UC San Diego

3D Deep Learning: An Overview based on My Work

Hao Su

Feb 23, 2018

Our world is 3D

Broad applications of 3D data

Roboti

Broad applications of 3D data

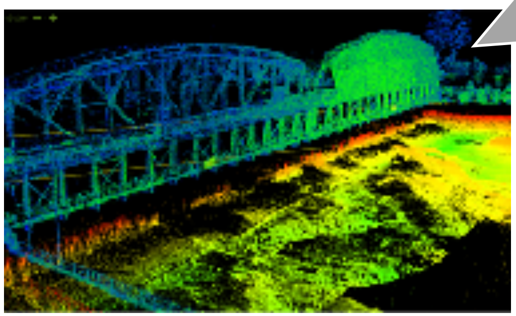
Roboti

Augmented

Broad applications of 3D data

Roboti

Augmented

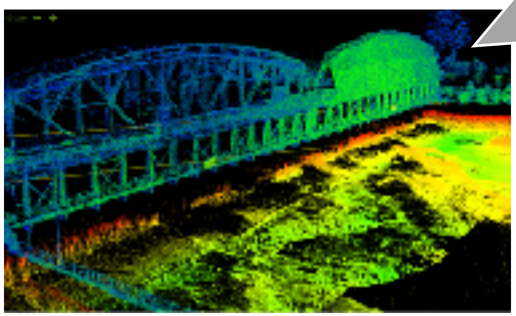


Autonomous

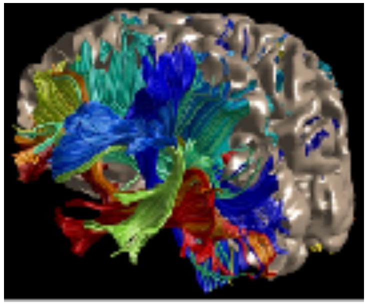
Broad applications of 3D data

Roboti

Augmented



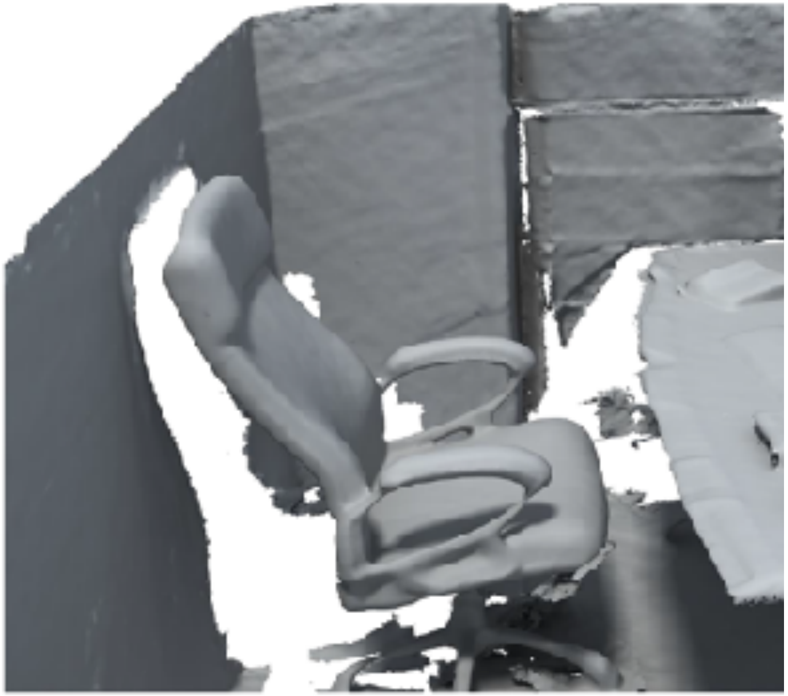
Autonomous



Medical Image Processing

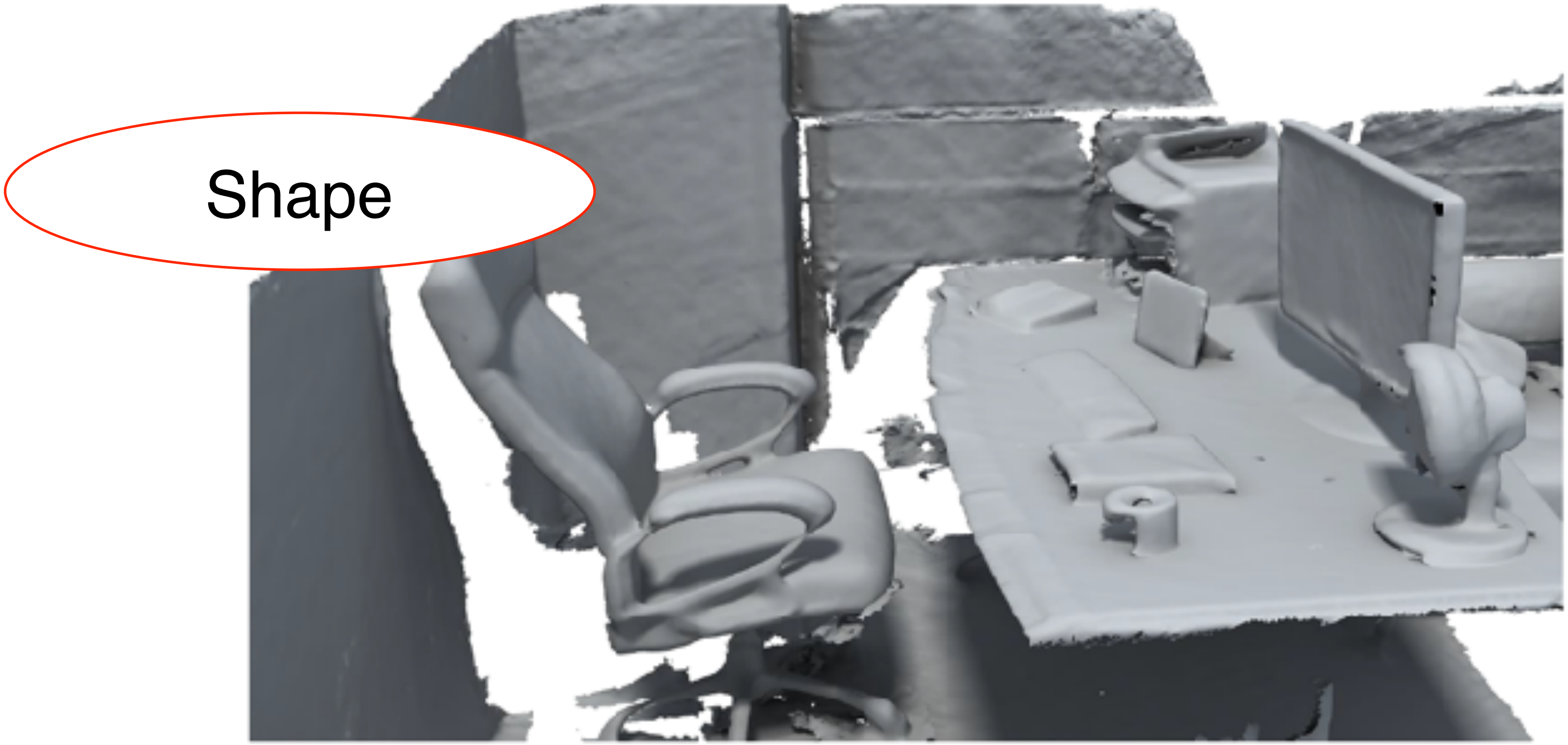
3D Understanding Enables Interactions

Example: 3D understanding for a robot

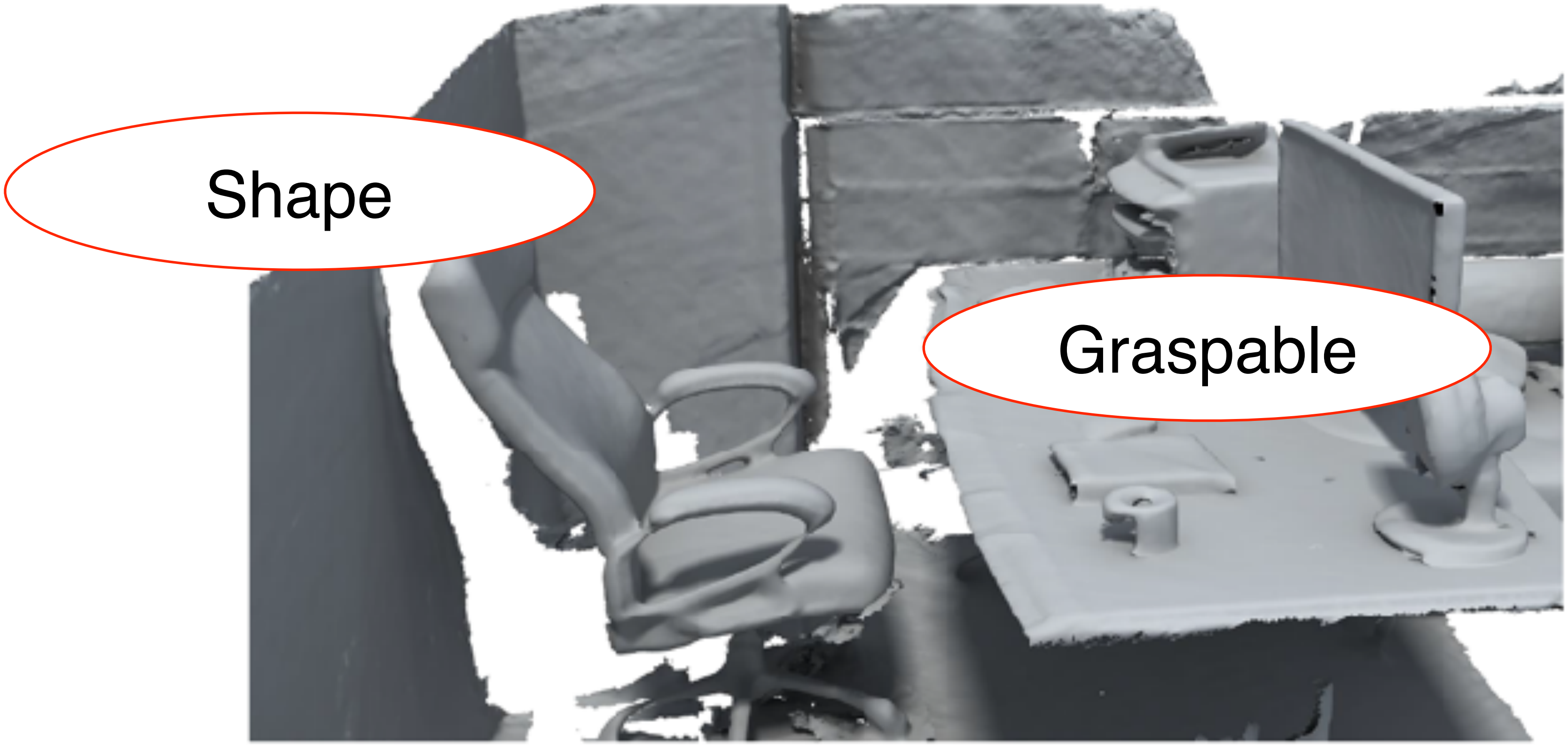


[SIGGRAPH Asia 2016]

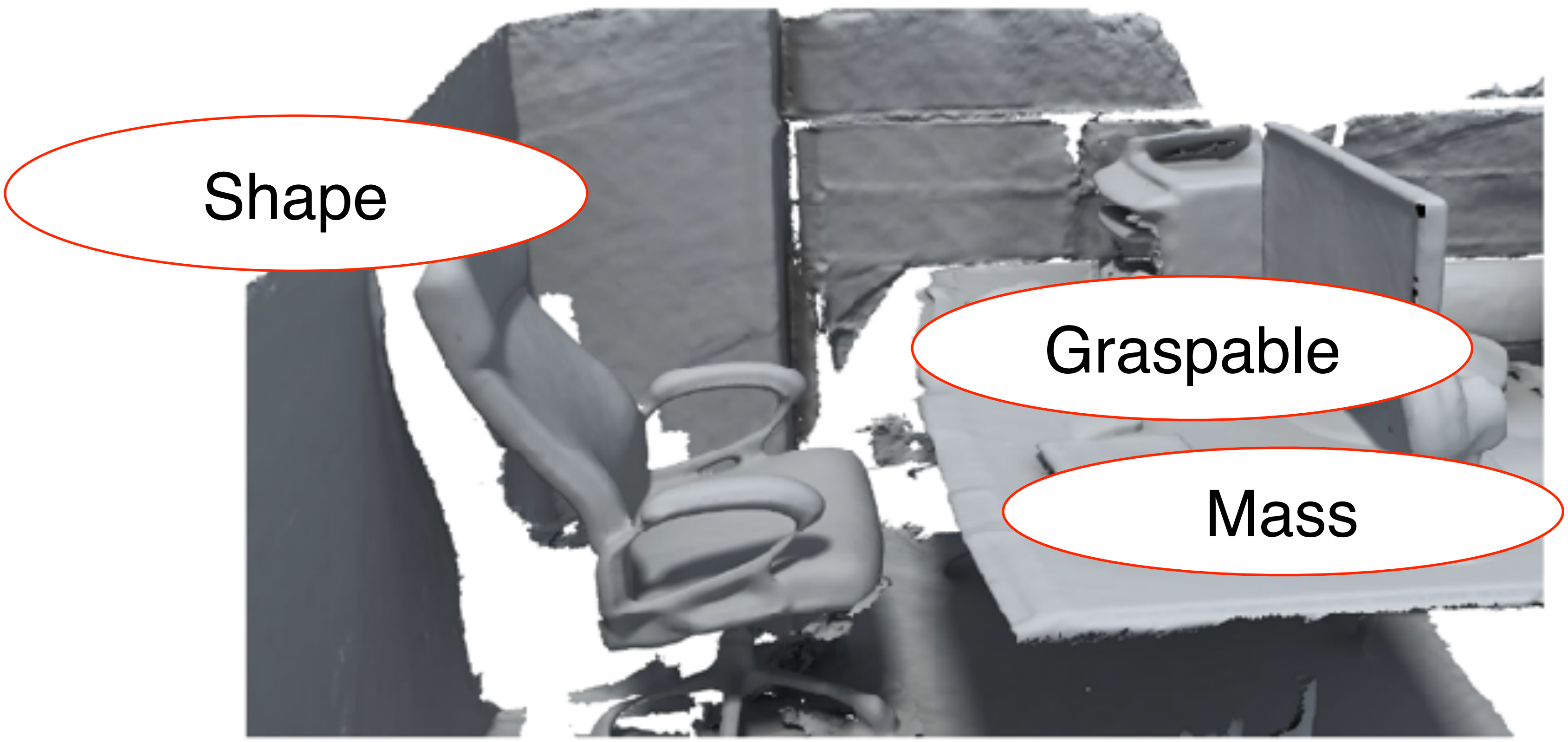
3D Understanding Enables Interactions



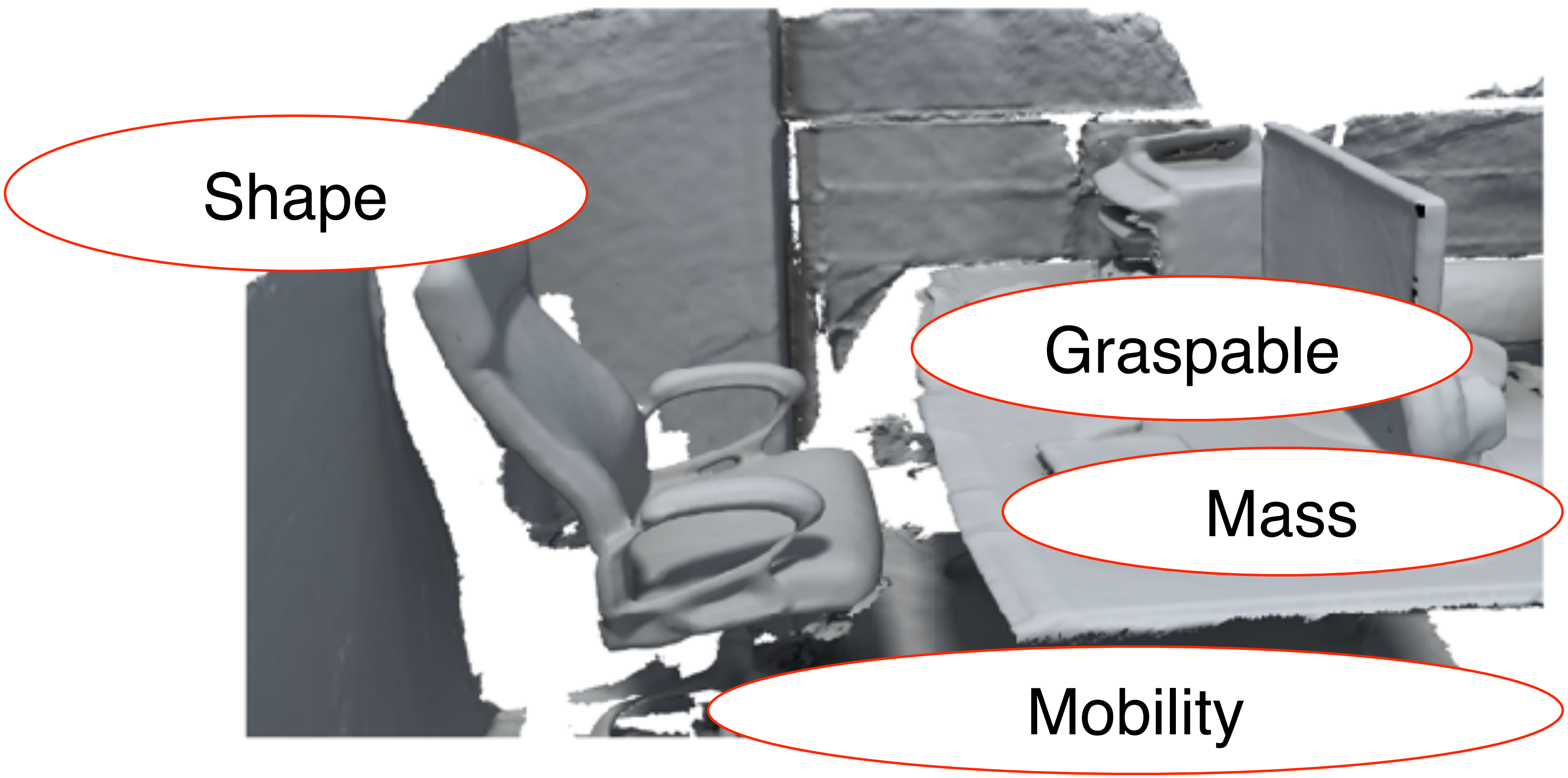
3D Understanding Enables Interactions



3D Understanding Enables Interactions



3D Understanding Enables Interactions

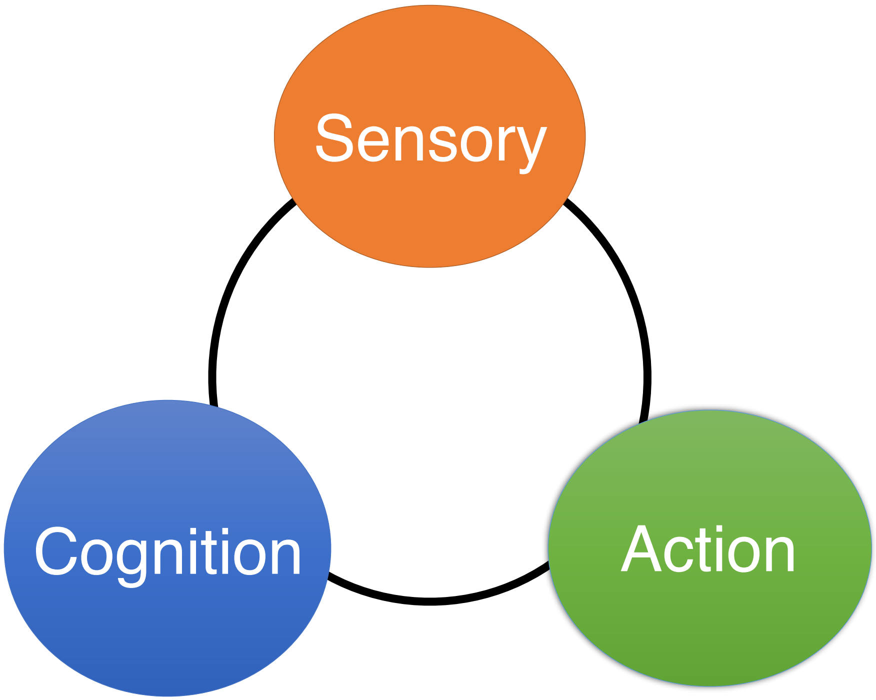


AI Perspective of 3D Understanding

See the world

Understand the world

Transform the world

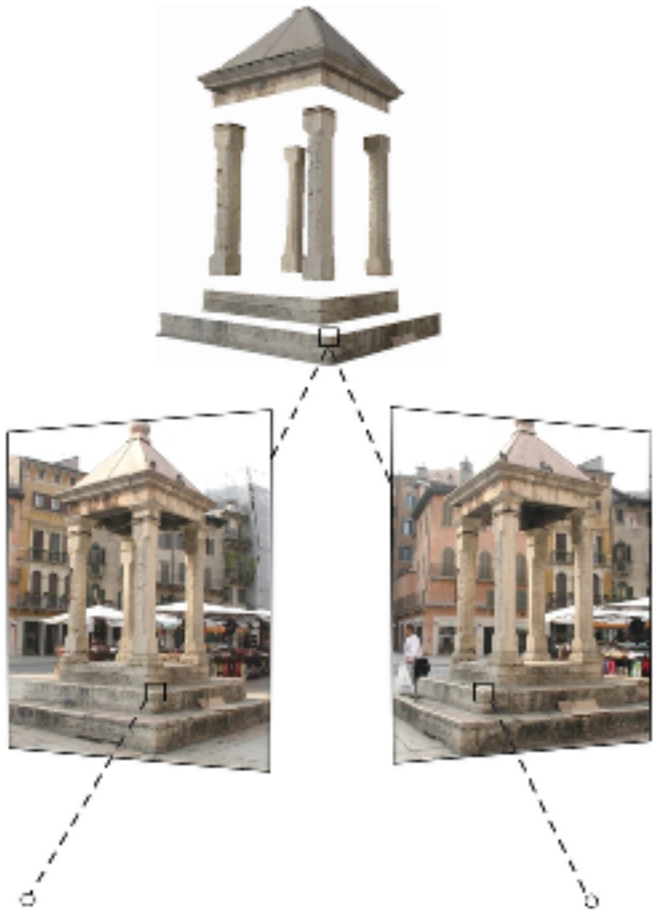


Towards **interaction** with the physical world,
3D is the key!

3D Perception requires "Knowledge" of 3D World

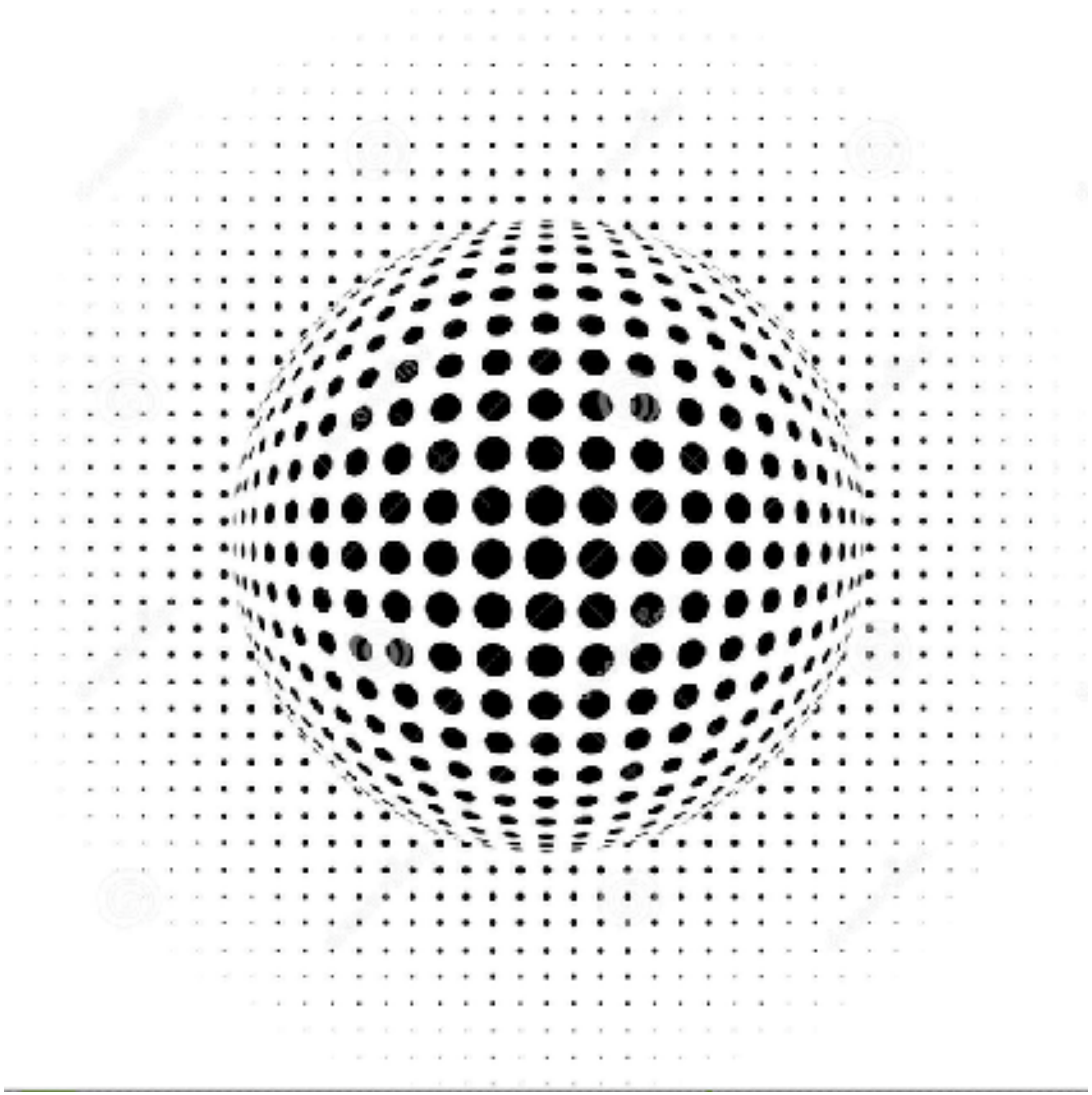
Traditional 3D Vision

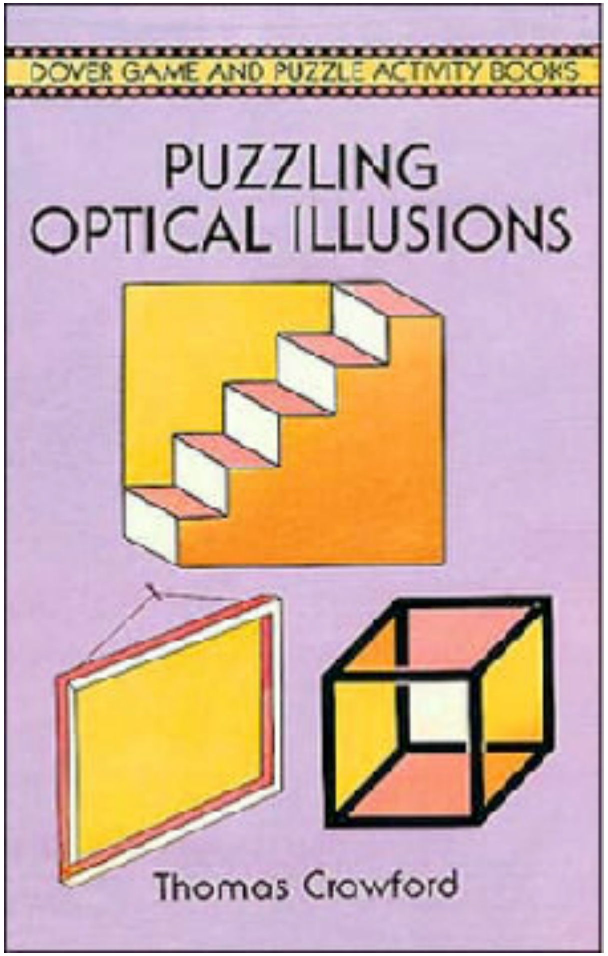
Multi-view Geometry: Physics based



3D Learning: Knowledge Based

3D Learning: Knowledge Based





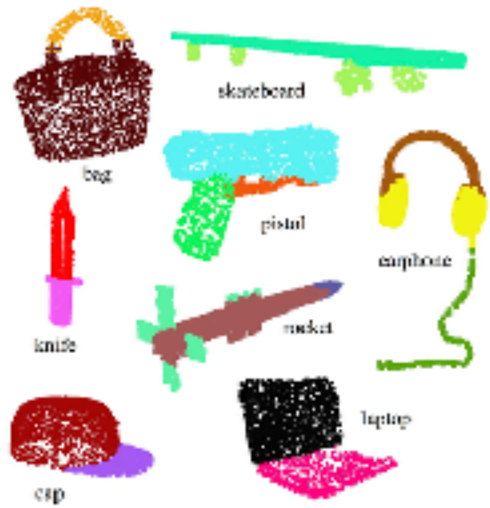
Acquire Knowledge of 3D World by Learning

A priori knowledge of the 3D world

3D Learning Tasks

3D Analysis

Classification



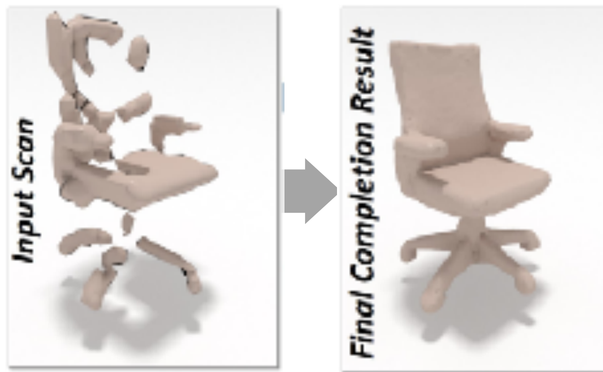
Segmentation
(object/scene)

Correspondence

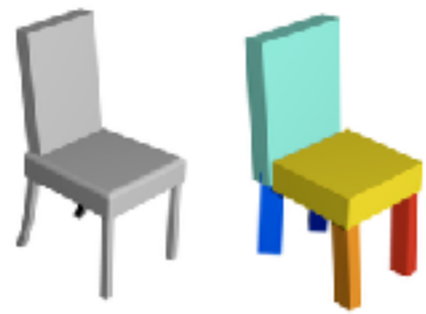
3D Learning Tasks

3D Synthesis

Monocular
3D reconstruction



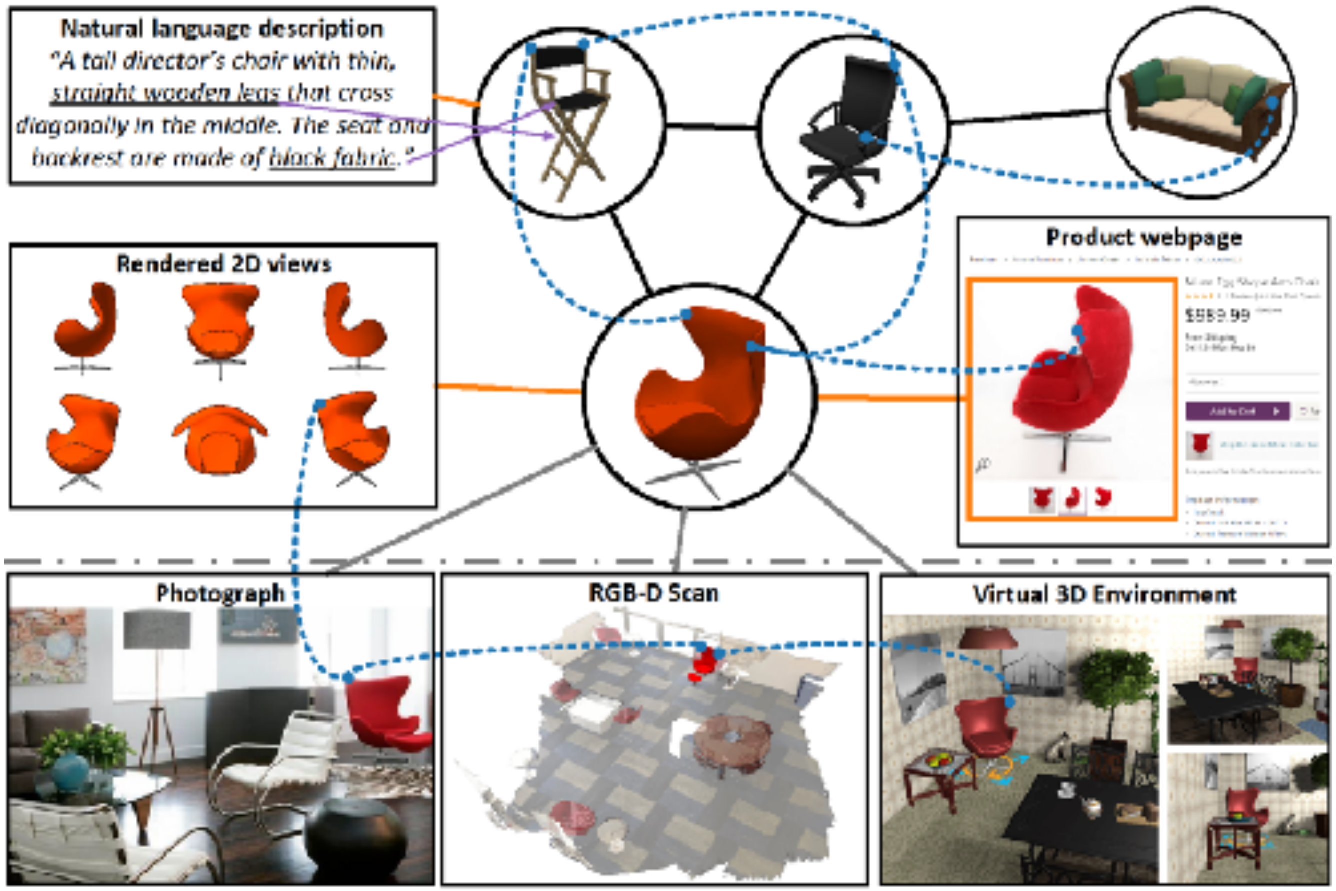
Shape completion



Shape modeling

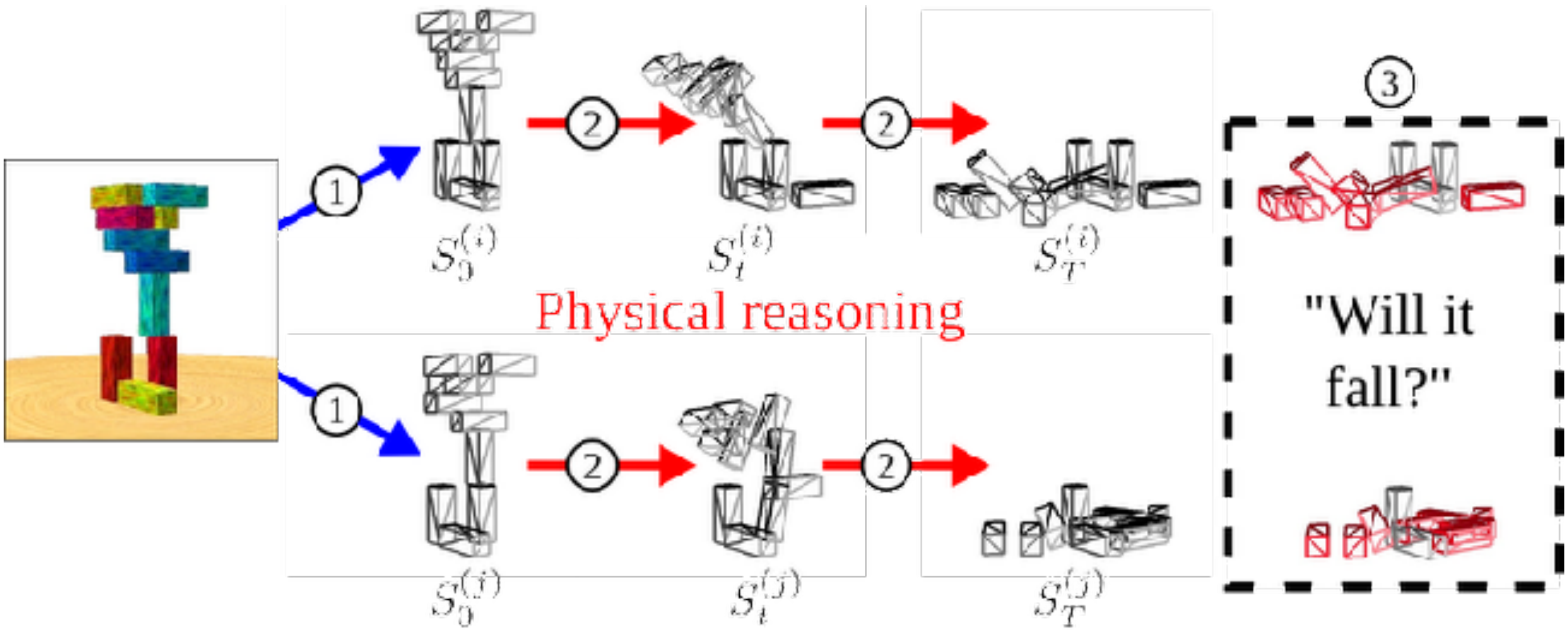
3D Learning Tasks

3D-based Knowledge Transportation

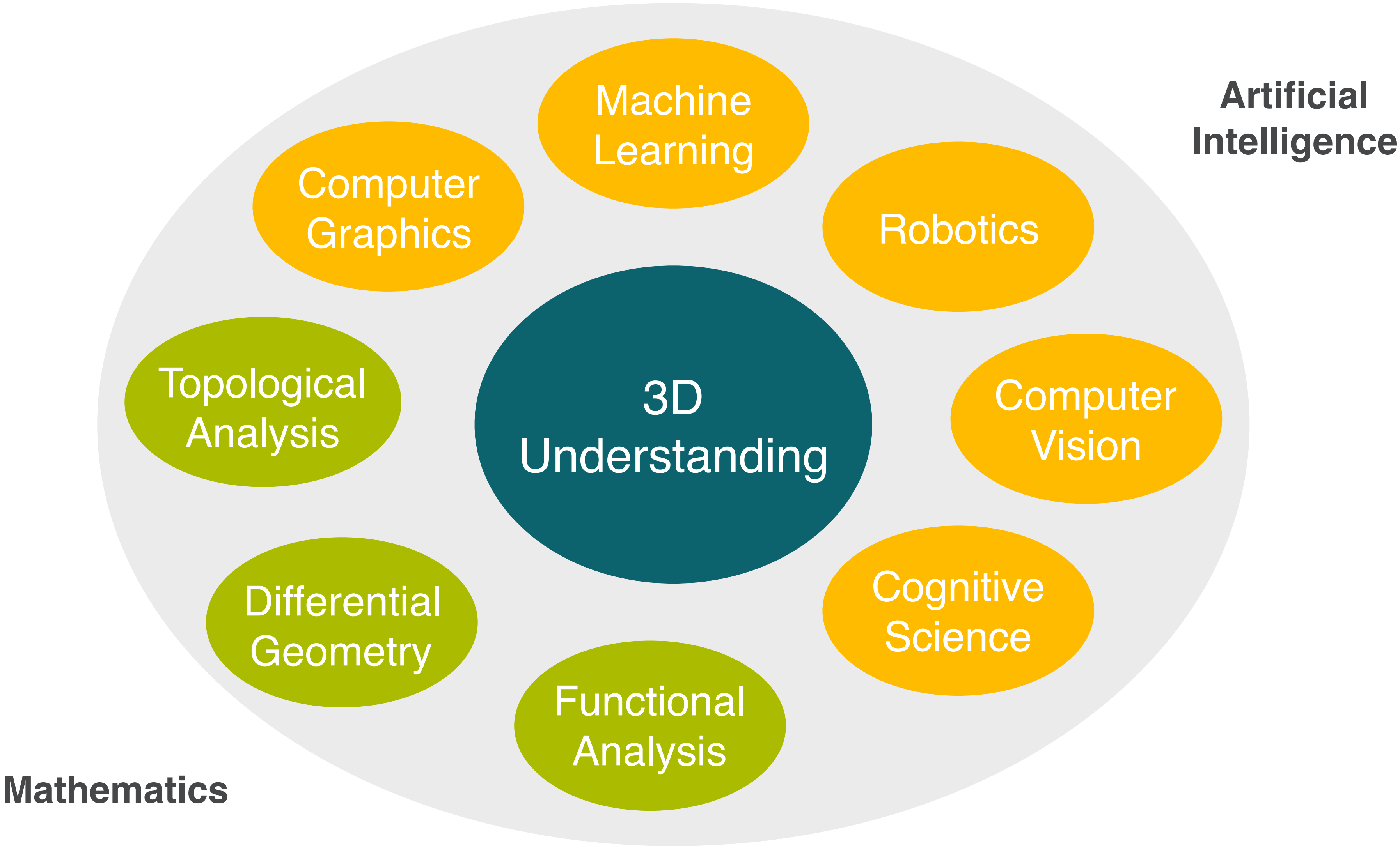


3D Learning Tasks

Intuitive Physics based on 3D Understanding



Deep Learning on 3D: A New Rising Field



Outline

Overview of 3D Deep Learning

3D Deep Learning Algorithms

The Representation Issue of 3D Deep Learning

Images: Unique representation with regular data structure

1	44	33	12	20	23	35	14
51	16	40	32	46	48	28	17
29	60	3	63	49	55	36	7
52	22	26	41	38	10	61	53
2	24	19	11	34	43	5	8
57	9	37	42	25	21	27	18
30	56	50	64	4	59	6	13
58	47	45	31	39	15	62	54

The Representation Issue of 3D Deep Learning

3D has many representations:

multi-view RGB(D) images

volumetric

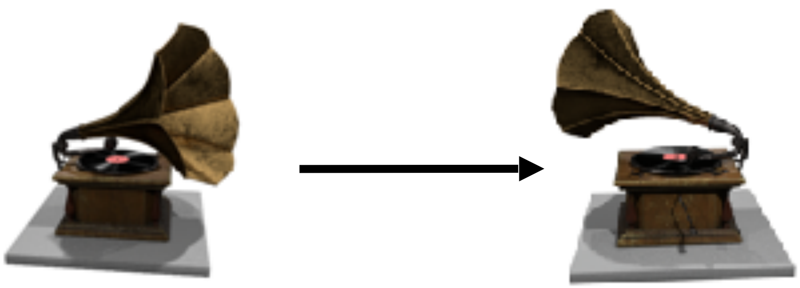
polygonal mesh

point cloud

primitive-based models

The Representation Issue of 3D Deep Learning

3D has many representations:



Novel view image synthesis

multi-view RGB(D) images

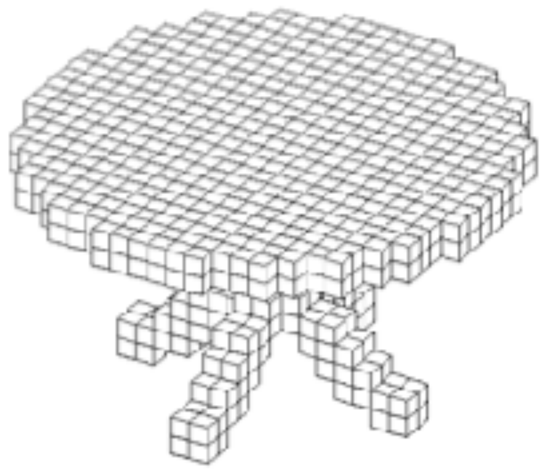
volumetric

polygonal mesh

point cloud

primitive-based models

The Representation Issue of 3D Deep Learning



3D has many representations:

multi-view RGB(D) images

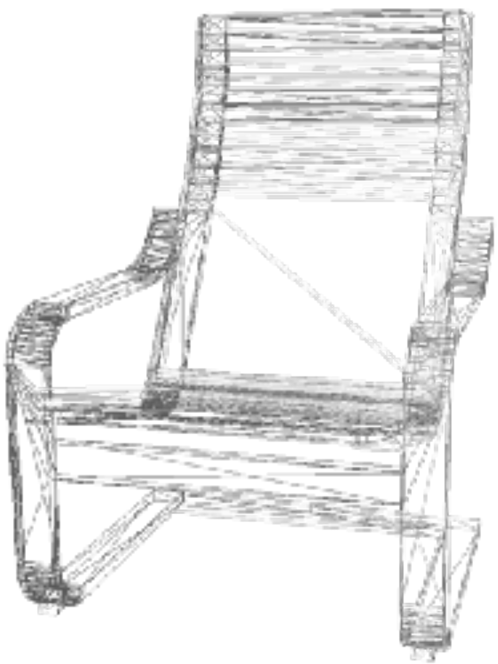
volumetric

polygonal mesh

point cloud

primitive-based models

The Representation Issue of 3D Deep Learning



3D has many representations:

multi-view RGB(D) images

volumetric

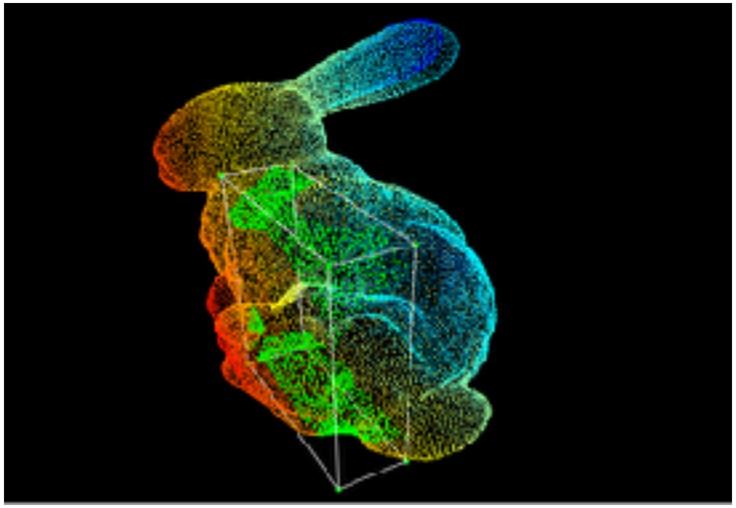
polygonal mesh

point cloud

primitive-based models

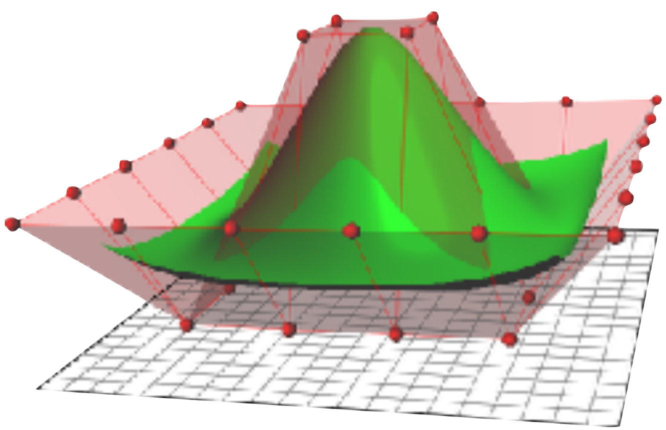
The Representation Issue of 3D Deep Learning

3D has many representations:



- multi-view RGB(D) images
- volumetric
- polygonal mesh
- **point cloud**
- primitive-based models

The Representation Issue of 3D Deep Learning



3D has many representations:

multi-view RGB(D) images

volumetric

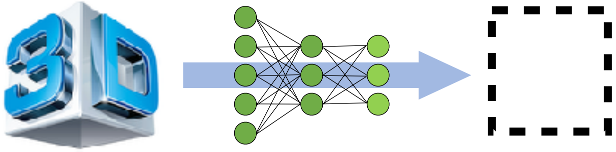
polygonal mesh

point cloud

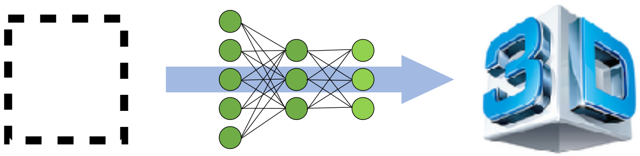
primitive-based models

Cartesian Product Space of “Task” and “Representation”

3D geometry analysis



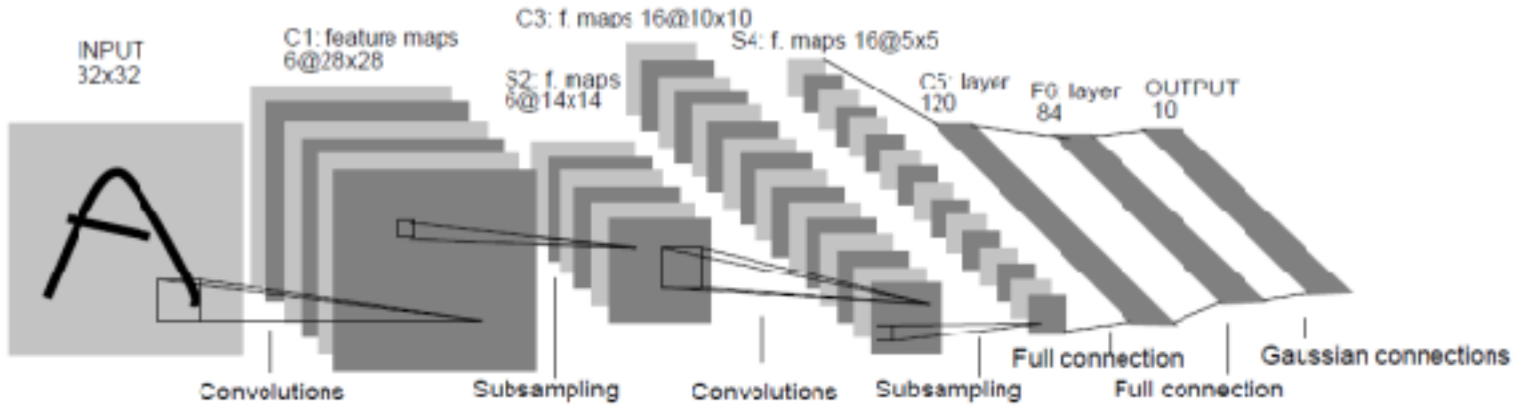
3D synthesis



Fundamental Challenges of 3D Deep Learning

Convolution needs an underlying structure

Can we directly apply CNN on 3D data?



Rasterized vs Geometric

3D has many representations:

Rasterized form (regular grids)

- Can directly apply CNN
- But has other challenges

multi-view RGB(D) images

volumetric

Fundamental Challenges of 3D Deep Learning

3D has many representations:

Rasterized form
(regular grids)

multi-view RGB(D) images

volumetric

Geometric form
(irregular)

Cannot directly apply CNN

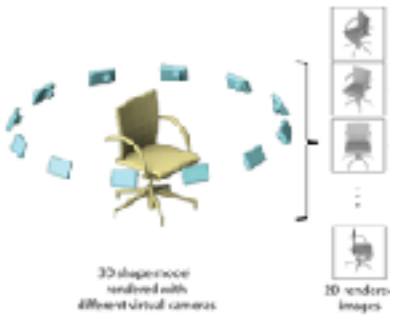
polygonal mesh

point cloud

primitive-based models

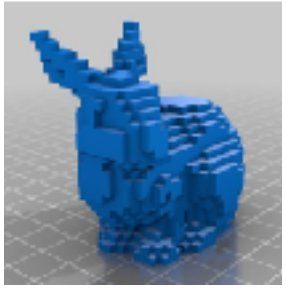
3D Deep Learning Algorithms (by Representations)

- Projection-based



[Su et al. 2015]
[Kalogerakis et al. 2016]
…

Multi-view

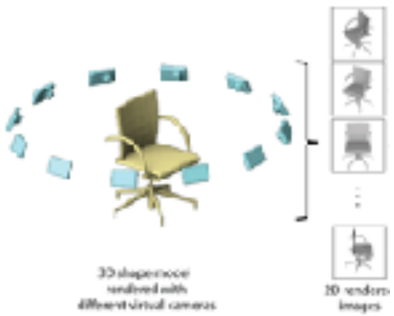


[Maturana et al. 2015]
[Wu et al. 2015] (GAN)
[Qi et al. 2016]
[Liu et al. 2016]
[Wang et al. 2017] (O-Net)
[Tatarchenko et al. 2017] (OGN)
…

Volumetric

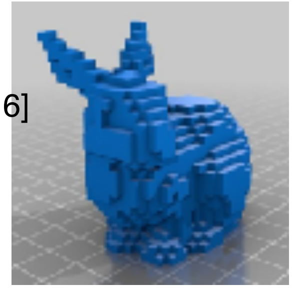
3D Deep Learning Algorithms (by Representations)

- Projection-based



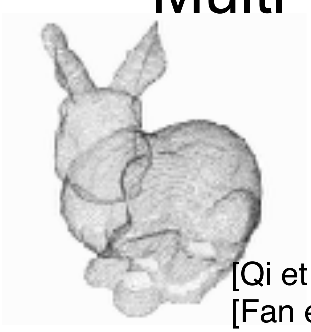
[Su et al. 2015]
[Kalogerakis et al. 2016]
…

Multi-view



[Maturana et al. 2015]
[Wu et al. 2015] (GAN)
[Qi et al. 2016]
[Liu et al. 2016]
[Wang et al. 2017] (O-Net)
[Tatarchenko et al. 2017] (OGN)
…

Volumetric

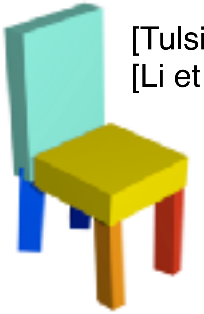


[Qi et al. 2017] (PointNet)
[Fan et al. 2017] (PointSetGen)

Point cloud

[Defferard et al. 2016]
[Henaff et al. 2015]
[Yi et al. 2017] (SyncSpecCN
…

Mesh (Graph CNN)



[Tulsiani et al. 2017]
[Li et al. 2017] (GRASS)

Part assembly

Fundamental Challenges of 3D Deep Learning

3D has many representations:

Rasterized form (regular grids)

- Can directly apply CNN
- But has other challenges

multi-view RGB(D) images

volumetric

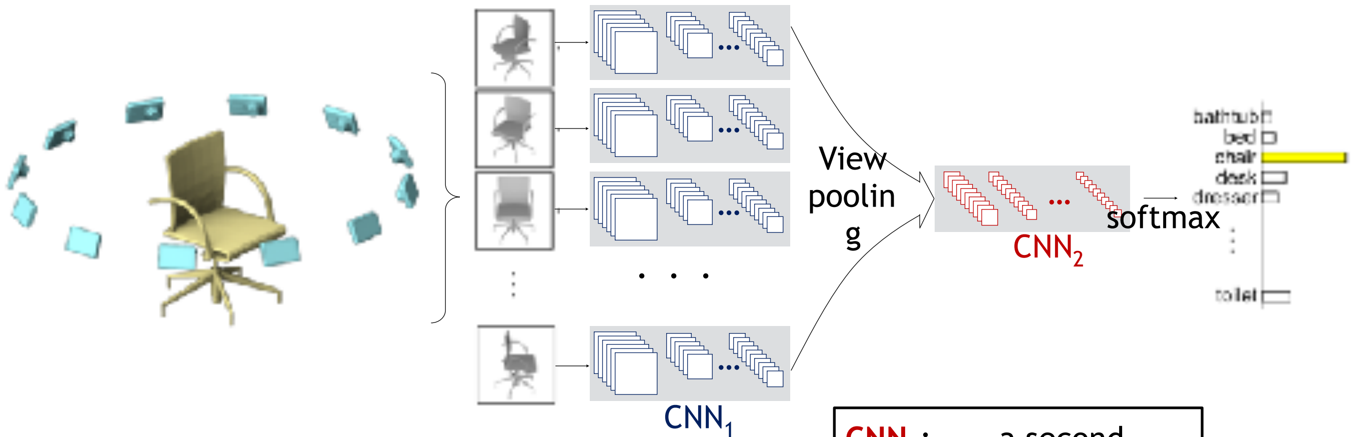
Deep Learning on Multi-view Representation

Multi-view Representation as 3D Input

- Leverage the huge CNN literature in image analysis

Multi-view Representation as 3D Input

- Classification



CNN$_2$: a second ConvNet producing shape descriptors

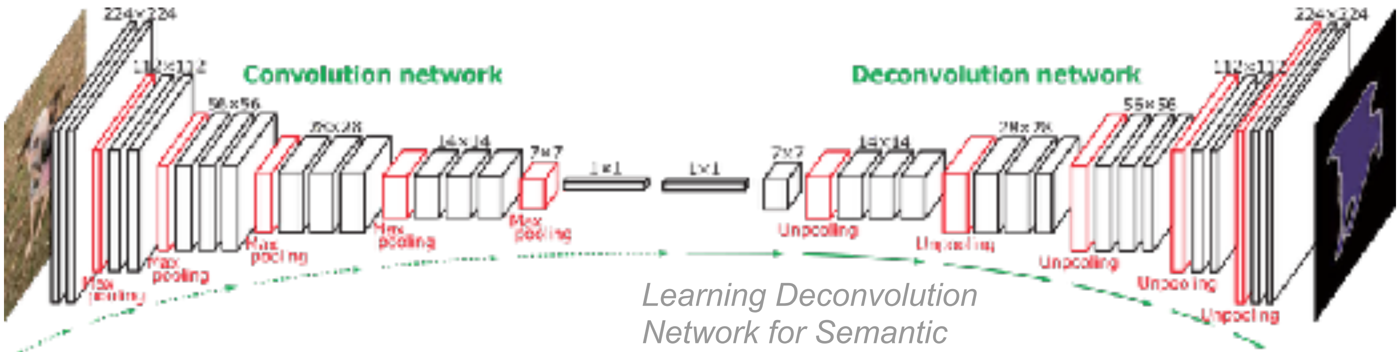
Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-Miller, "**Multi-view Convolutional Neural Networks for 3D Shape Recognition**", *Proceedings of ICCV 2015*

Multi-view Representation as 3D Output

- The Novel-view Synthesis Problem

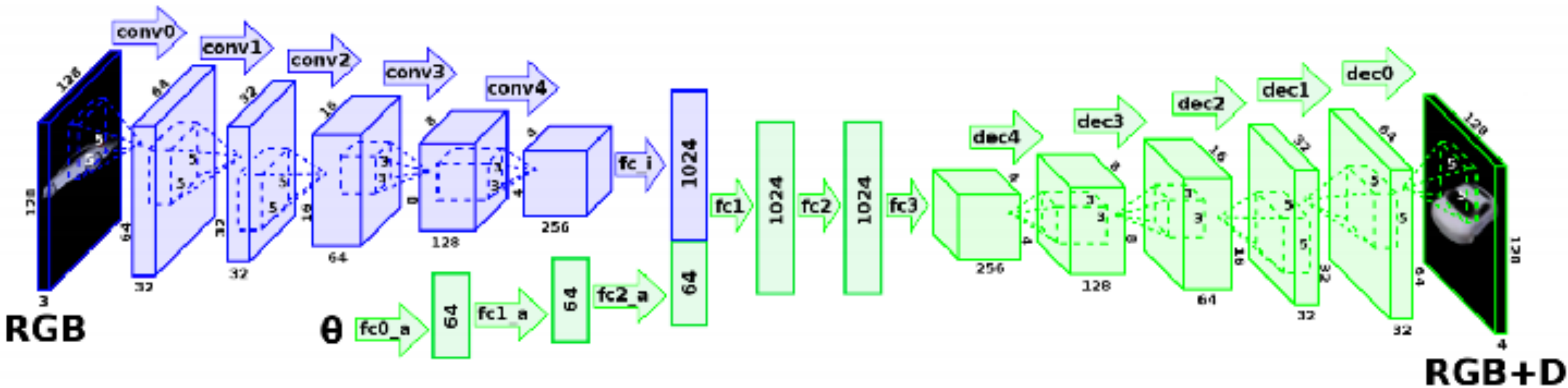
Fully Convolutional Network (FCN)

Segmentation:



Learning Deconvolution Network for Semantic Segmentation

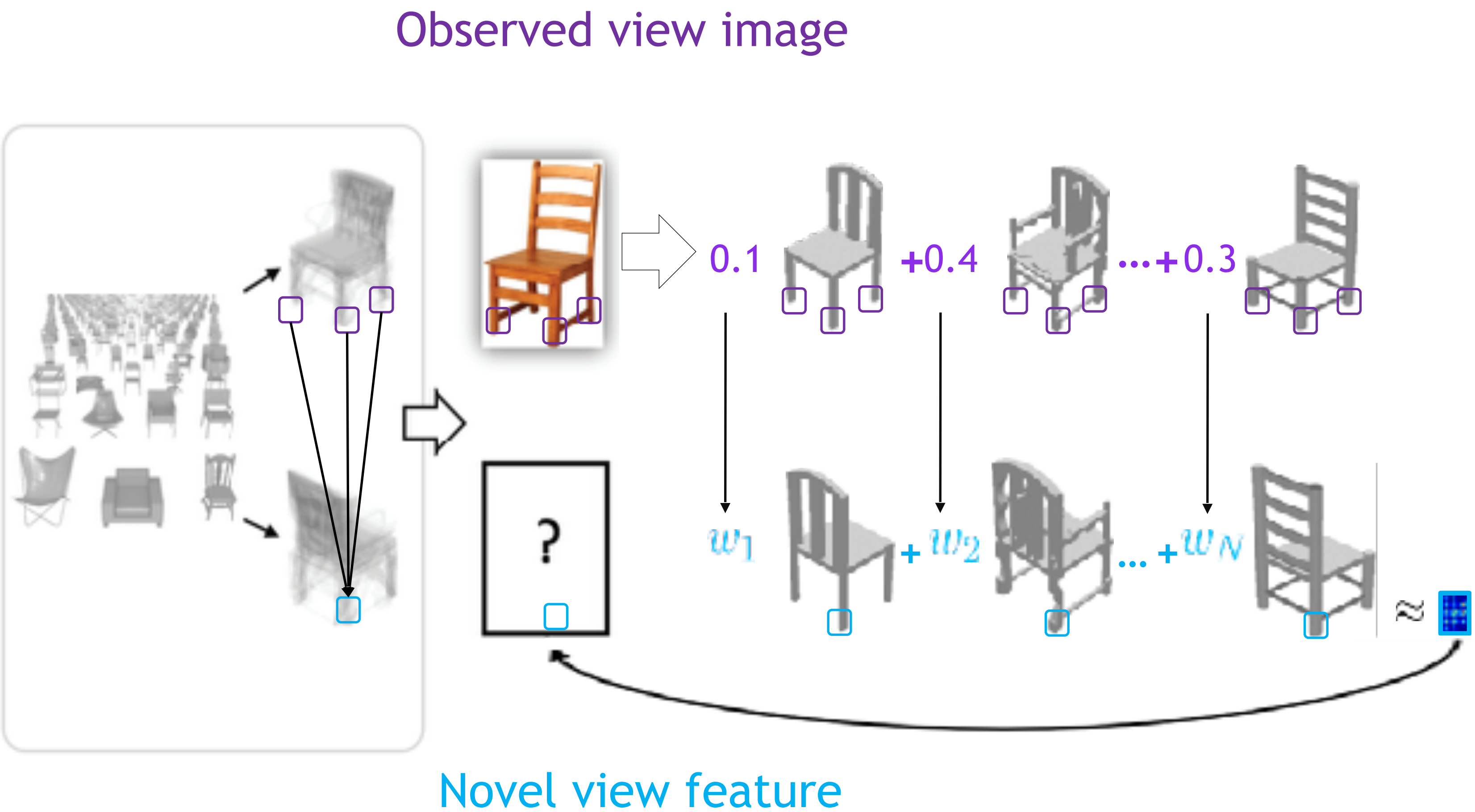
Idea 1: Direct Novel-view Synthesis



Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox,
"Multi-view 3D Models from Single Images with a Convolutional Network",
ECCV2016

Results are often Blurry

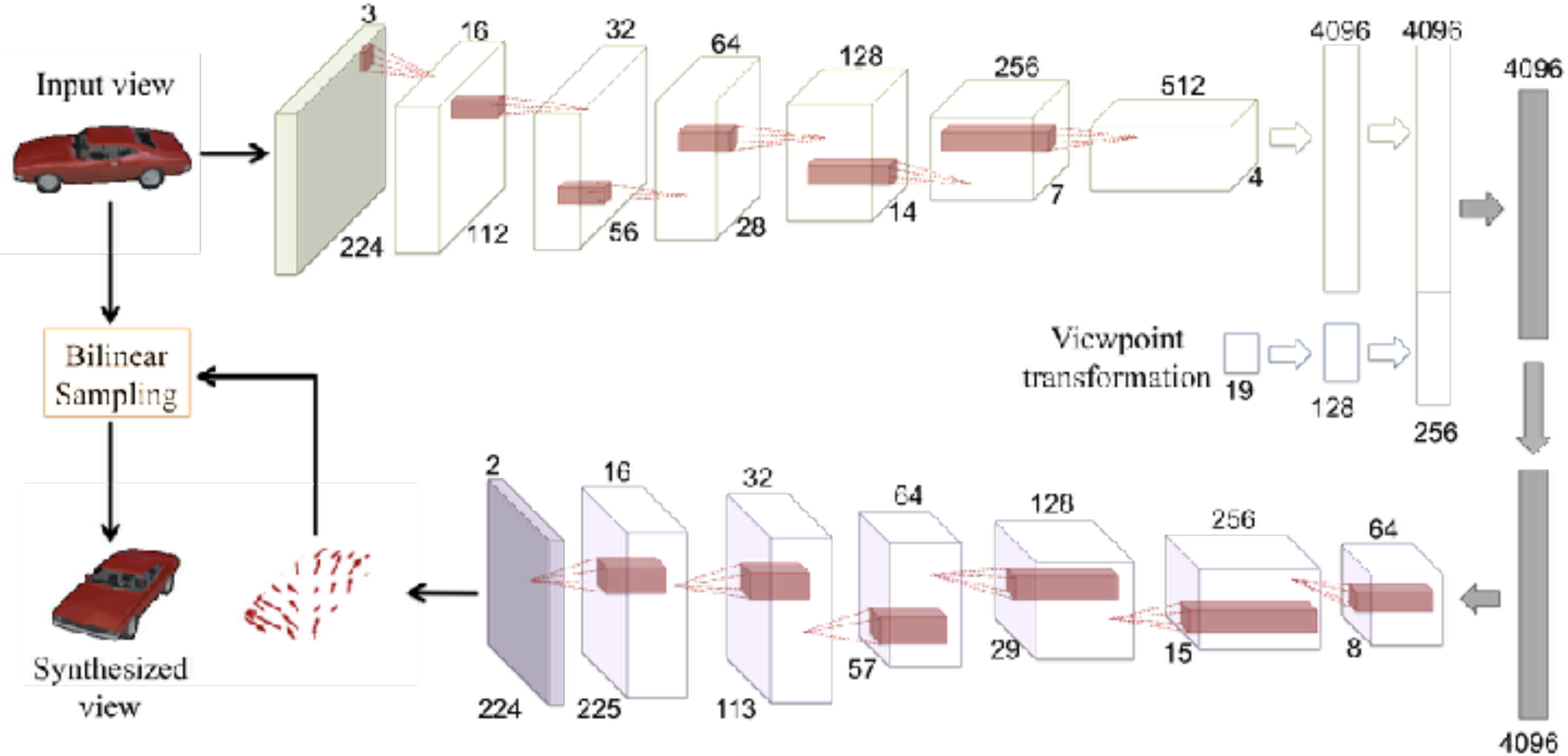
Idea 2: Explore Cross-View Relationship



Su et al, 3D-Assisted Image Feature Synthesis for Novel Views of an Object, ECCV 2016

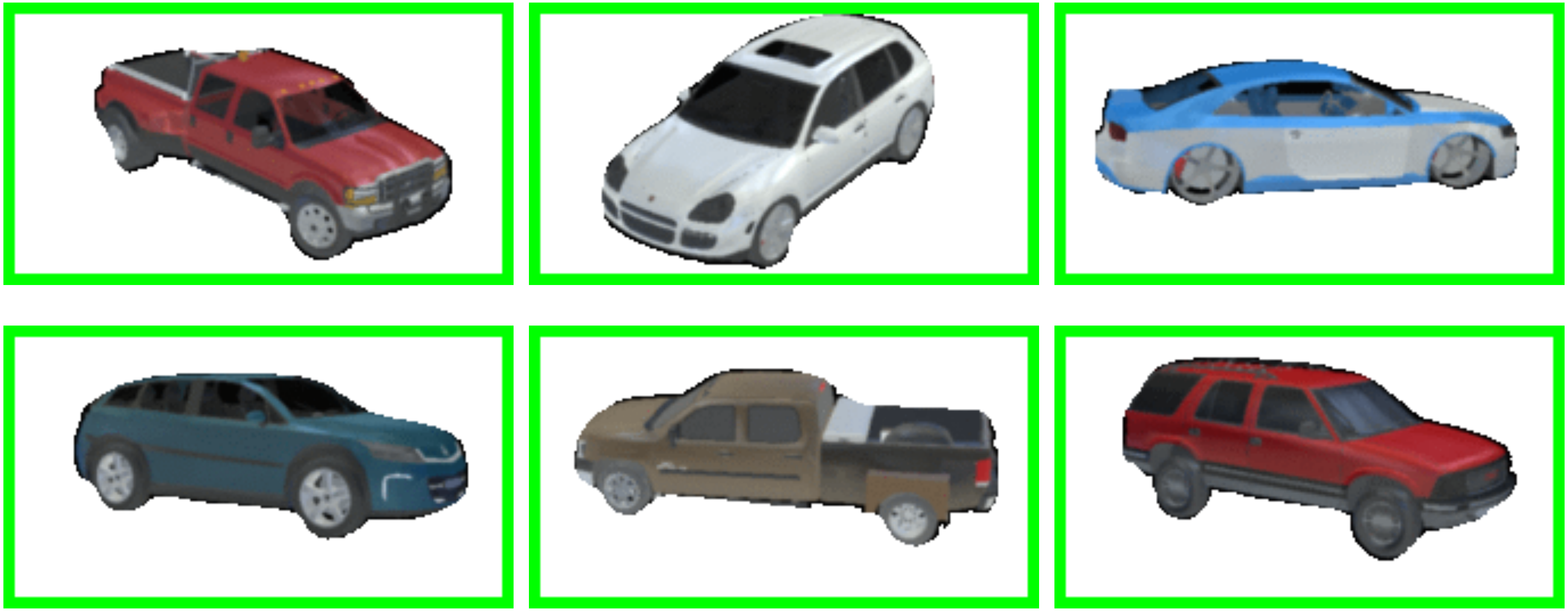
Idea 2: Explore Cross-View Relationship

Single-view network architecture:

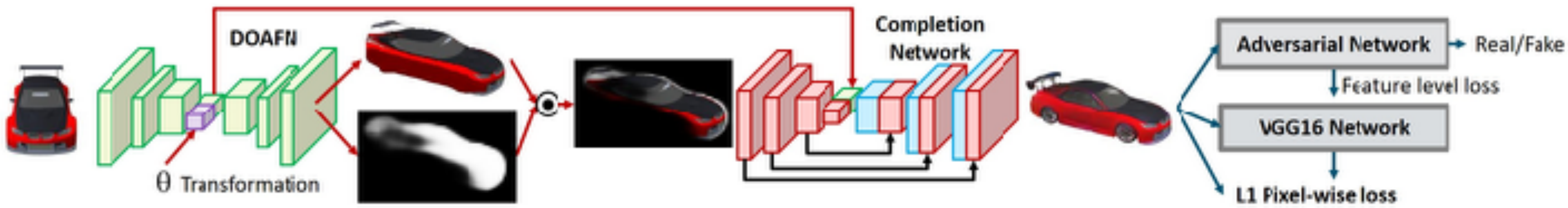


Zhou et al, View Synthesis by Appearance Flow, ECCV 2016

Idea 2: Explore Cross-View Relationship



Combine both ideas

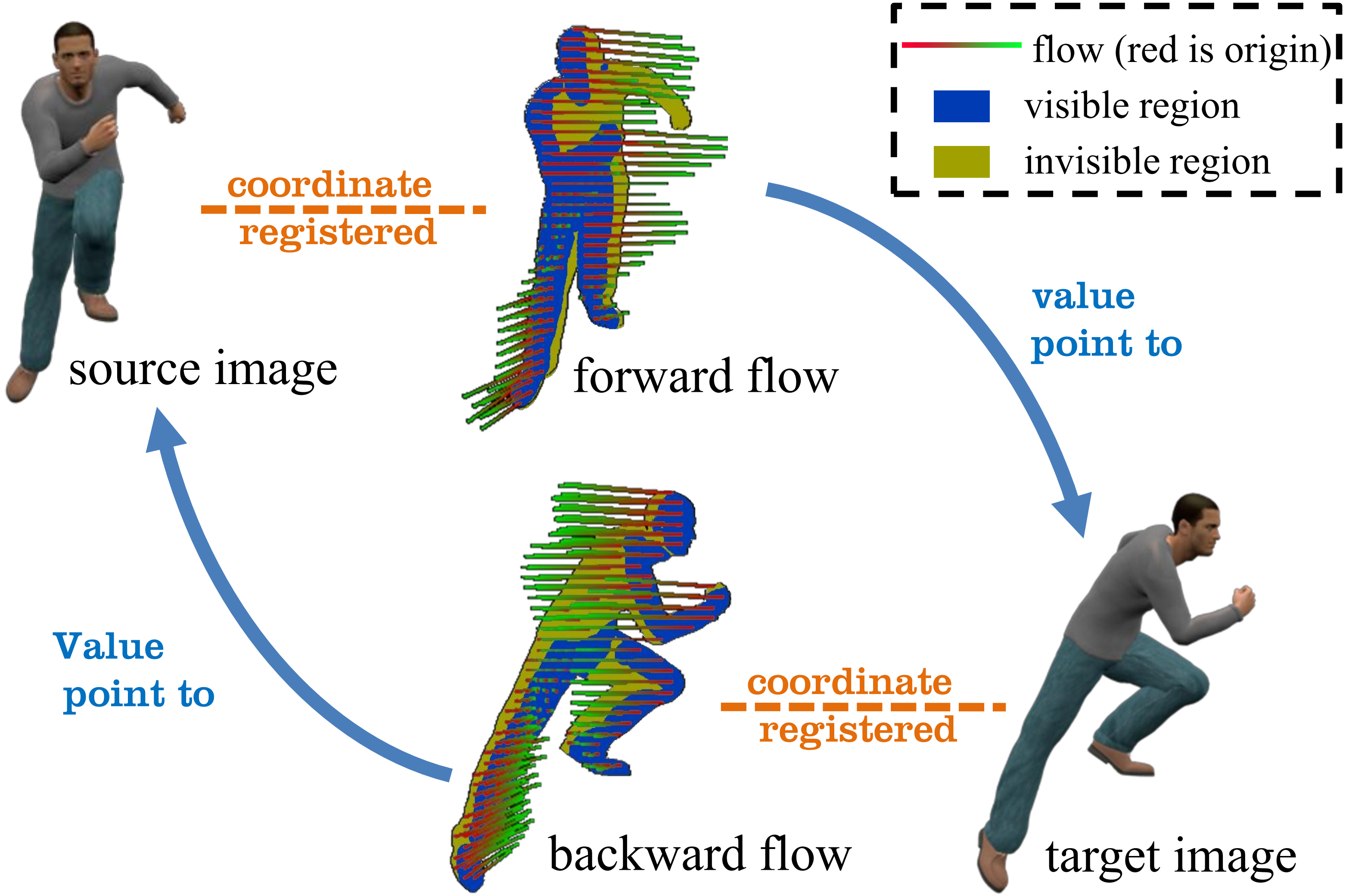


- First, apply flow prediction
- Second, conduct invisible part hallucination

Park et al, Transformation-Grounded Image Generation Network for Novel 3D View Synthesis, CVPR 2017

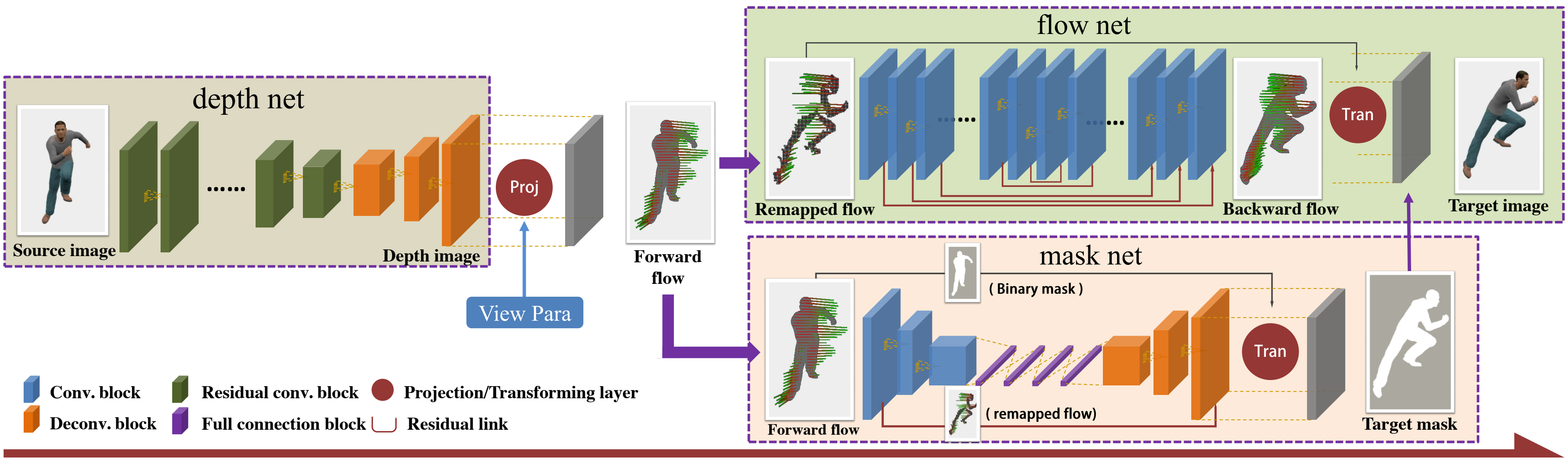
Combine both ideas

Articulated Shapes: Assist Flow Synthesis by Depth Estimation



My latest paper accepted by CVPR'18

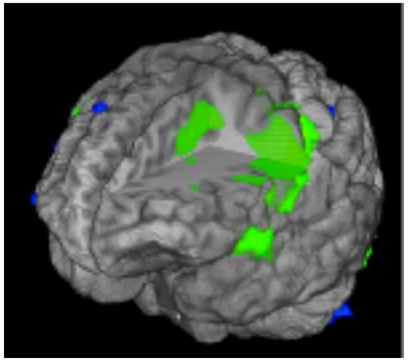
Articulated Shapes:
Assist Flow Synthesis by Depth Estimation



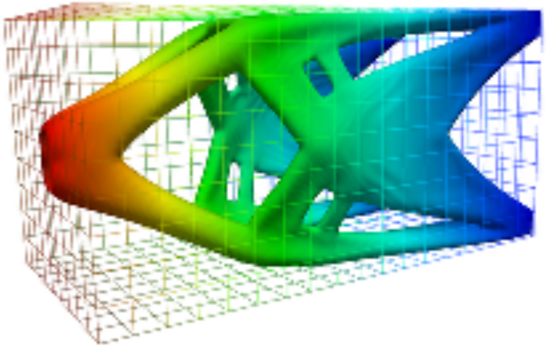
My latest paper accepted by CVPR'18

Deep Learning on Volumetric Representation

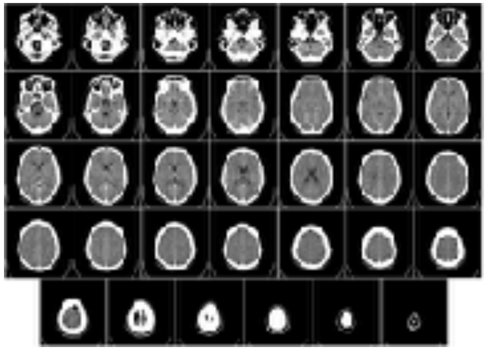
Popular 3D volumetric data



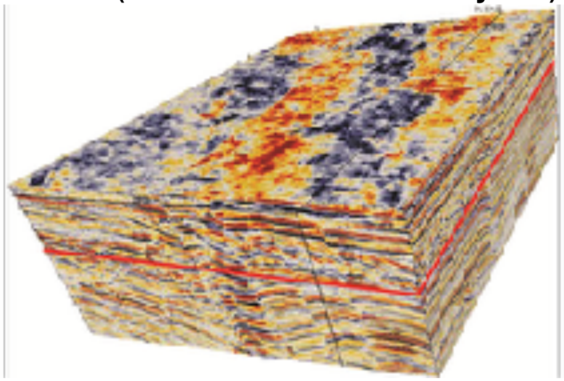
fMRI



Manufacturing
(finite-element analysis)



CT

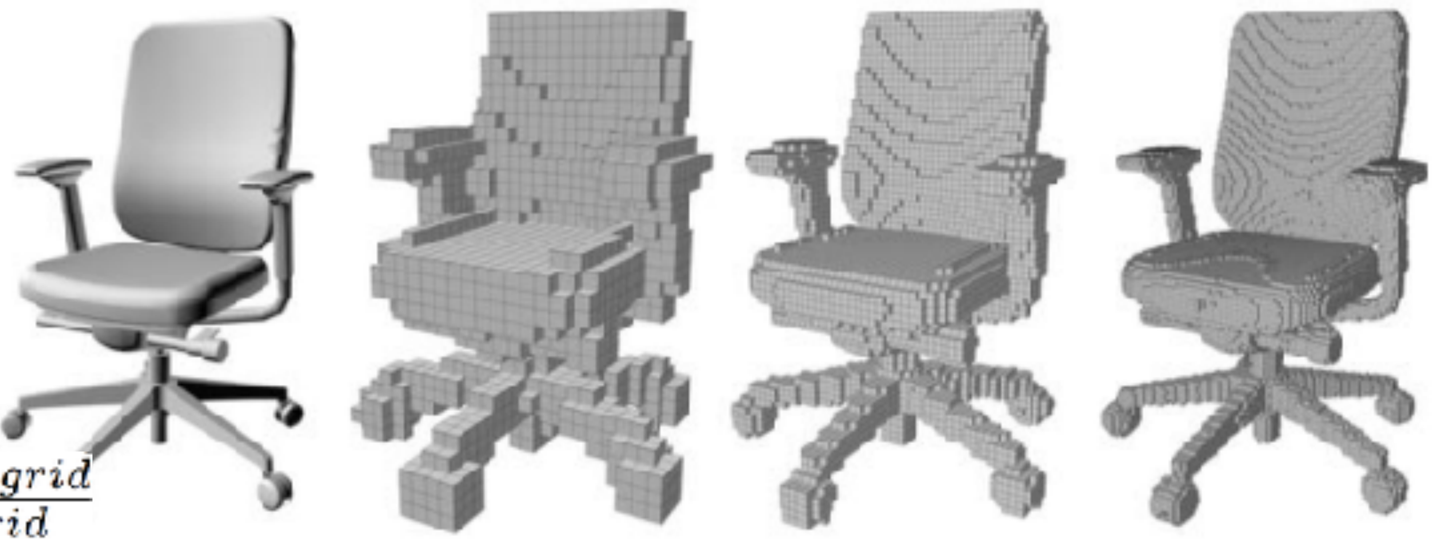


Geology

Volumetric Representation as 3D Input

- The main hurdle is Complexity

The Sparsity Characteristic of 3D Data

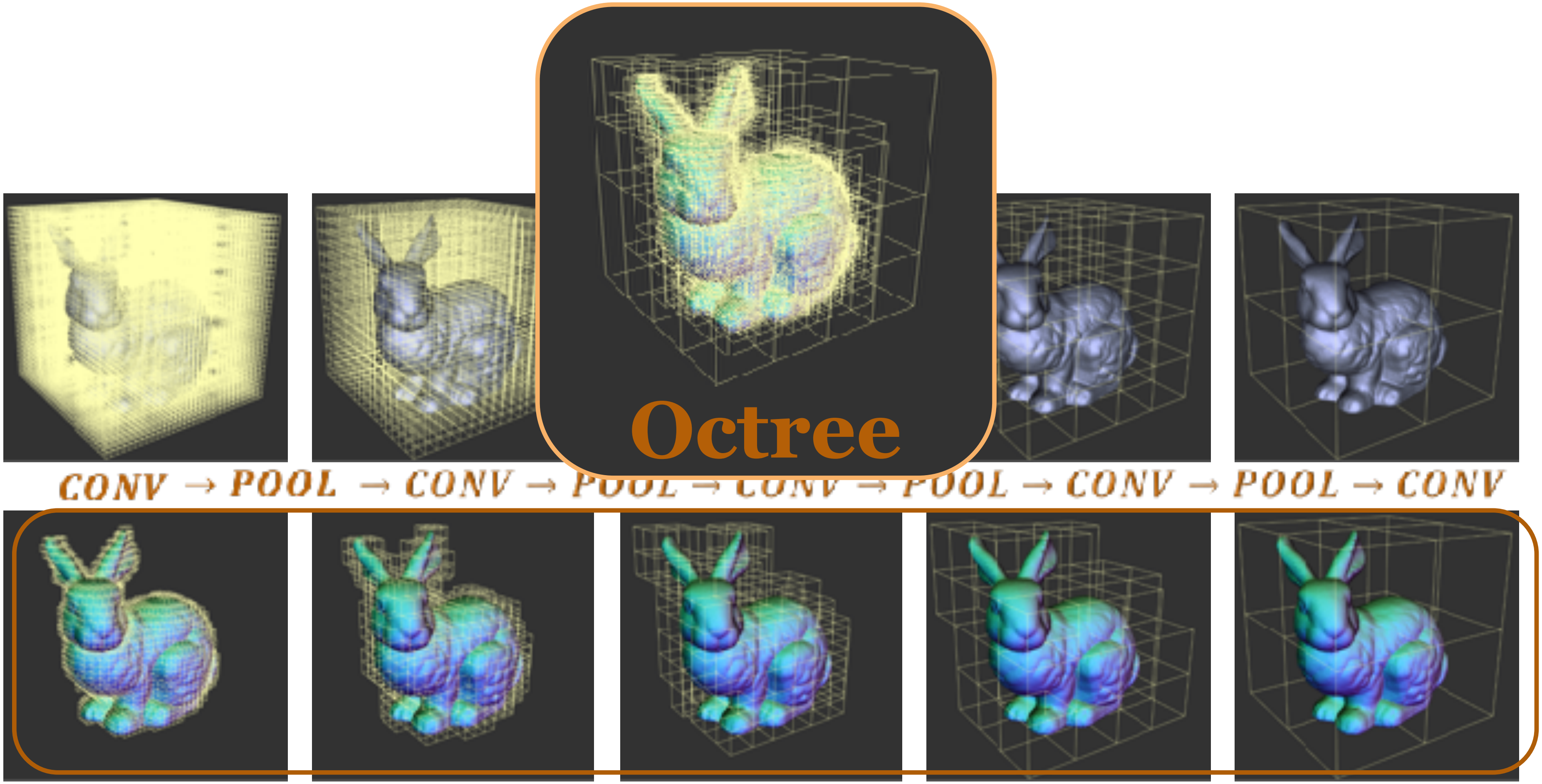


$\frac{\#occupied\ grid}{\#total\ grid}$

Occupancy:	10.41%	5.09%	2.41%
Resolution:	32	64	128

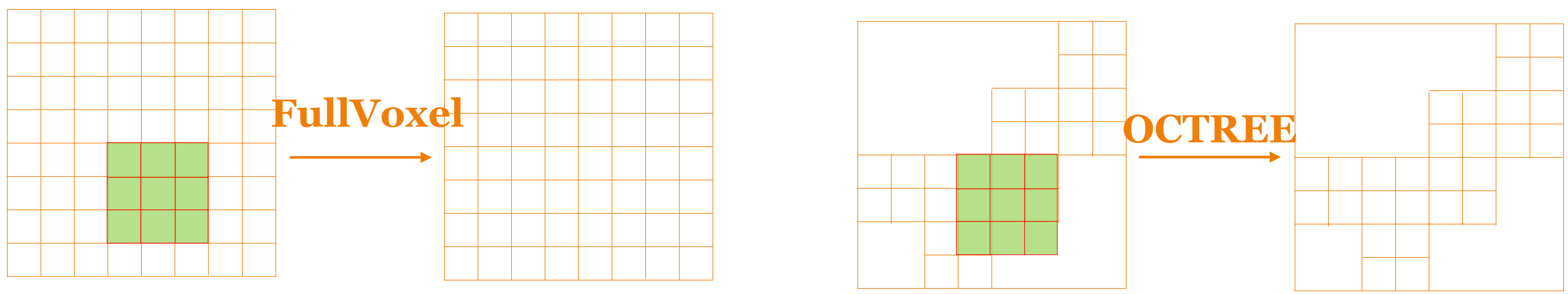
Li et, FPNN: Field Probing Neural Networks for 3D Data, NIPS 2016

Solution: Octree based CNN (O-CNN)



Convolution on Octree

- Neighborhood searching: Hash table

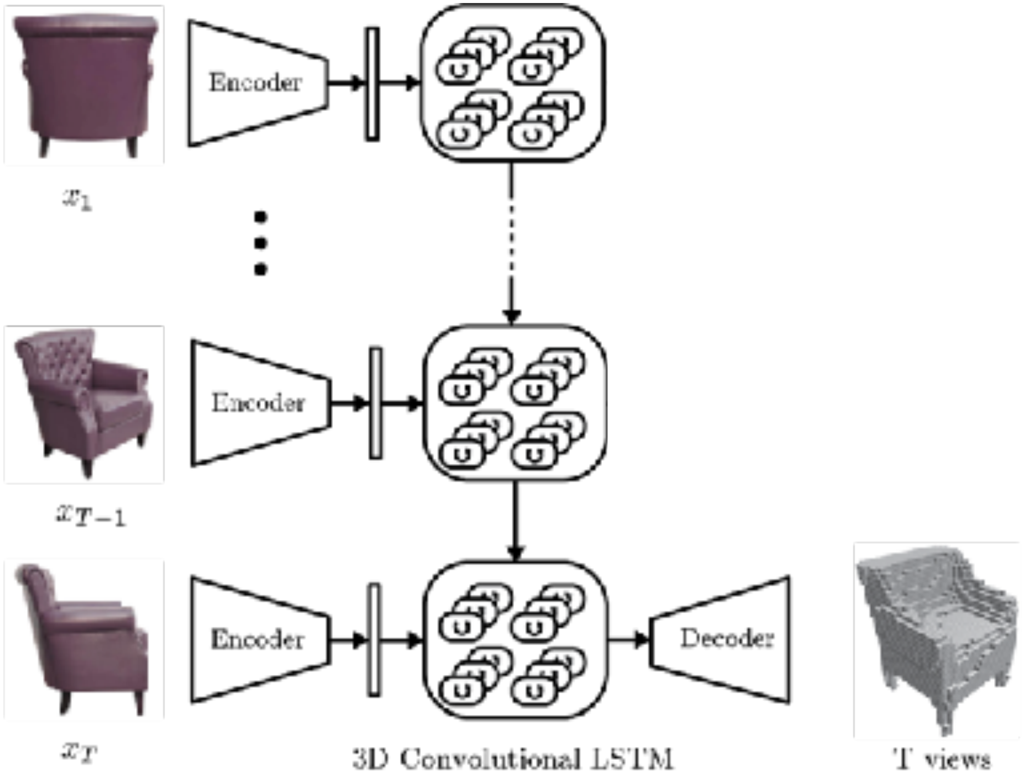


Gernot Riegler, Ali Osman Ulusoy, Andreas Geiger
“OctNet: Learning Deep 3D Representations at High Resolutions”
CVPR2017
Pengshuai Wwang, Yang Liu, Yuxiao Guo, Chunyu Sun, Xin Tong
“O-CNN: Octree-based Convolutional Neural Network for Understanding 3D Shapes”
SIGGRAPH2017

Volumetric Representation as 3D Input

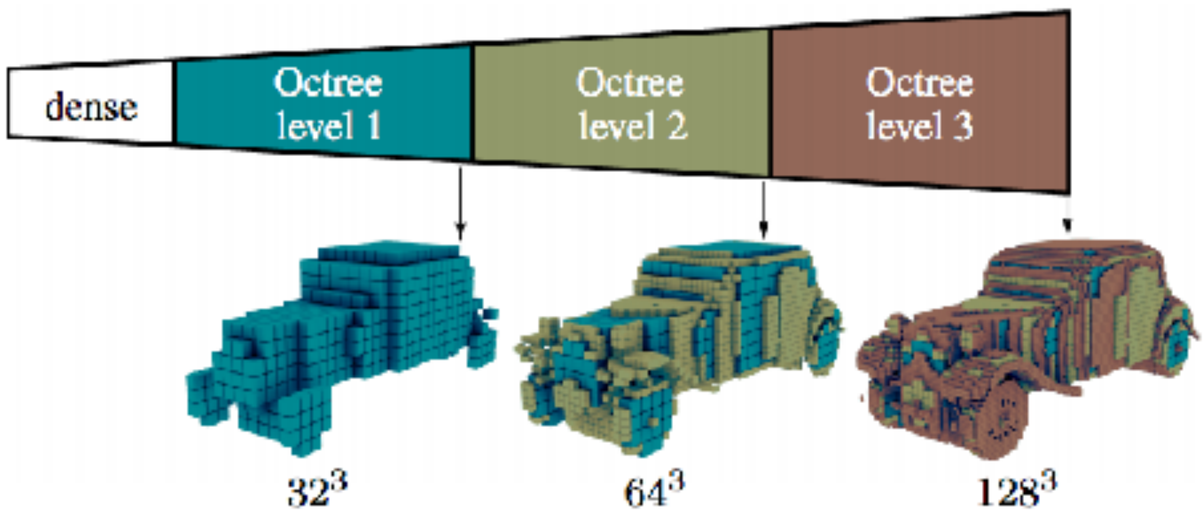
- The main hurdle is still Complexity

A Straight-forward Implementation



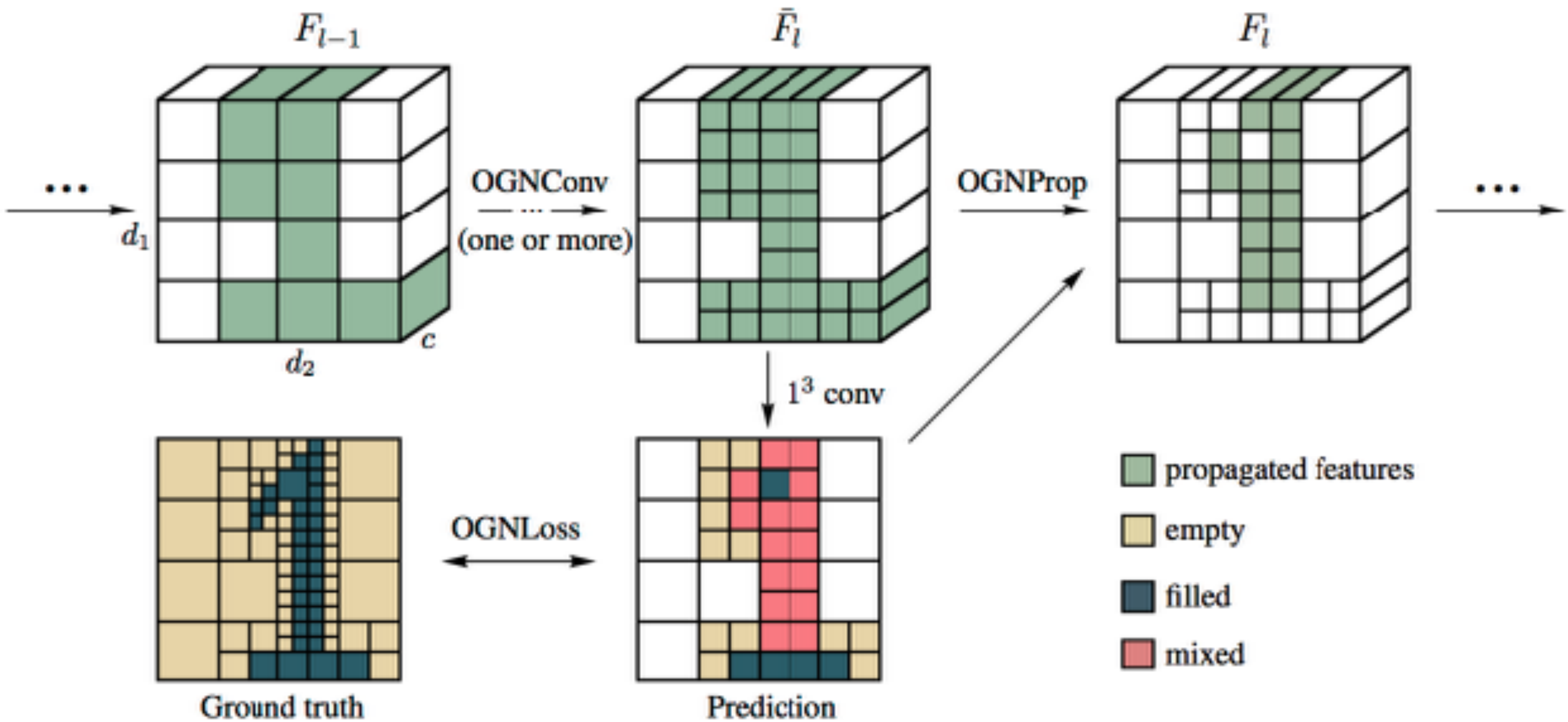
Choi et al. ECCV 2016

Towards Higher Spatial Resolution



Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox
“Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs”
arxiv (March, 2017)

Progressive Voxel Refinement



Fundamental Challenges of 3D Deep Learning

3D has many representations:

Rasterized form (regular grids)

multi-view RGB(D) images

volumetric

Geometric form (irregular)

Cannot directly apply CNN

polygonal mesh

point cloud

primitive-based models

Deep Learning on Polygonal Meshes

Mesh as 3D Input

- Deep Learning on Graphs

Geometry-aware Convolution can be Important

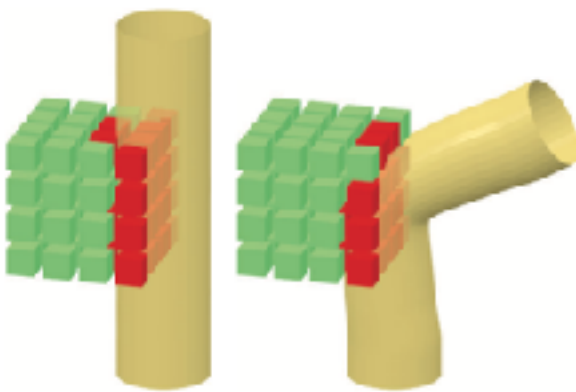
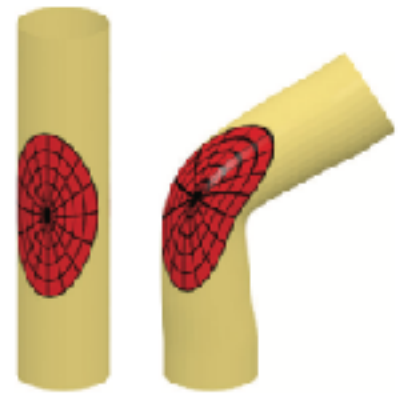


image credit: D. Boscaini, et al.

convolutional along spatial coordinates



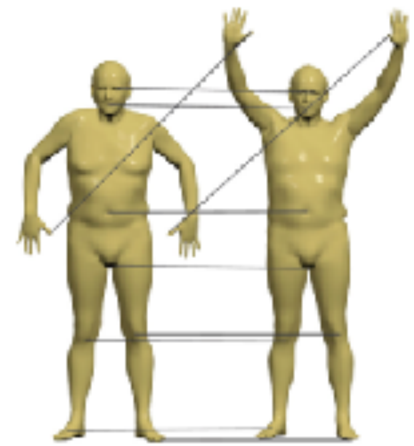
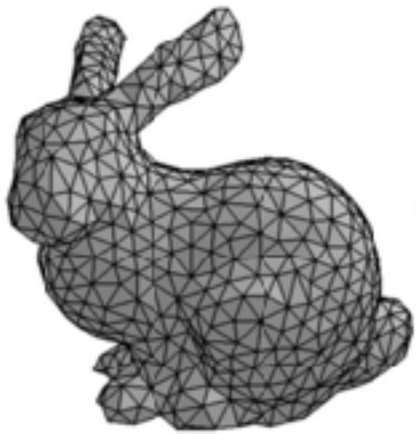


image credit: D. Boscaini, et al.

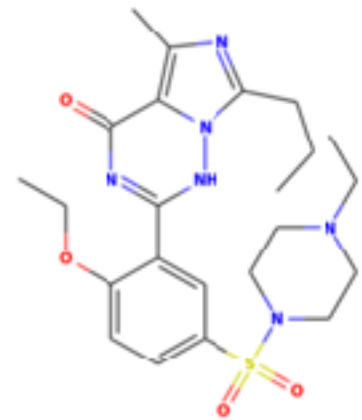
convolutional considering underlying geometry

Meshes can be represented as graphs



3D shape graph

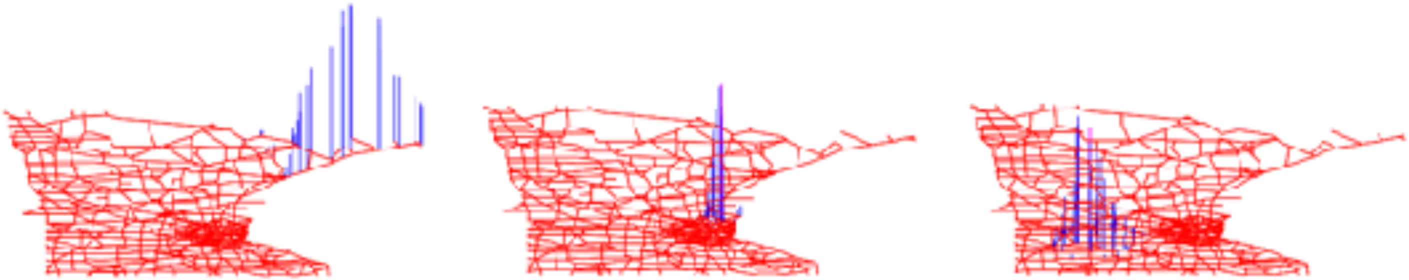
social network



molecules

How to define convolution kernel on graphs?

- Desired properties:
  - locally supported (w.r.t graph metric)
  - allowing weight sharing across different coordinates



from Shuman et al. 2013

Issues of Geodesic CNN

- The local charting method relies on a fast marching-like procedure requiring a triangular mesh.

- The radius of the geodesic patches must be sufficiently small to acquire a topological disk.

- No effective pooling, purely relying on convolutions to increase receptive field.

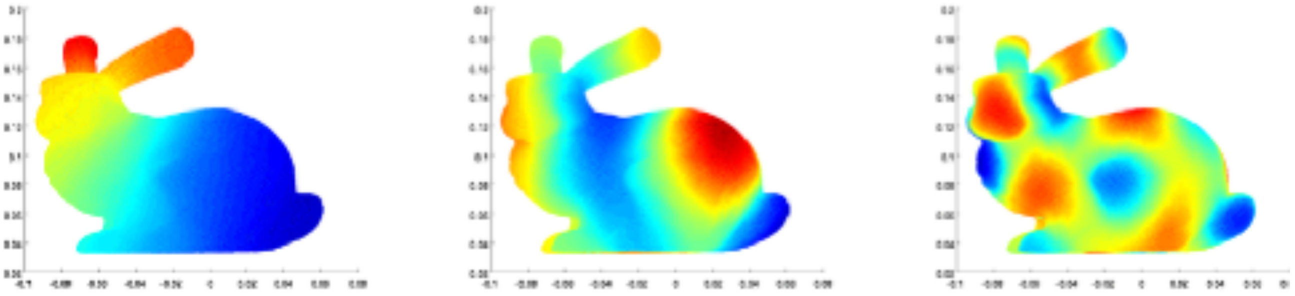
Spectral construction: Spectral CNN

Fourier analysis

Convert convolution to multiplication in spectral domain

Bases on meshes: eigenfunction of Laplacian-Bertrami operator

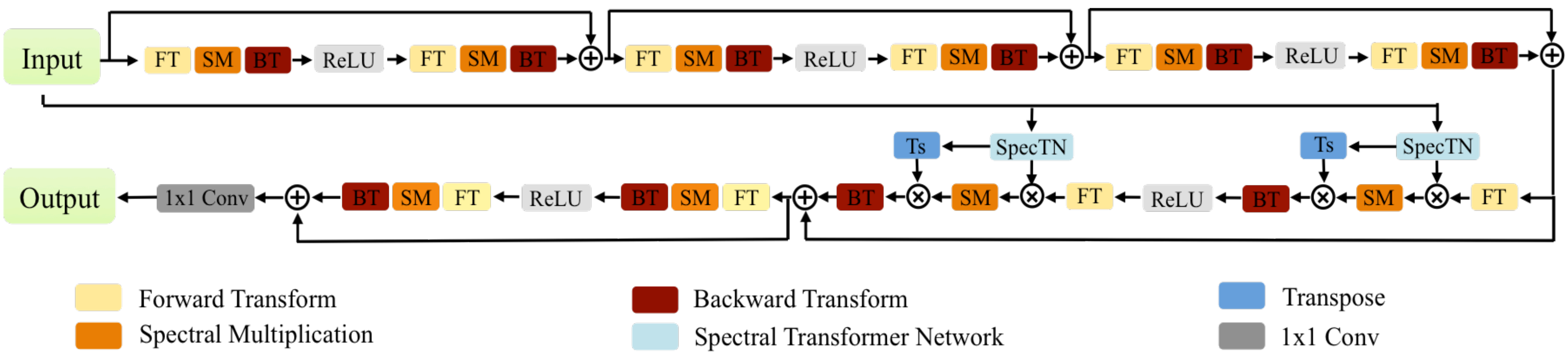
- "Fourier basis" of the graph: V : Eigenvectors of Δ



v_2 v_{10} v_{30}

Synchronization of functional space across meshes

Functional map



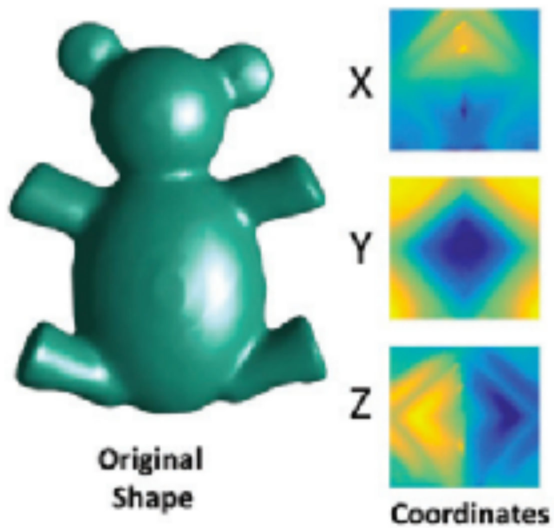
Li Yi, Hao Su, Xingwen Guo, Leonidas Guibas
"SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation"
CVPR2017 (spotlight)

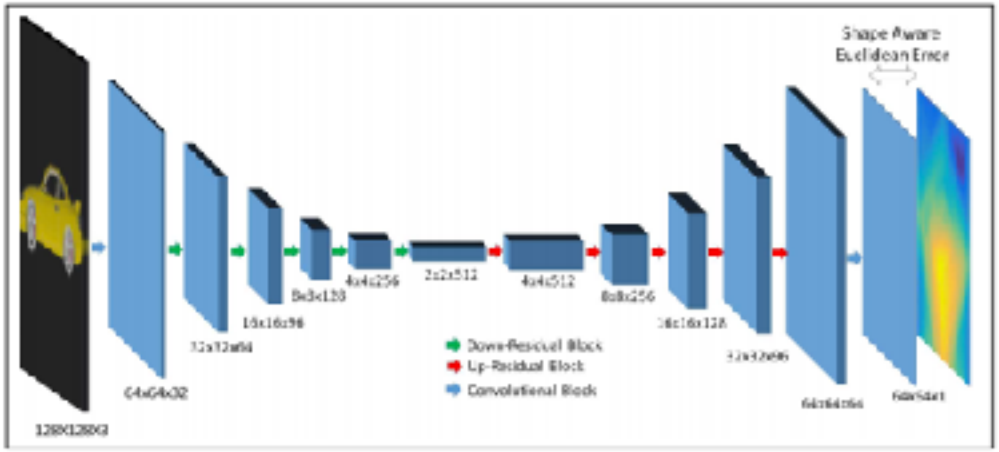
Mesh as 3D Output

- At the heart a surface parameterization problem

Deep learning on surface parameterization

Use CNN to predict the parameterization, then convert to 3D mesh





Step 1 Step 2

Ayan Sinha, Asim Unmesh, Qixing Huang, Karthik Ramani
“SurfNet: Generating 3D shape surfaces using deep residual networks”
CVPR2017

Deep Learning on Point Cloud Representation

Point Cloud: the Most Common Sensor Output

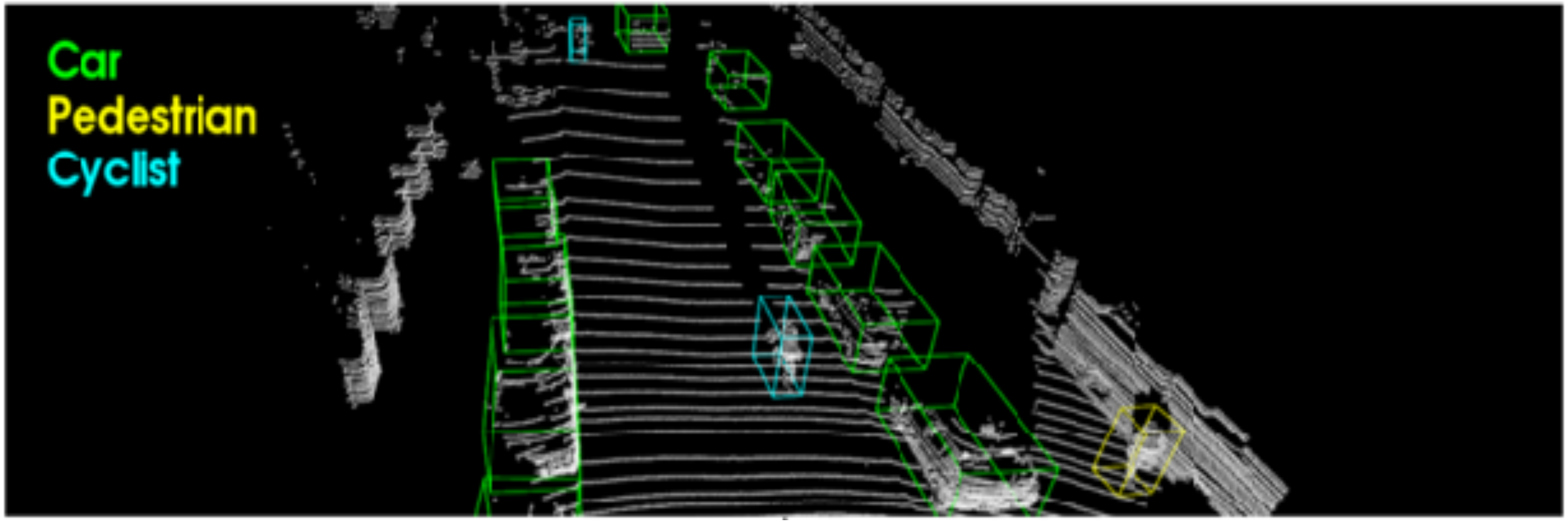
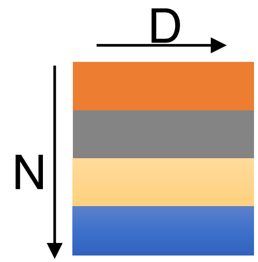


Figure from the recent VoxelNet paper from Apple.

Point Cloud as 3D Input

- Deep Learning on Sets (orderless)

Properties of a desired neural network on point clouds

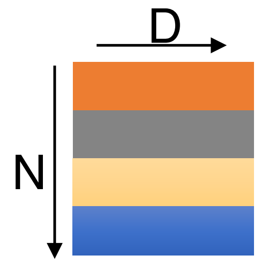


2D array representation

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

Hao Su, Charles Qi*, Kaichun Mo, Leonidas Guibas*
"PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation"
CVPR2017 (oral)

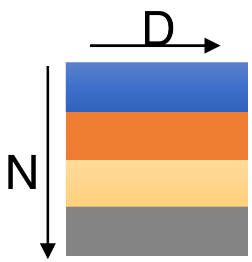
Properties of a desired neural network on point clouds



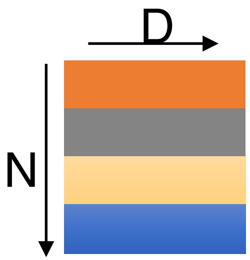
2D array representation

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

Properties of a desired neural network on point clouds



represents the same **set** as



2D array representation

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

Permutation invariance:

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

Examples:

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

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

$\ldots$

Construct symmetric function family

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

Construct symmetric function family

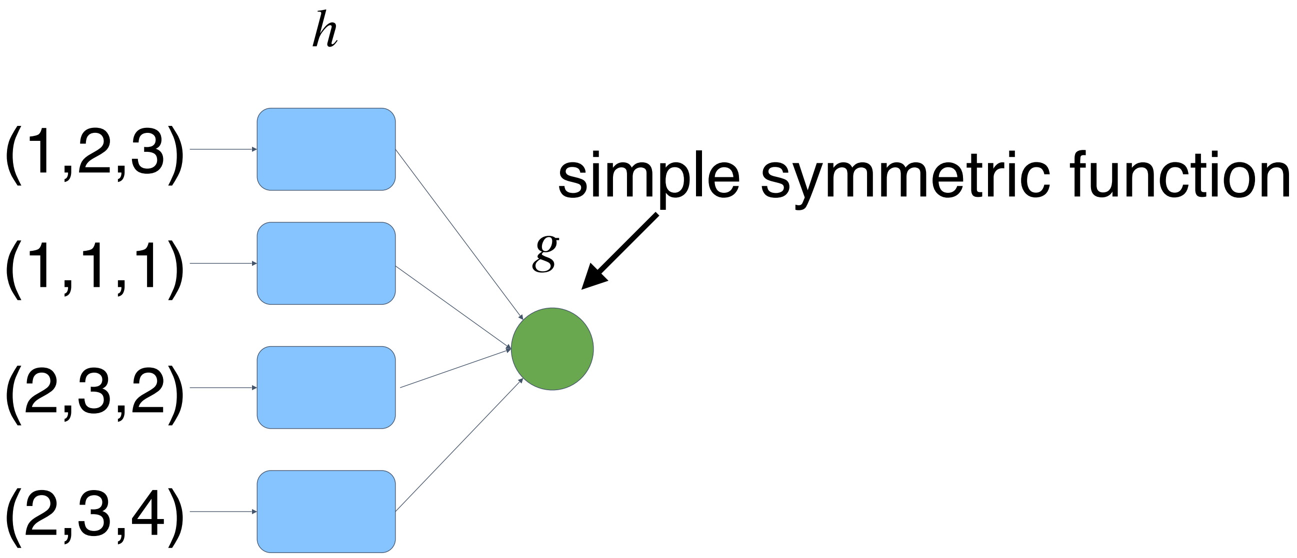
Observe:

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

Construct symmetric function family

Observe:

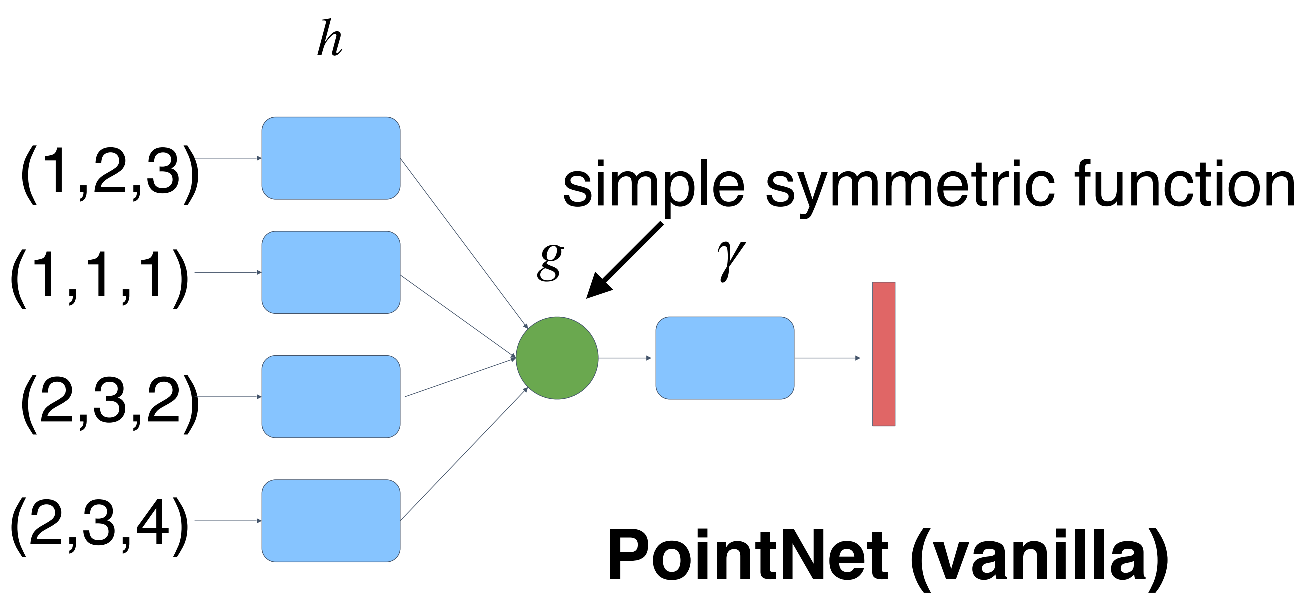
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



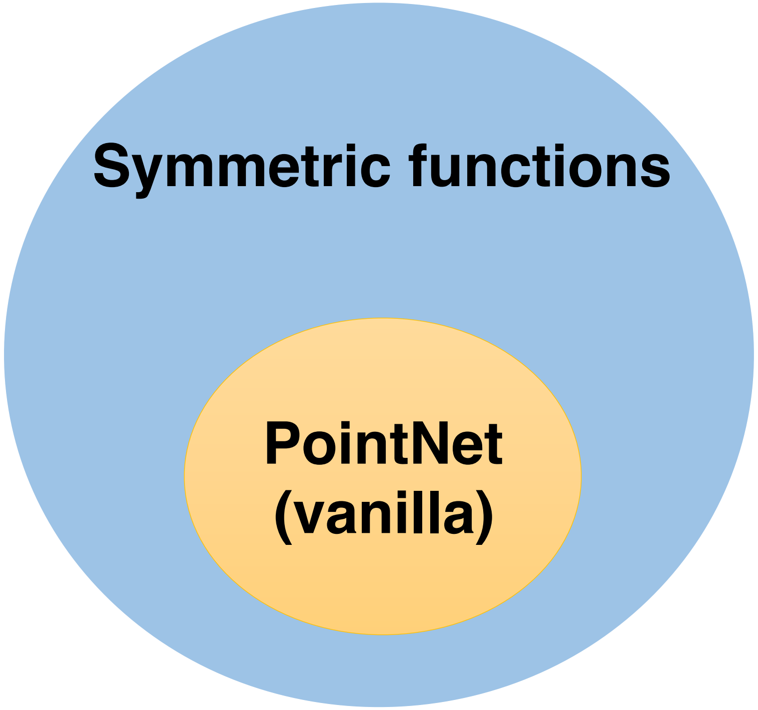
Construct symmetric function family

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



Q: What symmetric functions can be constructed by PointNet?



A: Universal approximation to continuous symmetric functions

Theorem:

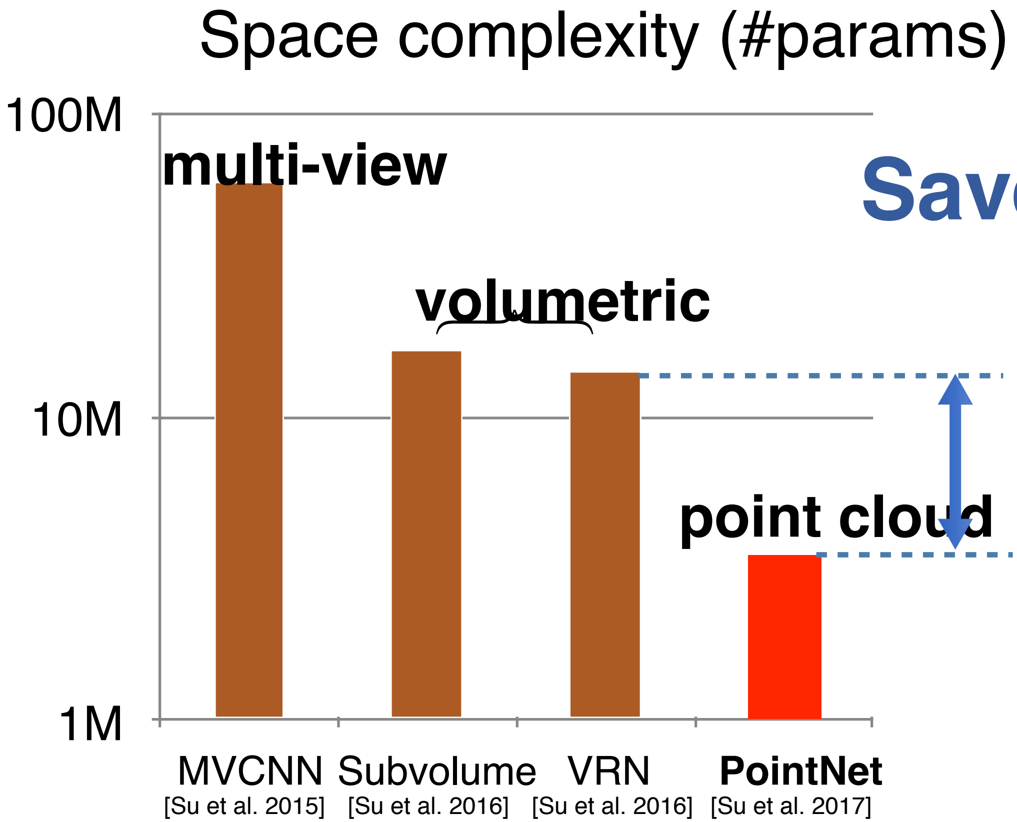
A Hausdorff continuous symmetric function $f : 2^{\mathcal{X}} \to \mathbb{R}$ can be arbitrarily approximated by PointNet.

$$\left| f(S) - \boxed{\gamma\left(\underset{x_i \in S}{\mathrm{MAX}}\left\{h(x_i)\right\}\right)} \right| < \epsilon$$

$S \subseteq \mathbb{R}^d$,

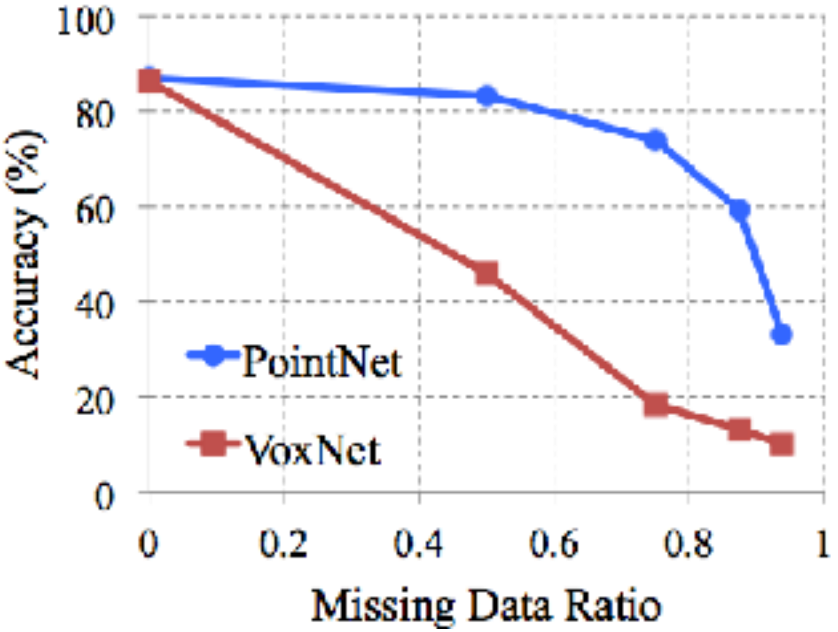
PointNet (vanilla)

PointNet is Light-weight

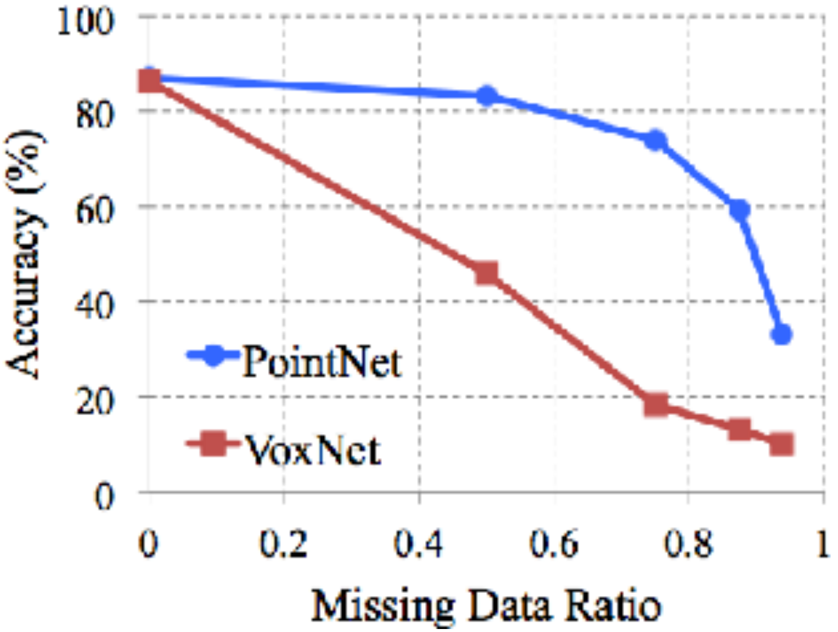


Saves 95% GPU memory

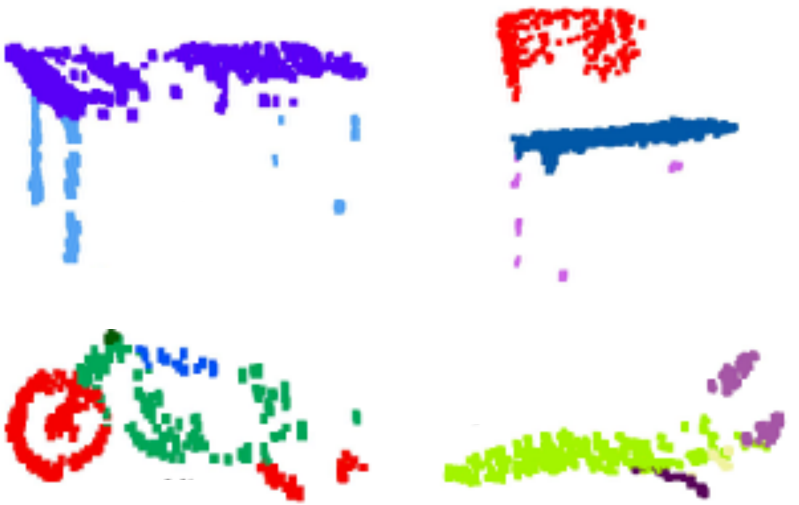
Robustness to data corruption



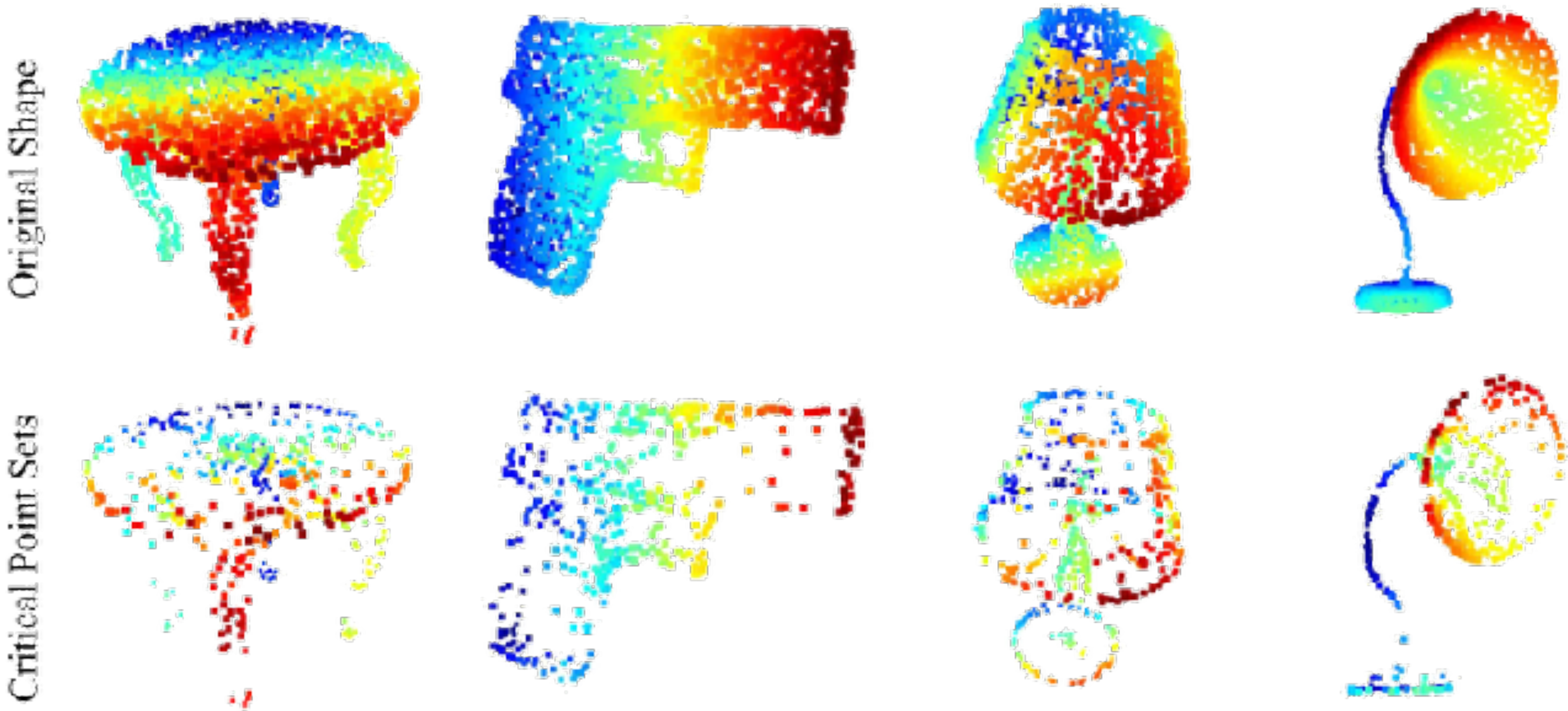
Robustness to data corruption



Segmentation from **partial scans**

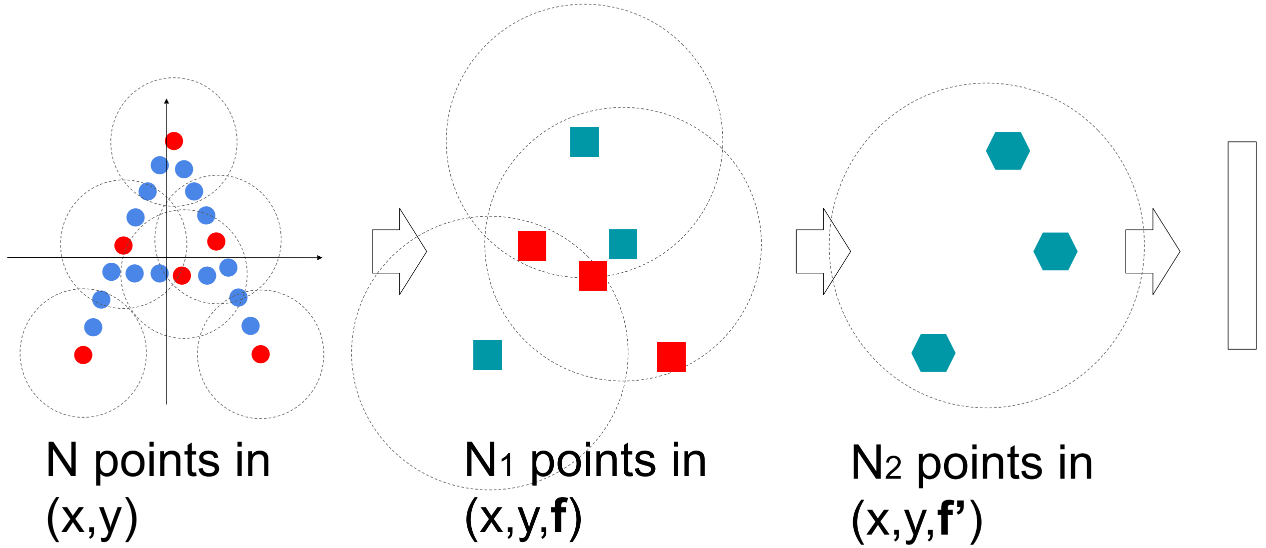


Visualize what is learned by reconstruction



Salient points are discovered!

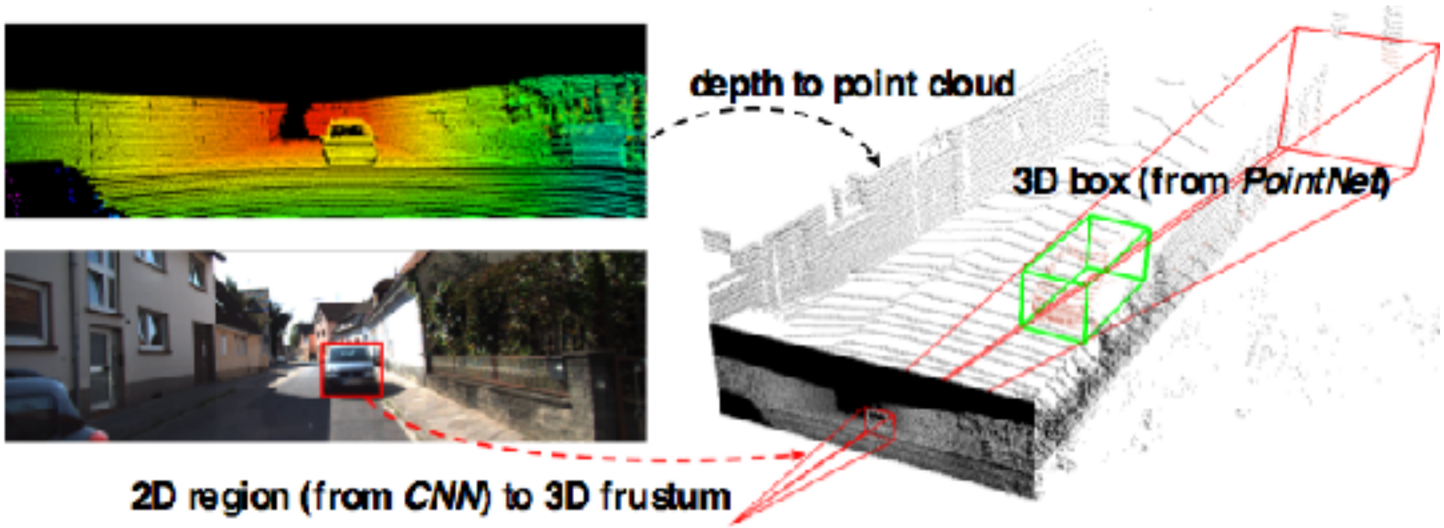
PointNet v2.0: Multi-Scale PointNet



1. Larger receptive field in higher layers ✓
2. Less points in higher layers (more scalable) ✓
3. Weight sharing ✓
4. Translation invariance (local coordinates in local regions) ✓

Charles Qi, Hao Su, Li Yi, Leonidas Guibas
"PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space"
NIPS 2017

Fuse 2D and 3D:
Frustum PointNets for 3D Object Detection



+ Leveraging mature 2D detectors for region proposal and 3D search space reduction
+ Solving 3D detection problem with 3D data and 3D deep learning architectures

My latest paper accepted at CVPR 2018

Our method ranks No. 1 on KITTI 3D Object Detection Benchmark

We get 5% higher AP than Apple's recent CVPR submission and more than 10% higher AP than previous SOTA in easy category

Car

	Method	Setting	Code	Moderate	Easy	Hard	Runtime	Environment	Compare
1	F-PointNet			70.39 %	81.20 %	62.19 %	0.17 s	GPU @ 3.0 Ghz (Python)	
2	VxNet(LIDAR)			65.11 %	77.47 %	57.73 %	0.23 s	GPU @ 2.5 Ghz (Python + C/C++)	
3	AVOD			65.02 %	75.48 %	57.87 %	0.08 s	Titan X (pascal)	
4	MV3D			62.35 %	71.09 %	55.12 %	0.36 s	GPU @ 2.5 Ghz (Python + C/C++)	
X. Chen, H. Ma, J. Wan, B. Li and T. Xia: Multi-View 3D Object Detection Network for Autonomous Driving. CVPR 2017.									
5	MV3D (LIDAR)			52.73 %	66.77 %	51.31 %	0.24 s	GPU @ 2.5 Ghz (Python + C/C++)	
X. Chen, H. Ma, J. Wan, B. Li and T. Xia: Multi-View 3D Object Detection Network for Autonomous Driving. CVPR 2017.									
6	F-PC_CNN			42.67 %	50.46 %	40.15 %	0.5 s	GPU @ 3.0 Ghz (Matlab + C/C++)	
7	SDN			21.36 %	34.05 %	18.59 %	0.07 s	GPU @ 1.5 Ghz (Python)	
8	LMNetV2			15.24 %	14.75 %	12.85 %	0.02 s	GPU @ 2.5 Ghz (C/C++)	
9	3dSSD			14.97 %	14.71 %	19.43 %	0.03 s	GPU @ 2.5 Ghz (Python + C/C++)	
10	LMnet			9.19 %	11.32 %	9.19 %	0.1 s	GPU @ 1.1 Ghz (Python + C/C++)	

⋮

Our method ranks No. 1 on KITTI 3D Object Detection Benchmark

We are also 1st place for smaller objects (ped. and cyclist) winning with even bigger margins.

Pedestrian

	Method	Setting	Code	Moderate	Easy	Hard	Runtime	Environment	Compare
1	F-PointNet			44.89 %	51.21 %	40.23 %	0.17 s	GPU @ 3.0 Ghz (Python)	
2	VxNet(LiDAR)			33.69 %	39.48 %	31.51 %	0.23 s	GPU @ 2.5 Ghz (Python + C/C++)	
3	AVOD			25.87 %	32.67 %	25.01 %	0.08 s	Titan X (pascal)	
4	3dSSD			17.35 %	20.22 %	17.20 %	0.03 s	GPU @ 2.5 Ghz (Python + C/C++)	

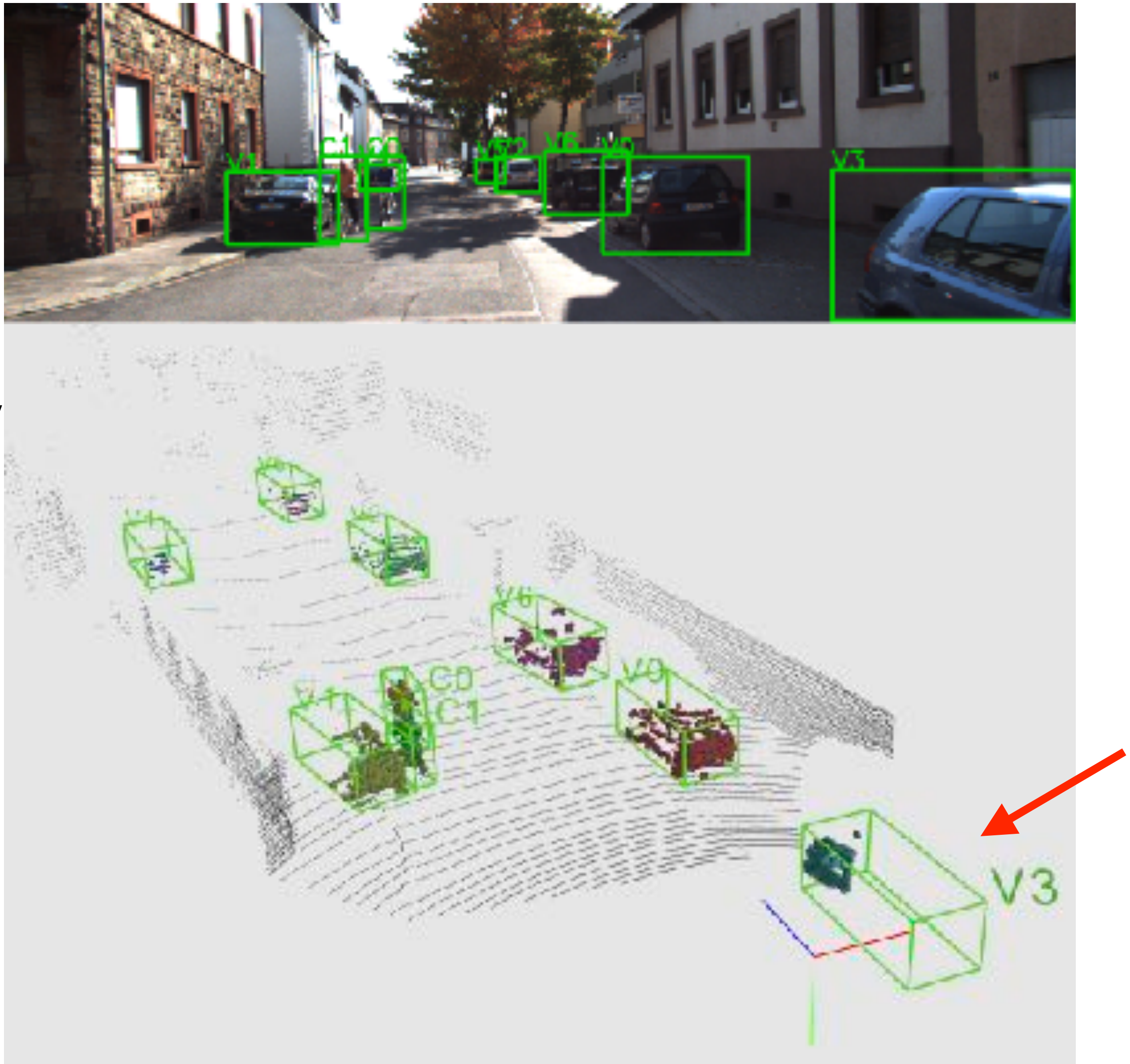
⋮

Cyclist

	Method	Setting	Code	Moderate	Easy	Hard	Runtime	Environment	Compare
1	F-PointNet			56.77 %	71.96 %	50.39 %	0.17 s	GPU @ 3.0 Ghz (Python)	
2	VxNet(LiDAR)			48.36 %	61.22 %	44.37 %	0.23 s	GPU @ 2.5 Ghz (Python + C/C++)	
3	AVOD			30.43 %	43.74 %	30.12 %	0.08 s	Titan X (pascal)	

⋮

Remarkable box estimation accuracy even with a dozen of points or with very partial point cloud



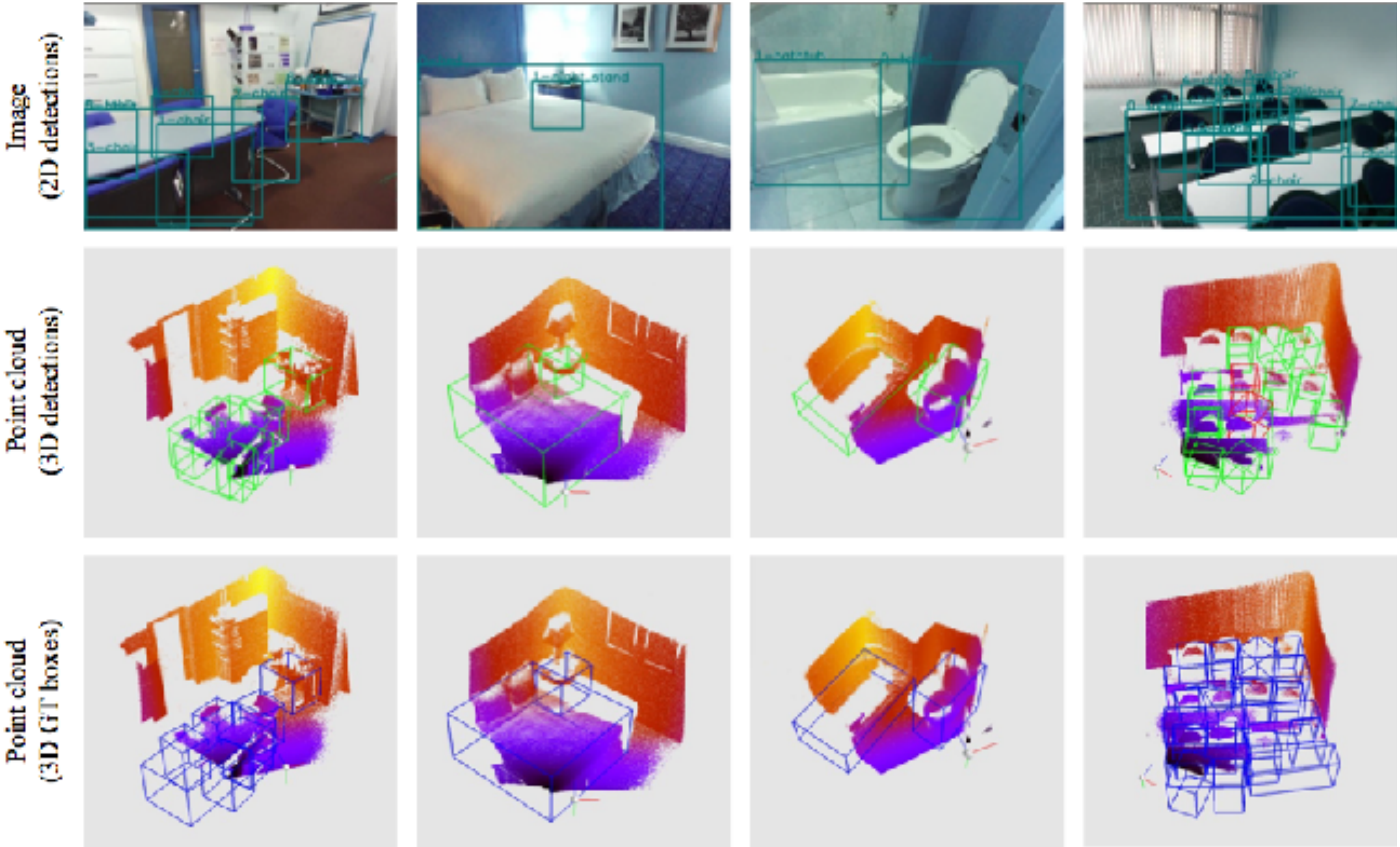
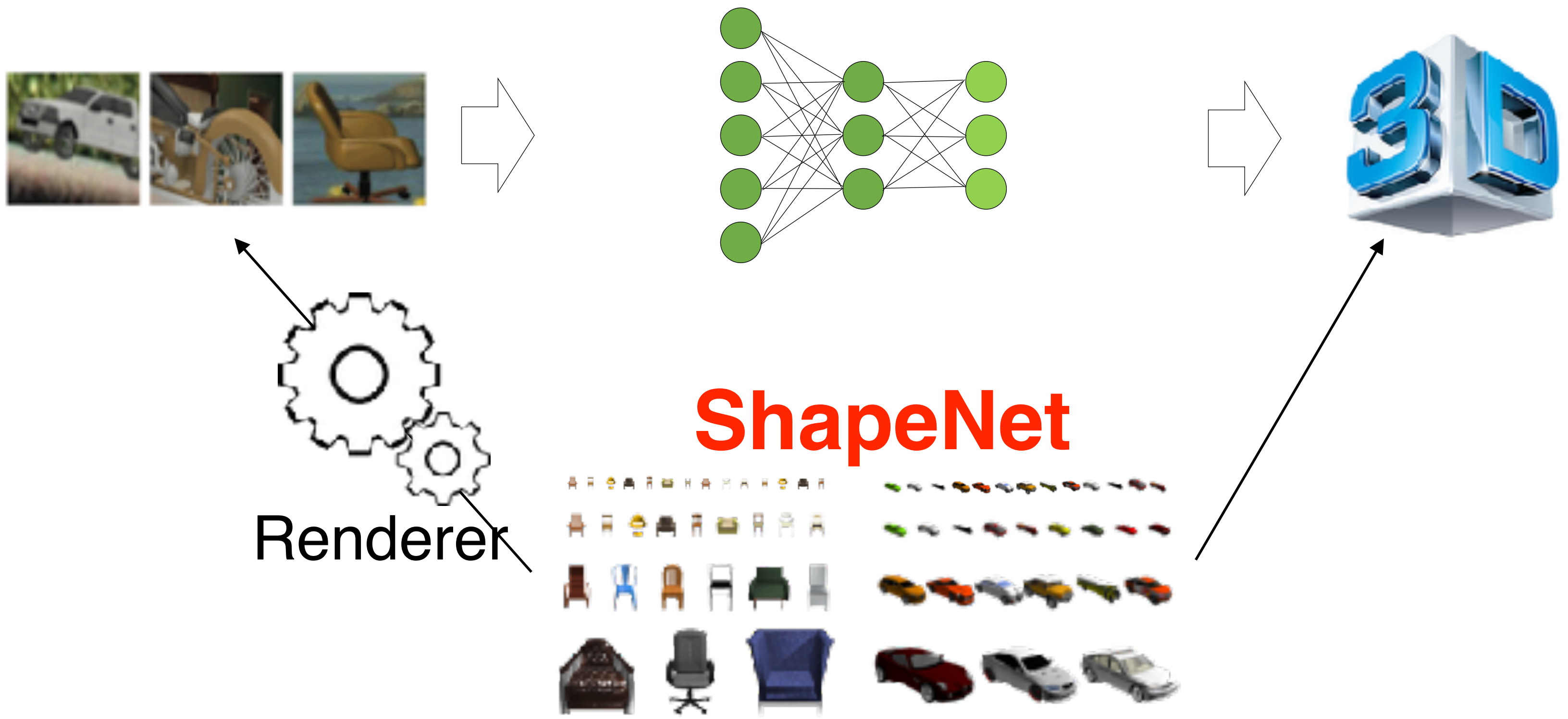


Image
(2D detections)
Point cloud
(3D detections)
Point cloud
(3D GT boxes)

Point Cloud as 3D Output

- Deep Learning to Generate Combinatorial Objects

Supervision from “Synthesize for Learning”



3D Representation: Point Cloud

✓ Describe shape for the whole object

? Usable as **network output?**

No prior works in the deep learning community!

3D Prediction by Point Clouds

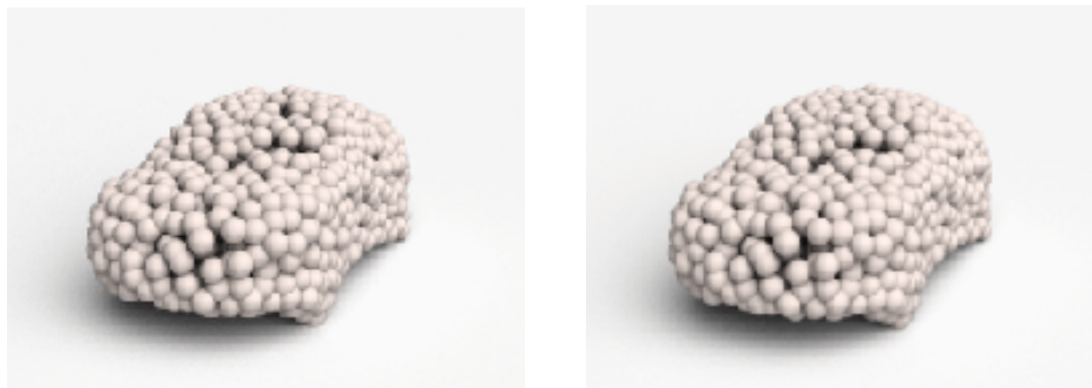
Input

Reconstructed 3D point cloud

Hao Su, Haoqiang Fan, Leonidas Guibas
"A Point Set Generation Network for 3D Object Reconstruction from a Single Image"
CVPR2017 (oral)

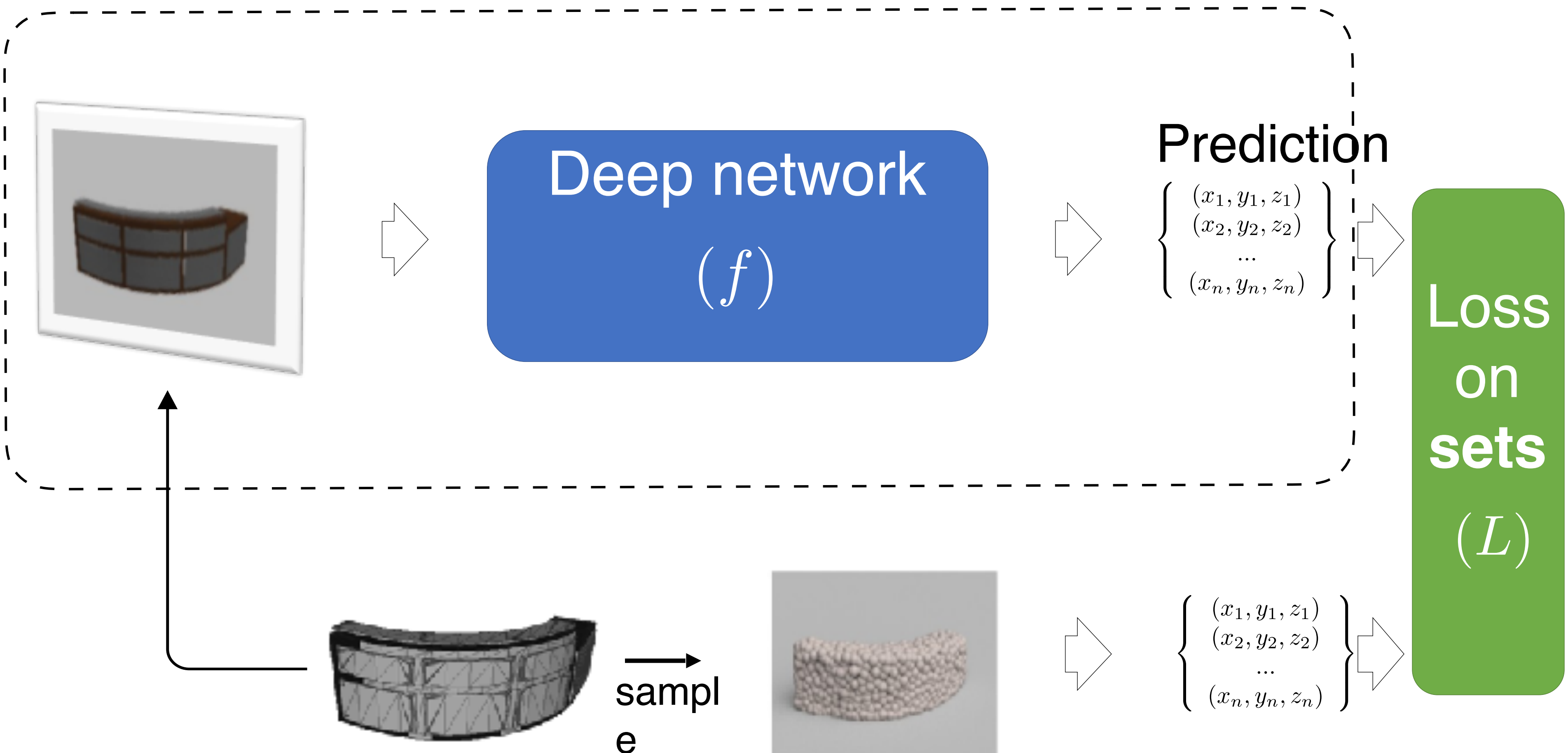
3D Prediction by Point Clouds

Input



Reconstructed 3D point cloud

Pipeline



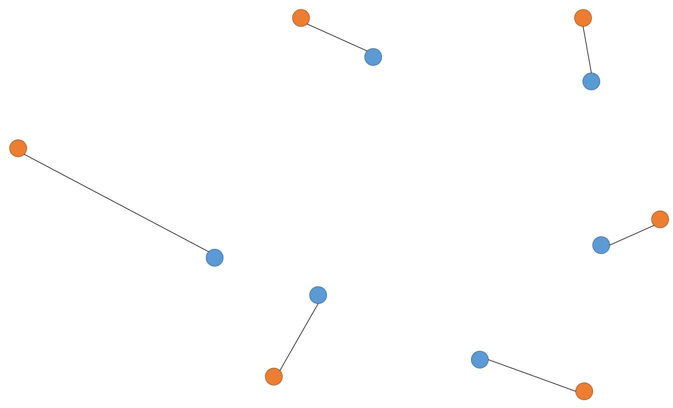
CVPR '17, Point Set Generation

Loss function: Earth Mover's Distance (EMD)

- Given two sets of points, measure their discrepancy:

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

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



Differentiable **Admit fast computation**

Generalization to Unseen Categories

input observed view 90° input observed view 90°

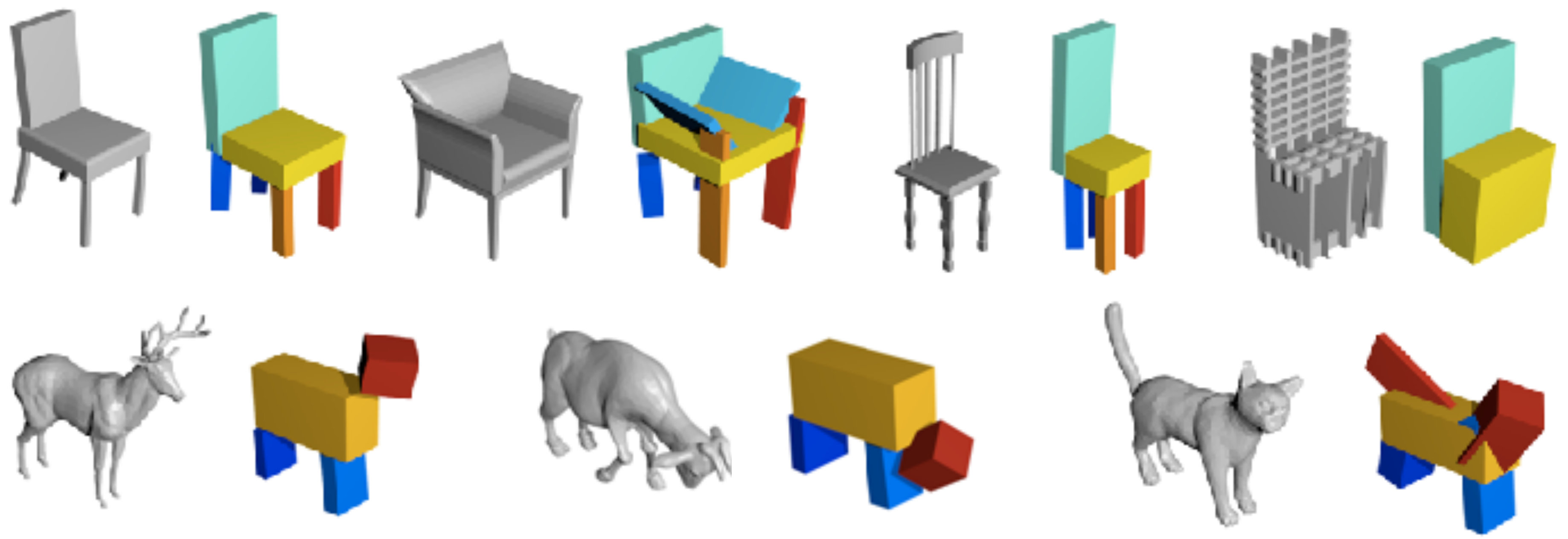
Out of training

Deep Learning on Primitives

Describe Shapes by Primitives

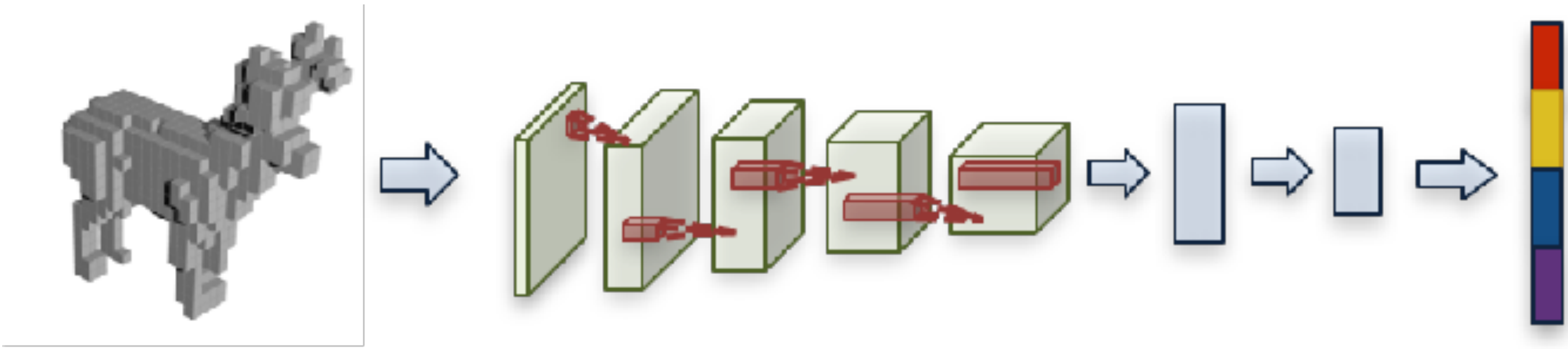
- What are parts? Reusable substructures!
- A Structure Mining Problem
- By DL, also a Meta-Learning Problem

Primitive-based Assembly



Shubham Tulsiani, Hao Su, Leonidas Guibas, Alexei A. Efros, Jitendra Malik
Learning Shape Abstractions by Assembling Volumetric Primitives
CVPR 2017

Approach



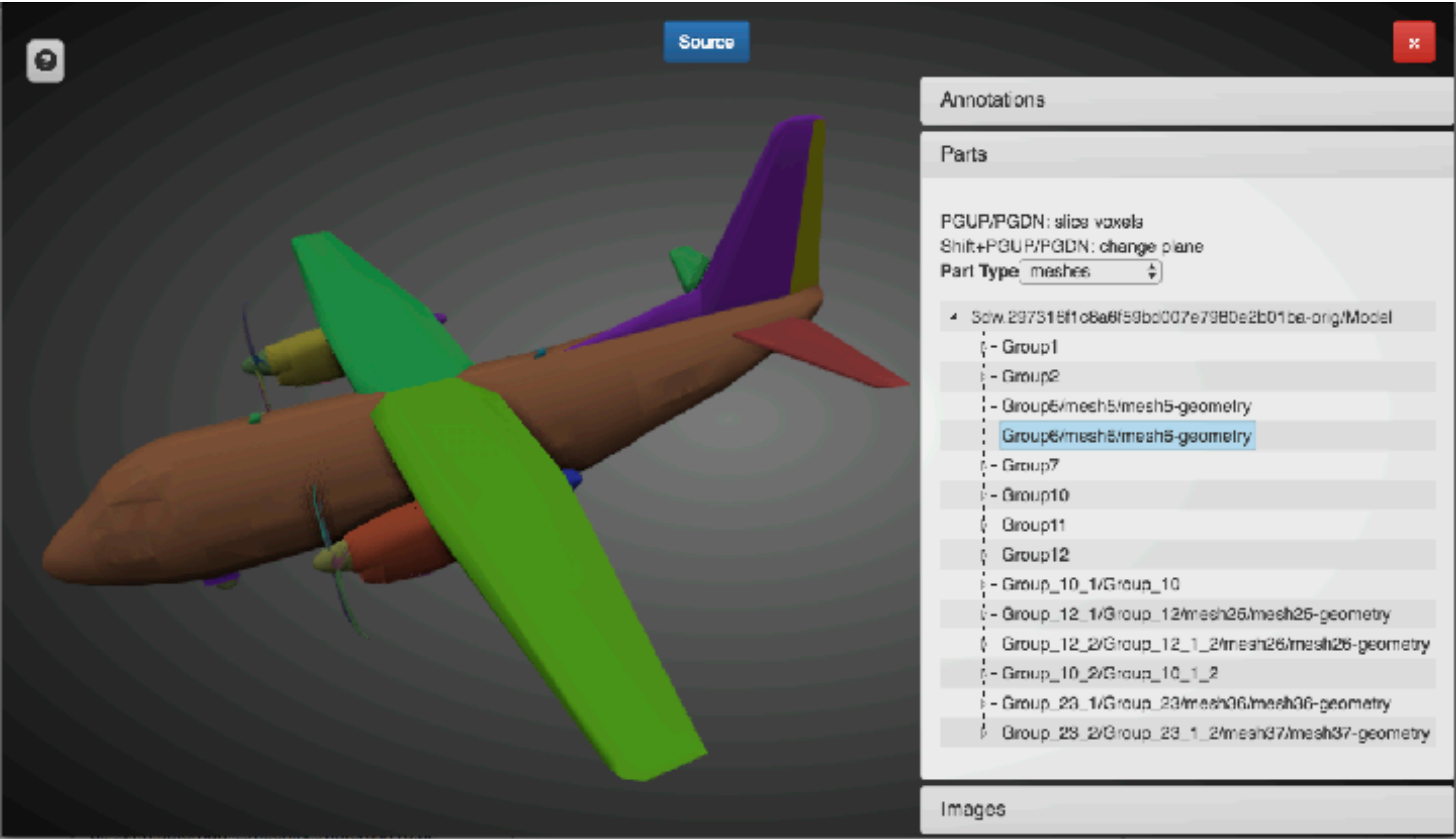
We predict primitive parameters: size, rotation, translation of M cuboids.

Variable number of parts? We predict “primitive existence probability”

Generative Models for Shapes by Reusing Primitives

- Incremental Assembly-based modeling
- "Transfer Learning" in the sense of reusing prior knowledge

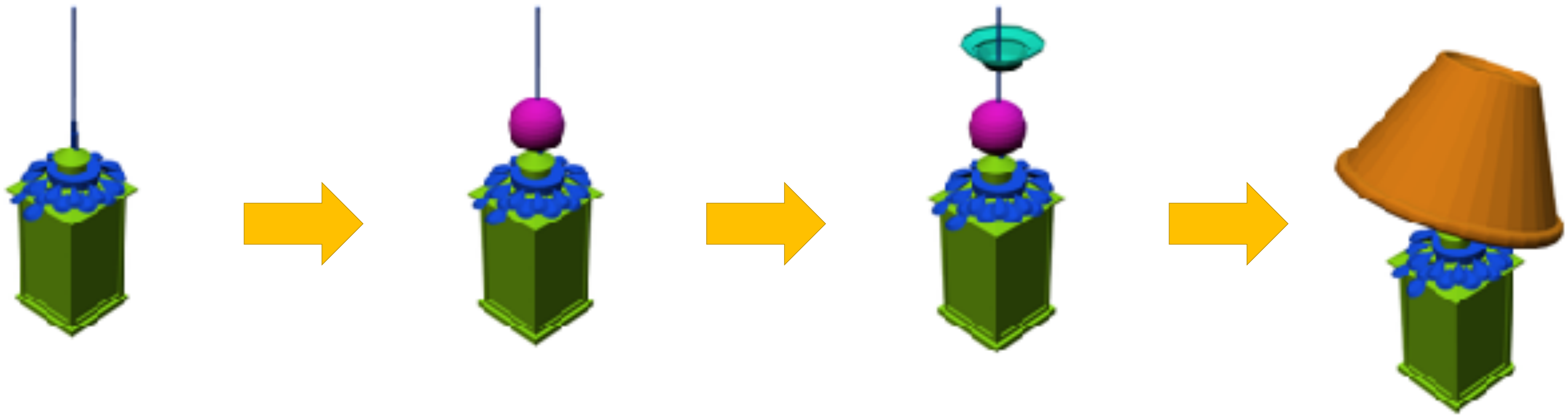
Primitive Space from ShapeNet Parts



Markov Modeling Process

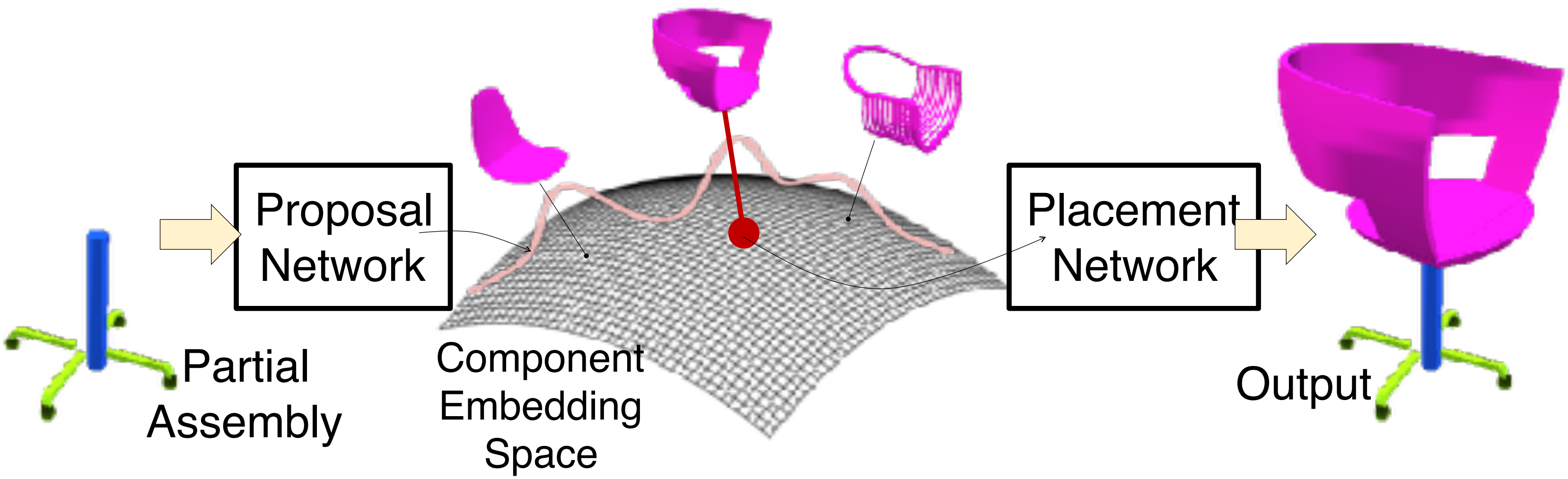
Part assembly:

Markov process – *Incrementally* assemble parts.



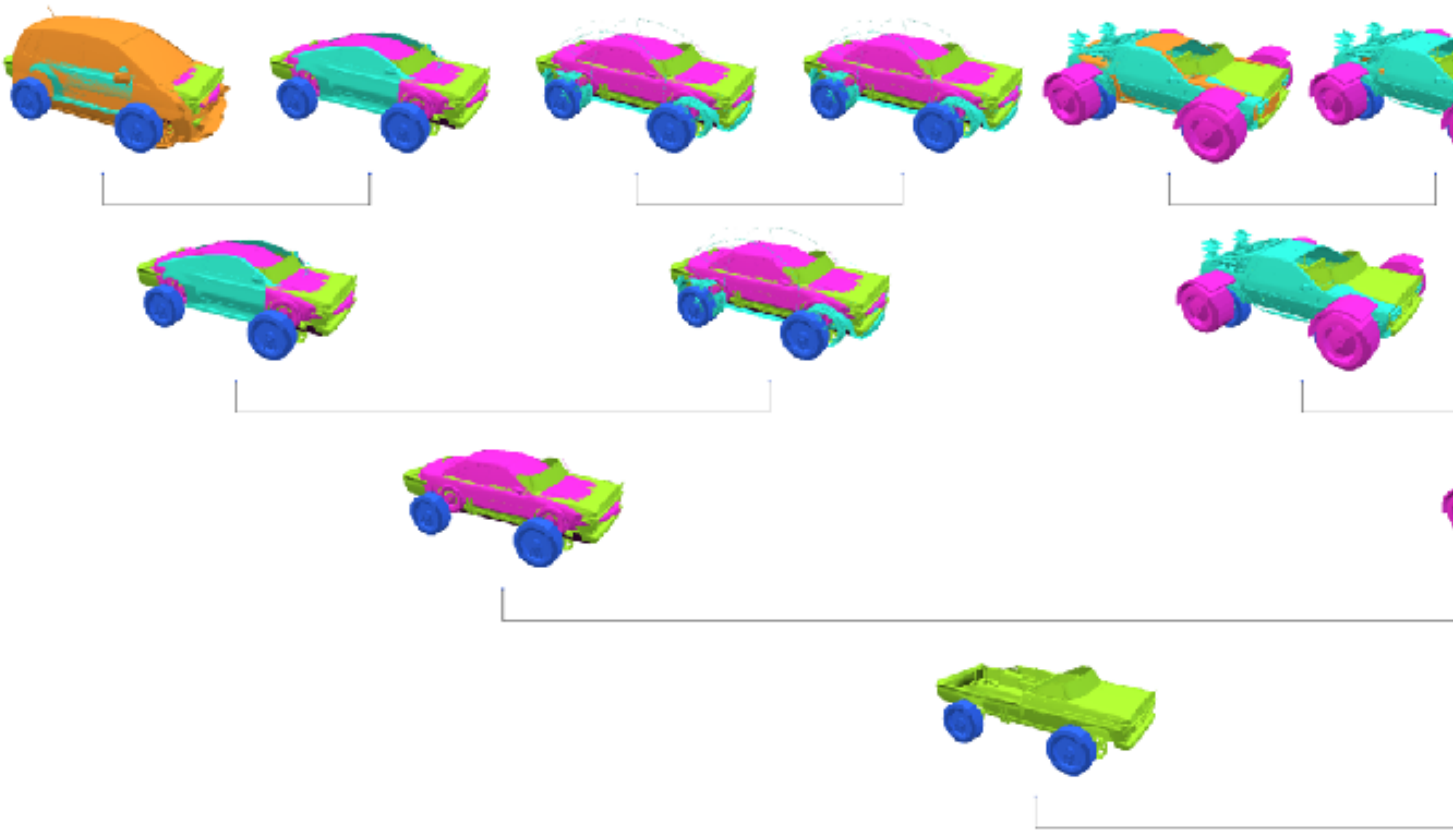
Sung et al, ComplementMe: Weakly-Supervised Component Suggestions for 3D Modeling SIGGRAPH Asia 2017

New part proposal by network



Automatic Shape Synthesis

Automatic Shape Synthesis




ASIMO
HONDA

Thank you!