3D Object Completion with RGB-D Data

Jing Dao Chen, Ching-An Cheng, Eric Wang

Objectives

1. 3D Object Completion: reason about the occluded parts of a 3D object due to incomplete view
2. Economic: low-cost depth image
3. Efficient: single-pass computation without sampling

Introduction

- 3D sensor data has become increasingly useful for perception tasks in both navigation and mapping applications.
- However, the captured data from sensors such as RGB-D camera and Lidar is often suboptimal due to occlusion and incomplete views.
- In the field of autonomous vehicle navigation [1], heavy occlusion is identified as the major problem which causes most pedestrian detection algorithms to fail.
- We approach this problem by performing 3D object completion through training over a large number of objects with known models. This approach has parallels in computer vision for the task of scene completion [2].

Related Work

- Our idea is mostly related to 3D Convolutional Deep Belief Network, 3D ShapeNets [3].
- It represents 3D geometry as a probability distribution over binary variables on 3D occupancy grid. The network is trained from ModelNet, a large-scale object dataset of 3D CAD models, and models the joint distribution of the 3D occupancy grid and the object category.
- The architecture has three 3D convolutional layers, with direct input of 3D occupancy grid data at the bottom, and one fully connected restricted Boltzman machine at the top.
- 3D ShapeNets perform objection completion using Gibbs sampling

Methods

- **Pre-processing**: A 2.5D depth image is represented in a 30x30x30 3D occupancy grid, where each occupancy box is labeled as 1, if it is observed; 0, if it is background; 0.5, if it is occluded.
- **3D Completion Network** : Our model is based on the architecture of 3D ShapeNets [3]. But instead of using Gibbs sampling, here we simplify the computation by performing a simple pass through the network and treating the completion problem as a multiclass classification problem (Figure 2)
  - The input data is 30-30-30 occupancy data representing the objects.
  - A forward convolution network includes 3 convolution layers and a fully connected layer and outputs a 1200-dimensional vector
  - Fully connected layers are used to increase the network capacity in order to learn the completion. In our experiments, we tried three different configurations of fully connected layers.
  - A backward convolution network mirrors the forward convolution network, having a fully connected layer and 3 deconvolution layers.
  - The output is a 30x30x30 3D occupancy grid with value denoting the probability of the object appearing the occupancy boxes.
  - All the activation functions used are sigmoid functions.
- **Output Masking**: An output mask is applied to the network output to render the object completion result based on the input occupancy grid which has correct information of the observed and background parts. The output mask extracts information only from the occluded part of the input to perform object completion.

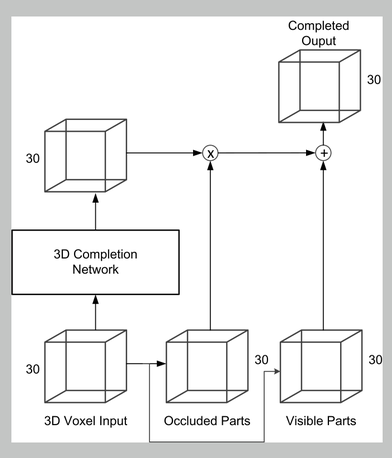


Figure 1: Object Completion

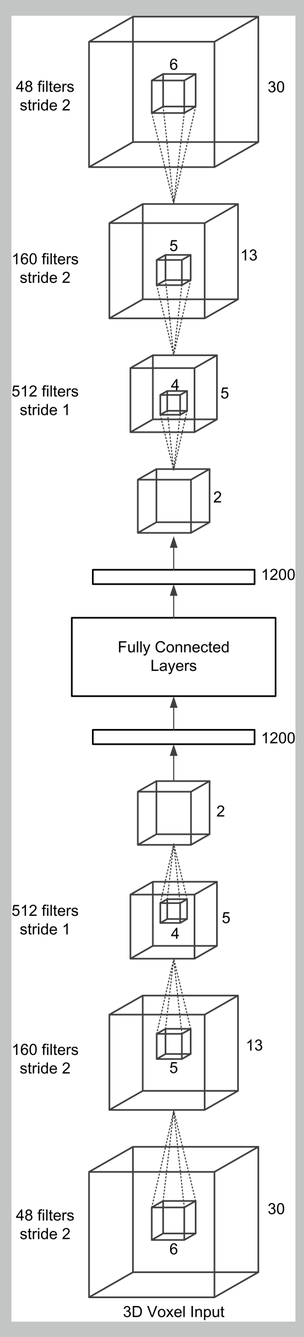
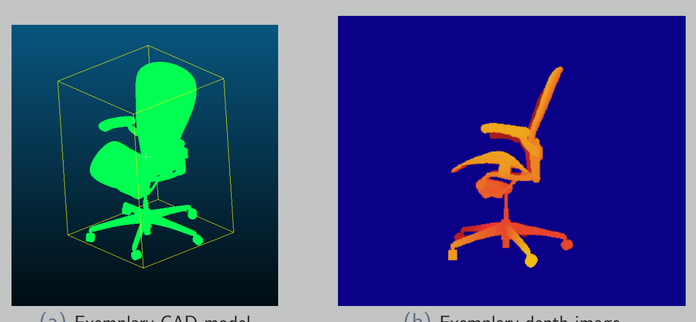


Figure 2: 3D Completion Network

Experiments

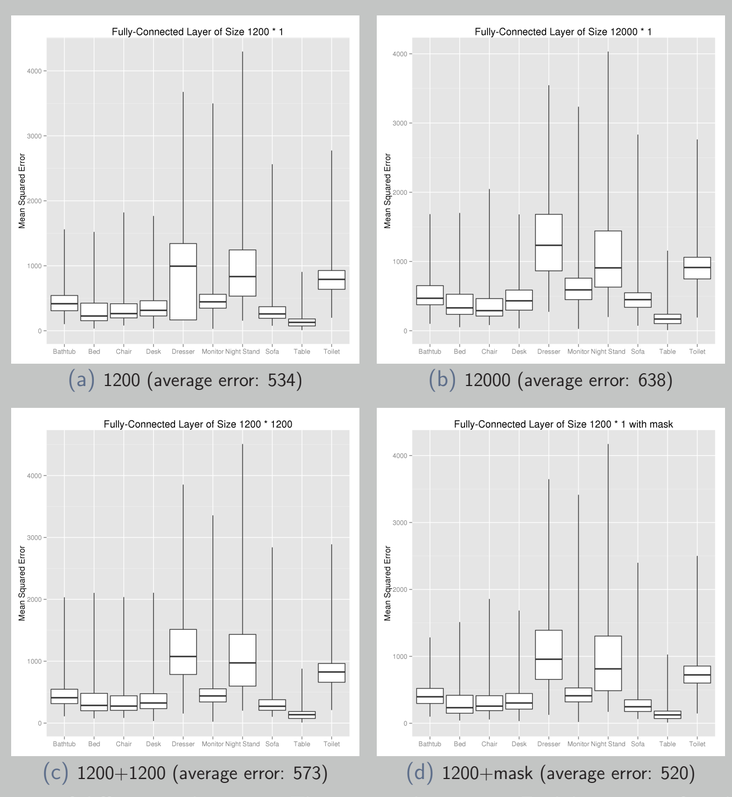
- Starting with 3D CAD models, we generate synthetic depth images and its corresponding occupancy grids representing partial view observations as the input data, and use the complete occupancy grid as the ground truth.
- We experimented with three architectures of the fully-connected layers: a single layer of 1200 hidden neurons, a single layer of 12000 neurons, and two layers of 1200 neurons. The error is calculated as the Euclidean error between the 3D Completion Network output and the ground truth. For the network with a single layer of 1200 hidden neurons, we also experimented with training the network using the error after output masking.
- The models were trained on the *table* category alone (4707 depth images), and tested on another test set (10896 depth images, in which 1200 are tables).
- Python and Caffe was used in the development of our project. We used an AWS EC2 g2.2xlarge instance equipped with NVIDIA GRID K520 GPU to experiment with different architectures of the neural network.



(a) Exemplary CAD model (b) Exemplary depth image

Figure 3: Example of dataset

Results: Table

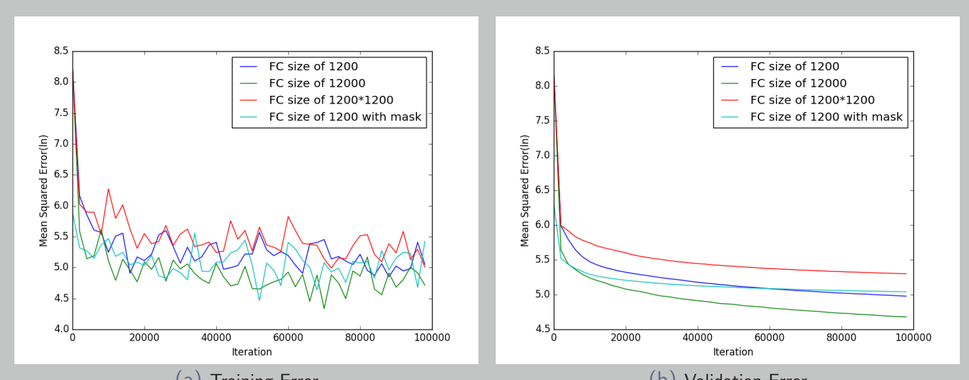


(a) 1200 (average error: 534) (b) 12000 (average error: 638)

(c) 1200+1200 (average error: 573) (d) 1200+mask (average error: 520)

Figure 4: Testing error of different architecture. The error is the average Euclidean error after the output masking

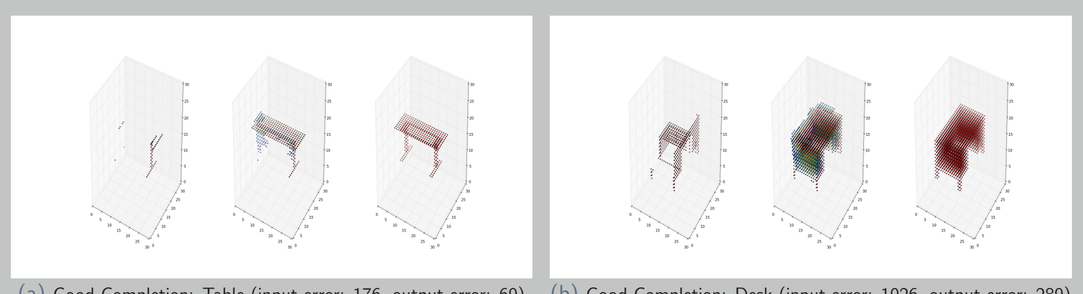
Results: Figure



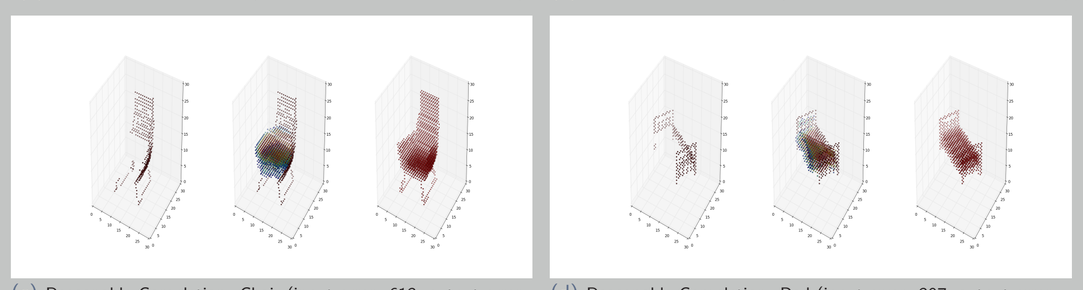
(a) Training Error (b) Validation Error

Figure 5: Training/Validation Error

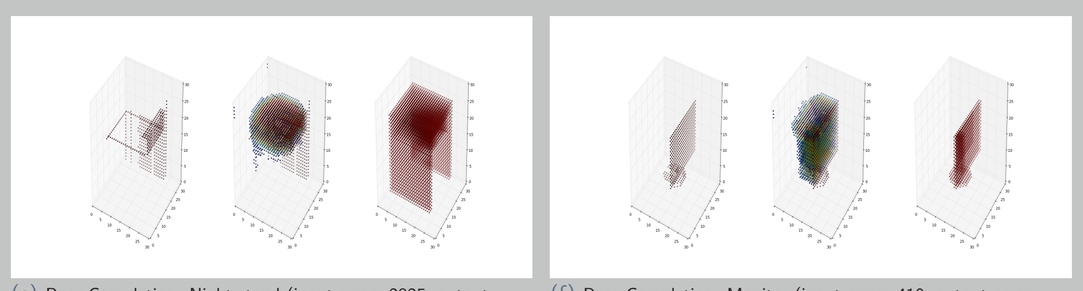
Results: Object Completion



(a) Good Completion: Table (input error: 176, output error: 69) (b) Good Completion: Desk (input error: 1026, output error: 289)



(c) Reasonable Completion: Chair (input error: 613, output error: 387) (d) Reasonable Completion: Bed (input error: 397, output error: 231)



(e) Poor Completion: Night stand (input error: 2825, output error: 1721) (f) Poor Completion: Monitor (input error: 410, output error: 636)

Figure 6: Object completion of network with 1200 neurons in the fully connected layers

Discussion

- Experimental results
  - The best architecture in terms of test error is the one with a single 1200-hidden-units fully-connected layer. Larger/deeper networks though perform better in training, but do not generalize as well.
  - Output masking improved completion result.
  - The network performs well to complete object in similar classes, such as desk and chair, to table.
  - The network performs poorly on dissimilar class, such as a monitor, since the smooth prior learned from the table category does not carry over.
- Future work
  - Train the network with output masks directly: End-to-end training
  - Train the network with data overall all categories

References

[1] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona.
Pedestrian detection: An evaluation of the state of the art.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(4):743–761, 2012.

[2] James Hays and Alexei A Efros.
Scene completion using millions of photographs.
ACM Transactions on Graphics, 26(3):4, 2007.

[3] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao.
3d shapenets: A deep representation for volumetric shapes.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.