3D Vision

Overview

Geometric Deep Learning

What you will learn in this course

Low-Level Vision

Image Processing:
Photometric Image formation
Point Operations
Histogram Equalization
Linear & Non-Linear filters
Convolution & Cross correlation

Feature Detection:
Edge detection
Corner detection
Line detection

Feature Description:
SIFT
Feature Matching
Image Stitching
Panorama

Geometric Vision

Transformation and Camera Mode:
Geometric Image Formation
Pinhole Camera Model

Camera Calibration:
Estimating intrinsics
Estimating extrinsics

Visual Understanding

Object Detection & Tracking:
Single-stage detectors
Two-stage detectors
YOLO
Object tracking

Image Segmentation:
Traditional Image Segmentation
Learning-based Image Segmentation

Machine Learning for CV

Machine Learning for Classification:
k-NN
Linear Classifier

Deep Learning:
Optimization
Neural Network
CNN
CNN architectures:

Generative & Geometric Learning

Generative Models:
AE, VAE, GAN

3D Vision and 3D Deep Learning:
3D shape representations
Geometric deep learning overview
3D Deep Learning models : pointnet

What we will learn today

- ❑ 3D Representations
- ❑ Point Cloud Processing
- ❑ 3D Deep Learning
  - ❑ Multiview images
  - ❑ Voxels
  - ❑ Point Cloud
  - ❑ Polygon Mesh
  - ❑ Signed Distance Function (SDF)

3D Vision

3D, 3D Everywhere

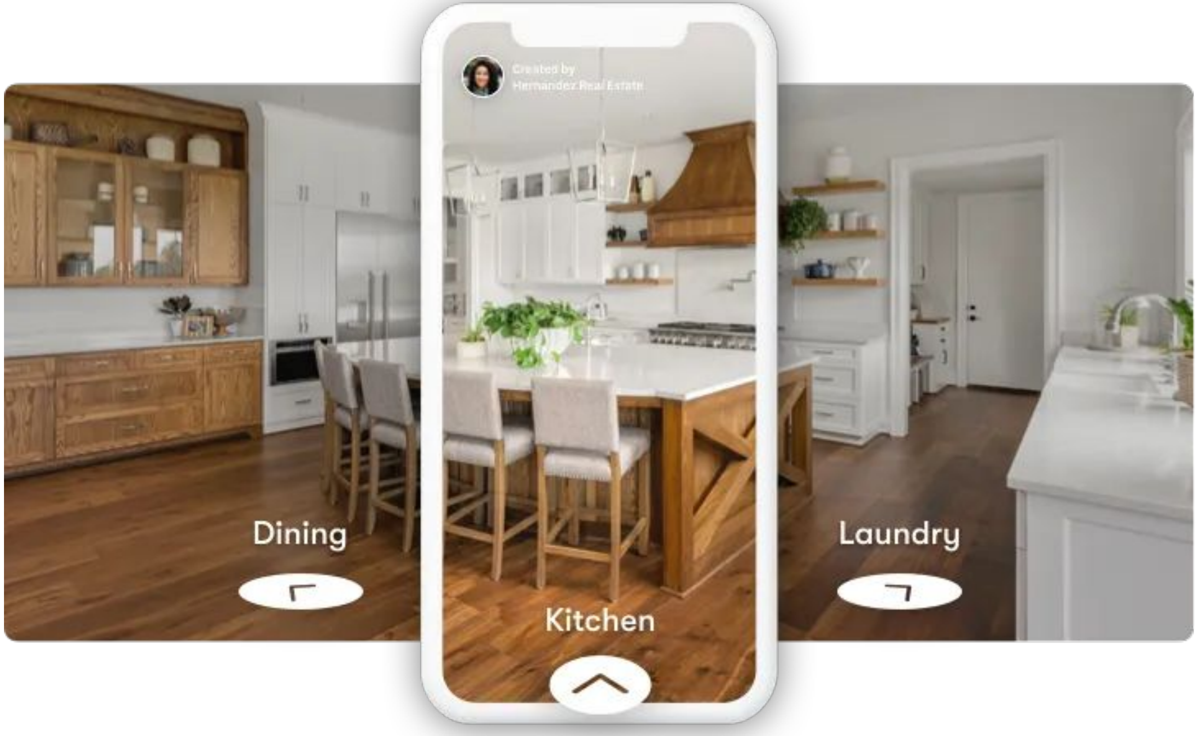


Image from: Zillow, 3D Virtual Tour

3D Vision

3D, 3D Everywhere

Image from: Advice

3D Vision

3D, 3D Everywhere

3D Vision

3D, 3D Everywhere

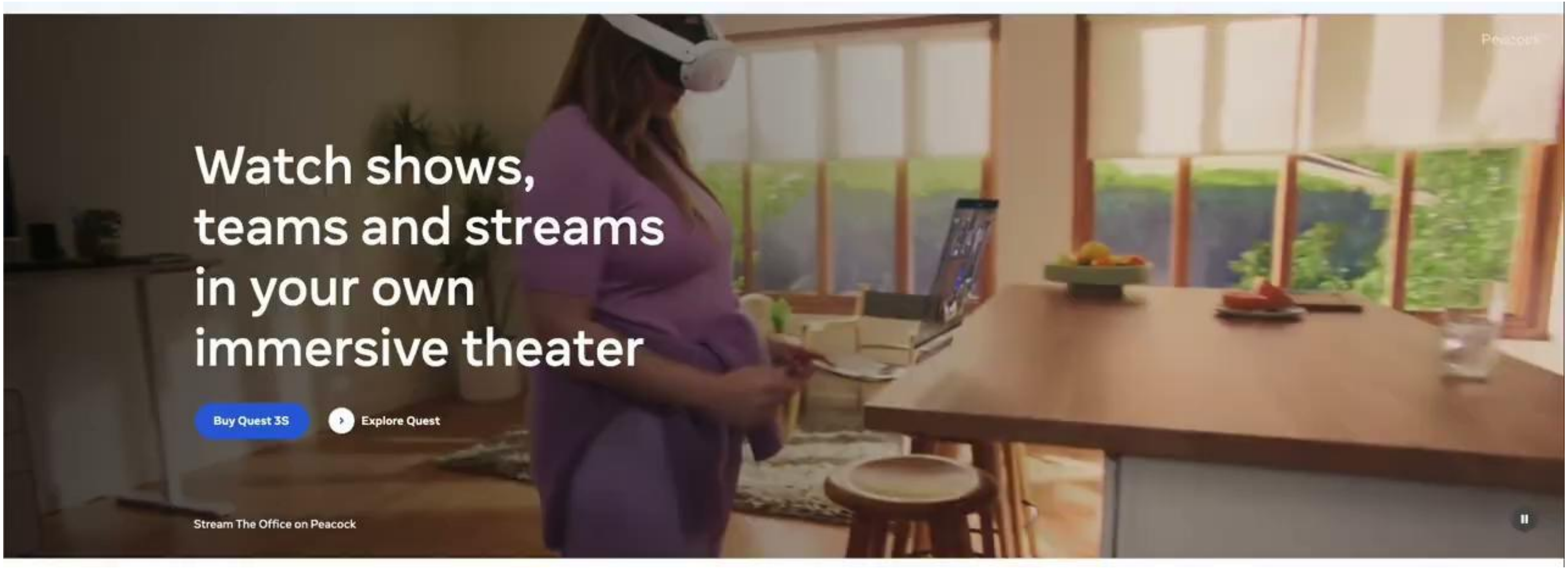
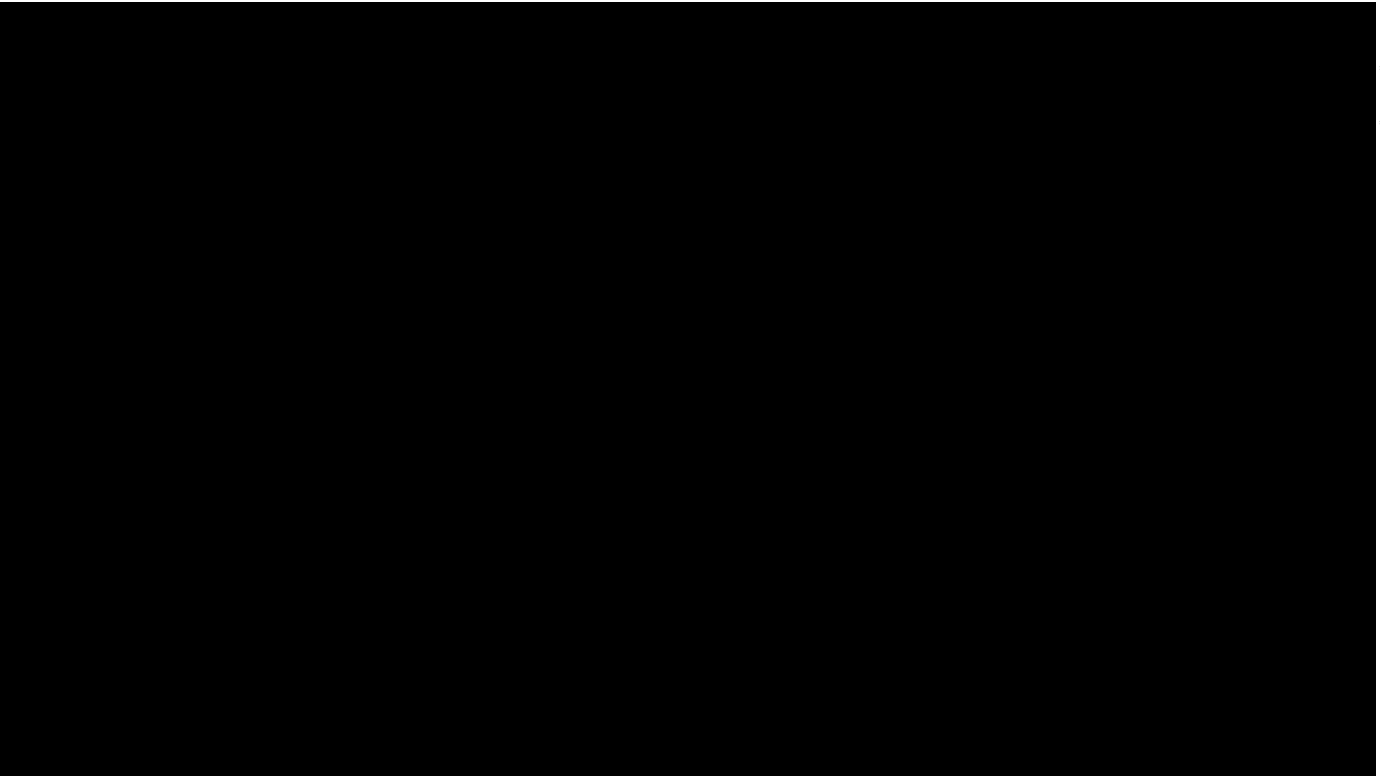
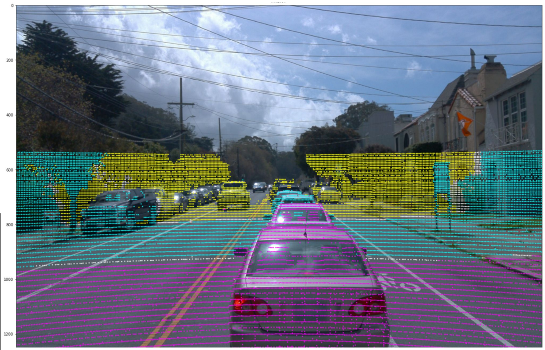


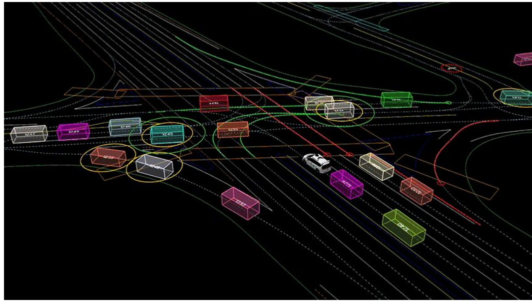
Image from: Meta Quest

3D Vision

3D, 3D Everywhere







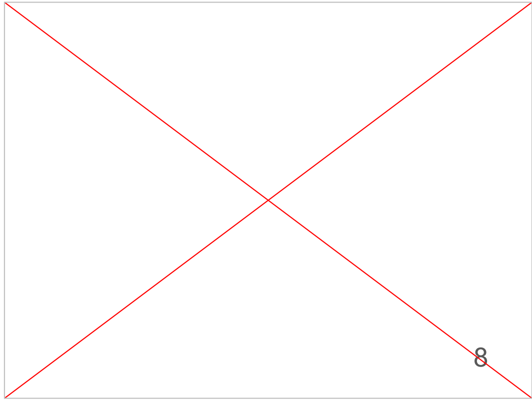


Image from: Waymo

3D Vision

3D, 3D Everywhere

Image from: Creaform3d

3D Representations

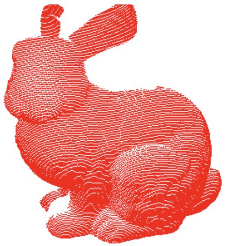
3D Representation

Many ways to represent geometry

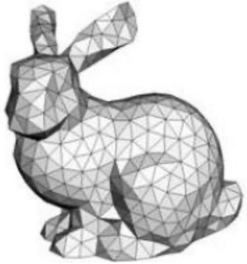
Explicit

Implicit

Non-parametric

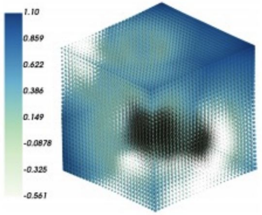


Points



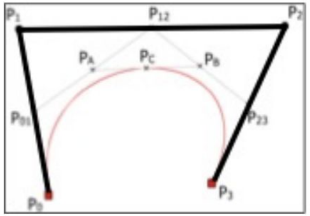
Meshes

Voxels

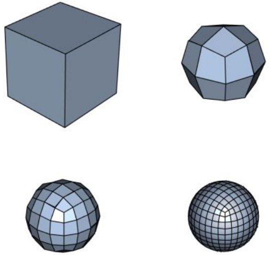


Level Sets

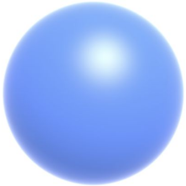
Parametric



Splines

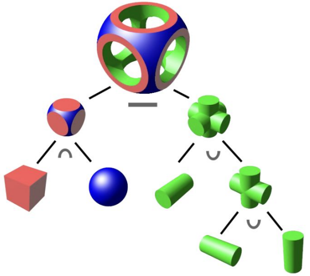


Subdivision Surfaces



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

Algebraic Surfaces



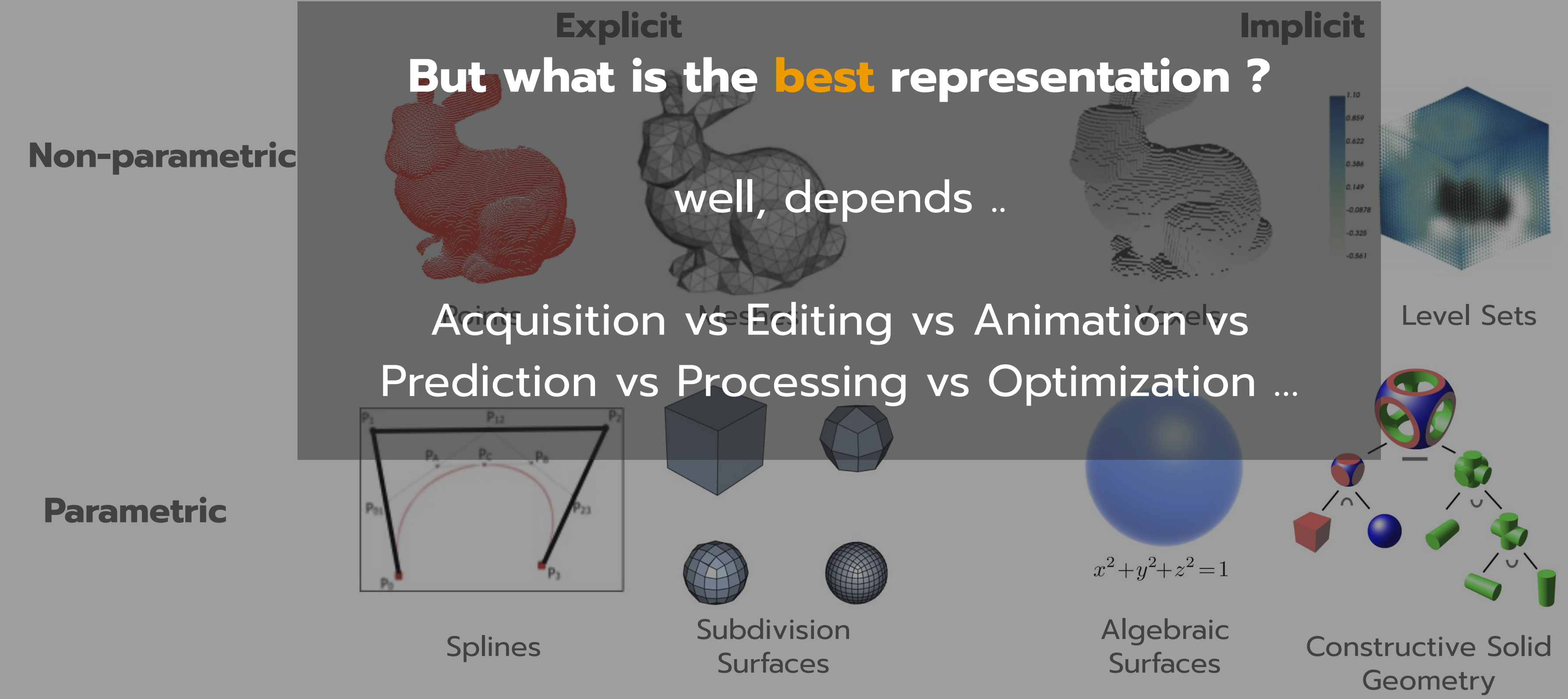
Constructive Solid Geometry

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

Cherdsak Kingkan

3D Representation

Many ways to represent geometry



But what is the best representation ?

well, depends ..

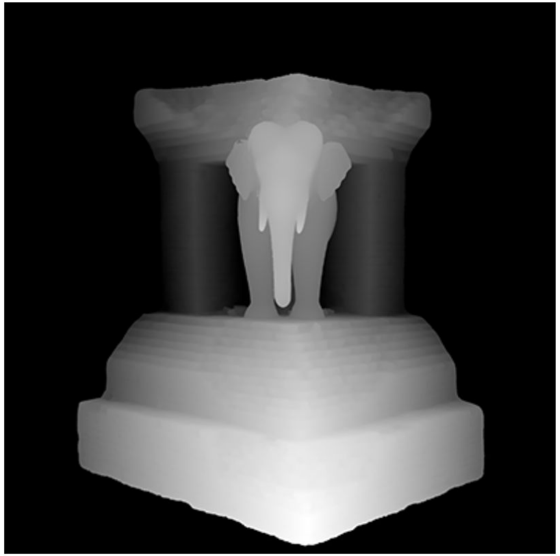
Acquisition vs Editing vs Animation vs Prediction vs Processing vs Optimization …

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

Cherdsak Kingkan

3D Representation

2.5D Representation: Depth Maps

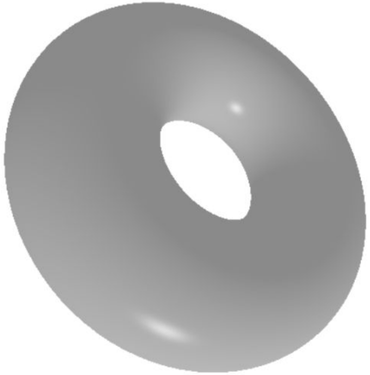
 

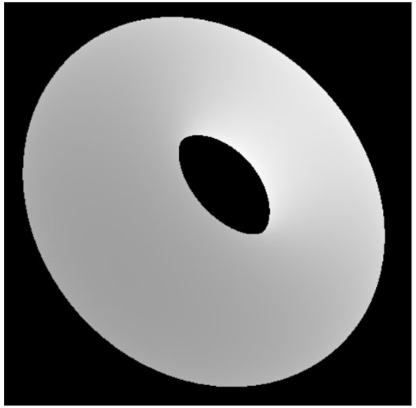
An **image** that represents how far each pixel ***p*** is $D[\mathbf{p}] \in \mathbb{R}^+$

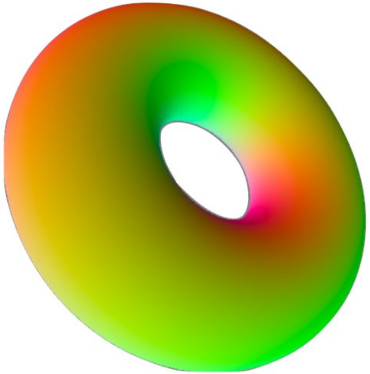
The 'standard' image processing tools can be used

3D Representation

2.5D Representation: Depth Maps





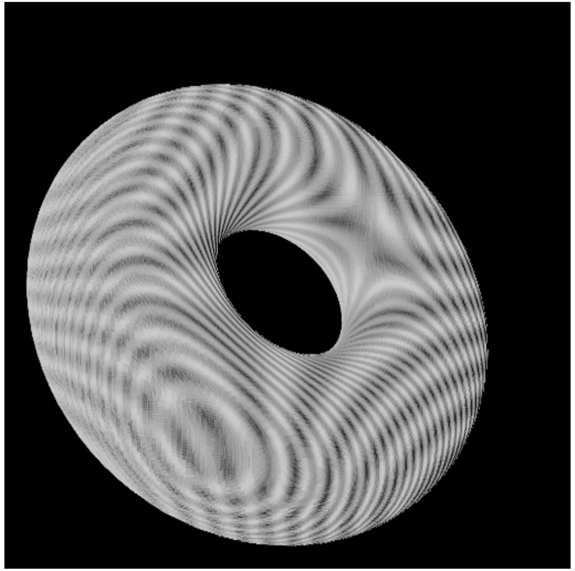


$$N[\mathbf{p}] \in \mathbb{S}^2$$

Normal Maps: Can extend to capture other surfaceproperties e.g. surface normals

3D Representation

2.5D Representation: Depth Maps



Representing the Visible

- Does not capture the 'full' 3D Structure
- '2.5D' representations – properties associated to image pixels

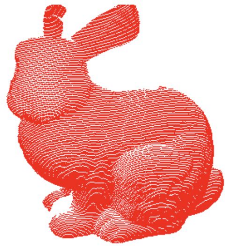
3D Representation

Many ways to represent geometry

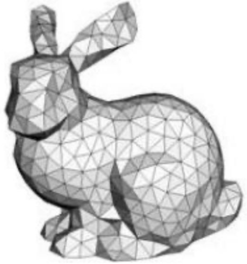
Explicit

Implicit

Non-parametric

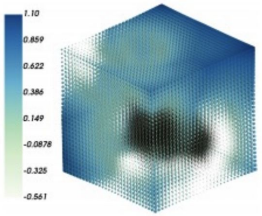


Points



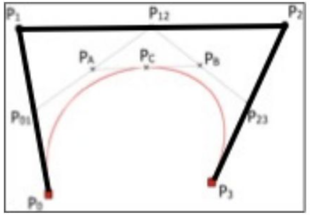
Meshes

Voxels

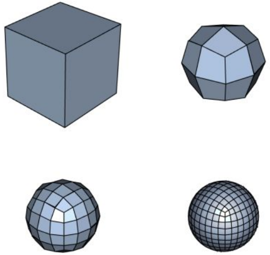


Level Sets

Parametric



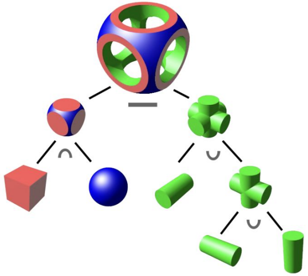
Splines



Subdivision Surfaces

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

Algebraic Surfaces



Constructive Solid Geometry

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025
Stanford University

Cherdsak Kingkan

3D Representation

Point Cloud

What is Point Cloud?

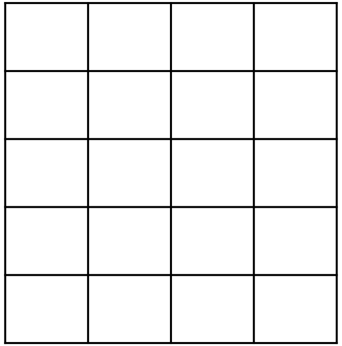
Point cloud is a set of points which can be represented by its coordinates (x, y, z) in 3D space.

3D Representation

Point Cloud

Organized Point Cloud

- Data is split into rows and columns.
- Advantages
  - Knowing relationship between points
  - Speed up the computation of some algorithm



Unorganized Point Cloud

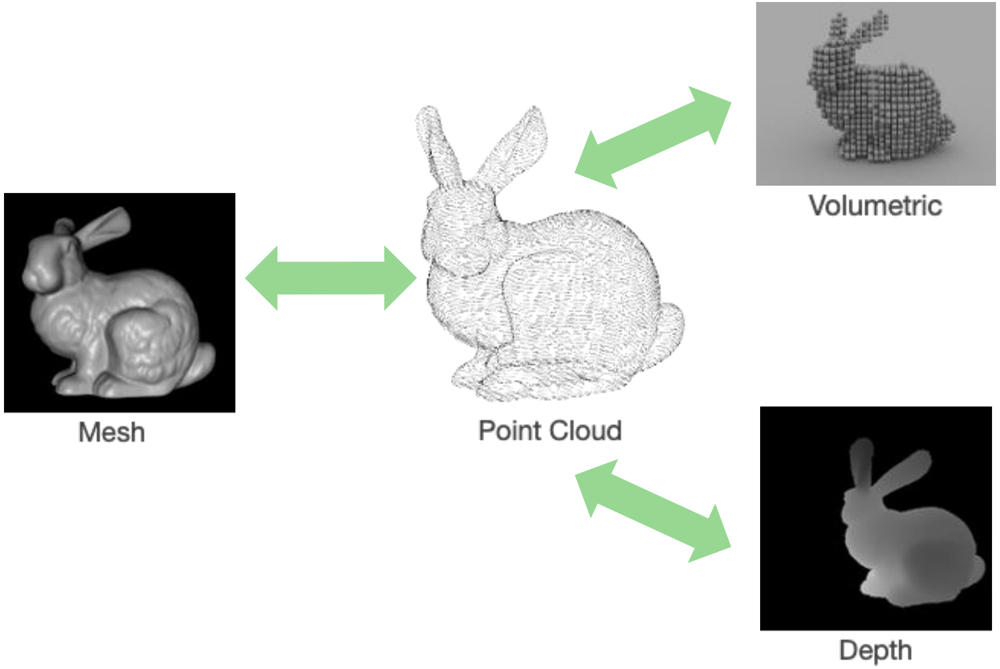
- The order of points in the data structure is random.

3D Representation

Point Cloud

Why Point Cloud

- Point cloud is close to raw sensor data and **easy to acquired** by low-cost 3D scanning sensors
- Point cloud is **simple, flexible and scalable** representation: only points, no connectivity.
- Point cloud is **canonical**

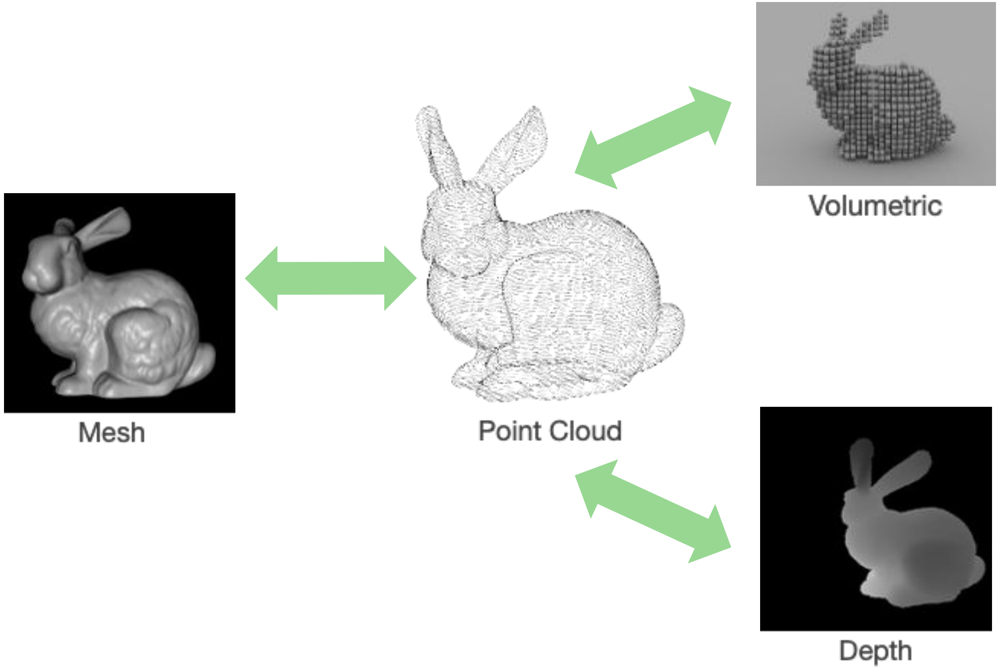


3D Representation

Point Cloud

Why Point Cloud

- Point cloud is close to raw sensor data and **easy to acquired** by low-cost 3D scanning sensors
- Point cloud is **simple, flexible and scalable** representation: only points, no connectivity.
- Point cloud is **canonical**



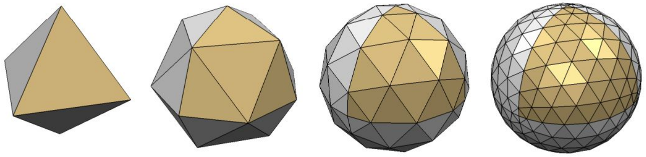
Limitations

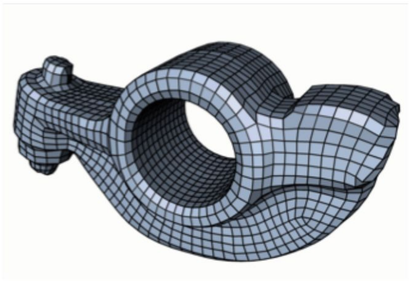
- No explicit 'connectivity' information
- No simplification or subdivision
- No direction smooth rendering

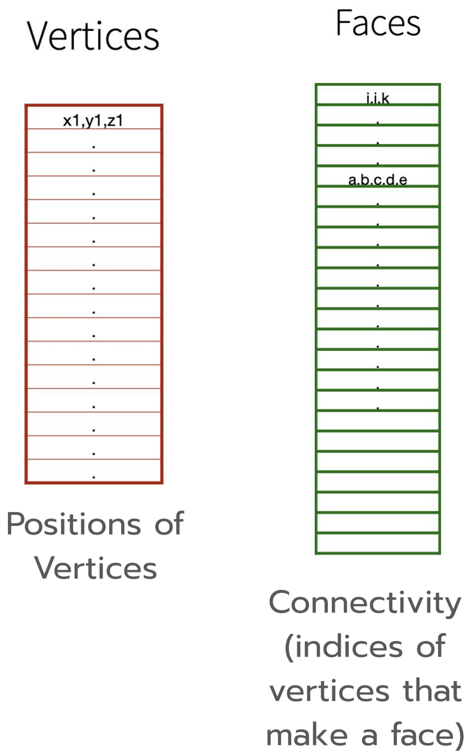
3D Representation

Polygonal Meshes

- Consists of vertices and faces
- Boundary representations of objects
- Piecewise linear approximations of underlying surface
- Triangular Meshes: a face consists of **<u>three</u>** vertices





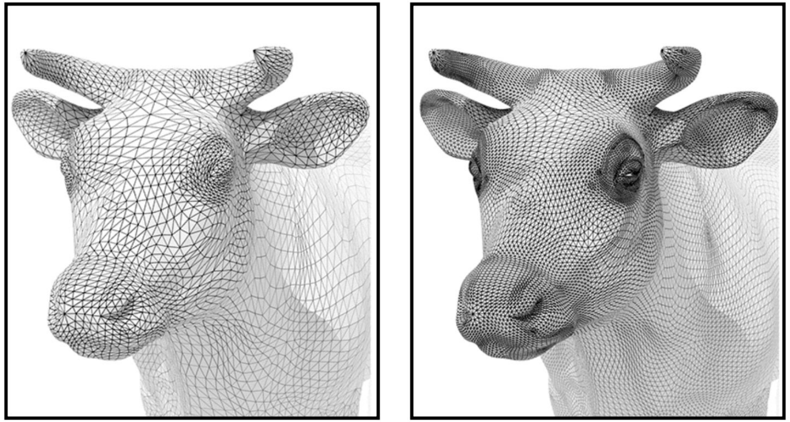


Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

3D Representation

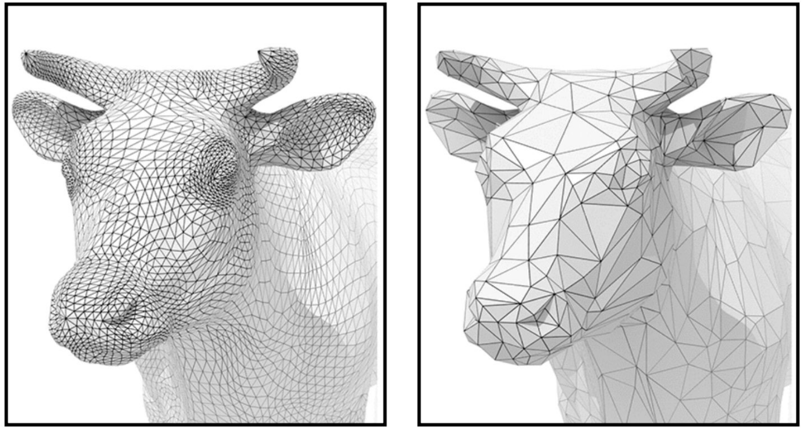
Polygonal Meshes - Mesh operations

Mesh Upsampling - Subdivision



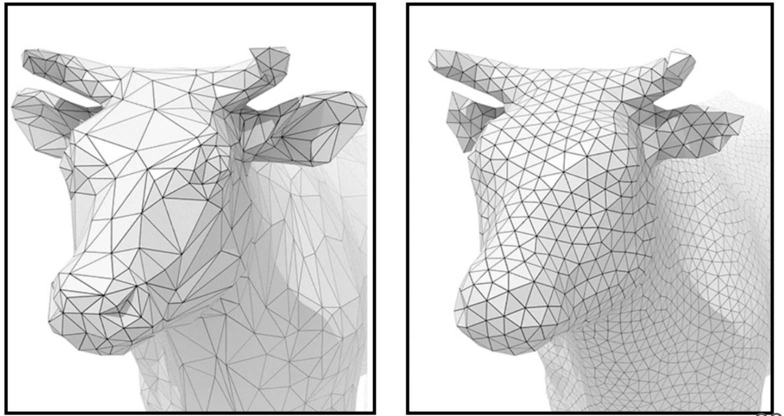
Increase resolution via interpolation

Mesh Downsampling - Simplification



Decrease resolution; try to preserve shape/appearance

Mesh Regularization

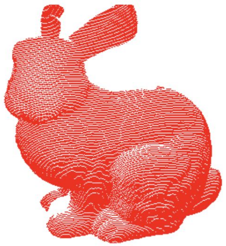


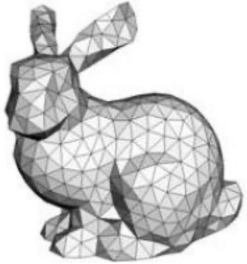
Modify sample distribution to improve quality

3D Representation

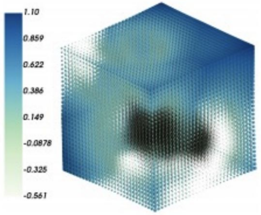
Many ways to represent geometry

	Explicit		Implicit	
Non-parametric	Points	Meshes	Voxels	Level Sets
Parametric	Splines	Subdivision Surfaces	Algebraic Surfaces	Constructive Solid Geometry

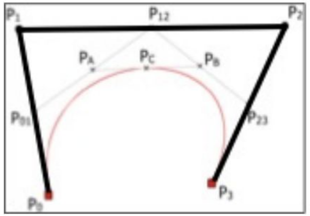

Points


Meshes

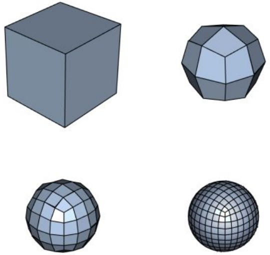

Voxels



Level Sets

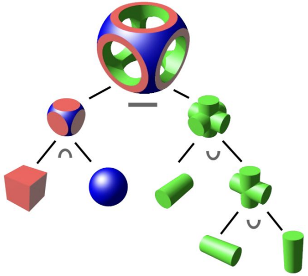


Splines


Subdivision Surfaces


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

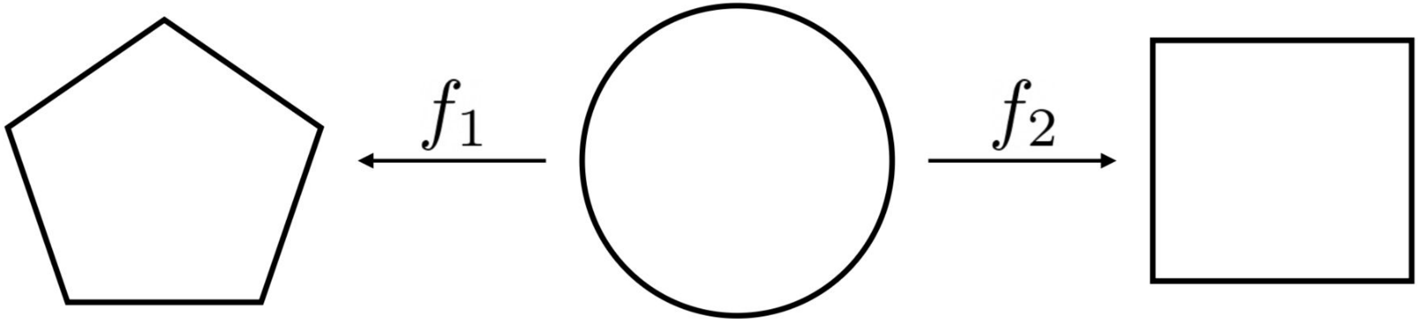
Algebraic Surfaces


Constructive Solid Geometry

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

3D Representation

Parametric Representation : Parametric Surfaces in 2D



$$f(\mathbf{u}) = \mathbf{p} \in \mathbb{R}^2; \ \mathbf{u} \in \mathbb{S}^1$$

3D Representation

Parametric Representation : Parametric Curves

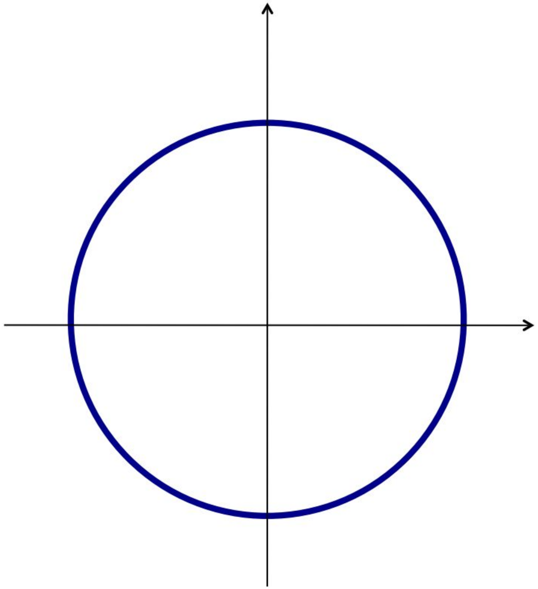
Explicit curve/circle in 2D

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

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

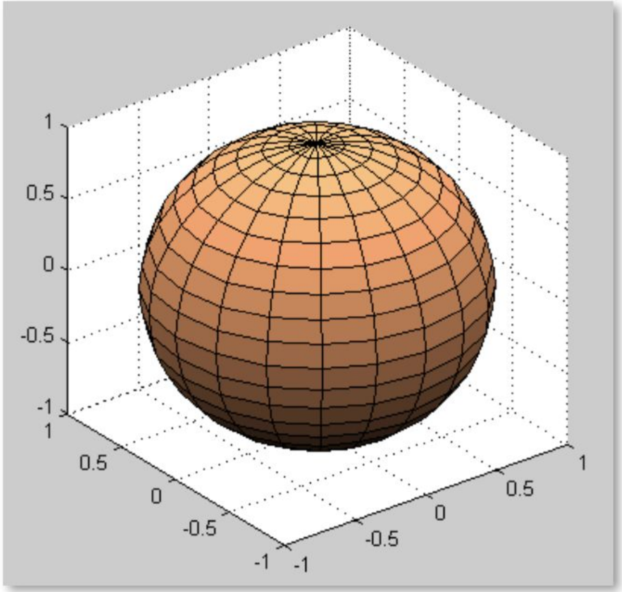
$$\mathbf{p}(t) = r\,(\cos(t), \sin(t))$$

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



3D Representation

Parametric Representation : Parametric Surfaces



Sphere in 3D

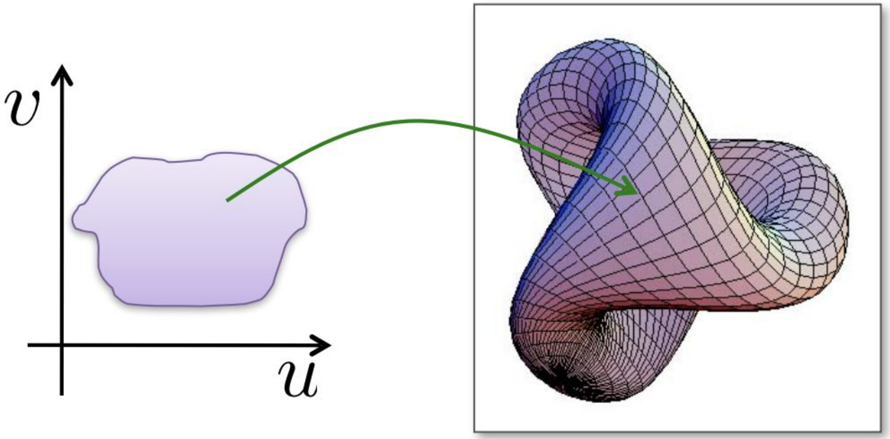
$$s : \mathbb{R}^2 \to \mathbb{R}^3$$

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

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

3D Representation

Parametric Representation : Parametric Surfaces



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

Range of a function :

$$f : X \to Y, X \subseteq \mathbb{R}^m, Y \subseteq \mathbb{R}^n$$

Surface in 3D : $m = 2, n = 3$

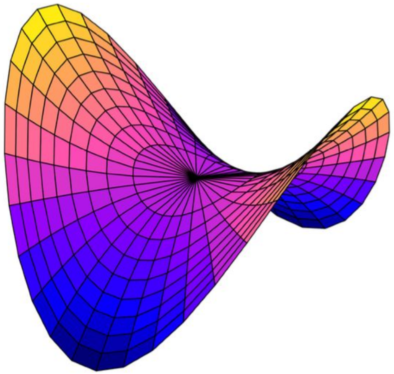
A continuous function *f* over a 2D manifold *M* can parametrize a surface.

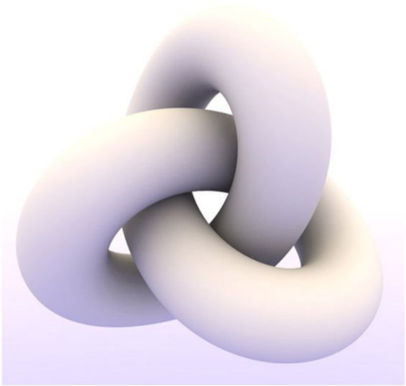
```
def f(u):
    '''
    Input:
        u: N-dimenstional vector from a 2D manifold
    Output:
        p: 3D point
    '''
```

Can be fixed analytics functions, parameterized family, or even a neural network!

3D Representation

Parametric Representation : Parametric Surfaces



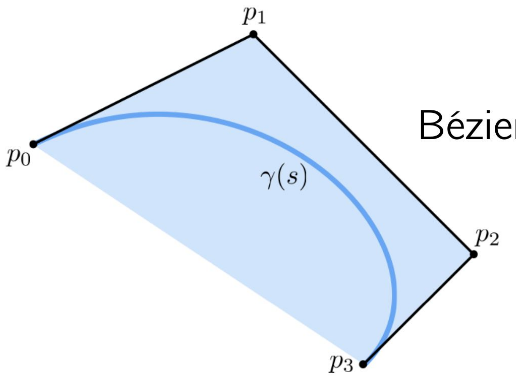


$$f((u,v)) = (u, v, v^2 - u^2)$$

```python
def trefoil(u, v):
    x = r * np.sin(3 * u) / (2 + np.cos(v))
    y = r * (np.sin(u) + 2 * np.sin(2 * u)) / (2 + np.cos(v + np.pi * 2 / 3))
    z = r / 2 * (np.cos(u) - 2 * np.cos(2 * u)) * (2 + np.cos(v)) * (2 + np.cos(v + np.pi * 2 / 3)) / 4
```

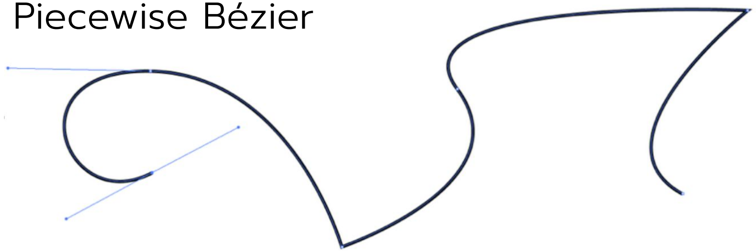
3D Representation

Parametric Representation : Bezier Curves/Surfaces

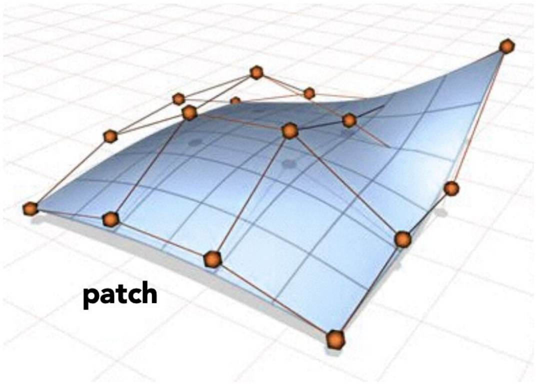


Bézier curves

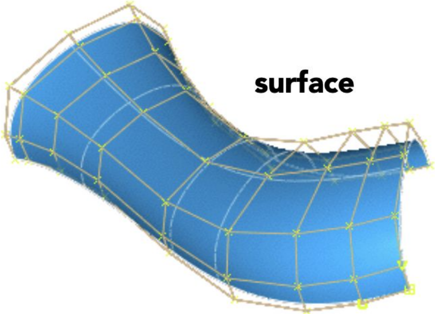
Piecewise Bézier



Use tensor product of Bézier curves to get a patch:



Multiple Bézier patches form a surface.

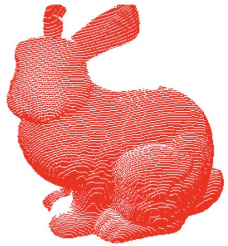


Adopted from: Shubham Tulsiani, Learning for 3D Vision, Fall 2025
Carnegie Mellon University

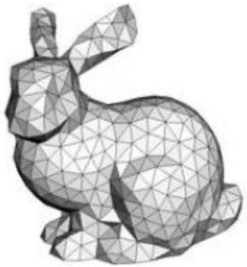
3D Representation

Many ways to represent geometry

	Explicit		Implicit	
Non-parametric	Points	Meshes	Voxels	Level Sets
Parametric	Splines	Subdivision Surfaces	Algebraic Surfaces	Constructive Solid Geometry

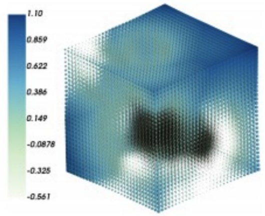


Points

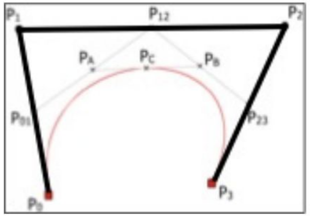


Meshes

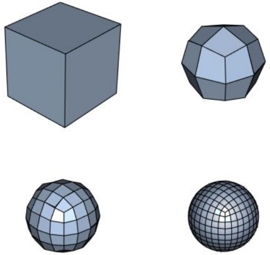
Voxels



Level Sets



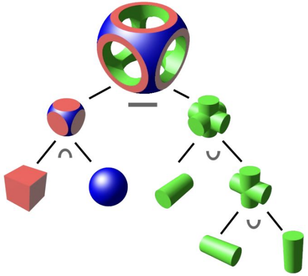
Splines



Subdivision Surfaces

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

Algebraic Surfaces



Constructive Solid Geometry

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025
Stanford University

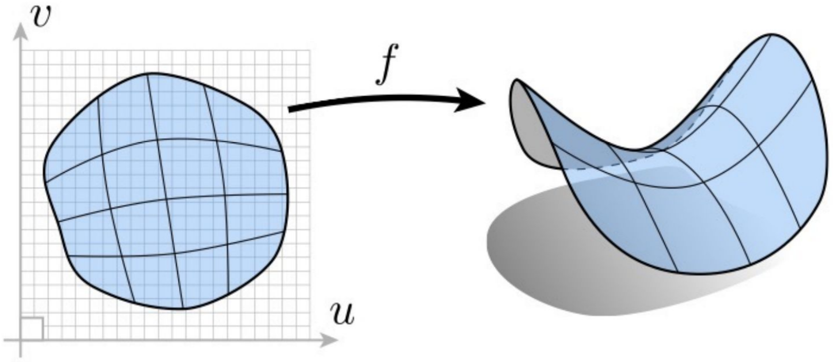
Cherdsak Kingkan

3D Representation

Explicit Representation of Geometry

All points are given directly.

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



3D Representation

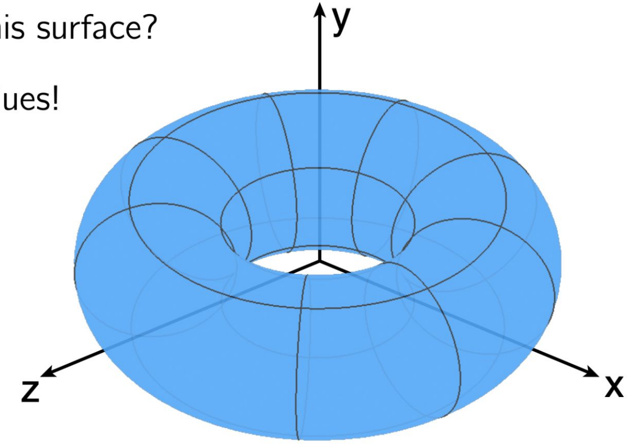
Explicit Representation of Geometry

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!



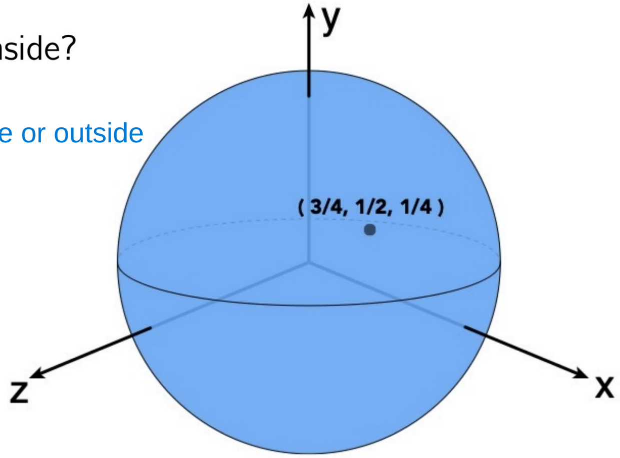
Explicit representations make some tasks easy.

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?

we don't know if it is inside or outside



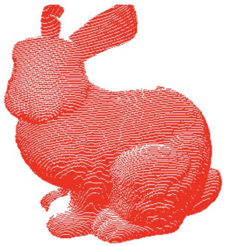
Some tasks are hard with explicit representations.

3D Representation

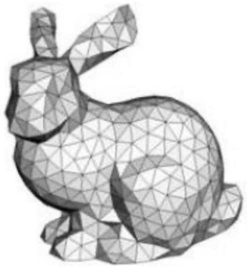
Many ways to represent geometry

	Explicit		**Implicit**	

Non-parametric

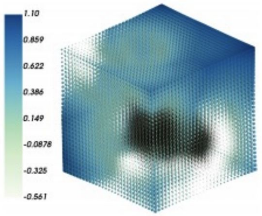


Points



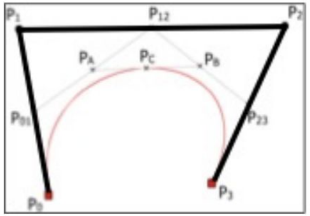
Meshes

Voxels

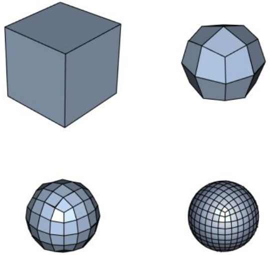


Level Sets

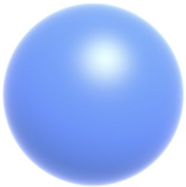
Parametric



Splines

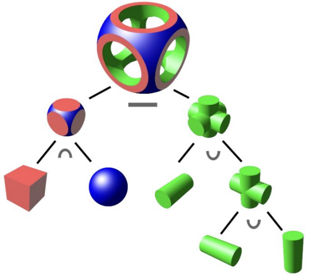


Subdivision Surfaces



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

Algebraic Surfaces



Constructive Solid Geometry

3D Representation

Implicit Representation of Geometry

Based on classifying points

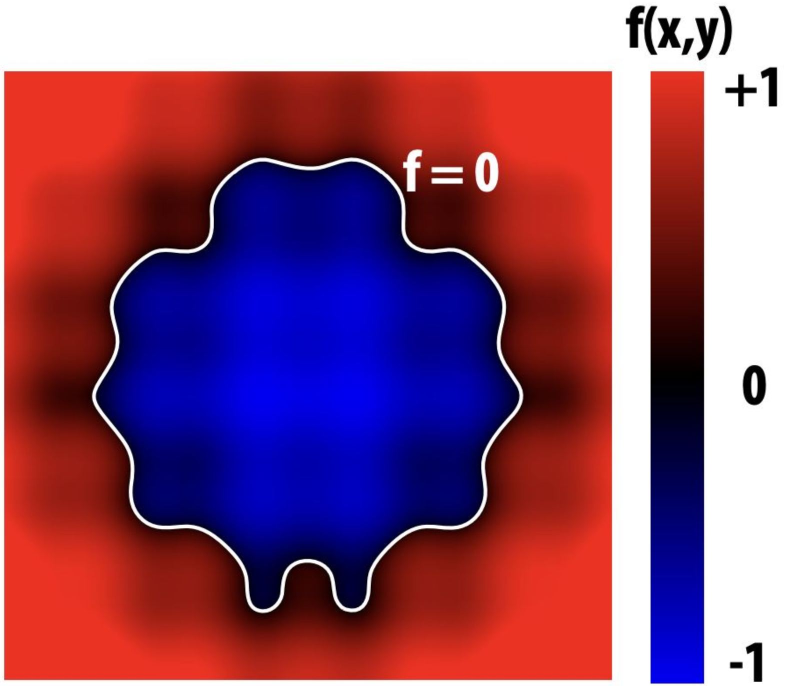
- Points satisfy some specified relationship.

E.g., sphere: all points in 3D, where

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

More generally,

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



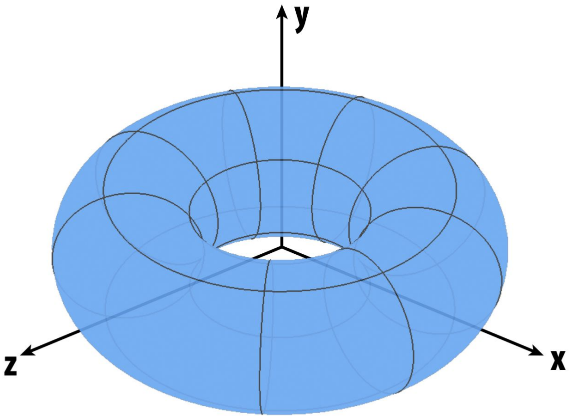
3D Representation

Implicit Representation of Geometry

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.

Inside/Outside Test **Easy**.

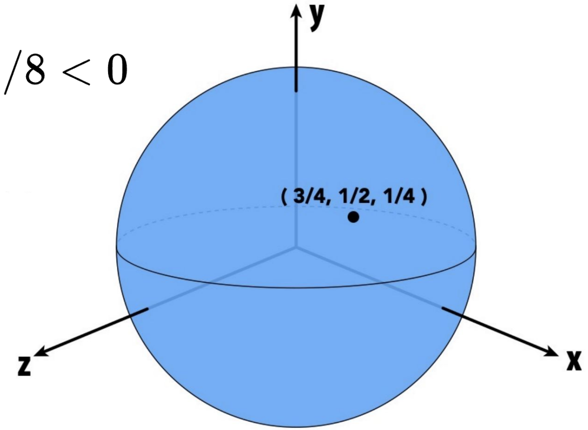
$$f(x,y,z) = x^2 + y^2 + z^2 - 1$$

Is (¾, ½, ¼) inside?

Just plug it in:

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

Yes, inside.



Implicit representations make **some tasks easy**.

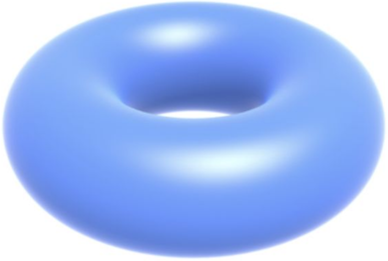
Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025
Stanford University

3D Representation

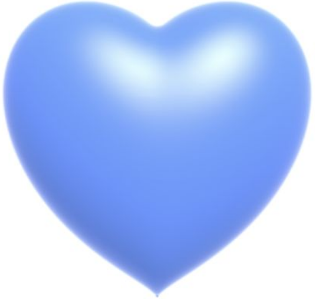
Implicit Representation : Algebraic Surfaces

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

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



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



$$(x^2+\frac{9y^2}{4}+z^2-1)^3 = x^2z^3+\frac{9y^2z^3}{80}$$

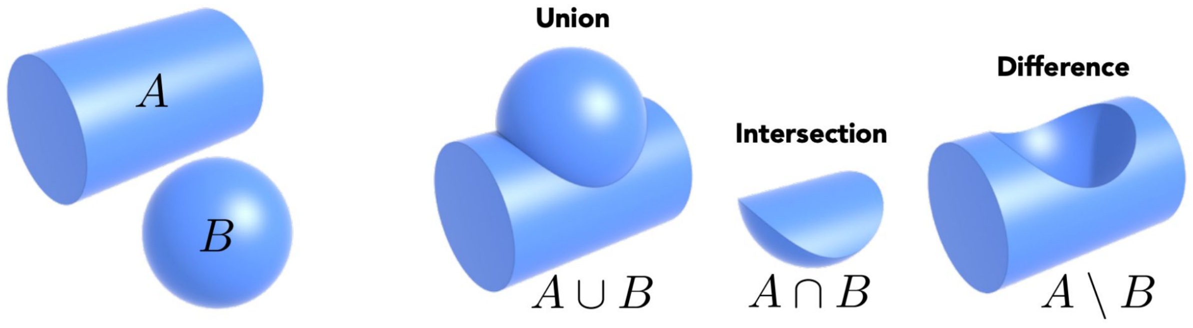
More complex shapes??

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

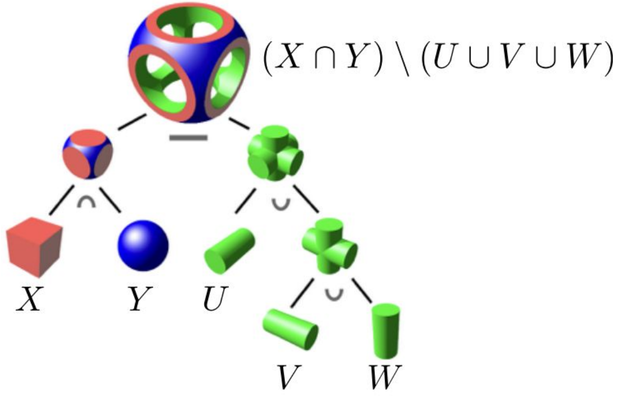
3D Representation

Implicit Representation : Algebraic Surfaces

Combine implicit geometry via Boolean operations.

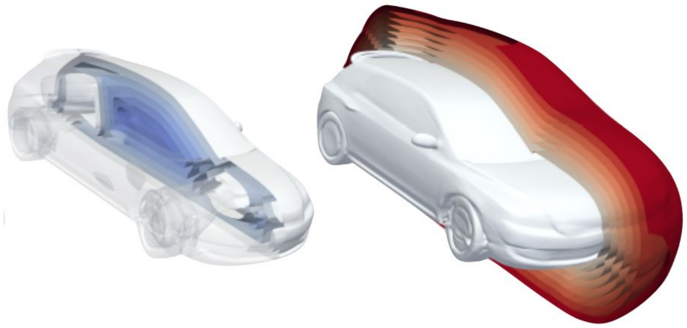


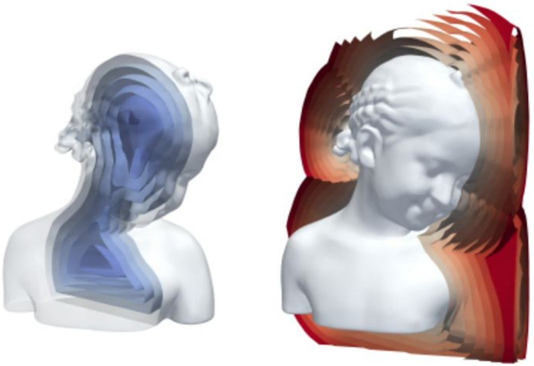
Boolean expressions:



3D Representation

Implicit Representation : Signed Distance Function





$$\{\mathbf{p} \mid f(\mathbf{p})\} = 0$$

Additional condition that ***f*** represents (signed) distance to surface from point ***p***

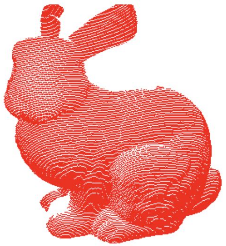
3D Representation

Many ways to represent geometry

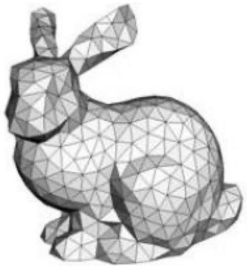
Explicit

Implicit

Non-parametric

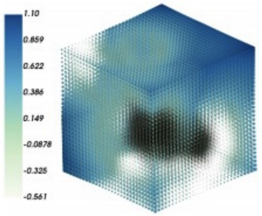


Points



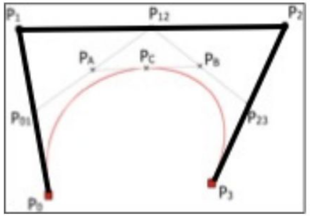
Meshes

Voxels

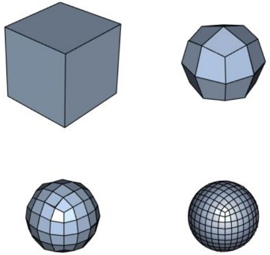


Level Sets

Parametric



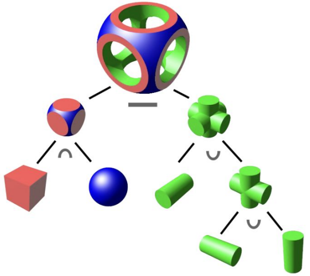
Splines



Subdivision Surfaces

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

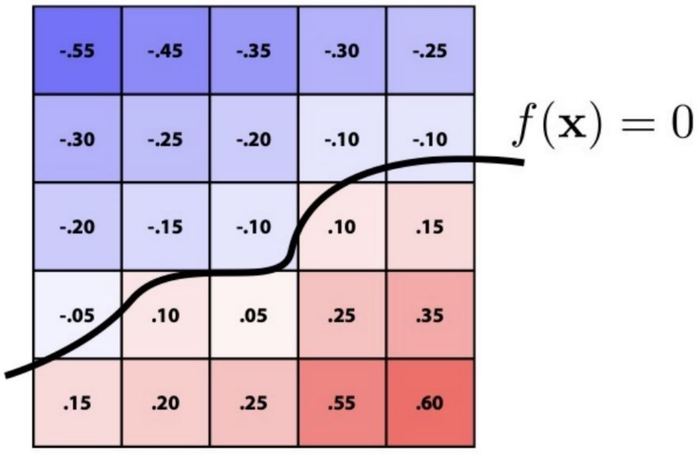
Algebraic Surfaces



Constructive Solid Geometry

3D Representation

Implicit Representation : Level Set Methods



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

But hard to describe complex shapes in closed form.

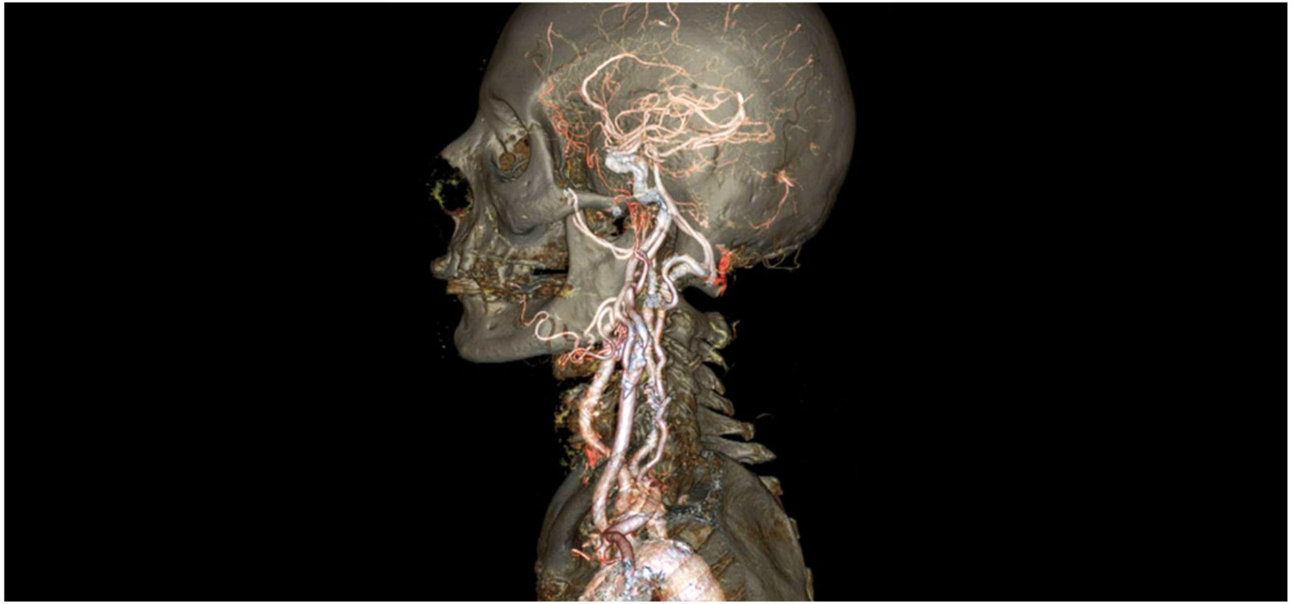
Alternative: store a grid of values approximating function.

Surface is found where interpolated values equal zero. Provides much more explicit control over shape (like a texture).

3D Representation

Implicit Representation : Level Set Methods

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



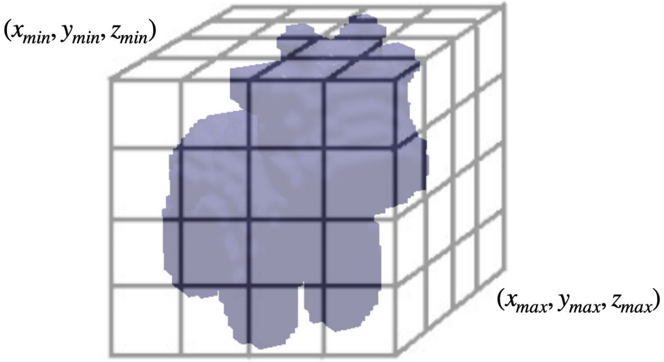
Level sets encode, e.g., constant tissue density

3D Representation

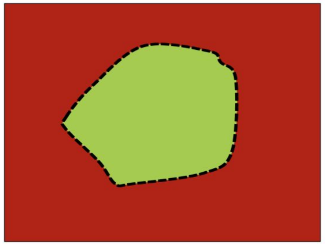
Implicit Representation : Voxels

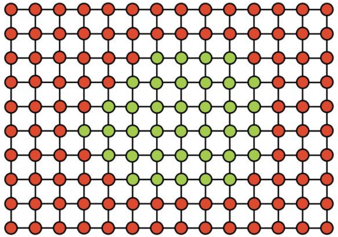
Discretized representation of the space demarcated via cuboid endpoints

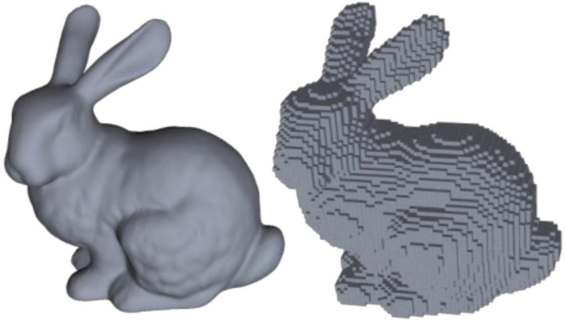
A (W X H X D) grid representing occupancy (or probability) at each cell



$$V[x, y, z] \in [0,1]$$

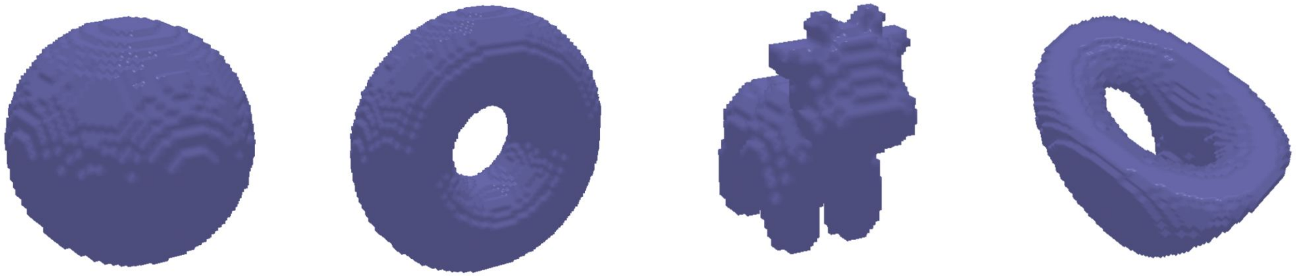






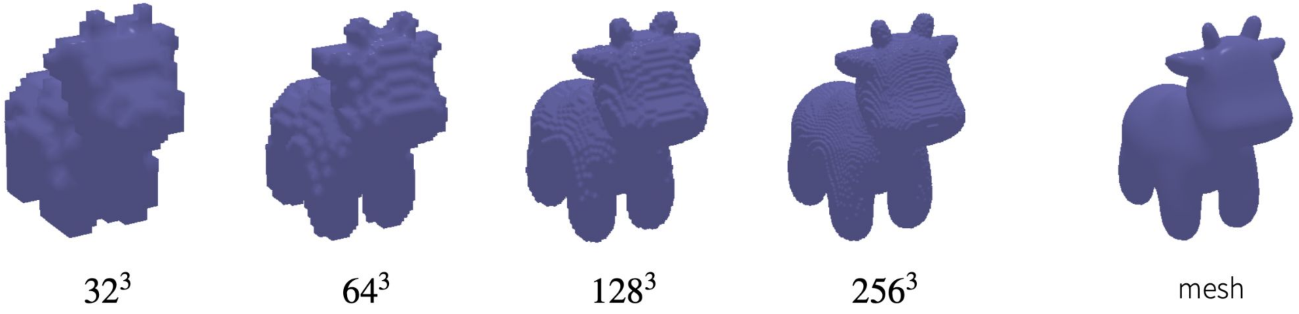
3D Representation

Implicit Representation : Voxels



A generic way for representing shapes across topologies

Computationally challenging to scale resolution



Adopted from: Shubham Tulsiani, Learning for 3D Vision, Fall 2025
Carnegie Mellon University

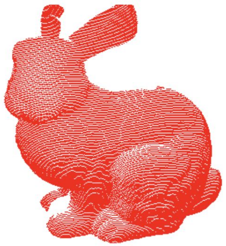
3D Representation

Many ways to represent geometry

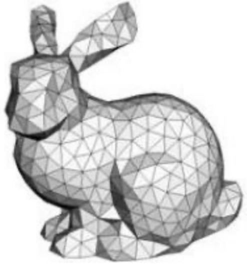
Explicit

Implicit

Non-parametric

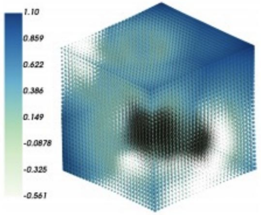


Points



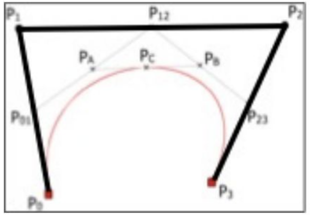
Meshes

Voxels

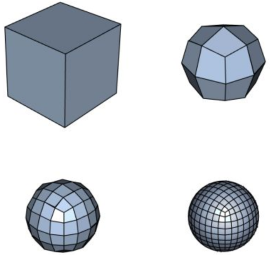


Level Sets

Parametric



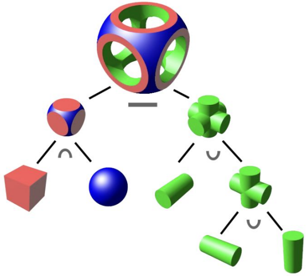
Splines



Subdivision Surfaces

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

Algebraic Surfaces



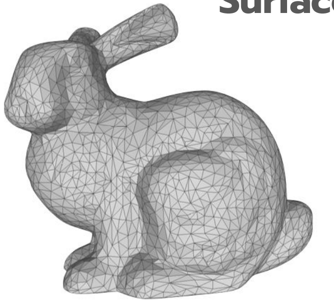
Constructive Solid Geometry

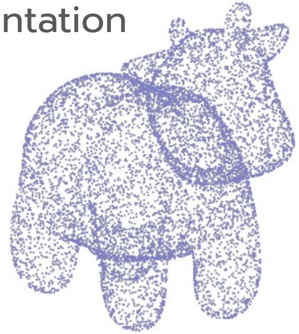
Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

3D Representation

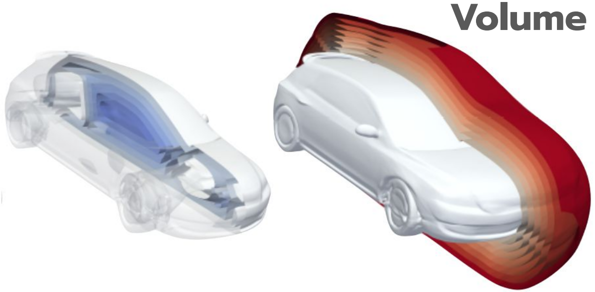
Surface vs. Volume Representations

Surface Representation





Volume Representation



Adopted from: Shubham Tulsiani, Learning for 3D Vision, Fall 2025
Carnegie Mellon University

Point Cloud

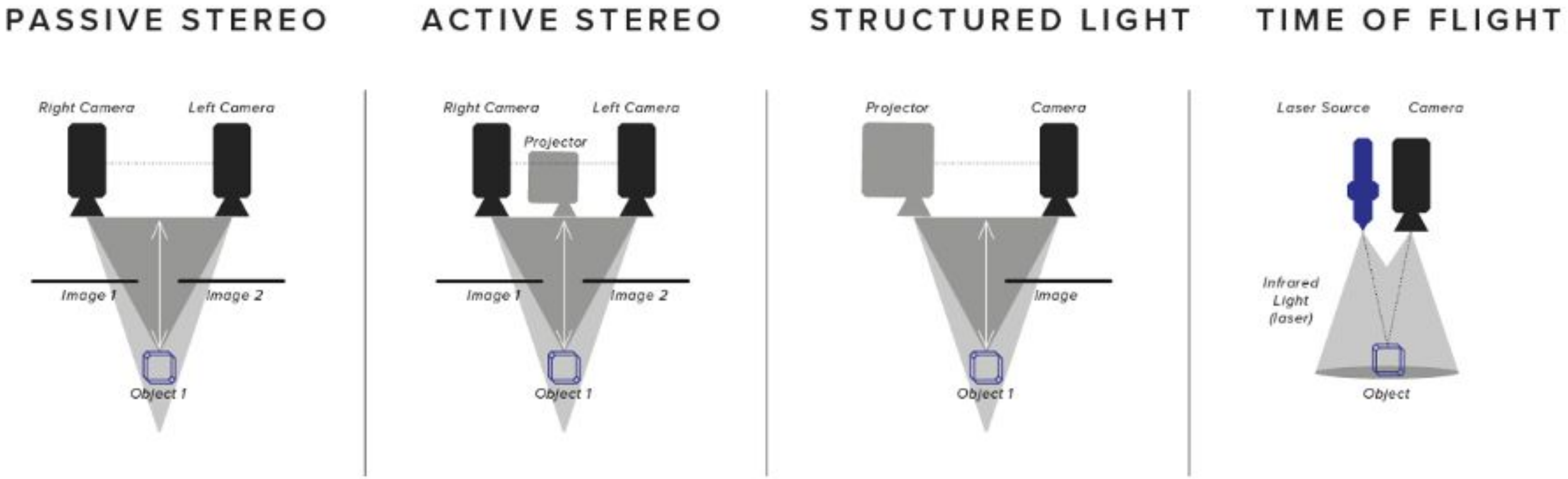
Processing

Point Cloud Processing

How point cloud is generated

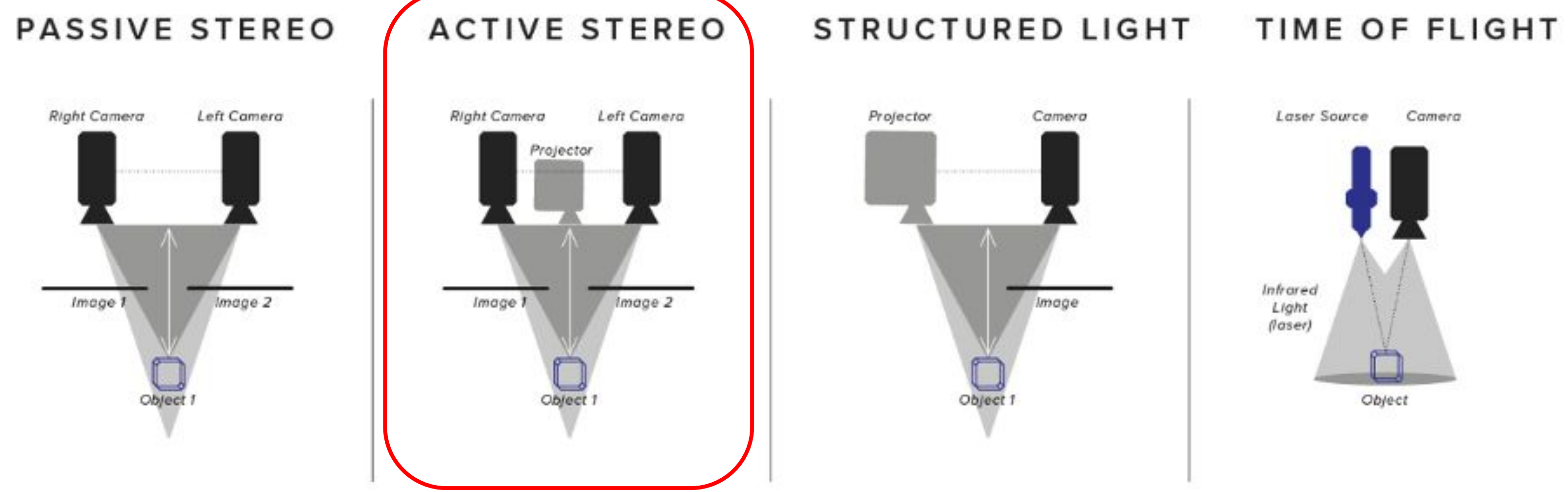
Point Cloud Processing

How point cloud is generated



Point Cloud Processing

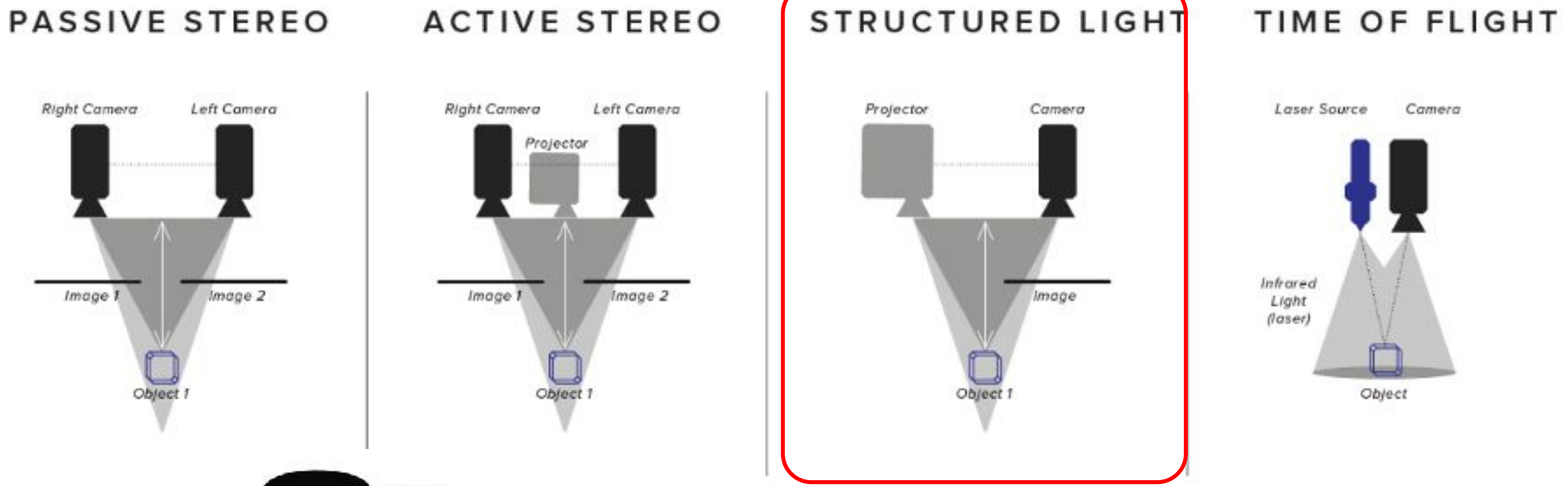
How point cloud is generated

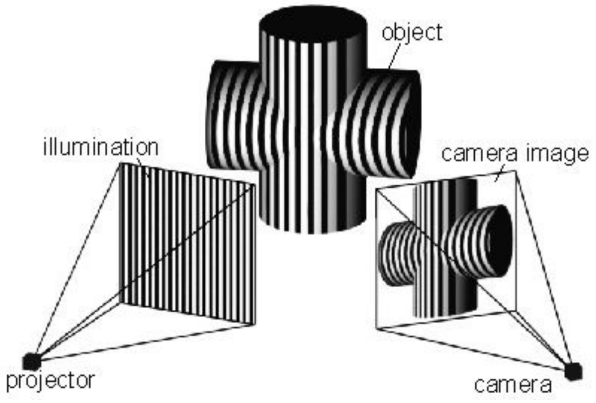


intel.
REALSENSE™

Point Cloud Processing

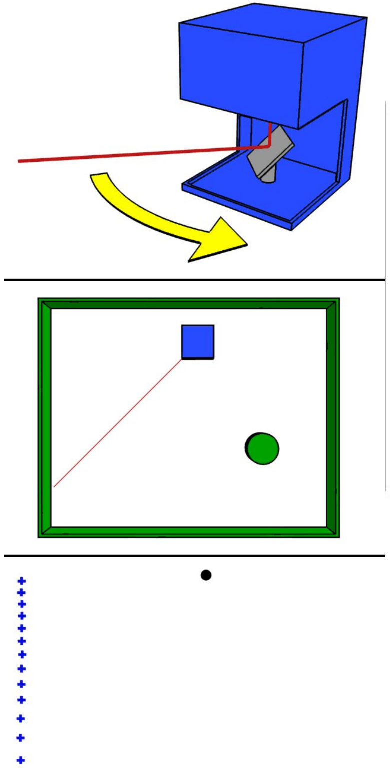
How point cloud is generated

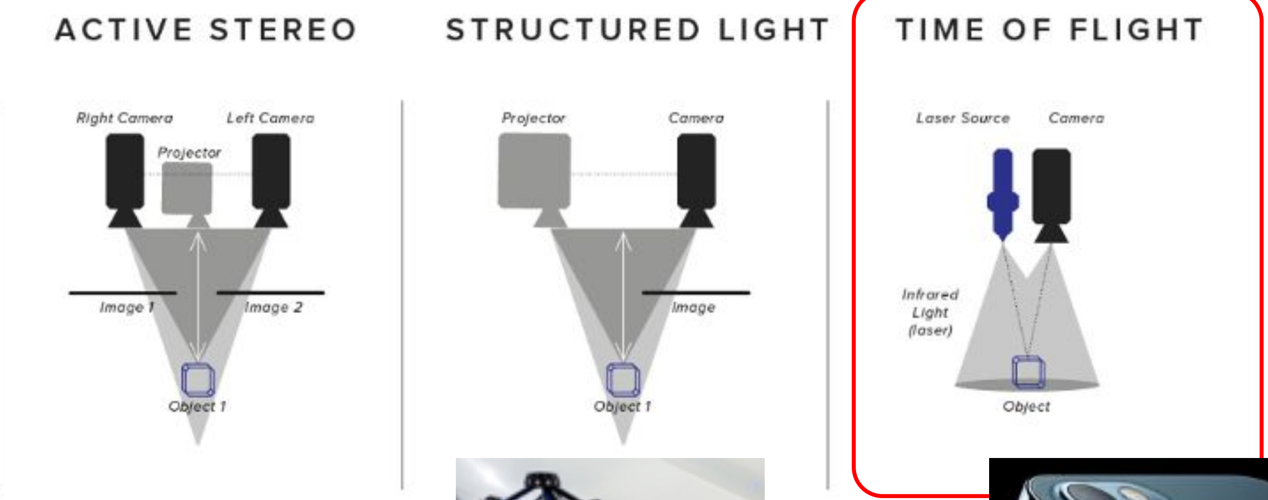


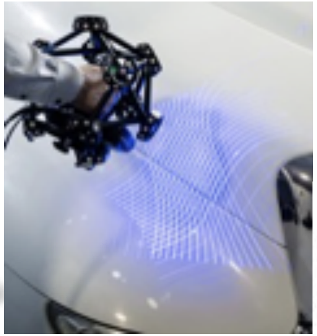


Point Cloud Processing

How point cloud is generated







Point Cloud Processing

What format is point cloud stored?

.PCD

```
# .PCD v.7 - Point Cloud Data file format
VERSION .7
FIELDS x y z rgb
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 213
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 213
DATA ascii
0.93773 0.33763 0 4.2108e+06
0.90805 0.35641 0 4.2108e+06
0.81915 0.32 0 4.2108e+06
0.97192 0.278 0 4.2108e+06
0.944 0.29474 0 4.2108e+06
0.98111 0.24247 0 4.2108e+06
0.93655 0.26143 0 4.2108e+06
0.91631 0.27442 0 4.2108e+06
0.81921 0.29315 0 4.2108e+06
0.90701 0.24109 0 4.2108e+06
0.83239 0.23398 0 4.2108e+06
0.99185 0.2116 0 4.2108e+06
0.89264 0.21174 0 4.2108e+06
0.85082 0.21212 0 4.2108e+06
0.81044 0.32222 0 4.2108e+06
0.74459 0.32192 0 4.2108e+06
0.69927 0.32278 0 4.2108e+06
0.8102 0.29315 0 4.2108e+06
0.75504 0.29765 0 4.2108e+06
0.8102 0.24399 0 4.2108e+06
```

```
ply
format ascii 1.0
comment author: Greg Turk
comment object: another cube
element vertex 8
property float x
property float y
property float z
property uchar red
property uchar green
property uchar blue
element face 7
property list uchar int vertex_inc
element edge 5
property int vertex1
property int vertex2
property uchar red
property uchar green
property uchar blue
end_header
0 0 0 255 0 0
0 0 1 255 0 0
0 1 1 255 0 0
0 1 0 255 0 0
1 0 0 0 0 255
1 0 1 0 0 255
1 1 1 0 0 255
1 1 0 0 0 255
3 0 1 2
3 0 2 3
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
0 1 255 255 255
1 2 255 255 255
2 3 255 255 255
3 0 255 255 255
2 0 0 0 0
```

.PLY

Point Cloud Processing

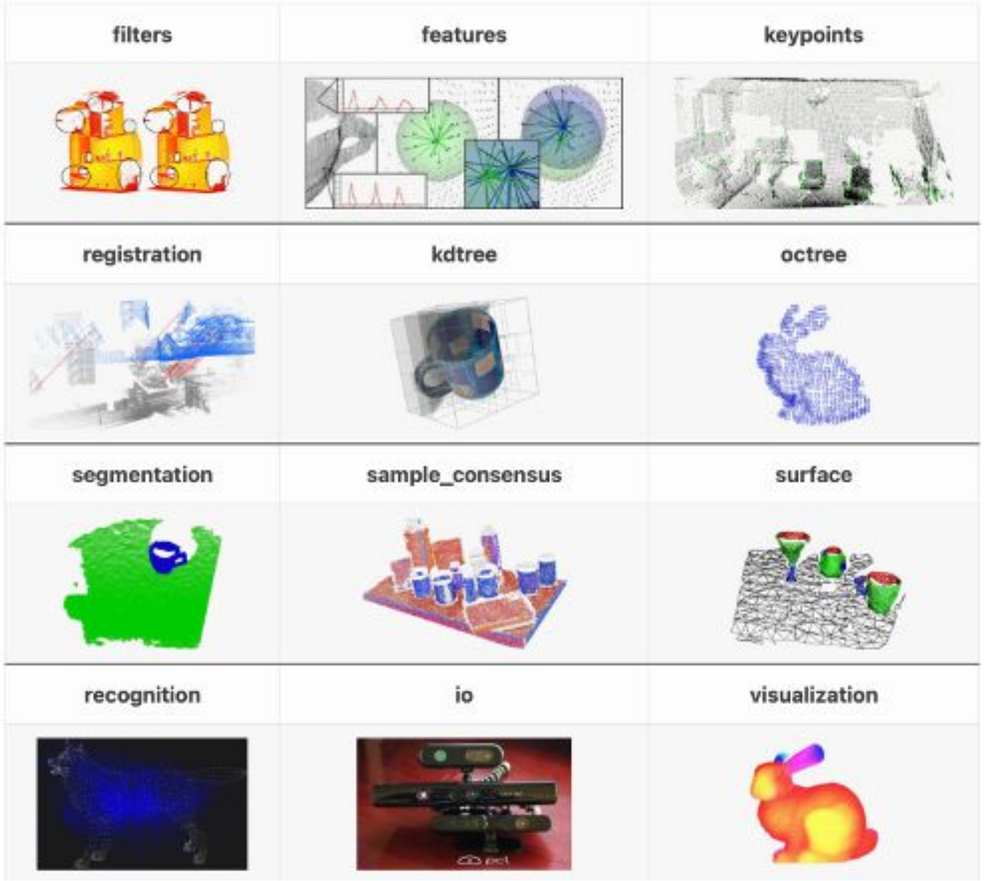


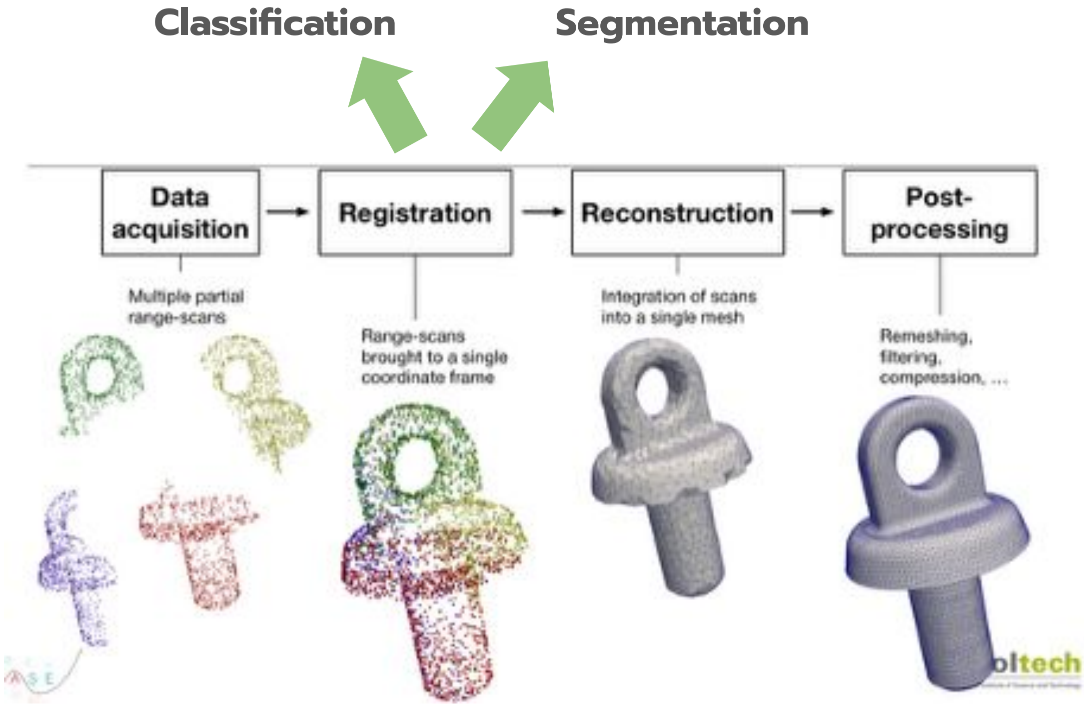
Image from: www.pointcloud.org

Point Cloud Processing

Registration

Point Cloud Processing

Registration



Point Cloud Processing

Registration

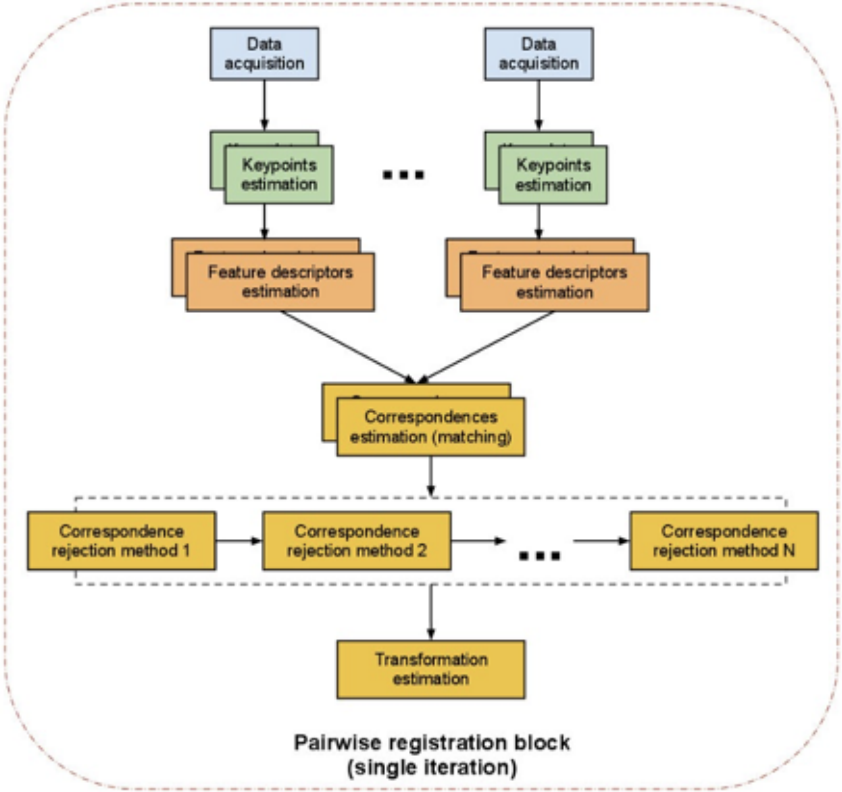
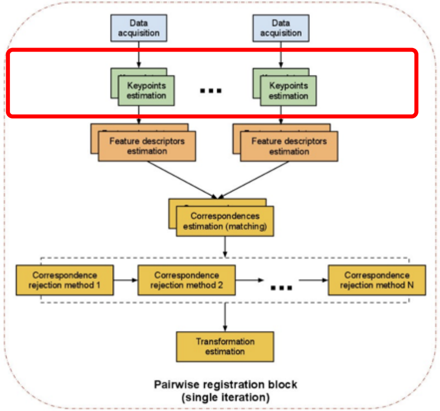


Image from: www.pointcloud.org

Point Cloud Processing

Registration - Keypoints



Good Keypoints

- Repeatability
- Distinctiveness

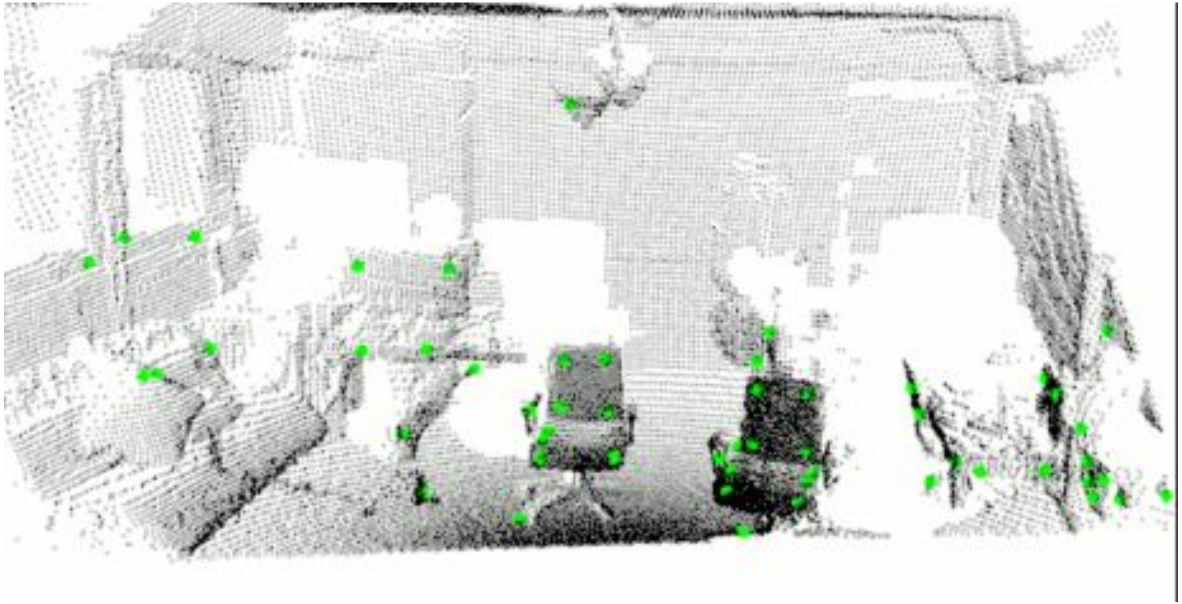
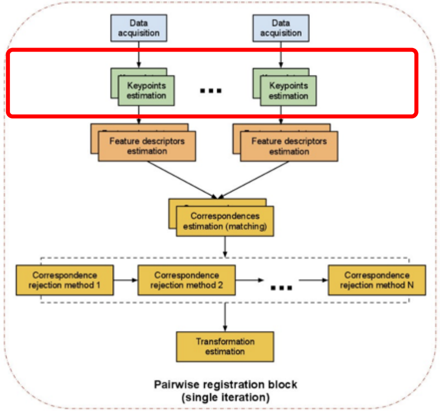


Image from: https://robotica.unileon.es/index.php?title=PCL/OpenNI_tutorial_4:_3D_object_recognition_(descriptors)#NARF

Point Cloud Processing

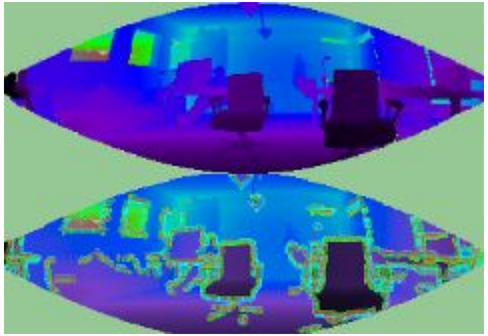
Registration - Keypoints



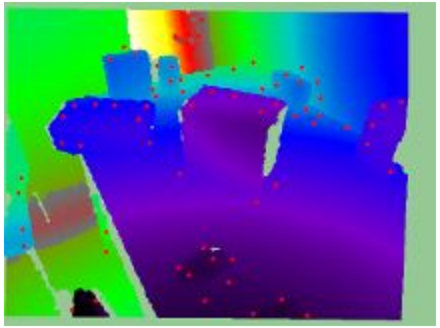
NARF- Normal Aligned Radial Feature

NARF keypoints are located near an object's corners

- Contains information about border and surface structure.
- Be reliably detected regardless to perspective.
- Suitable area for normal estimation



Extracting border

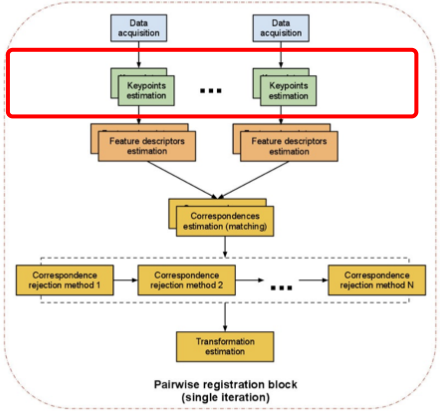


Keypoints

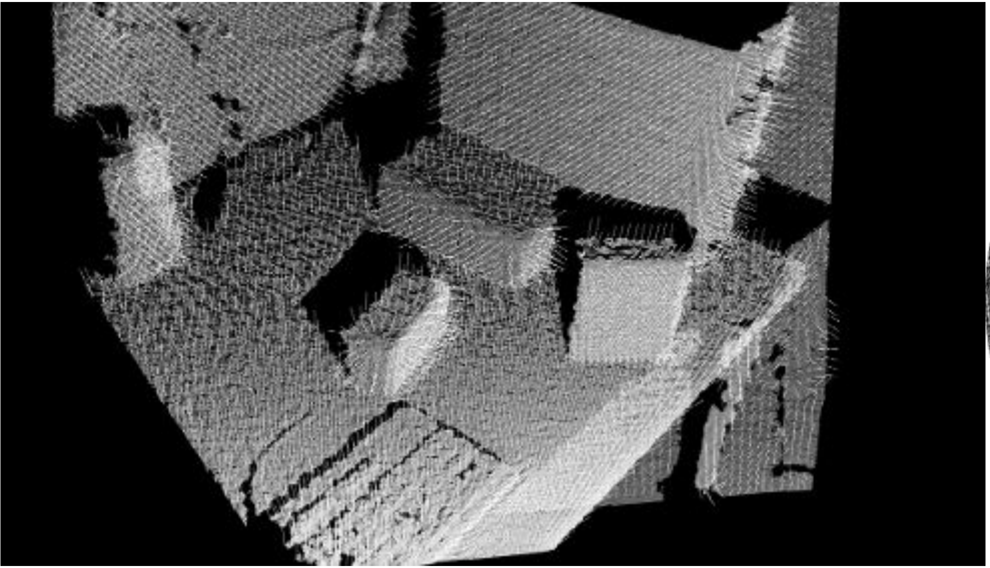
Image from: https://robotica.unileon.es/index.php?title=PCL/OpenNI_tutorial_4:_3D_object_recognition_(descriptors)#NARF

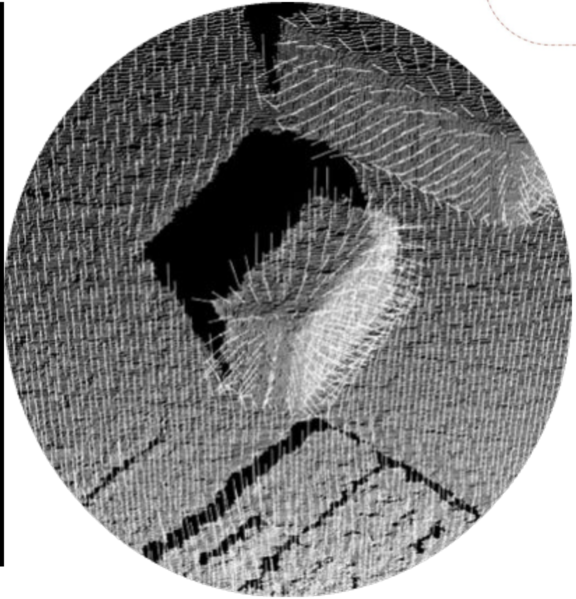
Point Cloud Processing

Registration - Keypoints



Normal Estimation





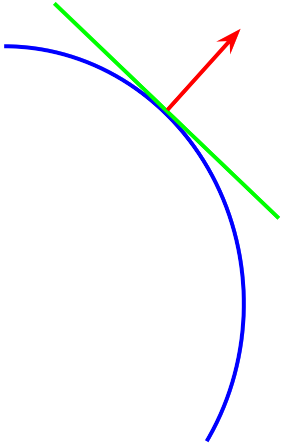


Image from:
https://robotica.unileon.es/index.php?title=PCL/OpenNI_tutorial_4:_3D_object_recognition_(descriptors)#NARF

Point Cloud Processing

Registration - Feature Descriptors

Name	Type
PFH (Point Feature Histogram)	Local
FPFH (Fast Point Feature Histogram)	Local
RSD (Radius-Based Surface Descriptor)	Local
3DSC (3D Shape Context)	Local
USC (Unique Shape Context)	Local
SHOT (Signatures of Histograms of Orientations)	Local
Spin image	Local
RIFT (Rotation-Invariant Feature Transform)	Local
NARF (Normal Aligned Radial Feature)	Local
RoPS (Rotational Projection Statistics)	Local
VFH (Viewpoint Feature Histogram)	Global
CVFH (Clustered Viewpoint Feature Histogram)	Global
OUR-CVFH (Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram)	Global
ESF (Ensemble of Shape Functions)	Global
GFPFH (Global Fast Point Feature Histogram)	Global
GRSD (Global Radius-Based Surface Descriptor)	Global

Feature Descriptors

- Signatures of points
- Encode or describe a lot of information about the surrounding geometry.

Good Descriptors

- Robust to transformation
- Robust to noise
- Resolution invariant

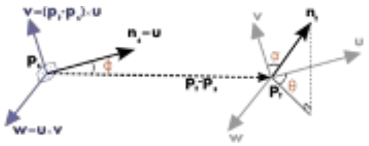
Image from: https://robotica.unileon.es/index.php?title=PCL/OpenNI_tutorial_4:_3D_object_recognition_(descriptors)

Point Cloud Processing

Registration - Feature Descriptors

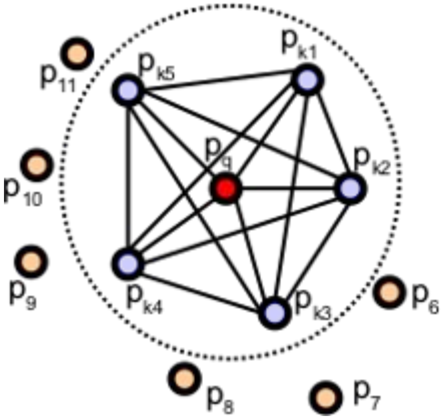
Local - PFH

To capture information of the geometry surrounding the point by analyzing the difference between the directions of the normals in the vicinity



$$u = n_i \qquad v = u \times \frac{(p_j - p_i)}{\|p_j - p_i\|_2} \qquad w = u \times v$$

$$\alpha = v \cdot n_j \qquad \phi = u \cdot \frac{(p_j - p_i)}{d} \qquad \theta = \arctan(w \cdot n_j,\, u \cdot n_j)$$



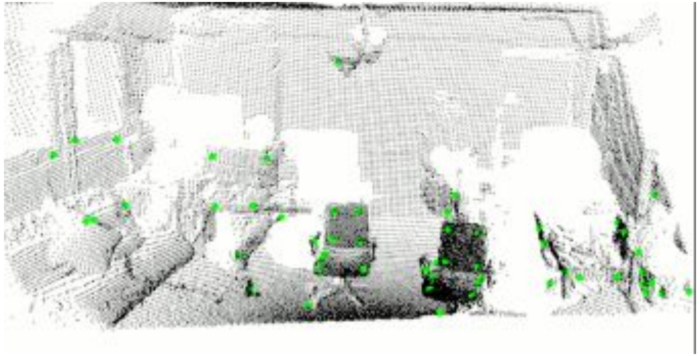
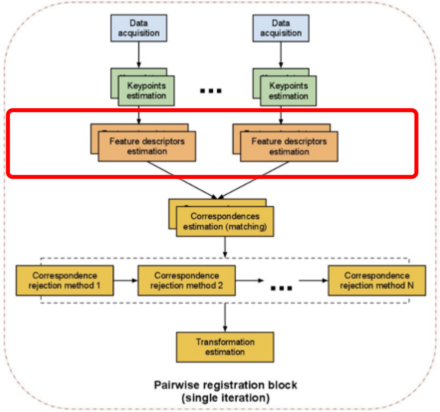


Image from: www.pointcloud.org

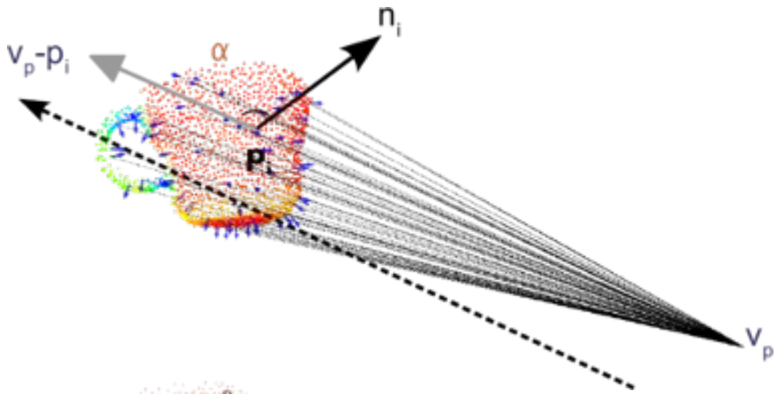
Cherdsak Kingkan

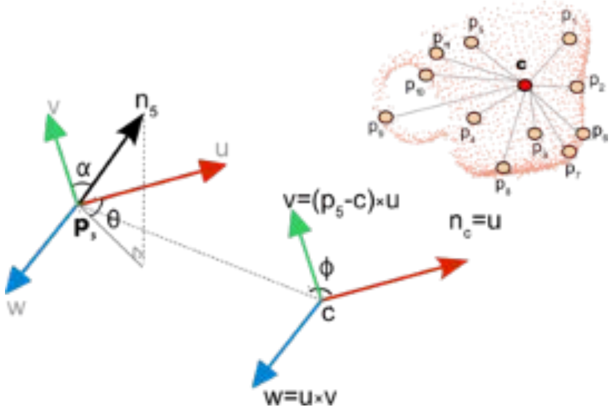
Point Cloud Processing

Registration - Feature Descriptors



Global - VFH





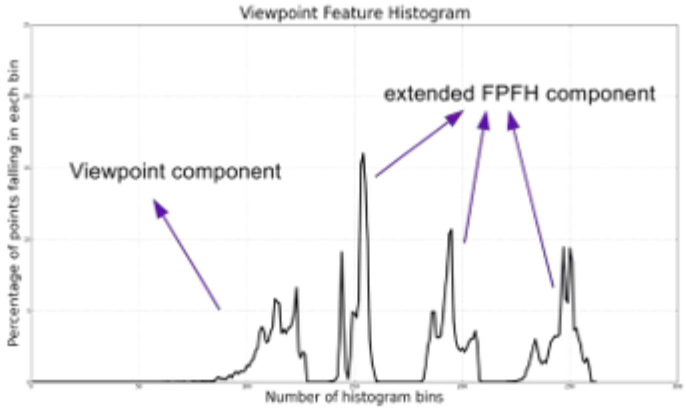
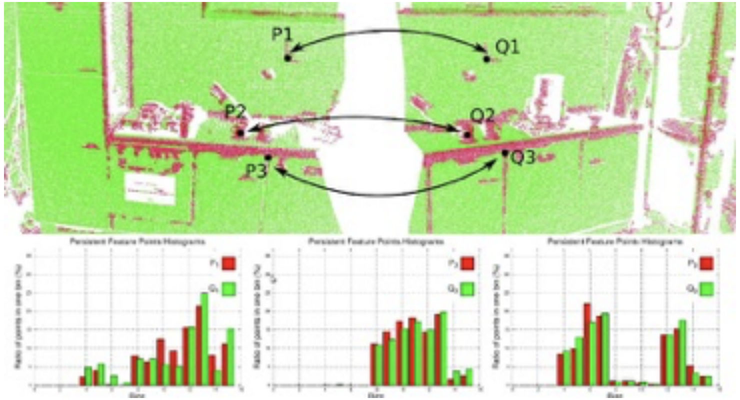


Image from: www.pointcloud.org

Point Cloud Processing

Registration - Correspondences Estimation



Point Matching

Using point coordinates as features

- brute force matching,
- kd-tree nearest neighbor search (FLANN),
- searching in the image space of organized data, and
- searching in the index space of organized data.

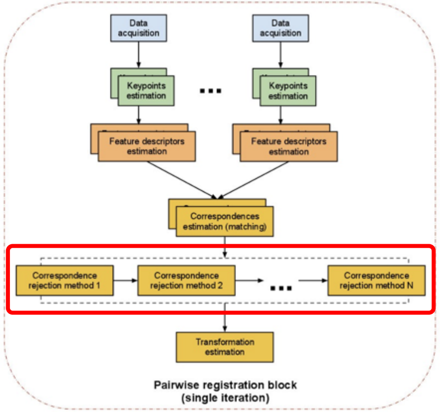
Feature Matching

- brute force matching and
- kd-tree nearest neighbor search (FLANN).

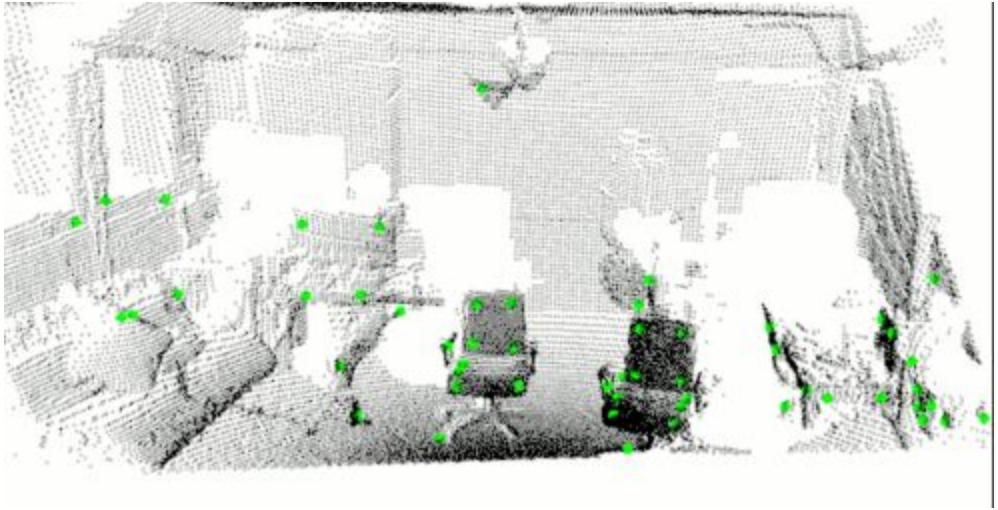
Image from: www.pointcloud.org

Point Cloud Processing

Registration - Correspondences Rejection



RANSAC – Random Sample Consensus



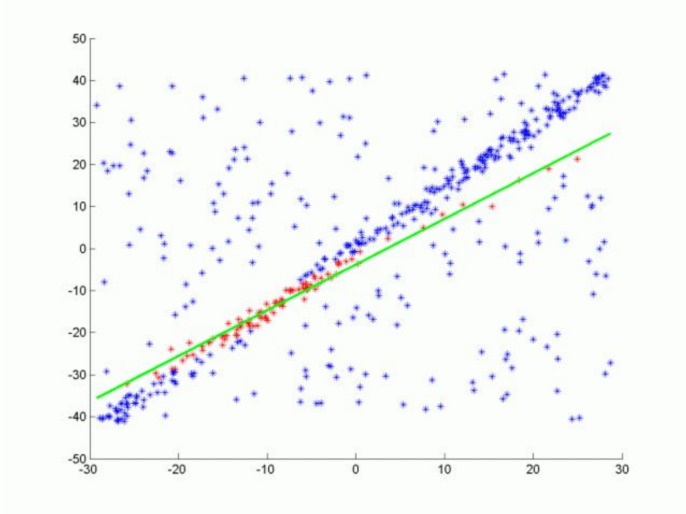
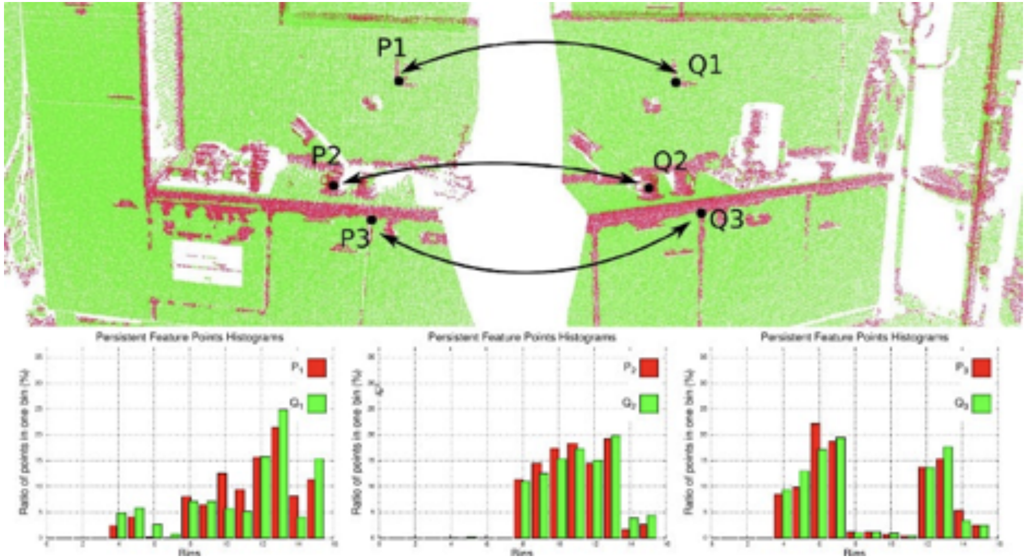


Image from: www.pointcloud.org

Point Cloud Processing

Registration - Transformation Estimation

RANSAC – Random Sample Consensus



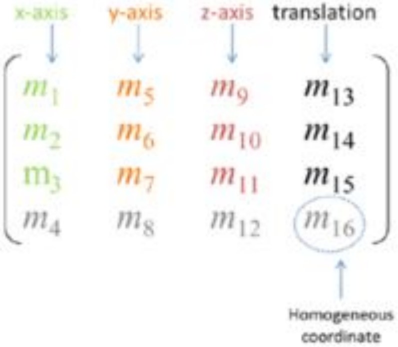
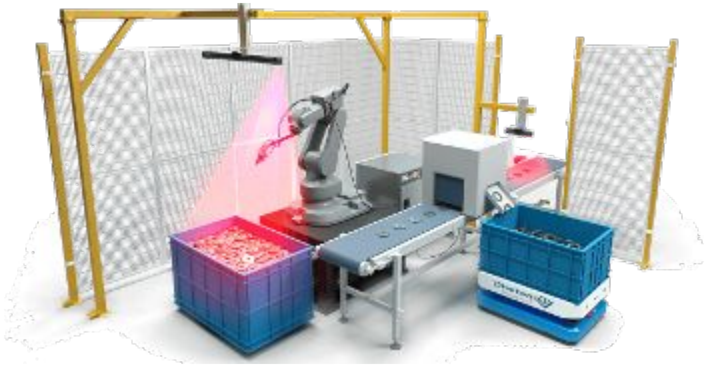
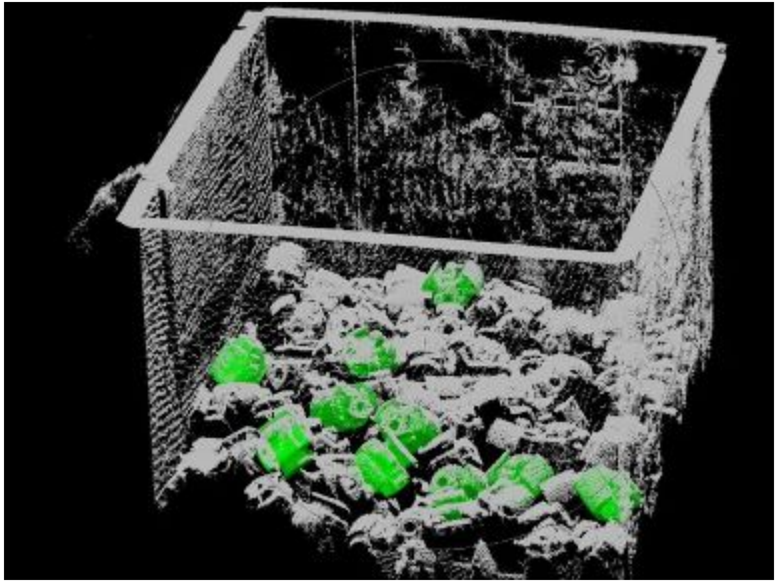


Image from: www.pointcloud.org

Point Cloud Processing

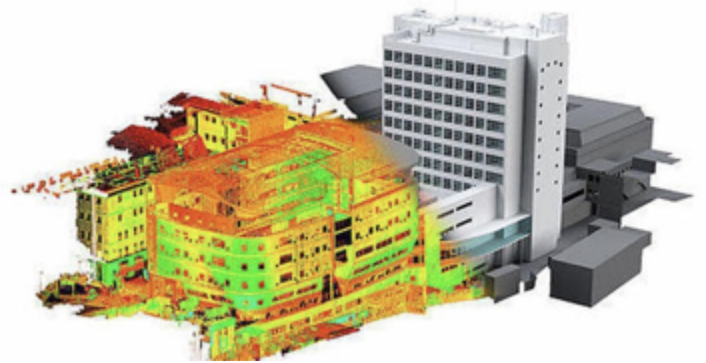
Registration - Applications

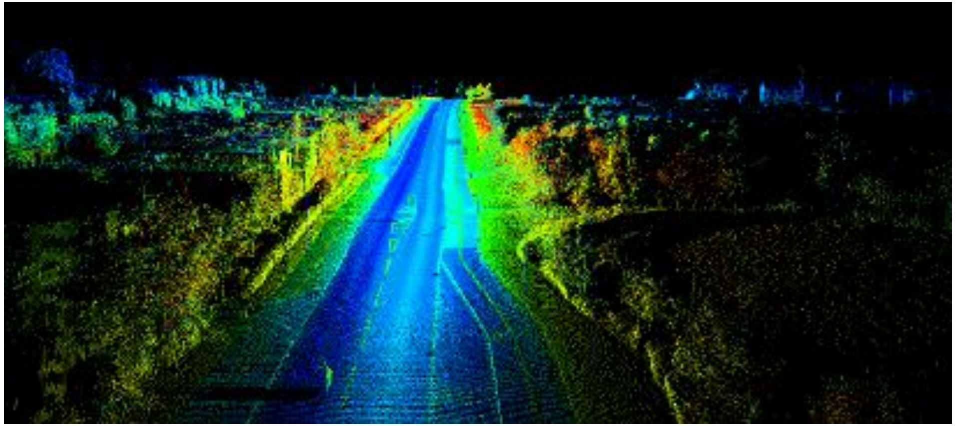




Point Cloud Processing

Applications



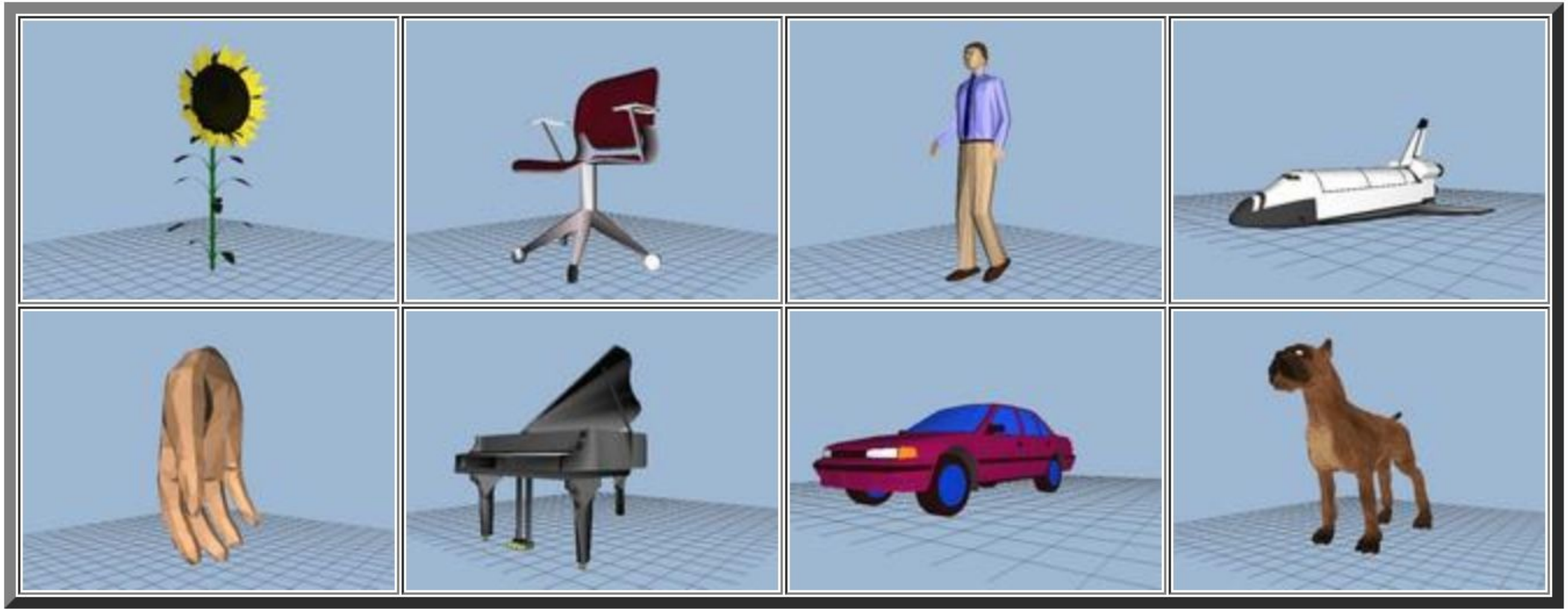


3D Deep Learning

Datasets

Princeton Shape Benchmark (2003)

- 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].

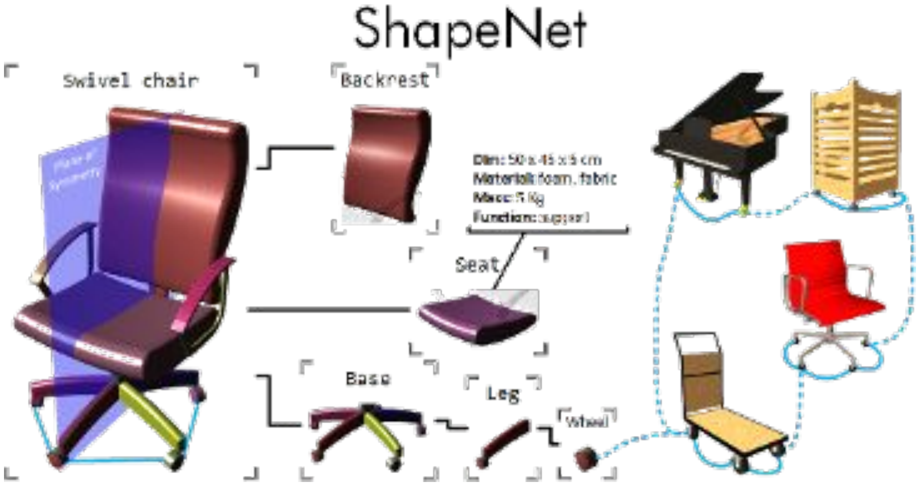
Datasets

Large-scale Synthetic Object Datasets

- 127,915 3D CAD models from 662 categories
- ModelNet10: 4,899 models from 10 categories
- ModelNet40: 1,2311 models from 40 categories, all are uniformly orientated

Datasets

Large-scale Synthetic Object Datasets



- 3 Million+ models and 4K+ categories.
- A dataset that is large in scale, well organized and richly annotated
- ShapeNetCore: 51,300 models for 55 categories.

Datasets

Large-scale Synthetic Object Datasets

ABC Dataset: A Big CAD Model Dataset For Geometric Deep Learning

- a collection of **one million** Computer-Aided Design (CAD) models for research of geometric deep learning methods and applications.
- Each model is a collection of explicitly parametrized curves and surfaces, providing ground truth for
    - differential quantities,
    - patch segmentation,
    - geometric feature detection,
    - shape reconstruction.

Datasets

Objverse (800K) and Objverse-XL (10M)

- Matt Deitke et al., Objaverse-XL: A Universe of 10M+ 3D Objects, 2023
- Matt Deitke et al., Objaverse: A Universe of Annotated 3D Objects, 2022

Datasets

ScanNN & ScanNet for indoor scenes

SceneNN (2016)

- 100+ indoor scene meshes with per-vertex and per-pixel annotation.

ScanNet (2017)

- An RGB-D video dataset containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations.

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

Datasets

A Large Dataset of Object Scans (2016)

- 10K scans in RGBD + reconstructed 3D models in .PLY format.

Datasets

CO3D: Common Objects in 3D: Large-Scale Learning and Evaluation of Real-life 3D Category Reconstruction

- Videos: 18,619
- Categories: 50 MS-COCO
- Camera-annotated frames: 1.5 million
- Point-cloud-annotated videos: 5,625

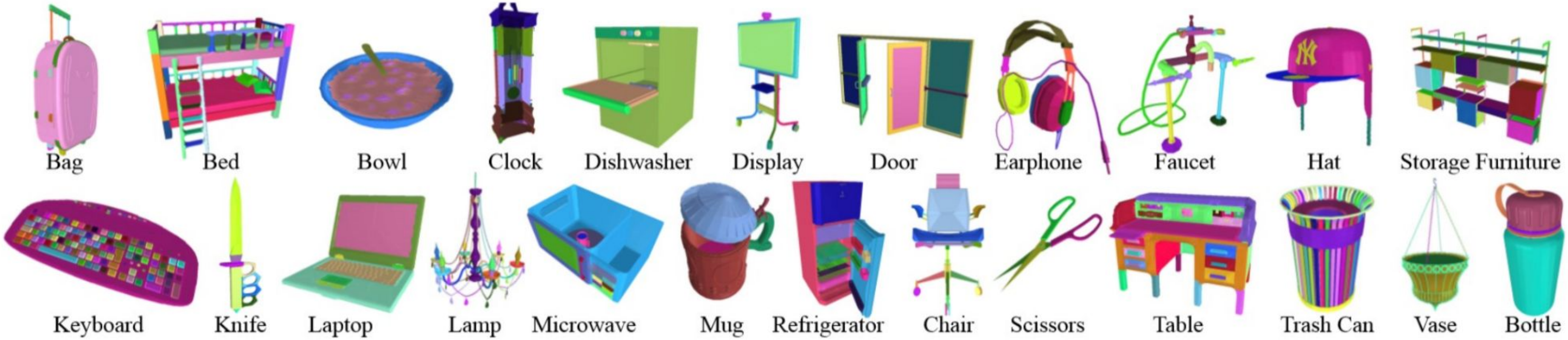
Source from: https://ai.meta.com/datasets/co3d-dataset/, https://github.com/facebookresearch/co3d/tree/main

Datasets

3D Object Parts

Fine-grained Parts: PartNet

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

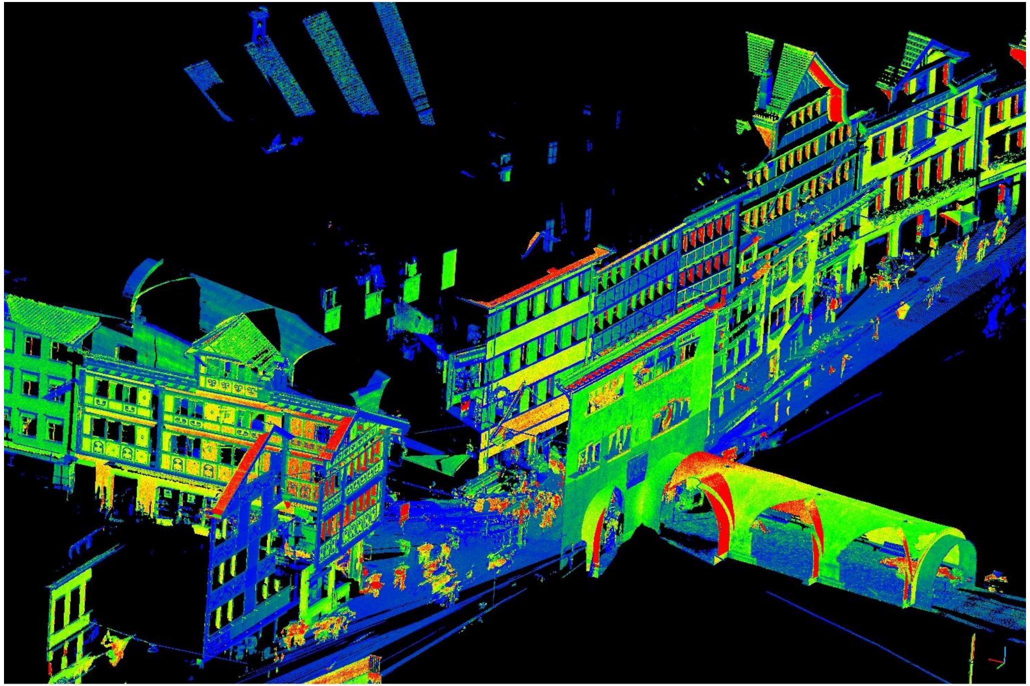


Datasets

Point Cloud Datasets

Semantic3D

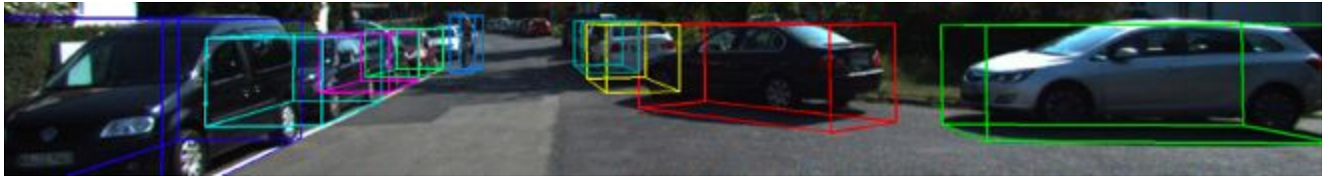
Large-Scale Point Cloud Classification Benchmark, which provides a large labelled 3D point cloud data set of natural scenes with over **4 billion points** in total, and also covers a range of diverse urban scenes.



Datasets

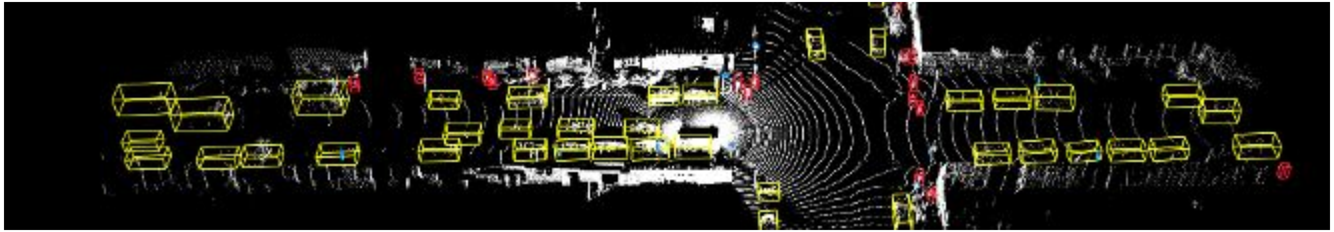
Point Cloud Datasets

KITTI: LiDar data, labeled by 3D bounding box



Semantic KITTI: LiDar data, labeled per point

Waymo Open Dataset: LiDar data, labeled by 3D bounding box



Tools

TensorFlow Graphics

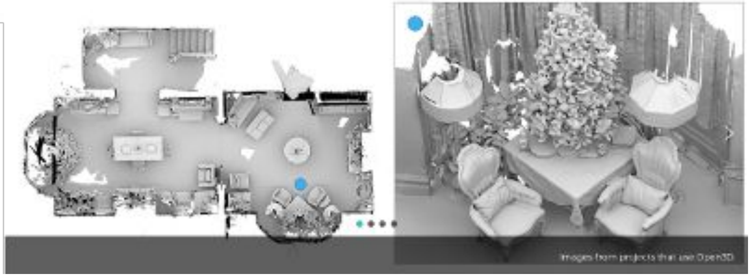
Jraph - library for graph neural networks in jax

PyTorch3D

A LIBRARY FOR DEEP LEARNING WITH 3D DATA

Open3D

A Modern Library for 3D Data Processing



PyTorch geometric

trimesh

DGL DEEP GRAPH LIBRARY

AI + Geometry

Tasks

***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

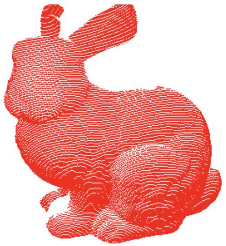
- Large-scale 2D datasets & very good pretrained models
- Differentiable projection/back-projection & differentiable/neural rendering

Joint modeling of multi-modal data beyond visual (e.g., text)

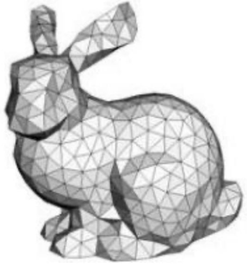
AI + Geometry

Which representation ?

	Explicit		Implicit	
Non-parametric	Points	Meshes	Voxels	Level Sets
Parametric	Splines	Subdivision Surfaces	Algebraic Surfaces	Constructive Solid Geometry

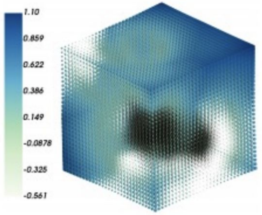


Points

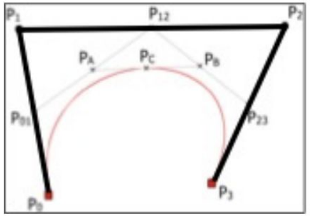


Meshes

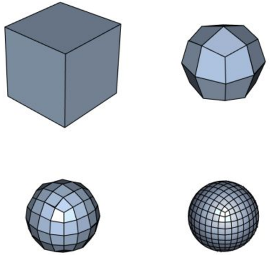
Voxels



Level Sets



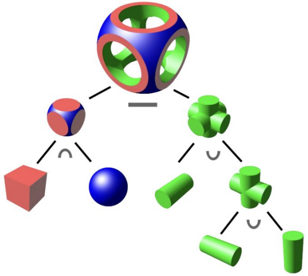
Splines



Subdivision Surfaces

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

Algebraic Surfaces



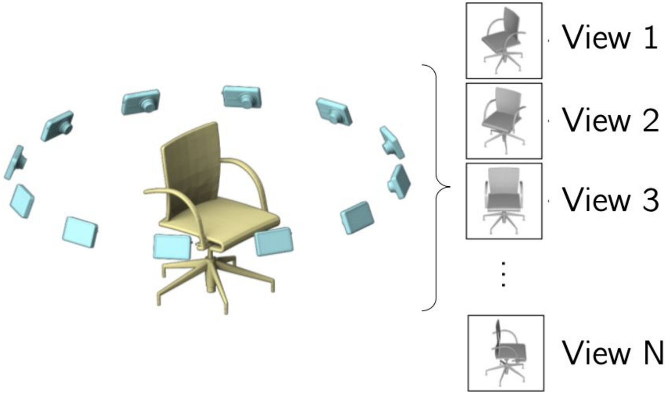
Constructive Solid Geometry

3D Deep Learning

Image-based: Multi-view CNN

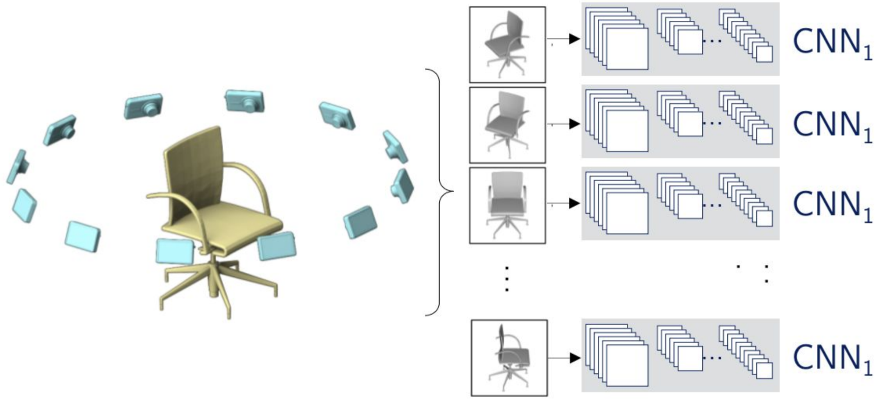
3D Deep Learning

Image-based: Multi-view CNN



3D Deep Learning

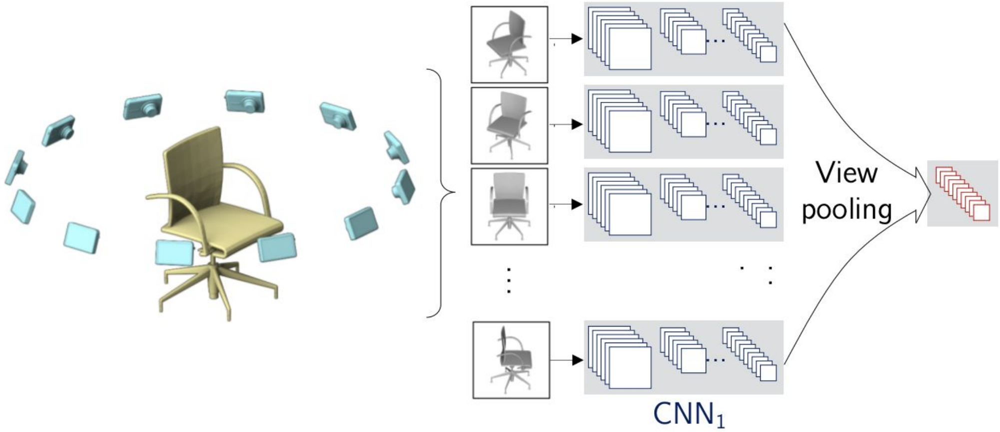
Image-based: Multi-view CNN



CNN_1: a ConvNet extracting image features

3D Deep Learning

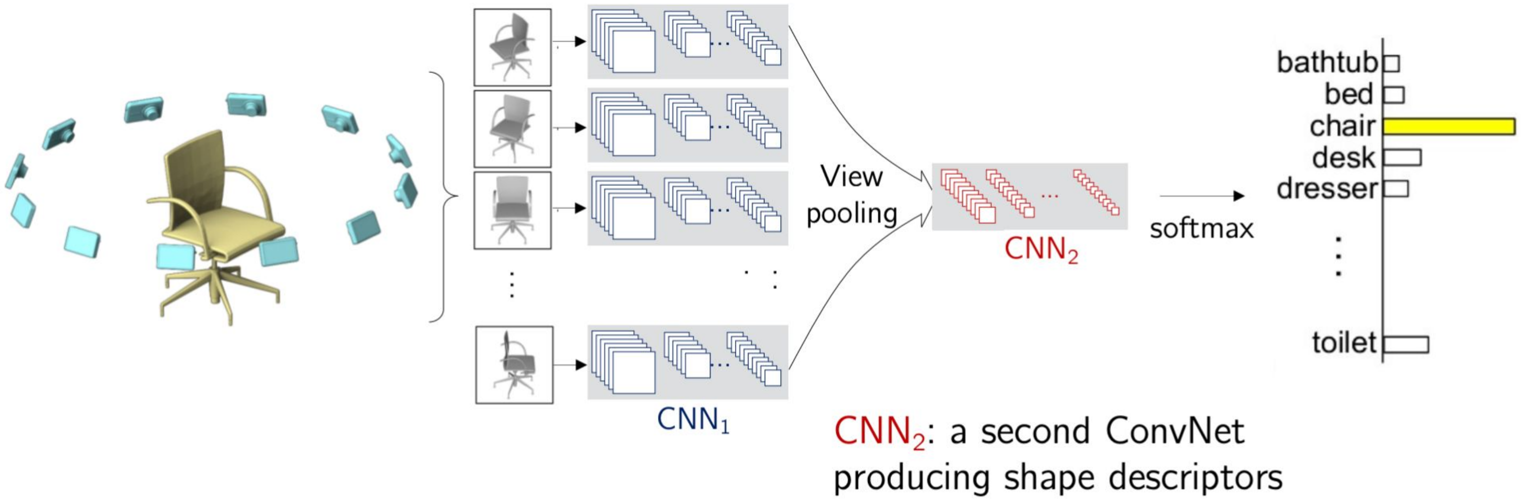
Image-based: Multi-view CNN



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

3D Deep Learning

Image-based: Multi-view CNN



3D Deep Learning

Image-based: Multi-view CNN

Classification & Retrieval

	Method	Classification (Accuracy)	Retrieval (mAP)
Non-deep	SPH	68.2%	33.3%
Non-deep	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 40

Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025
Stanford University

3D Deep Learning

Image-based

Advantages

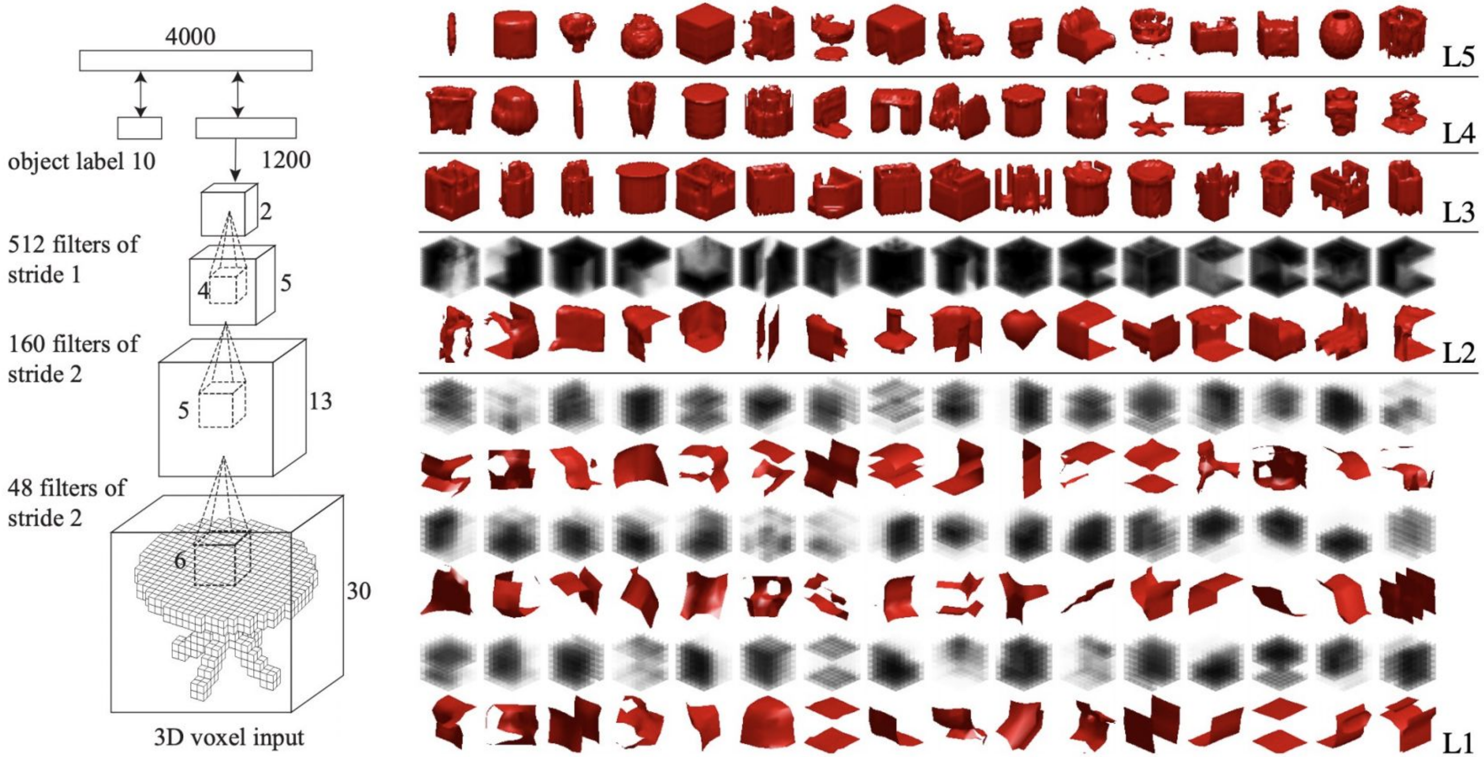
- Gives good performance
- Simple, re-use standard components of CNNs
- Can use pretrained features

Disadvantages

- Memory
- Not geometric
- No invariance
- Need projection
- What if the input is noisy and/or incomplete? e.g., point cloud

3D Deep Learning

Pixels → Voxels: 3D Conv Deep Belief Networks (CDBN)

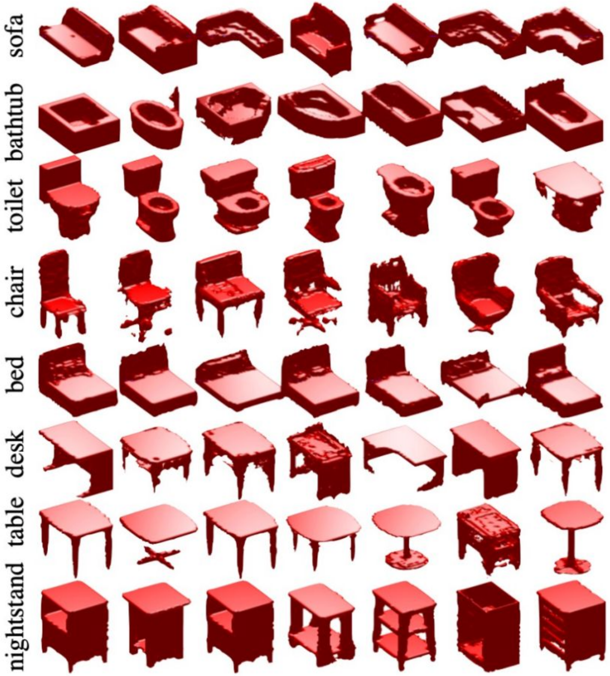


Adopted from: Jianjun Wu, CS231n: Deep Learning for Computer Vision, Spring 2025 Stanford University

3D Deep Learning

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.**

3D Deep Learning

Pixels → Voxels: 3D GANs

Generative Modeling

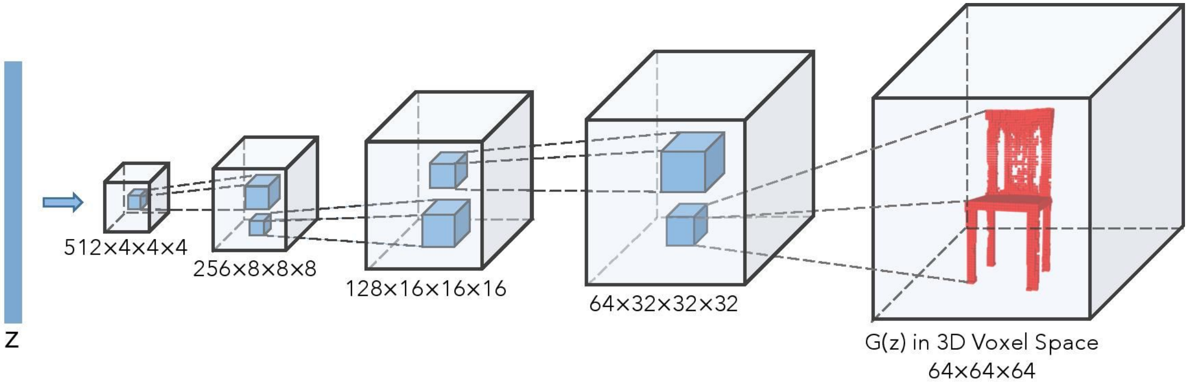


Image from: Wu et al., Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling, 2016

3D Deep Learning

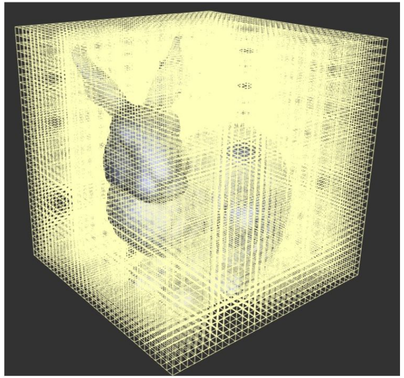
Pixels → Voxels:

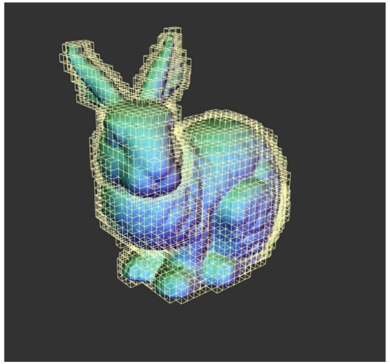
- Represent a shape with a V x V x V grid of occupancies
- (+) Conceptually simple: just a 3D grid!
- (-) Need high spatial resolution to capture fine structures
- (-) Scaling to high resolutions is nontrivial!

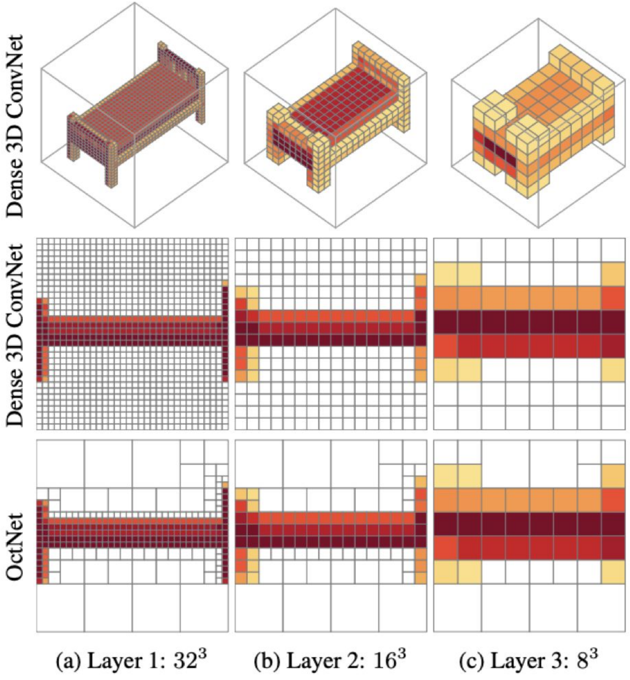
Image from: Wu et al., Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling, 2016

3D Deep Learning

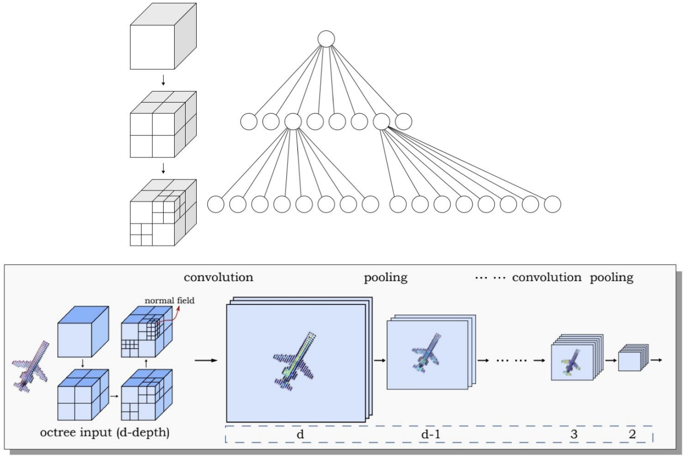
Pixels → Voxels → Octree: Classification







Riegler et al. OctNet. CVPR 2017

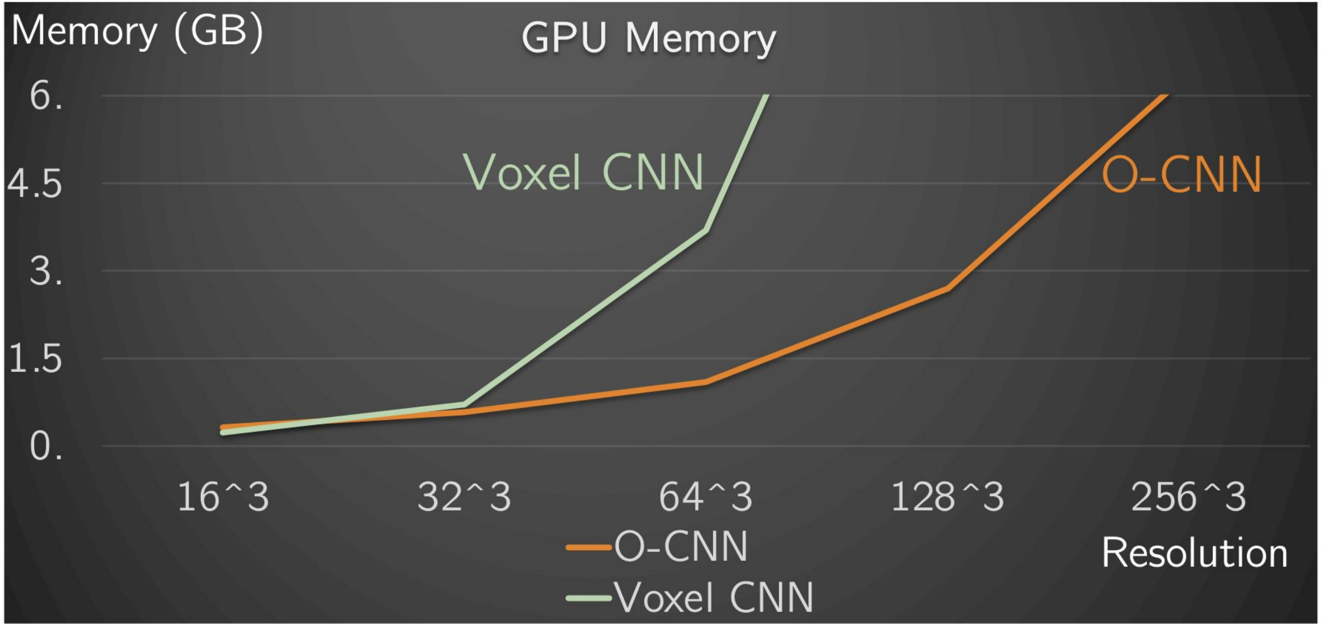


Wang et al. O-CNN. SIGGRAPH 2017

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

3D Deep Learning

Pixels → Voxels → Octree:

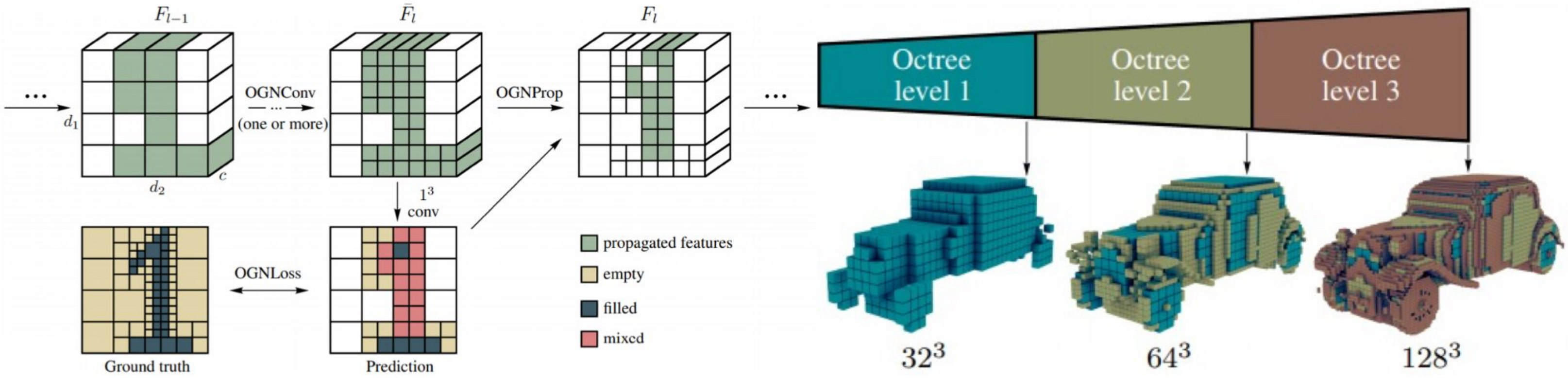


3D Deep Learning

Pixels → Voxels → Octree: Generative Network

Avoid $O(n^3)$ reconstruction

- Octree representation of shapes
- Generate the octree layer by layer



3D Deep Learning

Point Cloud: PointNet

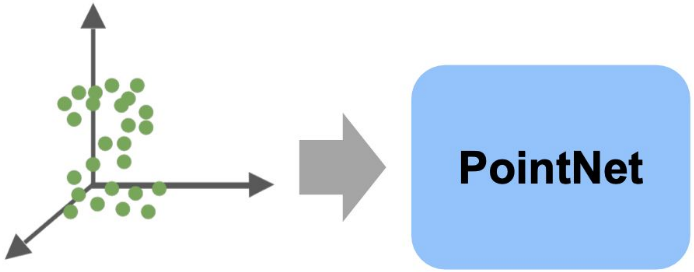
Research Question:

Can we achieve effective feature learning directly on point clouds?

3D Deep Learning

Point Cloud: PointNet

End-to-end learning for **scattered, unordered** point data



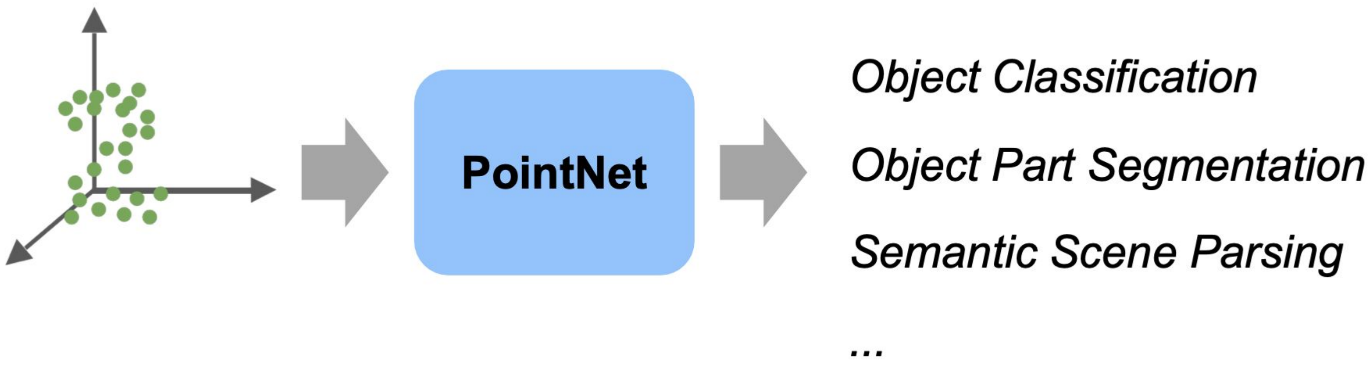
Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

End-to-end learning for **scattered, unordered** point data

Unified framework for various tasks



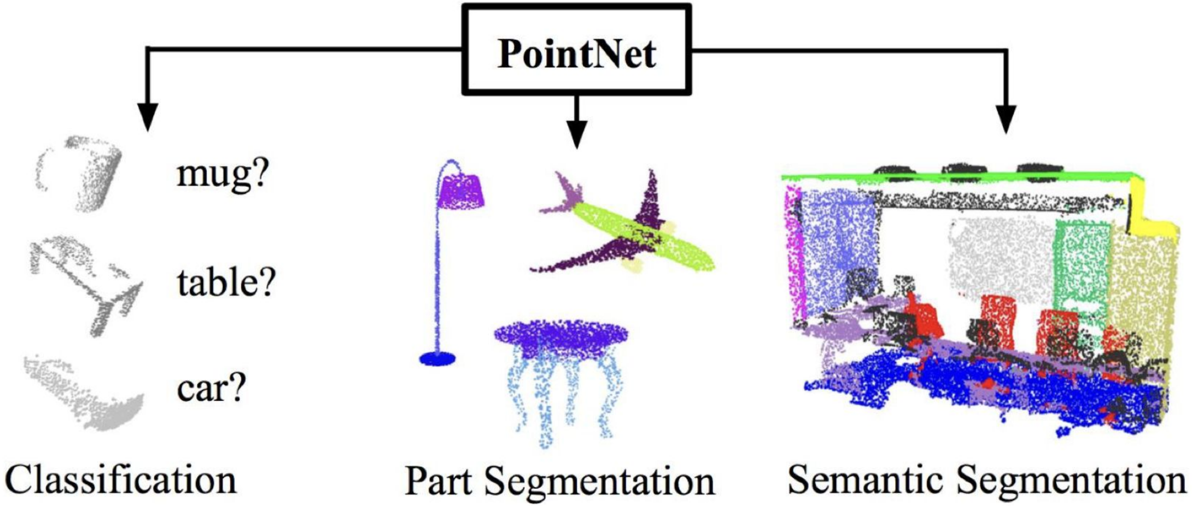
Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

End-to-end learning for **scattered, unordered** point data

Unified framework for various tasks



Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Challenges

- **Unordered point set as input**
  Model needs to be invariant to N! Permutations.

- **Invariance under geometric transformations**
  Point cloud rotations should not alter classification results.

translation / rotation

different size / different orientation

3D Deep Learning

Point Cloud: PointNet

Challenges

- **Unordered point set as input**
  Model needs to be invariant to N! Permutations.

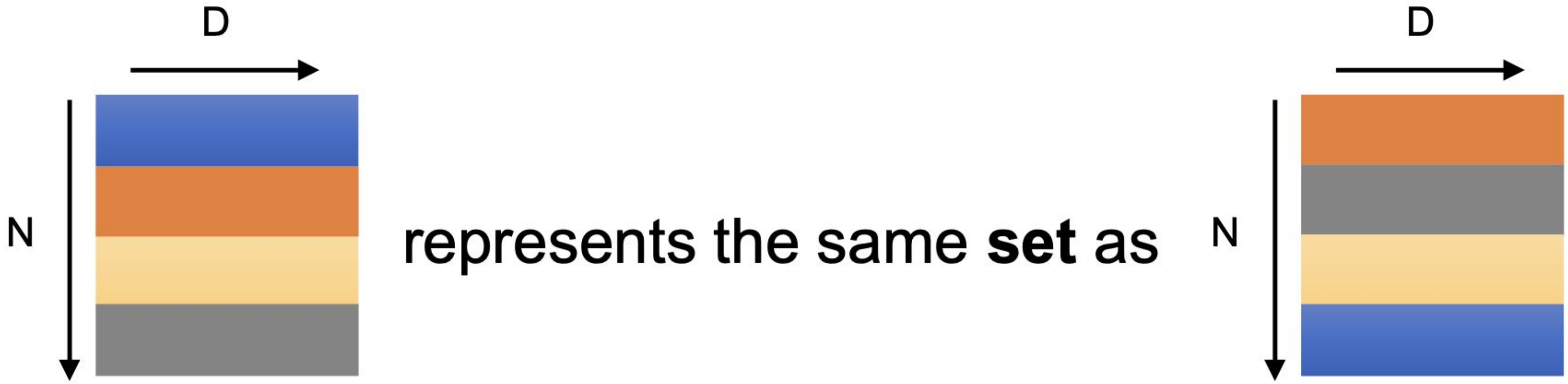
- **Invariance under geometric transformations**
  Point cloud rotations should not alter classification results.

3D Deep Learning

Point Cloud: PointNet

<u>Unordered</u> point set as input

Point cloud: N **orderless** points, each represented by a D dim vector.



Model needs to be invariant to N! Permutations.

Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

Permutation Invariance: Symmetric Function

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

Examples:

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

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

...

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

Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

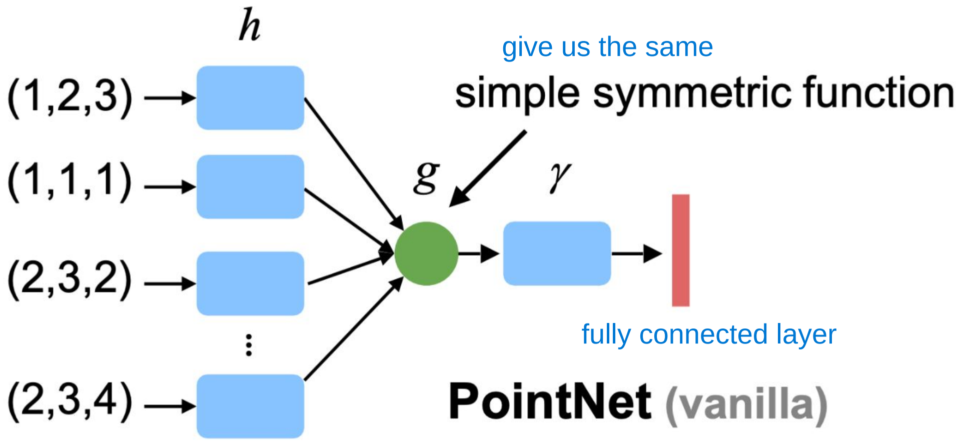
3D Deep Learning

Point Cloud: PointNet

Permutation Invariance: Symmetric Function

Observe:

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

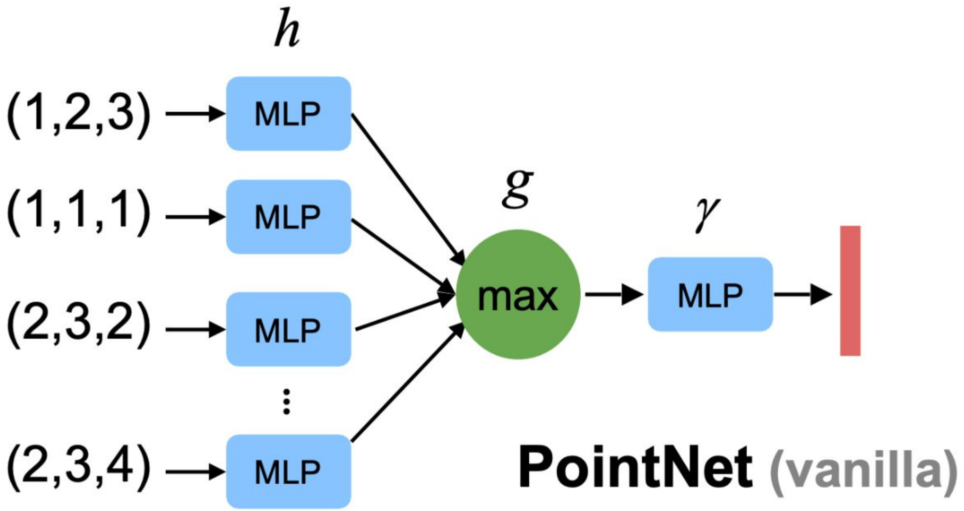


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

Point Cloud: PointNet

Permutation Invariance: Symmetric Function

Empirically, we use **multi-layer perceptron (MLP)** and **max pooling**:



Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Challenges

- **Unordered point set as input**
  Model needs to be invariant to N! Permutations.

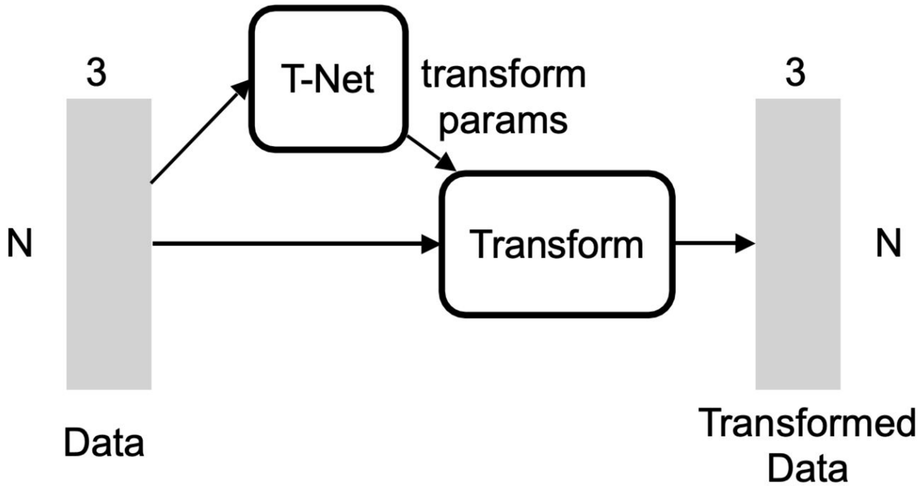
- **Invariance under geometric transformations**
  Point cloud rotations should not alter classification results.

3D Deep Learning

Point Cloud: PointNet

Input Alignment by Transformer Network

Idea: Data dependent transformation for automatic alignment

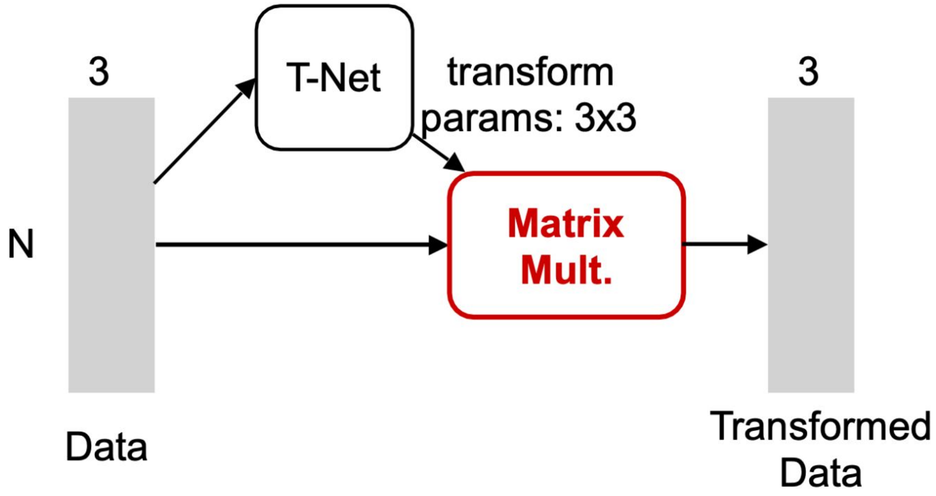


3D Deep Learning

Point Cloud: PointNet

Input Alignment by Transformer Network

The transformation is just matrix multiplication!



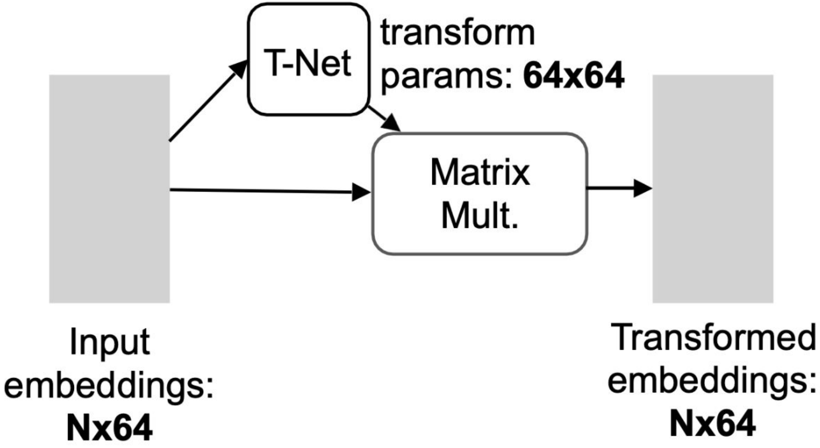
Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Input Alignment by Transformer Network

Embedding Space Alignment



Regularization:

Transform matrix A 64x64 close to orthogonal:

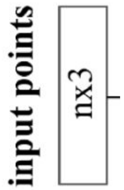
$$L_{reg} = \|I - AA^T\|_F^2$$

Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification

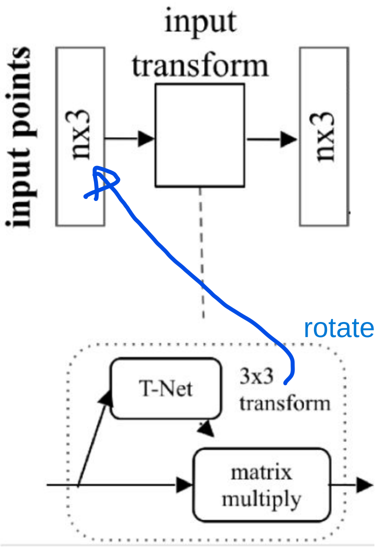


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification



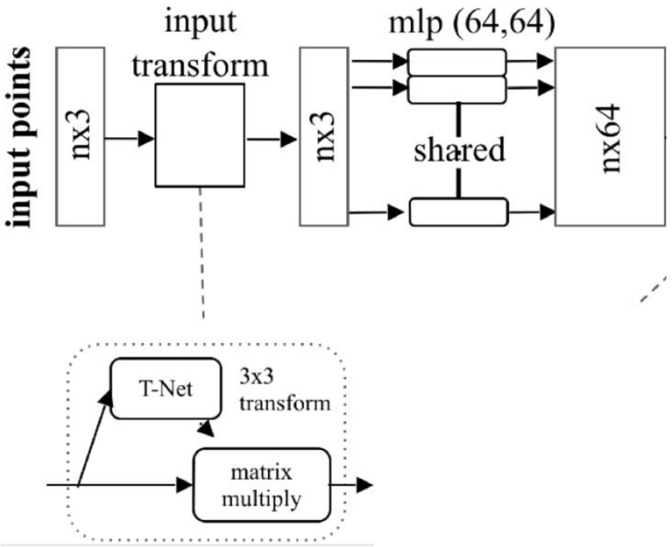
rotate the object with the thing of what we are going to learn

Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification

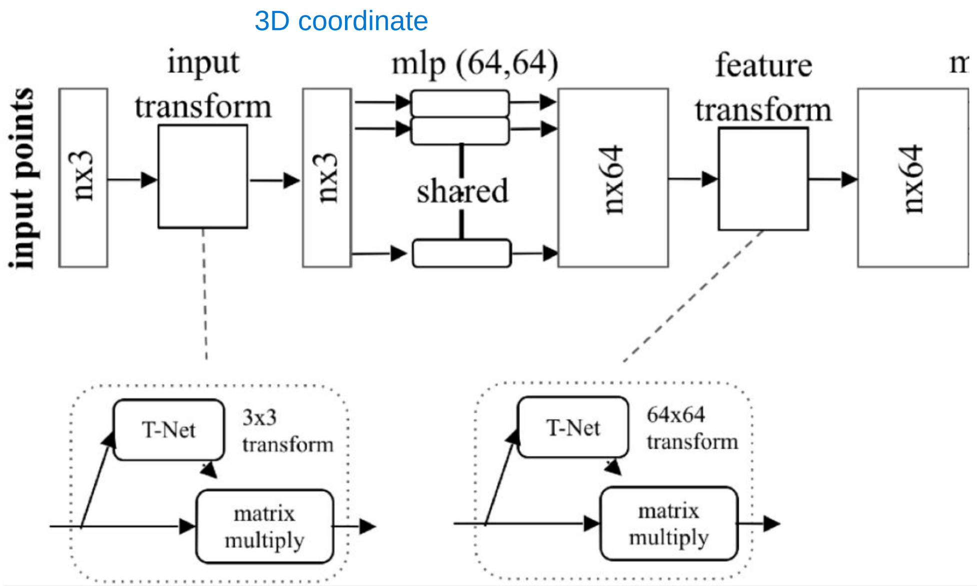


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification

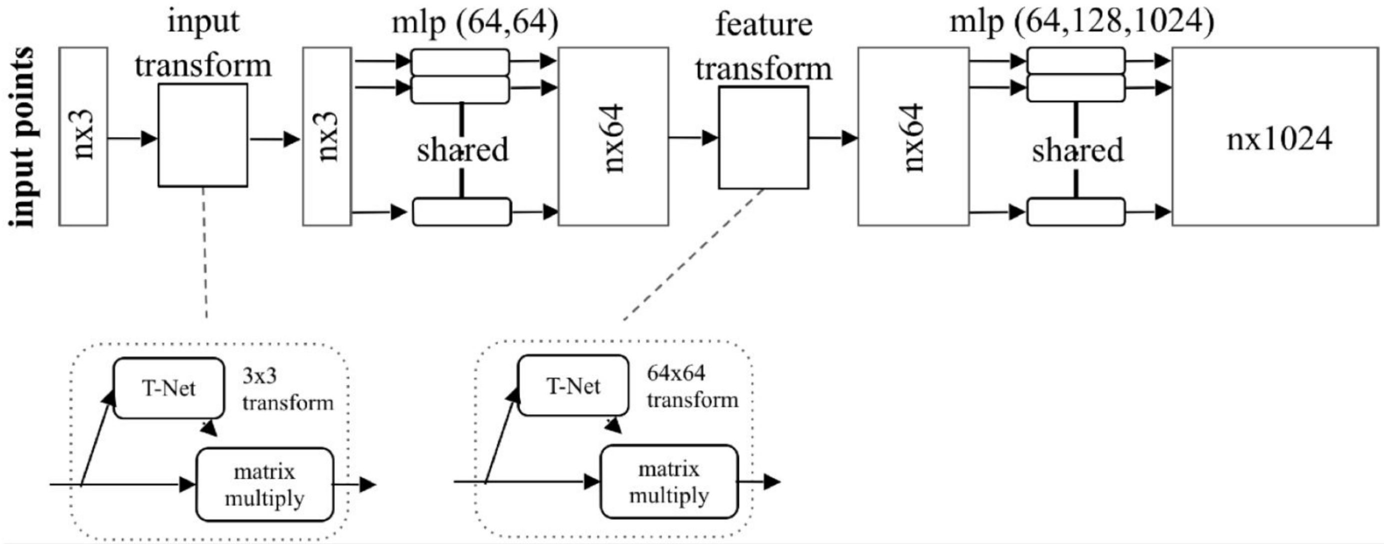


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification

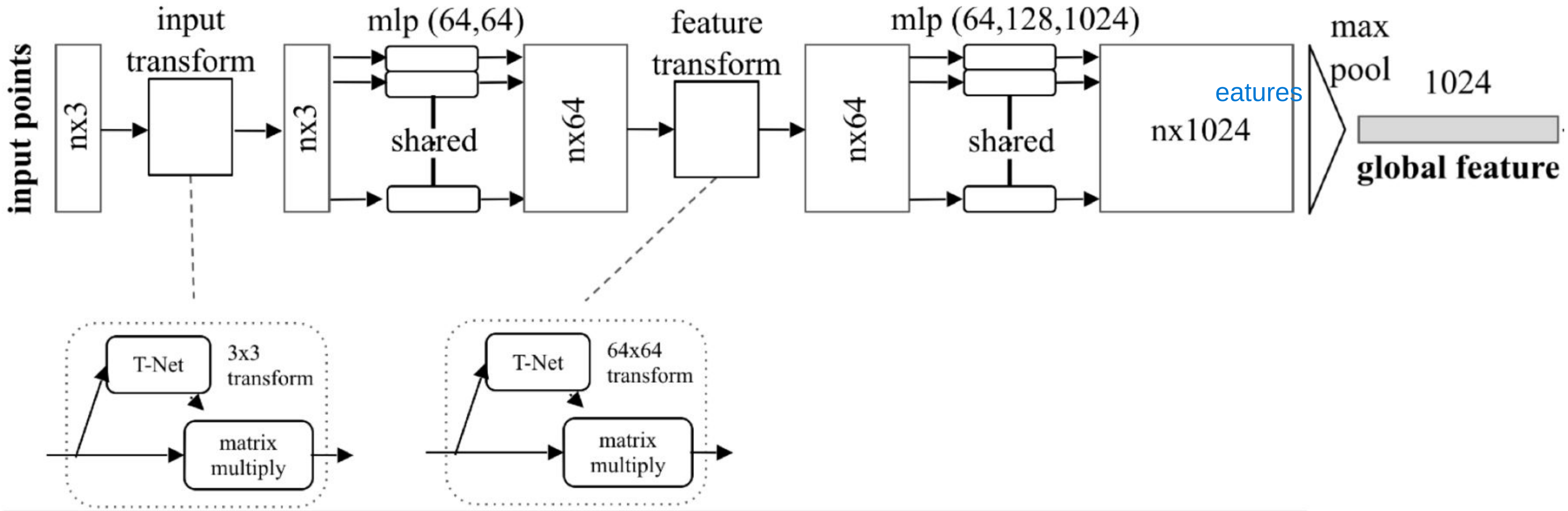


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification

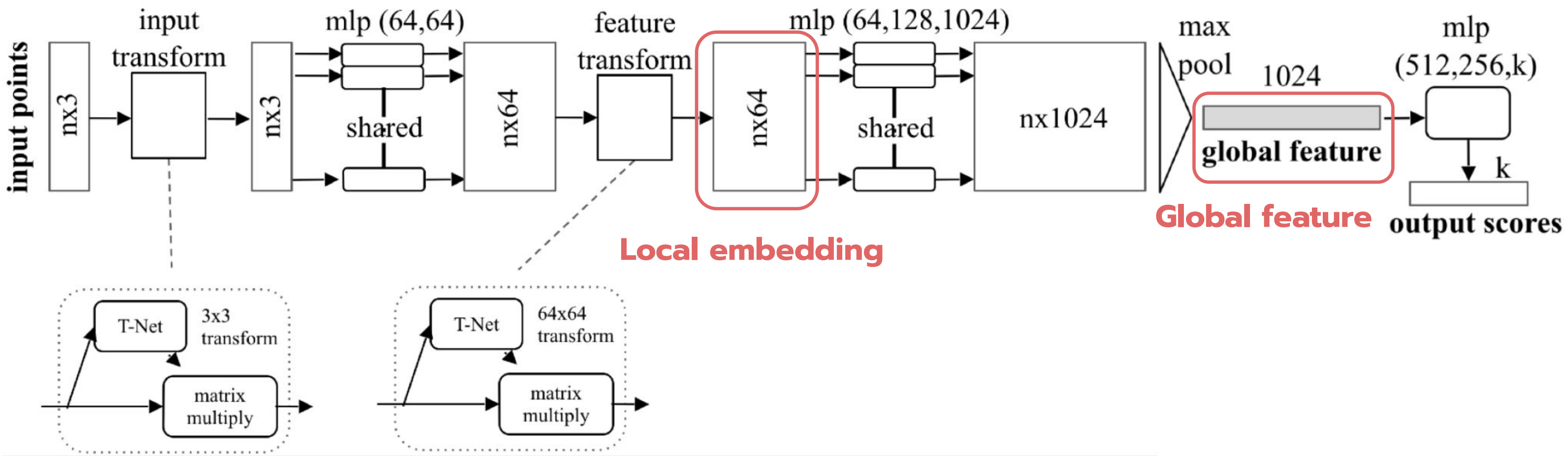


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Classification

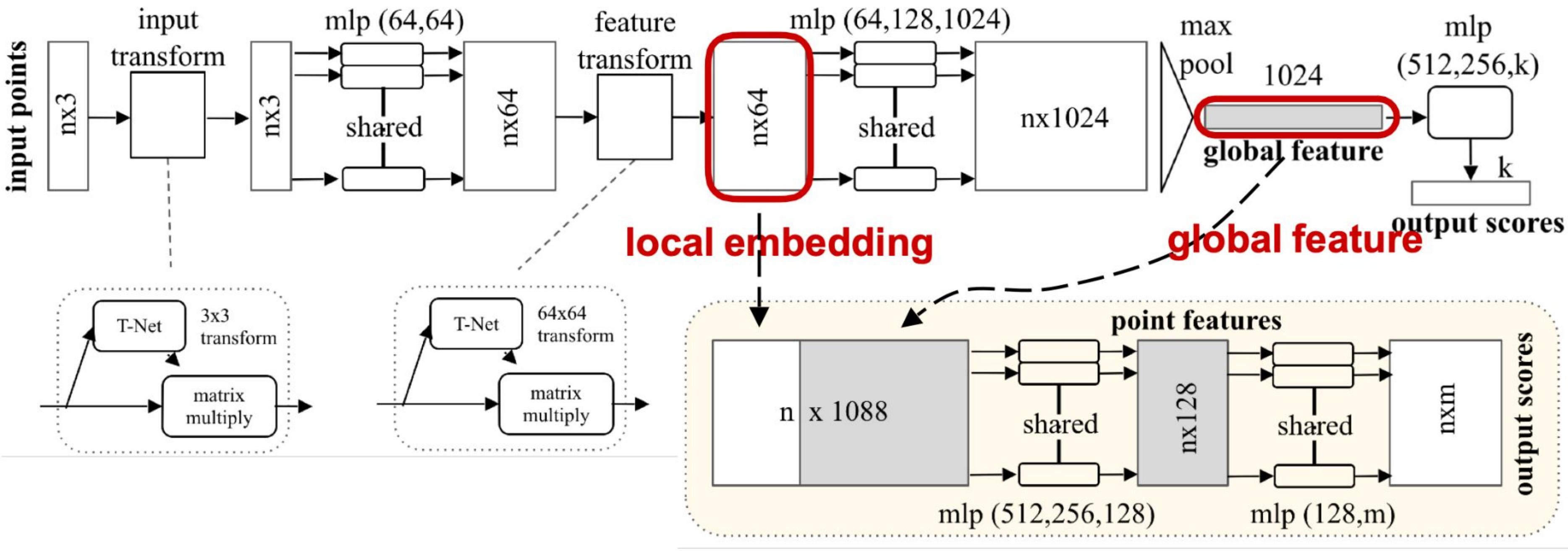


Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet

Segmentation



Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet - Results

Object Classification

3D CNNs

	input	#views	accuracy avg. class	accuracy overall
SPH [12]	mesh	-	68.2	
3DShapeNets [29]	volume	1	77.3	84.7
VoxNet [18]	volume	12	83.0	85.9
Subvolume [19]	volume	20	86.0	**89.2**
LFD [29]	image	10	75.5	-
MVCNN [24]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

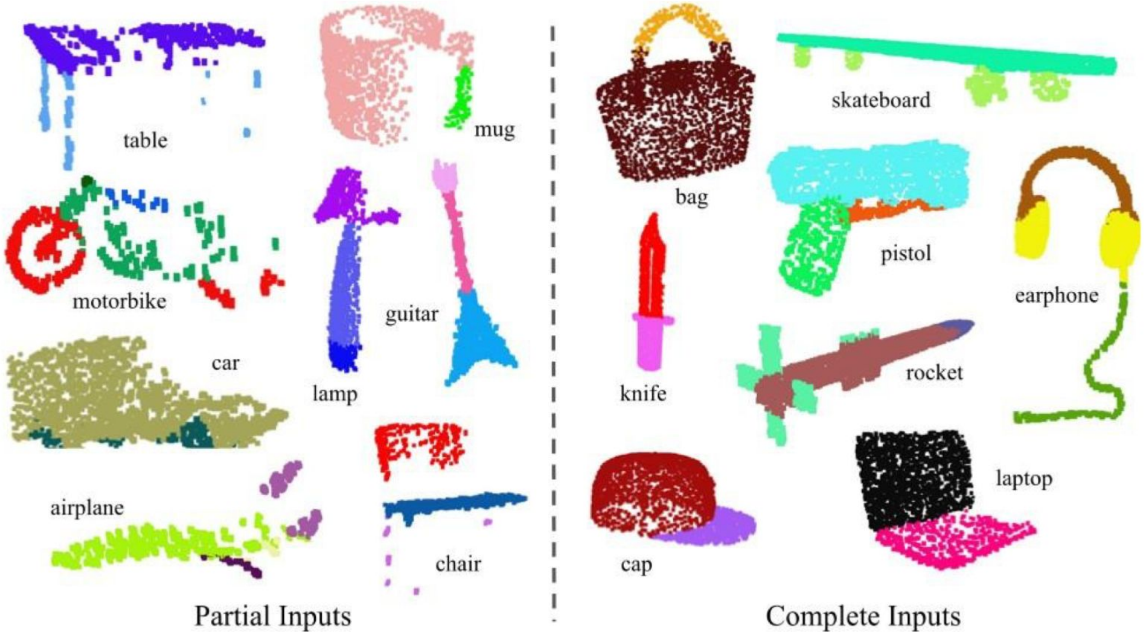
dataset: ModelNet40; metric: 40-class classification accuracy (%)

Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet - Results

Object Part Segmentation



Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: PointNet - Results

Semantic Scene Parsing

dataset: Stanford 2D-3D-S (Matterport scans)

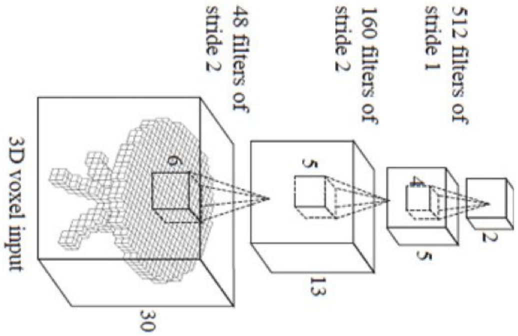
Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

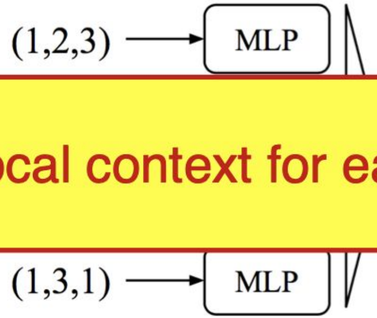
Point Cloud: PointNet - Limitations

Hierarchical feature learning
Multiple levels of abstraction

Global feature learning
Either **one** point or **all** points



3D CNN (Wu et al.)

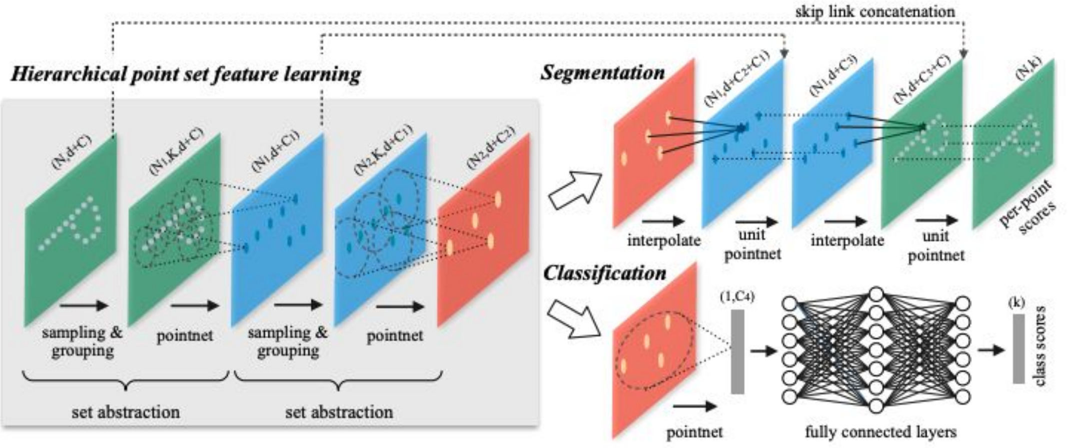


No local context for each point!

PointNet (vanilla) (Qi et al.)

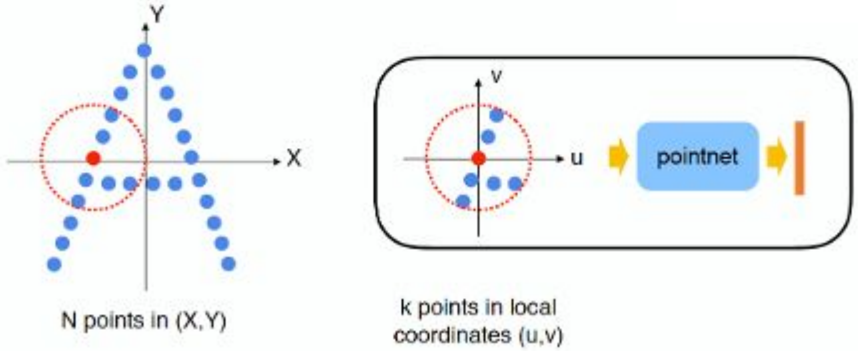
3D Deep Learning

Point Cloud: PointNet



Basic idea: Apply pointnet at local regions.

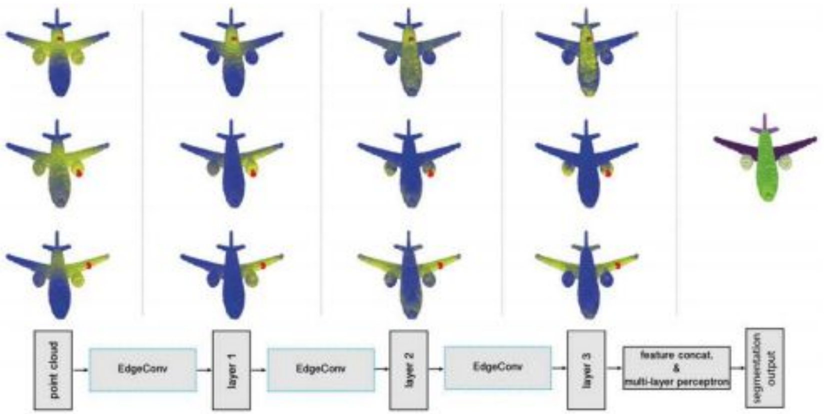
- ✓ Hierarchical feature learning
- ✓ Translation invariant
- ✓ Permutation invariant

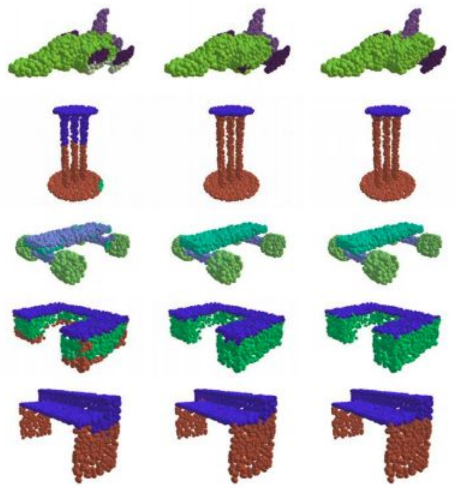


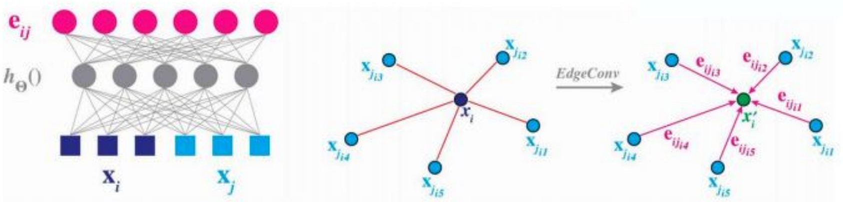
Adopted from: Hao Su, A Tutorial on 3D Deep Learning, CVPR 2017

3D Deep Learning

Point Cloud: Graph NN







Edge convolution operator is defined as

$$h_\theta(x_i, x_j) = h_\theta(x_i, x_j - x_i)$$

$$\mathbf{x}_i' = \mathop{\square}_{j:(i,j)\in\mathcal{E}} h_\Theta(\mathbf{x}_i, \mathbf{x}_j).$$

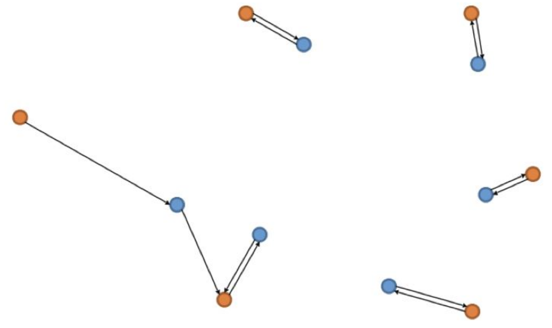
aggregation operation $\square$ (e.g., $\sum$ or max)

3D Deep Learning

Distance Metrics for Point Cloud

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

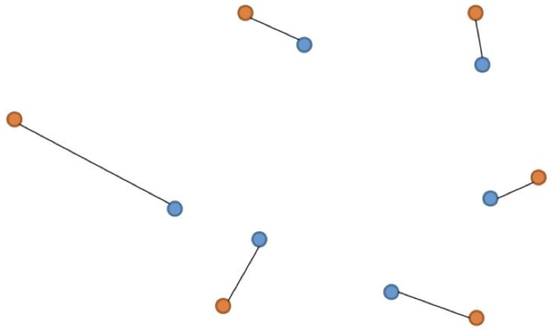
$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2$$



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:

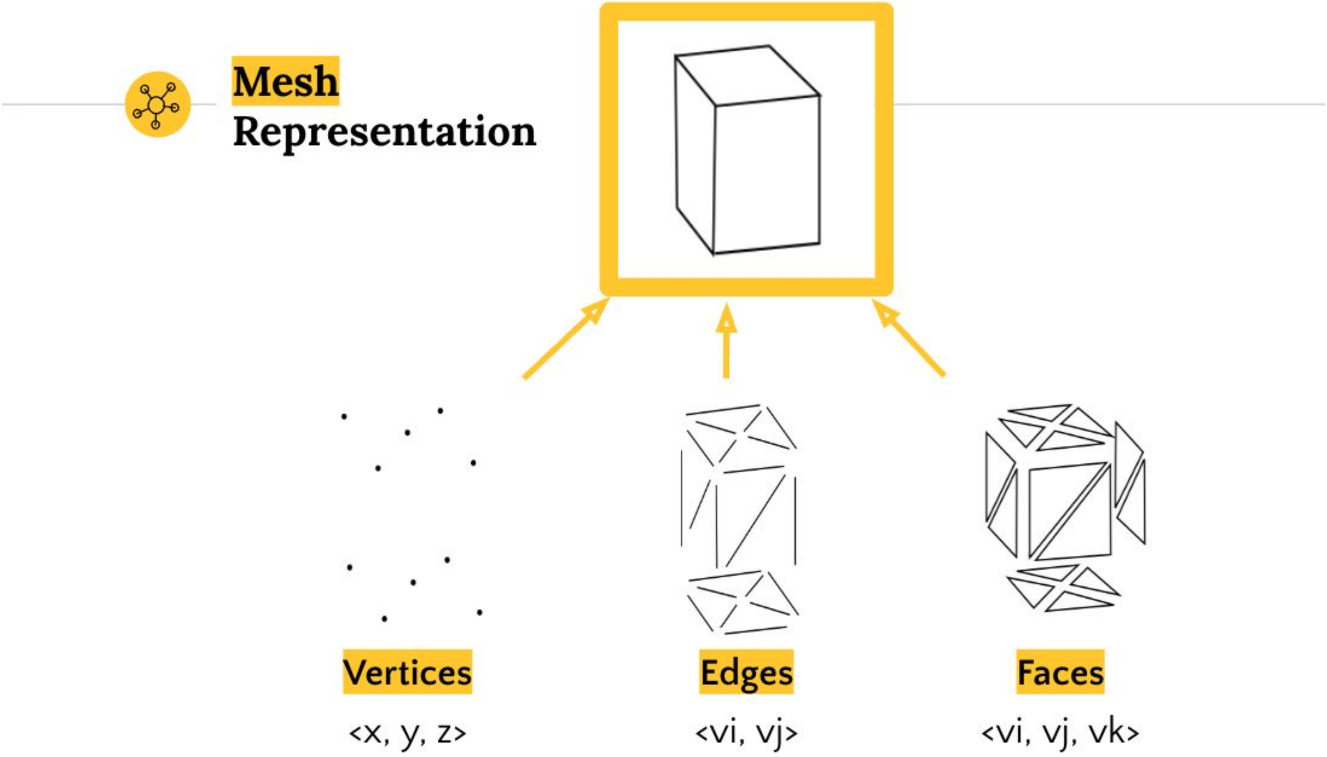
$$d_{EMD}(S_1, S_2) = \min_{\phi : S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

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



3D Deep Learning

Mesh R-CNN:



Adopted from: https://ranahanocka.github.io/MeshCNN/

3D Deep Learning

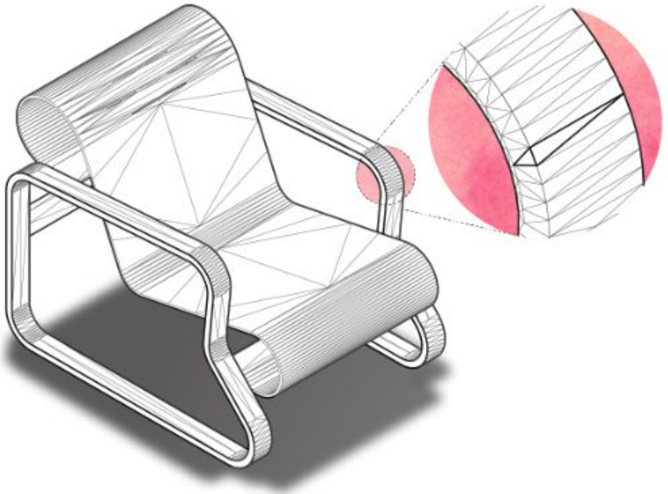
Mesh R-CNN:

Meshes are Irregular

Adapt to the surface

- → Large polygons in flat regions
- → Small polygons in detailed regions

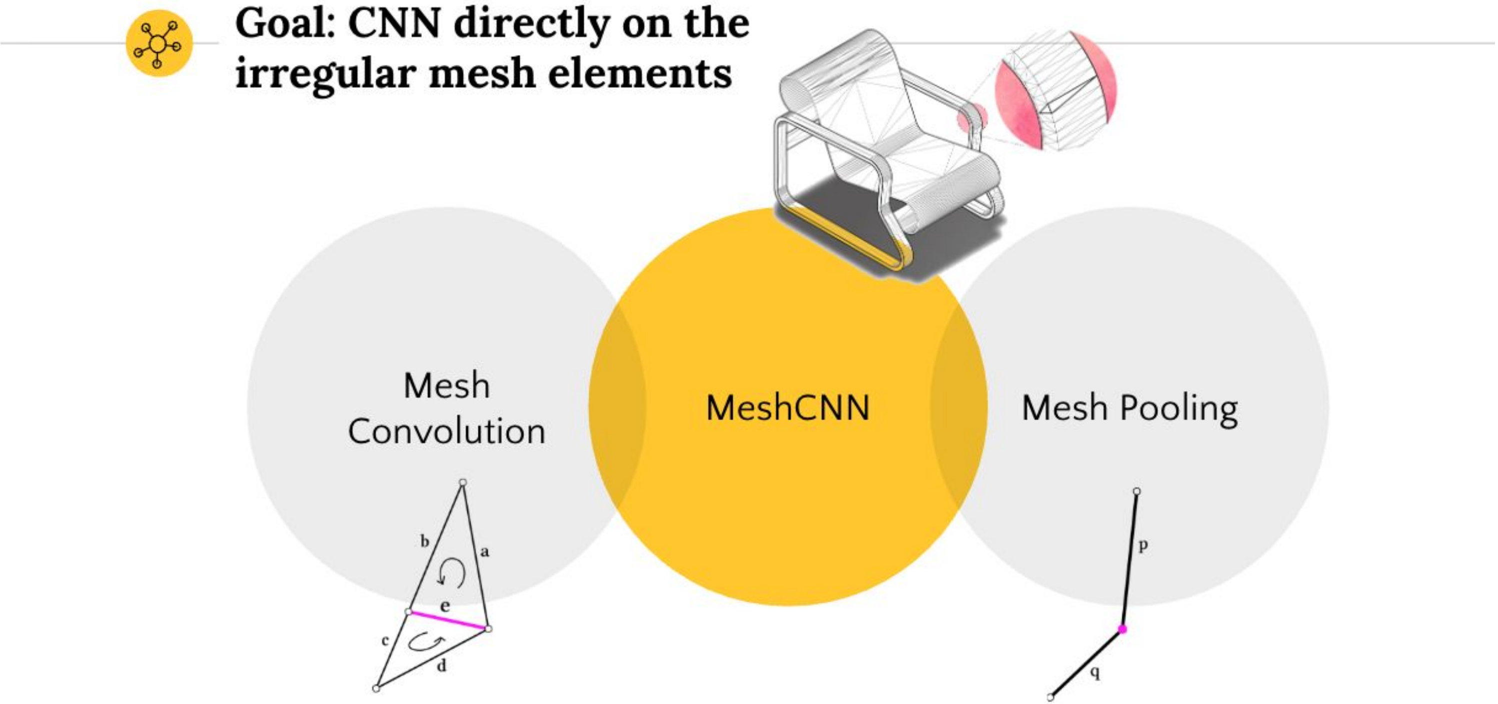
Non-Uniform & Efficient



Adopted from: https://ranahanocka.github.io/MeshCNN/

3D Deep Learning

Mesh R-CNN:



Adopted from: https://ranahanocka.github.io/MeshCNN/

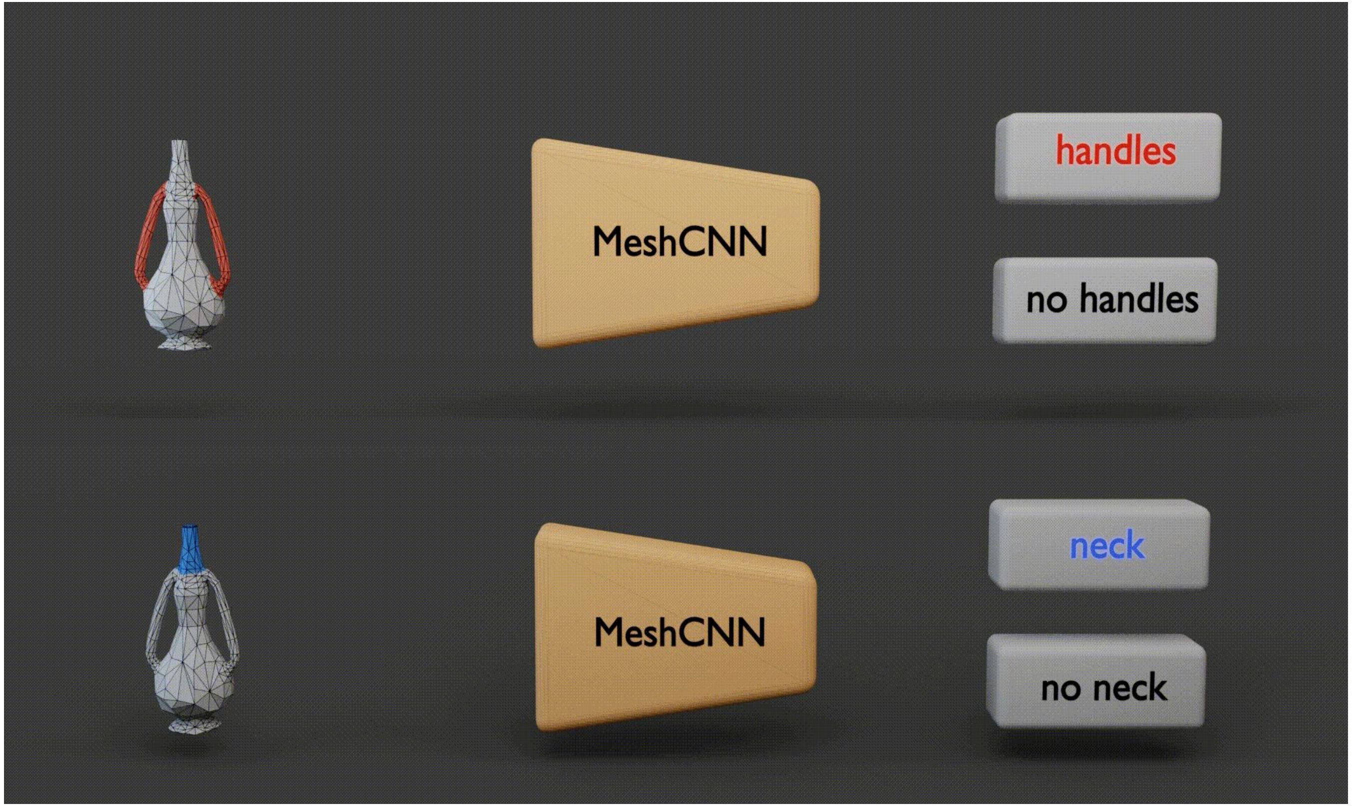
3D Deep Learning

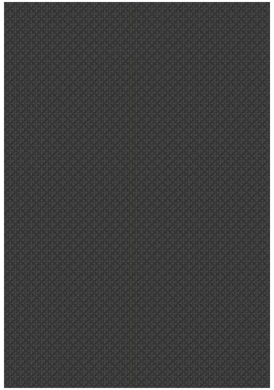
Mesh R-CNN:

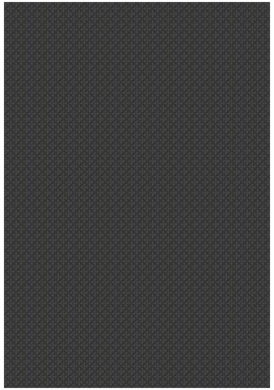
Adopted from: https://ranahanocka.github.io/MeshCNN/

3D Deep Learning

Mesh R-CNN:





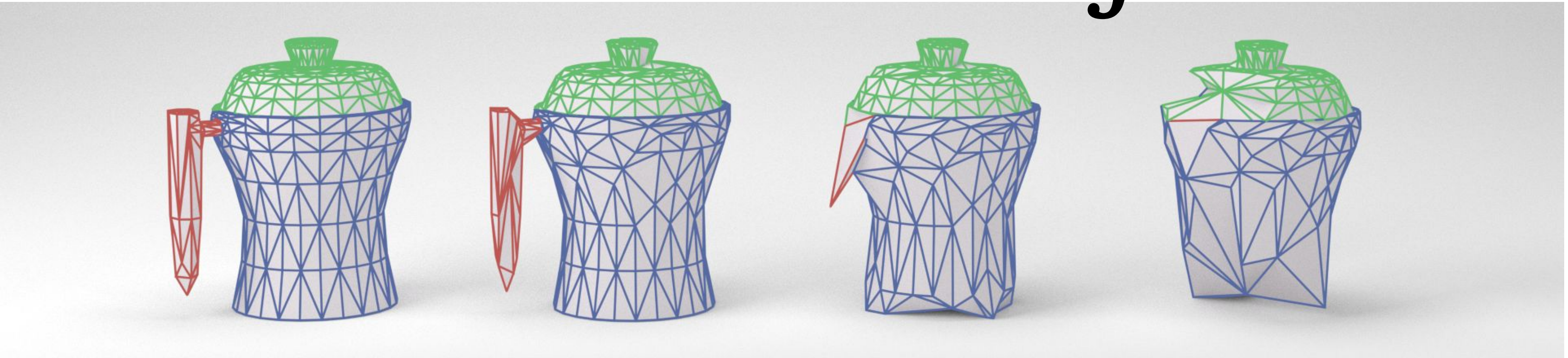


Adopted from: https://ranahanocka.github.io/MeshCNN/

3D Deep Learning

Mesh R-CNN: Results

Segmentation



Human Segmentation

Adopted from: https://ranahanocka.github.io/MeshCNN/

3D Deep Learning

Implicit Representation: Signed Distance Function (SDF)

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation

Jeong Joon Park[1,3†] Peter Florence [2,3†] Julian Straub[3] Richard Newcombe[3] Steven Lovegrove[3]

[1]University of Washington [2]Massachusetts Institute of Technology [3]Facebook Reality Labs

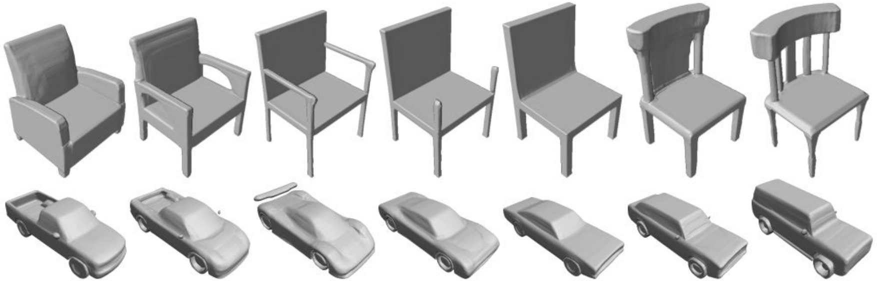
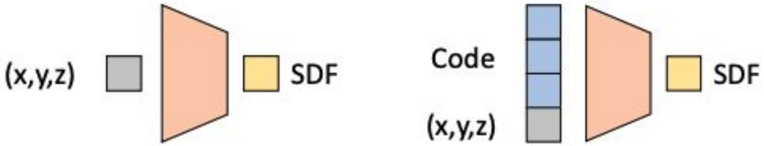
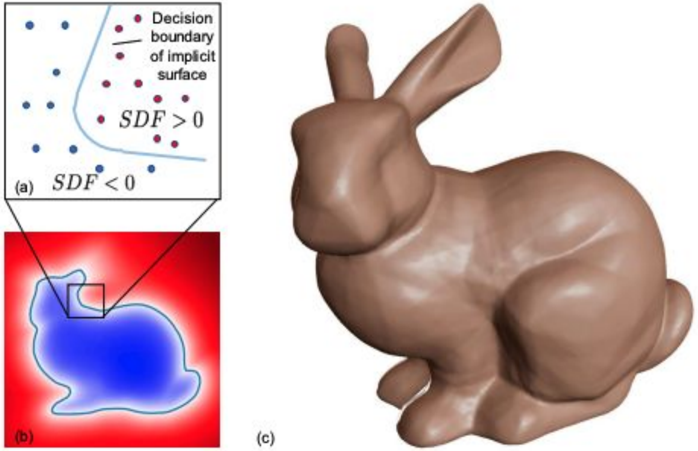
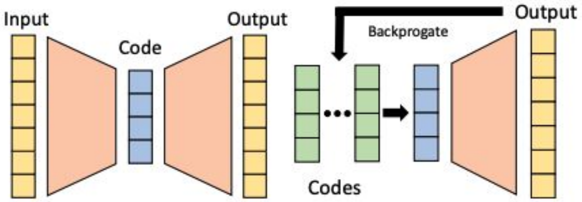


Figure 1: DeepSDF represents signed distance functions (SDFs) of shapes via latent code-conditioned feed-forward decoder networks. Above images are raycast renderings of DeepSDF interpolating between two shapes in the learned shape latent space. Best viewed digitally.



(a) Single Shape DeepSDF **(b)** Coded Shape DeepSDF





(a) Auto-encoder **(b)** Auto-decoder

3D Deep Learning

Implicit Representation: Signed Distance Function (SDF)

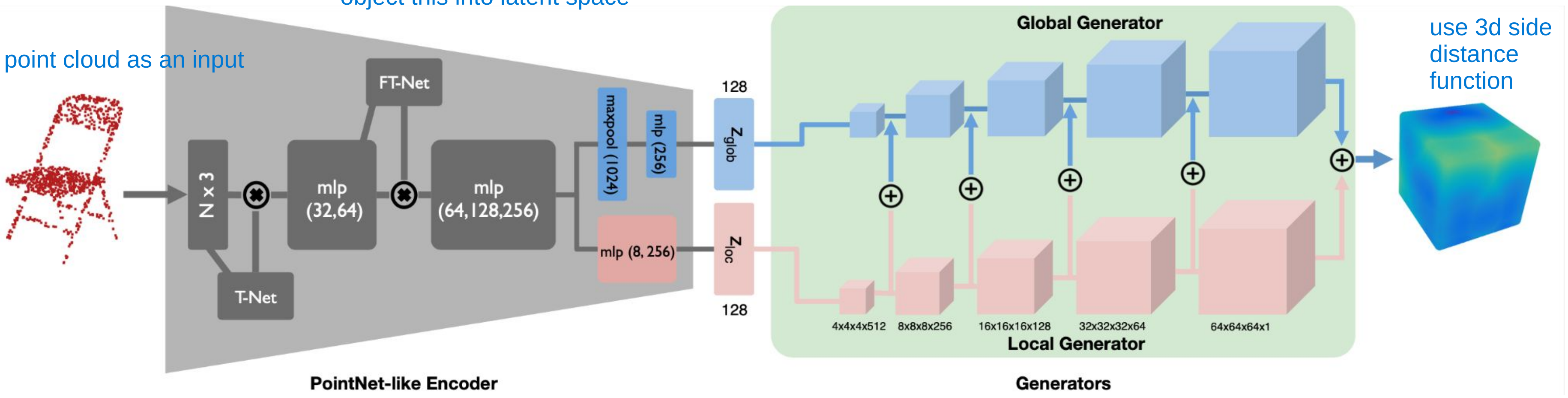
Generating Mesh-based Shapes from Learned Latent Spaces of Point Clouds with VAE GAN

Cherdsak Kingkan, and Koichi Hashimoto
Int. Conf. Pattern Recognition (ICPR2018)

object this into latent space

point cloud as an input

use 3d side distance function



3D Deep Learning

Implicit Representation: Signed Distance Function (SDF)

trun into polygon mest using 3D meshing cube

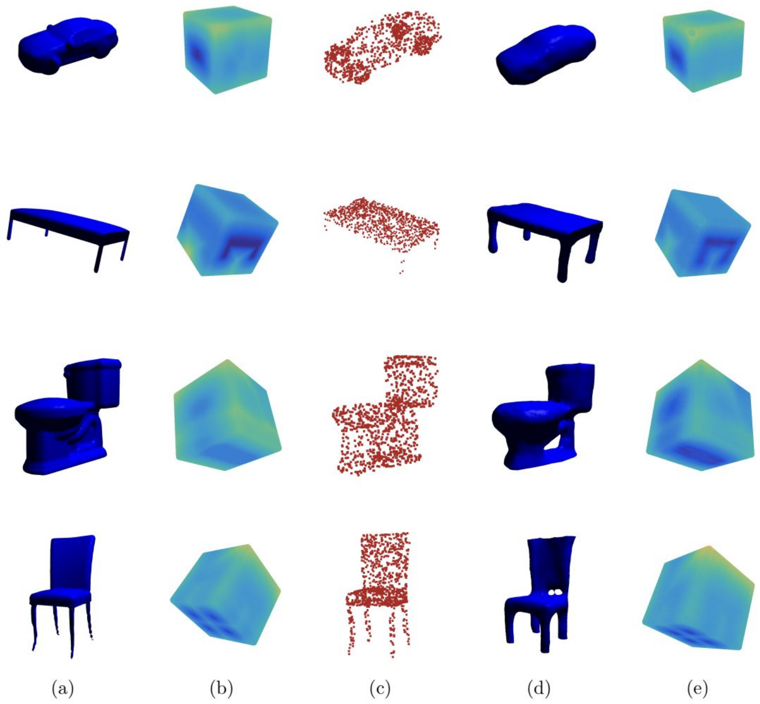


Figure 2.14: Examples showing mesh, SDFs, and point clouds of objects and its corresponding reconstructed mesh and SDFs. (a)-(c) show the mesh, SDFs, and point clouds of objects. (d)-(e) show the corresponding reconstructed mesh-based and SDFs of objects.