Object Detection in 3D Scenes Using CNNs in Multi-view Images

Ruizhongtai (Charles) Qi
Department of Electrical Engineering
Stanford University
rqi@stanford.edu

Abstract

Semantic understanding in 3D scenes will become an important topic for future VR and AR applications. Our work explores object detection in 3D scenes. Instead of following traditional methods that directly detect objects in 3D with hand-crafted features and assumptions that 3D models exist for observed objects, we lift 2D detection results in multi-view images to a common 3D space. By taking advantage of the state-of-the-art CNN (Convolutional Nerual Nets) detectors and aggreate information from hundreds of frames in 3D, we can robustly detect objects and generate a heat map showing probabilities of object existence in each spatial location.

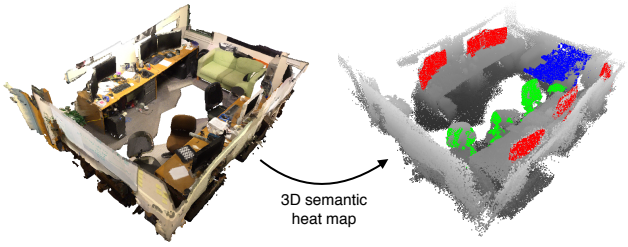

Figure 2. Our algorithm can generate a semantic heat map of a 3D scene (only points in voxels above a score threshold are shown). Left: point cloud of an office room. Right: semantic heat map showing three categories, where monitors are in red, chairs in green and sofas in blue. The heat map can enable instance segmentation, 3D bounding box detection and 3D/2D alignment.

1. Introduction

Recently, we have witnessed a lot of industrial momentum in commercializing virtual reality and augmented reality. Head mounted devices (HMD) like Microsoft HoloLens are expected to have huge impacts in entertainment, design and medical treatment. While a lot of R&D focus is on low-level computer vision (e.g. localization, reconstruction) and optics design, we believe for powerful future VR and AR applications, *semantic understanding* of 3D environment will play a very important role.

Without semantic understanding, the computer generated world is actually separate from the real world. To support interaction of the objects/agents from both worlds, computer needs to assign "meanings" to the pixels and points. For example, to create a ball rolling in a room, we have to know where the floor is. If we want to create a virtual character sitting beside us on a sofa, we have to know where the sofa is in the room and which part of it is for sitting. If we want to make a horror movie in AR to have a ghost coming out from a screen, the algorithm should figure out whether there are any monitors in the room. All of these applications require semantic understanding.

It's easy for the virtual world since everything is generated and all its properties (like category, shape, material etc.) are known, while it can be very challenging for the real world. This project will focus on how to give meanings to real data, i.e. to fill the semantic gap. Specifically, we want to achieve *object detection in 3D scenes*.

While traditional object detection algorithms are available for RGB images, they are not robust enough and cannot directly be applied to 3D cases. On the other side, mainstream object recognition methods on point clouds are not data-driven. Usually, keypoint detection (using hand-crafted features), RANSAC and ICP methods are used and they rely heavily on the assumption that a very similar model or point cloud to the object in real scene is available, which is usually not the case for generic scenes. Our method will avoid drawbacks of these two approaches but take their strengths. We use an approch that takes advantages of (1) advances in CNNs for understanding RGB images and (2) 3D as a coherant information aggregation space. By applying object detection on RGB images, back-project detection scores to 3D voxel grids and post-filtering and global adjustment, we are able to achieve robust object detection in 3D scenes. In Section 3 we present details of the algorithm and in Section 4 we show output results of each step of the pipeline.

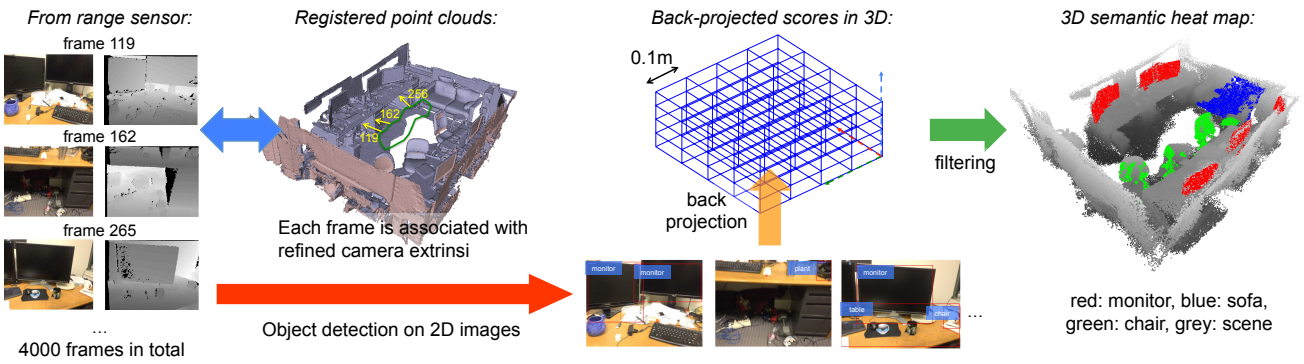

Figure 1. **Detection Pipeline.** Input to the system is RGB-D frames from scanning. Each of the frame has known camera pose. We first apply object detectors on RGB frames and then back-project scores into a coherent 3D voxel grid space. After filtering, we obtain a 3D semantic heat map.

2. Related Works

Object Detection in 2D and 2.5D In computer vision, object detetion is often formulated as finding rectangular bounding boxes for objects in certain categories. Some of the recent best detection system consist of R-CNN [1] and Faster-RCNN [5]. Detectors of similar flavor have also been developed for depth images [2], which treat depth data as 2D images with differnt channels then RGB. There methods are limited on 2D images and do not have robust outputs. Our problem differs from them since we desire very robust object detection directly in 3D space.

Object Detection in 3D There is a growing interest in SLAM (Simultaneous Localizaiton And Mapping) community in acquring semantics for their data. In some recent works including SLAM++ [7] and objection recognition with monucular SLAM [4], researchers tried to achieve object detection along with SLAM. However, they either depend on having exactly the same models to the object in the scene or rely on hand-crafted features. Our method has no assumption on having similar models and makes use of CNN learned features.

3D Mesh Classificaiton Recently, there are some works on semantic understanding of 3D data in terms of mesh classification using CNNs [9][8]. The best performing method MV-CNN is still based image CNNs. Our method can be viewed as an attempt to apply MV-CNN to 3D scenes through back-projecting 2D detections into coherent 3D voxel grids.

3. Object Detection Pipeline

As shown in Fig 1, input to our system is a series of RGB-D images of a 3D scene with camera poses for each of them, which can be obtained by a reconstruction pipeline. The output is a heat map for probabilities of appearance of each categories (like sofa, chair, monitor etc.) in 3D voxel grids. On top of the heat map we can separate object instances, fit 3D bounding boxes or align 3D model to the sensor data.

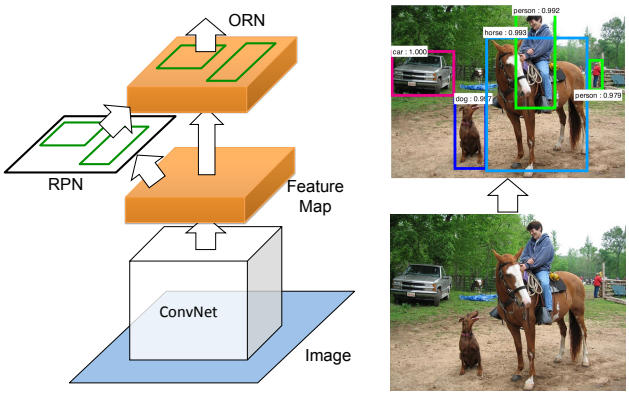

Figure 3. Illustration of the faster-RCNN [5] network structure (left cartoon) and its examplar input and output (right images).

Our method first uses the state-of-the-art object detector on RGB images to achieve bounding box proposals along with their scores (note that there are a lot of false positives and false negatives in this stage). Then since we have camera pose and depth image for each frame, we can *back project* the proposal into 3D voxel grid (from image coordinate to camera coordinate and then to world coordinate). All points in the bounding boxes within the depth image are projected to their corresponding voxel grid, where information is aggregated. Then we conduct statistical filtering and global adjustment (by excluding object collisions) to achieve a clean and final heat map. For details of each step, see the following subsections.

3.1. Object Detection in RGB Images

In Faster-RCNN [5], given an input RGB image, it's able to detect objects of certain categories in the image by placing a rectangular bounding box over the objects. Convolutional neural networks are used for proposing object locaitons (bounding box) in the image through a Region Proposal Network (RPN). Another network called Object Recognition Network (ORN) is used to classify the object in each bounding box as either background or valid object category. RPN and ORN will share some lower convolution layers. An illustration of the faster-RCNN network is shown in Fig 3. There are 20 valid categories used, which are from PASCAL VOC dataset, in which the relevant categories we will look at include monitor, chair, sofa, pottedplant and bottle.

Although faster-RCNN has achieved state-of-the-art performance on several benchmark dataset, real use reflects that it is still not robust enough. As shown in the mid-bottom images in Fig 1, the earphone is wrongly detected as plant and a cup is detected as a chair, and the bounding box for monitors are not accurate. In summary, there are two common cases in the detectin results (all detection results are provided through donwloadable links): false positive (wrong proposal), false negative (miss the object). We find it's difficult to locate objects in an image frame with limited field of view, especially for objects that are truncated or occluded or in rare viewpoint that's not covered enough in training data. In next subsection, we will discuss how to use these inaccurate proposals from images to achieve a robust detection result in 3D.

3.2. Back Projection to 3D Voxel Grids

Since we know camera pose for each RGB-D image frame, we can back-project all points in the depth image back to 3D. Obviously we can project other meta data along with the points where the most naive case is color information. Here we will project object detection scores/confidence instead of colors.

$$\begin{aligned} x &= \frac{(u-uc)z}{f \cdot D} \\ y &= \frac{(v-vc)z}{f \cdot D} \\ z &= z \\ p' &= [x', y', z', 1]^T = [R|T][x, y, z, 1]^T \end{aligned} \tag{1}$$

For each bounding box proposal from 2D object detector mentioned in Sec 3.1, we set a threshold for confidence, if the confidence score is higher then the threshold λ, all points in the depth image that fall into the bounding box are associated with the proposal category and score. Then we use camera intrinsics and extrinsics to compute the 3D world coordinates of each point. In Eq 1, u and v are coordinates of pixels in image, uc and vc are coordinates for center pixel, f is focal length, D is conversion from depth value to real depth and z is depth value in depth images. x, y, z are coordinates as to the camera and x', y' and z' are point coordinates in 3D world.

$$VS(i,j,k,c) = \frac{\sum_P S(I(p),c)1[V(p)=(i,j,k)]}{\sum_P 1[V(p)=(i,j,k)]} \tag{2}$$

To aggregate detection proposals in 3D we discretize the 3D space into voxel grids with fixed dimension $a \times a \times a$. A typical value for a is 10cm. Then we can accumulate proposal scores from points in all frames. To compute scores for voxel at position i,j,k for category c as in Eq 2, where P represent all points in all frames and function V is mapping from point to its belonging voxel index, S is mapping from point and category to its associate score. There are several variations in aggregation method.

(1) Looking at positive signals only. For this method, different from denominators in Eq 2, we only count for points within proposal bounding boxes, which result in a smaller denominator. We will see in experiments section that this method is not ideal since both positive and negative signals are worthwhile to capture.

(2) As in current Eq 2, we count denominator for both positive and negative points, as long as we observe a point in the voxel the denominator will count one. In this way, if we have two objects, one is classified as chair everytime it appears while the other is wrongly classified as chair for only one time among its 100 apperances but with a relatively high score. Using previous counts, the average voxel score might be similar, but if we count negative point as well, the first object will have much higher score, which is what we want.

(3) As to the accumulation of score, we may want to apply a non-linear operation such as taking exponential of the score to amplify weights of confident proposals.

3.3. Filtering and Global Adjustment

Even we can aggregate detection results from hundreds of images, there are still much noises due to the limit of viewpoint coverage and 2D detectors. Here we introduce two approaches (they can be used together) to post-process the aggregation results in 3D space. The first one is applied to scores in each category and the second is applied to scores from all categories jointly.

The first approach is statistical outlier removal [6]. The algorithm has two passes. In the first pass, it computes the average distance of each point to its nearest K neighbors. Then the mean μ and standard deviation δ of all these distances are computed. Then we can set a threshold $\mu + n\delta$

where n is a parameter specified by user, such that in second pass all points with distance below the threshold will be classified as inliers.

The second approach is based on the observation that two object cannot occupy the same voxel if the dimension of the voxel grid is approaching zero. In an approximate case, we can say in a setting with reasonably small voxel (e.g. 10cm dimension), each voxel only belongs to one object. Therefore, we can filter in the global 3D space for all categories by picking up the category wtih highest weighted score in each voxel. Thus outliers that in the overlapped regions will be resolved.

4. Experiments

In this section, we first introduce the dataset we used. Next we present the output 3D semantic heat map from our algorithms and compare different methods for generating it. At last, we show some applications that can be built on top of the heat map, including instance segmentation and 3D/2D alignments.

4.1. Dataset

In thie project, We use a scanning dataset containing RGB-D scans of an office[1]. There are 4003 frames in total from 4 indivisual trajectories. There is also a reconstructed point cloud with colors by method in [3]. Camera poses of frames are globally optimized. For our experiments, we use a random subset of 1000 frames. For a glimpse of the dataset, see Fig 1.

4.2. Qualitative Results of 3D Semantic Heat Maps

Back projection. In Fig 4, we see a visualization of aggregated scores in 3D voxel space for monitor category. Although at this stage there are still lots of noises, we can already see a strong pattern of where the monitors could be. This voxel grid map is the result of projecting scores from 1000 images into the common 3D space.

Effect of different voting methods and filtering. In Sec 3.2 we mentioned there are three candidate methods for score voting in voxel grids. Fig 5 compared them, where we can see that introducing counting of negative signals (points that are not detected as any category), we can successfully eliminate large amount of irrelavant points such as points on walls and desks. Taking exponential on the score slightly improve the result. The bottom right image in the figure shows the output after a statistical outlier removal algorithm with $K = 50$ and $n = 1$, we see that most outliers are removed.

Multi-category result and global adjustment. In Fig 6, we show points of three categories (monitors in red, chairs in green and sofa in blue) in the common 3D space. To achieve that we used results from plane detection and removed the points on the floor and desk plane. Also we gurantee that there is no collision among the three categories.

[1]dataset provided by Matthias Niessner

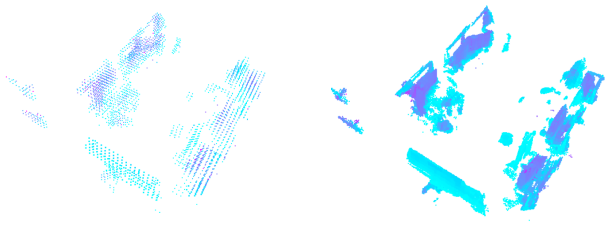

Figure 4. Left: visualization of scores of monitors in 3D voxel grid. Right: visualization of points in those voxel grids. The darker the color the higher the score.

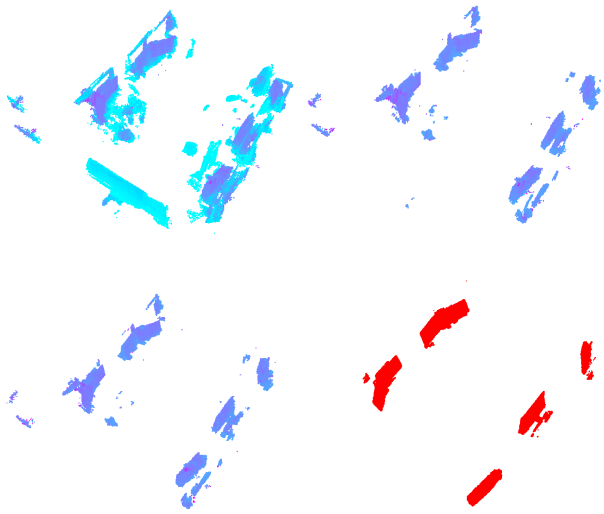

Figure 5. Improvement of using negative voting, exponential score and statistical outlier removal (for monitors). Top left: only count positive singal. Top right: count negative signal as well. Bottom left: based on top right, use exponential score. Bottom right: based on bottom left, use statistical outlier removal.

4.3. Applications of 3D Semantic Heat Map

Given a 3D semantic heat map, we can develop many interesting applications on top of it. Here we will demonstrate two of them: instance segmentation and 3D model alignment. Note that the algorithm here is semi-automatic and mainly serves the purpose for presenting the opportunities of using our pipeline's output.

In Fig 7, we take chairs for example. By K-means clustering based on the coordinates of the chair points in voxels that pass a score threshold, we can cluster chair points into four clusters which correspond to four individual chairs (shown in different colors). Note that K-means is sensitive to initialization, so we can randomly re-initialize sev-

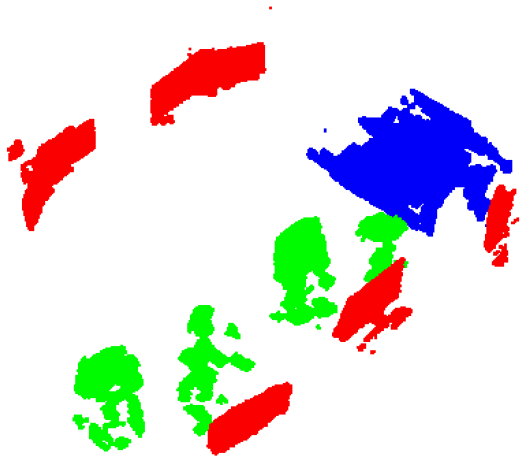

Figure 6. Visualization of heat map of chairs, sofas and monitors. Instead of showing probabilities we set a threshold to voxel scores and present points in voxels that pass the threshold for the three categories.

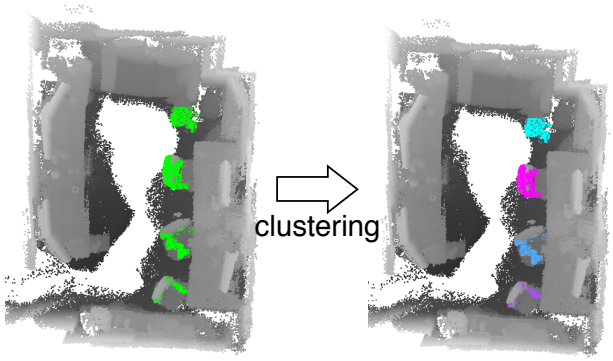

Figure 7. Application of instance segmentation (for chairs).

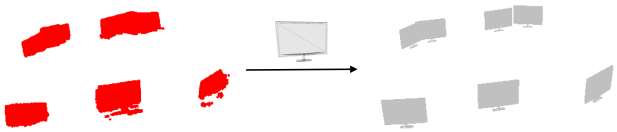

Figure 8. Application of 3D alignment (for monitors).

eral times and find the solution with minimum error. Also, it could be tricky to determine the number of clusters. For the chair example here we are able to try different number of clusters and the errors will be very different. But for the case of monitors as in Fig 6, when two monitors attach to each other, it's difficult to get them apart.

Idealy the ultimate goal of detection is to align 3D models to the objects, as visualized in Fig 8. Once alignment is achieved, we can naturally get 3D bounding box, orientation and instance segmentation.

5. Conclusions

In this work, we demonstrated how to detect objects (finding an object heat map) in 3D scene by taking advantage of state-of-the-art object detectors in RGB images and aggregating signals in a common 3D space through back-projections. We have presented our modules for detection score back-projection into 3D voxel grids, a few methods for score aggregation and filtering, along with two possible applications built on top of our pipeline. While VR, AR and robotics are quickly being comercialized, 3D semantic understanding will play an important role and we believe the ideas and tools in our work will be very applicable.

Although steady steps have been made, we think our work is still preliminary and there are many interesting future works. For example, how to fully automatically accomplish instance segmentation and 3D model alignment will be a big topic to explore. At lower levels, currently we randomly pick up 1000 frames but there can be smart ways to pick such as uniformly pick frames on the camera trajectory or pick the most "discriminative" frames etc. Also it would be interesting to study how to propagate scores to voxels that fall below the confidence threshold. For the chair case, we see only fractions of the chair backs and seats are highlighed in Fig 6, if we can propaget them to the entire chair it will be helpful for both segmentation and alignment.

Acknowledgement

I'd like to express my gratitude to the wonderful teaching team of the course. Many thanks to Gordon Wetzstein for great lectures and valuable project discussion, to TA Liang and Orly for helpful Q&A and discussions. Special thanks to Matthias Niessner for providing the dataset and valuable feedbacks on the project.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[2] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[3] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013.

[4] S. Pillai and J. Leonard. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732*, 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[6] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[7] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[8] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015.

[9] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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.