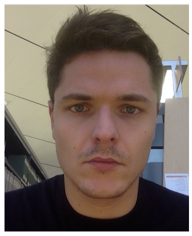
3D Object Recognition with Deep Networks

3D Photography Project Proposal
Supervised by: Martin Oswald, Pablo Speciale
March 7, 2015

GROUP MEMBERS

Tobias Grundmann Adrian Schneuwly Johannes Oswald

I. DESCRIPTION OF THE PROJECT

The goal of this project is to successfully implement 3D object recognition using a Convolutional Neural Network (CNN) approach [3]. Such networks can be trained to recognize complex 3D shapes and determine the corresponding object class given a volumetric representation. This project will mainly use the 3D CAD model data set called ModelNet [4]. Conceivably, volumetric data can also originate from inexpensive 2.5D depth sensors like Microsoft Kinect, Google Project Tango or Intel RealSense (Fig. 1).

II. WORK PACKAGES AND TIMELINE

A. Prerequisites in March

The first goal of the team is to fully understand the approach described in [3]. Therefore everyone in the team works through the Udacity deep learning course [2] individually in order to gain knowledge and hands-on experience with deep networks using 2D data. Further, we get familiar with TensorFlow[1] (in Python), which is used to implement the network architecture. As a first simple exercise to approach the problem we reproduce the character recognition example [1] using the MNIST data set. In order to gain experience for the 3D case, we further extend that example and make it rotation invariant.

The division of work among the 3 project members will happen after this initial project phase, because at this point in time it is hard to foresee how the work could be split meaningfully.

[1]open source library for machine intelligence

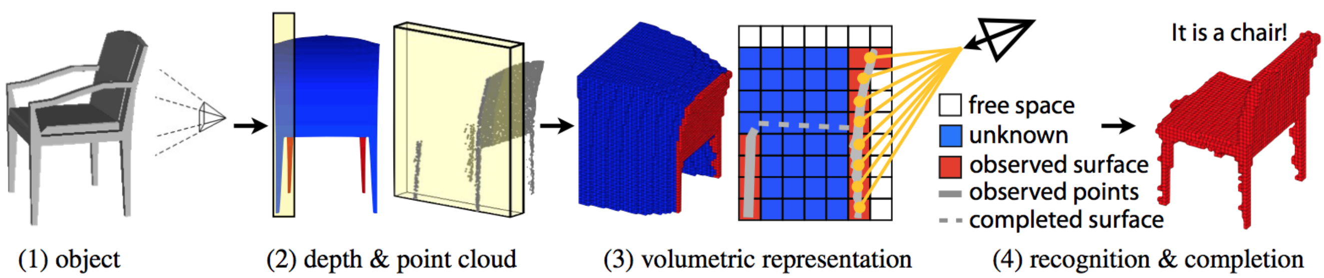


Fig. 1. 2.5D object recognition

B. Modeling & Data Preparation in April

We begin implementing the VoxNet Convolutional Neural Network architecture [3] in Python while using the framework provided by the Udacity deep learning course [2] as a base. Once the basic network is performing well, rotation invariance is added using the approach described in [3]. The input to the network is a small voxel volume (30x30x30) and the output is an object class. Part of the ModelNet is already provided in volumetric representation. If necessary, additional CAD models can be converted to a voxel data structure using the Matlab script provided by [4].

C. Training & Testing in June

After the successful implementation, we train the deep network with the ModelNet40 data set, which consists of 40 common categories with 100 unique CAD models per category (voxel format). The data is split into two parts, namely a training and testing set. For the network training only the training set is used. In this phase we fiddle around with network parameters while constantly evaluating the performance. Due to the computationally expensive nature of the training, we expect this process to be time consuming.

III. Outcomes and Demonstration

Goal of the project is to build up the VoxNet [3] architecture and train the network to recognize objects. We hope to achieve similar positive recognition results as described in the paper [3]. In the live demo, we feed the network with data from the testing set (ModelNet40) or possibly real Google Tango data and then try to classify the objects.

References

[1] https://www.tensorflow.org/versions/r0.7/tutorials/mnist/beginners/index.html.

[2] https://www.udacity.com/course/deep-learning–ud730.

[3] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. *International Conference on Intelligent Robots and Systems (IROS2015)*, 2015.

[4] Z. Wu, S. Song, A. Khosla, L. Zhang F. Yu, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, 2015.