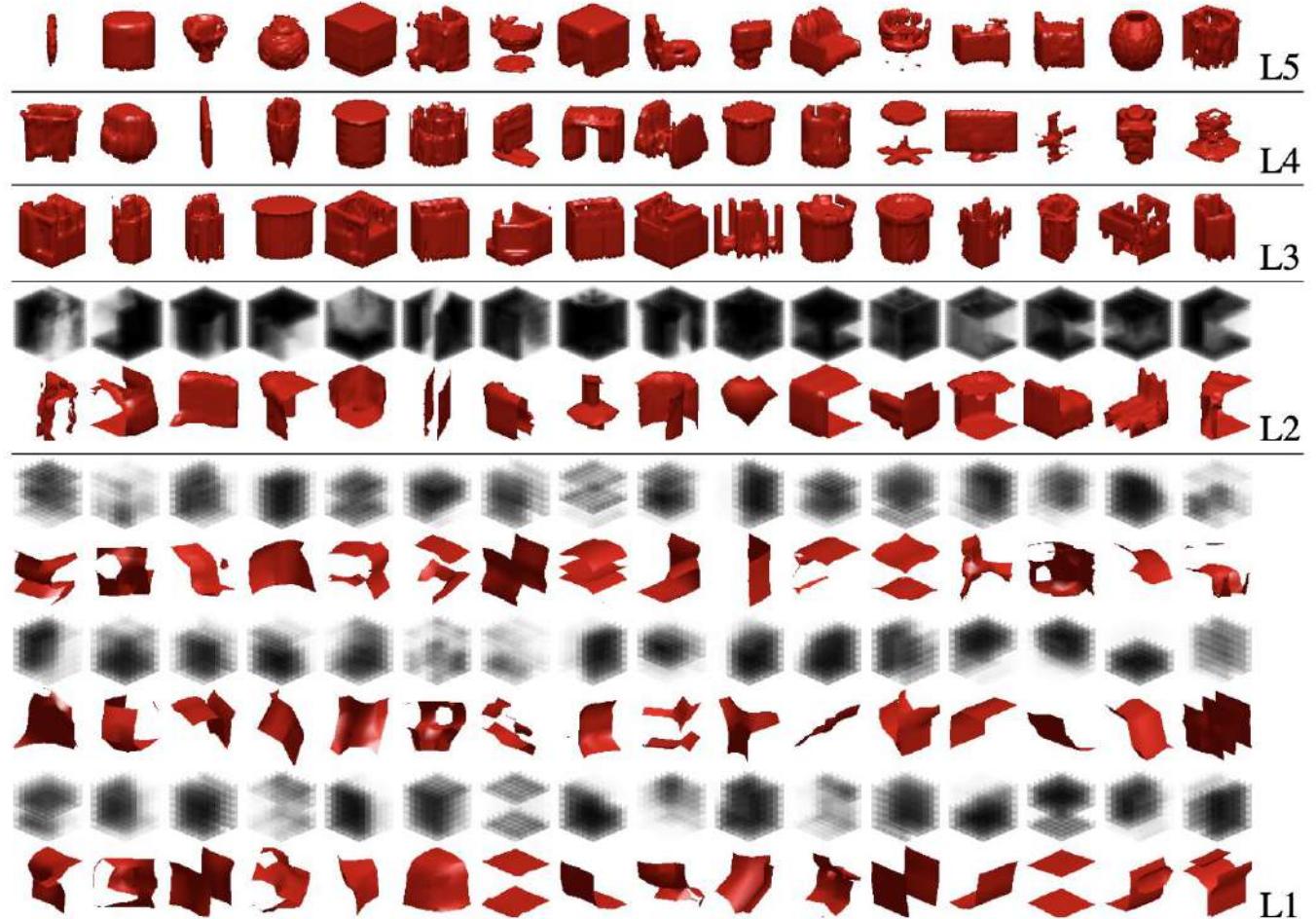
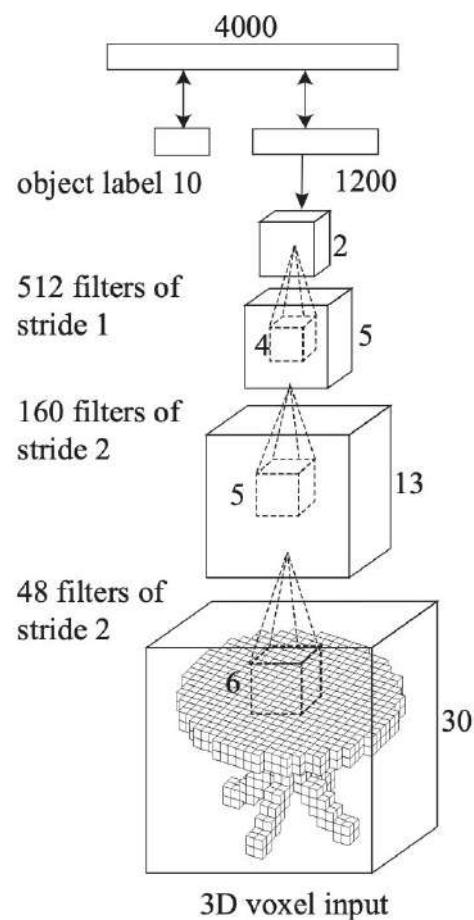
Multi-View Representations

- Indeed gives good performance
- Can leverage vast literature of image classification
- Can use pertained features

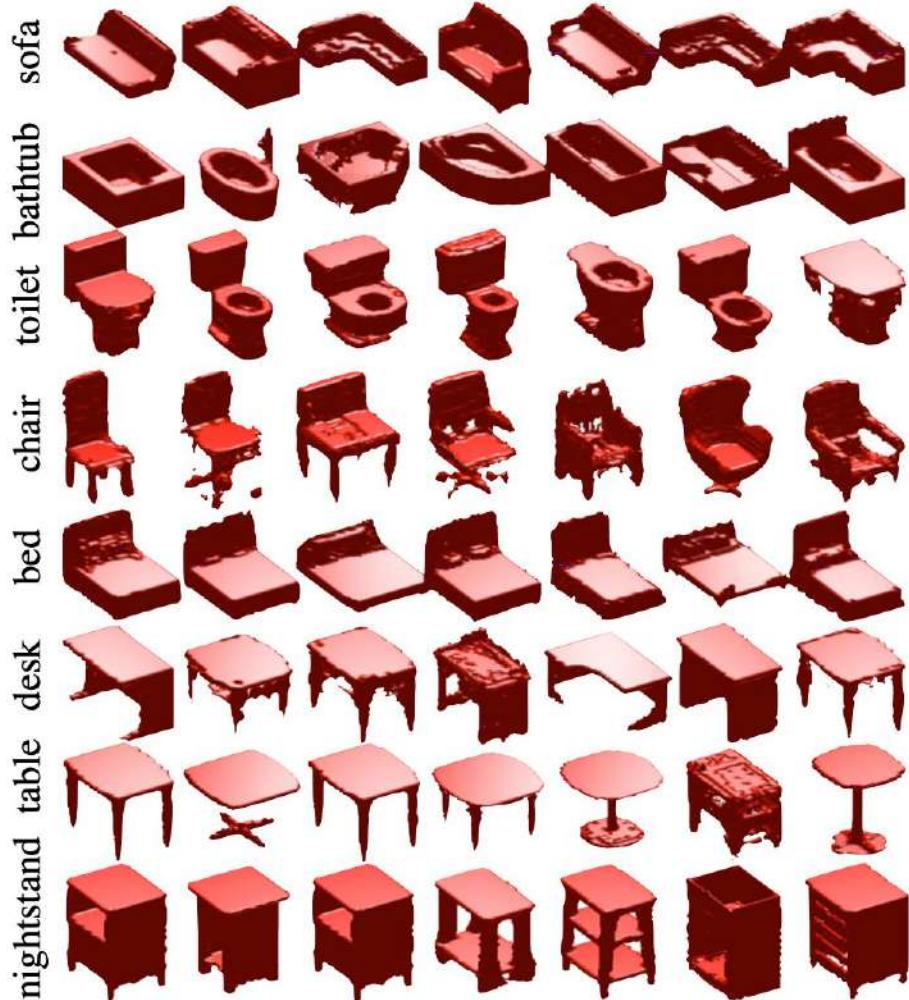
- Need projection
- What if the input is noisy and/or incomplete? e.g., point cloud

Pixels -> Voxels

- 3D Conv Deep Belief Networks (CDBN)



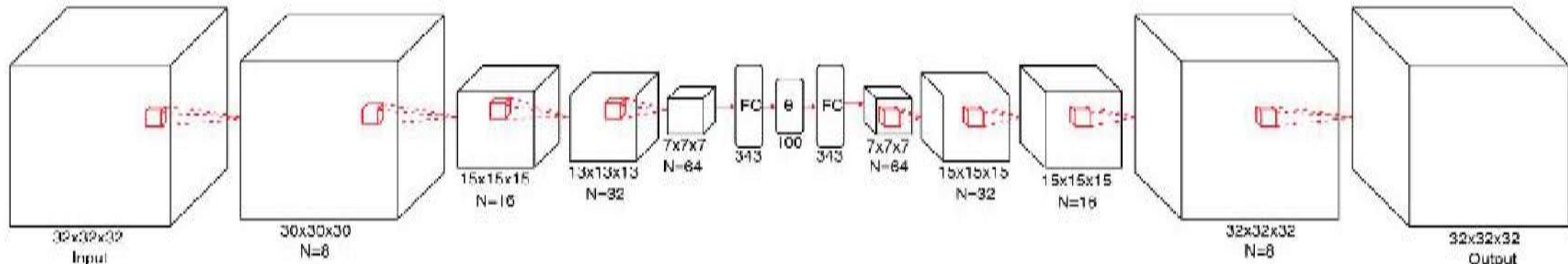
Generative Modeling



	10 classes	SPH [18]	LFD [8]	Ours
classification	79.79 %	79.87 %	83.54%	
retrieval AUC	45.97%	51.70%	69.28%	
retrieval MAP	44.05%	49.82%	68.26%	
	40 classes	SPH [18]	LFD [8]	Ours
classification	68.23%	75.47%	77.32%	
retrieval AUC	34.47%	42.04%	49.94%	
retrieval MAP	33.26%	40.91%	49.23%	

Table 1: Shape Classification and Retrieval Results.

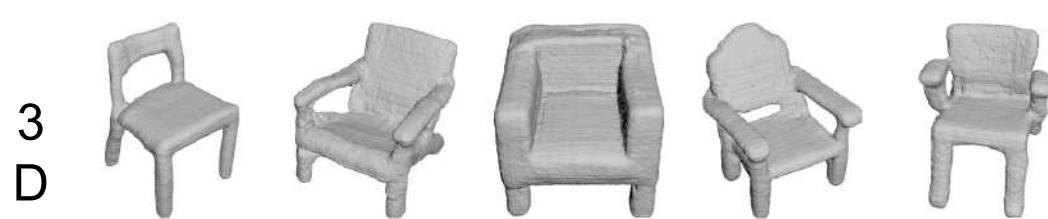
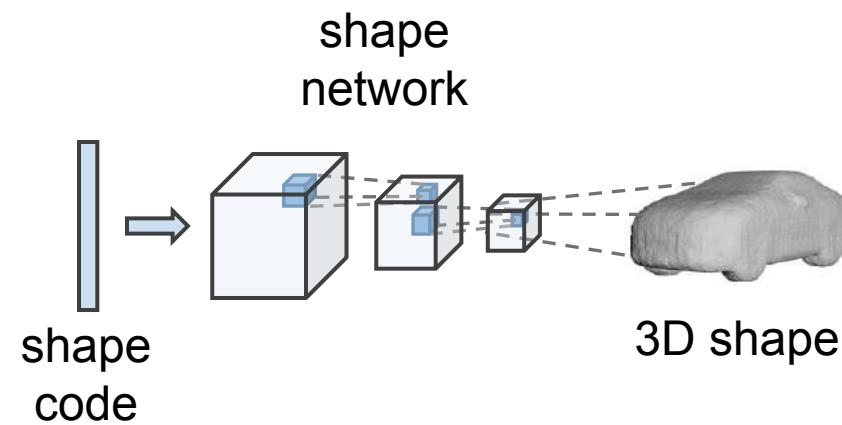
Volumetric Autoencoders



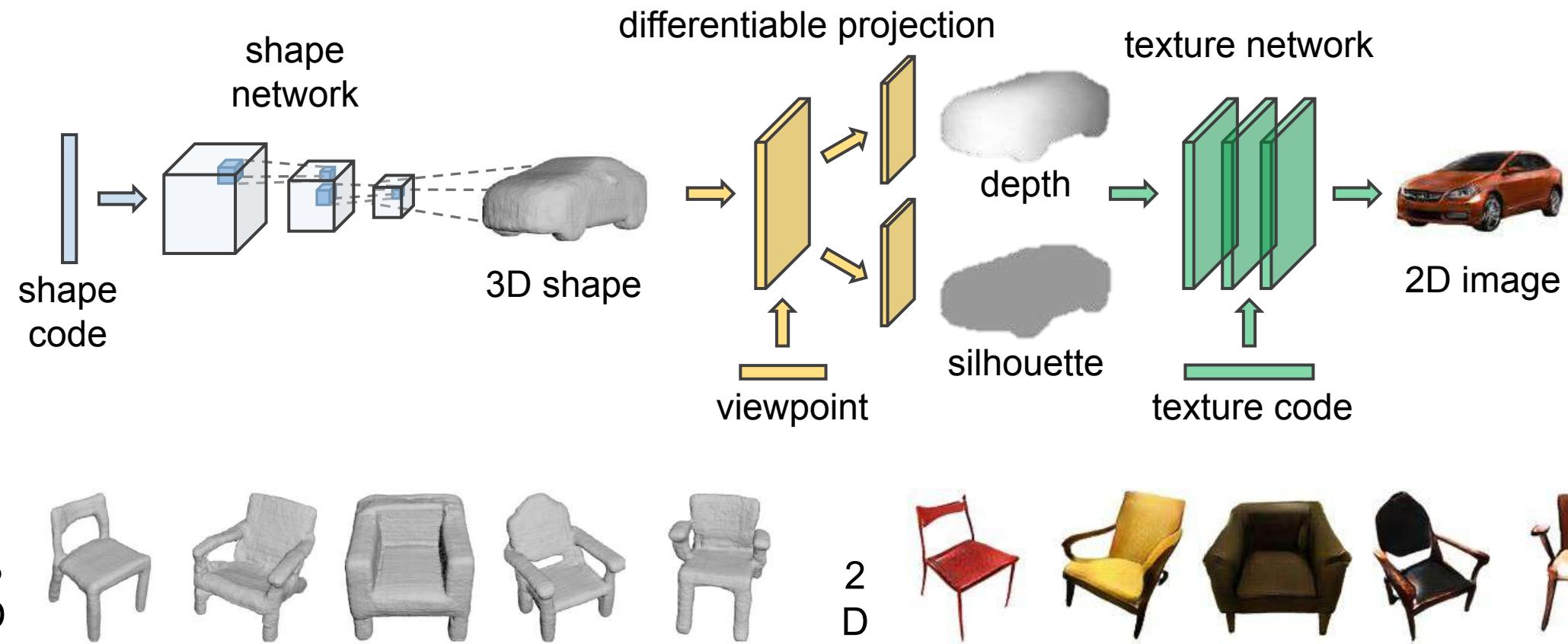
Binary Cross-Entropy Loss: $\mathcal{L} = -t \log(o) - (1-t) \log(1-o)$



3D-GANs



Visual Object Networks



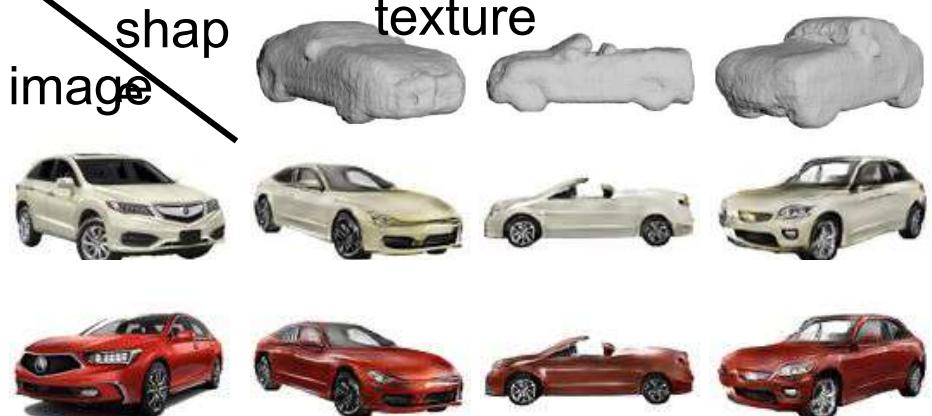
Editing viewpoint, shape, and texture



Interpolation in the latent space

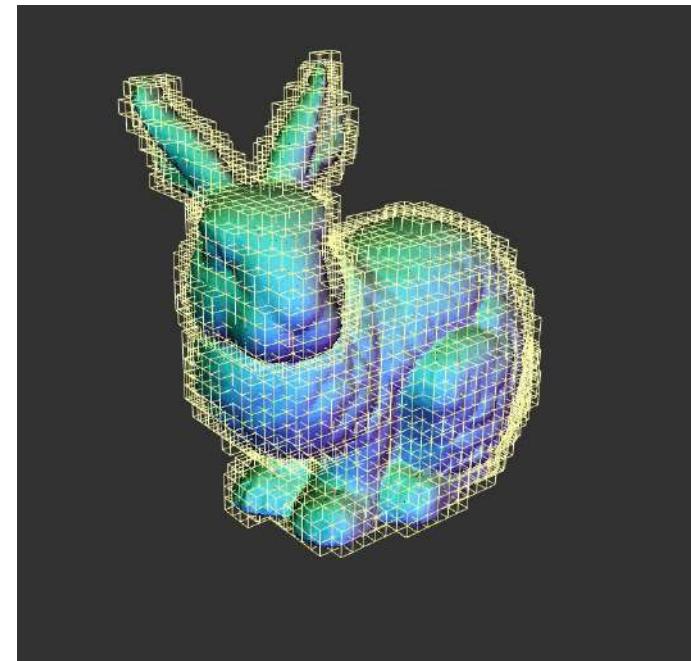
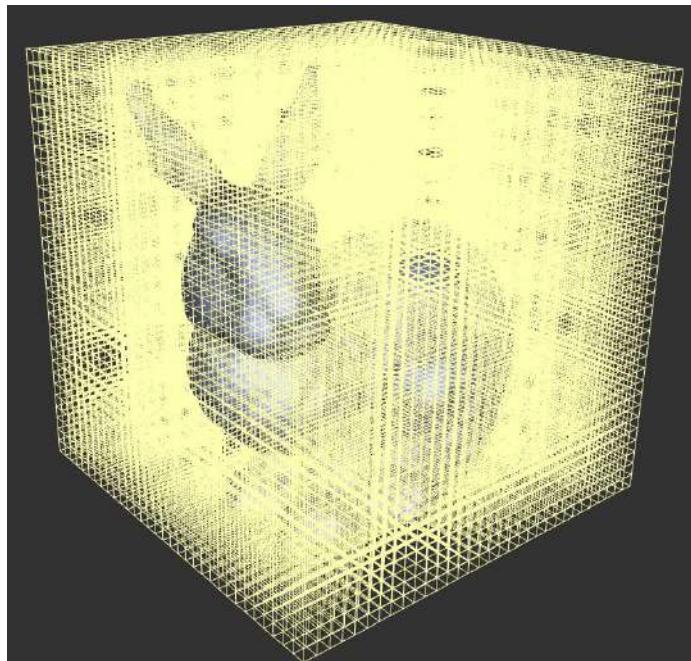


Transferring shape and texture



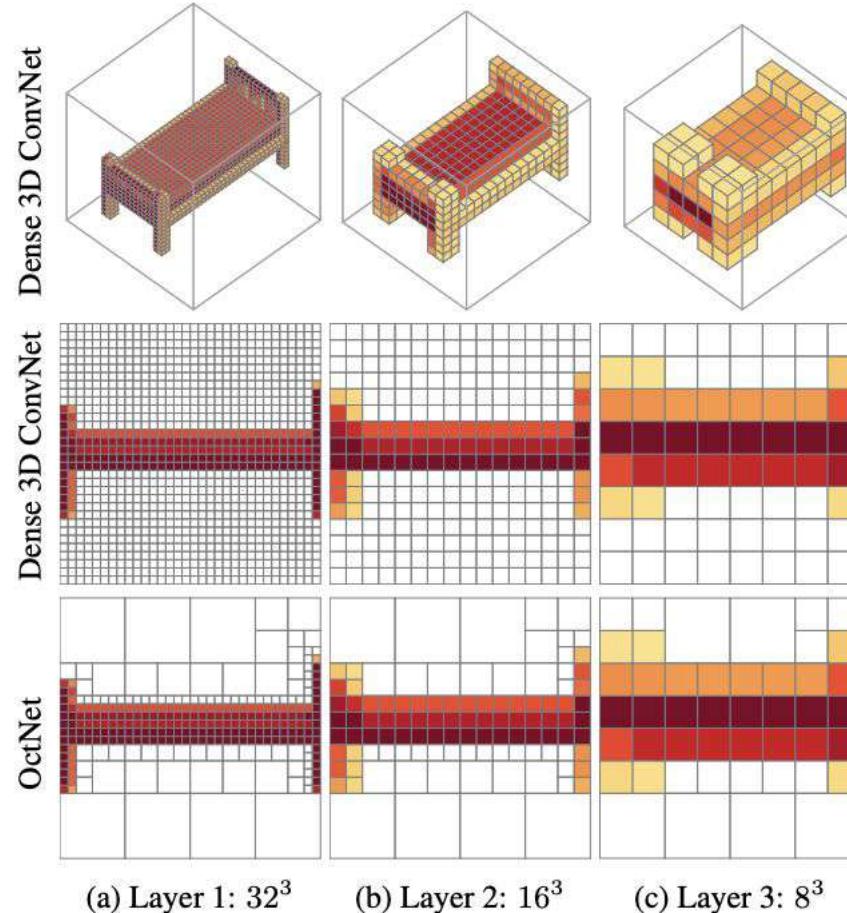
Octave Tree Representations

- Store the sparse surface signals
- Constrain the computation near the surface

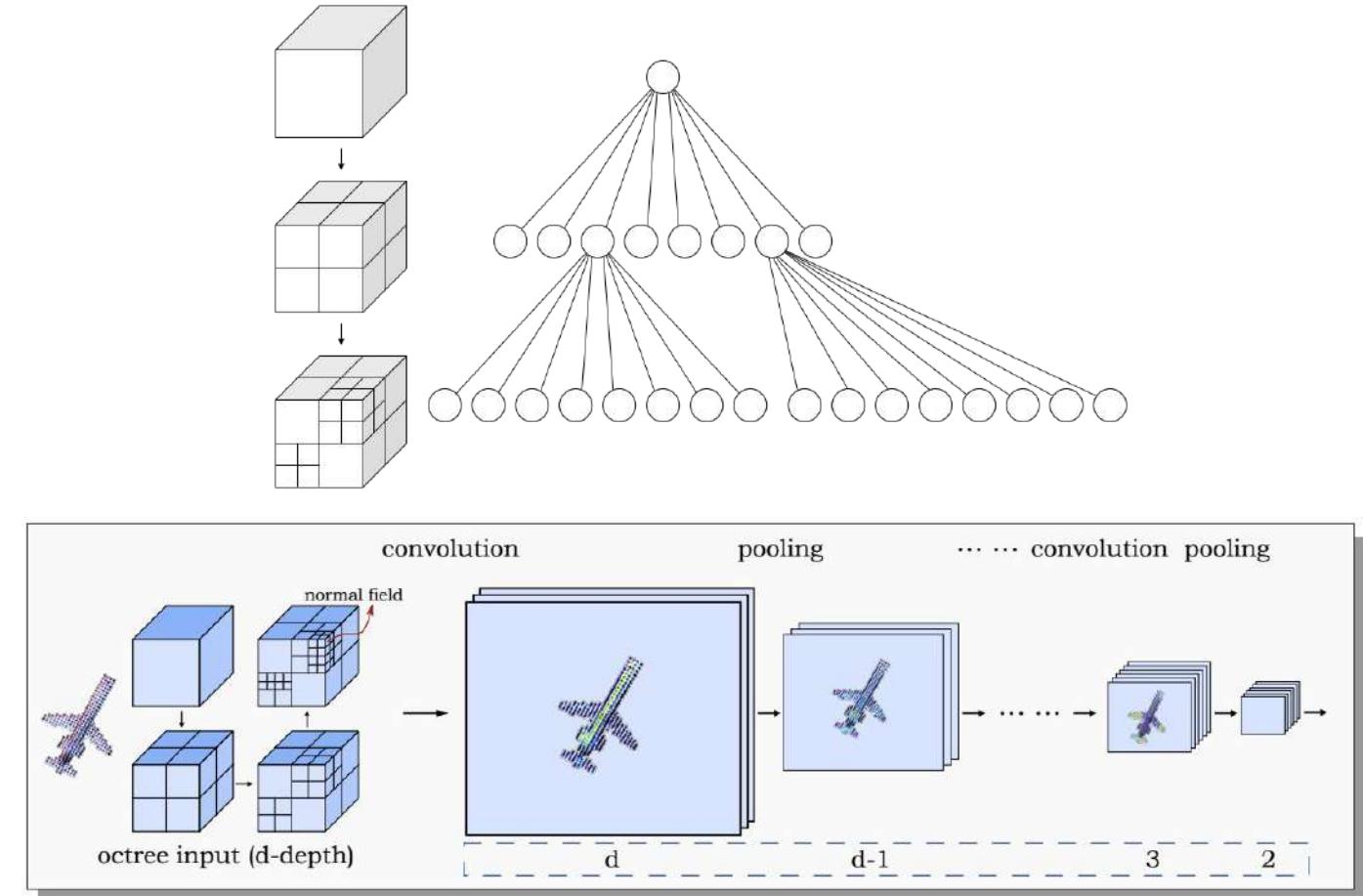


Slide Credit: Hao Su

Octree: Recursively Partition the Space

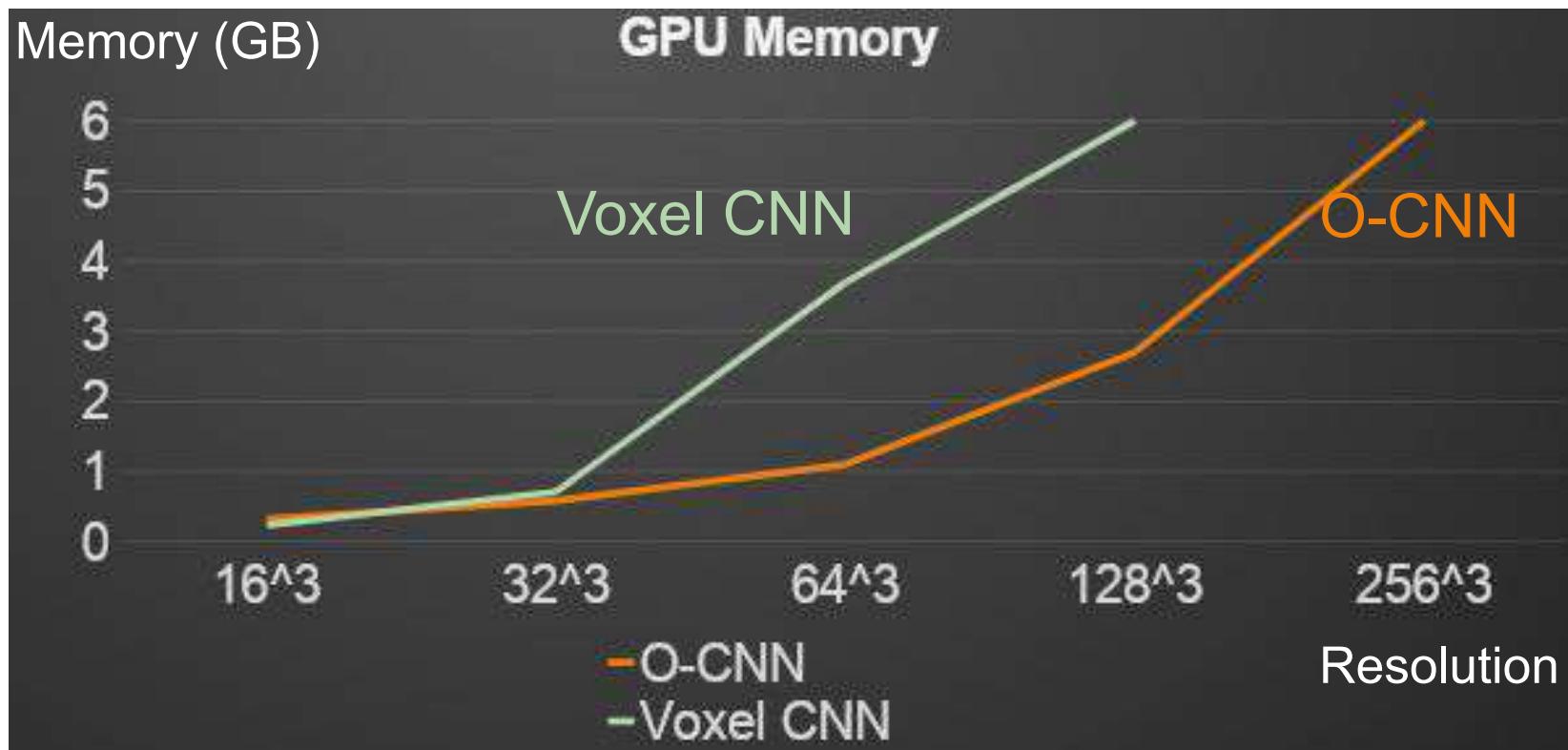


Riegler et al. OctNet. CVPR 2017

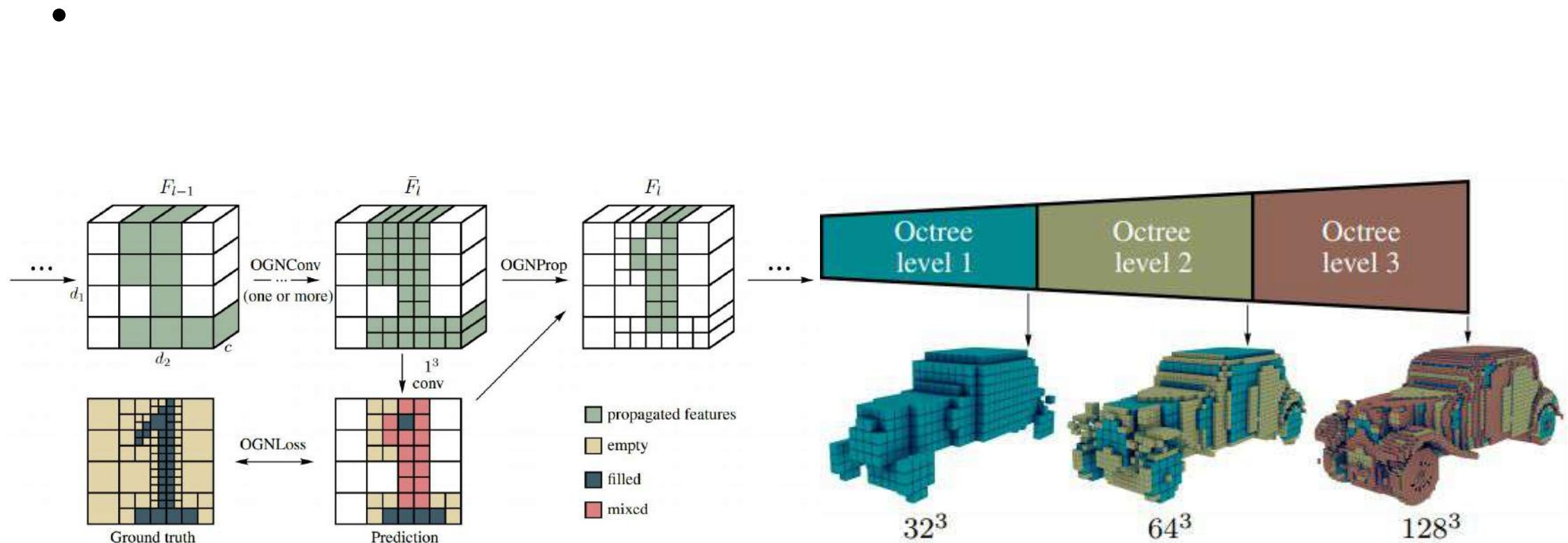


Wang et al. O-CNN. SIGGRAPH 2017

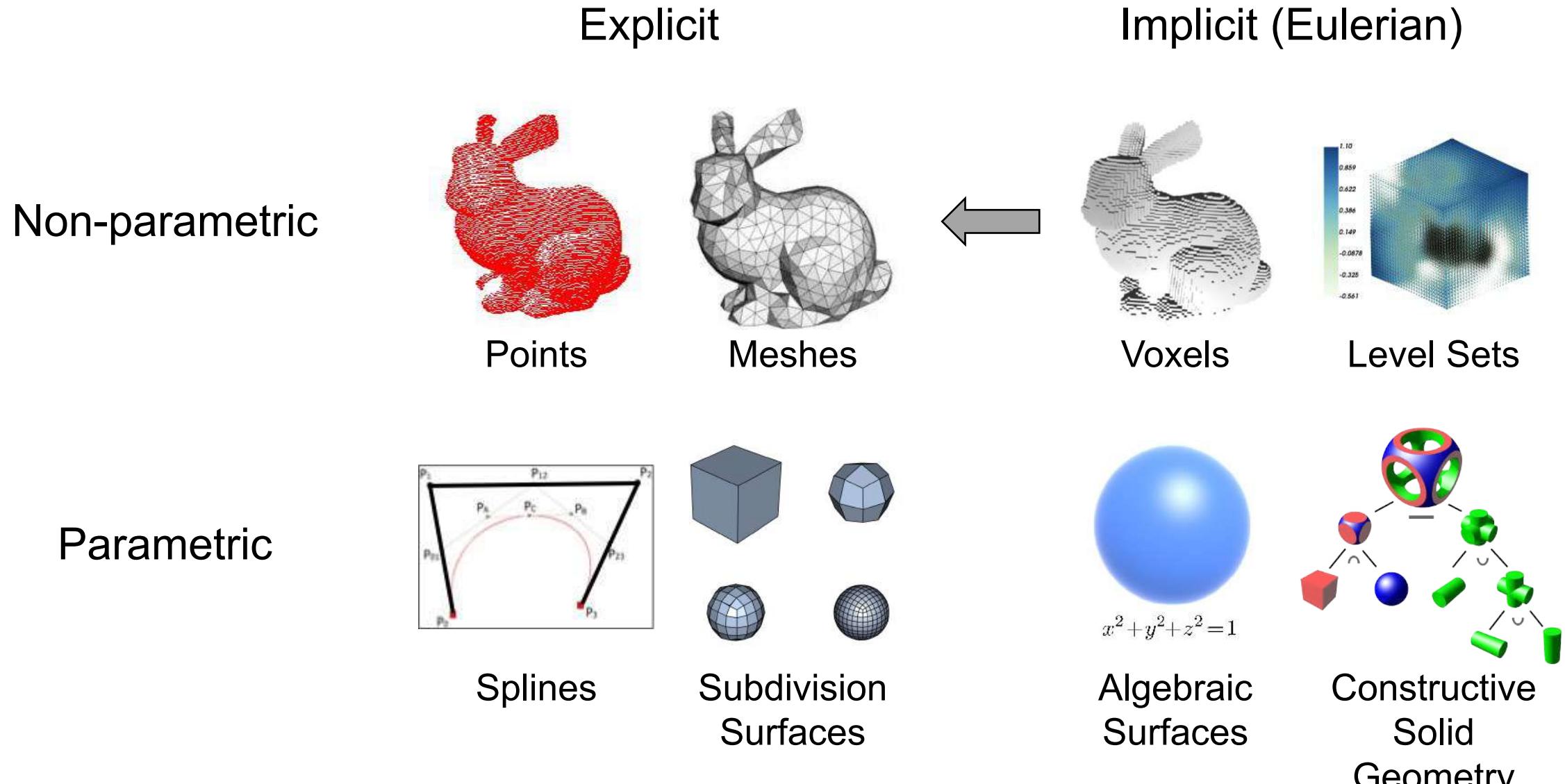
Memory Efficiency



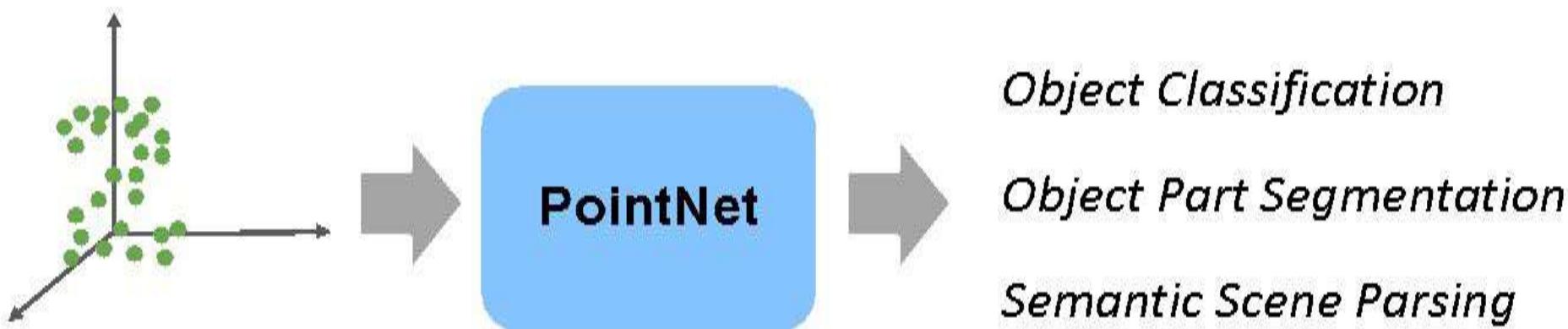
Octree Generating Networks



Eulerian \rightarrow Lagrangian

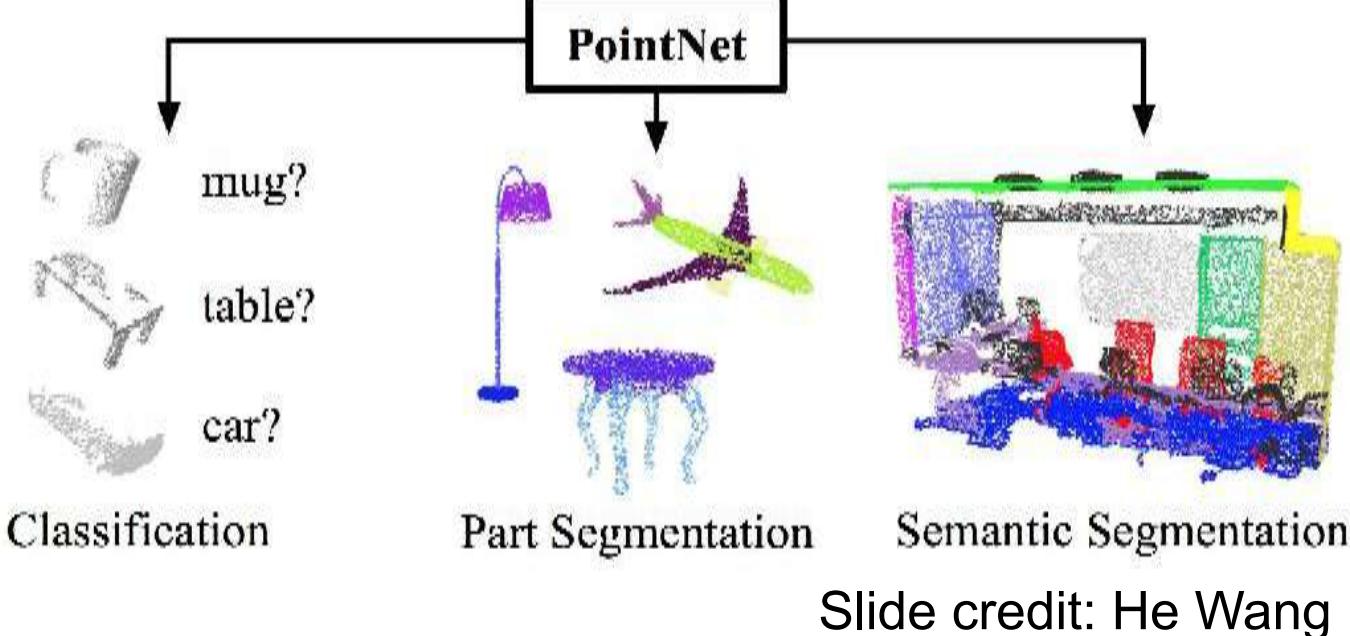


PointNet: First Learning Tool for Point Clouds



End-to-end learning for irregular point data

Unified framework for various tasks



Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas.
PointNet: Deep Learning on Point Sets for 3D
Classification and Segmentation. (CVPR '17)

Slide credit: He Wang

Invariances

The model has to respect key desiderata for point clouds:

Point Permutation Invariance

Point cloud is a set of **unordered** points

Sampling Invariance

Output a function of the underlying geometry and **not the sampling**

Permutation Invariance: Symmetric Functions

$$f(x_1, x_2, \dots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \dots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \dots, x_n) = \max\{x_1, x_2, \dots, x_n\}$$

$$f(x_1, x_2, \dots, x_n) = x_1 + x_2 + \dots + x_n$$

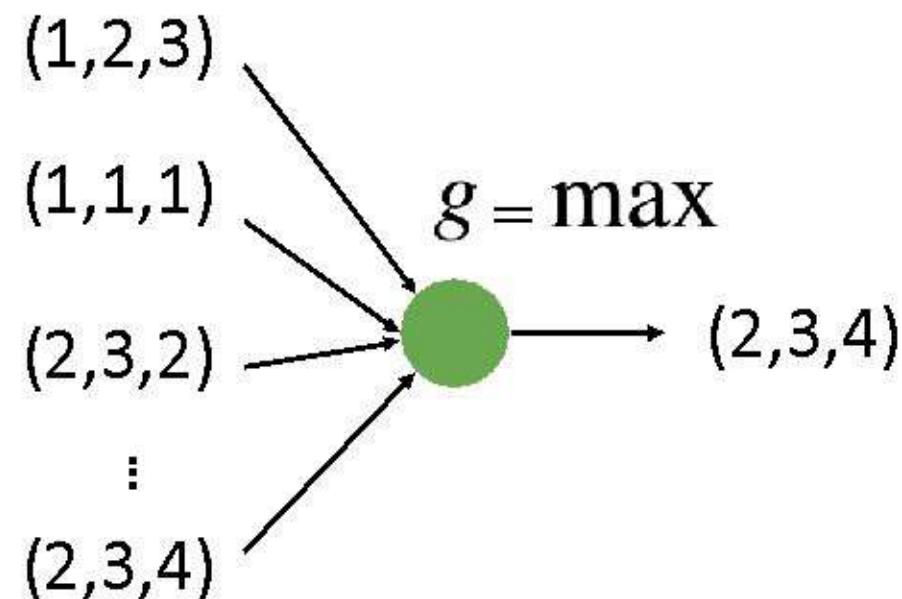
...

How can we construct a universal family of symmetric functions by neural networks?

Construct Symmetric Functions by Neural Networks

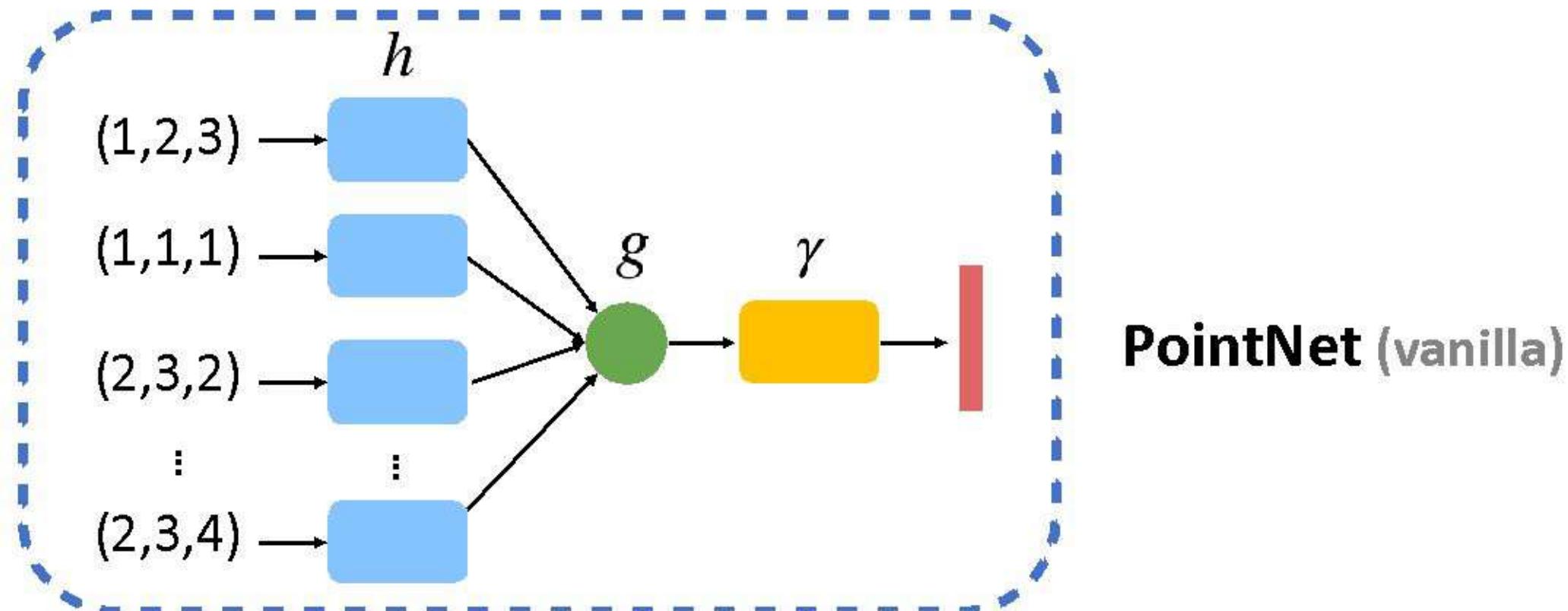
Simplest form: directly aggregate all points with a symmetric operator g

Just discovers simple extreme/aggregate properties of the geometry.



Construct Symmetric Functions by Neural Networks

$f(x_1, x_2, \dots, x_n) = \gamma \circ g(h(x_1), \dots, h(x_n))$ is symmetric if g is symmetric

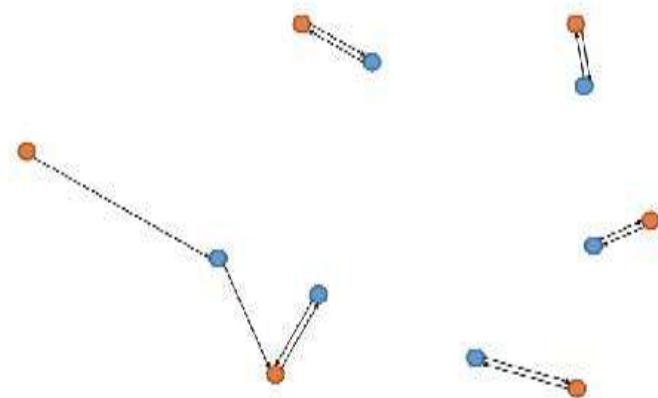


PointNet (vanilla)

Distance Metrics for Point Cloud

Chamfer distance We define the Chamfer distance between $S_1, S_2 \subseteq \mathbb{R}^3$ as:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2$$



Distance Metrics for Point Cloud

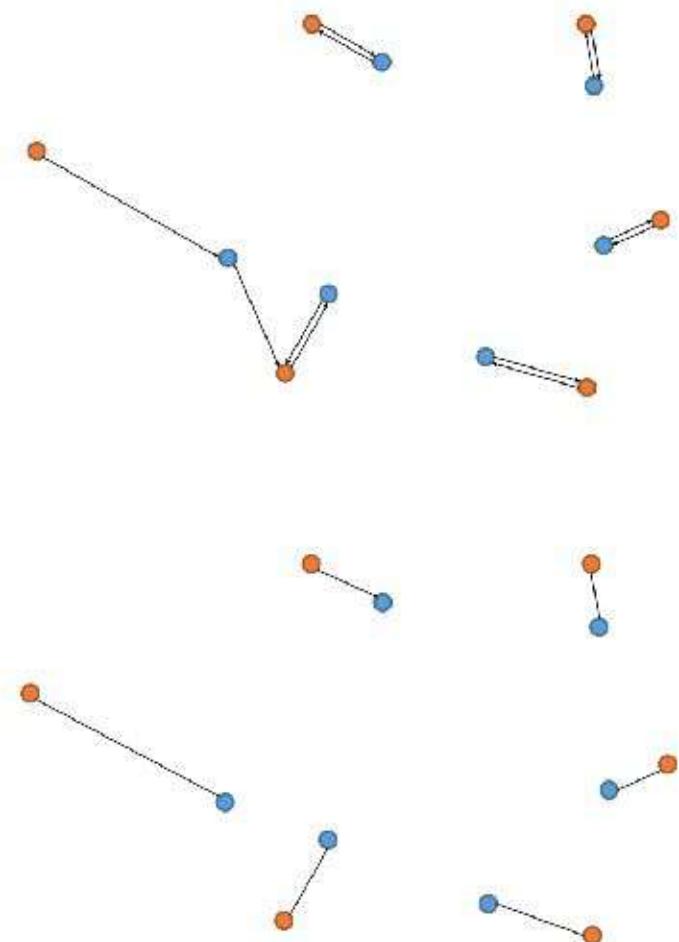
Chamfer distance We define the Chamfer distance between $S_1, S_2 \subseteq \mathbb{R}^3$ as:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2$$

Earth Mover's distance Consider $S_1, S_2 \subseteq \mathbb{R}^3$ of equal size $s = |S_1| = |S_2|$. The EMD between A and B is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where $\phi : S_1 \rightarrow S_2$ is a bijection.



Distance Metrics for Point Cloud

Chamfer distance We define the Chamfer distance between $S_1, S_2 \subseteq \mathbb{R}^3$ as:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2$$

Earth Mover's distance Consider $S_1, S_2 \subseteq \mathbb{R}^3$ of equal size $s = |S_1| = |S_2|$. The EMD between A and B is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where $\phi : S_1 \rightarrow S_2$ is a bijection.

Sum of closest distances

In insensitive to sampling
(only relatively)

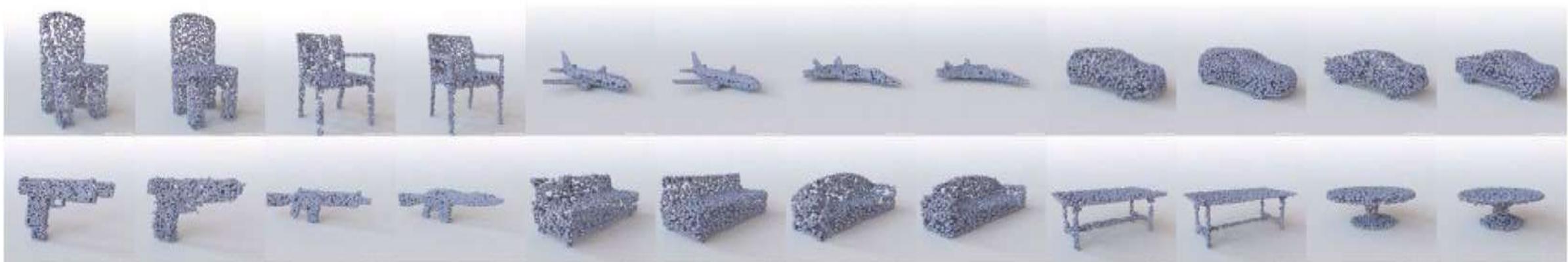
Sum of matched closest distances

Sensitive to sampling

Point Cloud AE

Encoder: PointNet

Decoder: MLP

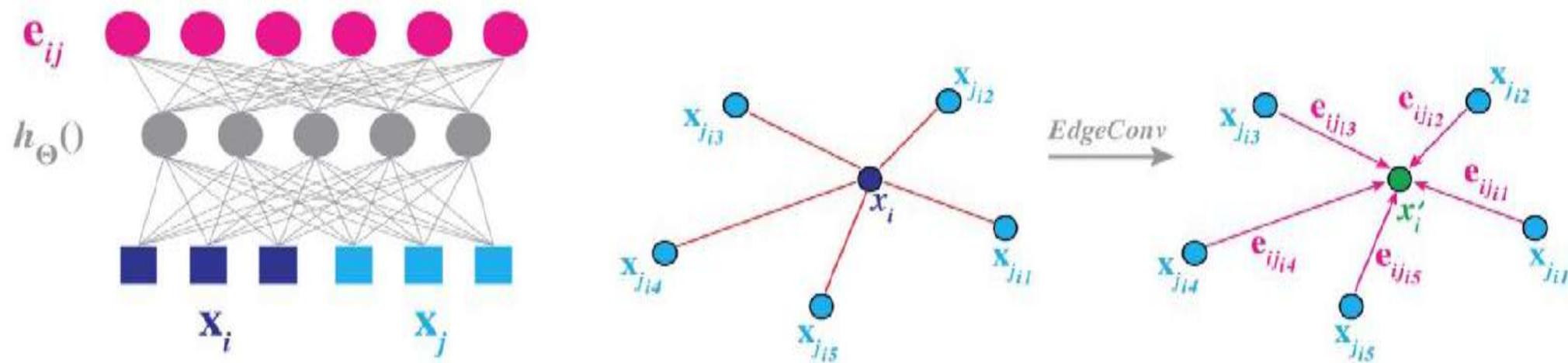


ICML 2018, Learning Representations and Generative Models for
3D Point Clouds, Panos Achlioptas, et. al.

Slide credit: He Wang

Graph NNs on Point Clouds

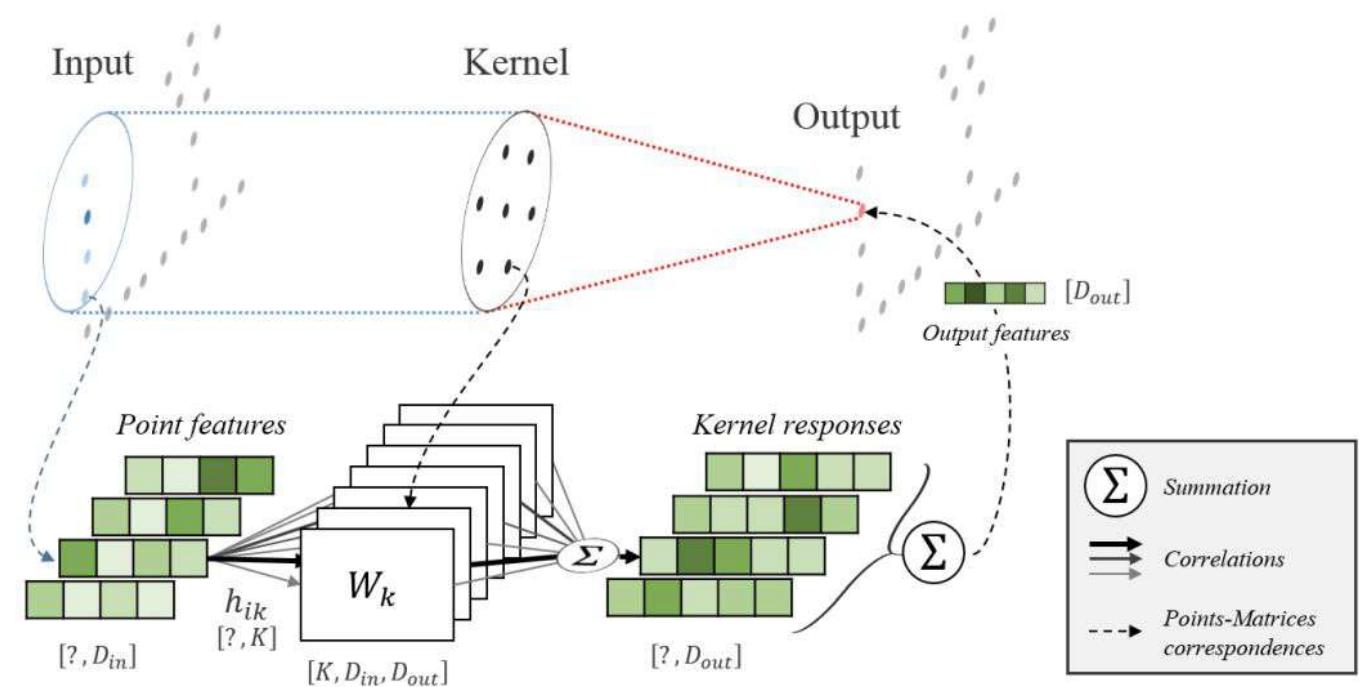
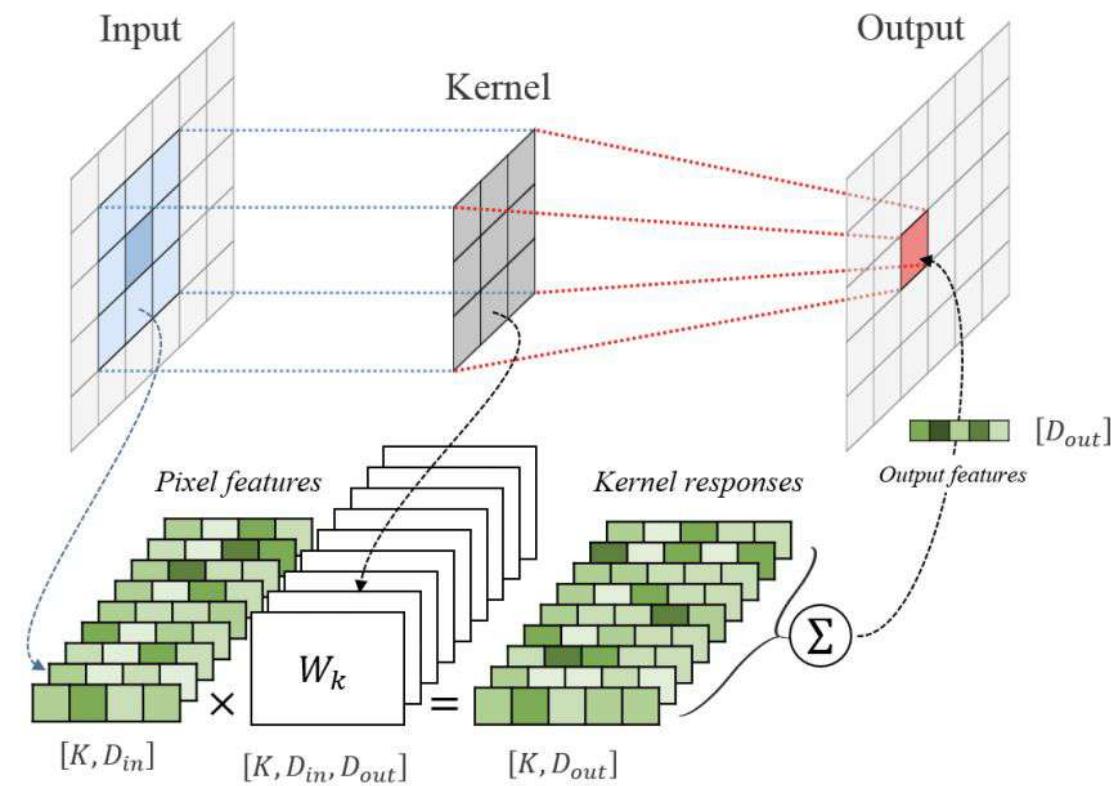
- Points -> Nodes
- Neighborhood -> Edges
- Graph NNs for point cloud processing



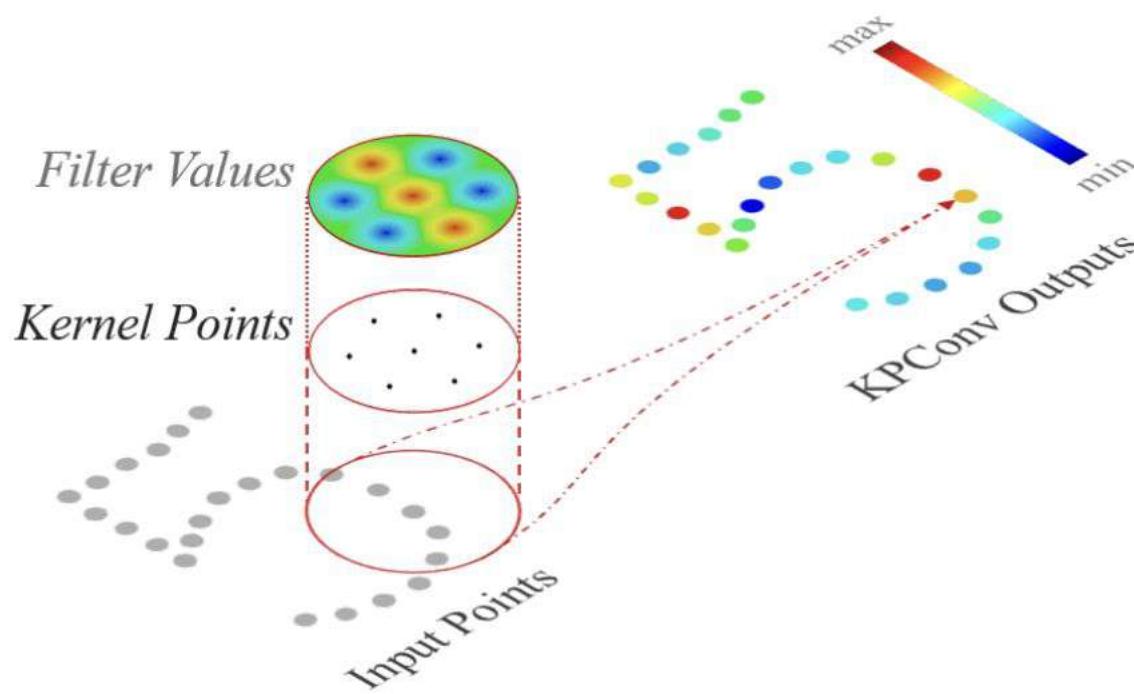
Message-Passing GNNs are not Geometry-Aware

- Points are **sampled** from surfaces.
- Ideally, features describe the geometry of underlying surface. They should be sample invariant.
- Message-passing GNNs do not address sample invariance.
- **Solution:** Estimate the continuous kernel and point density for continuous convolution (KPConv)

Kernel Point Convolution (KPConv)

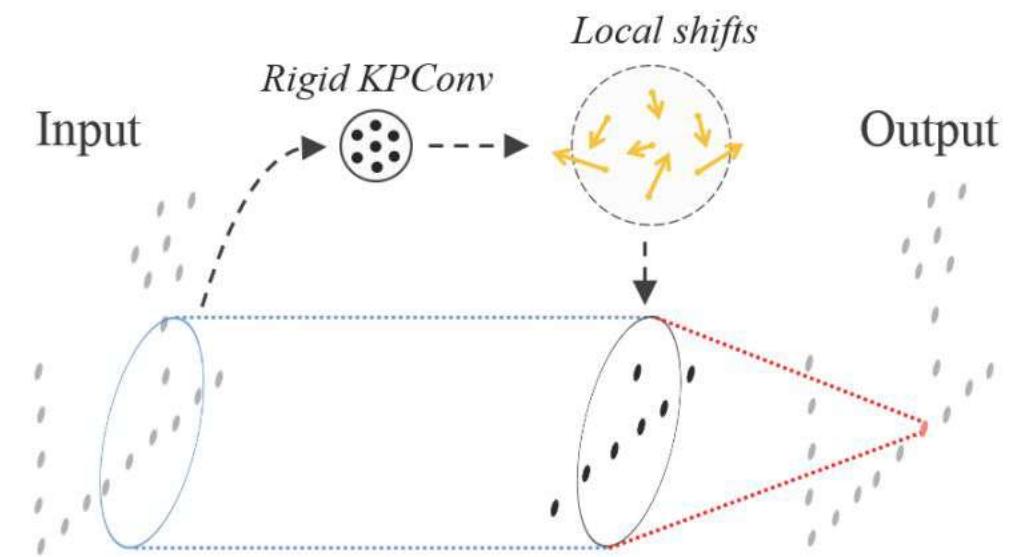


Deformable Kernel for Deformable Objects



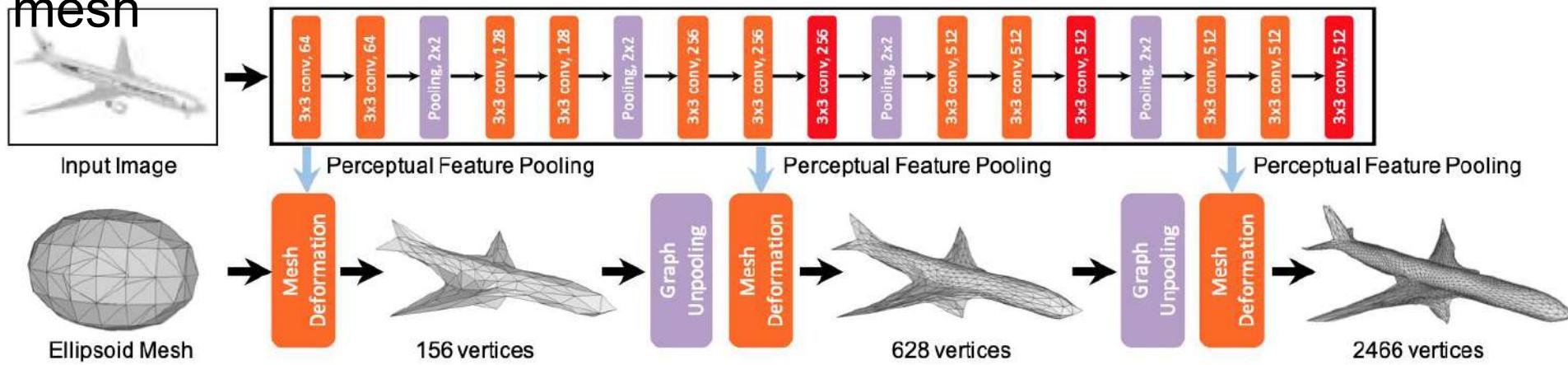
Deformable point-based kernel

- 3D version of 2D deformable convolution

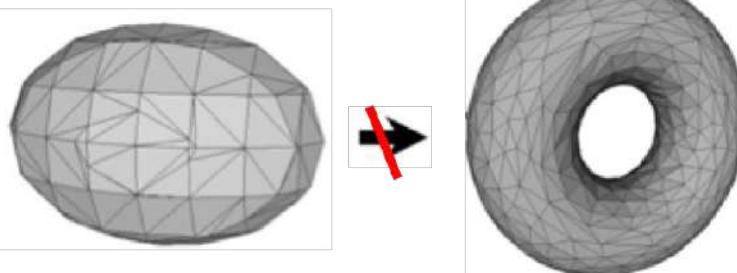


Pixel2Mesh

Learn to deform a template mesh

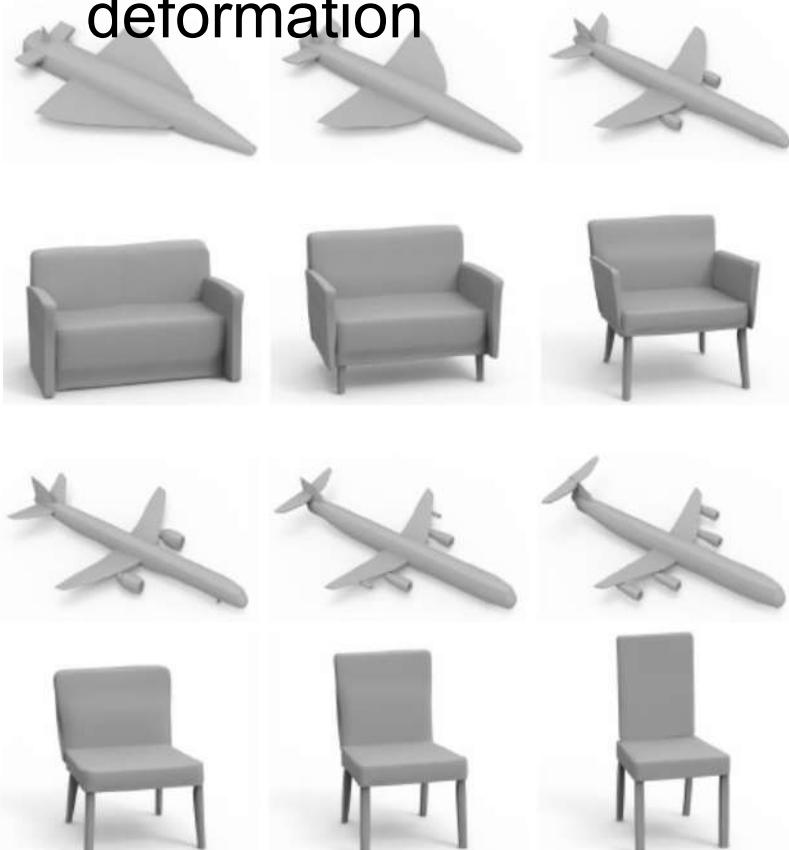


Cannot change the topology of the template mesh

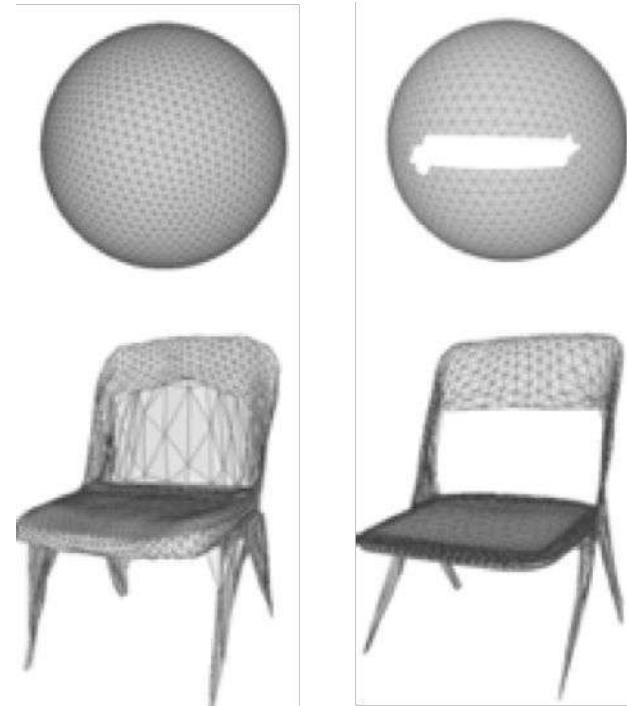


More on Mesh Deformation

Part-level
deformation



Modify the topology of
the template mesh

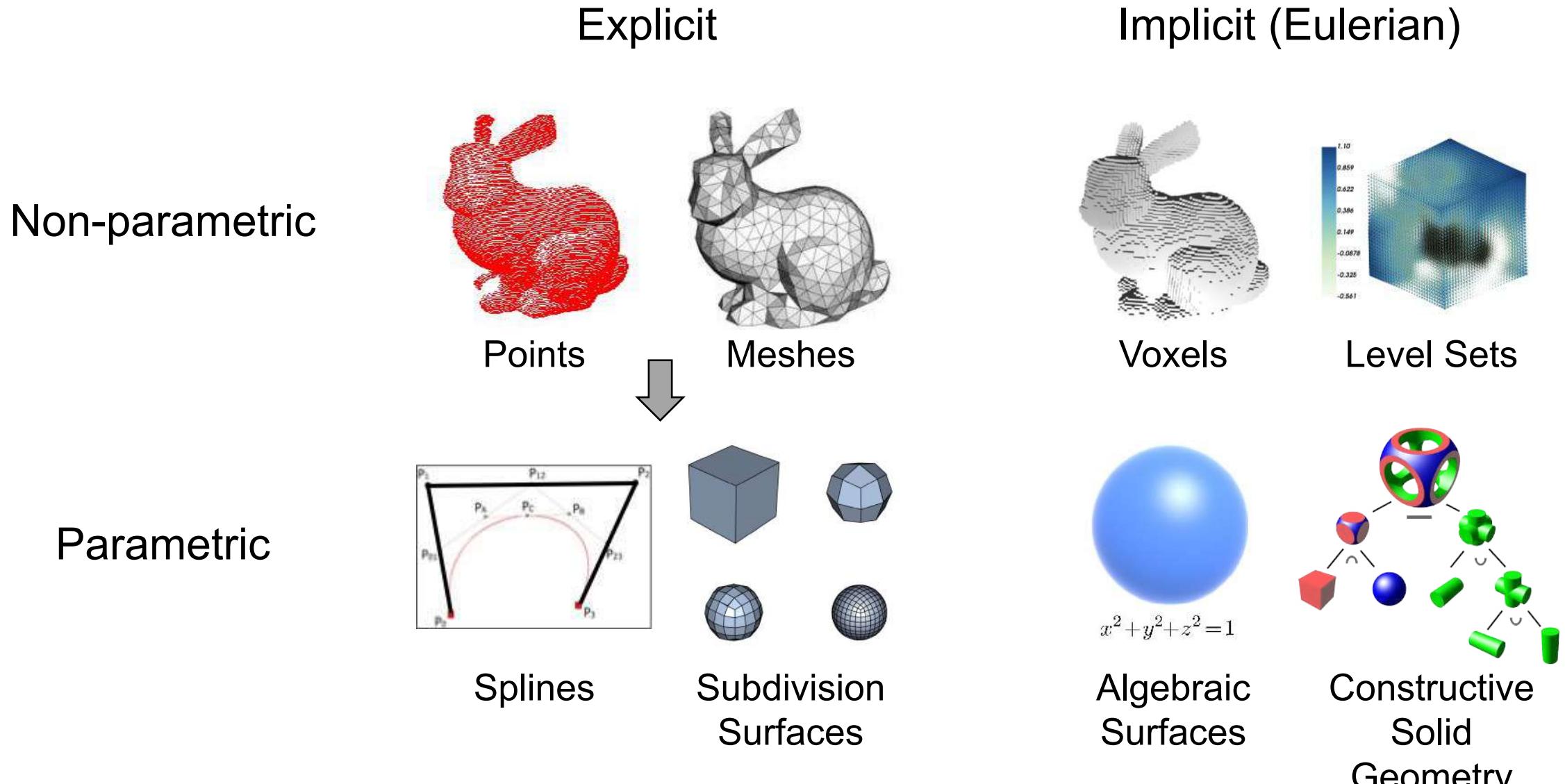


Gao et al. SDM-NET: Deep generative network for structured deformable mesh. SIGGRAPH Asia 2019

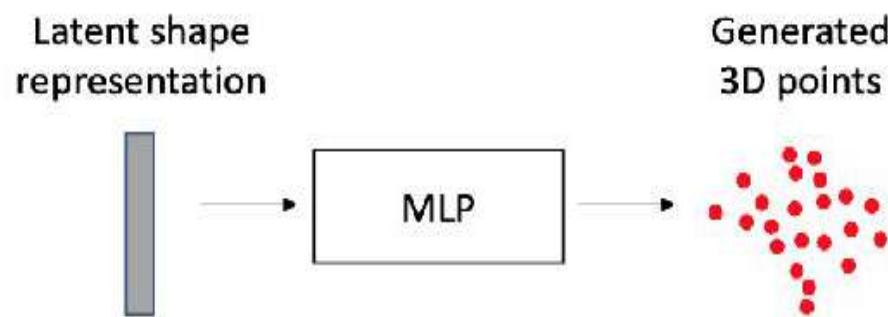
Pan et al., Deep Mesh Reconstruction from Single RGB Images via Topology Modification Networks. ICCV 2019

Slide: Hao Su

Non-Parametric → Parametric

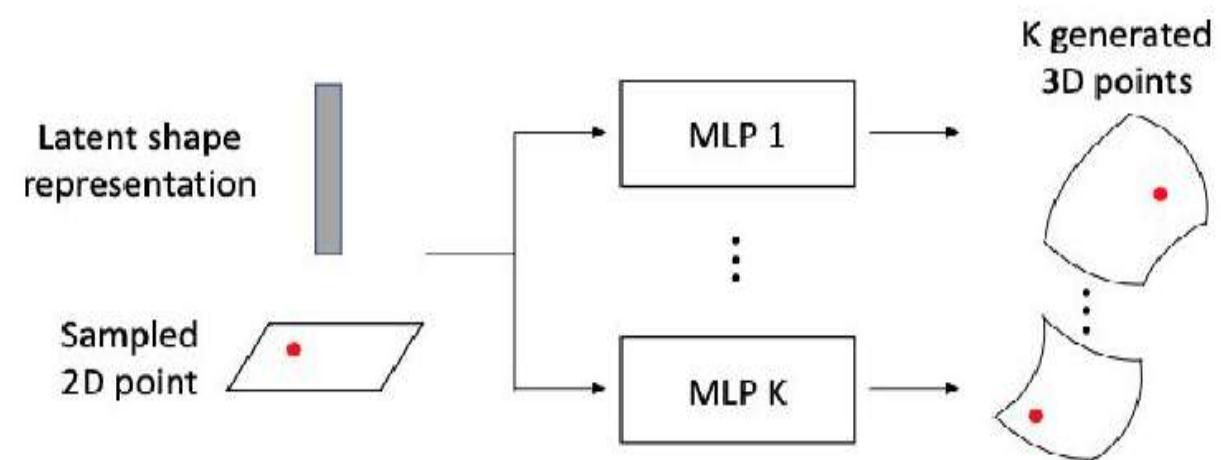
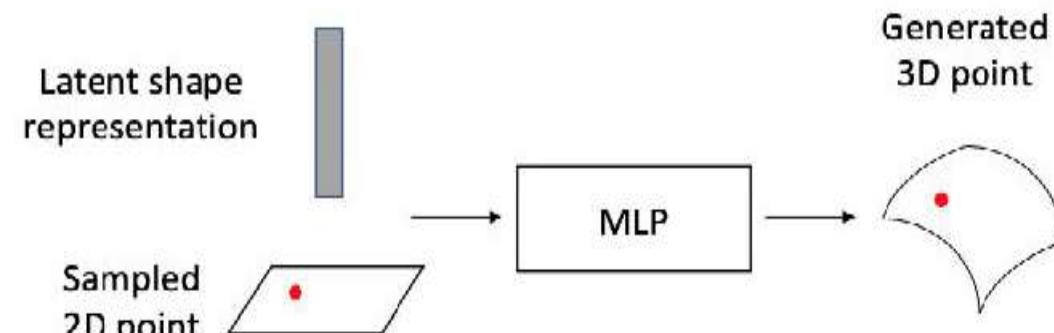


Parametric Decoder: AtlasNet



Given the output points form a smooth surface,
enforce such a parametrization as input.

$$\text{MLP}(z, u, v) \rightarrow \text{point}$$

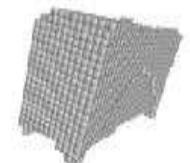
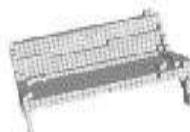


Results

Input image



Voxel



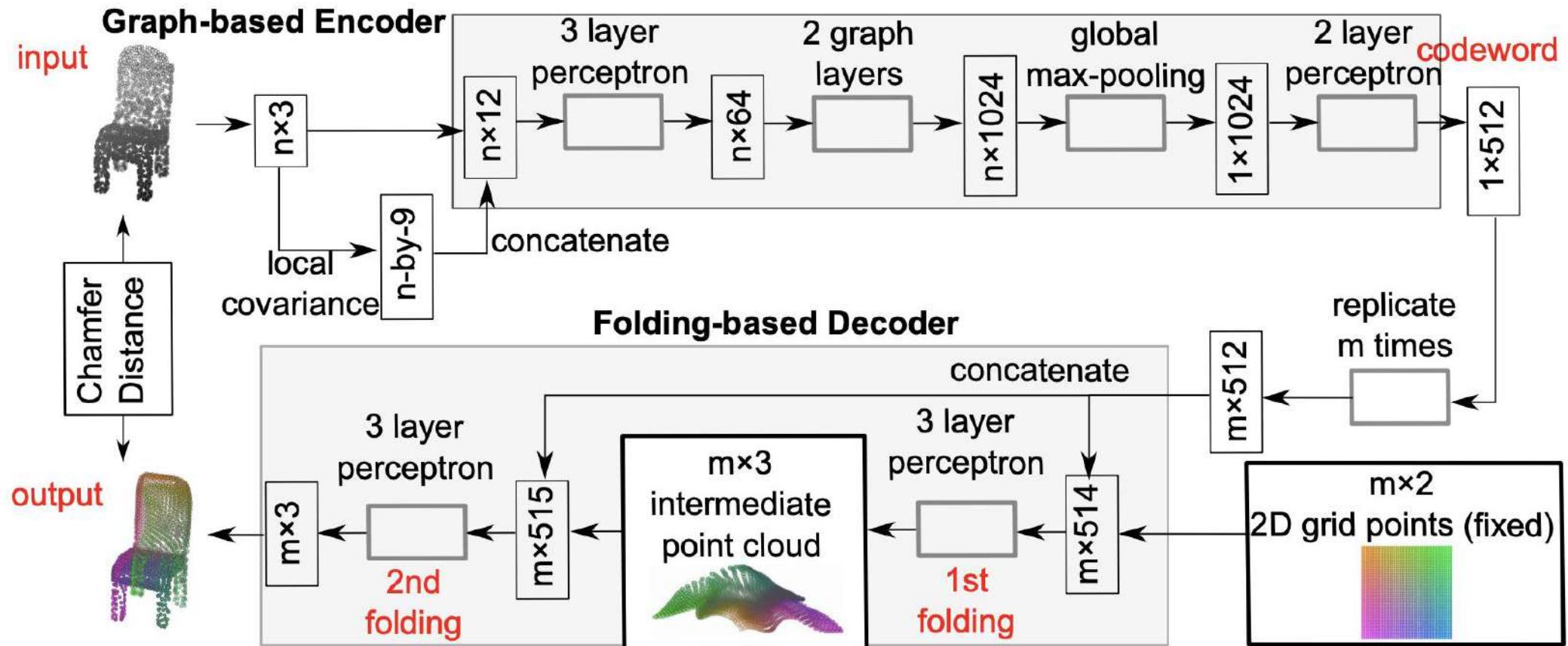
Point cloud



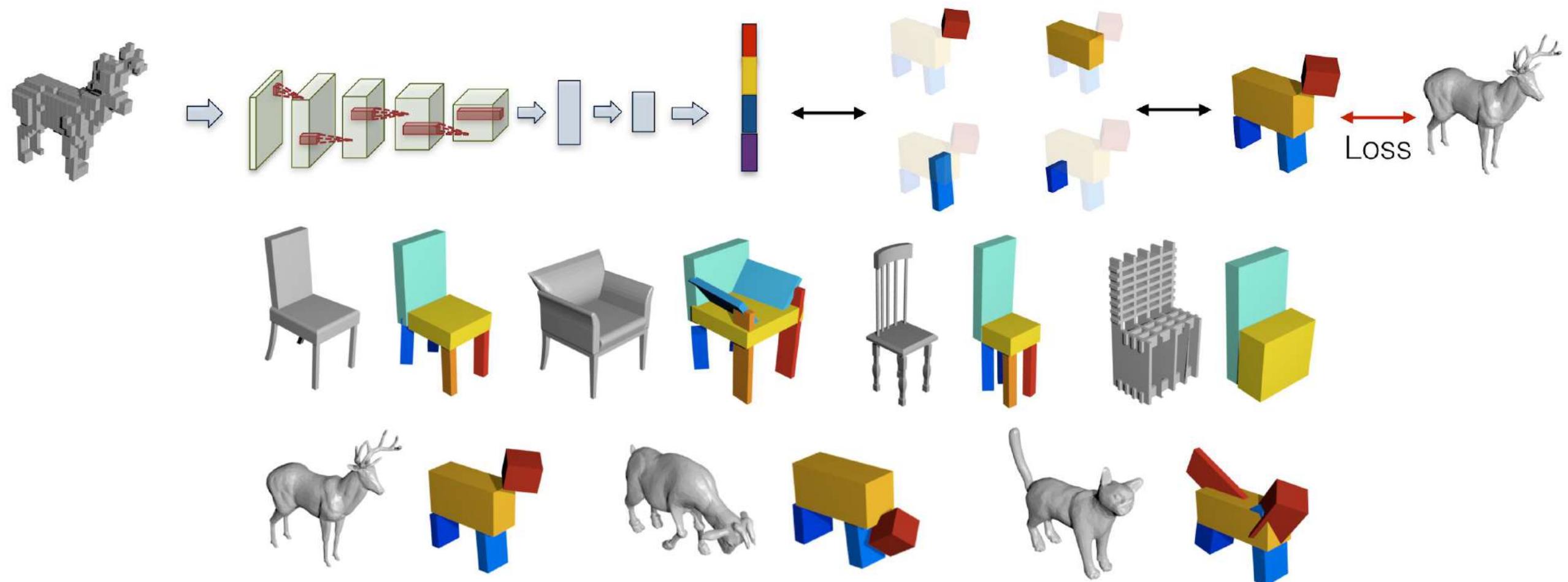
AtlasNet



FoldingNet

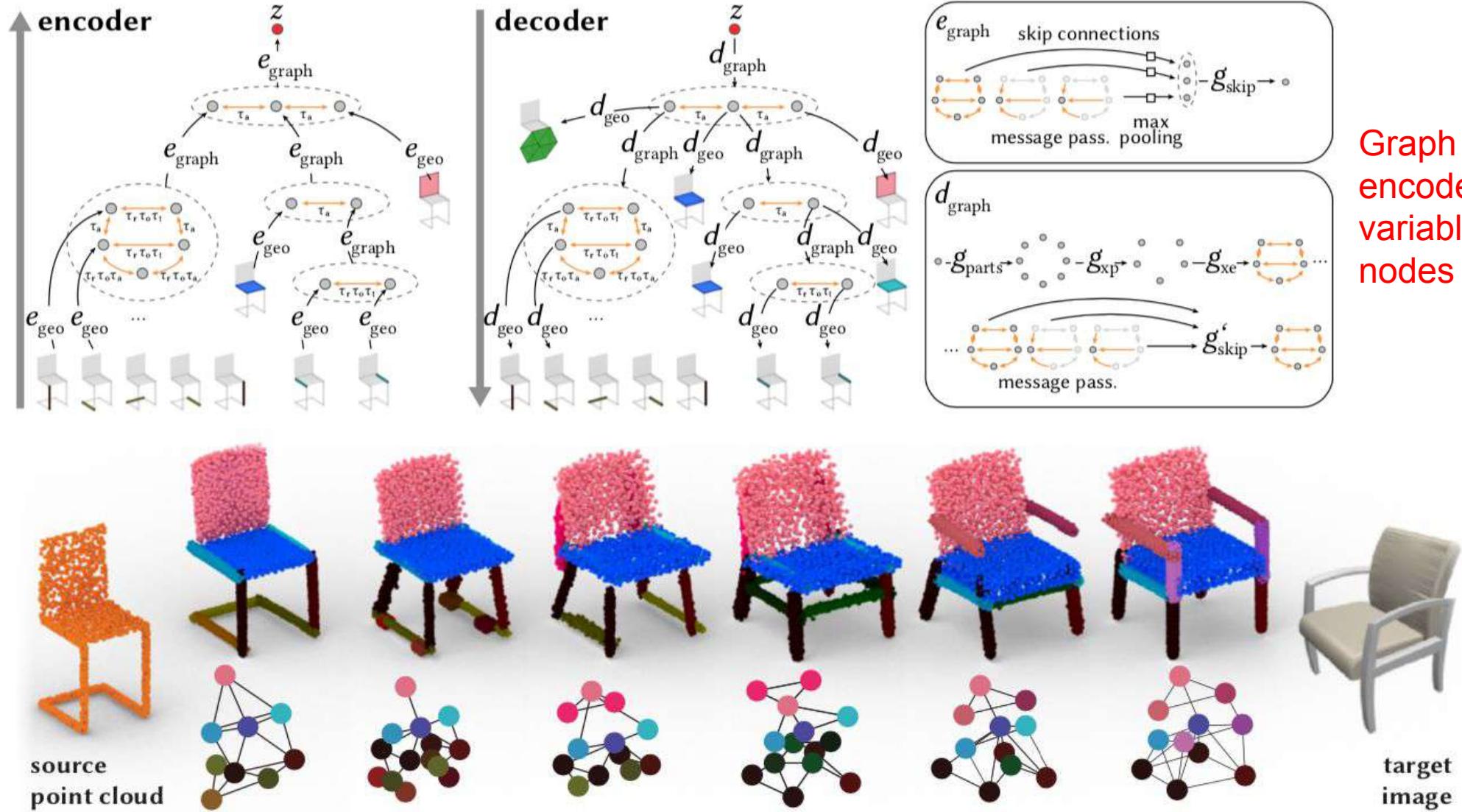


Assembling Volumetric Primitives

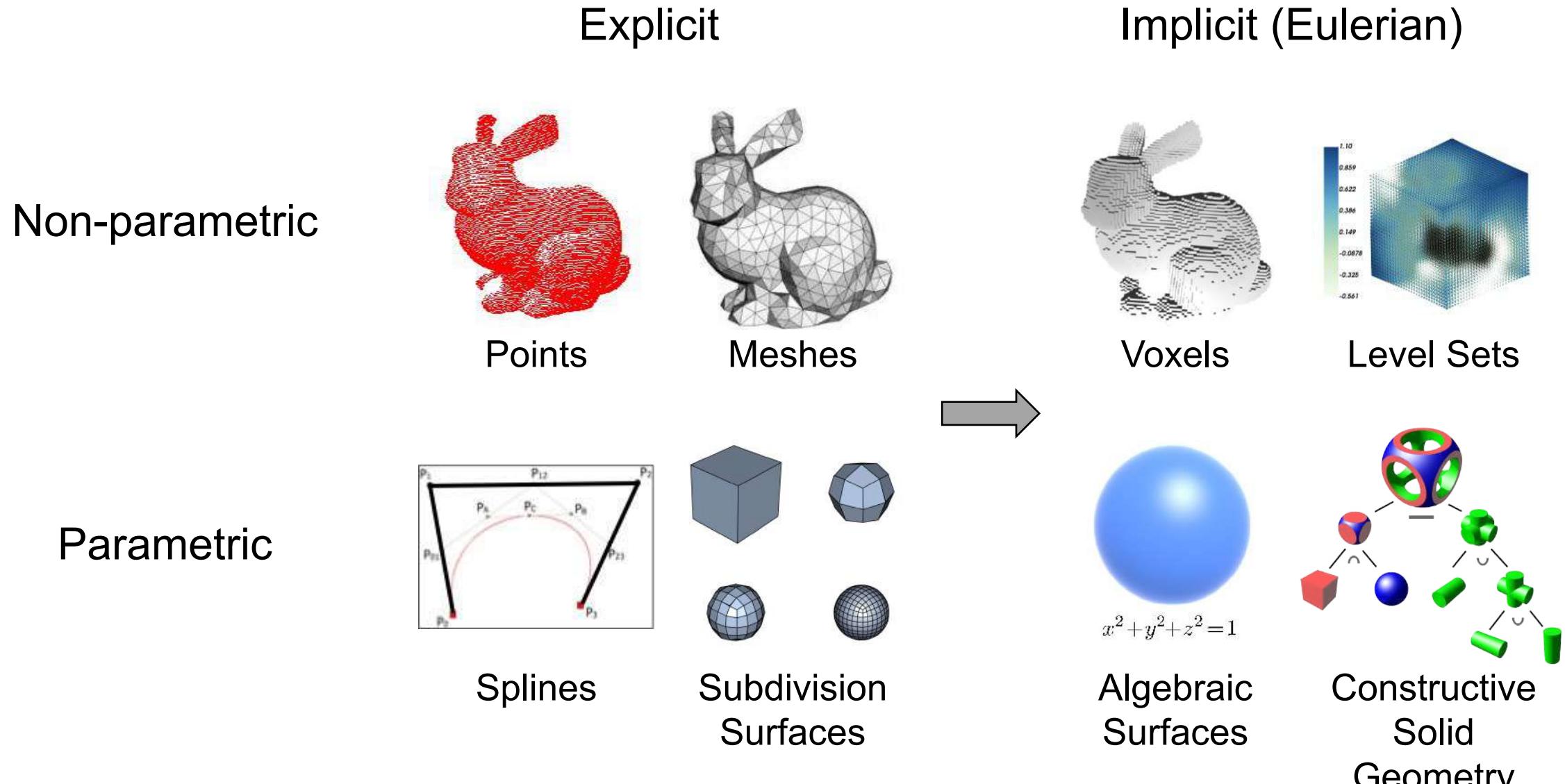


Incorporating Shape Structure

StructureNet



Explicit \rightarrow Implicit

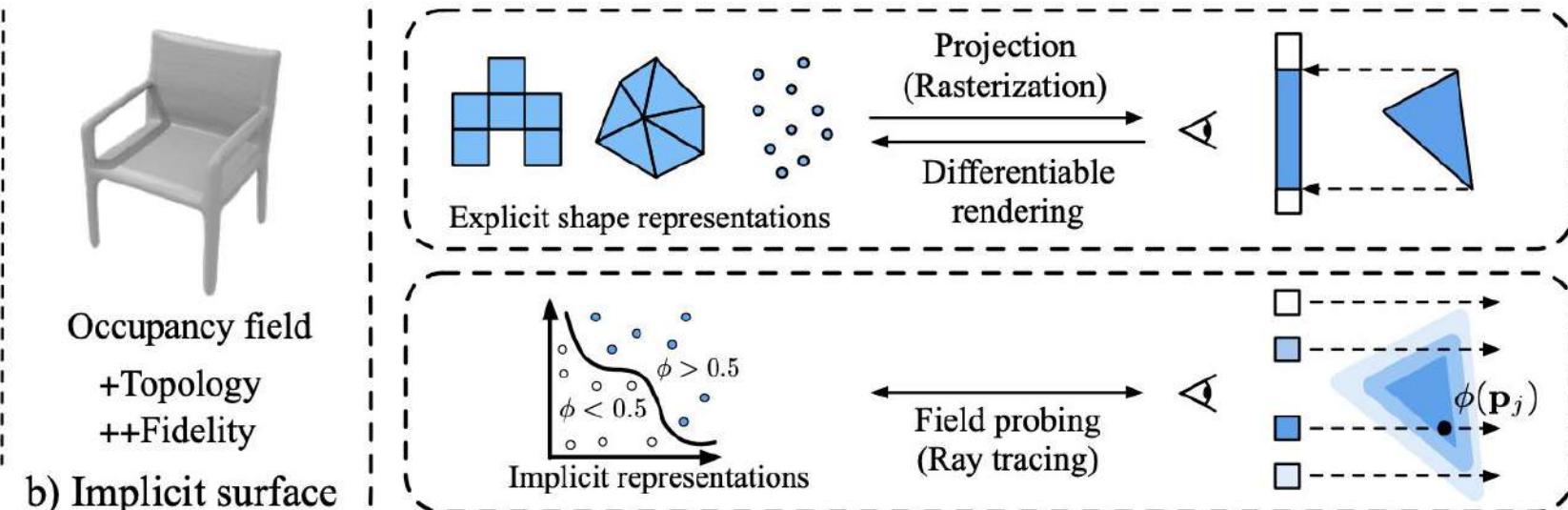


Deep Implicit Functions



Voxel
+Topology
-Fidelity
Point cloud
+Topology
-Fidelity
Mesh
-Topology
+Fidelity

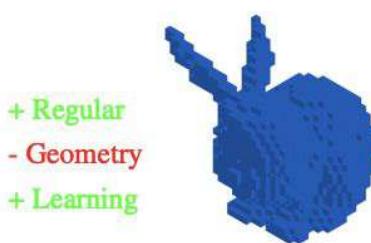
a) Explicit representation



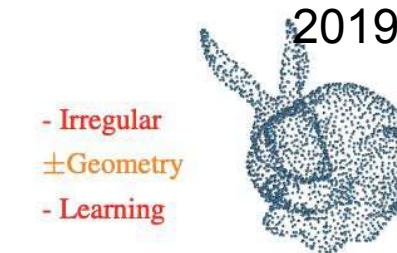
Liu et al. Learning to Infer Implicit Surfaces without 3D Supervision. NeurIPS



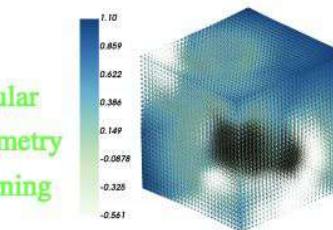
(a) Explicit representations



(b) Voxels



(c) Point cloud



(d) Level set

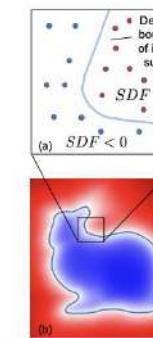
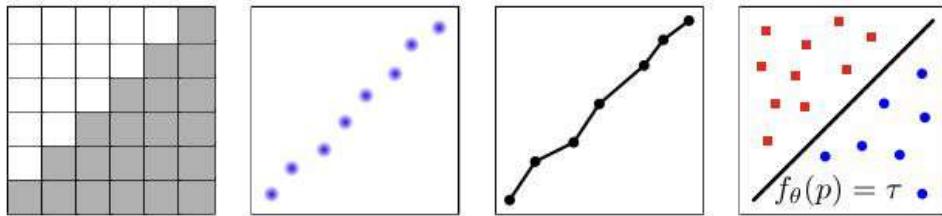


Figure 2. Four common representations of 3D shape along with their advantages and disadvantages.

Deep Level Sets: Implicit Surface Representations for 3D Shape Inference.
2019

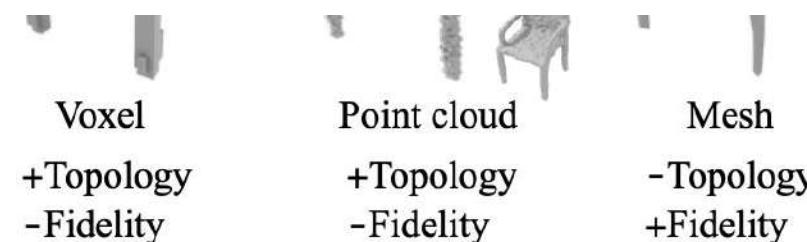
DeepSDF. CVPR
2019



Occupancy Networks
CVPR 2019



(a) Voxel (b) Point (c) Mesh (d) Ours

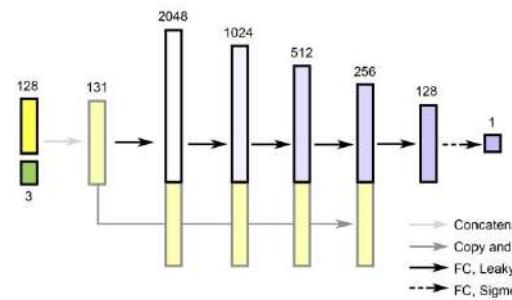


a) Explicit representation

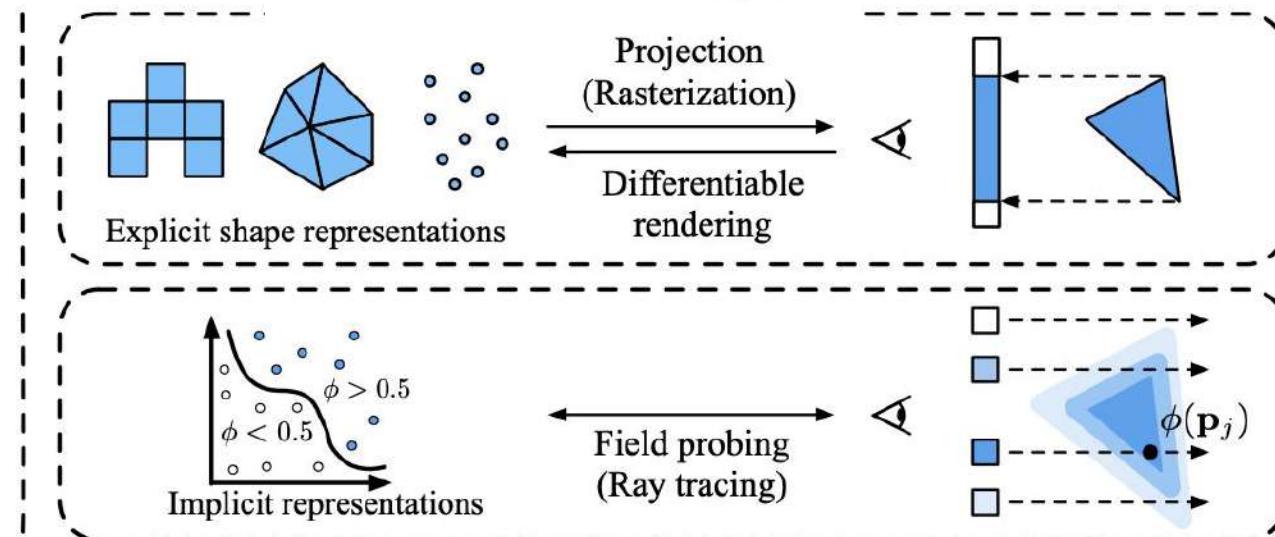


Occupancy field
 +Topology
 ++Fidelity

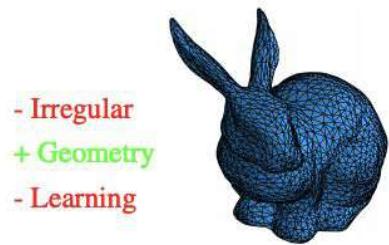
b) Implicit surface



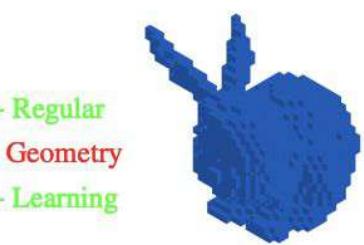
Chen and Zhang.
Learning Implicit
Fields
CVPR 2019



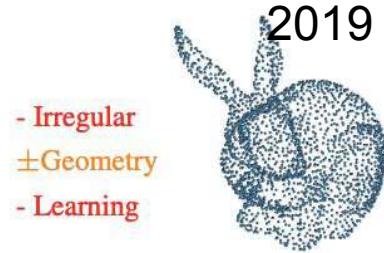
Liu et al. Learning to Infer Implicit Surfaces without 3D Supervision. NeurIPS



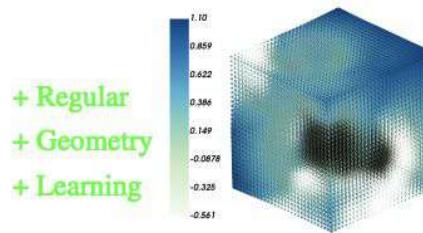
(a) Explicit representations



(b) Voxels



(c) Point cloud



(d) Level set

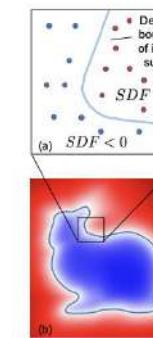


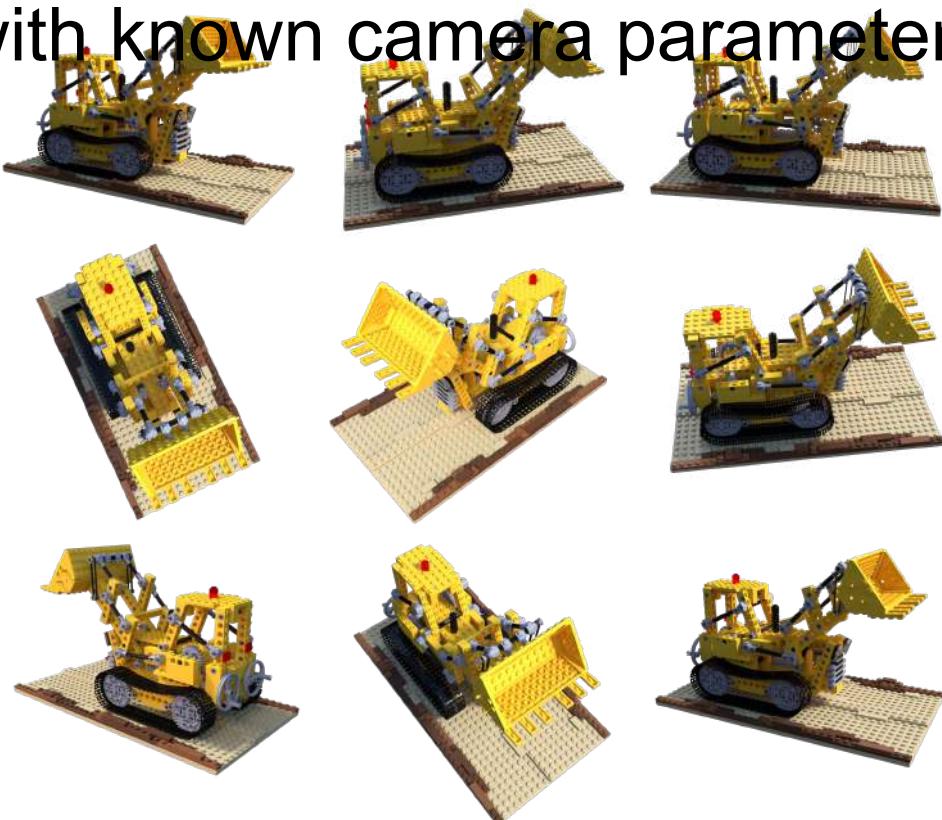
Figure 2. Four common representations of 3D shape along with their advantages and disadvantages.

Deep Level Sets: Implicit Surface Representations for 3D Shape Inference.
2019

DeepSDF. CVPR
2019

Neural Radiance Fields (NeRF) for View Synthesis

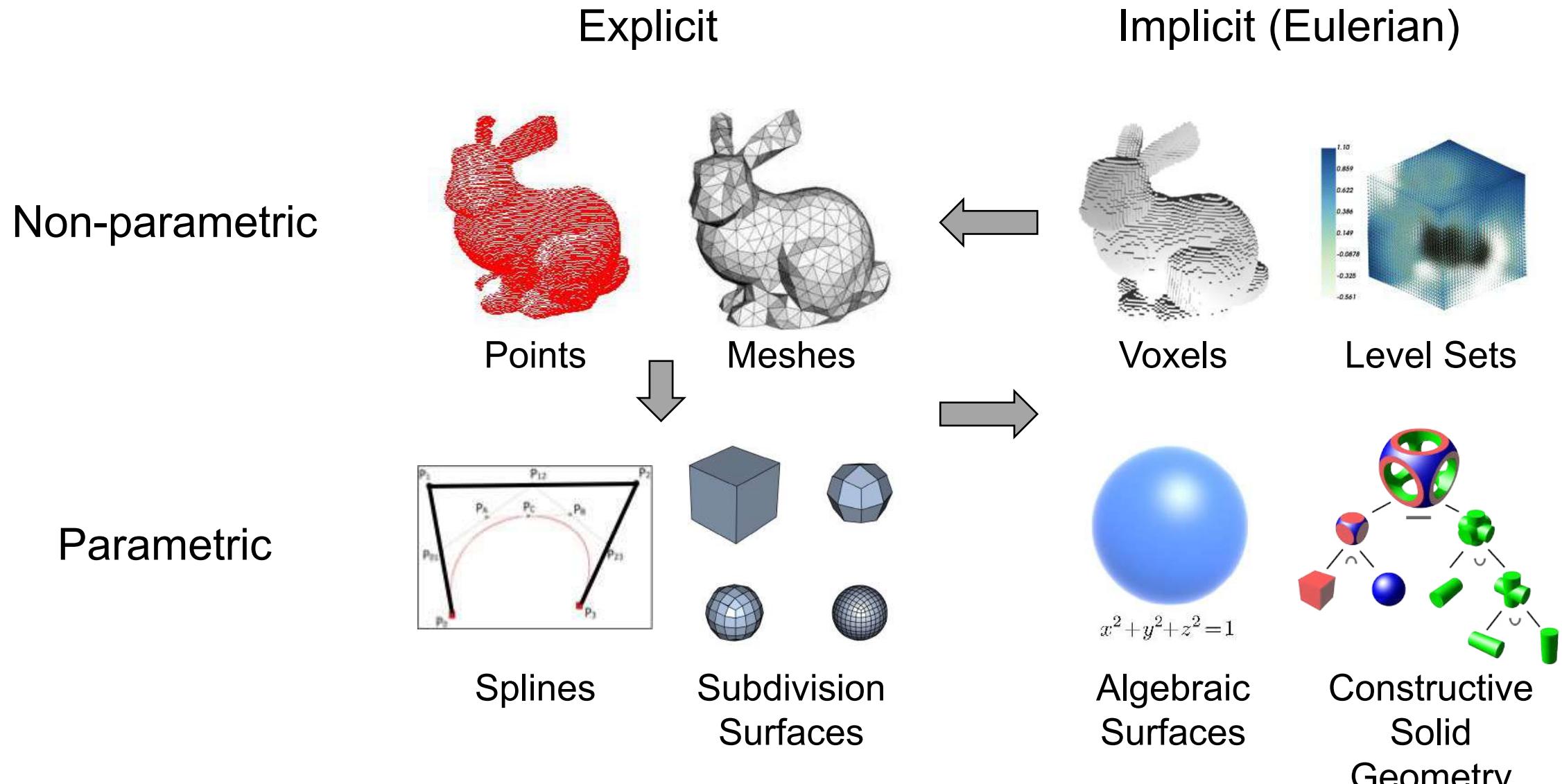
Input: Many images of the same scene
(with known camera parameters)



Output: Images showing the scene from novel viewpoints



Shape Representations



Next:
Human-Centered Artificial Intelligence