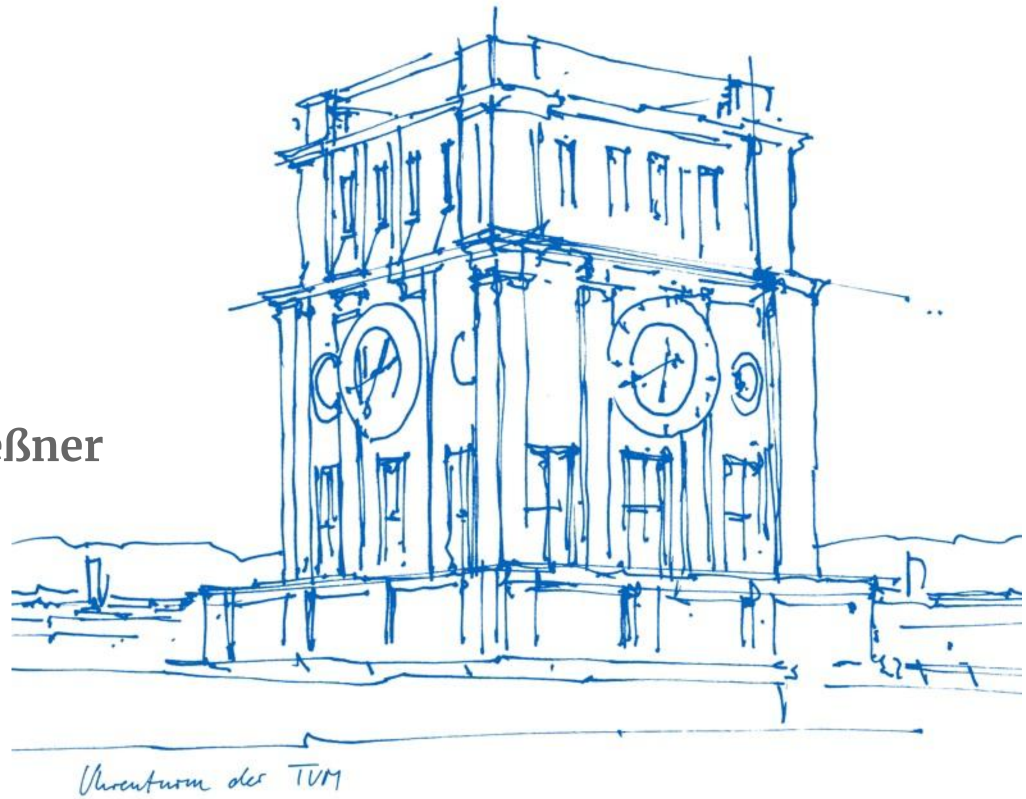
Visual Computing Lab
Department of Informatics
Technical University of Munich

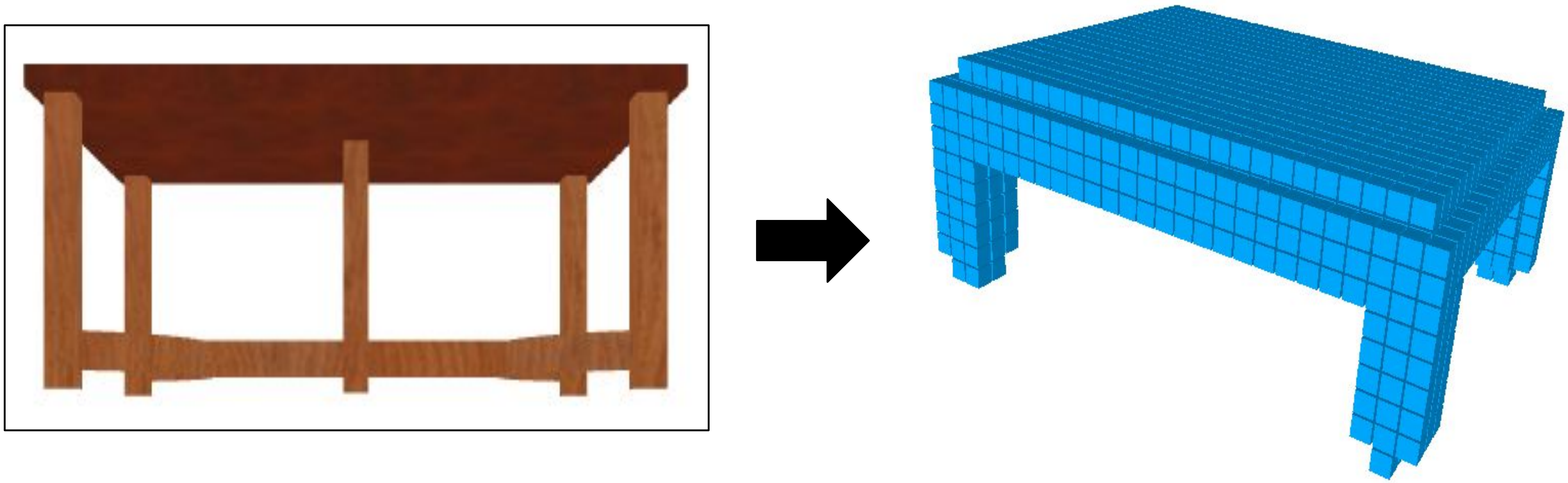
Single View 3D Object Reconstruction

(Guided Research)

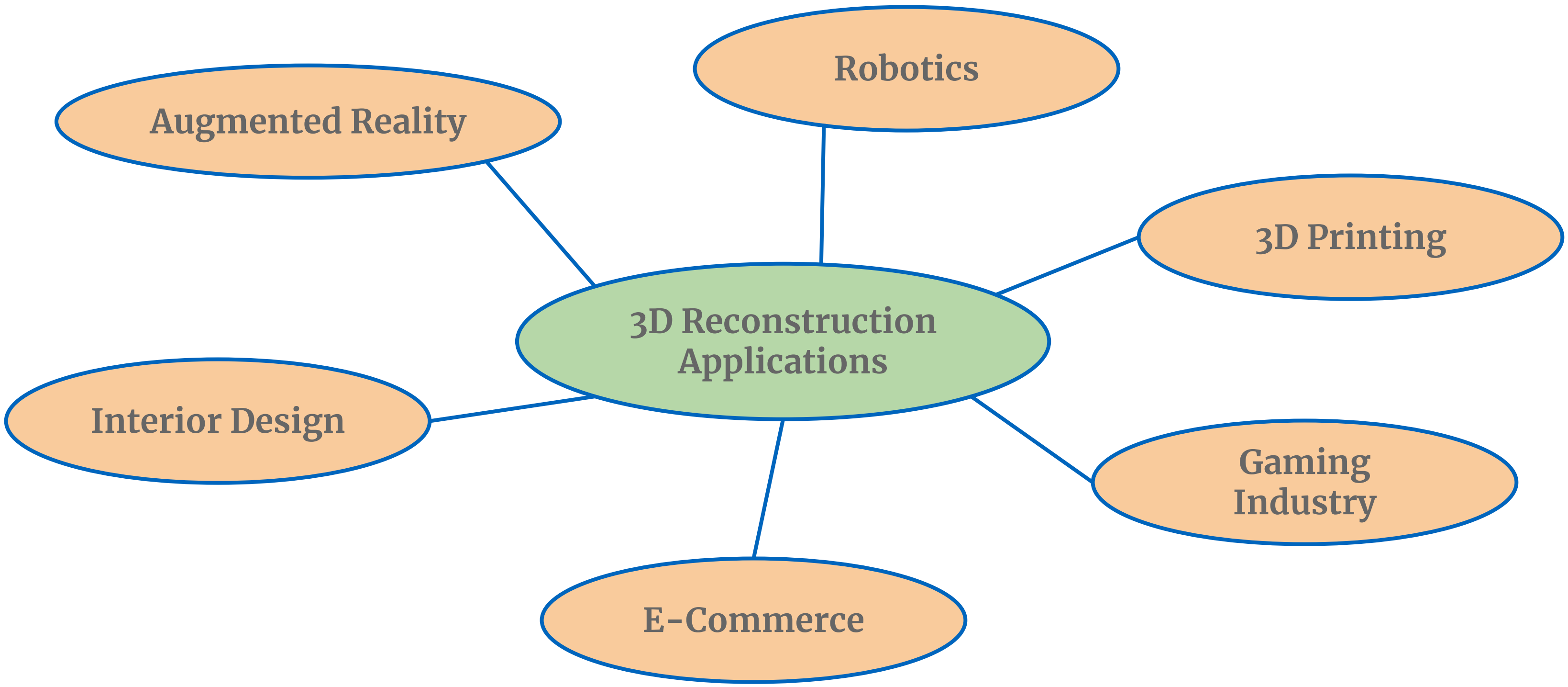


Parika Goel
Advisor : Dr. Angela Dai
Supervisor : Prof. Dr. Matthias Nießner

Goal : 3D Reconstruction from Images



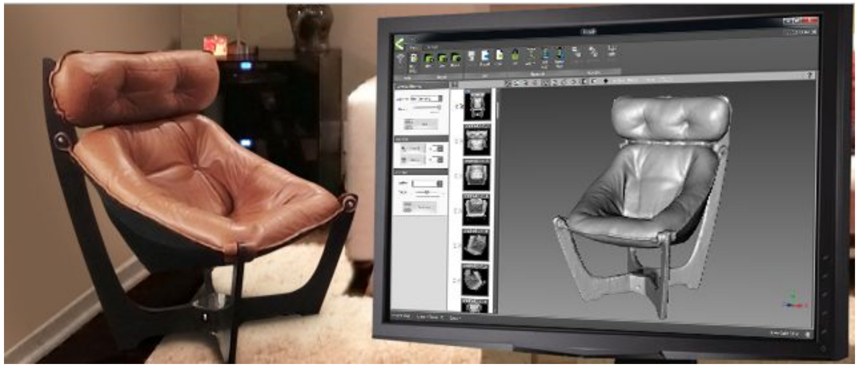
Motivation



Robotics
Augmented Reality
3D Printing
3D Reconstruction Applications
Interior Design
Gaming Industry
E-Commerce

Motivation : 3D Printing

Motivation : Interior Design



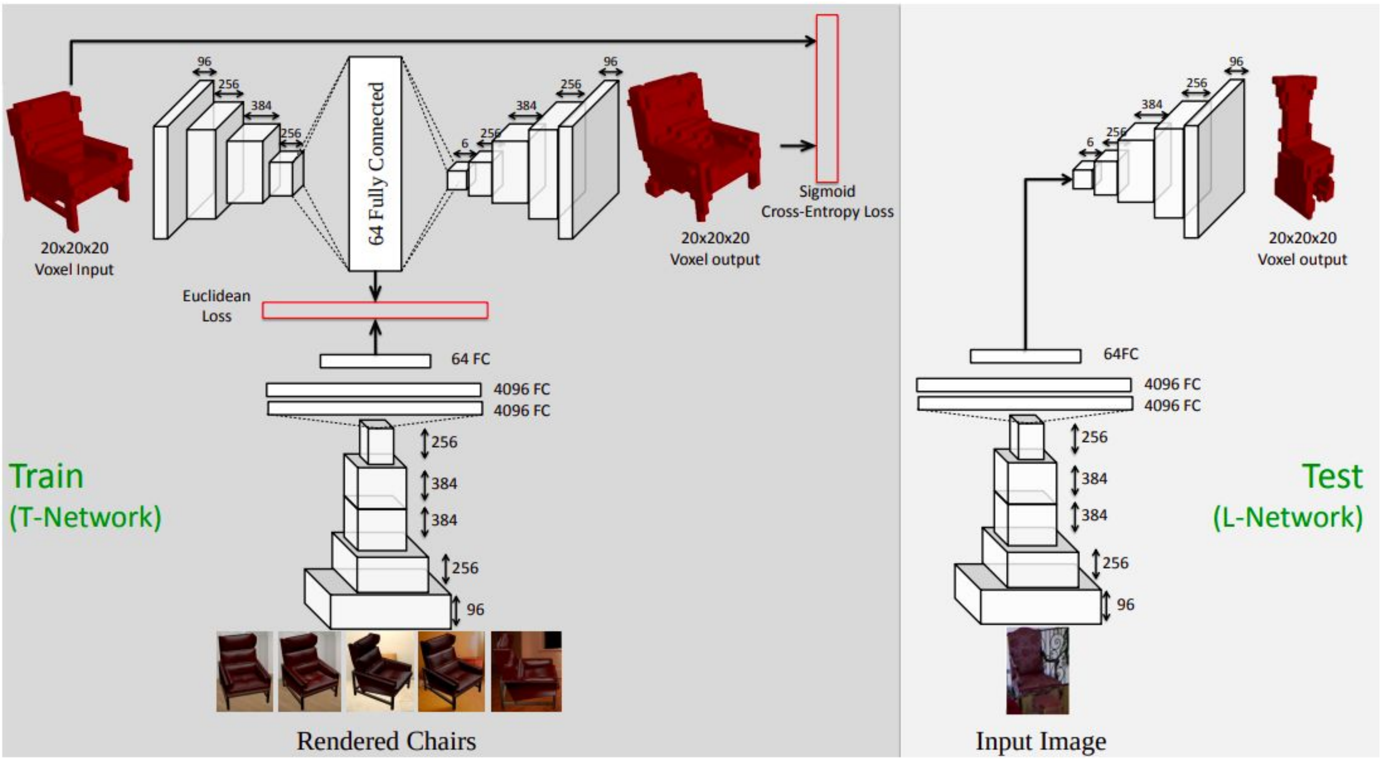
https://sketchfab.com/

Motivation : Augmented Reality

https://program-ace.com/blog/photogrammetry-in-ar/

Related Work

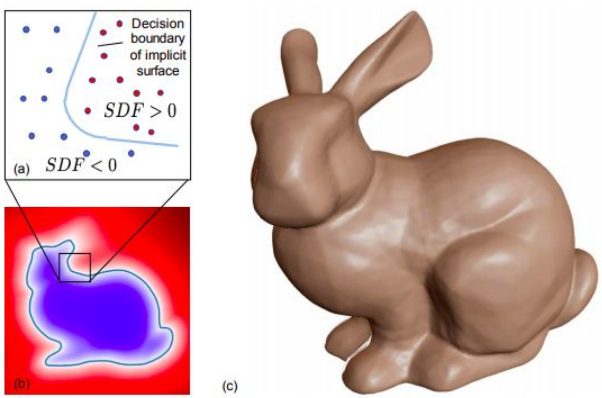
Learning a Predictable and Generative Vector Representation for Objects. Girdhar et al. 2016

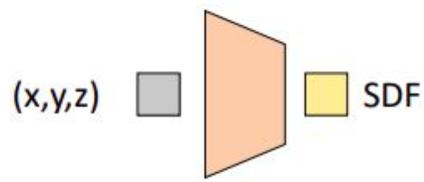


https://arxiv.org/pdf/1603.08637.pdf

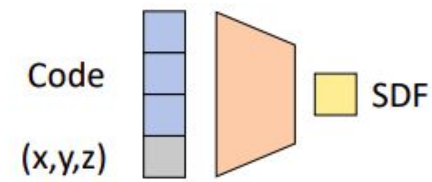
Related Work

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. Park et al. 2019

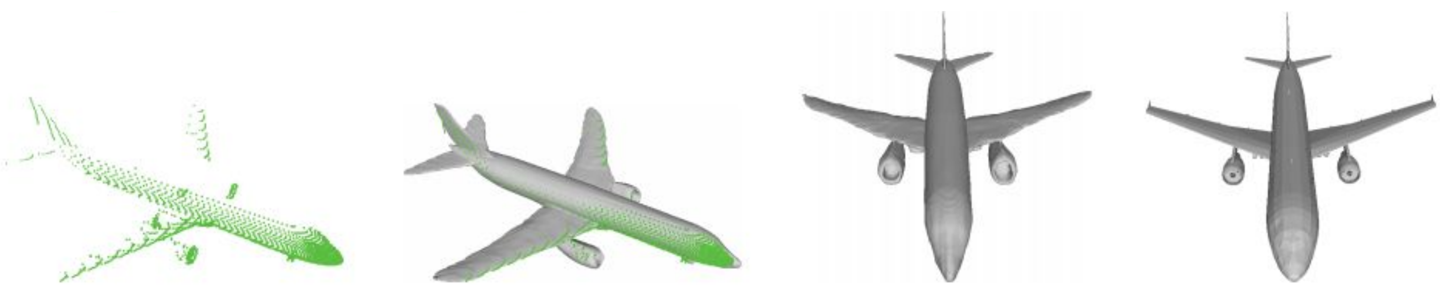




(a) Single Shape DeepSDF



(b) Coded Shape DeepSDF

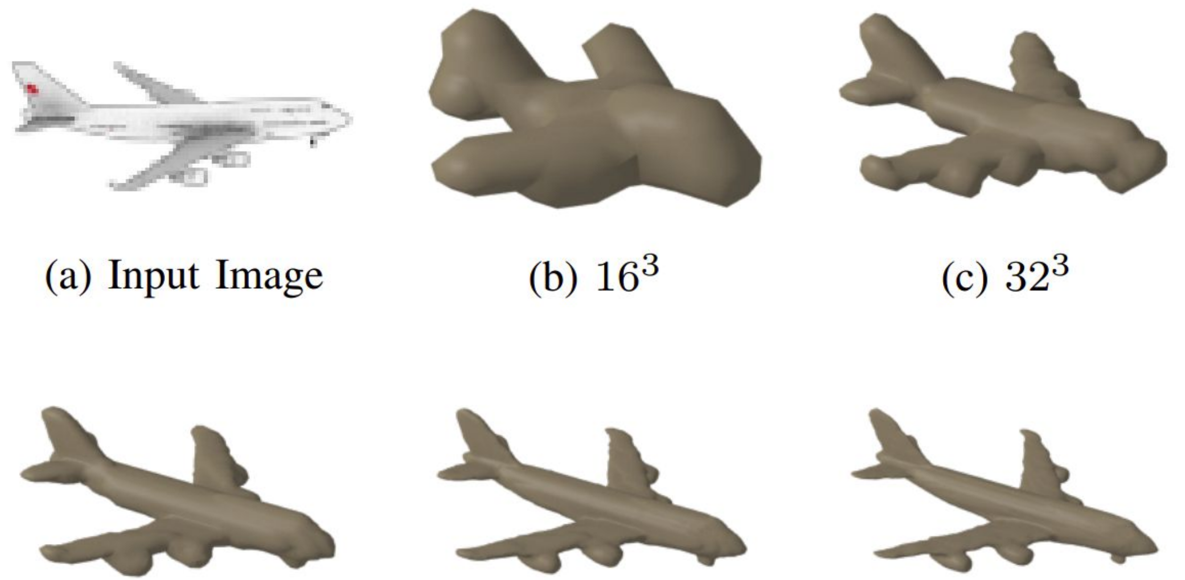


(a) Input Depth **(b)** Completion (ours) **(c)** Second View (ours) **(d)** Ground truth

https://arxiv.org/pdf/1901.05103.pdf

Related Work

Hierarchical Surface Prediction for 3D Object Reconstruction. Hane et al. 2017



(a) Input Image (b) 16^3 (c) 32^3

(d) 64^3 (e) 128^3 (f) 256^3

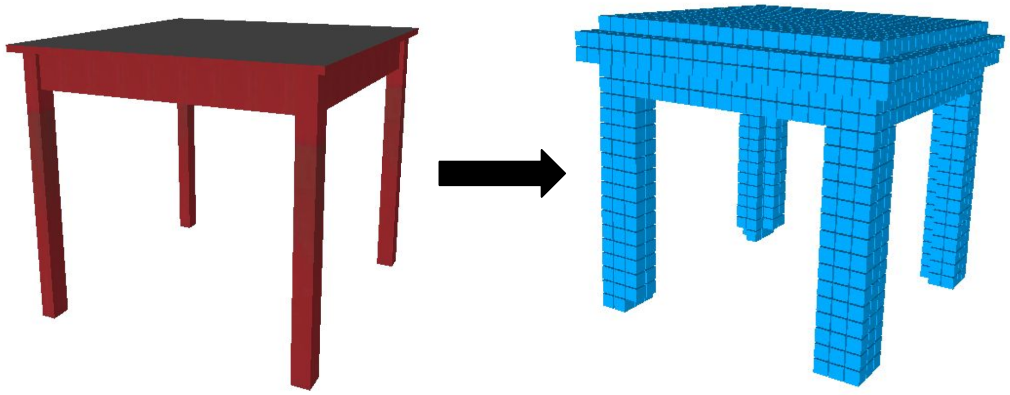
https://arxiv.org/pdf/1704.00710.pdf

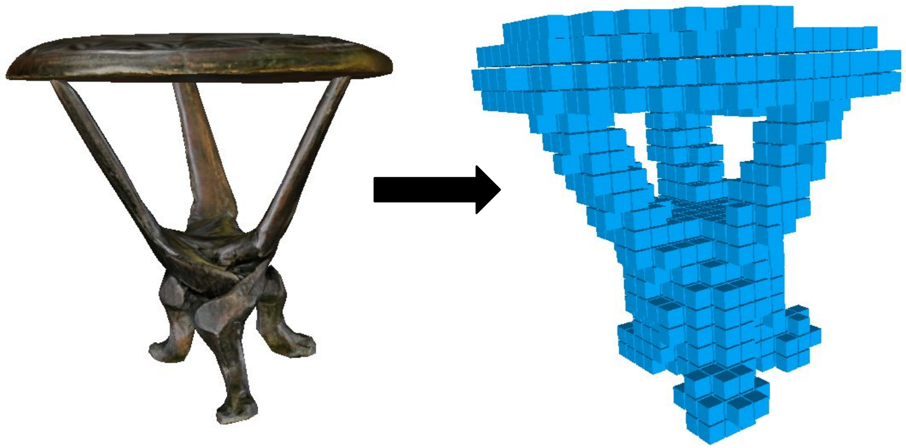
Contribution

- Comparison of Occupancy Grid and Distance Field Representation for the task of 3D Reconstruction
- Comparison of two network architectures (Net3D vs UNet3D) for the task of 3D Reconstruction

Data Generation

Ground Truth: Voxelize shapenet table from .obj

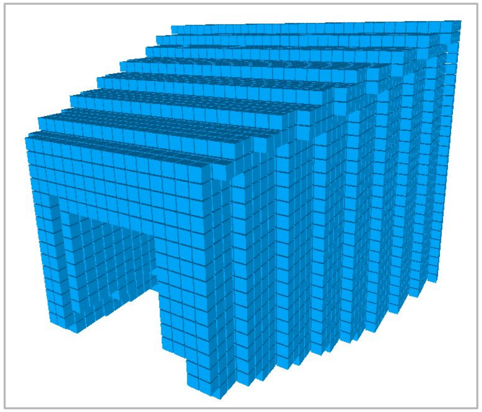


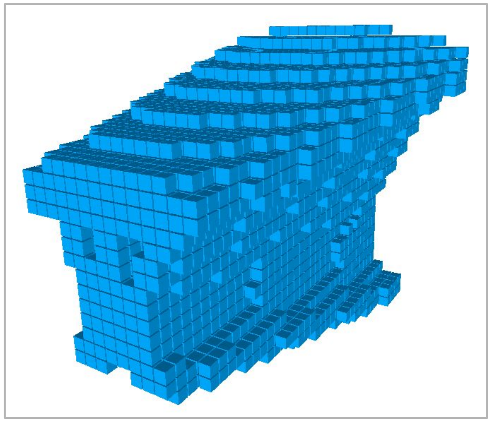


https://github.com/christopherbatty/SDFGen

Data Generation

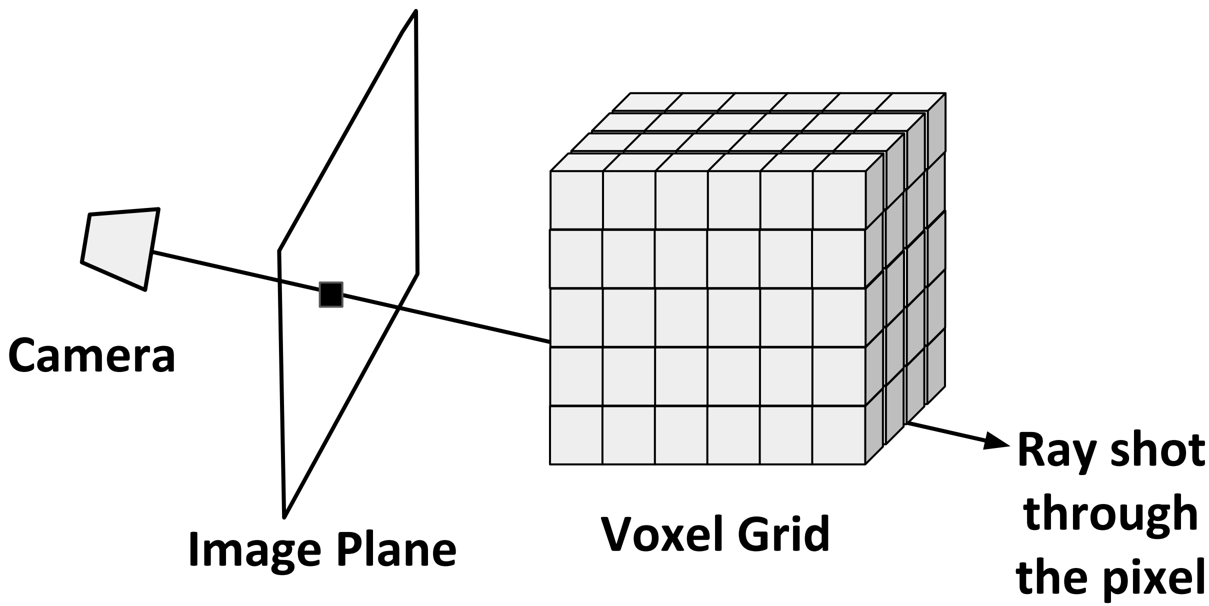
3D Input : Generate Visual Cones from input 2D images





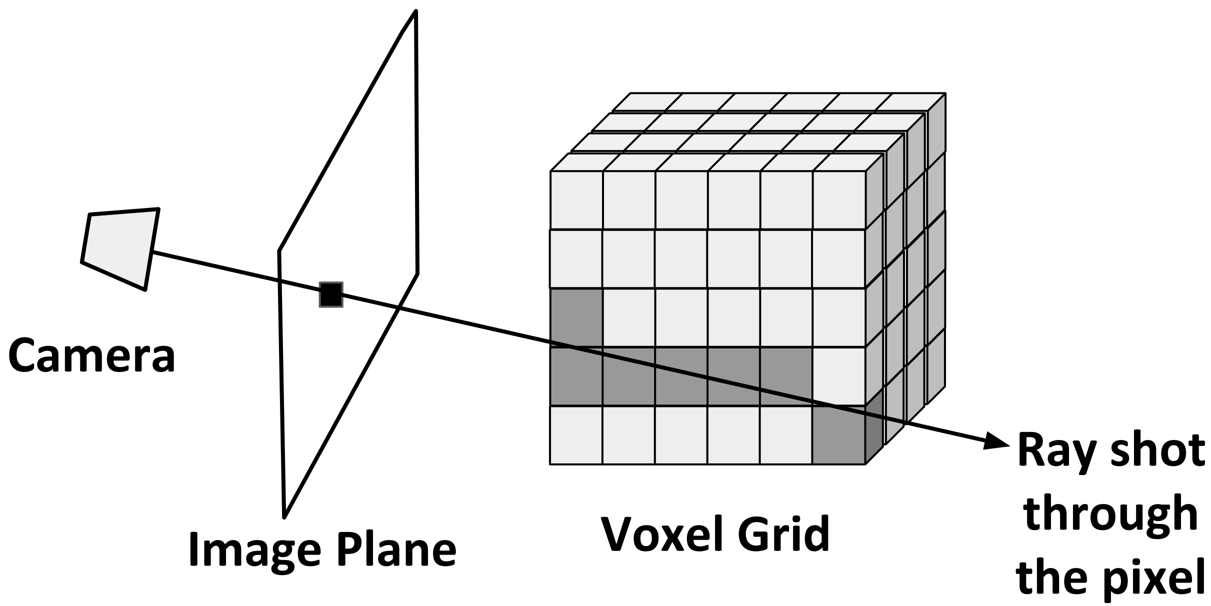
Data Generation

3D Input : Generate Visual Cones from input 2D images



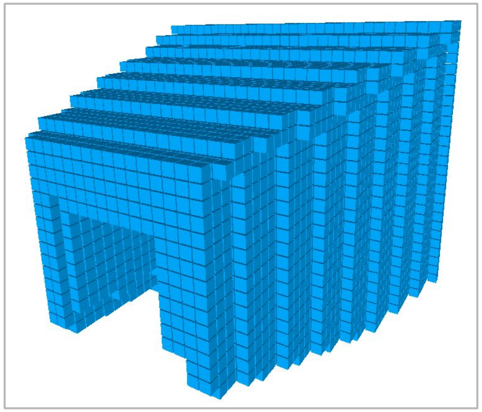
Data Generation

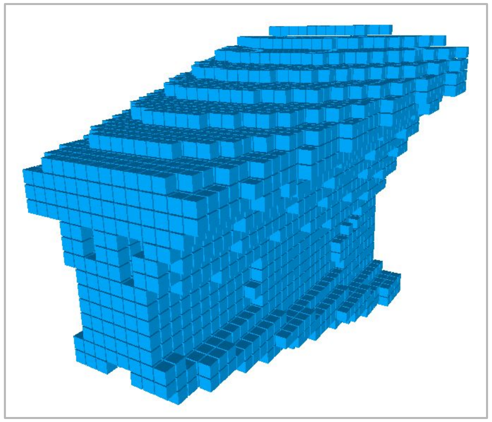
3D Input : Generate Visual Cones from input 2D images



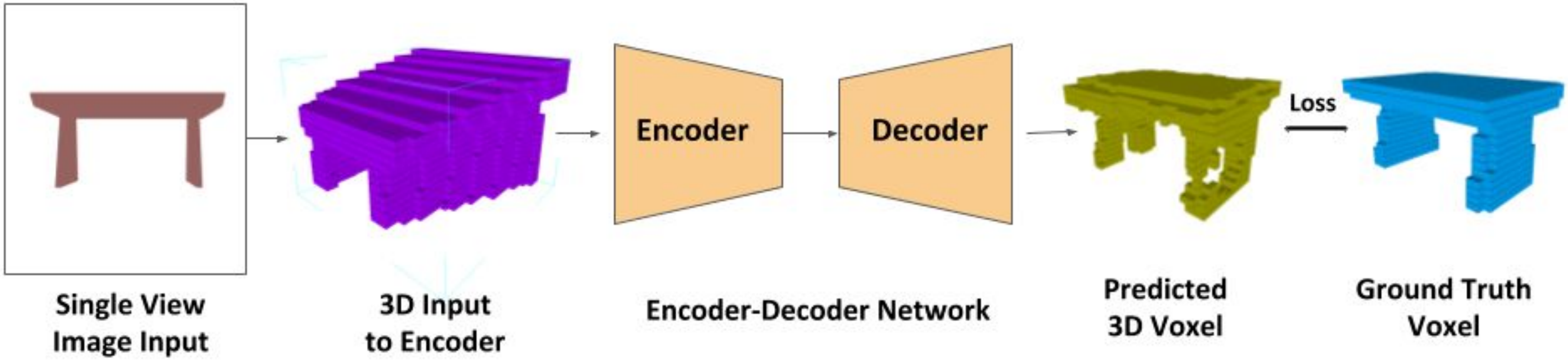
Data Generation

3D Input : Generate Visual Cones from input 2D images

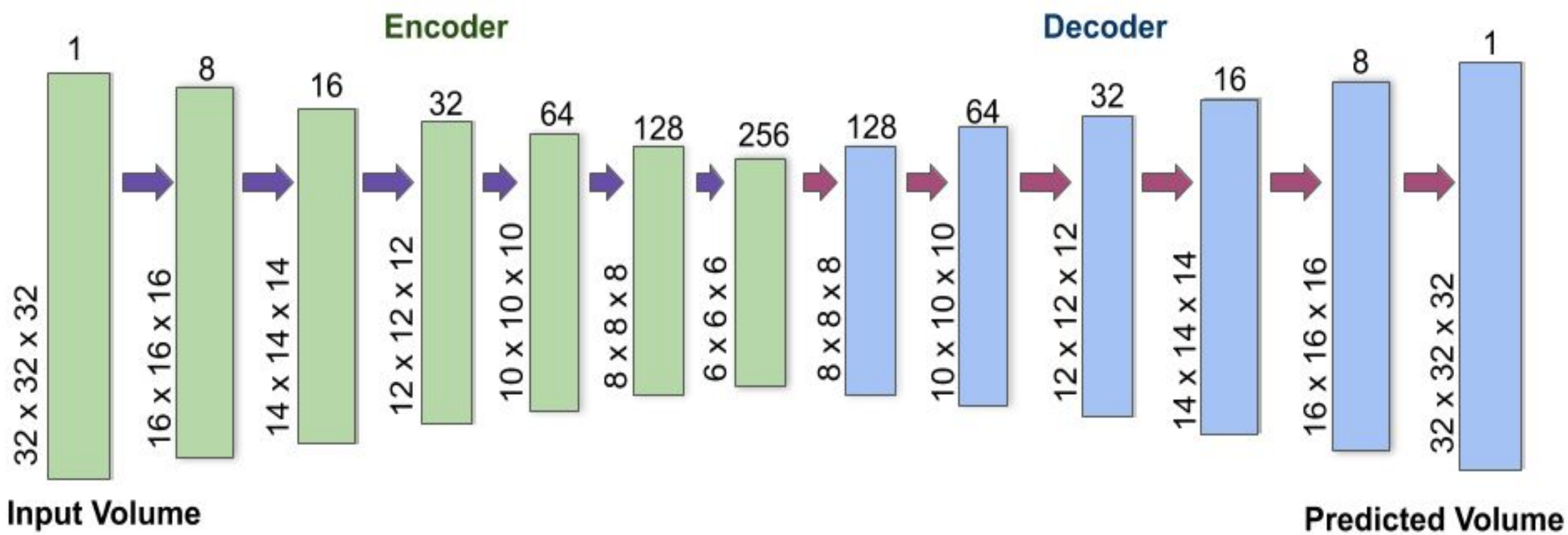




System Pipeline

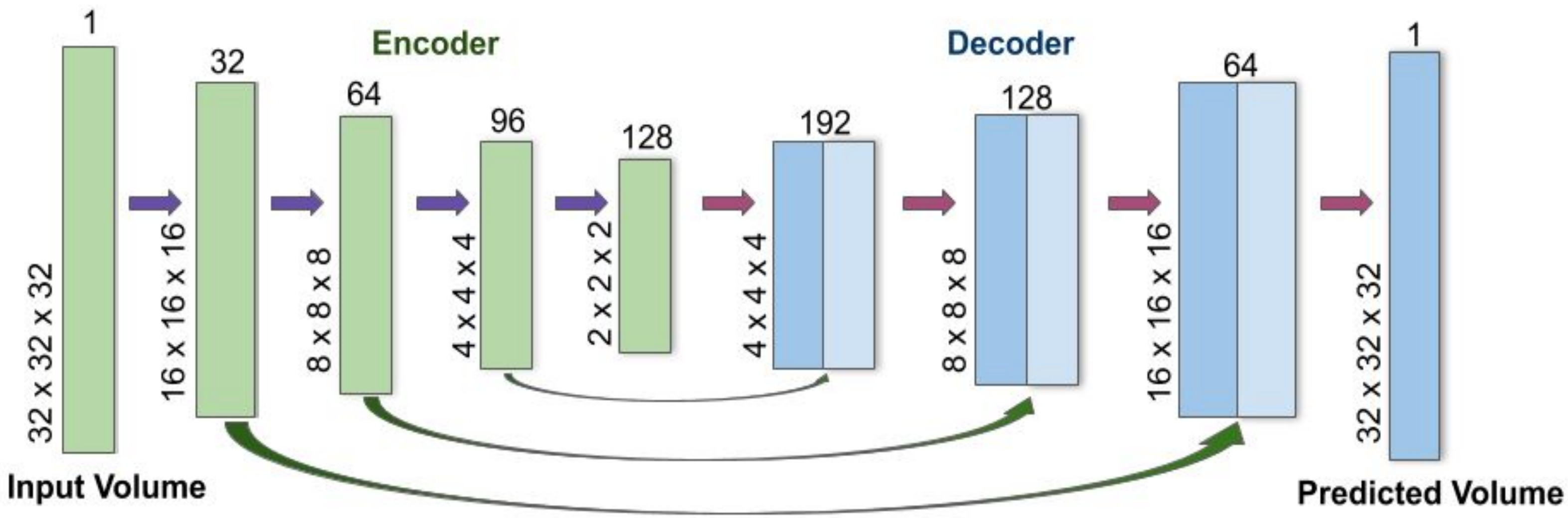


Network Architecture : Net3D



- All layers have Instance normalization and ReLU
  - Except the last layer

Network Architecture: UNet3D

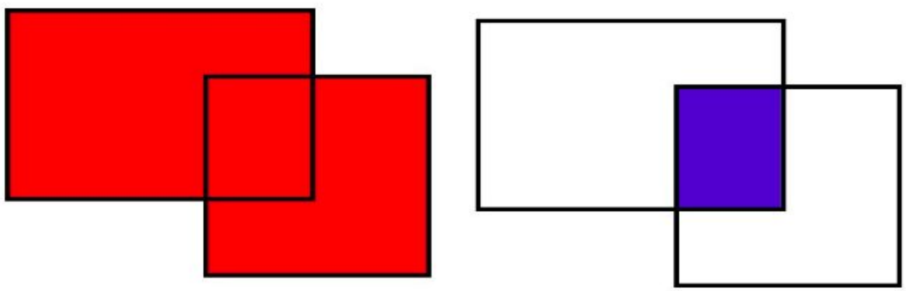


- All layers have Instance normalization and ReLU
    - Except the last layer

Evaluation Metrics

- Intersection Over Union (IoU)

$$IoU = \frac{p \cap t}{p \cup t} = \frac{\sum_{0 \leq i < n} (p_i \geq 0.5).t_i}{\sum_{0 \leq i < n} [(p_i \geq 0.5) + t_i]} \qquad \begin{array}{l} p_i \in [0,1] \\ t_i \in \{0,1\} \end{array}$$



The Red Area is Union

The Blue Area is Intersection

- L1 Error Metric

$$L_1 = \frac{1}{n} \sum_{0 \leq i < n} |p_i - t_i|$$

Evaluation : Occupancy Grid vs Distance Field

	IoU ↑	L1-Error ↓
Occupancy	*0.592*	*0.118*
TDF	*0.583*	*0.114*
TDF(log)	***0.597***	***0.111***

Evaluation : Occupancy Grid vs Distance Field

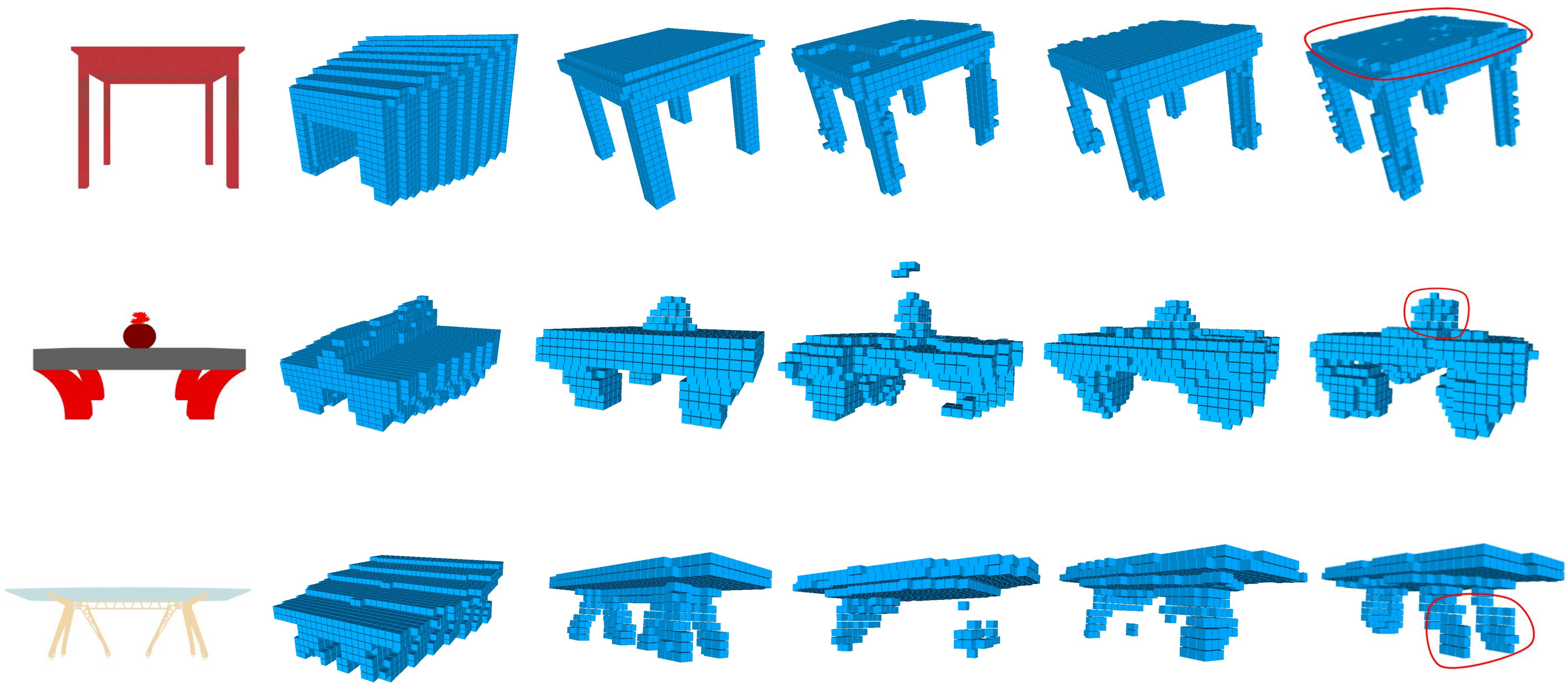
- Distance Field with log weighting performs better than other two versions.
    - more weightage to the voxels near the surface.

- Occupancy representation performs better than Distance Field representation
    - Lesser to learn
    - Input is in the form of occupancy

- In terms of speed, Occupancy representation performs better than other two versions
    - Occupancy: 26 epochs
    - Distance Field with log weighting: 43 epochs

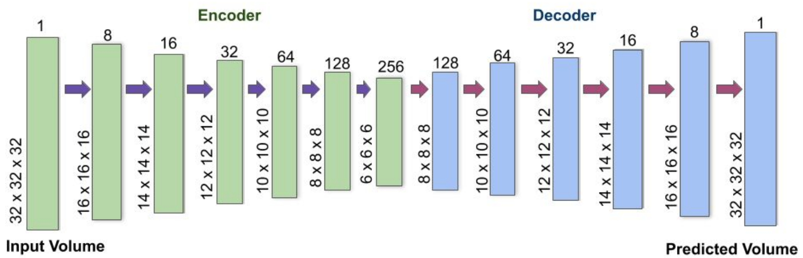
Evaluation : Occupancy Grid vs Distance Field

Input Image | *Input Volume* | *Ground Truth* | *Occupancy* | *TDF* | *TDF(log)*

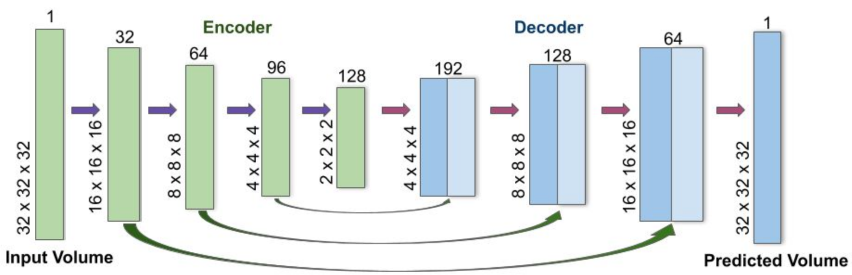


Evaluation : Net3D vs UNet3D

- ➔ Net3D
  - ◆ No Skip Connections
  - ◆ Approx. 2.3 million parameters



- ➔ UNet3D
  - ◆ Skip Connections
  - ◆ Approx. 1.3 million parameters



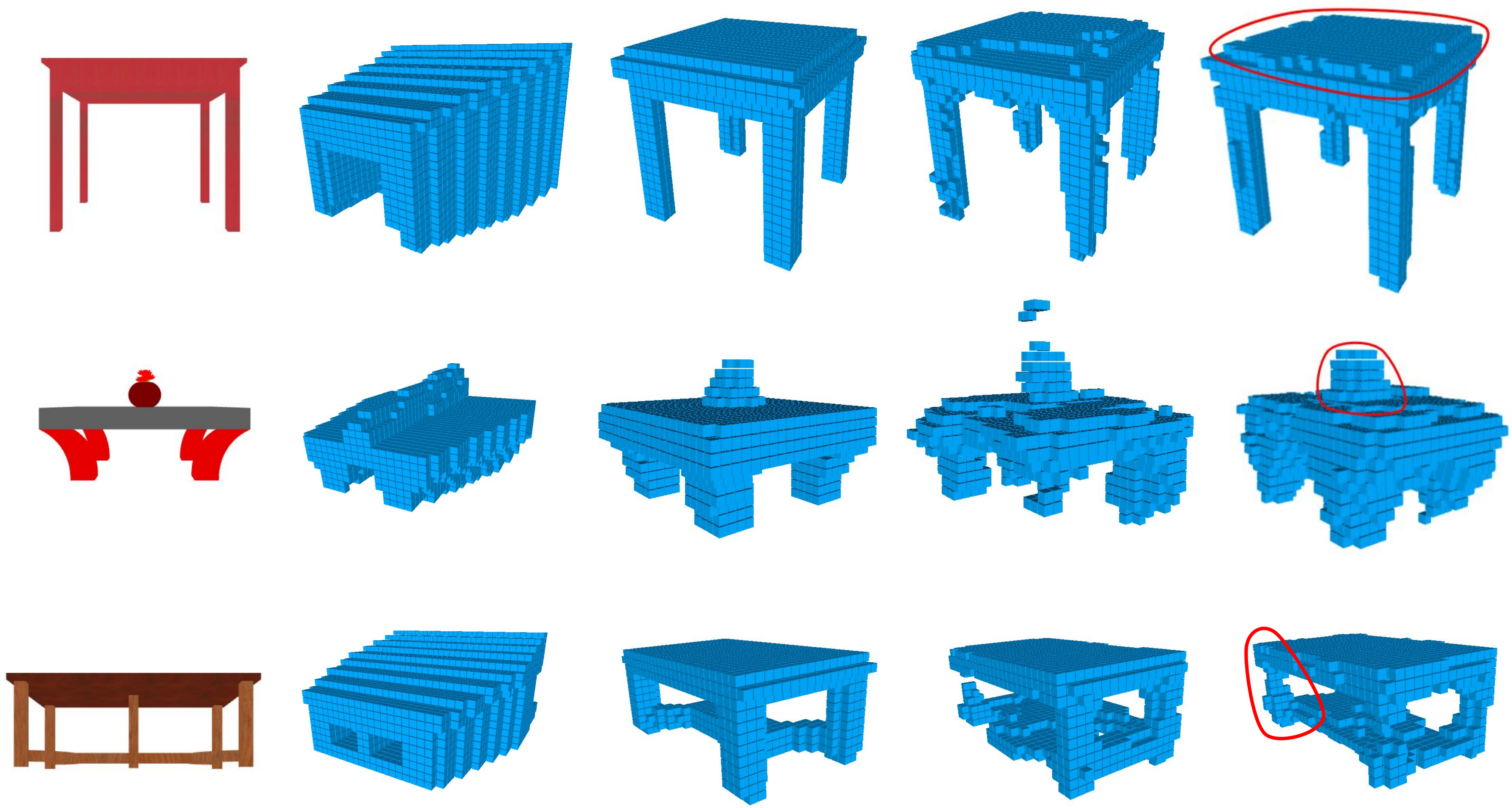
Evaluation : Net3D vs UNet3D

	Occupancy		TDF	
	IoU ↑	L1-Error ↓	IoU ↑	L1-Error ↓
Net3D (2.3M params)	*0.592*	*0.118*	***0.597***	*0.111*
UNet3D (1.3M params)	***0.596***	***0.118***	*0.584*	***0.110***

- UNet3D gives comparable performance with lesser parameters
  - Skip Connections
- In terms of speed, UNet3D performs better
  - UNet3D : 12 epochs
  - Net3D: 26 epochs

Evaluation : Net3D vs UNet3D

Input Image | *Input Volume* | *Ground Truth* | *Net3D* | *UNet3D*

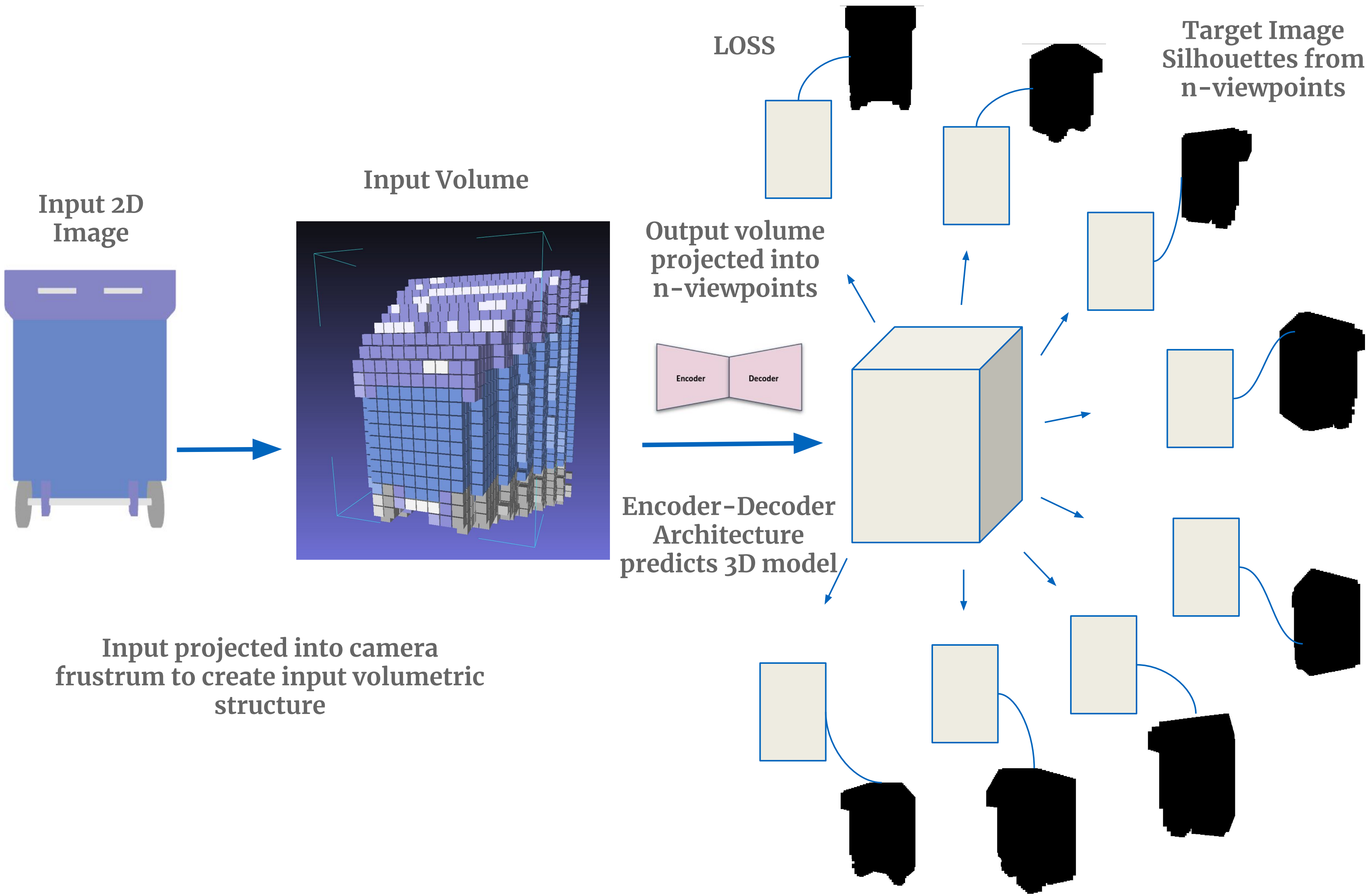


Conclusion

- Occupancy Grid vs Distance Field (DF) Representation
  - DF(log) - Better reconstructions
  - Occupancy - Faster convergence

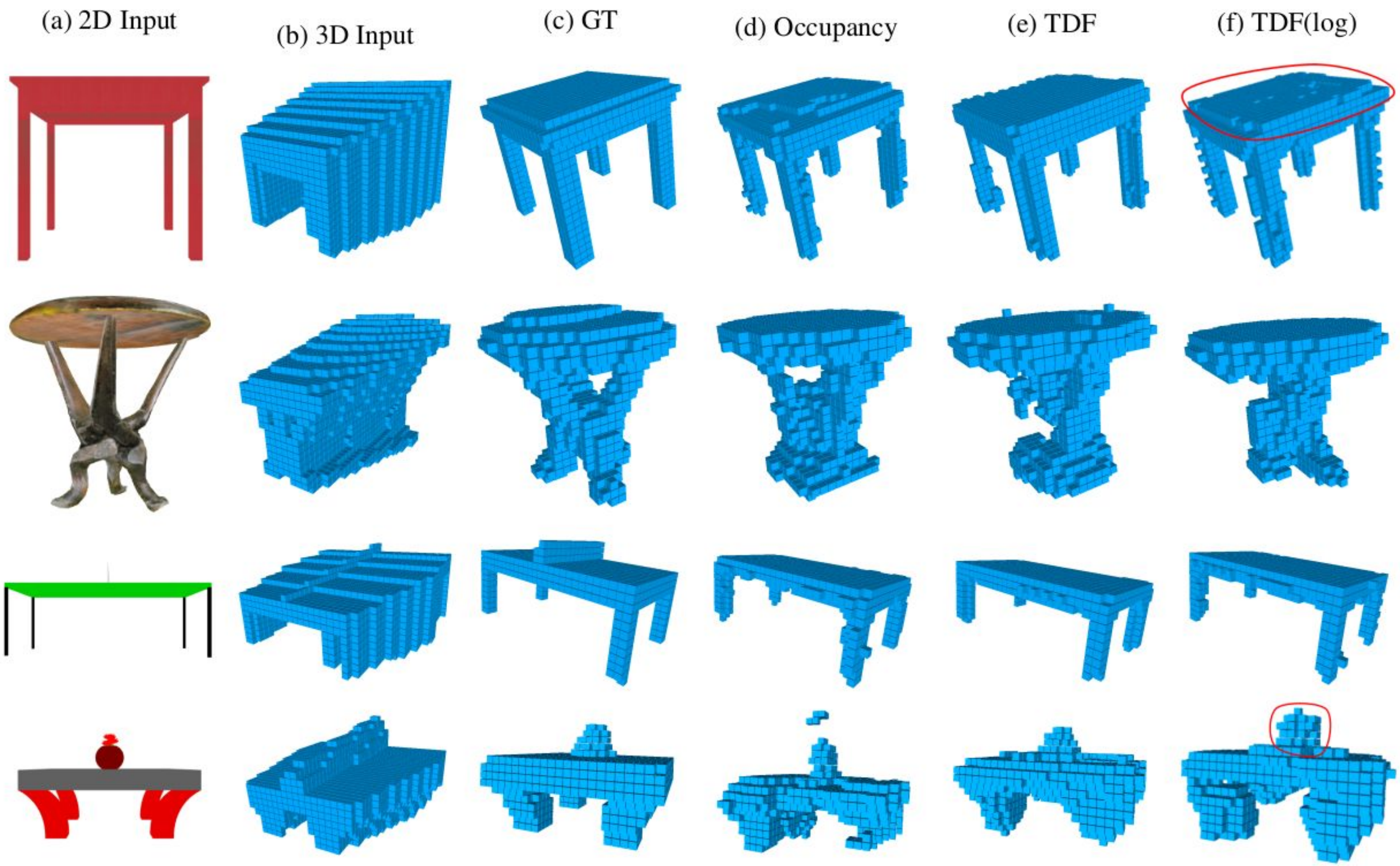
- Net3D vs UNet3D - Skipp connections improve performance
  - Faster convergence
  - Lesser parameters
  - Better reconstructions

Future Work

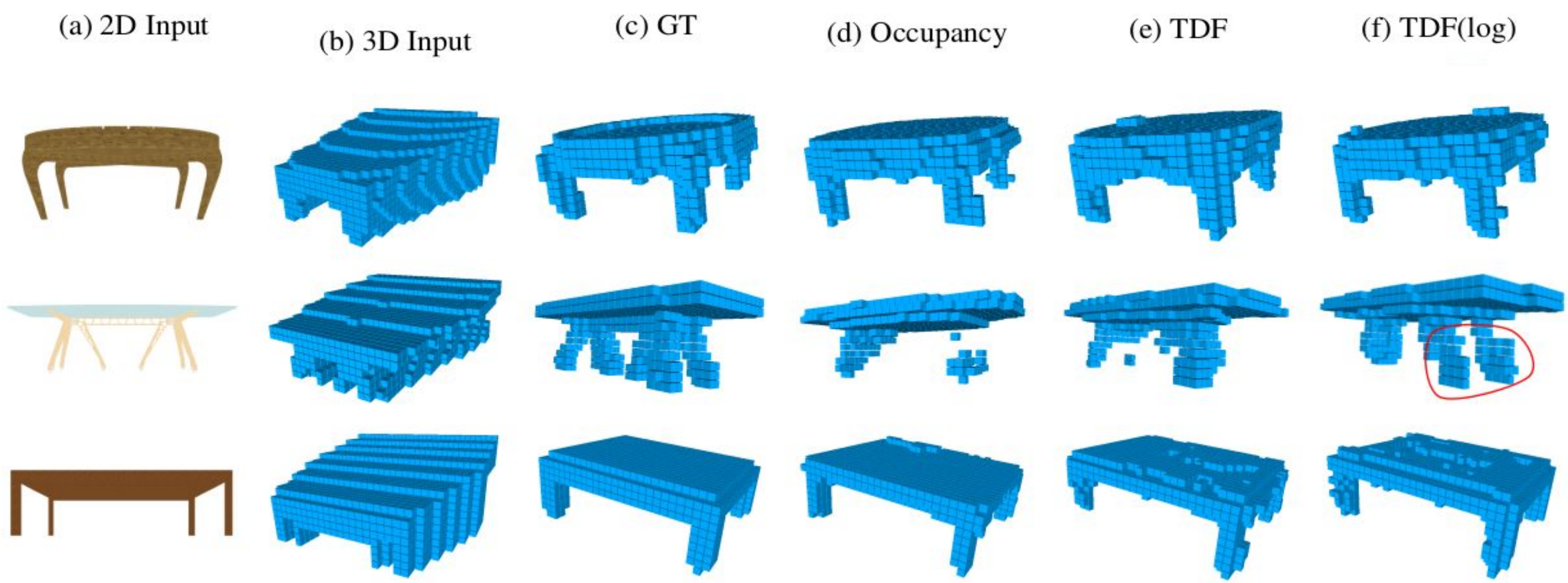

Input 2D Image
Input Volume
LOSS
Target Image Silhouettes from n-viewpoints
Output volume projected into n-viewpoints
Encoder
Decoder
Encoder-Decoder Architecture predicts 3D model
Input projected into camera frustrum to create input volumetric structure

Questions?

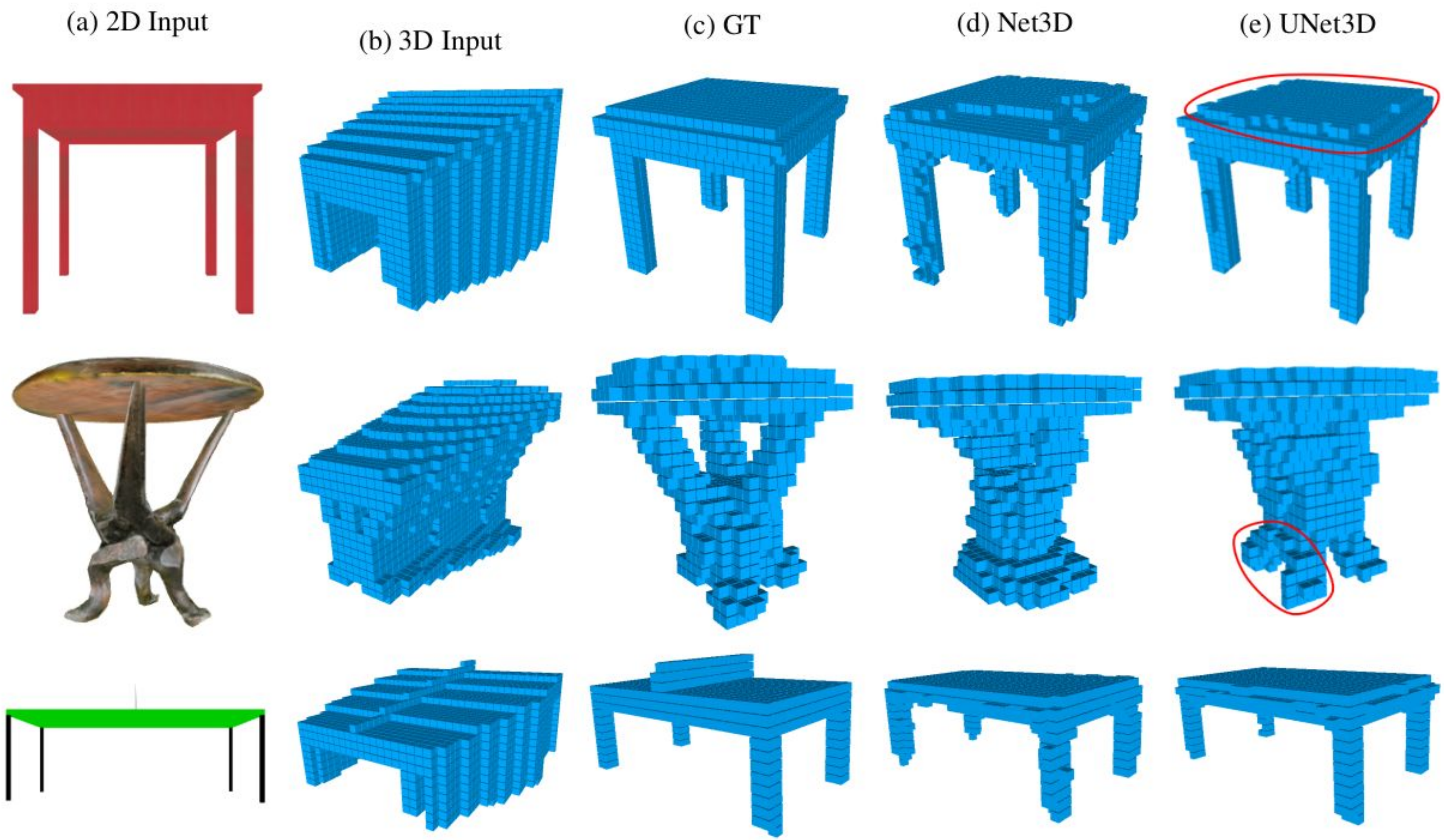
Appendix : Occupancy Grid vs Distance Field



Appendix : Occupancy Grid vs Distance Field



Appendix : Net3D vs UNet3D



Appendix : Net3D vs UNet3D