

Advanced Deep Learning Architectures

COMP 5214 & ELEC 5680

Instructor: Dr. Qifeng Chen

<https://cqd.io>

Spatial transformer networks

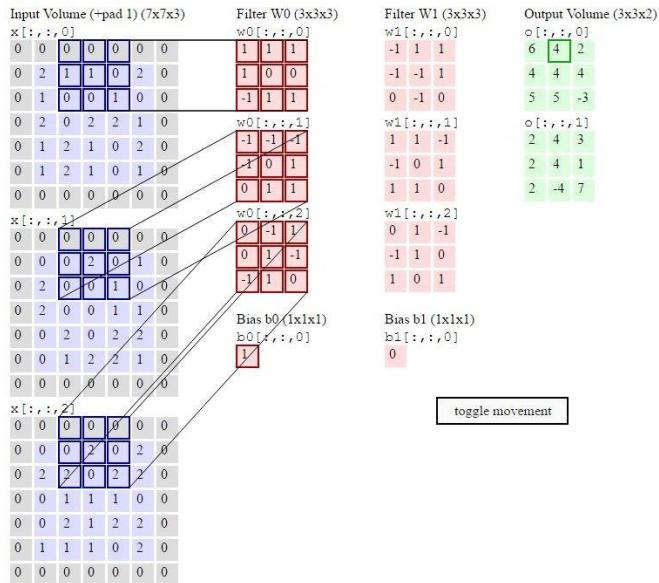
Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu

Why do we need Spatial transformer networks?

Are Convolutional Neural Networks invariant to...

- Scale?
- Rotation?
- Translation?

Why do we need Spatial transformer networks?



CS231n: [Convolutional Neural Networks for Visual Recognition](#) (Stanford)

Why do we need Spatial transformer networks?

Are Convolutional Neural Networks invariant to...

- Scale? **No**
- Rotation?
- Translation?

Why do we need Spatial transformer networks?

Are Convolutional Neural Networks invariant to...

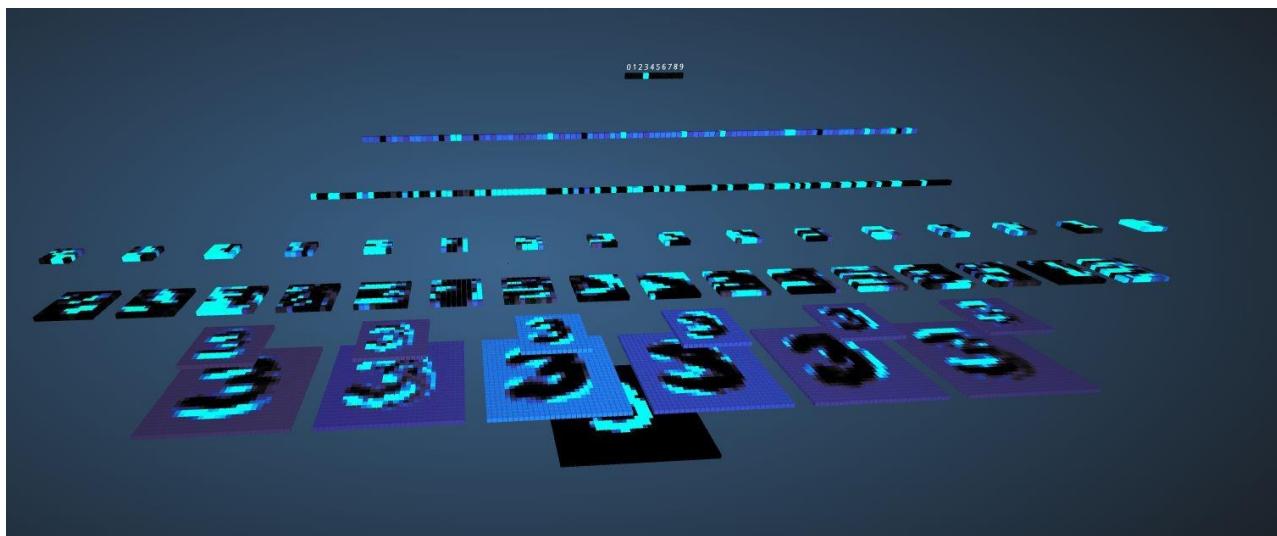
- Scale? **No**
- Rotation? **No**
- Translation?

Why do we need Spatial transformer networks?

Are Convolutional Neural Networks invariant to...

- Scale? **No**
- Rotation? **No**
- Translation? **Partially**

Why do we need Spatial transformer networks?

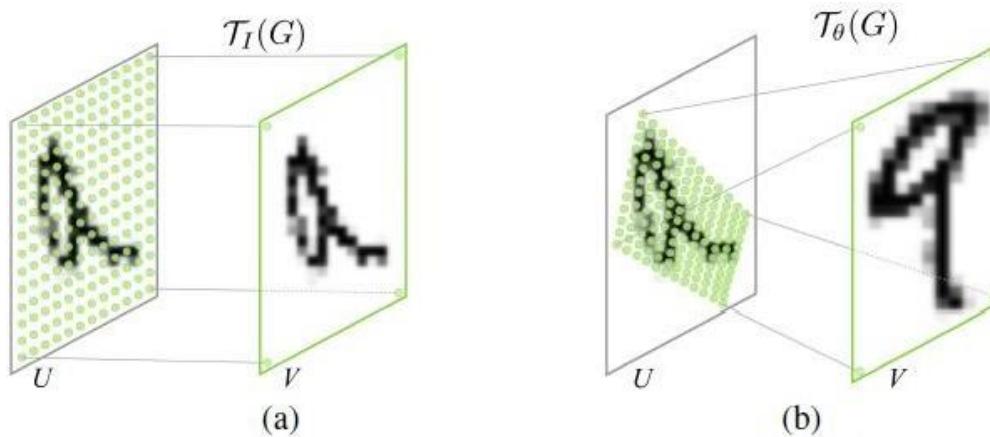


A. W. Harley, "[An Interactive Node–Link Visualization of Convolutional Neural Networks](#)," in ISVC, pages 867–877, 2015

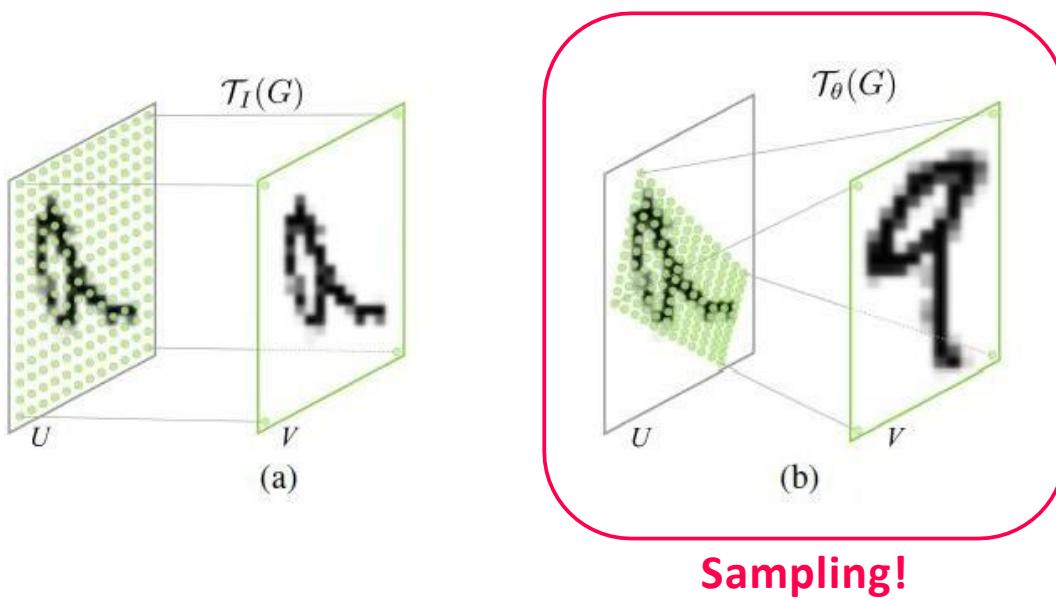
Intuition behind Spatial transformers



Intuition behind Spatial transformers



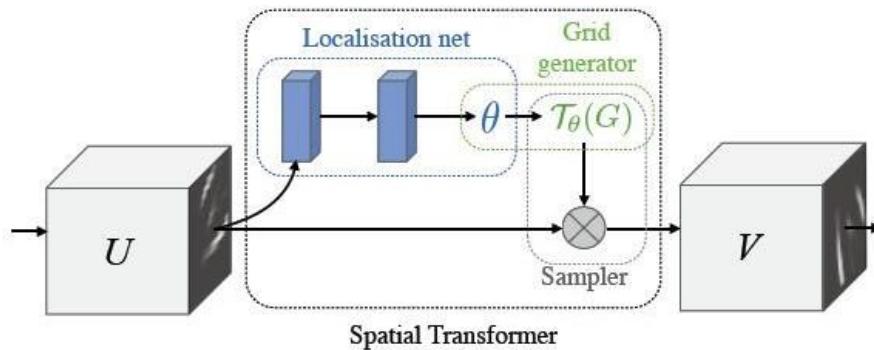
Intuition behind Spatial transformers



Formulating Spatial transformers

Three main **differentiable** blocks:

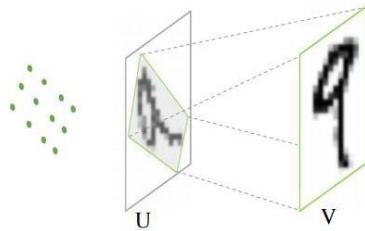
- Localisation network
- Grid generator
- Sampler



Grid generator: Examples

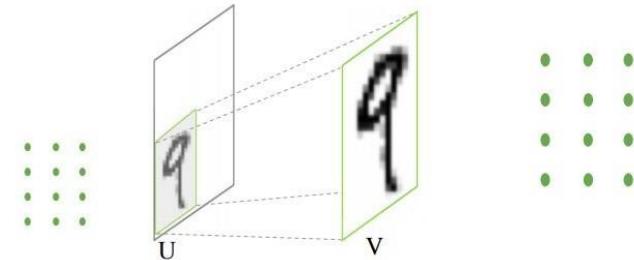
Affine transform

$$\begin{pmatrix} x_i^s \\ y_i^s \\ 1 \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathbf{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



Attention model

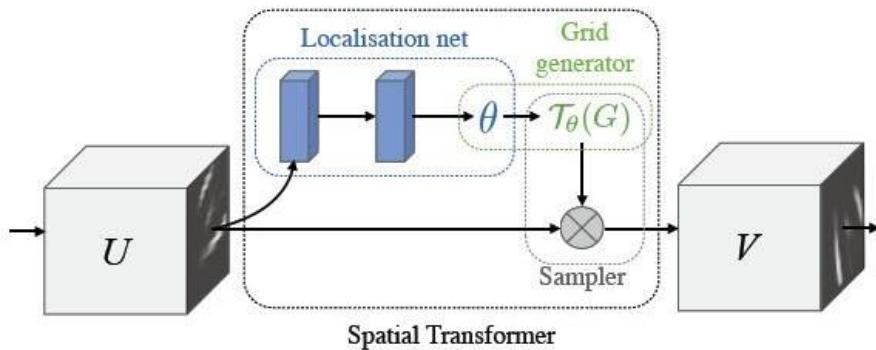
$$\begin{pmatrix} x_i^s \\ y_i^s \\ 1 \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathbf{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$



(x_i^t, y_i^t) – coordinates in the target (output) feature map

(x_i^s, y_i^s) – coordinates in the source (input) feature map

Sampler: Mathematical formulation



$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \quad \forall i \in [1 \dots H'W'] \quad \forall c \in [1 \dots C]$$

Annotations:

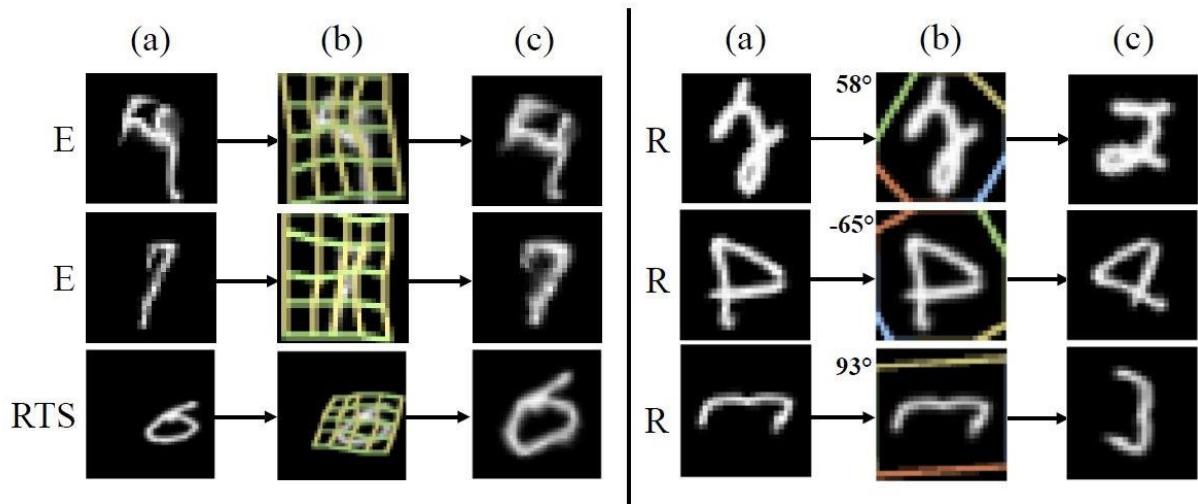
- Generic sampling kernel (green arrow pointing to $k(x_i^s - m; \Phi_x)$)
- From the grid generator (red arrow pointing to $y_i^s - n$)
- All pixels in the output feature map (orange box around $\forall i \in [1 \dots H'W']$)

Experiment: Distorted MNIST

0 0 0 1 1 1 1 1 1 2
2 2 2 2 2 2 2 3 3 3
3 4 4 4 4 4 5 5 5 5
6 6 7 7 7 7 7 8 8 8
8 8 9 9 9 9 9 9 9 9

Experiment: Distorted MNIST

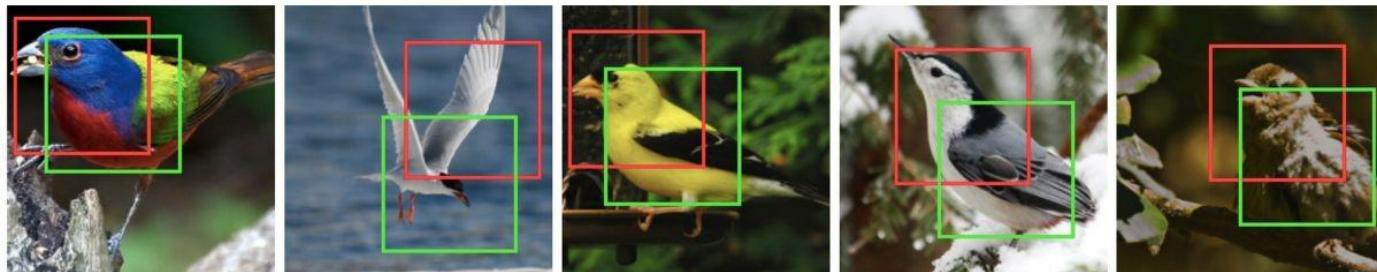
Model	MNIST Distortion			
	R	RTS	P	E
FCN	2.1	5.2	3.1	3.2
CNN	1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5
	Proj	1.3	0.9	1.4
	TPS	1.1	0.8	1.4
ST-CNN	Aff	0.7	0.5	0.8
	Proj	0.8	0.6	0.8
	TPS	0.7	0.5	0.8



Distortions: Rotation, Translation, Projective, Elastic
 Transformations: Affine, Projective, Thin Plate Spline (TPS)

Other experiments: Applications of spatial transformers

- Street View House Numbers
- Fine-grained classification



Deep Learning for 3D Vision

Topics

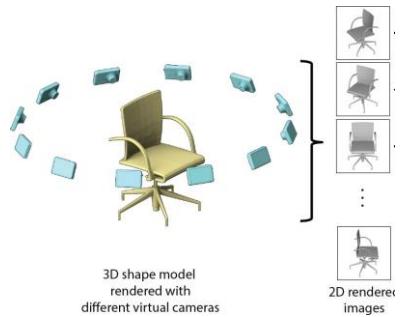
- **3D Data**
- Classification
- Segmentation and Detection
- Reconstruction

The Representation Challenge of 3D Deep Learning

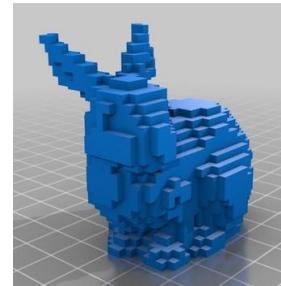
Rasterized form
(regular grids)

Geometric form
(irregular)

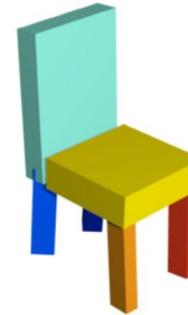
The Representation Challenge of 3D Deep Learning



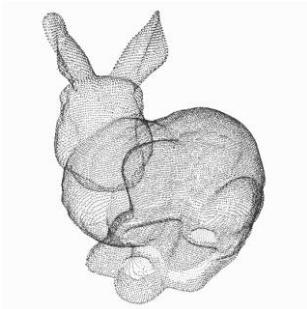
Multi-view



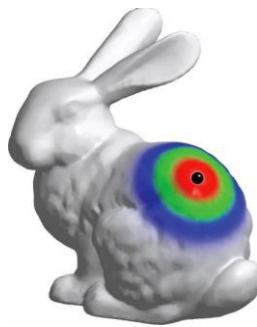
Volumetric



Part Assembly



Point Cloud



Mesh (Graph CNN)

$$F(x) = 0$$

Implicit Shape

Datasets for 3D Objects

Large-scale Synthetic Objects: ShapeNet



3DScan: Consumer-grade 3D scanning (click to open)

ModelNet: absorbed by ShapeNet

Chang et al., "ShapeNet: An Information-Rich 3D Model Repository", *arXiv*

Wu et al., "3D ShapeNets: A deep representation for volumetric shapes", *CVPR 2015*

Choi et al., "A Large Dataset of Object Scans", *arXiv*

Datasets for 3D Object Parts

Fine-grained Part: PartNet (ShapeNetPart2019)

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



Datasets for Indoor 3D Scenes

Large-scale Synthetic Scenes: SceneNet

- 3D meshes
- 5M Photorealistic Images



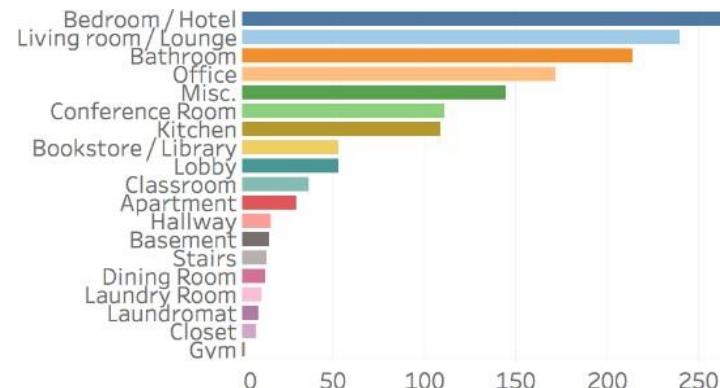
Ankur et al., "Understanding RealWorld Indoor Scenes with Synthetic Data", CVPR 2016

McCormac et al., "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?", ICCV 2017

Datasets for Indoor 3D Scenes

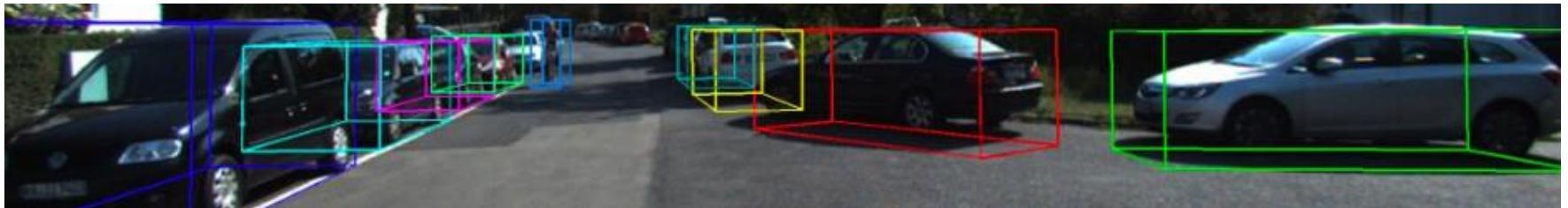
Large-scale Scanned Real Scenes: ScanNet

- 2.5 M Views in 1500 RGBD scans
- 3D camera poses
- surface reconstructions
- Instance-level semantic segmentations



Datasets for Outdoor 3D Scenes

KITTI: LiDAR data, labeled by 3D b.boxes



Semantic KITTI: LiDAR data, labeled per point



Waymo Open Dataset: LiDAR data, labeled by 3D b.boxes



Topics

- 3D Data
- Classification
- Segmentation and Detection
- Reconstruction

Task: 3D Classification



This is a chair!

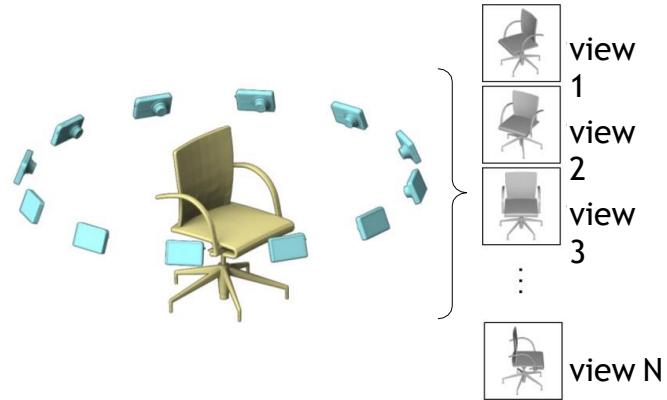
Covered methods: Volumetric CNN, OctNet, O-CNN, SparseConvNet, PointNet, PointNet++, RS CNN, DGCNN, Point ConvNet, KPConv, Monte Carlo Point Convolution, PConv, Multi-View CNN, Spectral CNN, Synchronized Spectral CNN, Spherical CNN

Multi-View CNN

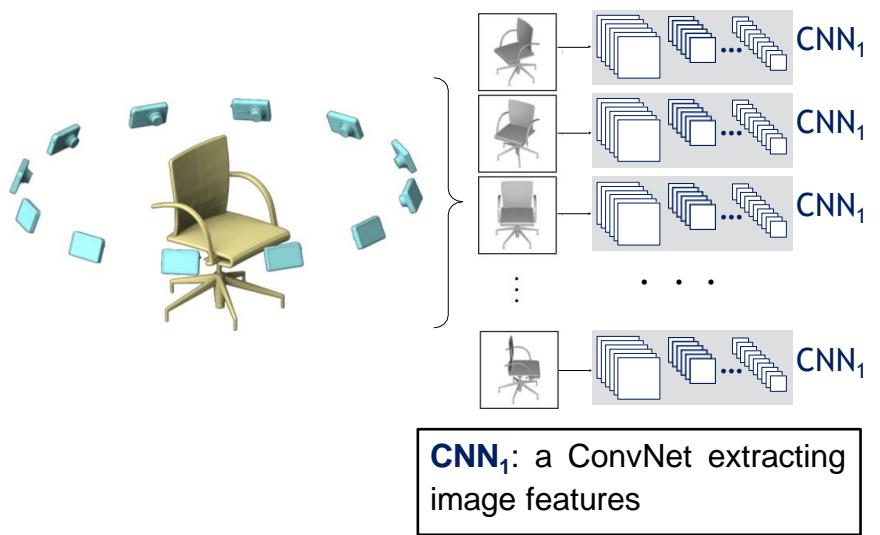
Given an Input Shape



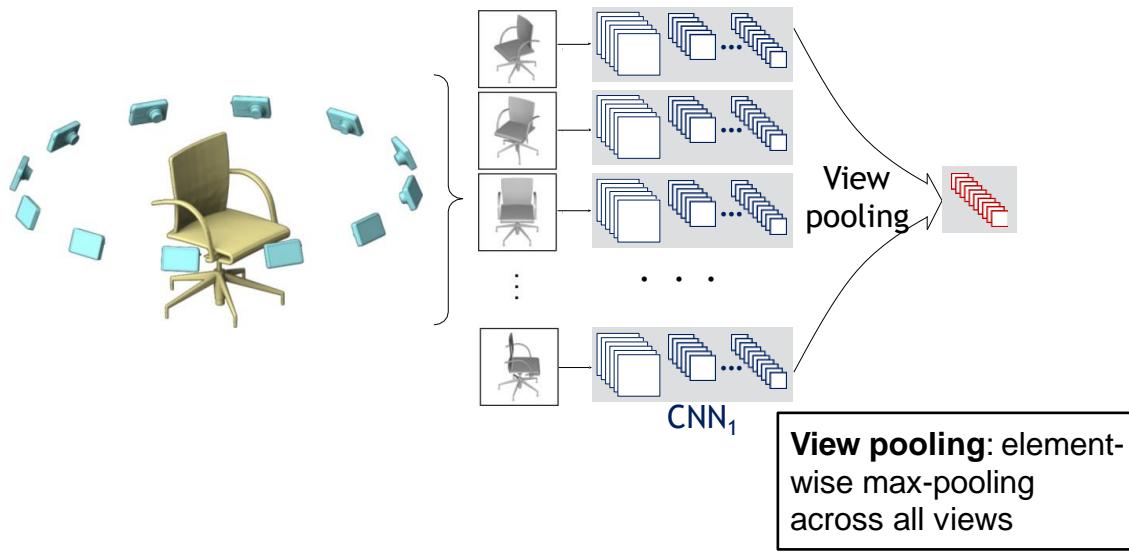
Render with Multiple Virtual Cameras



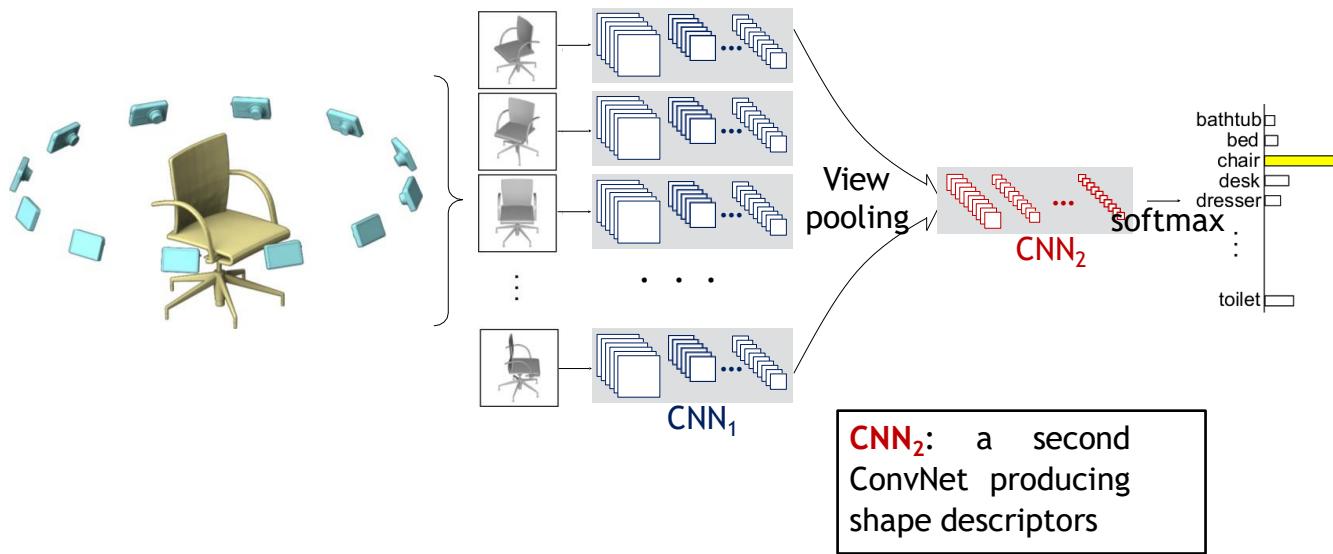
The Rendered Images are Passed through CNN_1 for Image Features



All Image Features are Combined by View Pooling



... and then Passed through CNN_2 and to Generate Final Predictions



Experiments – Classification & Retrieval

Method	Classification (Accuracy)	Retrieval (mAP)
Non-deep {	SPH [16]	68.2%
	LFD [5]	75.5%
	3D ShapeNets [37]	77.3%
	FV, 12 views	84.8%
	CNN, 12 views	88.6%
	MVCNN, 12 views	89.9%
	MVCNN+metric, 12 views	89.5%
	MVCNN, 80 views	90.1%
	MVCNN+metric, 80 views	90.1%
		79.5%

On ModelNet40

[credit: Hang Su]

- 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

Volumetric CNN

Can we use CNNs without 2D-3D projection?

Straight-forward idea: 3D native convolution

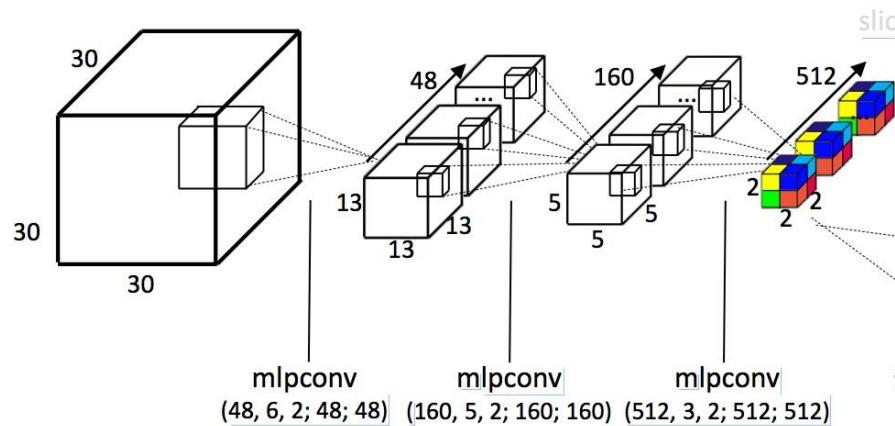
Voxelization

Represent the occupancy of regular 3D grids

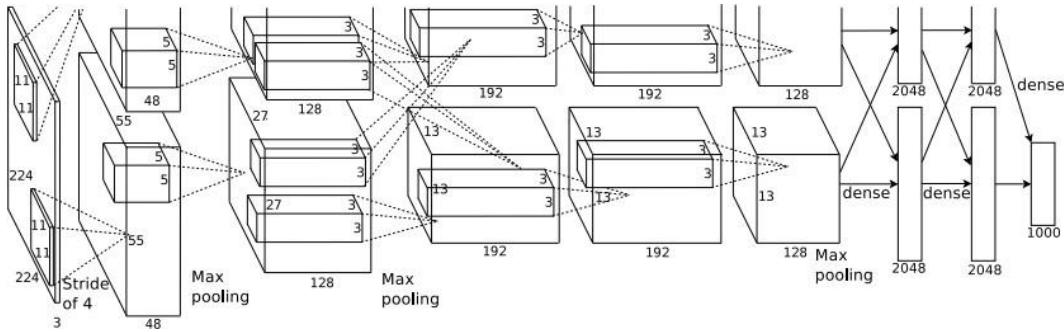


3D CNN on Volumetric Data

3D convolution uses 4D kernels



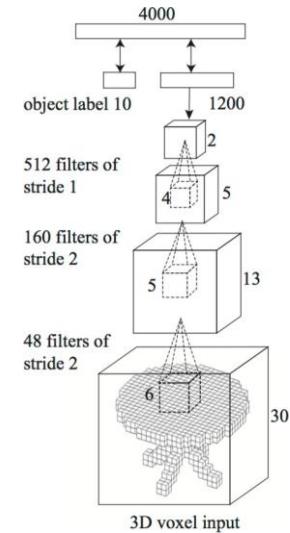
Complexity Issue



AlexNet, 2012

Input resolution: 224x224

$$224 \times 224 = 50176$$

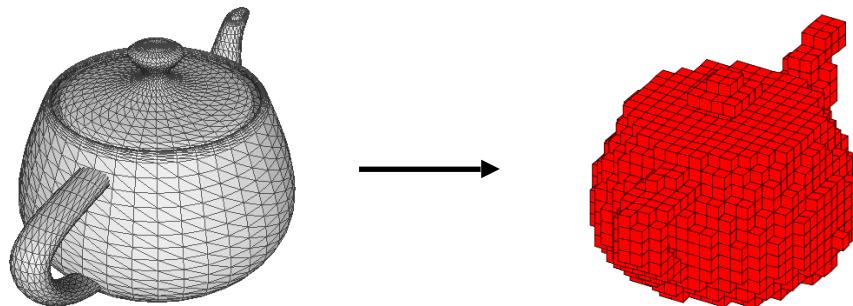


3DShapeNets,
2015

Input resolution: 30x30x30

$$224 \times 224 = 27000$$

Complexity Issue



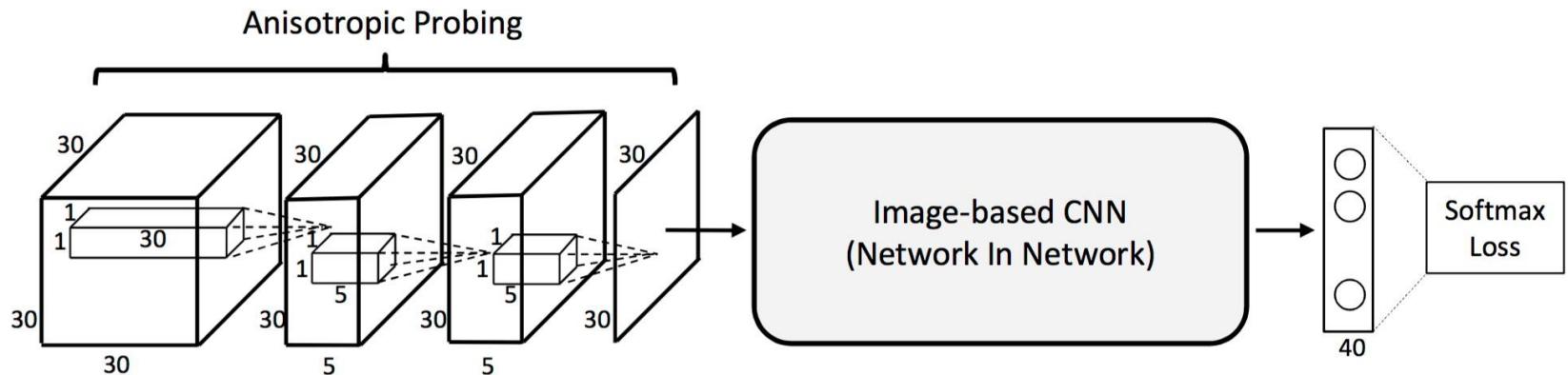
Polygon Mesh

Occupancy Grid
 $30 \times 30 \times 30$

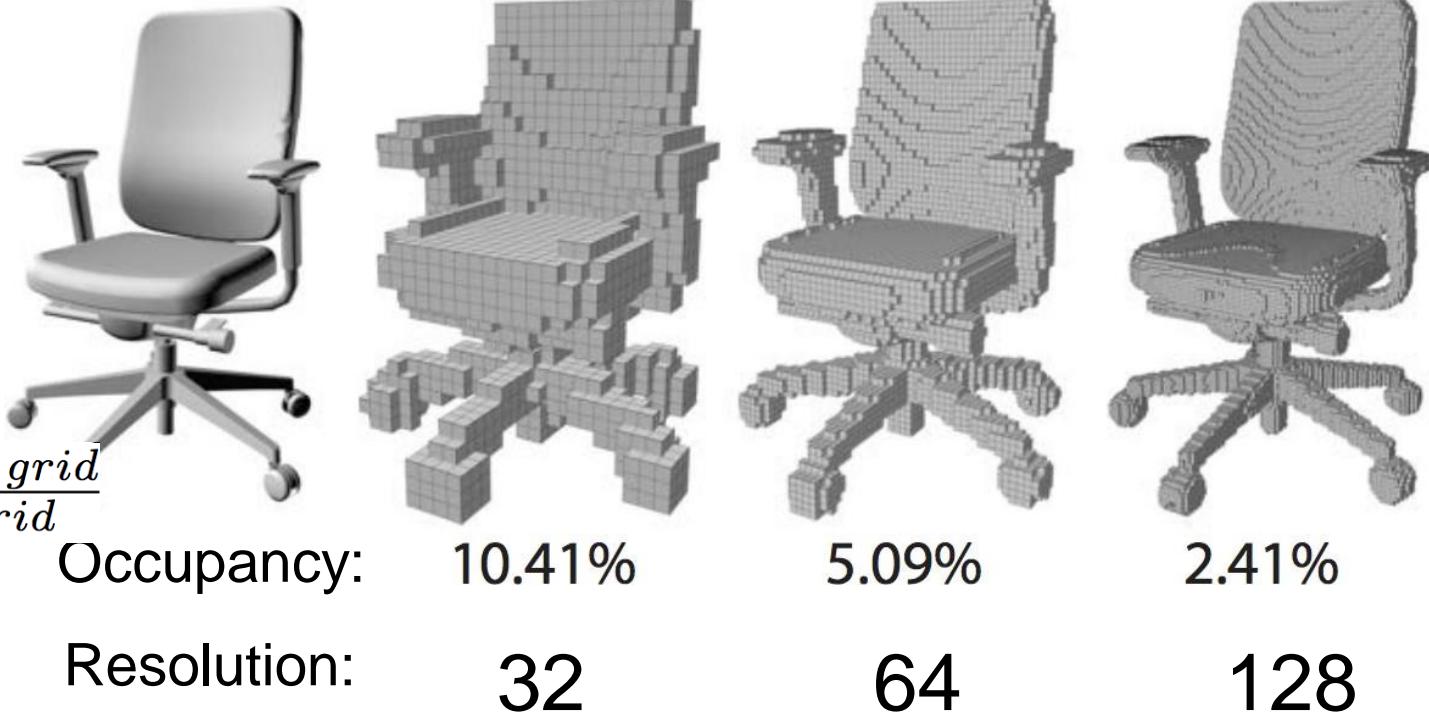
Information loss in voxelization

Idea 1: Learn to Project

*Idea: “X-ray” rendering + Image (2D) CNNs
very low #param, very low computation*

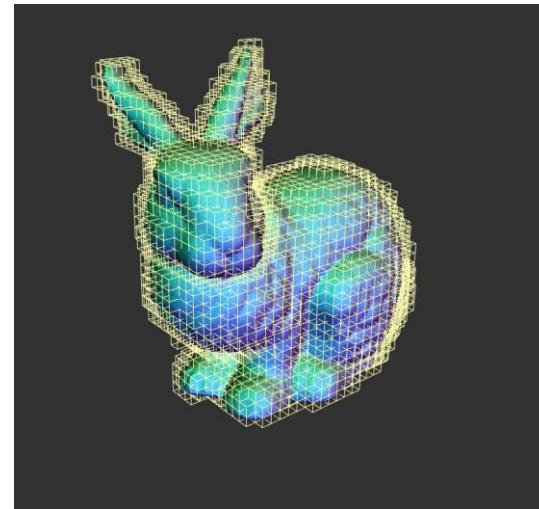
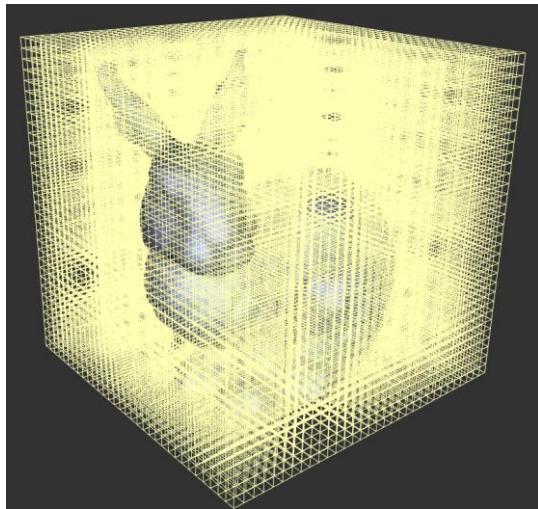


More Principled: Leverage Sparsity of 3D Surface Data



Store only the Occupied Grids

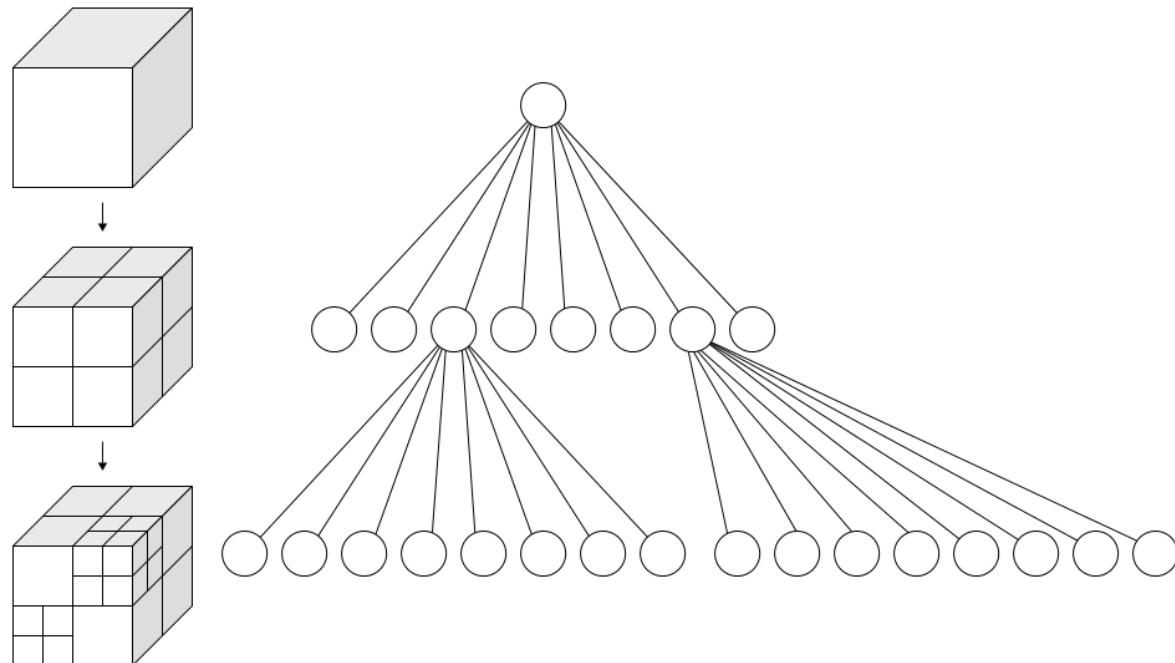
- Store the sparse surface signals
- Constrain the computation near the surface



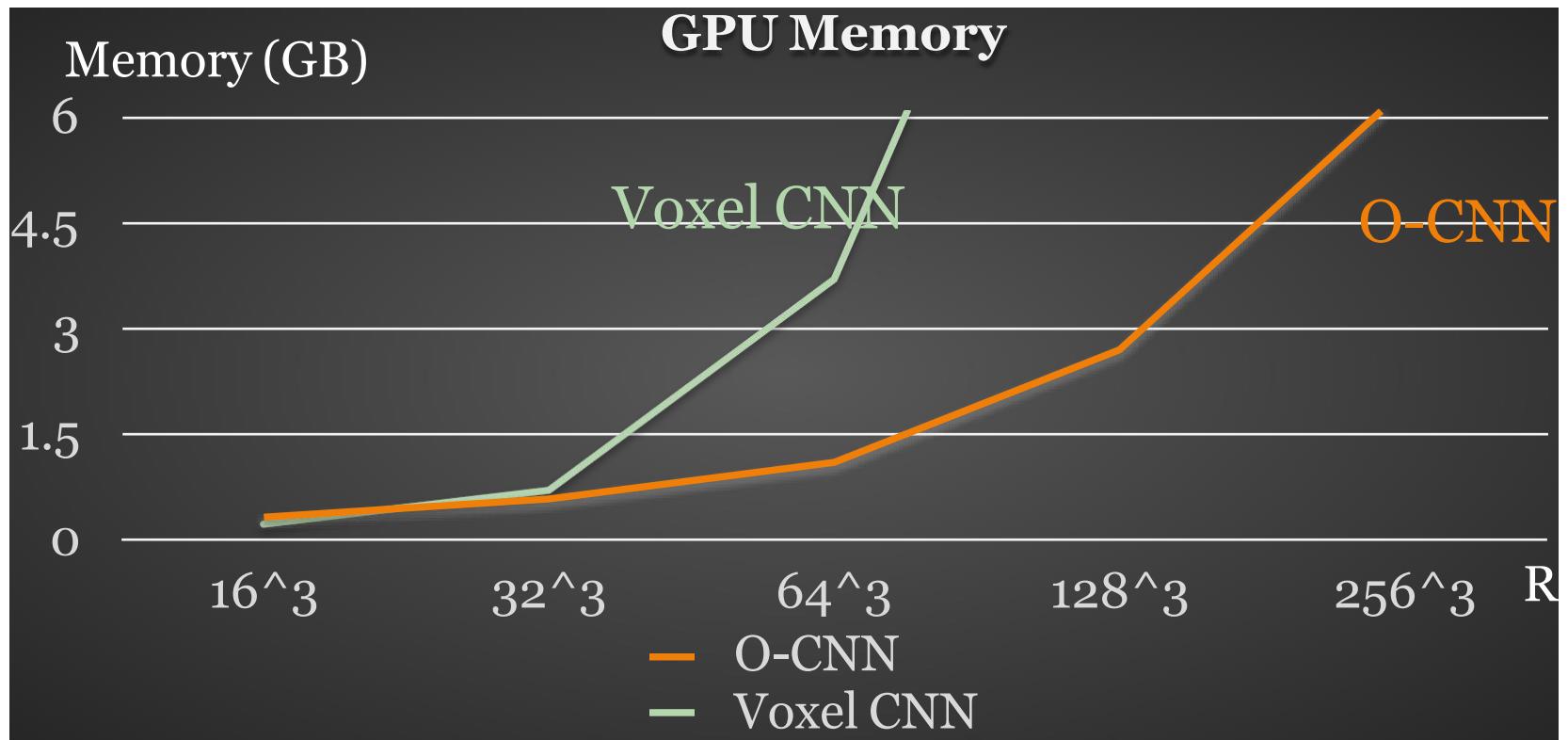
Octree: Recursively Partition the Space

Each internal node has exactly eight children

Neighborhood searching: Hash table



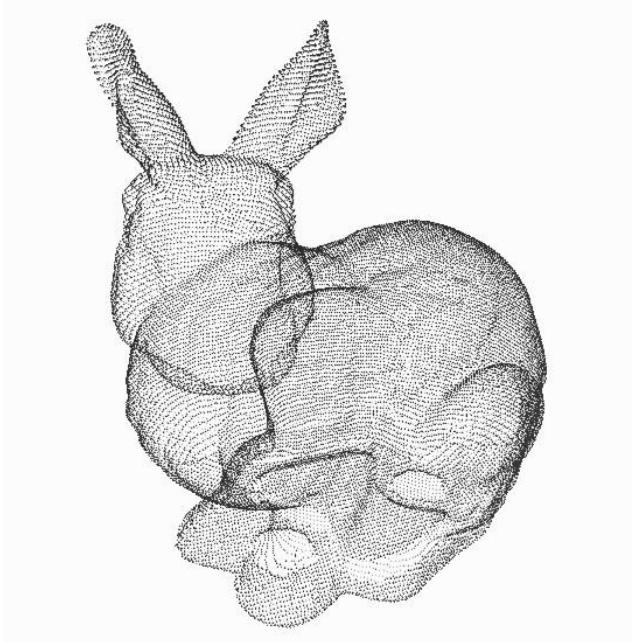
Memory Efficiency



Implementation

- SparseConvNet
 - [https://github.com/facebookresearch/
SparseConvNet](https://github.com/facebookresearch/SparseConvNet)
 - Uses ResNet architecture
 - State-of-the-art for 3D analysis
 - Takes time to train

Point Networks



Point cloud
(The most common 3D sensor data)

Directly Process Point Cloud Data

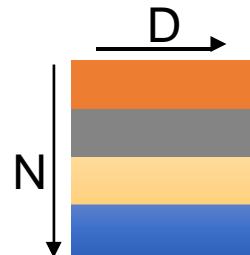
End-to-end learning for **unstructured**,
unordered point data



Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation", CVPR
2017
Zaheer, Manzil, et al. "Deep sets", NeurIPS 2017

Permutation invariance

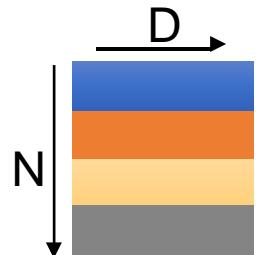
Point cloud: N **orderless** points, each represented by a D dim coordinate



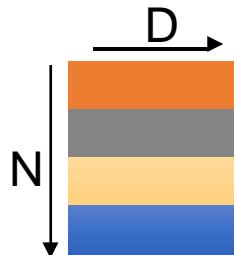
2D array representation

Permutation invariance

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



represents the same **set** as



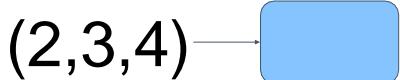
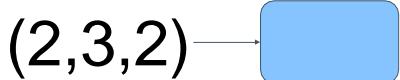
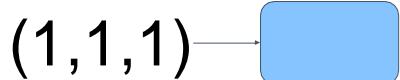
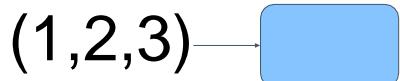
2D array representation

Construct a Symmetric Function

Observe:

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

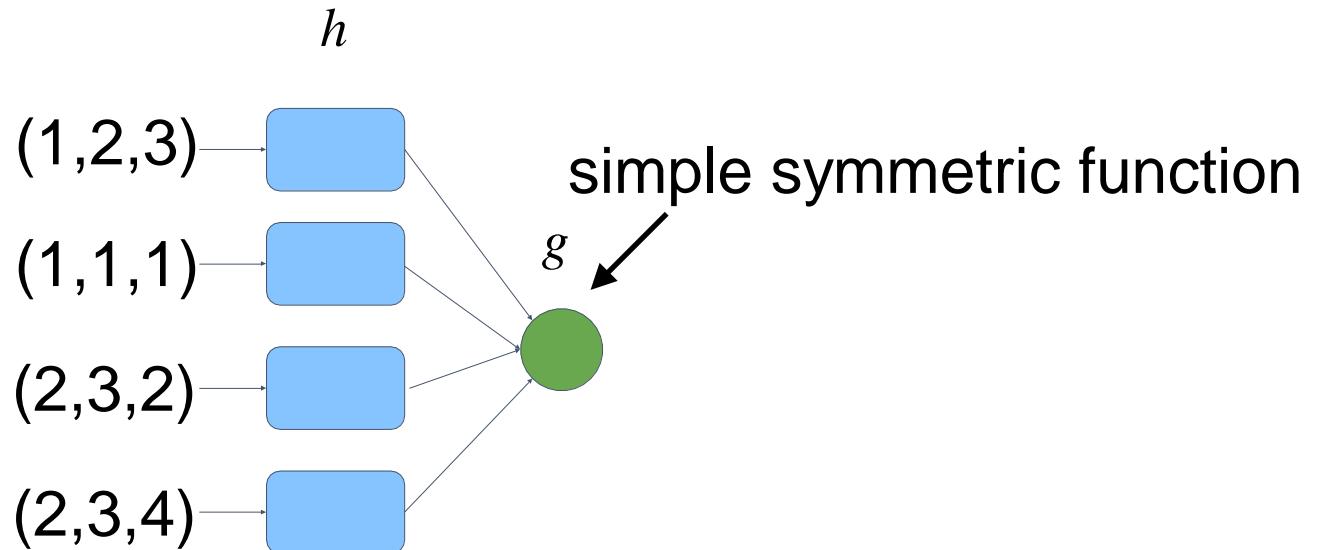
h



Construct a Symmetric Function

Observe:

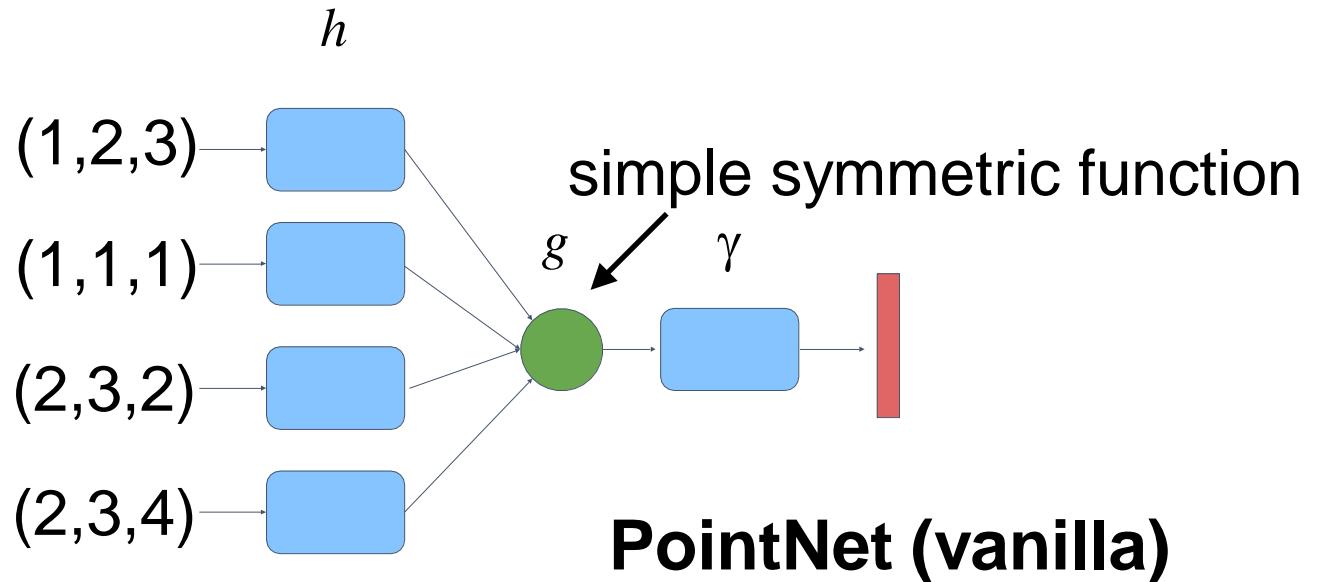
$f(x_1, x_2, \dots, x_n) = \gamma \circ g(h(x_1), \dots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

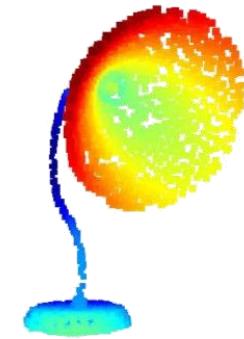
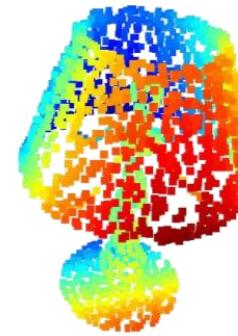
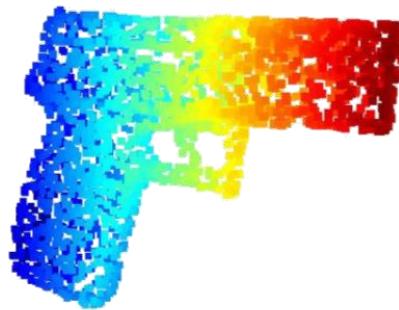
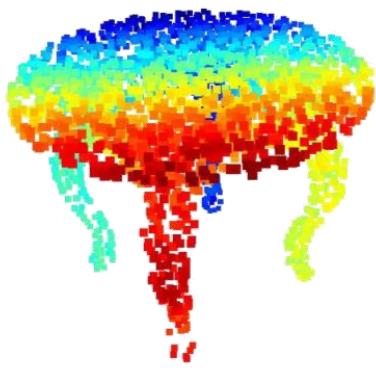
Observe:

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

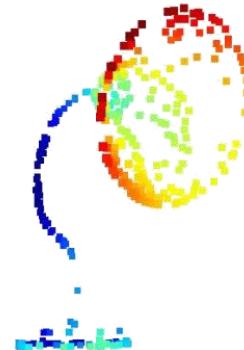
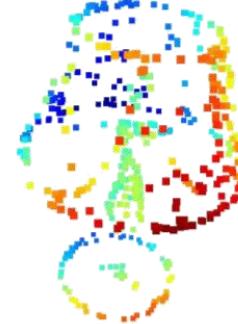
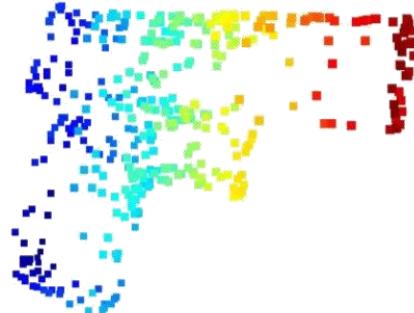
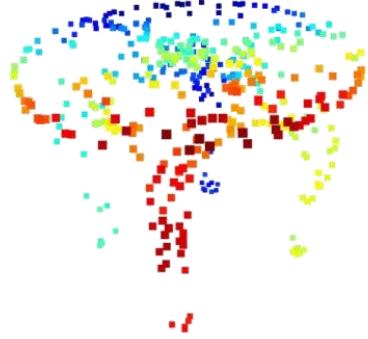


Visualize What is Learned by Reconstruction

Original Shape



Critical Point Sets



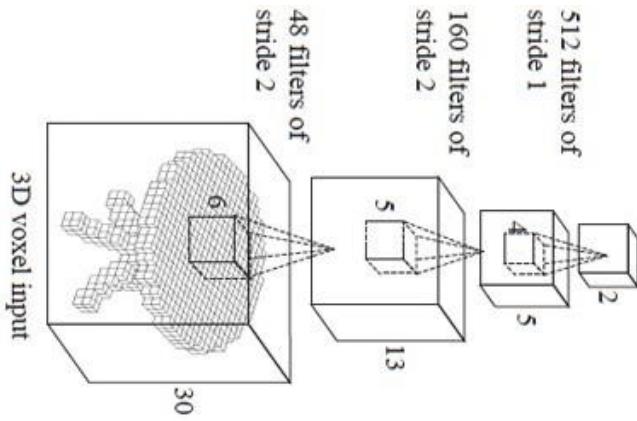
Salient points are discovered!

Limitations of PointNet

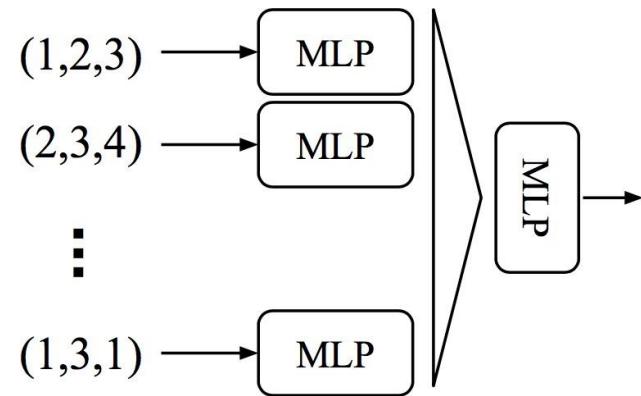
Hierarchical feature learning
Multiple levels of abstraction

v.s.

Global feature learning
Either one point or all points



3D CNN (Wu et al.)



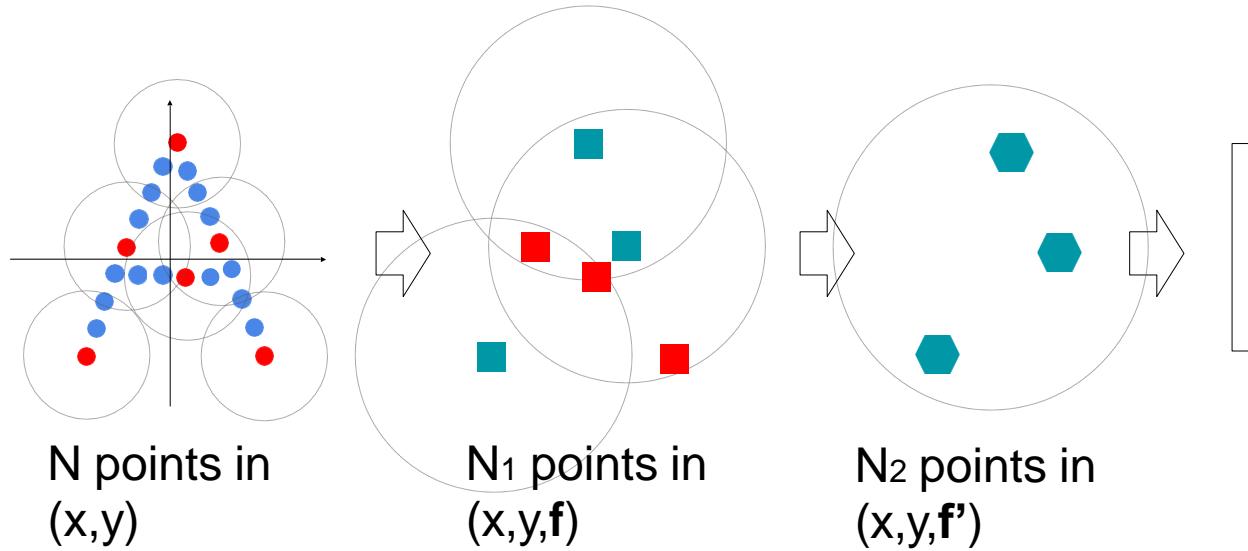
PointNet (vanilla) (Qi et al.)

- No local context for each point!
- Global feature depends on absolute coordinate. Hard to generalize to unseen scene configurations!

Points in Metric Space

- Learn “kernels” in 3D space and conduct convolution
- Kernels have compact spatial support
- For convolution, we need to find neighboring points
- Possible strategies for range query
 - Ball query (results in more stable features)
 - k-NN query (faster)

PointNet v2.0: Multi-Scale PointNet

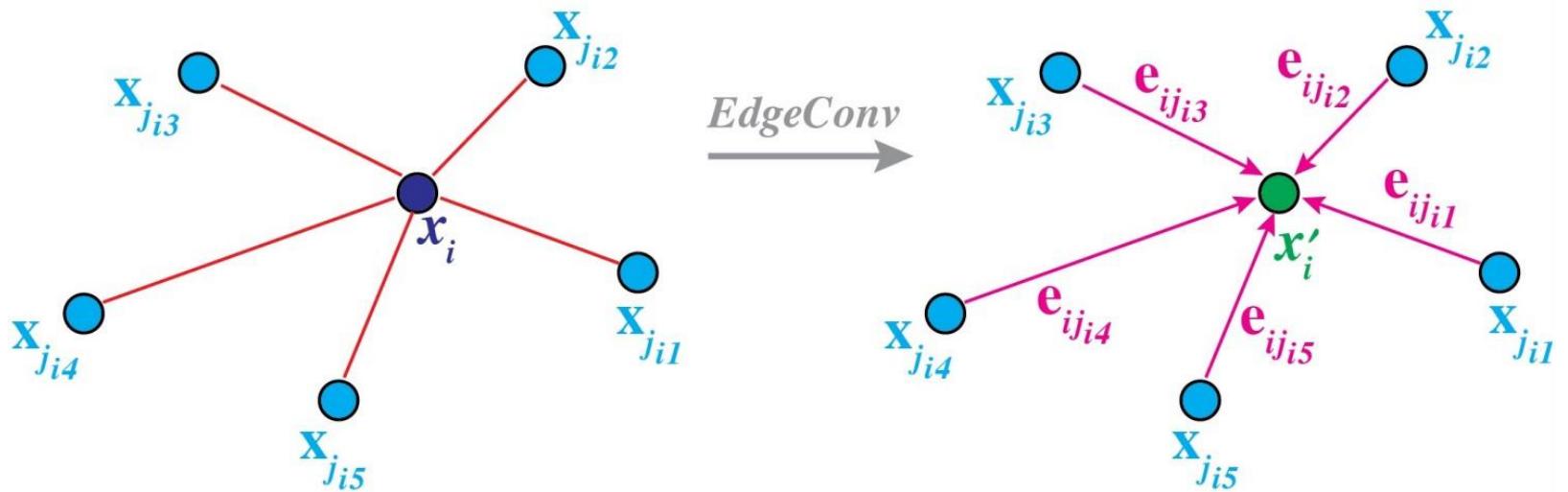


Repeat

- Sample anchor points
- Find neighborhood of anchor points
- Apply PointNet in each neighborhood to mimic convolution

Point Convolution As Graph Convolution

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



Wang et al., “Dynamic Graph CNN for Learning on Point Clouds”,
Transactions on Graphics, 2019

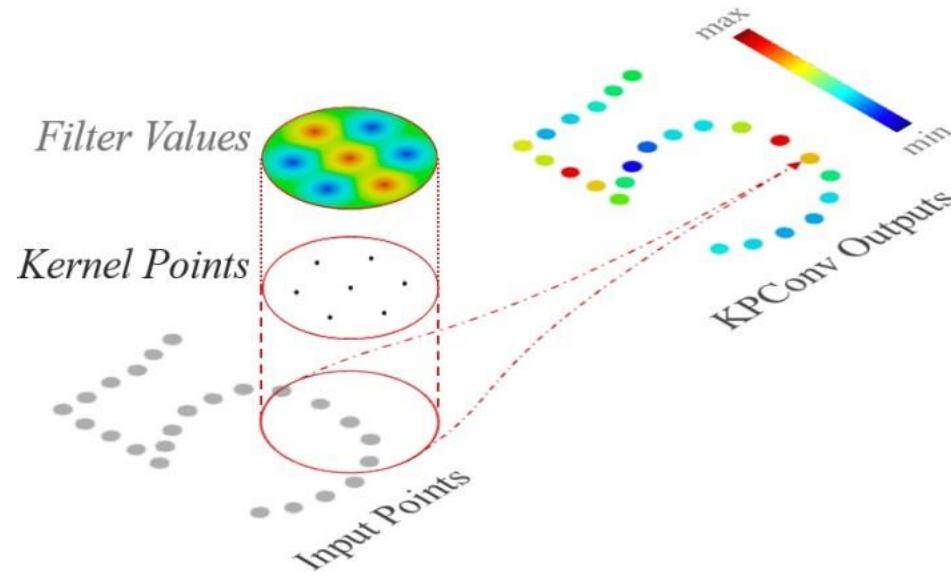
Liu et al., “Relation-Shape Convolutional Neural Network for Point
Cloud Analysis”, CVPR 2019

Standard GCNs are not Geometry-Aware

- Note that points are **sampled** from surfaces
- Ideally, features describe the geometry of underlying surface. Should be sample invariant
- But GCNs lack design to address sample invariance
- Remind us “density estimation” from a population
- Rescue: Estimate the continuous kernel and point density for continuous convolution

Mathematically Proper Conv. Discretization

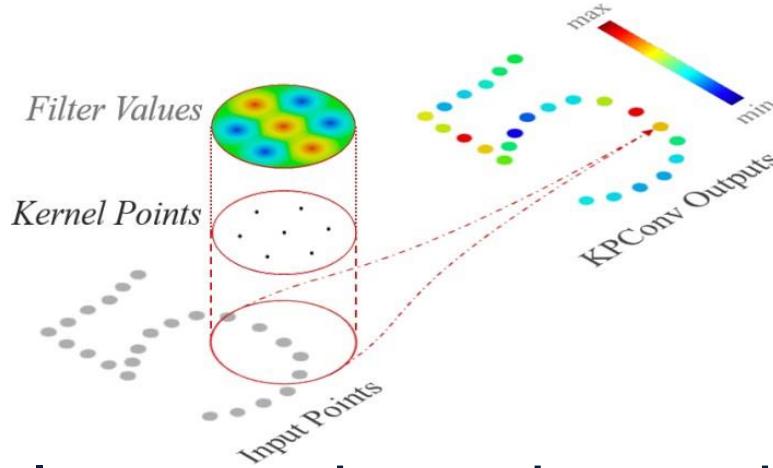
- Continuous conv: $(\mathcal{F} * g)(x) = \int g(y - x)f(y)dy$
- Empirical conv: $(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x)f_i$



Interpolated Kernel for Convolution

- Continuous conv: $(\mathcal{F} * g)(x) = \int g(y - x)f(y)dy$

- Empirical conv: $(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x)f_i$



- Learn kernel value at anchor points and interpolate to build continuous kernel

$$\kappa_{jm}(z) = \sum_l k_{ljm} \Phi(|z - y_l|)$$

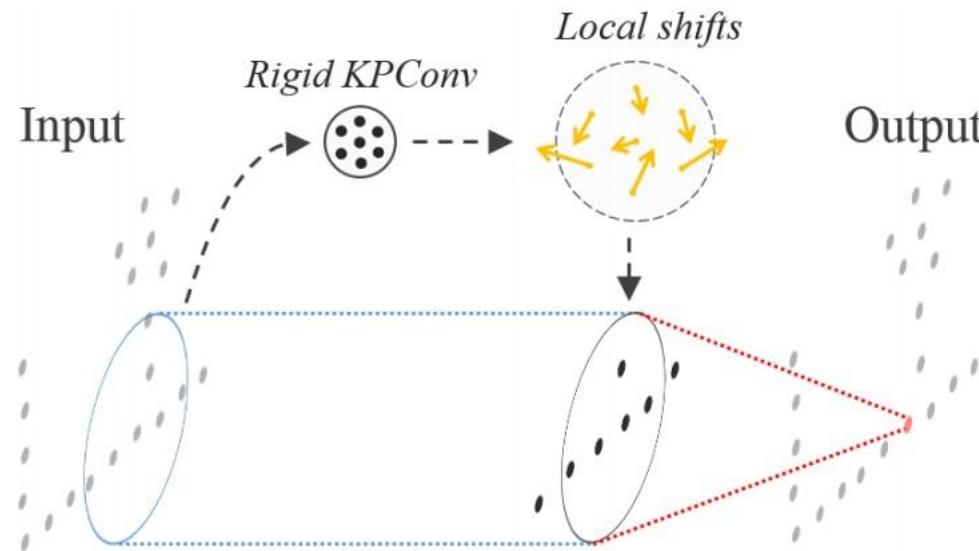
\emptyset RBF kernel

Atzmon et al., “Point Convolutional Neural Networks by Extension Operators”, *Trans. on Graphics*, 2018

Thomas et al., “KPConv: Flexible and Deformable Convolution for Point Clouds”, *ICCV* 2019

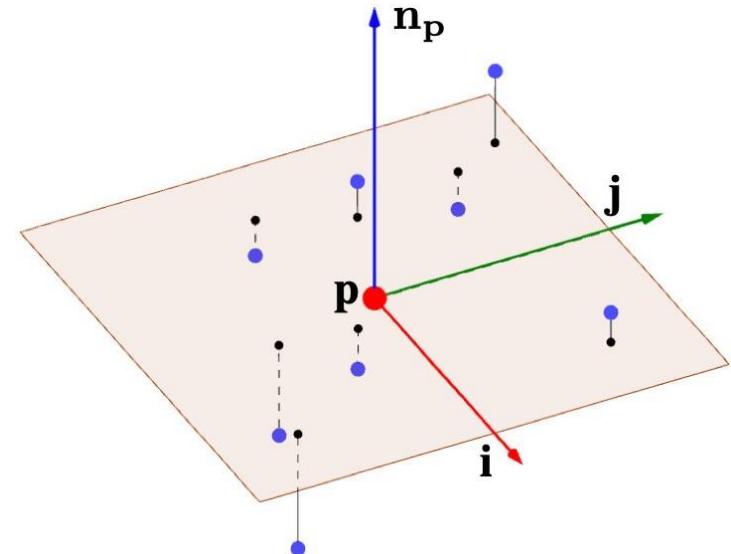
Deformable Kernel for Deformable Objects

- Deformable point-based kernel
- The 3D version of 2D deformable convolution

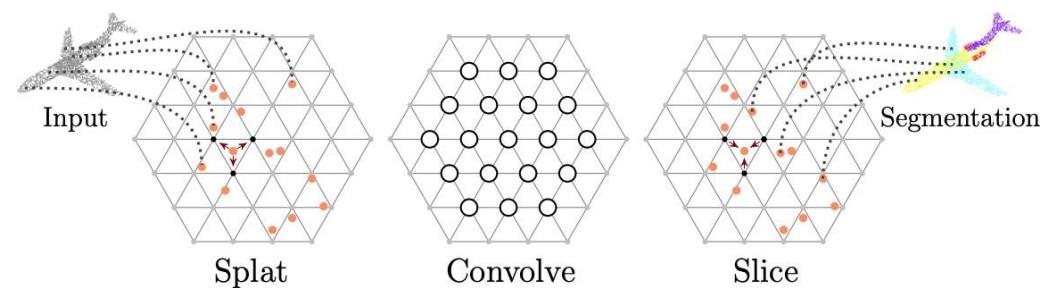


Other Ways to Address 3D Conv

- Tangent convolution:
 - Project & interpolate local points to the tangent plane (PCA for orientation)
- Lattice
 - high-dimensional space



Tatarchenko, Maxim, et al. "Tangent convolutions for dense prediction in 3d", CVPR 2018.

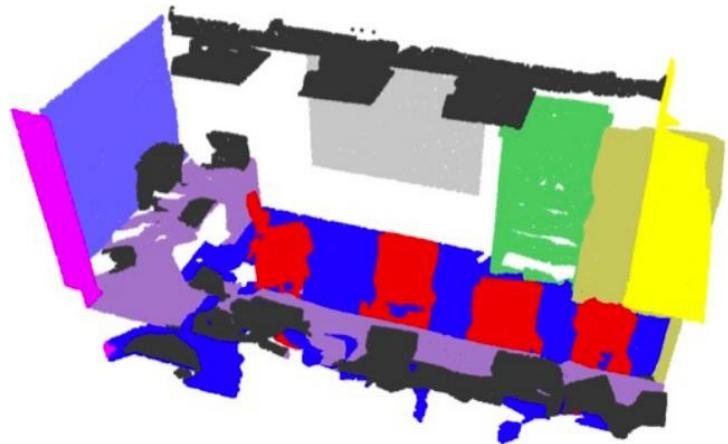


Su, Hang, et al. "Splatnet: Sparse lattice networks for point cloud processing", CVPR 2018

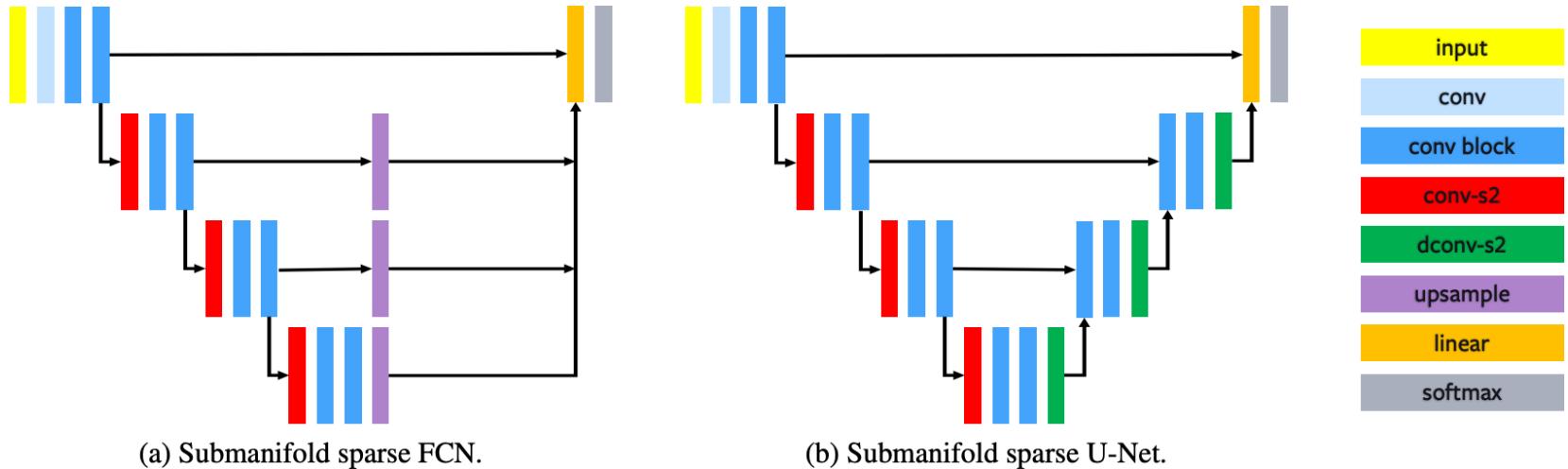
Topics

- 3D Data
- Classification
- Segmentation and Detection
- Reconstruction

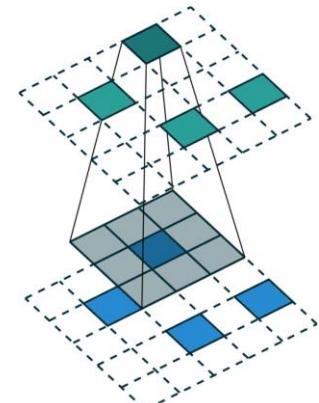
Task: 3D Semantic Segmentation



Encoder-Decoder



Sparse Convolution

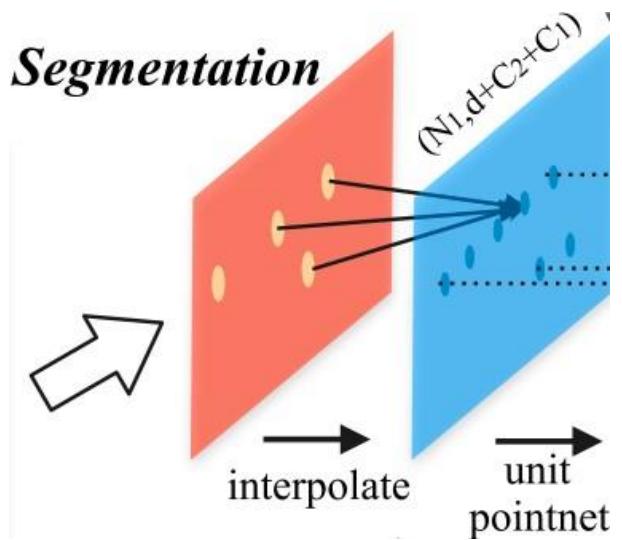


Graham, Benjamin, Martin Engelcke, and Laurens van der Maaten. "3d semantic segmentation with submanifold sparse convolutional networks." CVPR 2018.

Choy, Christopher, et al. "4D Spatio-Temporal Convnets: Minkowski Convolutional Neural Networks." CVPR 2019

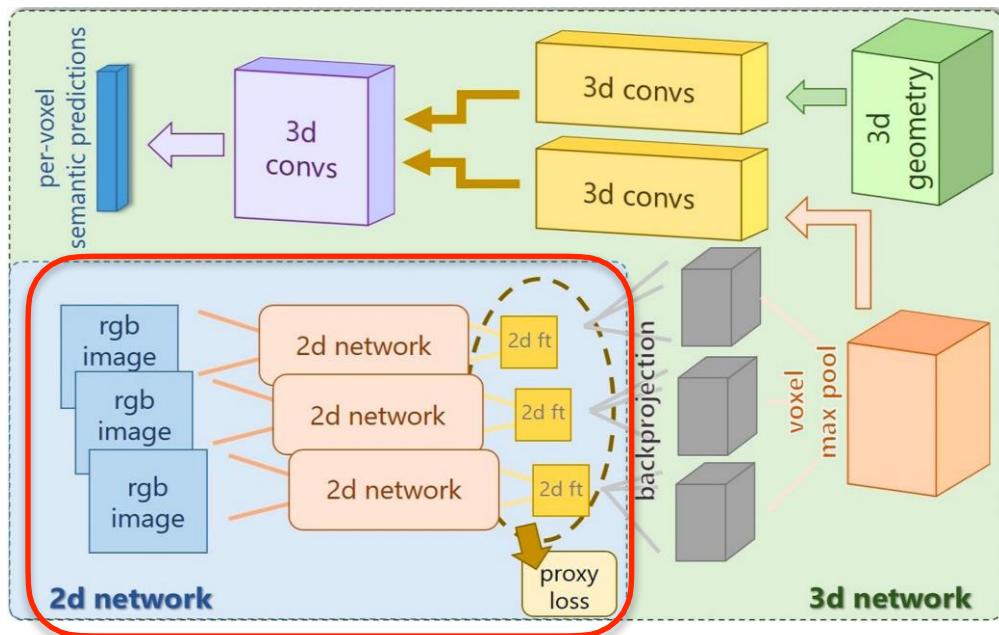
Sparse Deconvolution

Encoder-Decoder



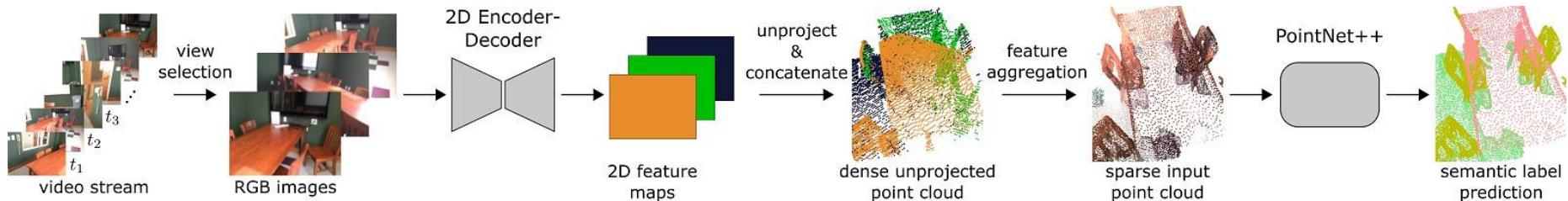
Upsample point cloud features by interpolating from 3 nearest points in the point cloud of lower resolution.

Multimodal



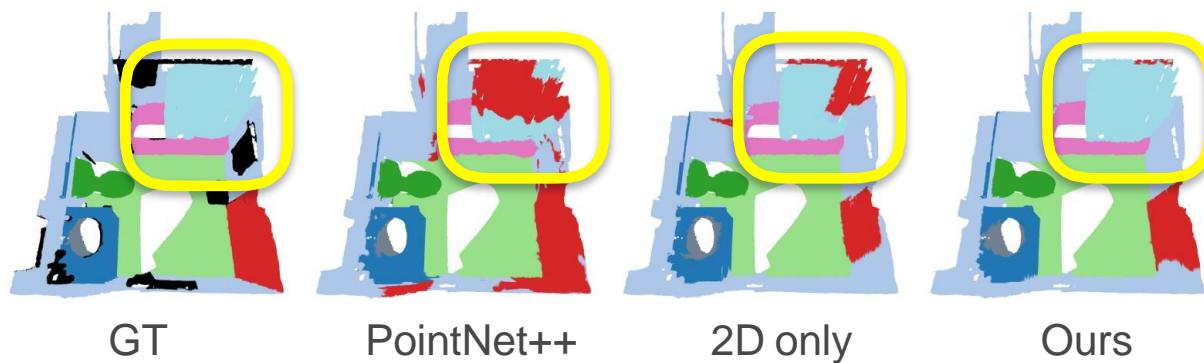
- Backproject 2D features to 3D voxels
- Apply voxel-wise max-pooling across multiple views
- Fuse 2D and 3D features at the intermediate level

Multimodal

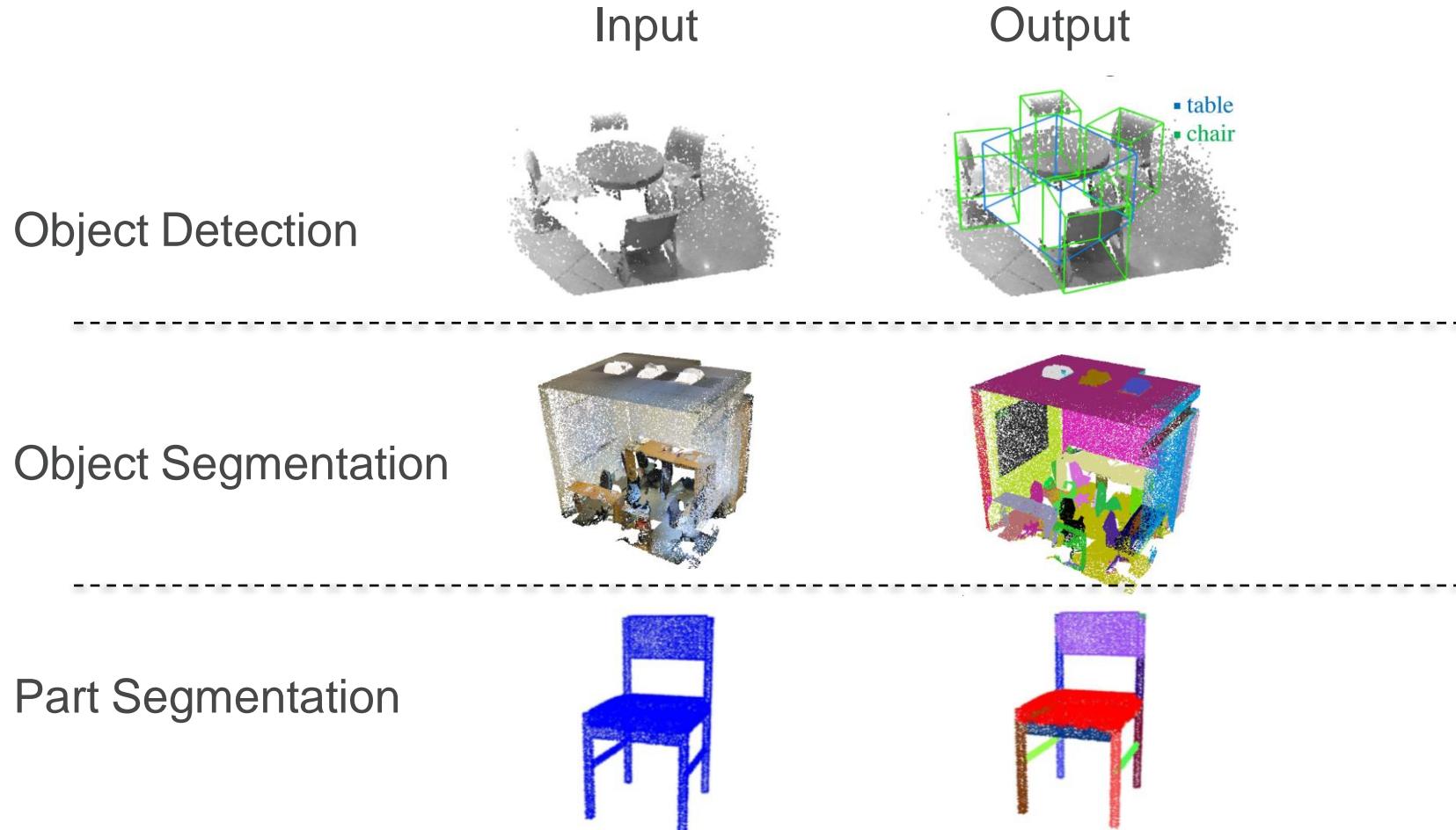


- Backproject 2D features to 3D points
- Apply PointNet to aggregate multi-view features
- Fuse 2D and 3D features as input to 3D network

ignore wall floor cabinet bed chair sofa table door window bookshelf picture
counter desk curtain refrigerator shower curtain toilet sink bathtub otherfurniture



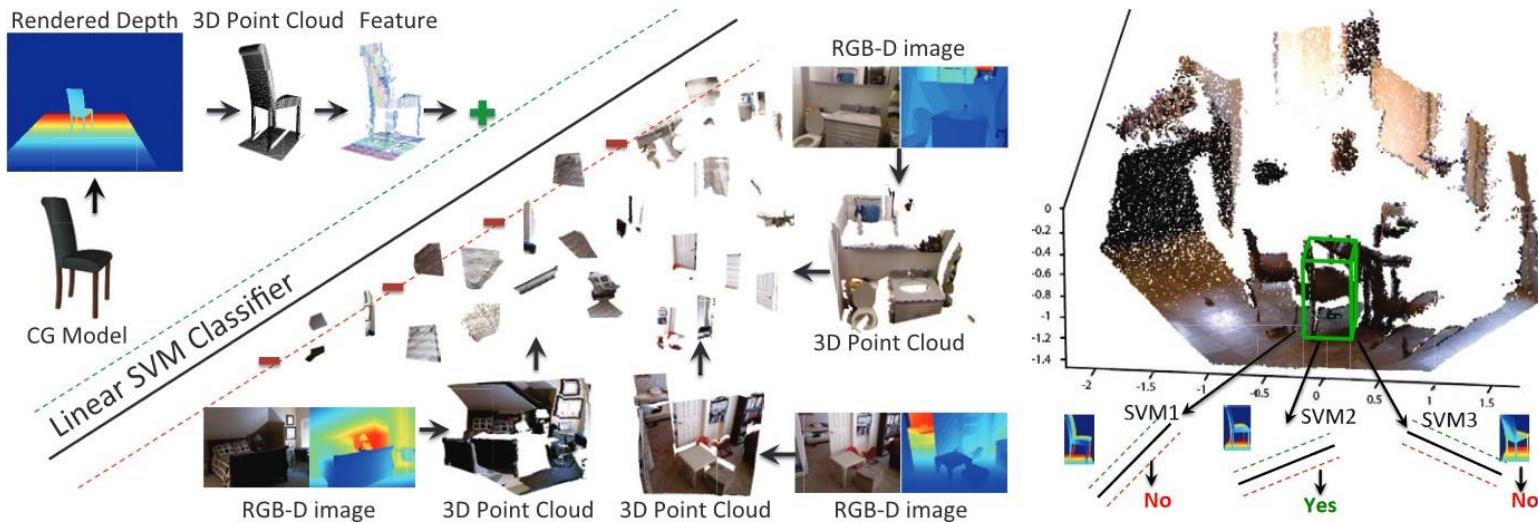
Task: Instance-level Understanding



Top-down Methods

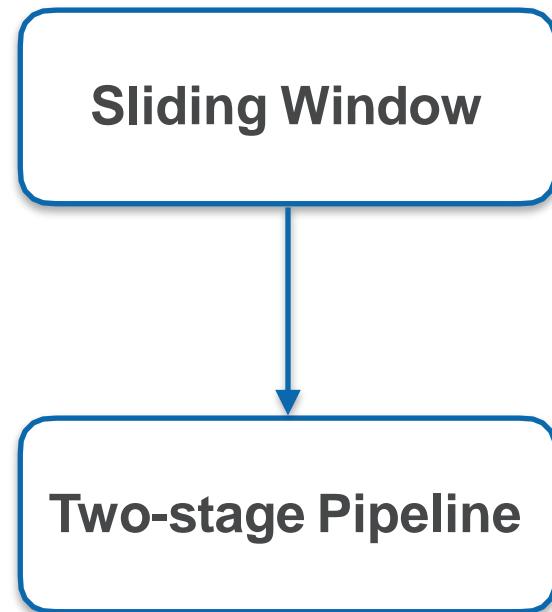
- Does this point cloud contain an object?
- How to select point clouds to classify objectness?

Sliding Shapes



Sliding window to walk over
the entire space

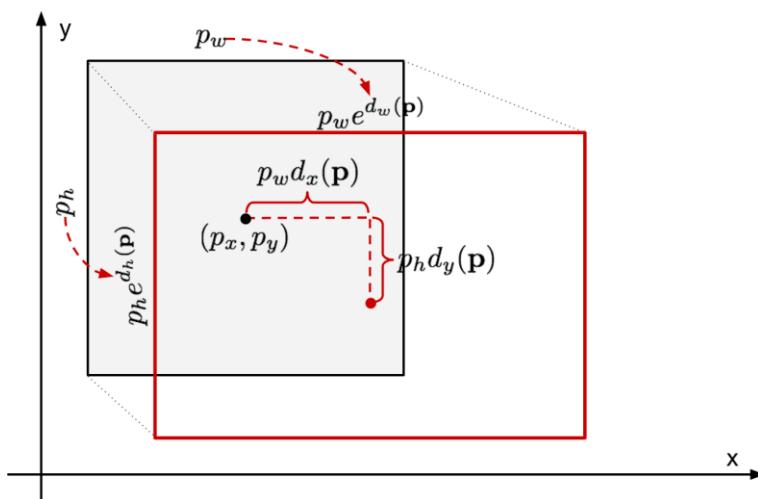
Expensive !



First stage: Proposal
Second stage: Refinement

Localization

- 3D bounding box regression
- Predefined bounding boxes (anchors) at each sliding window
- Re-parameterize as relative offsets



$$\Delta_x = \frac{\mu - \mu_{anchor}}{\phi_{anchor}}$$

Center shift

$$\Delta_w = \ln\left(\frac{\phi}{\phi_{anchor}}\right)$$

Size scale

From Box to Instance Segmentation

- Box includes background points or other instances
- foreground/background segmentation

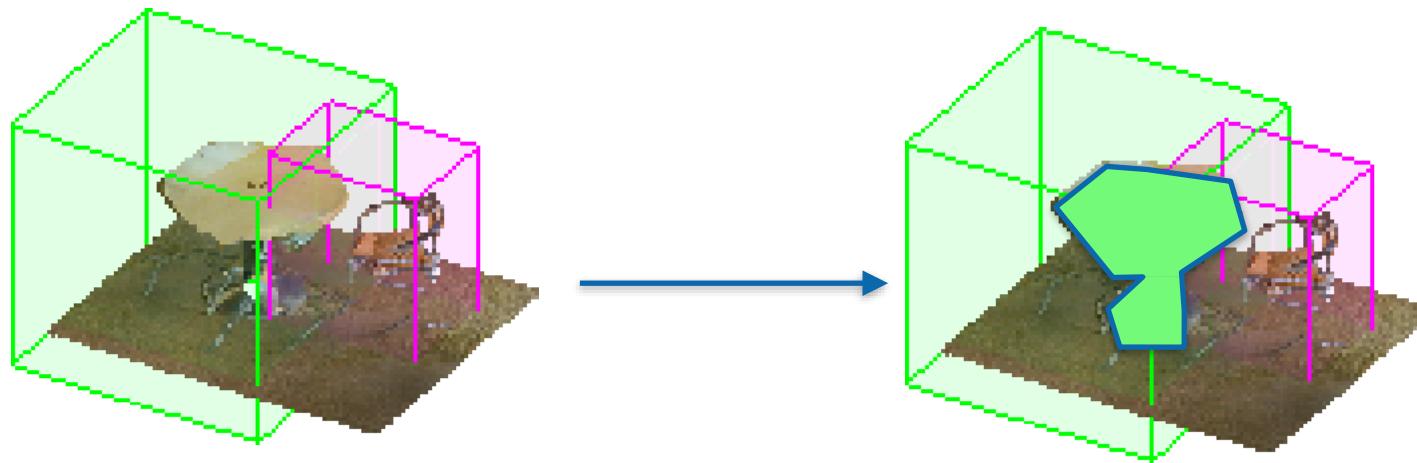
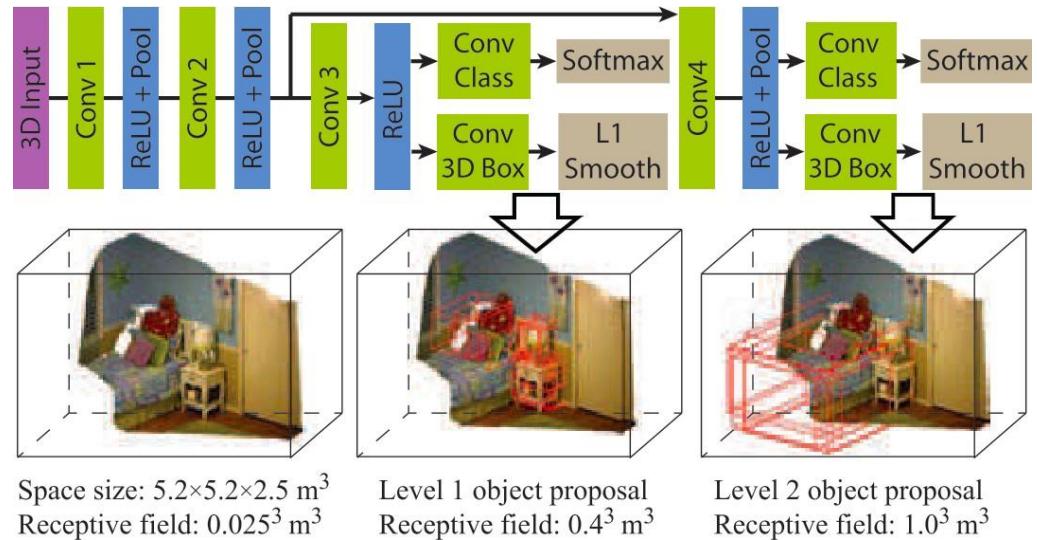


Figure from “Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds”

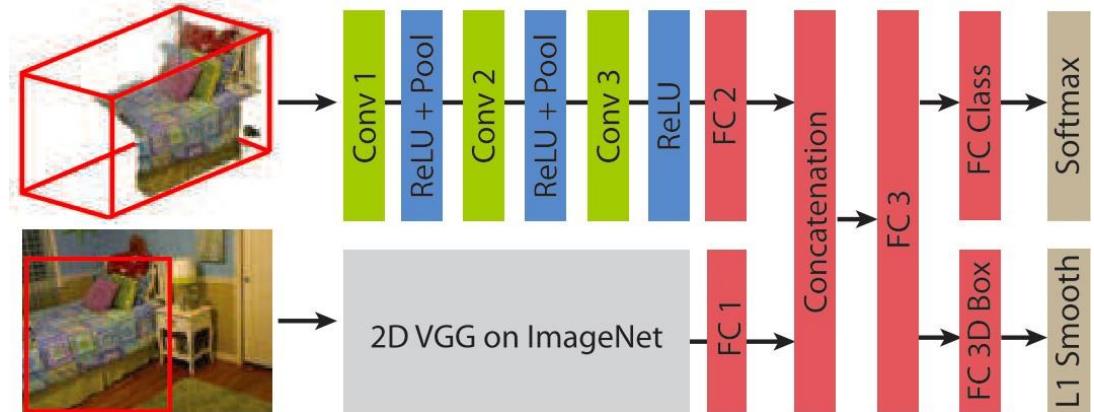
Volumetric R-CNN

Stage1: 3D Region Proposal Network

TSDF



Stage2: Joint Object Recognition Network



Topics

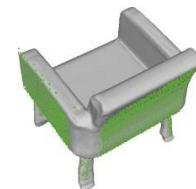
- 3D Data
- Classification
- Segmentation and Detection
- Reconstruction
 - Generative Model
 - Multi-View Stereo

Task

Conditional generation

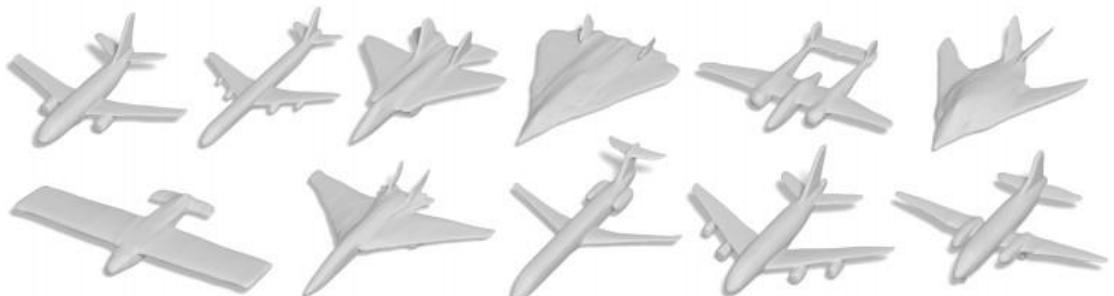
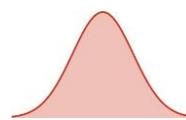


Single-image
3D reconstruction



Shape Completion

Free generation



Gaussian Noise

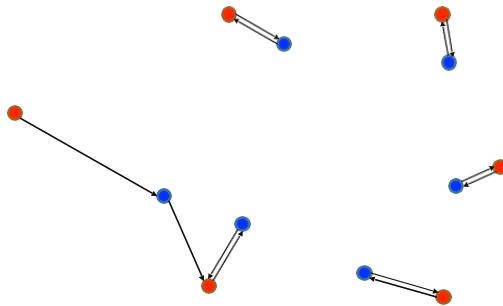
Metric

First of all,

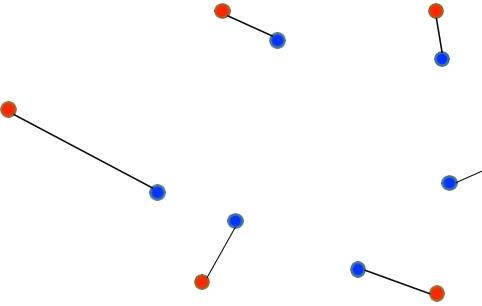
how to evaluate the generated shapes?

Metric for Point Clouds

Chamfer Distance



Earth Mover's Distance



$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

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

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

F-score

precision and recall are calculated by checking the percentage of points in one point cloud that can find a neighbor from the other point cloud within a threshold. F-score is then calculated as the harmonic mean.

Normal Consistency

dot product of the normals of each point and its nearest neighbor

Fan et al., "A Point Set Generation Network for 3D Object Reconstruction from a Single Image", CVPR 2017

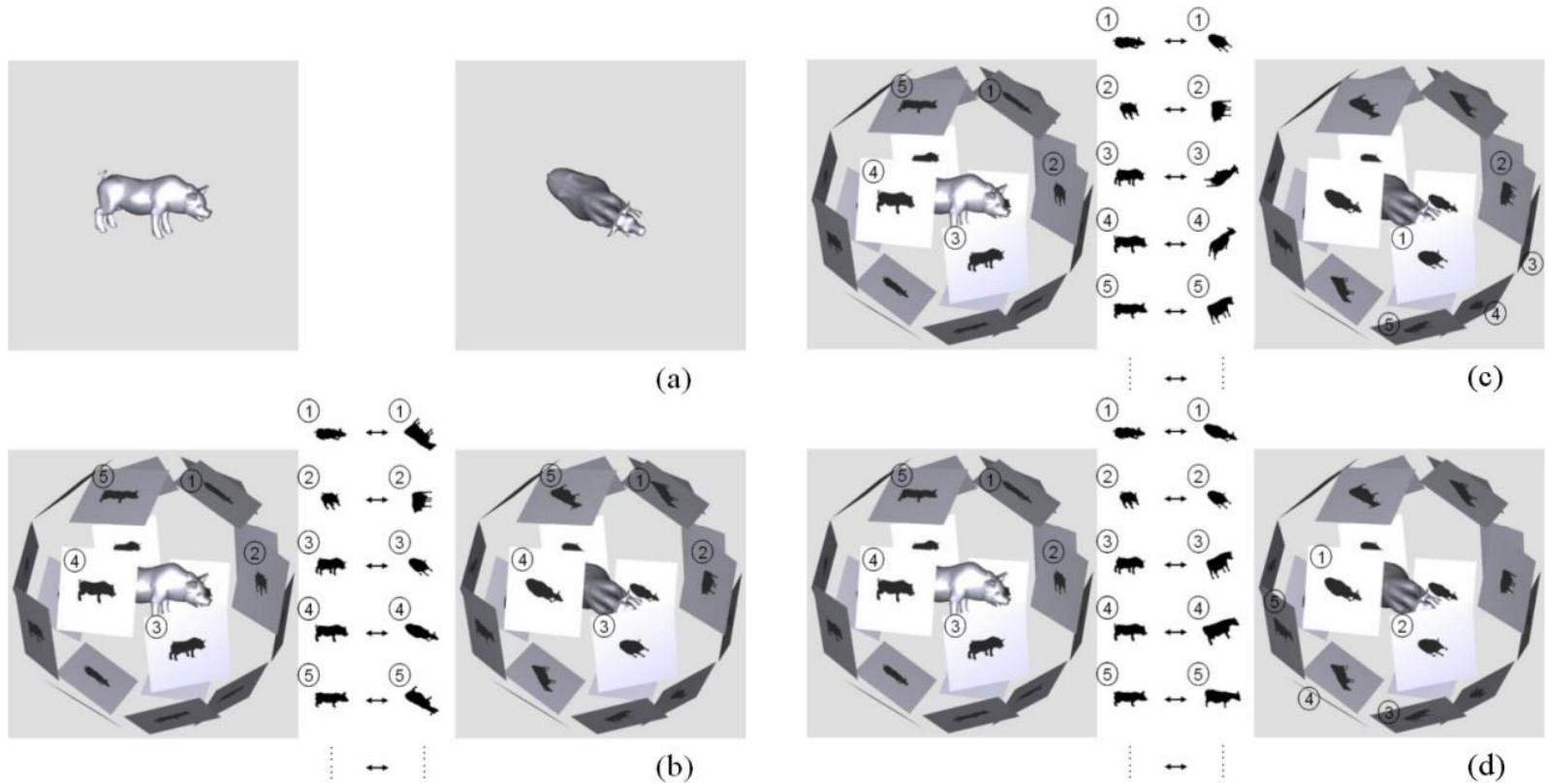
Wang et al., "Pixel2mesh: Generating 3d mesh models from single rgb images", ECCV 2018

Mescheder et al., "Occupancy Networks: Learning 3D Reconstruction in Function Space", CVPR 2019

Metric by Projection

Light Field Descriptor (LFD)

- Extract features from orthogonal projections

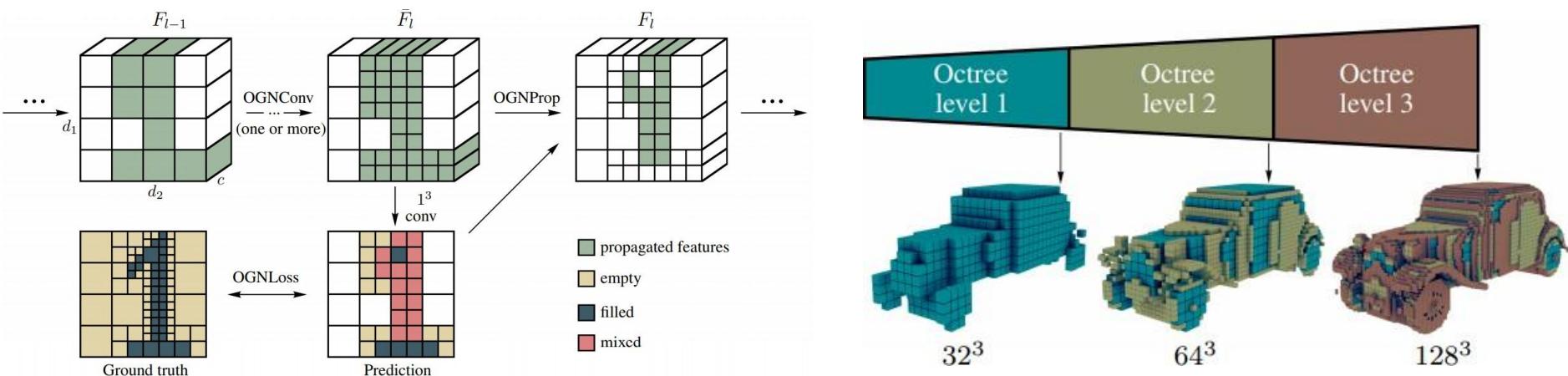


Algorithm for Conditional Generation

From Single Image to Volume

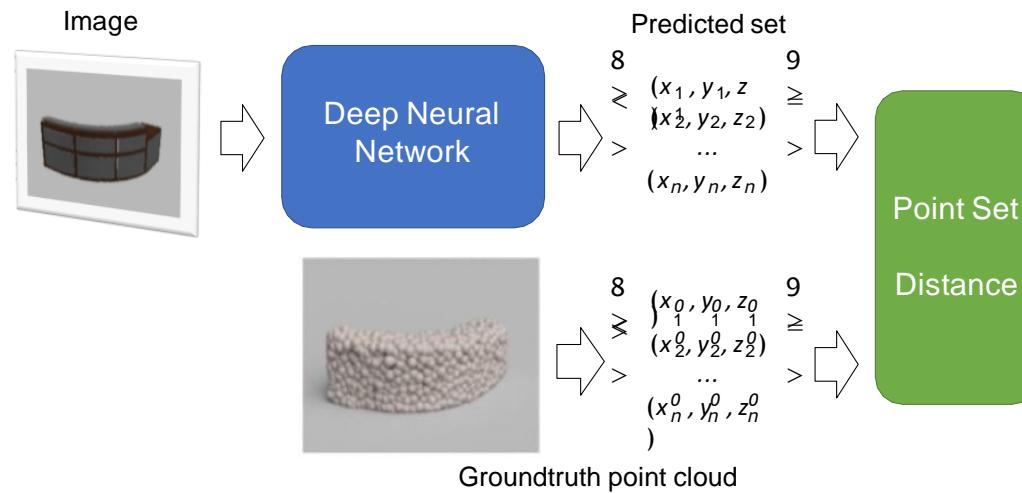
Avoid $\mathcal{O}(n^3)$ reconstruction

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



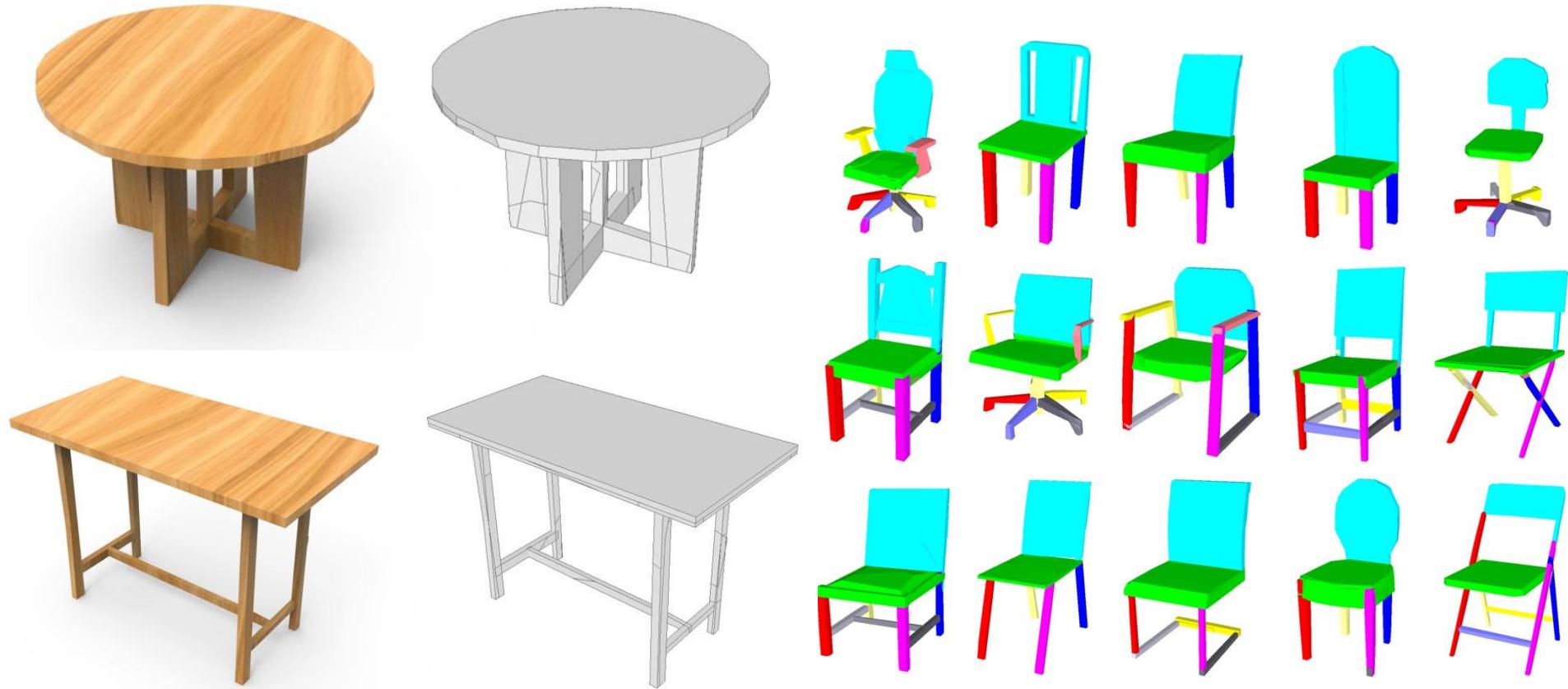
From Single Image to Point Cloud

- It is possible to generate a **set** (permutation invariant)



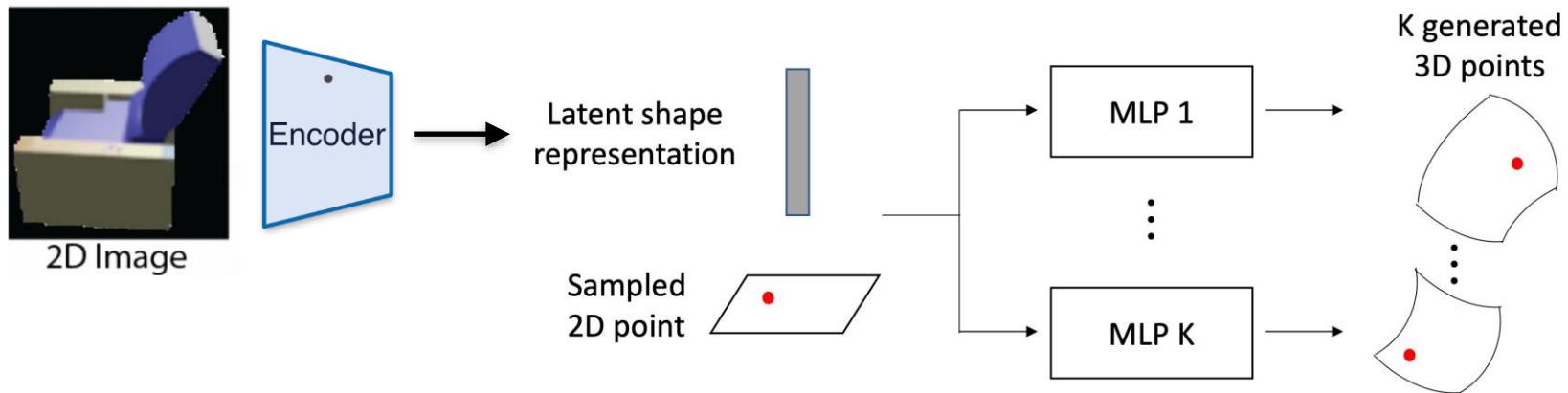
From Image to Shape

- Planes -> convex polytopes -> shape



From Image to Surface

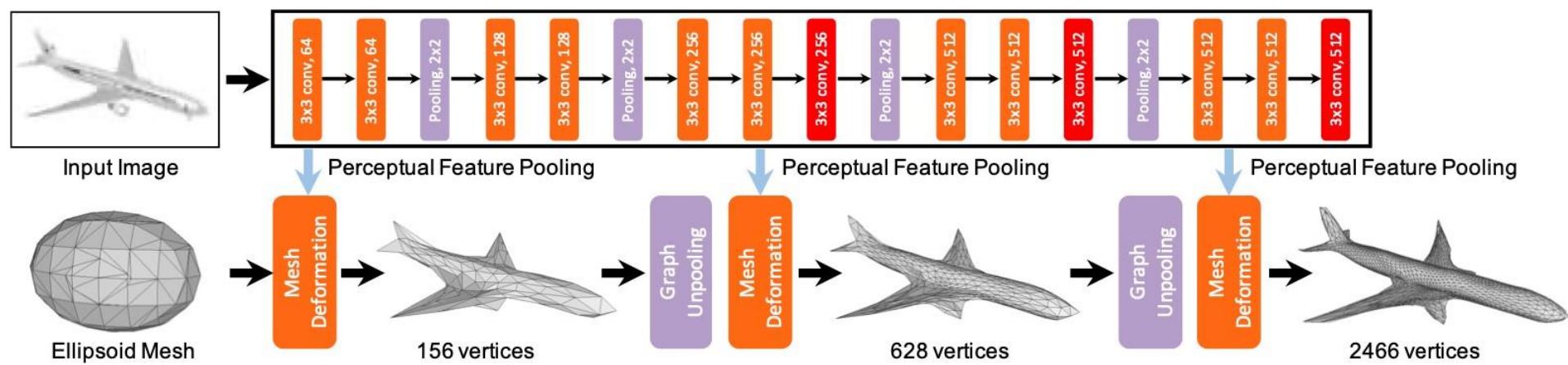
- Learn to warp a plane to surface



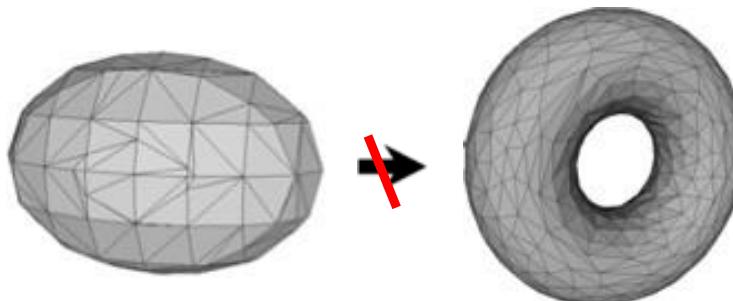
Groueix et al., "AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation", CVPR 2018
Yang et al., "Foldingnet: Point cloud auto-encoder via deep grid deformation", CVPR 2018

From Image to Surface

- Polygon meshes are irregular
- Learn to deform a template mesh



cannot change the topology of the template mesh

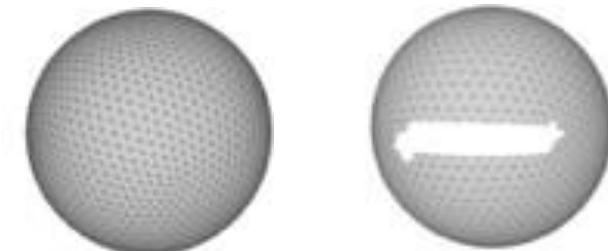


From Image to Surface

part-level deformation

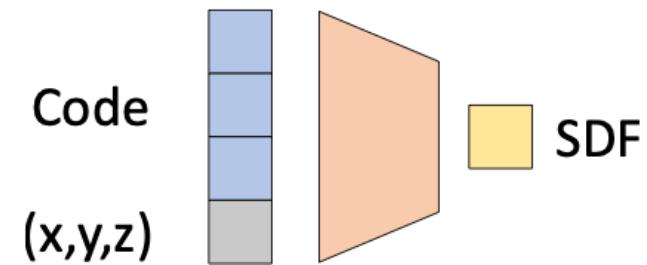
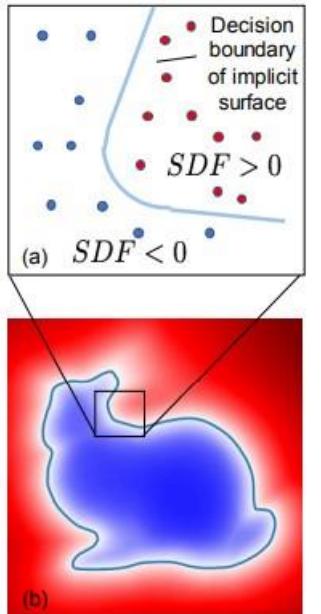


modify the topology of
the template mesh



Implicit Surface Reconstruction

- Implicit field function $F(x)$ (e.g., signed distance)
- Extract the iso-surface $F(x) = 0$



Park et al., “DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation”, CVPR 2019

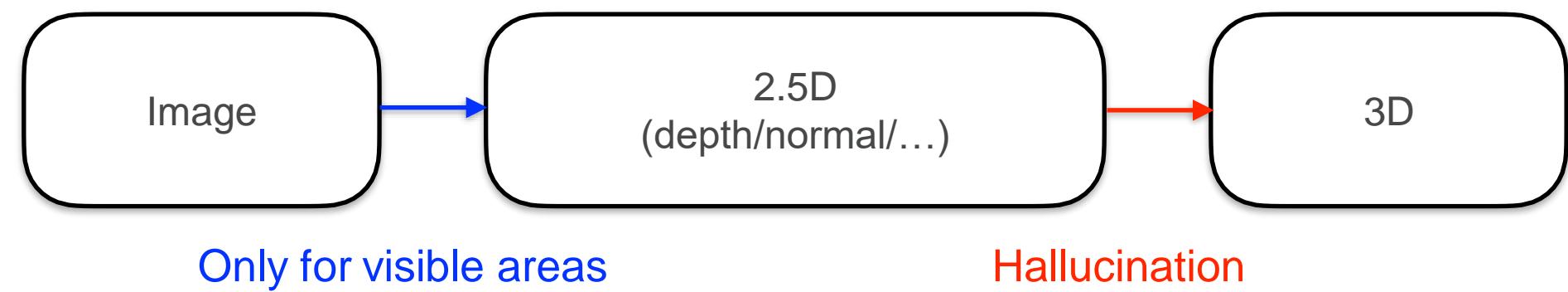
Other two similar paper on implicit representation:

Mescheder et al., “Occupancy Networks: Learning 3D Reconstruction in Function Space”, CVPR 2019

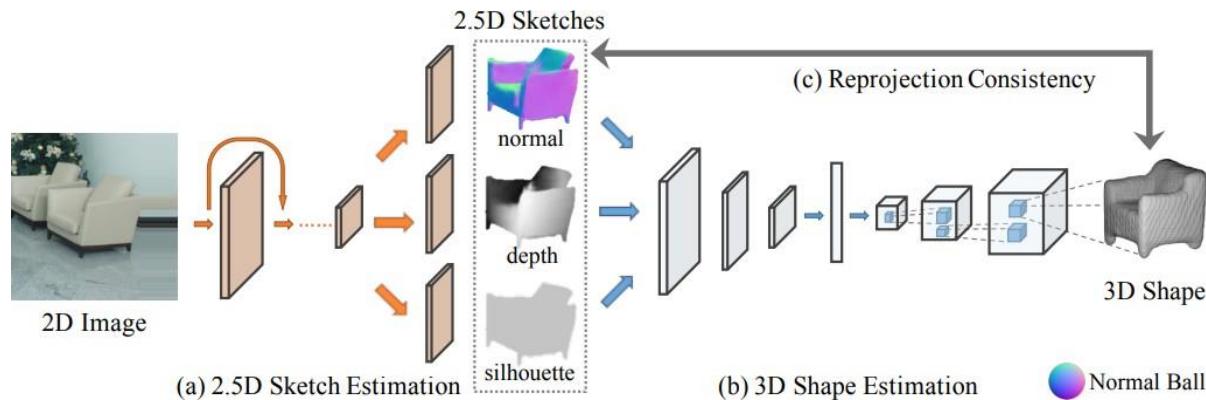
Chen et al., “Learning Implicit Fields for Generative Shape Modeling”, CVPR 2019

- In general,
 - First map the input to a shape embedding
 - Then reconstruct by decoding
- Limitation
 - Output is not explicitly grounded on the input
 - Structures of 3D objects are not explicitly leveraged
 - Cannot generalize to unseen objects

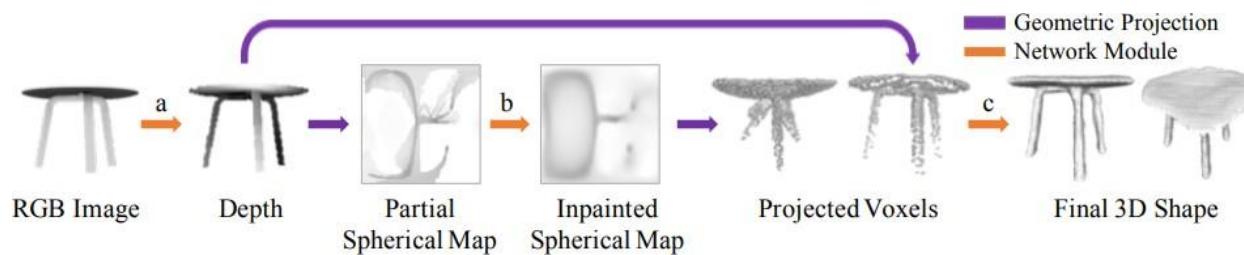
Visually Grounded Prediction: 2.5D to Bridge



Visually Grounded Prediction: 2.5D to Bridge



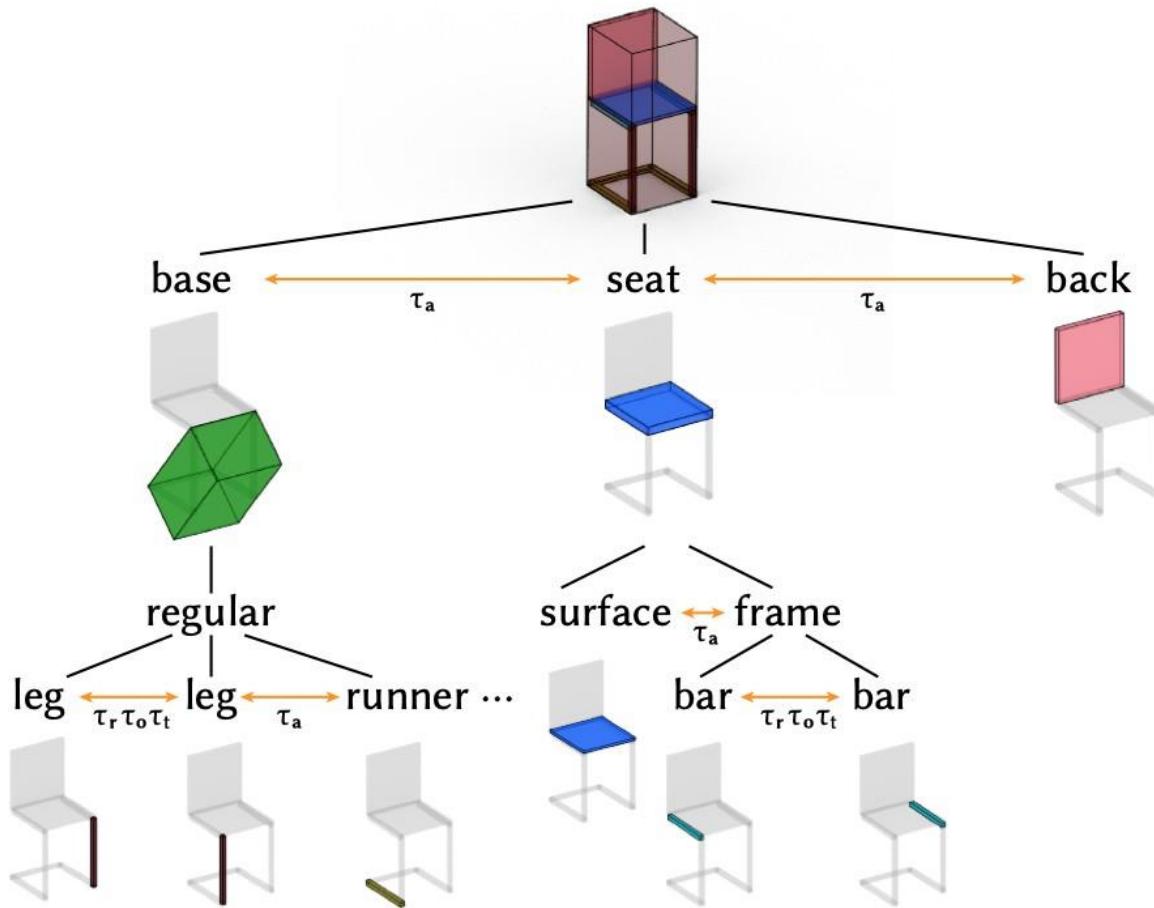
Wu et al., “**MarrNet: 3D Shape Reconstruction via 2.5D Sketches**”, NeurIPS 2017



Zhang et al., “**Learning to Reconstruct Shapes from Unseen Classes**”, NeurIPS 2018

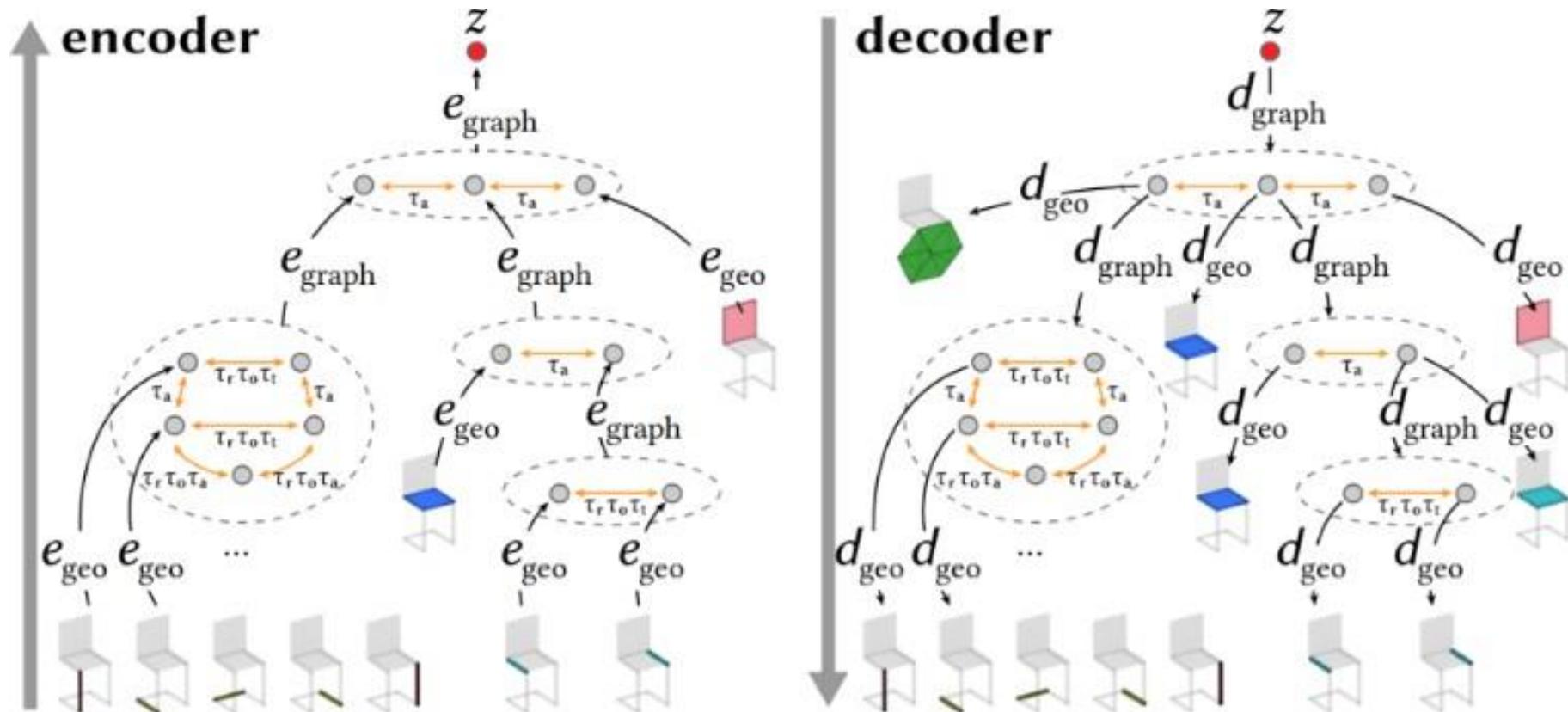
Structured Prediction: Part-based

Hierarchical Graph



Structured Prediction: Part-based

Recursive Network for Hierarchical Graph AE



Structured Prediction: Part-based



Mo et al., “**StructureNet**, a hierarchical graph network for learning PartNet shape generation”, *Siggraph Asia 2019*

Algorithm for Free Generation (GAN)

Challenges

Similar challenges as GAN for images:

- Good by human eye v.s. Good by objective metric

Metrics

Comparison between two sets of point clouds A and B:

- A: generated point clouds
- B: a held-out test set

Geometry Quality of Generated Shape

- e.g., minimum matching ($B \rightarrow A$) distance for CD/EMD

Coverage

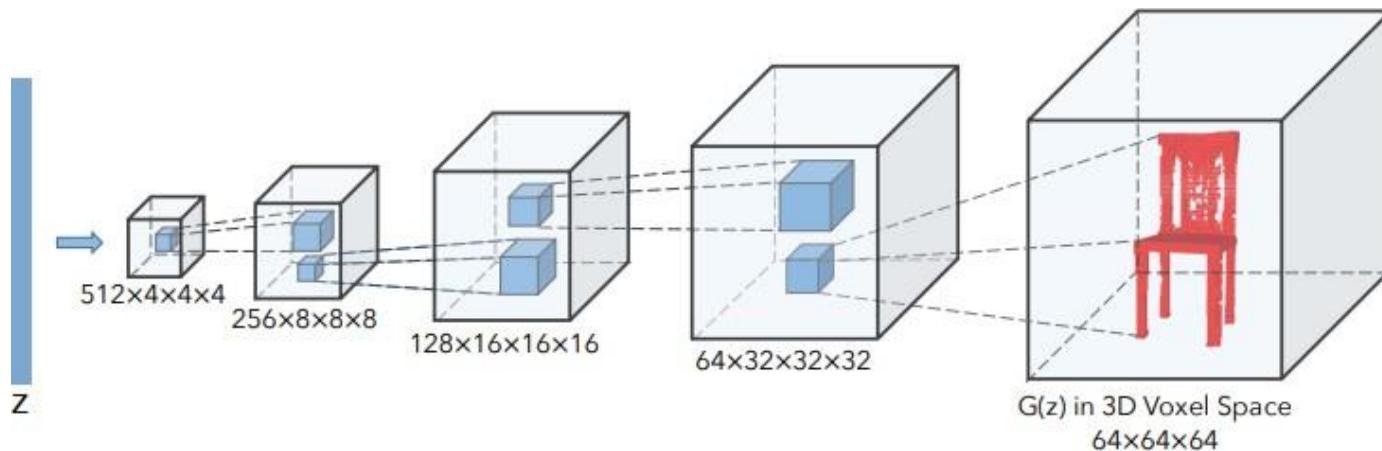
- The fraction of the shapes in B that were matched to shapes in A

Perceptually Correct

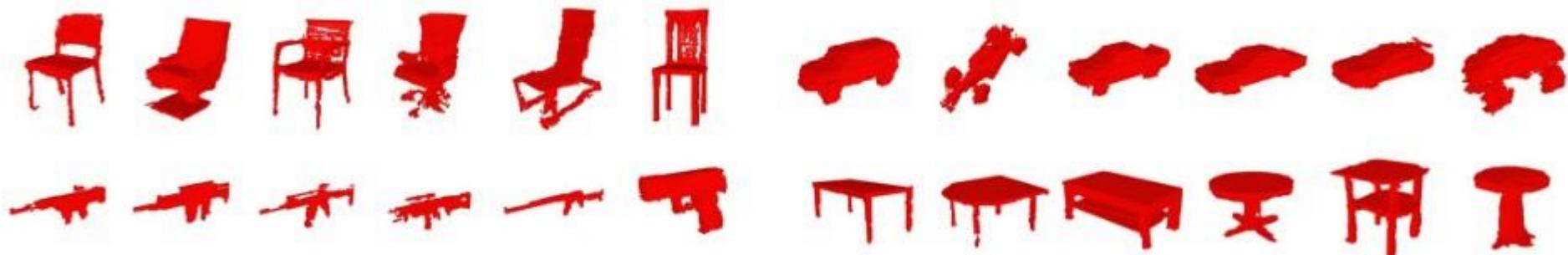
- Feature distribution distance (e.g. Frechet Point Cloud Distance)

$$\text{FPD}(\mathbb{P}, \mathbb{Q}) = \|\mathbf{m}_{\mathbb{P}} - \mathbf{m}_{\mathbb{Q}}\|_2^2 + \text{Tr}(\Sigma_{\mathbb{P}} + \Sigma_{\mathbb{Q}} - 2(\Sigma_{\mathbb{P}} \Sigma_{\mathbb{Q}})^{\frac{1}{2}})$$

Volumetric Generation



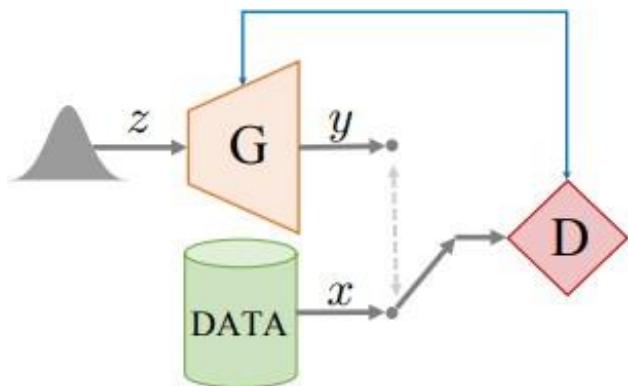
$$\log D(x) + \log(1 - D(G(z)))$$



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

Point Cloud Generation

- FC layer as generator
- PointNet as discriminator
- WGAN



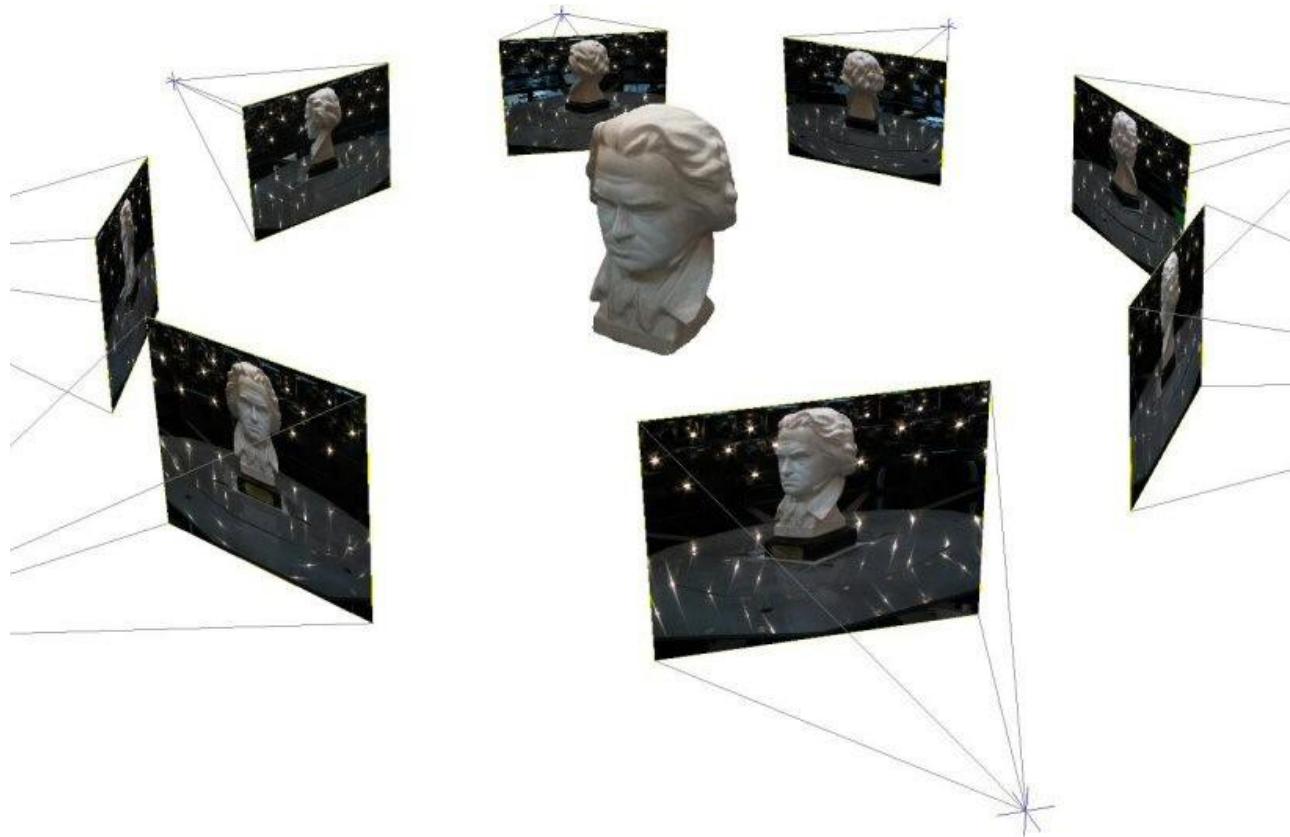
Many Issues

- Still cannot generate high quality local details
- Still hard to generate complex structures
- For point clouds, using strong classifiers (than PointNet) as discriminator is highly tricky

Topics

- 3D Data
- Classification
- Segmentation and Detection
- Reconstruction
 - Generative Model
 - Multi-View Stereo

Task: Reconstruction

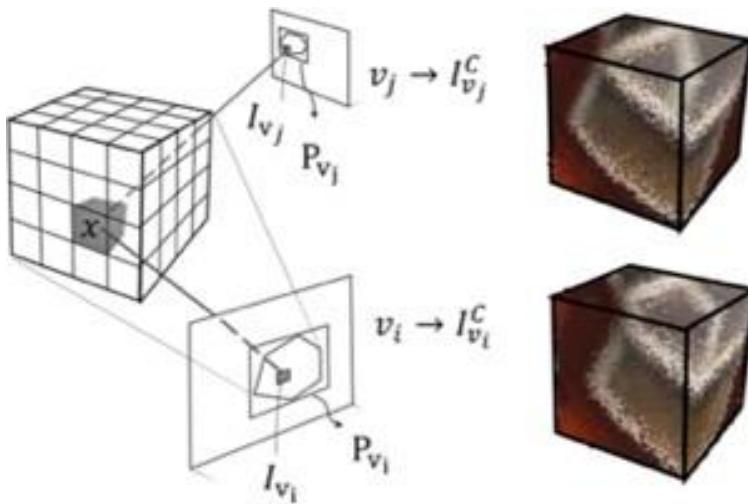


[image: oswald]

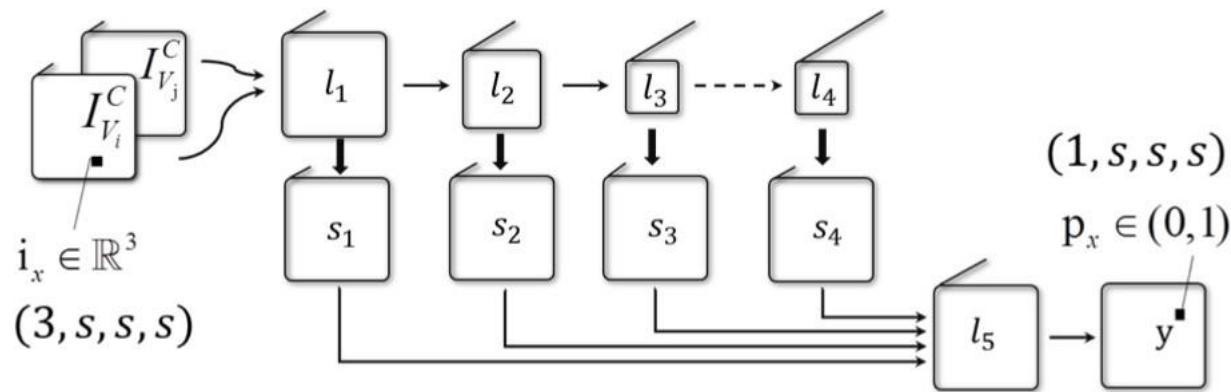
Covered methods: SurfaceNet, LSM, GC-Net, MVSNet, R-MVSNet, PointMVSNet, BA-Net

Surface Reconstruction as Voxel Occupancy Prediction

project along viewing rays to build colored voxel cubes.



Predict the surface confidence for each voxel:



$$\begin{aligned} L(I_{v_i}^C, I_{v_j}^C, \hat{S}^C) = \\ - \sum_{x \in C} \{\alpha \hat{s}_x \log p_x + (1 - \alpha)(1 - \hat{s}_x) \log(1 - p_x)\} \end{aligned}$$

binarization as post-processing

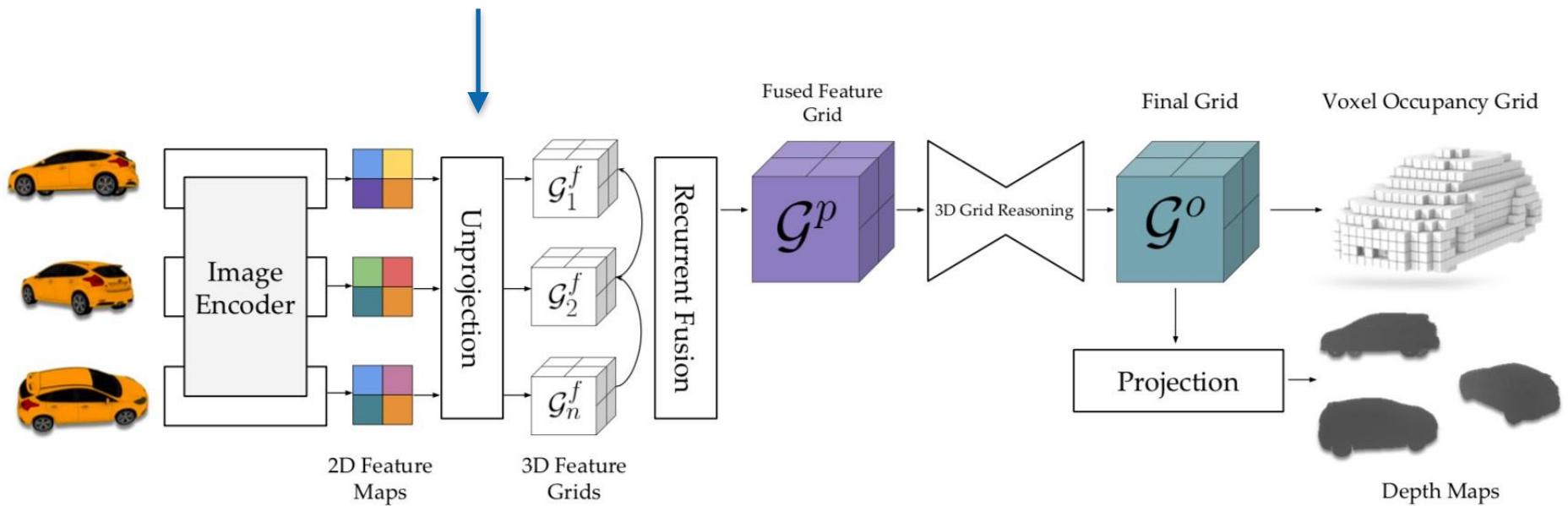
Limitations:

- Pre-computed voxel cubes can only take RGB colors at coarse resolution
- Voxel binarization introduces quantization errors.

Learning-Based Stereopsis

End-to-end learning of deep features for each image pixel

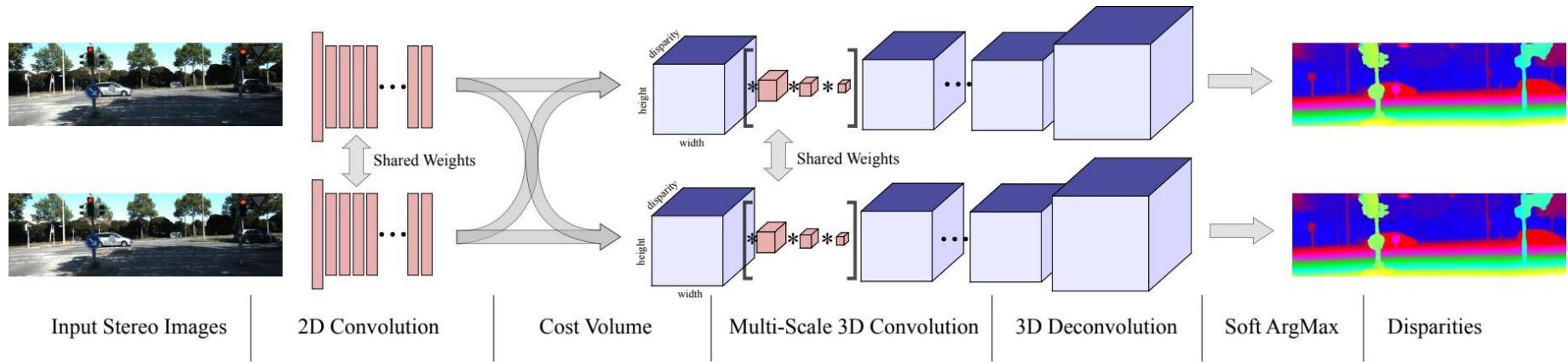
Unproject image features into 3D grids





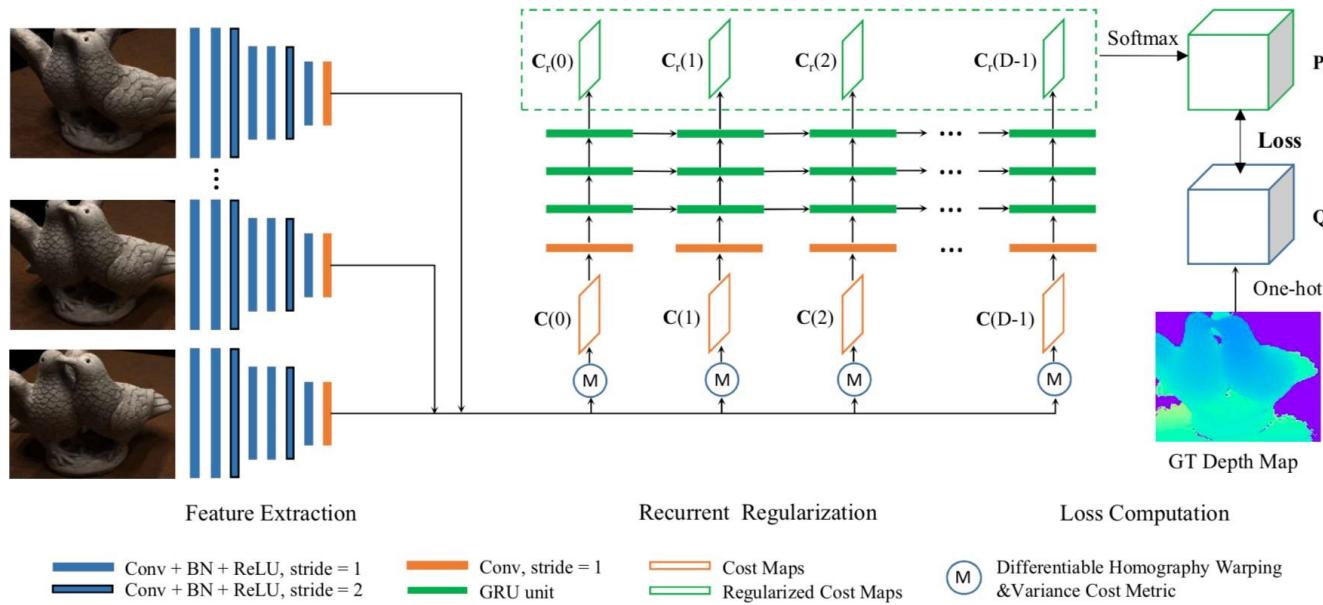
Still very coarse resolution (32x32x32) due to volumetric representation.

- View-aligned cost-volume construction
- Differentiable soft-argmin to achieve sub-pixel accuracy



$$\text{soft argmin} := \sum_{d=0}^{D_{\max}} d \times \sigma(-c_d)$$

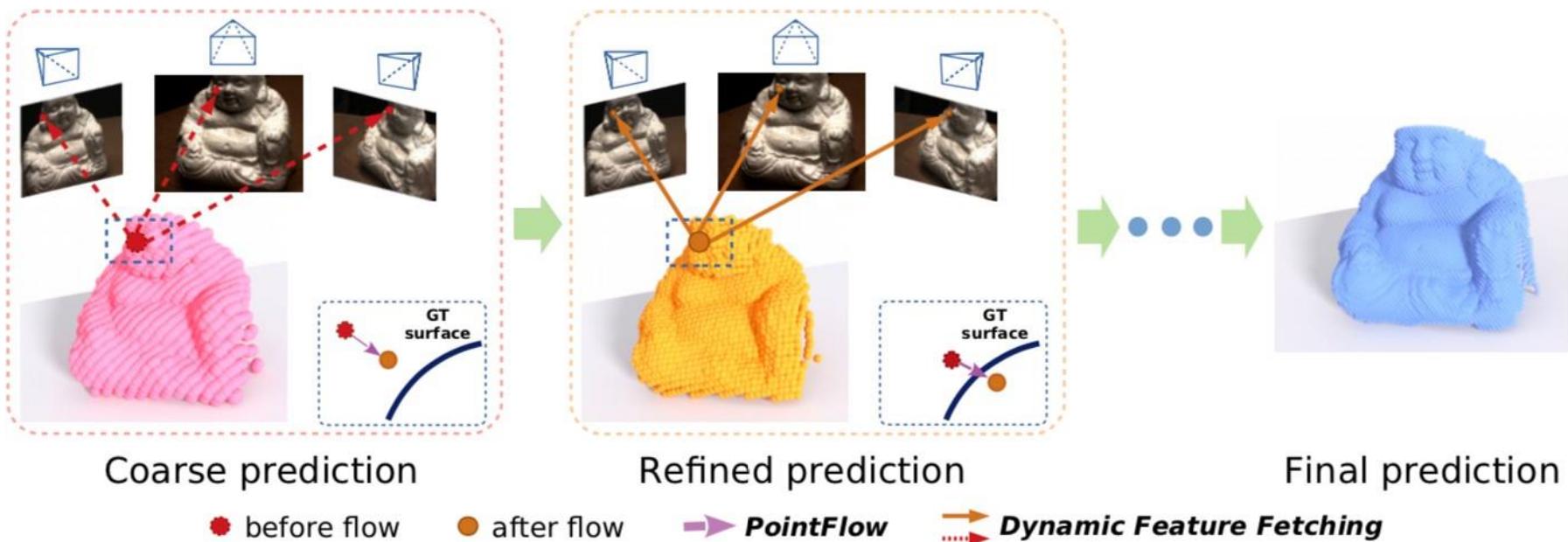
Idea 1: Slide-by-Slide Processing of Cost Volume by Recurrent Neural Network



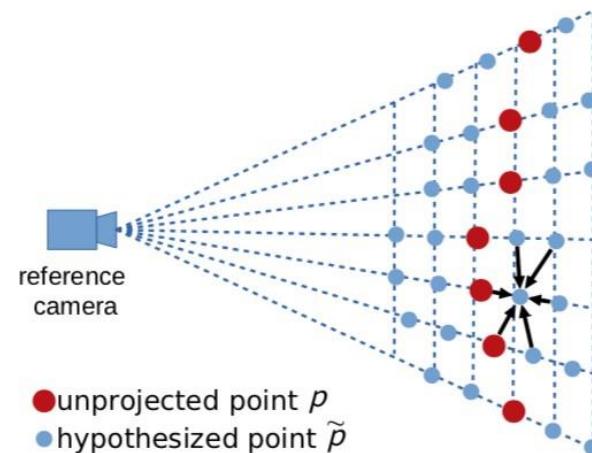
The cost volume is sequentially regularized along the depth direction.

Idea 2: Point-based MVS

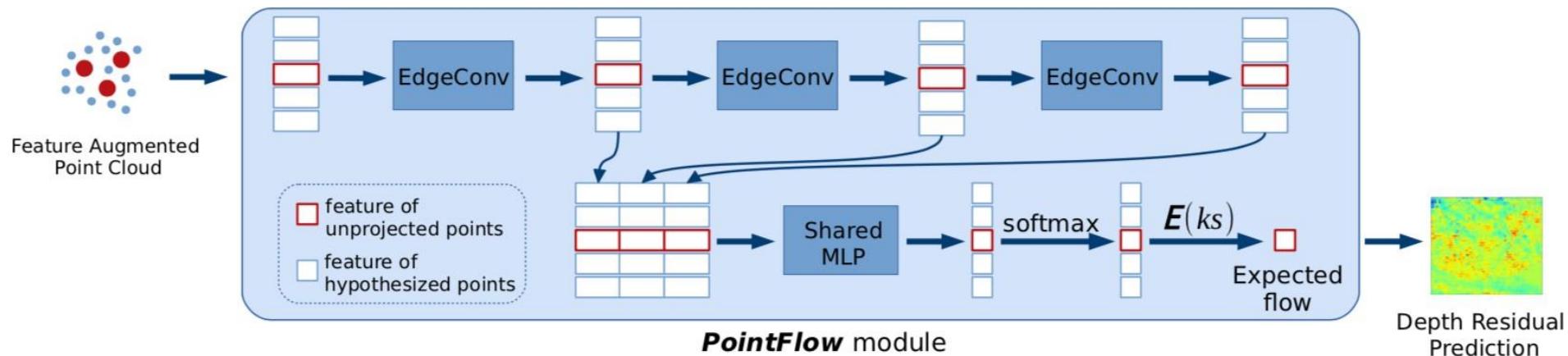
- Point-based representation for computational efficiency.
- Iteratively update the location of points and spawn more points.
- More flexible and accurate.



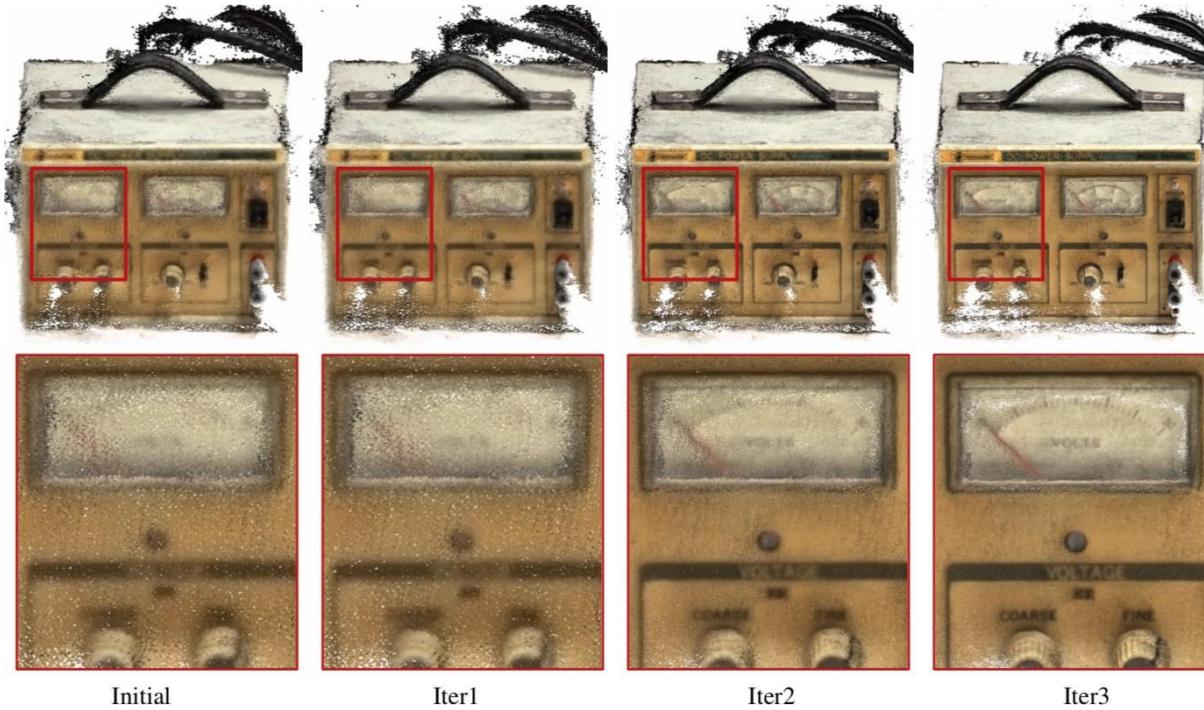
hypothesized point generation
along camera direction:



- PointFlow: iteratively pull all the points to the right position.



Iterative refinement:



Results on DTU benchmark

Iter.	Acc. (mm)	Comp. (mm)	Overall (mm)	0.5mm <i>f-score</i>	Depth Map Res.	Depth Interval (mm)	GPU Mem. (MB)	Runtime (s)
-	0.693	0.758	0.726	47.95	160×120	5.30	7219	0.34
1	0.674	0.750	0.712	48.63	160×120	5.30	7221	0.61
2	0.448	0.487	0.468	76.08	320×240	4.00	7235	1.14
3	0.361	0.421	0.391	84.27	640×480	0.80	8731	3.35
MVSNet[29]	0.456	0.646	0.551	71.60	288×216	2.65	10805	1.05

The End