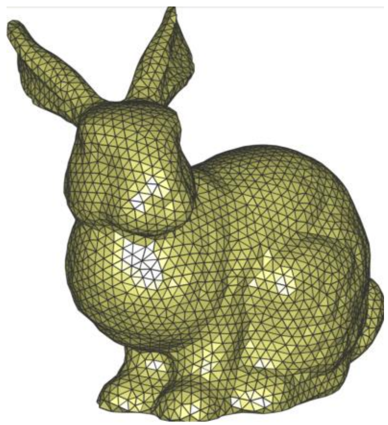
Data-Driven Geometry Processing
3D Deep Learning II

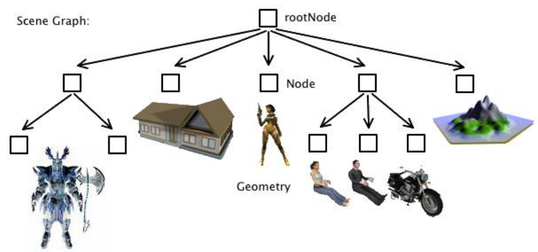
Qixing Huang

March 28th 2017

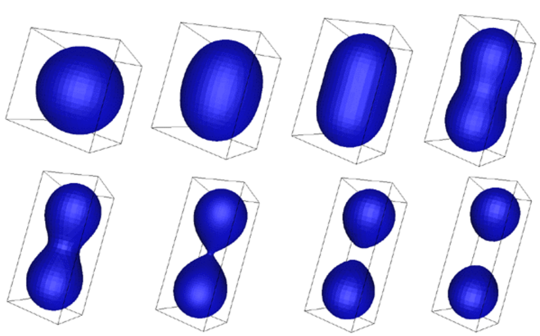
3D Surface Representations



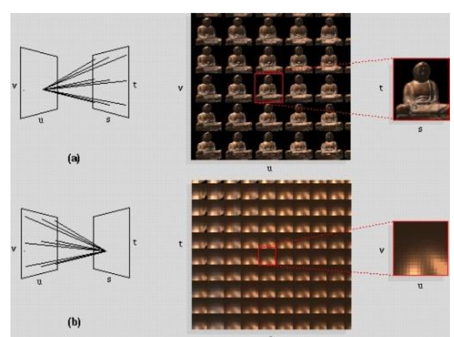
Triangular mesh



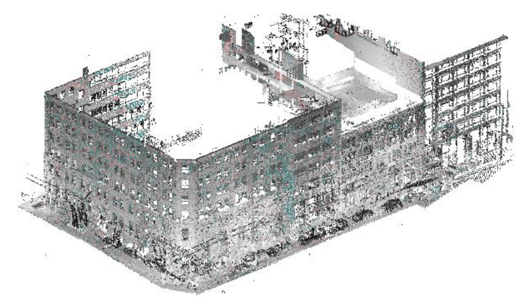
Part-based models



Implicit surface



Light Field Representation

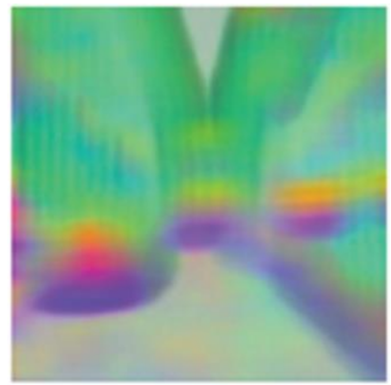


Point cloud

Matching in Embedding Spaces [CVPR' 16]

Existing methods usually follow a two-step approach (e.g., SIFT flow)

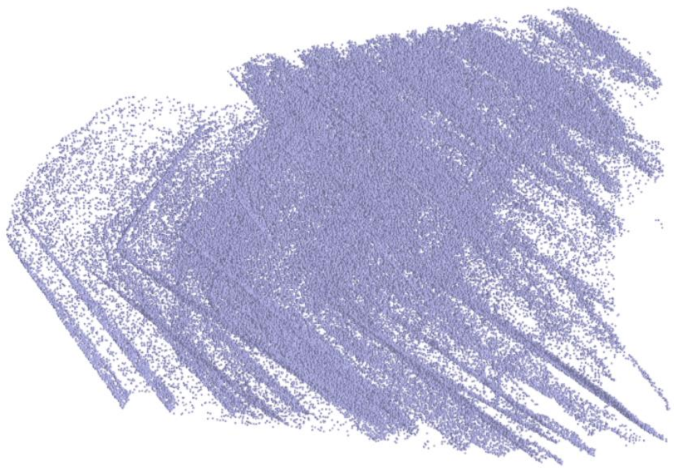
- Local descriptor computation

- Dense pixel labeling via MRF inference
  - Preserve descriptors
  - Preserve smoothness

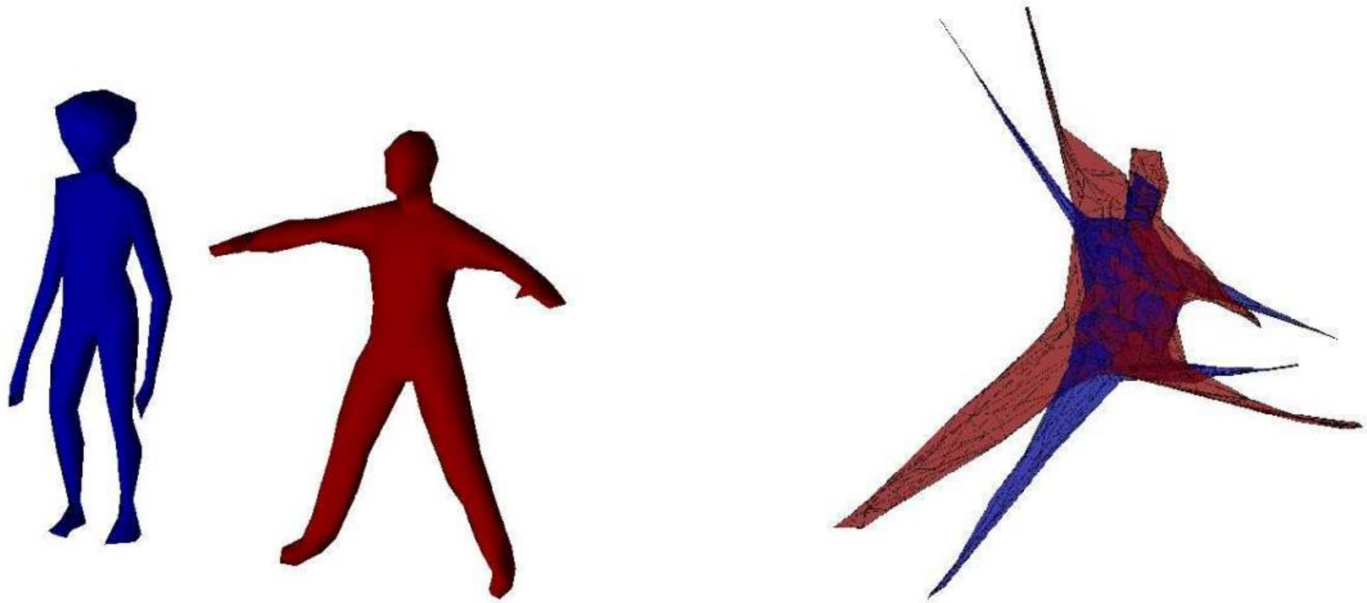
Issues of such two-step approach

Partial similarity



Inefficient when
matching multiple objects

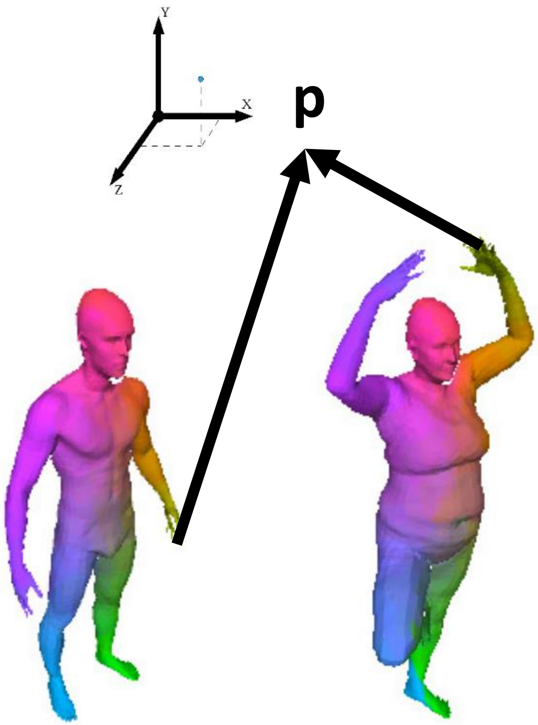
Embedding --- establishing correspondences in the embedding space



Spectral embedding [Liu et al. 06]

Sensitive to 1) partial similarity, and 2) geometric and topological changes

Properties of the desired embedding space



Corresponding points are matched in the embedding space

Embedding preserves continuity

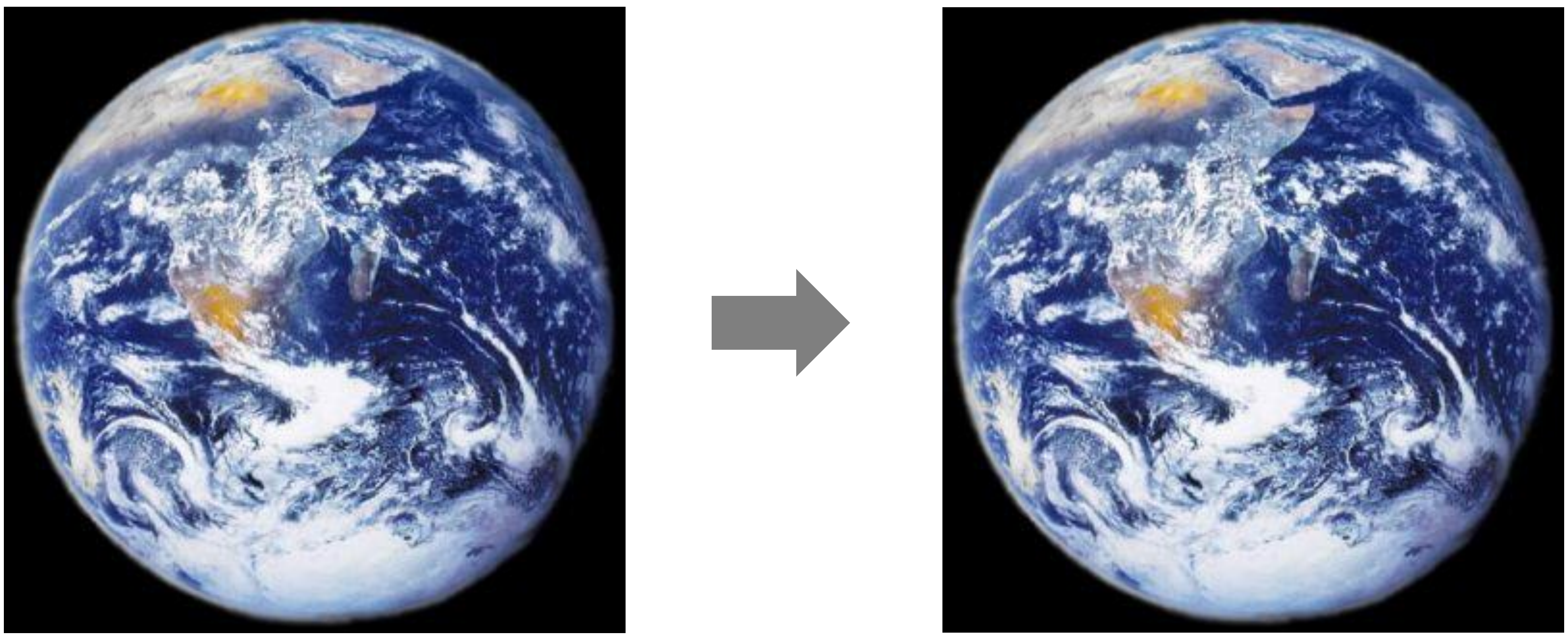
The benefits of object embedding

- Correspondences become nearest neighbor query
  - Efficiency for multiple object matching

    O(n) embeddings + O(n^2) queries
  - Partial similarity
  - Fuzzy correspondences

The biggest message of deep neural networks

- Approximate any function given sufficient data



Focus on depth images

- Scanning devices generate depth images

- Complete shape embedding are aggregated from depth image embeddings
  - 3D convolution is not ready yet

Architecture

	0	1	2	3	4	5	6	7	8	9	10
layer	**image**	**conv**	**max**	**conv**	**max**	**2×conv**	**conv**	**max**	**2×conv**	**int**	**conv**
filter-stride	-	11-4	3-2	5-1	3-2	3-1	3-1	3-2	1-1	-	3-1
channel	1	96	96	256	256	384	256	256	4096	4096	16
activation	-	relu	lrn	relu	lrn	relu	relu	idn	relu	idn	relu
size	512	128	64	64	32	32	32	16	16	128	512
num	1	1	4	4	16	16	16	64	64	1	1

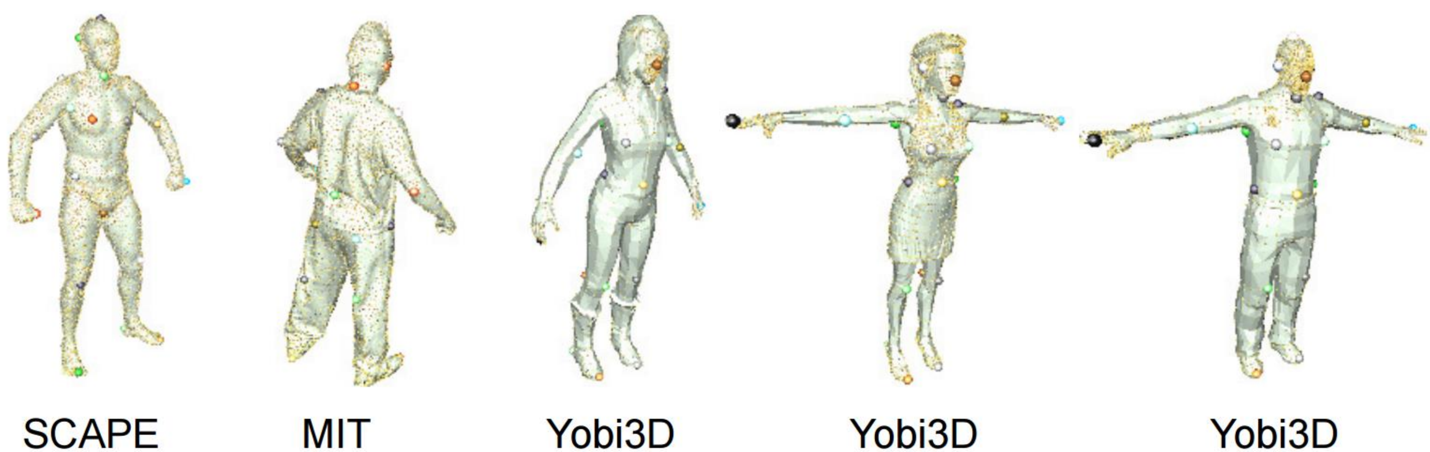
The input is a depth image

The output is a per-pixel descriptor (dim 16)

Convolution + Deconvolution

Training data

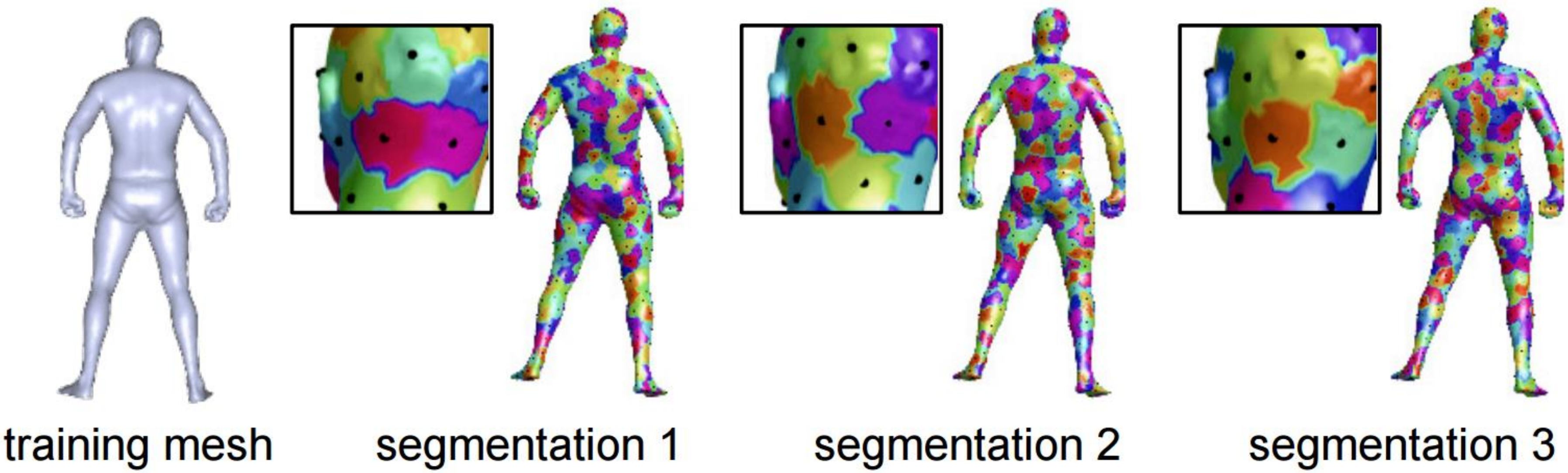
- 4 animation sequences (dense correspondences)
- 2500 shapes from Yobi3D (33 feature points)



Direct versus Indirect

- Descriptor learning (e.g., triplet loss [Schroff et al. 15])
- Classification loss (e.g., the second last layer of AlexNet)

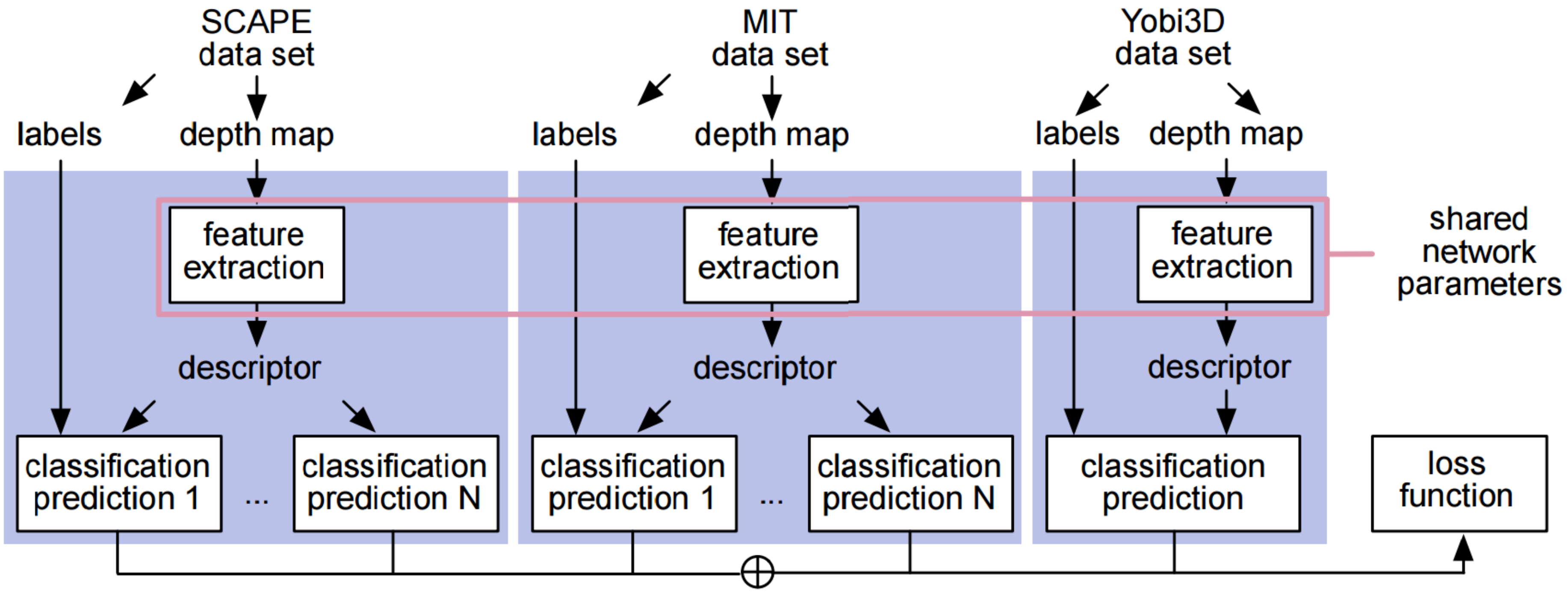
We employ a classification loss



Classes are defined in terms of super-patches

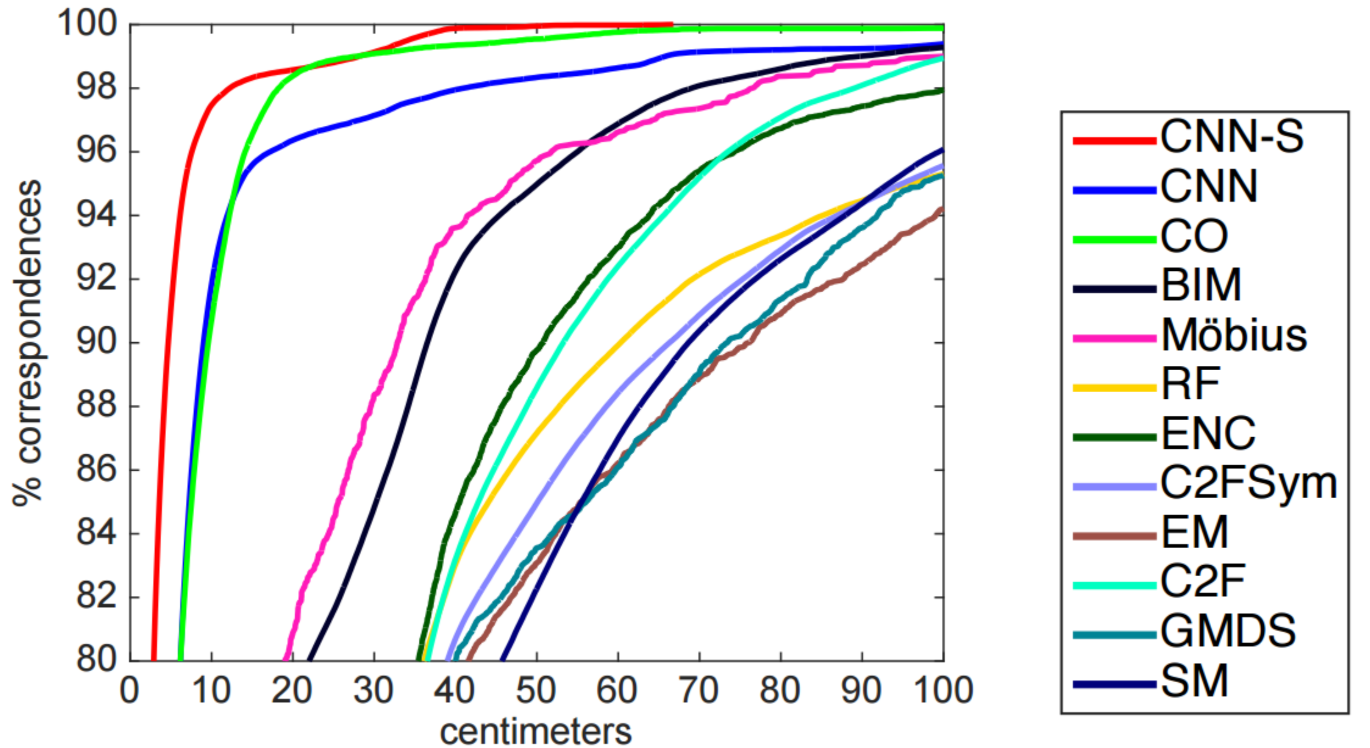
We use multiple segmentations --- so the probability of two points belong to the same segment is related to their distance

We employ the classification loss



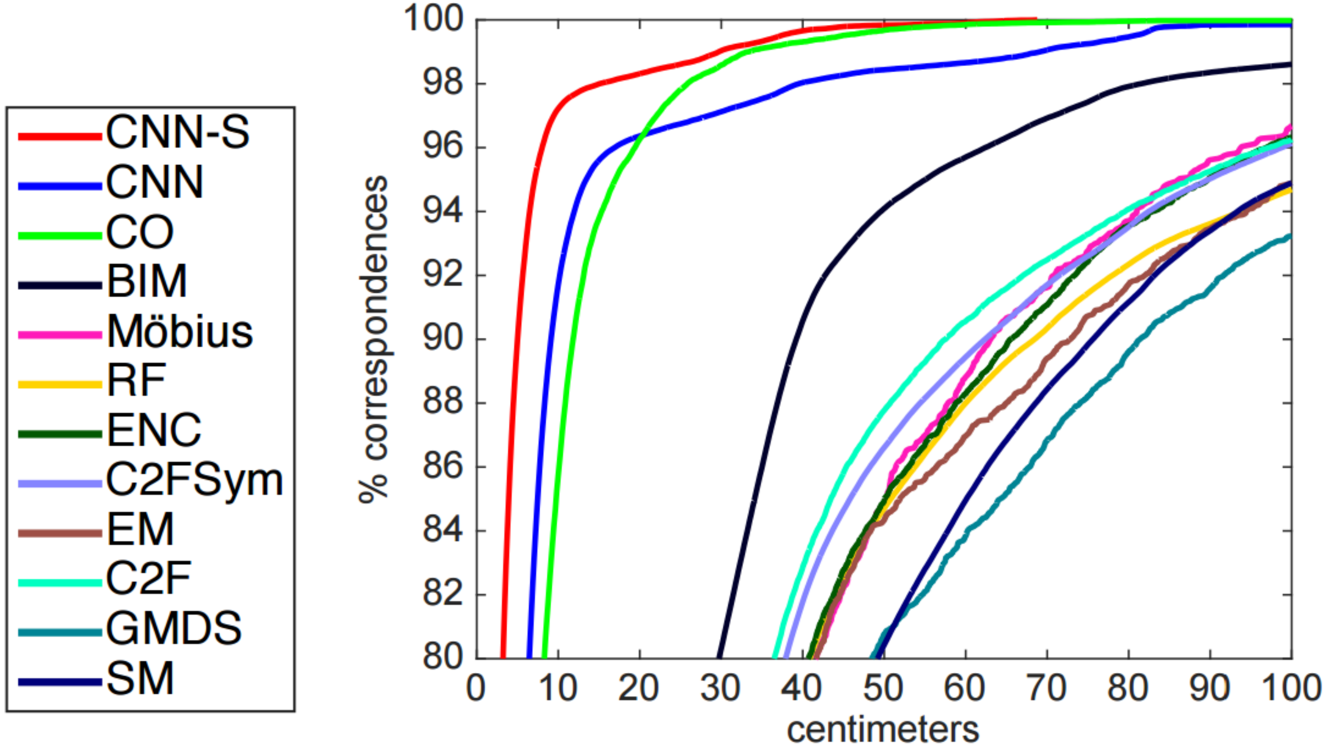
$$\{\mathbf{w}_i^\star\}, \mathbf{w}^\star = \arg\min_{\{\mathbf{w}_i\},\mathbf{w}} \sum_{i=1}^{M} l(\mathbf{w}_i, \mathbf{w})$$

Evaluation on the FAUST dataset



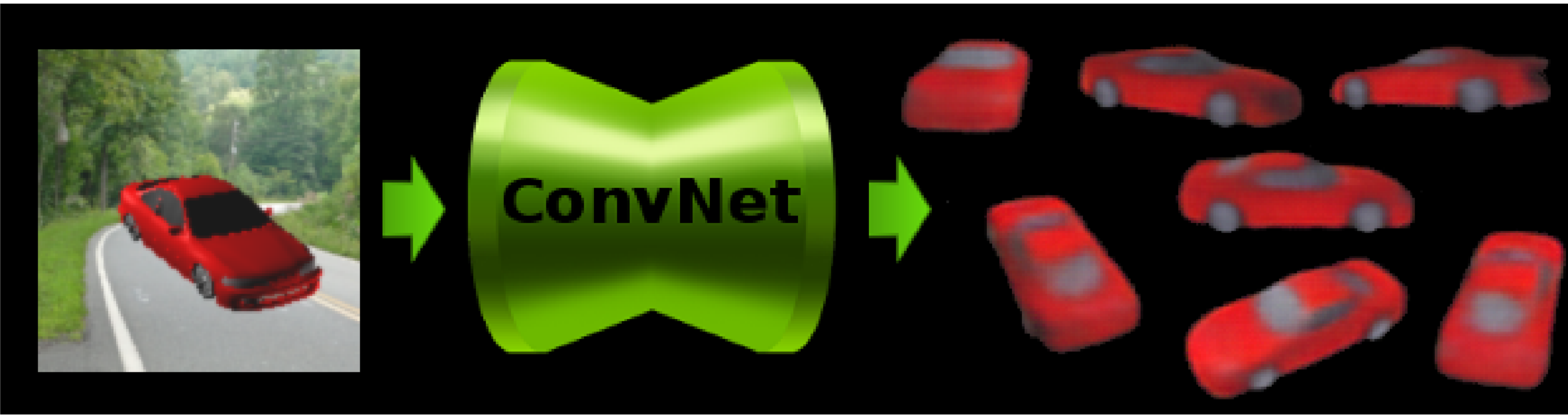
Cumulative error distribution, intra-subject

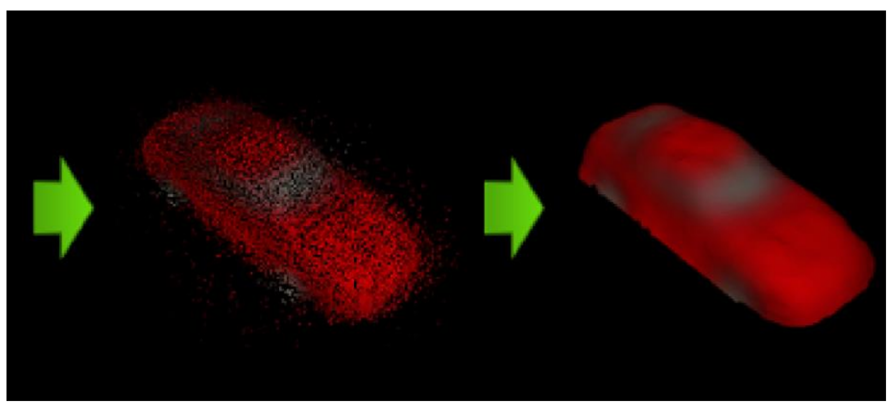
Evaluation on the FAUST dataset

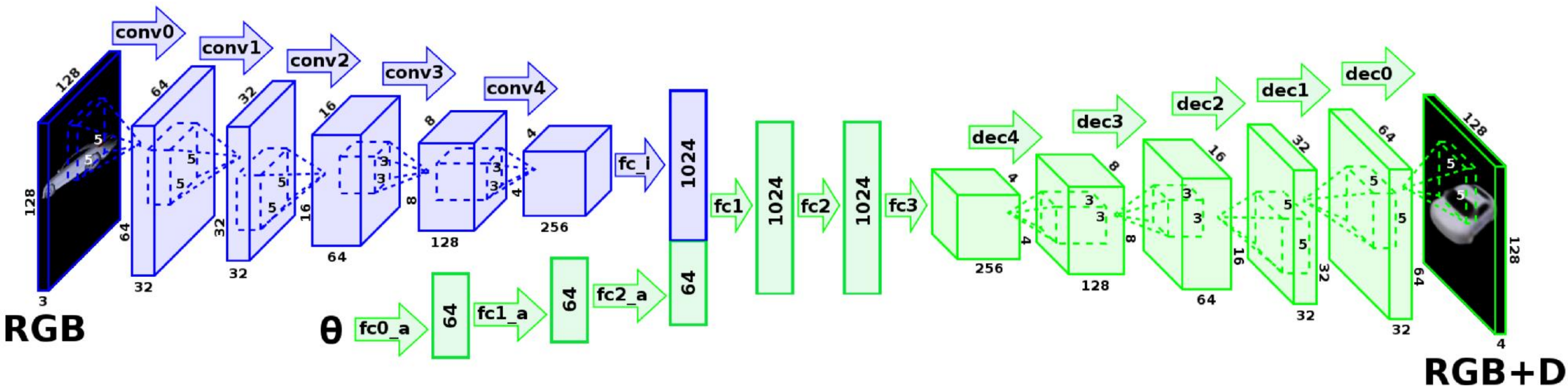


Cumulative error distribution, inter-subject

Multi-view 3D Models from Single Images With a Convolutional Network [ECCV’ 16]


ConvNet





conv0
conv1
conv2
conv3
conv4
fc_i
1024
fc1
1024
fc2
1024
fc3
dec4
dec3
dec2
dec1
dec0
RGB
θ
fc0_a
64
fc1_a
64
fc2_a
64
RGB+D

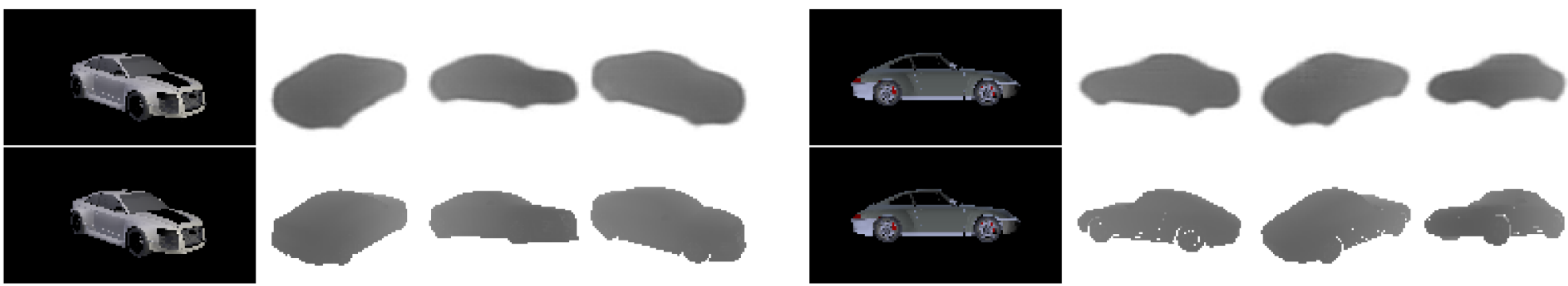


Fig. 5. Depth map predictions (**top row**) and the corresponding ground truth (**bottom row**). The network correctly estimates the shape.

Multi-view 3D Models from Single Images with a Convolutional Network

Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox

Department of Computer Science
University of Freiburg
{tatarchm, dosovits, brox}@cs.uni-freiburg.de

ECCV 2016

Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision [Yan et al. 16]

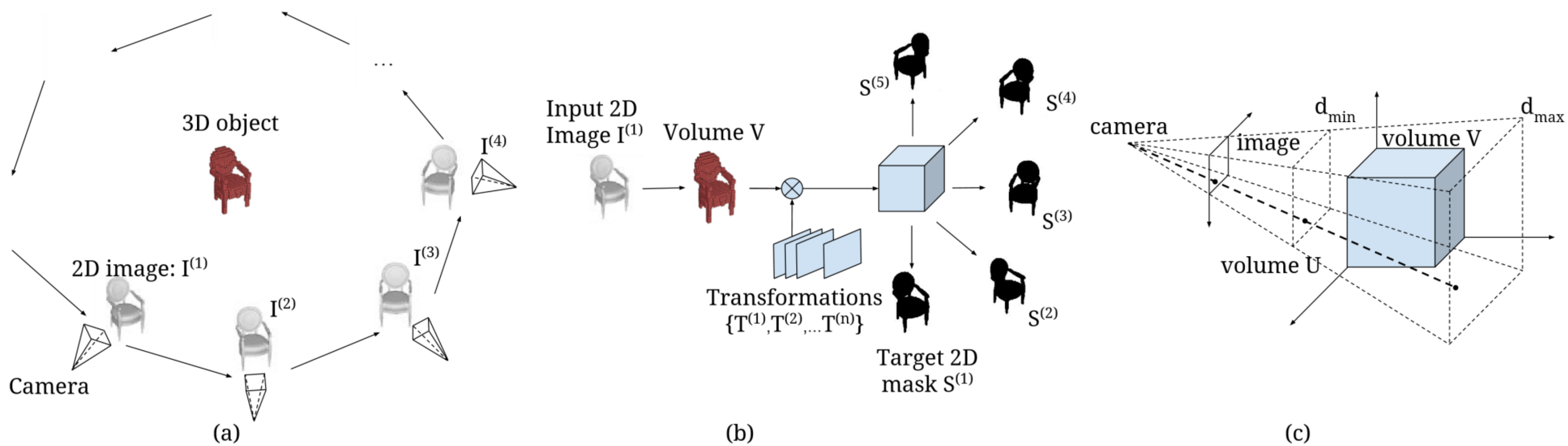
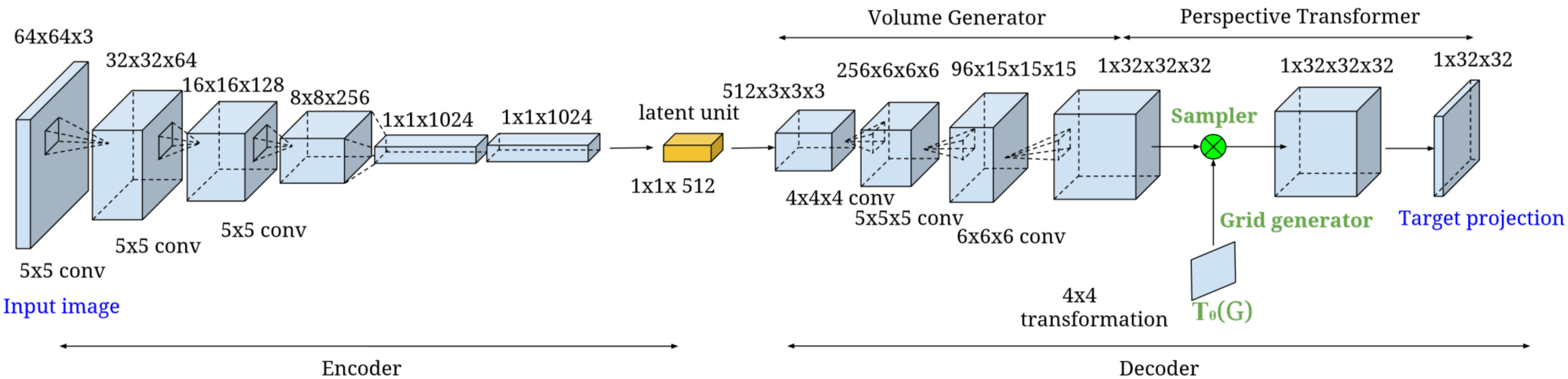


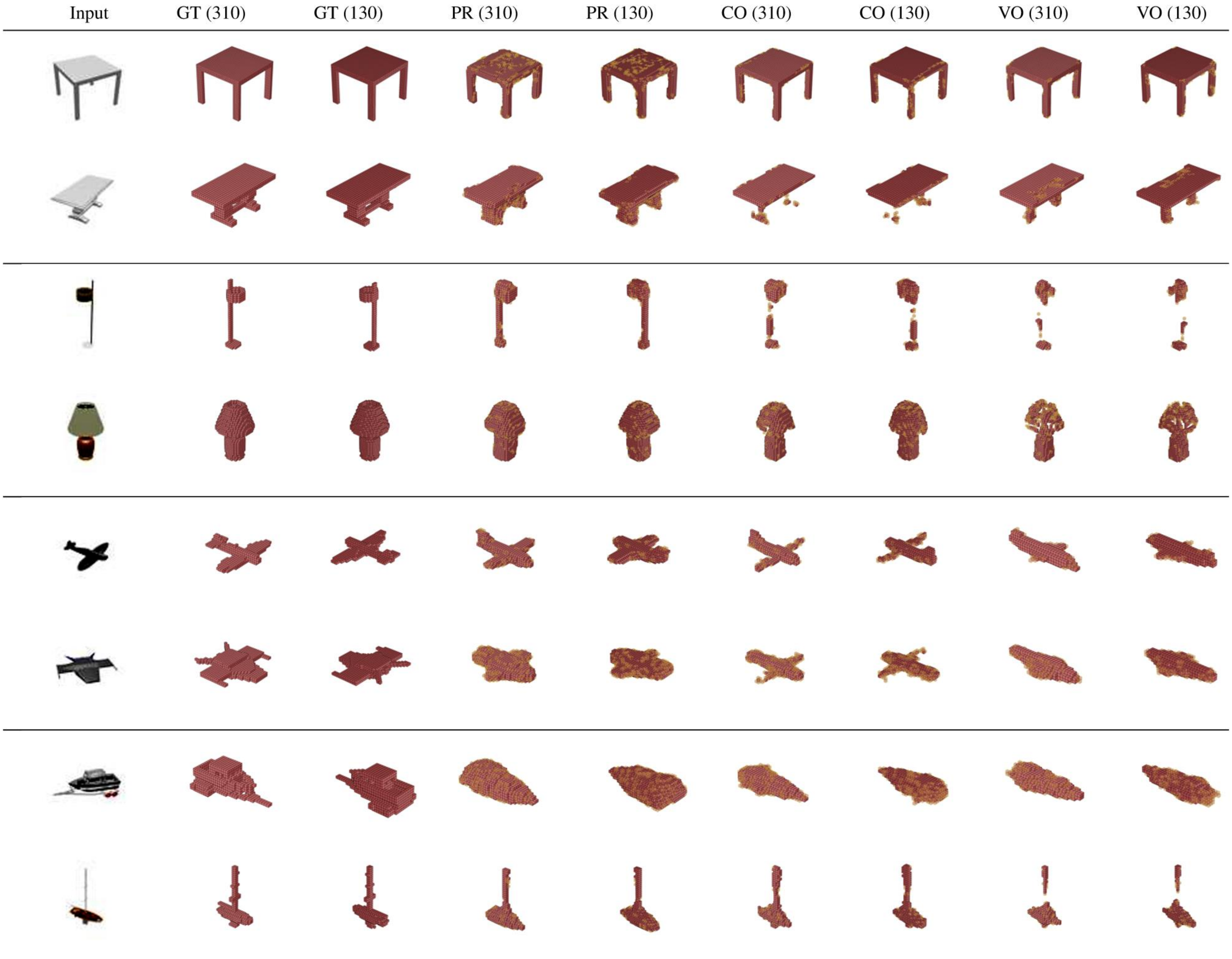
Figure 1: (a) Understanding 3D object from learning agent's perspective; (b) Single-view 3D volume reconstruction with perspective transformation. (c) Illustration of perspective projection. The minimum and maximum disparity in the screen coordinates are denoted as d_{min} and d_{max}.

$$\mathcal{L}_{vol}(I^{(k)}) = ||f(I^{(k)}) - \mathbf{V}||_2^2$$

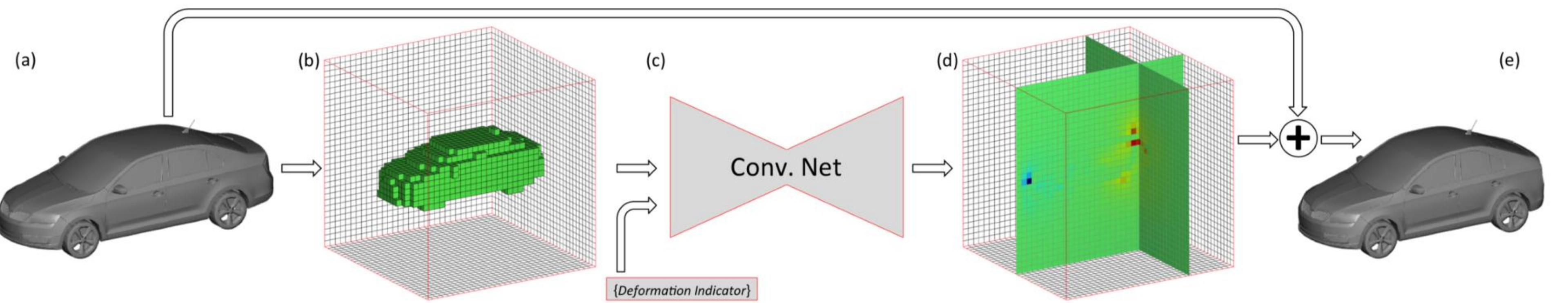
$$\mathcal{L}_{proj}(I^{(k)}) = \sum_{j=1}^{n} \mathcal{L}_{proj}^{(j)}(I^{(k)}; S^{(j)}, \alpha^{(j)}) = \frac{1}{n} \sum_{j=1}^{n} ||P(f(I^{(k)}); \alpha^{(j)}) - S^{(j)}||_2^2$$

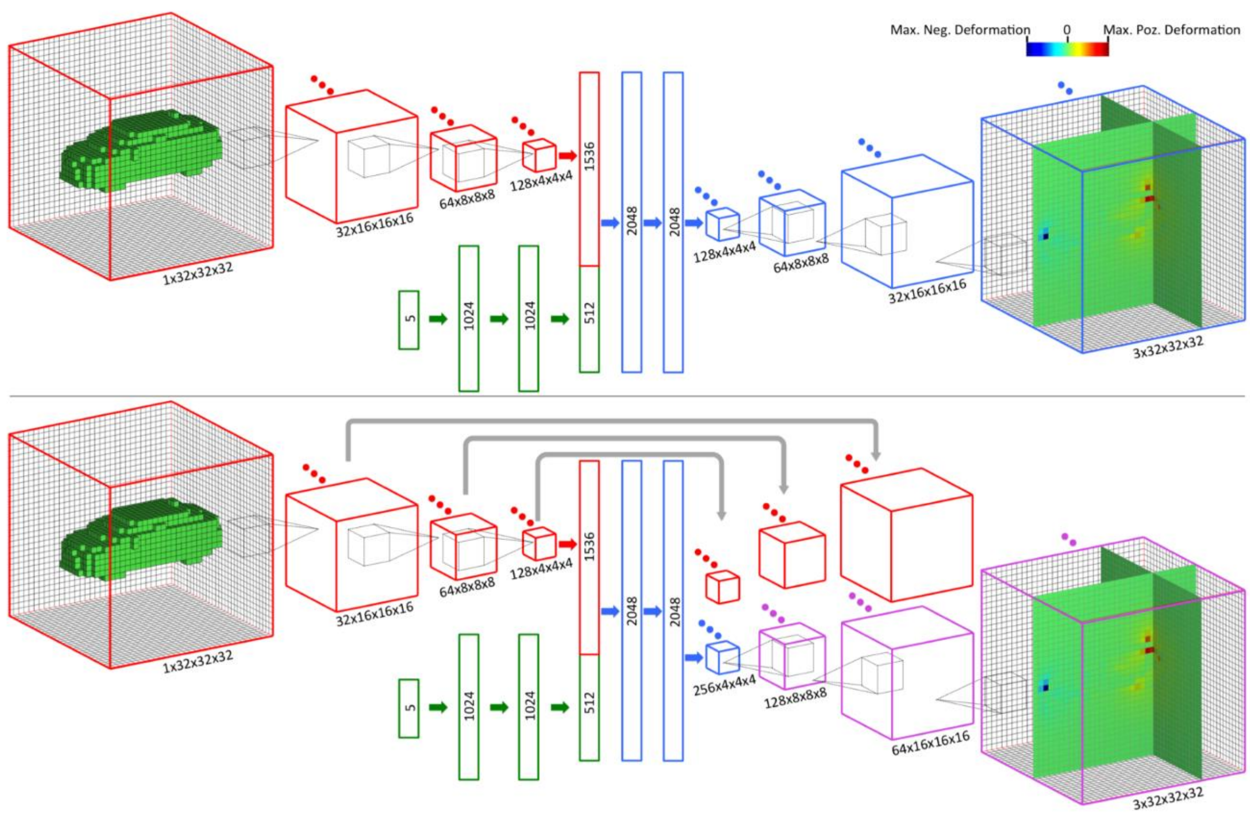
$$\mathcal{L}_{comb}(I^{(k)}) = \lambda_{proj}\mathcal{L}_{proj}(I^{(k)}) + \lambda_{vol}\mathcal{L}_{vol}(I^{(k)})$$

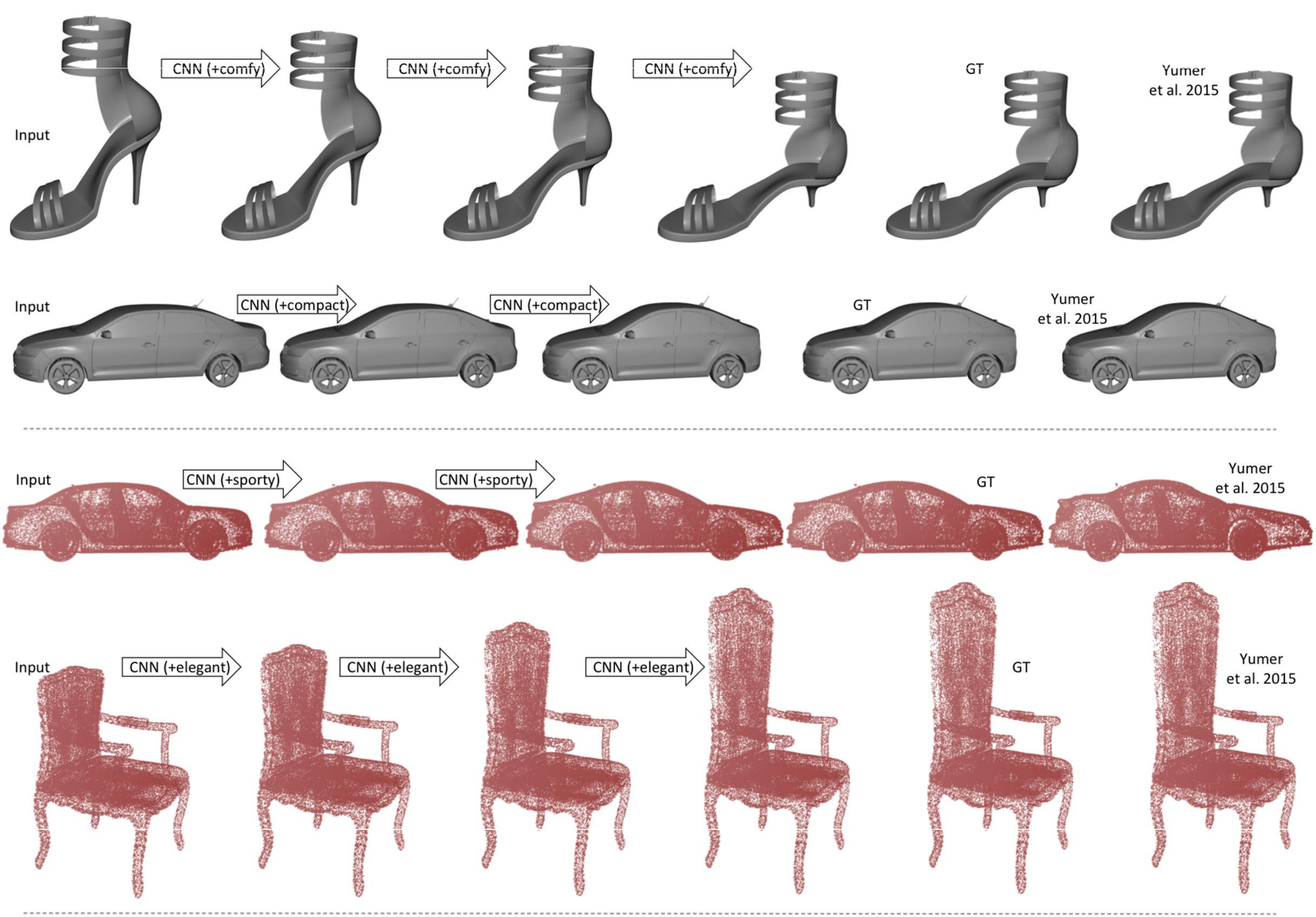

Volume Generator
Perspective Transformer
64x64x3
32x32x64
16x16x128
8x8x256
1x1x1024
1x1x1024
latent unit
1x1x 512
512x3x3x3
256x6x6x6
96x15x15x15
1x32x32x32
1x32x32x32
1x32x32
Sampler
4x4x4 conv
5x5x5 conv
6x6x6 conv
Grid generator
Target projection
5x5 conv
5x5 conv
5x5 conv
Input image
4x4
transformation
Tθ(G)
Encoder
Decoder


Input
GT (310)
GT (130)
PR (310)
PR (130)
CO (310)
CO (130)
VO (310)
VO (130)

Learning Semantic Deformation Flows with 3D Convolutional Networks [Yumer and Mitra 2016]

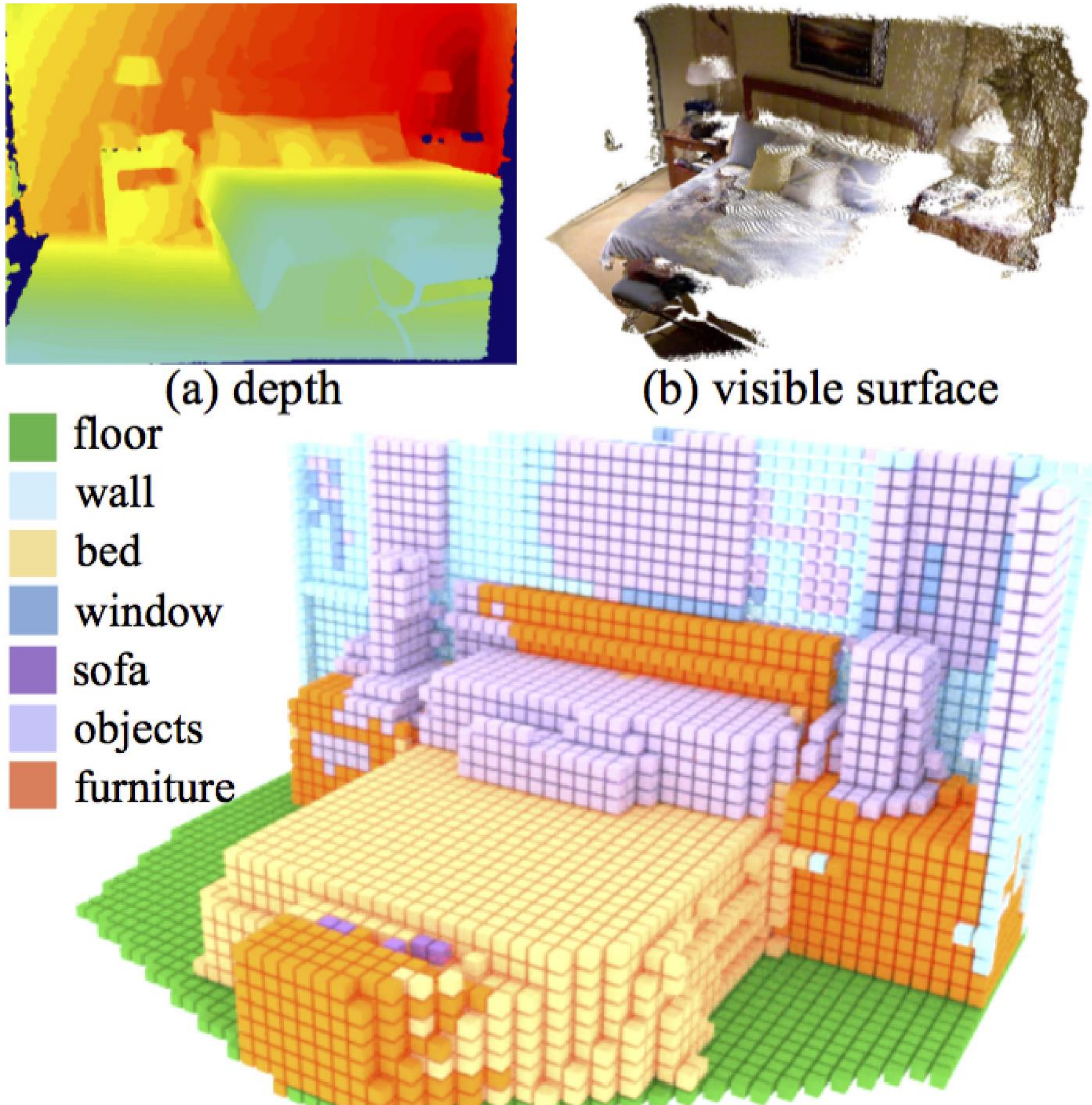

(a)
(b)
(c)
(d)
(e)
Conv. Net
{Deformation Indicator}


Max. Neg. Deformation
0
Max. Poz. Deformation
1x32x32x32
32x16x16x16
64x8x8x8
128x4x4x4
1536
512
2048
2048
5
1024
1024
128x4x4x4
64x8x8x8
32x16x16x16
3x32x32x32
1x32x32x32
32x16x16x16
64x8x8x8
128x4x4x4
1536
512
2048
2048
5
1024
1024
256x4x4x4
128x8x8x8
64x16x16x16
3x32x32x32



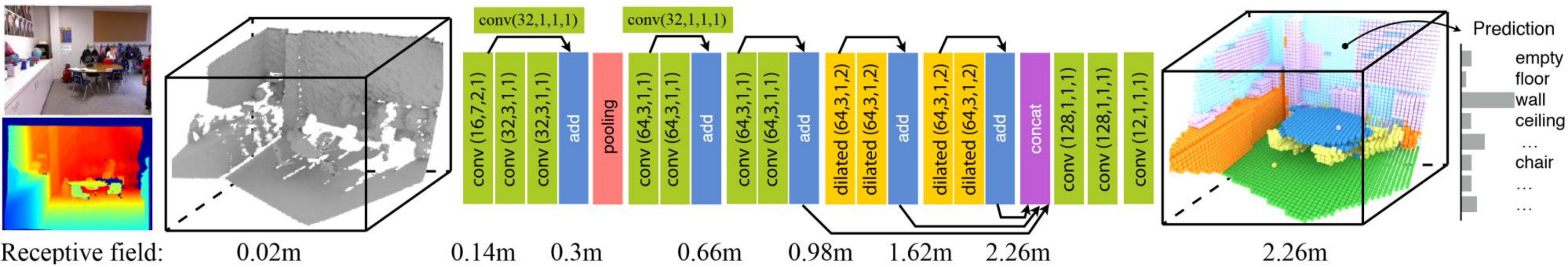
Input
CNN (+comfy)
CNN (+comfy)
CNN (+comfy)
GT
Yumer et al. 2015
Input
CNN (+compact)
CNN (+compact)
GT
Yumer et al. 2015
Input
CNN (+sporty)
CNN (+sporty)
GT
Yumer et al. 2015
Input
CNN (+elegant)
CNN (+elegant)
CNN (+elegant)
GT
Yumer et al. 2015

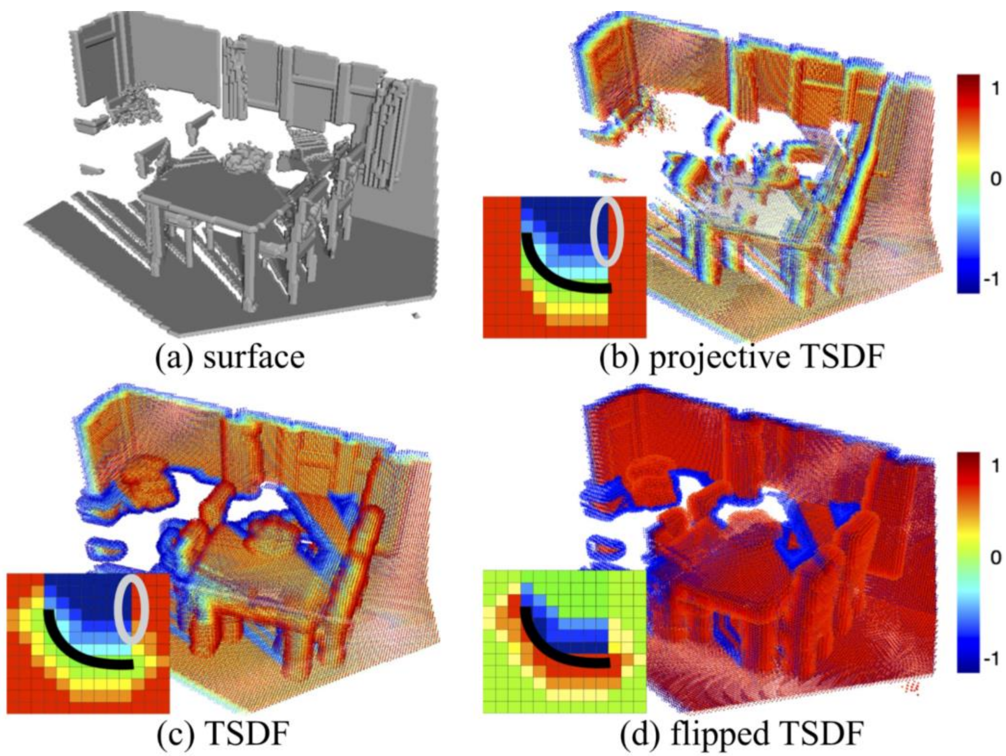
Semantic Scene Completion from a Single Depth Image [Song et al. 17]

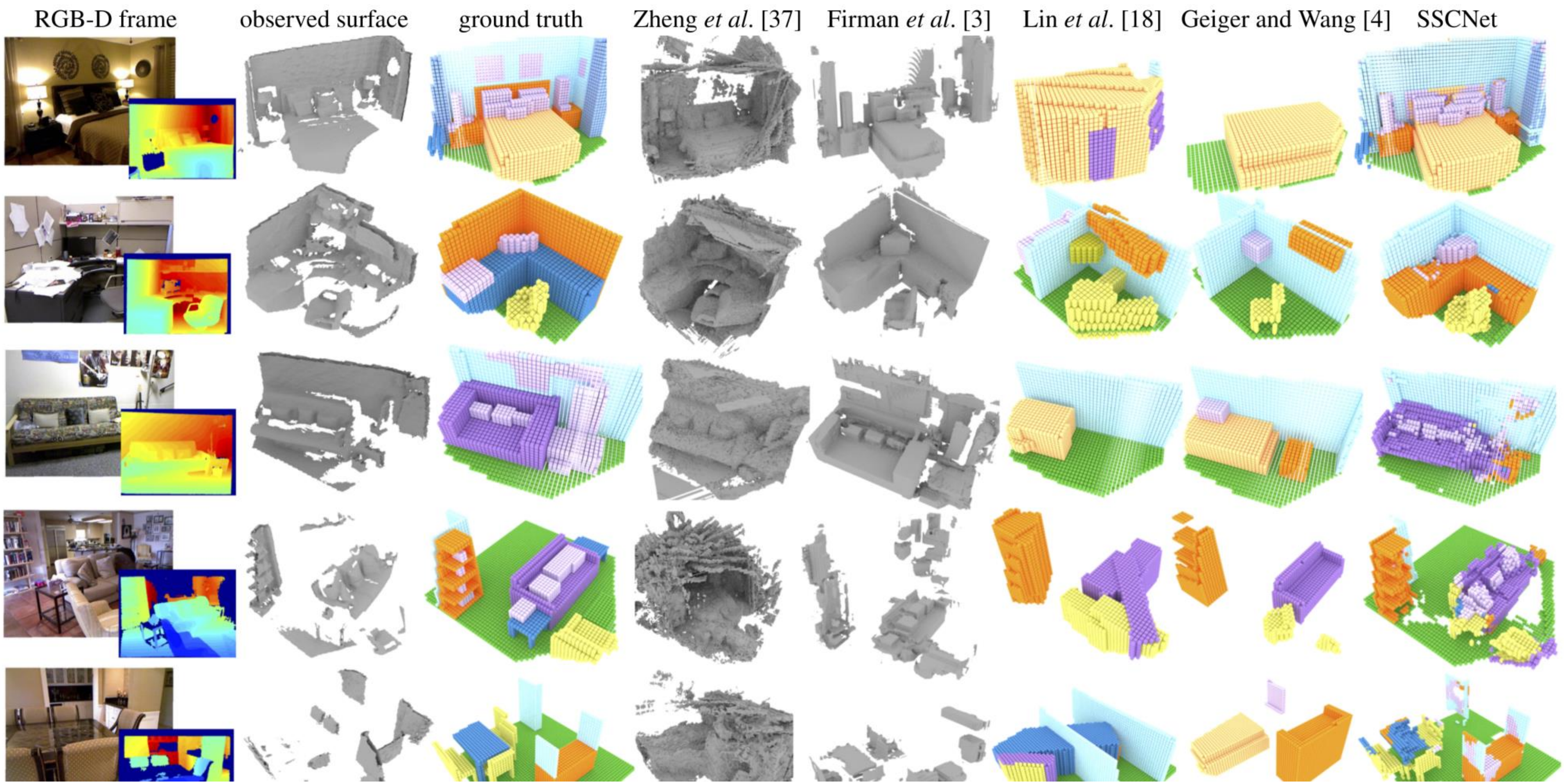


(a) depth (b) visible surface

(c) output


conv(32,1,1,1)
conv(32,1,1,1)
conv (16,7,2,1)
conv (32,3,1,1)
conv (32,3,1,1)
add
pooling
conv (64,3,1,1)
conv (64,3,1,1)
add
conv (64,3,1,1)
conv (64,3,1,1)
add
dilated (64,3,1,2)
dilated (64,3,1,2)
add
dilated (64,3,1,2)
dilated (64,3,1,2)
add
concat
conv (128,1,1,1)
conv (128,1,1,1)
conv (12,1,1,1)
Prediction
empty
floor
wall
ceiling
…
chair
…
…
Receptive field:
0.02m
0.14m
0.3m
0.66m
0.98m
1.62m
2.26m
2.26m


1
0
-1
(a) surface
(b) projective TSDF
1
0
-1
(c) TSDF
(d) flipped TSDF

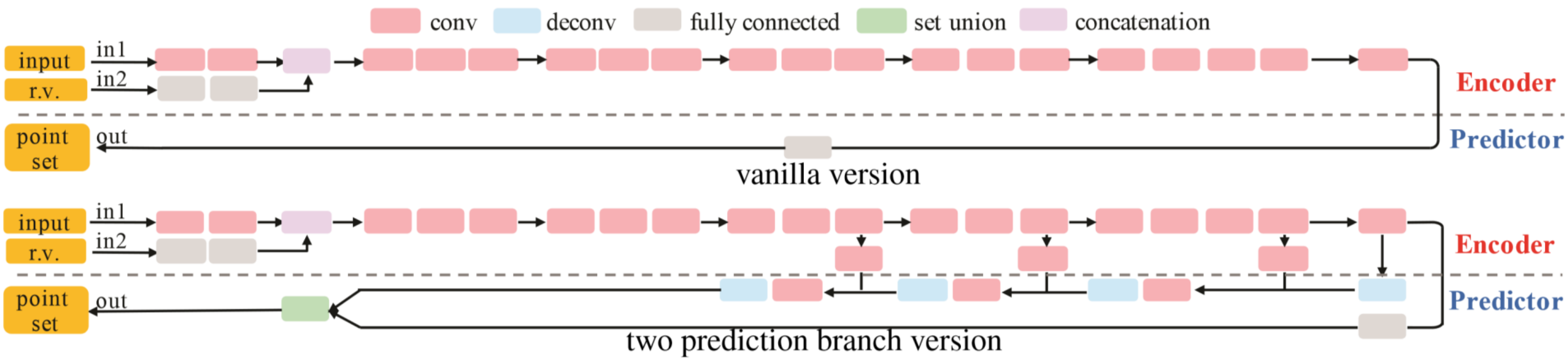

RGB-D frame
observed surface
ground truth
Zheng *et al*. [37]
Firman *et al*. [3]
Lin *et al*. [18]
Geiger and Wang [4]
SSCNet

A Point Set Generation Network for 3D Object Reconstruction from a Single Image [Fan, Su, Guibas, 2017]

Input

Reconstructed 3D point cloud

Network Architecture



Distance Metrics

- Chamfer distance

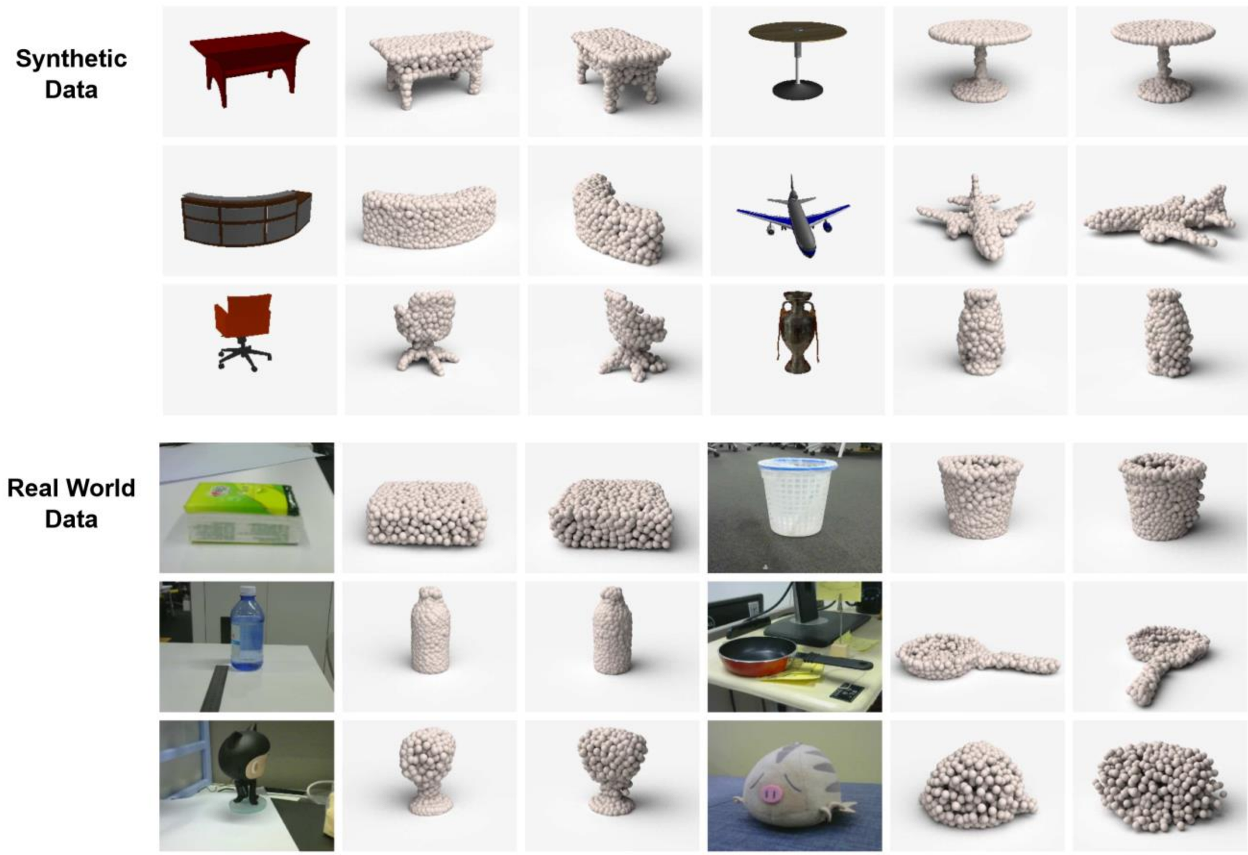
$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

- Earth Mover’s distance

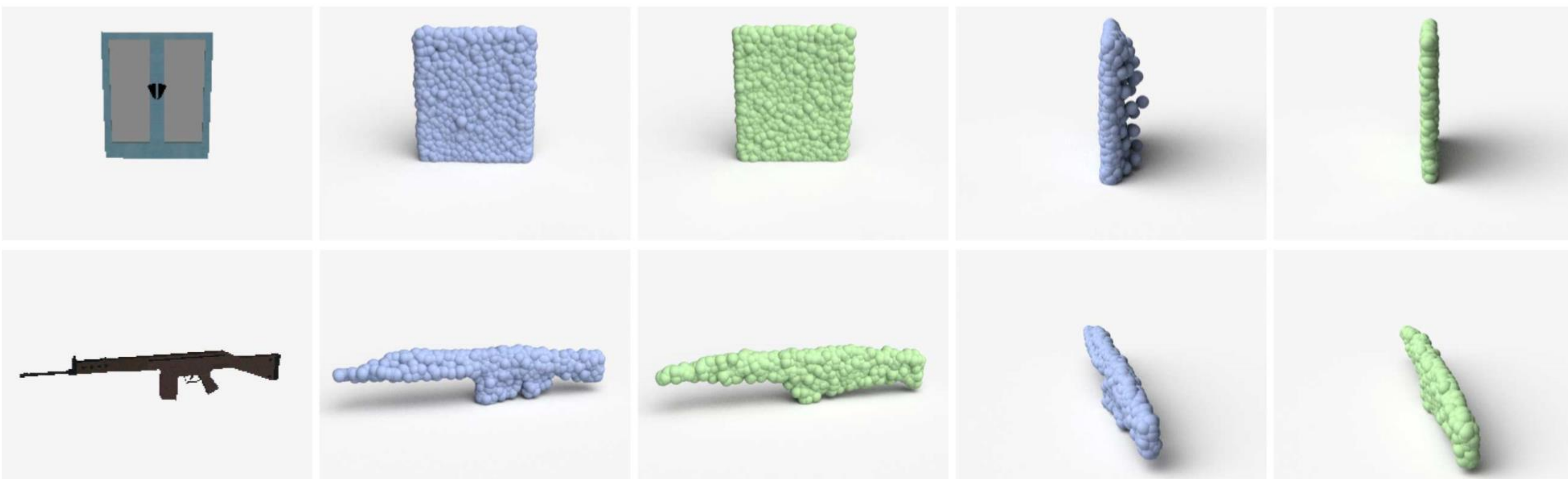
$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

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

Visual results



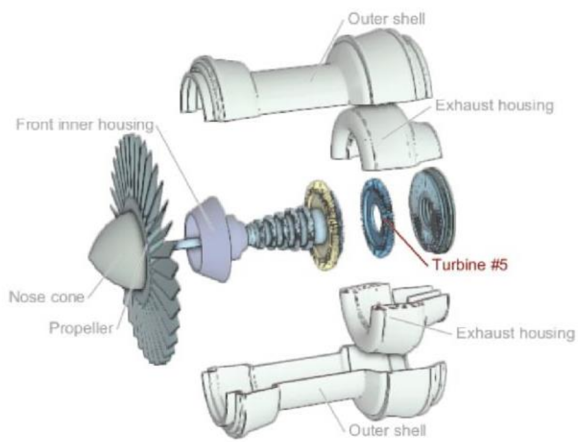
CD (Left) versus EMD (Right)

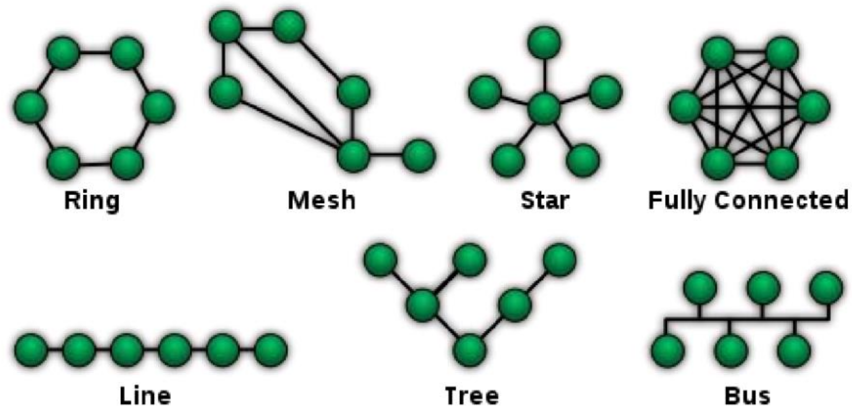


GRASS: Generative Recursive Autoencoders for Shape Structures
[Li, Xu, Chaudhuri, Yumer, Zhang, Guibas, SIGGRAPH’ 17]

Huge Variety of (Attributed) Graphs

- Arbitrary numbers/types of vertices (parts), arbitrary numbers of connections (adjacencies/symmetries)

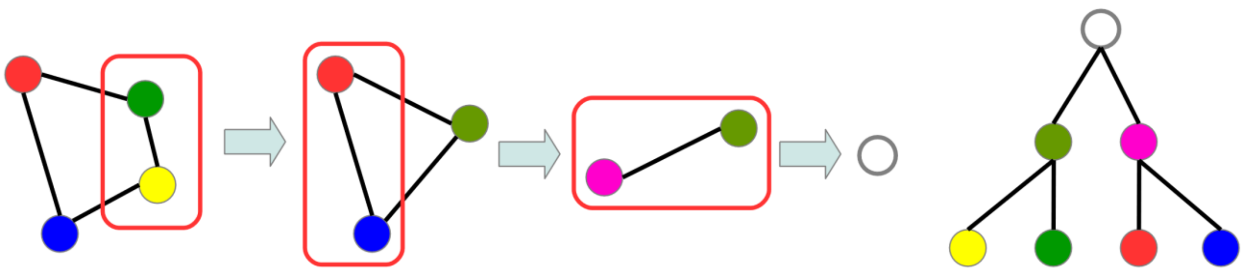




- For linear graphs (chains) of arbitrary length, we can use a recurrent neural network (RNN/LSTM)

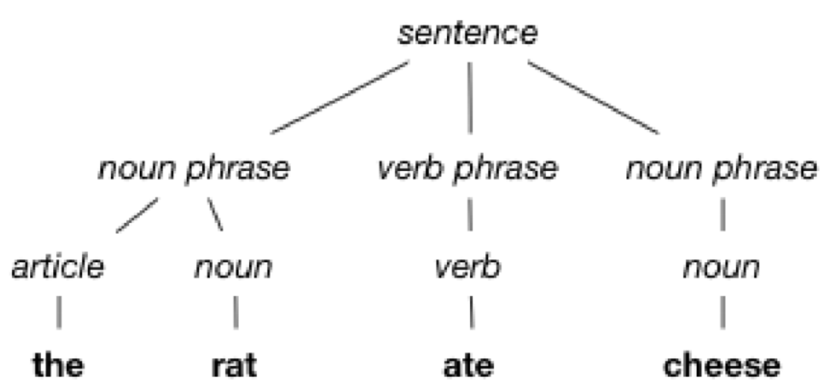
Key Insight

- Edges of a graph can be collapsed sequentially to yield a hierarchical structure



- Looks like a parse tree

  for a sentence!

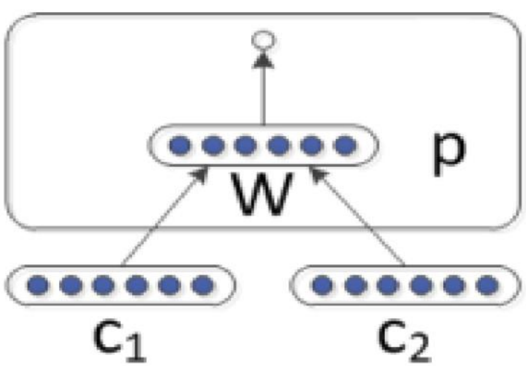


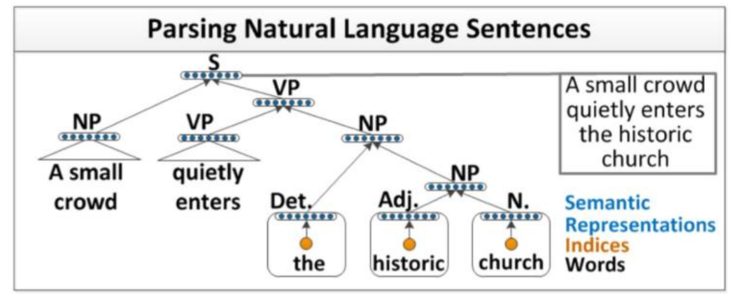
Recursive Neural Network (RvNN)

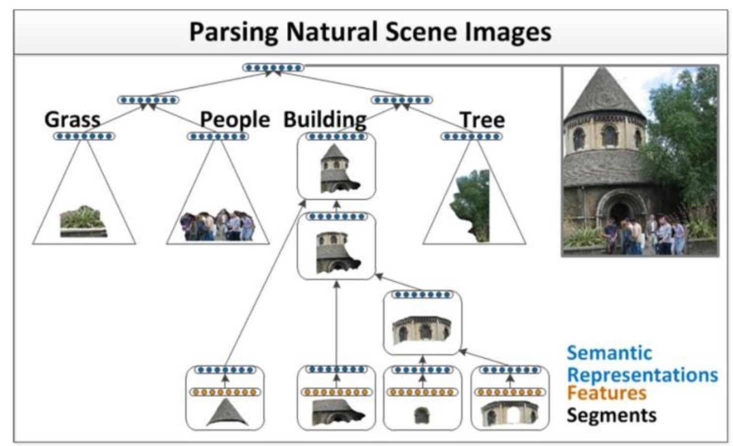
- Repeatedly merge two nodes into one

- Each node has an n-D feature vector, computed recursively

$$p = f(W[c_1;c_2] + b)$$

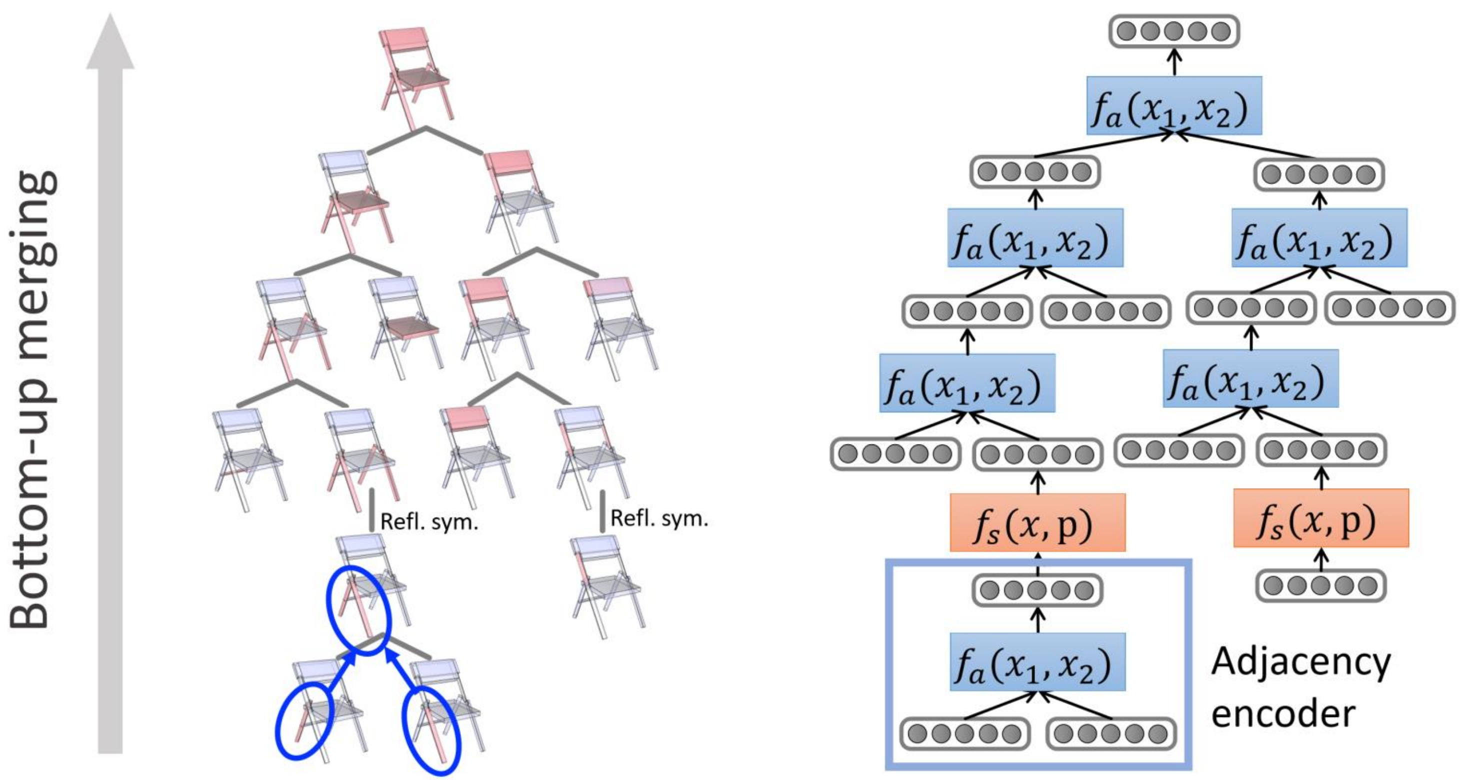




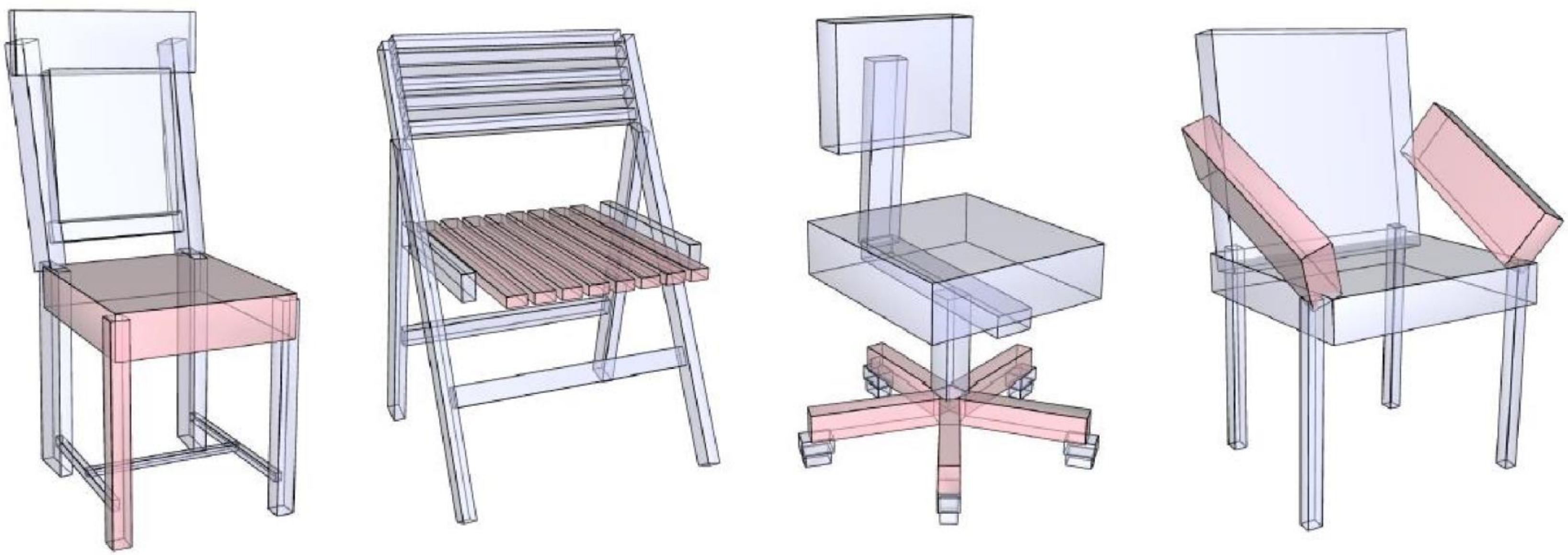


Socher et al. 2011

Recursively Merging Parts

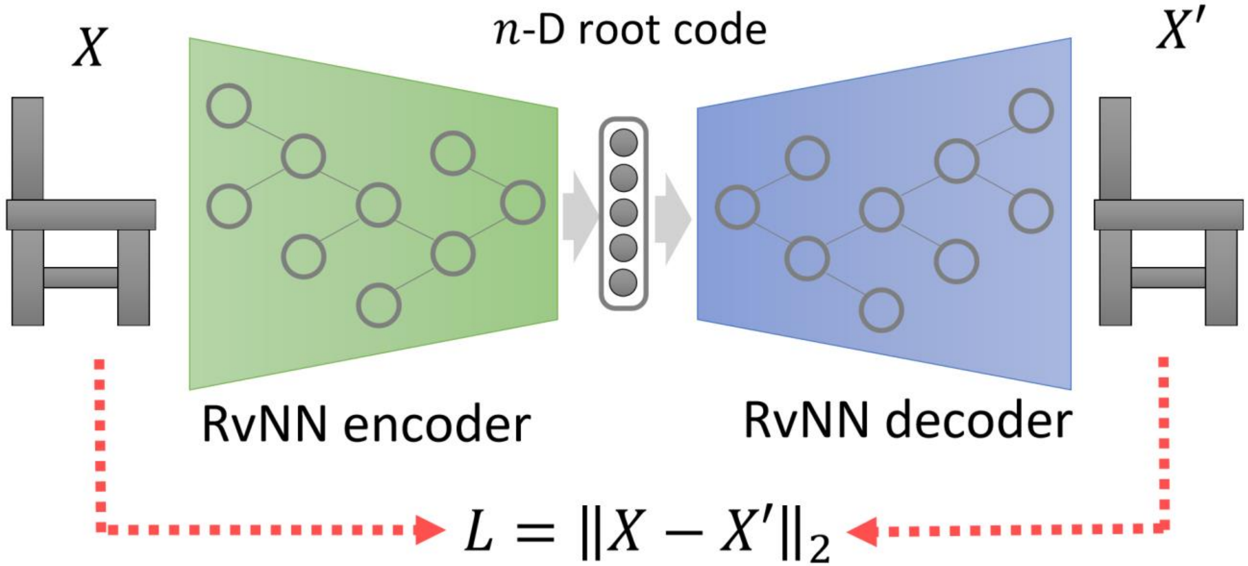


Different types of merges, varying cardinalities!



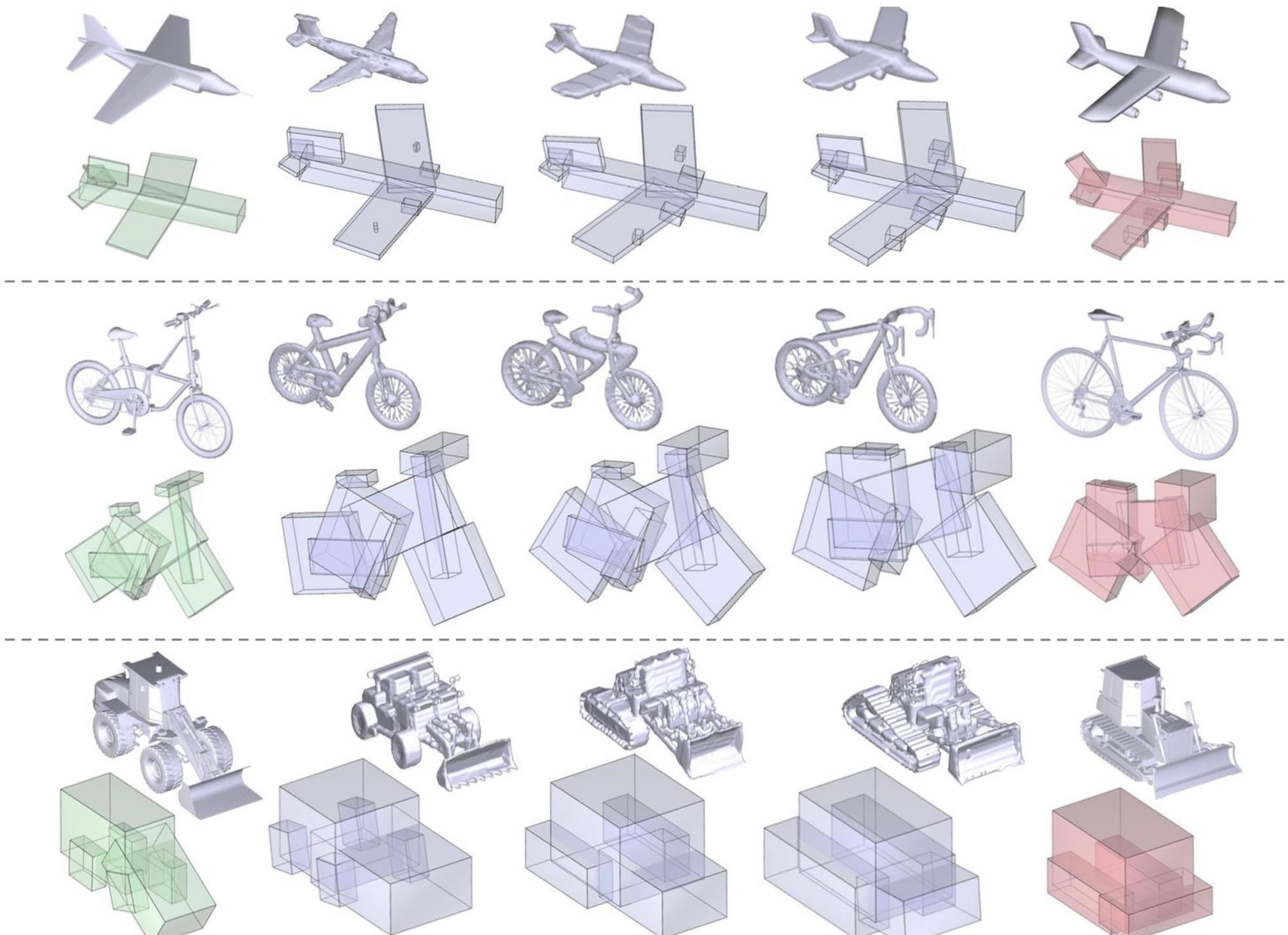
Adjacency | Translational symmetry | Rotational symmetry | Reflectional symmetry

Training with Reconstruction Loss



- Learn weights from a variety of randomly sampled merge orders for each box structure

Results: Shape interpolation



Discussion