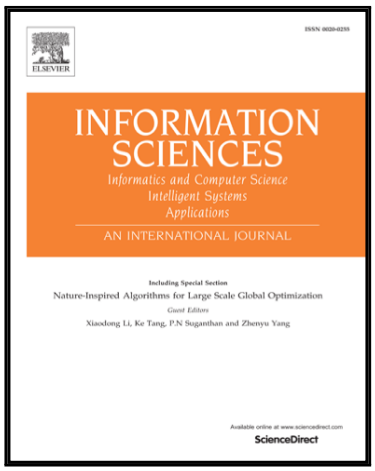

3D Object Detection: Learning 3D Bounding Boxes from Scaled Down 2D Bounding Boxes in RGB-D Images

Mohammad Muntasir Rahman, Yanhao Tan, Jian Xue, Ling Shao, Ke Lu

PII: S0020-0255(18)30754-0
DOI: https://doi.org/10.1016/j.ins.2018.09.040
Reference: INS 13951

To appear in: *Information Sciences*

Received date: 28 February 2018
Revised date: 18 September 2018
Accepted date: 19 September 2018

Please cite this article as: Mohammad Muntasir Rahman, Yanhao Tan, Jian Xue, Ling Shao, Ke Lu, 3D Object Detection: Learning 3D Bounding Boxes from Scaled Down 2D Bounding Boxes in RGB-D Images, *Information Sciences* (2018), doi: https://doi.org/10.1016/j.ins.2018.09.040

3D Object Detection: Learning 3D Bounding Boxes from Scaled Down 2D Bounding Boxes in RGB-D Images

Mohammad Muntasir Rahman[a,b], Yanhao Tan[a], Jian Xue[a], Ling Shao[c], Ke Lu[a,*]

[a]*School of Engineering Science, University of Chinese Academy of Sciences, No.19A Yuquan Road, Beijing 100049, China*
[b]*Department of Computer Science and Engineering, Islamic University, Kushtia 7003, Bangladesh*
[c]*Inception Institute of Artificial Intelligence, Abu Dhabi, UAE*

Abstract

3D object detection in RGB-D images is a vast growing research area in computer vision. In this paper, we study the problems of amodal 3D object detection in RGB-D images and present an efficient 3D object detection system that can predict object location, size, and orientation. Unlike existing methods that either uses multistage point cloud processing or pre-computed segmentation mask to generate the 3D bounding boxes, we only leverage 2D region proposals for this task. Given a pair of color and depth image as input, we first predict 2D region proposals from the designed multimodal fusion region proposal networks and then we propose an efficient method to generate 3D bounding boxes from those region proposals by scaling down the 2D bounding boxes with a scale factor and project it to 3D space. We evaluate our system on challenging NYUv2 and SUN RGB-D dataset and compare with the state-of-the-art detection methods. The experimental results show that our method outperforms the state-of-the-art by a remarkable margin with faster detection time. We achieve the best results on the NYUv2 dataset on a 19-class object detection task while performing comparably faster detection performances on the SUN RGB-D dataset on a 10-class object detection task.

Keywords: 3D Object Detection, RGB-D data, Deep Neural Networks, Multi-modal Region Proposal Networks, Deep Feature Learning.

1. Introduction

Object detection is one of the fundamental challenges in computer vision. In the past few years, 2D object detection has extensively investigated which currently shown enormous performance due to the availability of large-scale annotated datasets and impressive advances in deep convolutional neural networks (CNNs) [12, 11, 36]. 2D object detection aims to identify and localize all objects by drawing rectangles on the image planes. However, going beyond 2D plane information, accurate 3D object detection in indoor scenes in the 3D world remains an open problem compared to its counterpart. 3D understanding is very important in numerous real-world applications such as autonomous navigation, housekeeping robots, autonomous driving and augmented reality (AR). To facilitate this for the indoor scenario, the fundamental problems are to be able to reliably classify and localize amodal 3D objects in 3D space, where the amodal 3D object detection aims to draw a 3D bounding box of a complete object even if part of the object is occluded or truncated. 3D object detection enables machines (*i.e.,* robots) to interact with the indoor 3D environment. A variety of attempt has been made in the past decade to localize 3D objects from monocular images [29, 10] or inserting CAD models [13, 45, 47]; or extend 2D detections into the 3D [27, 18], but these techniques require prior knowledge of the environment.

*Corresponding author.
Email address: luk@ucas.ac.cn (Ke Lu)

Recently, with the rapid technological development and popularity of low-cost 3D sensing equipment (*i.e.,* Microsoft Kinect, Xtion Pro-live etc.), that provide depth along with color information, indoor scene understanding in RGB-D images becomes an active research area [6, 48, 46]. Depth images provide more geometrical information which is invariant to color, illumination, rotation, and scale compared to RGB images. Therefore, leveraging the power of RGB-D data with Deep Neural Networks can significantly improve the performances of image/video classification tasks [39, 16, 33]. However, due to the lack of large-scale RGB-D annotated dataset like ImageNet [7] and massive occlusions in indoor scenes, 3D object detection itself is a challenging task with great difficulty for indoor scenarios. Moreover, the main drawback of the 3D is the addition of an extra spatial dimension makes the search space for objects significantly large and computationally intensive even if operating on modern powerful hardware accelerator (GPU). As a result, the state-of-the-art 3D object detection methods still tend to be much slower than 2D object detection methods.

The dominant object detection frameworks in the literature contain two important components: proposal generation stage [36, 44] and region-wise object recognition stage [12, 20]. Unlike traditional sliding window based methods [9], recent effort in proposal generation methods aims to propose a moderate number of candidate regions which covers the most of the ground truth objects with the benefit from the deep learning techniques (*i.e.*, region proposal networks [36]. However, starting from 3D point cloud processing [44, 26, 37] or operating on the 3D convolutional neural networks [44, 23] is computationally intensive in 3D object detection. Most existing works in 3D object detection task started with 3D point clouds by converting it to images [45, 31] or to volumetric grids [47, 31, 28] or directly use them [32, 30] as input to convolutional neural networks to train. For instance, [44] converted point cloud of an entire scene into the 3D voxel grid and used 3D ConvNet for object proposals and classification. Despite those works achieved efficiency in several 3D understanding tasks, creating point clouds from the depth data obtained from indoor depth sensors are often noisy, sparse and incomplete in nature. Therefore, if an object is missing from the point clouds due to this noisy depth data or a majority of its area on the depth map is empty, the 3D anchor boxes cannot identify that object and resulting performance degrade. Moreover, the computational cost of those methods to convert a point cloud of an entire scene into another form and searching candidate box for objects on the 3D ConvNet is usually quite high. More recent approaches [21, 8] tried to solve the 3D object detection problem from the 2D point of view. However, [21] requires hand-crafted features, which are feed into a multilayer perceptron (MLP) network to regress 3D box location and pose. [8] takes 2D bounding boxes around superpixels together with RGB-Depth data as input. But, their method needs segmentation mask information for each 2D proposals generated from multiscale combinatorial grouping (MCG) algorithm [14], which requires extra and offline computation.

In this paper, inspired by [44, 8], we revisit the problem of amodal 3D object detection in the indoor scenes and propose a novel 3D object detection framework that takes only RGB-D imagery to detect multiple 3D objects. Without starting from point cloud processing or using segmentation mask, we propose a simple and efficient method of generating 3D object proposals that only requires 2D region proposals information for objects. To generate 2D region proposals, we design a multimodal fusion region proposal networks (MF-RPN) by incorporating both color and depth data as input. Finally, for each 2D region proposal, we first scale down the 2D bounding box by a scale factor towards its center and then compute median of the depth data inside the scale down region and project it to 3D space to initialize 3D bounding boxes which are then regressed to predict multiple 3D objects. When compared with previous state-of-the-art works, our method has achieved 6.8% and 2.2% better 3D mAP than [44] and [8] on the NYUv2 RGB-D dataset with 65× and 2.5× faster testing time, respectively. Our method also suits well with SUN RGB-D dataset, where we attain comparable detection result with more faster detection time compared to the state-of-the-art method.

The key contributions of the paper are as follows:

- Given color and depth image, we propose an efficient 3D object detection framework based on CNNs that does not require 3D volumetric input or segmentation proposals.
- We design a multimodal fusion region proposal networks (MF-RPN) which leverage both color and depth as input to predict 2D region proposals. No 3D ConvNet or extra offline computation is needed (*i.e.* MCG algorithm [14]) to generate object proposals.

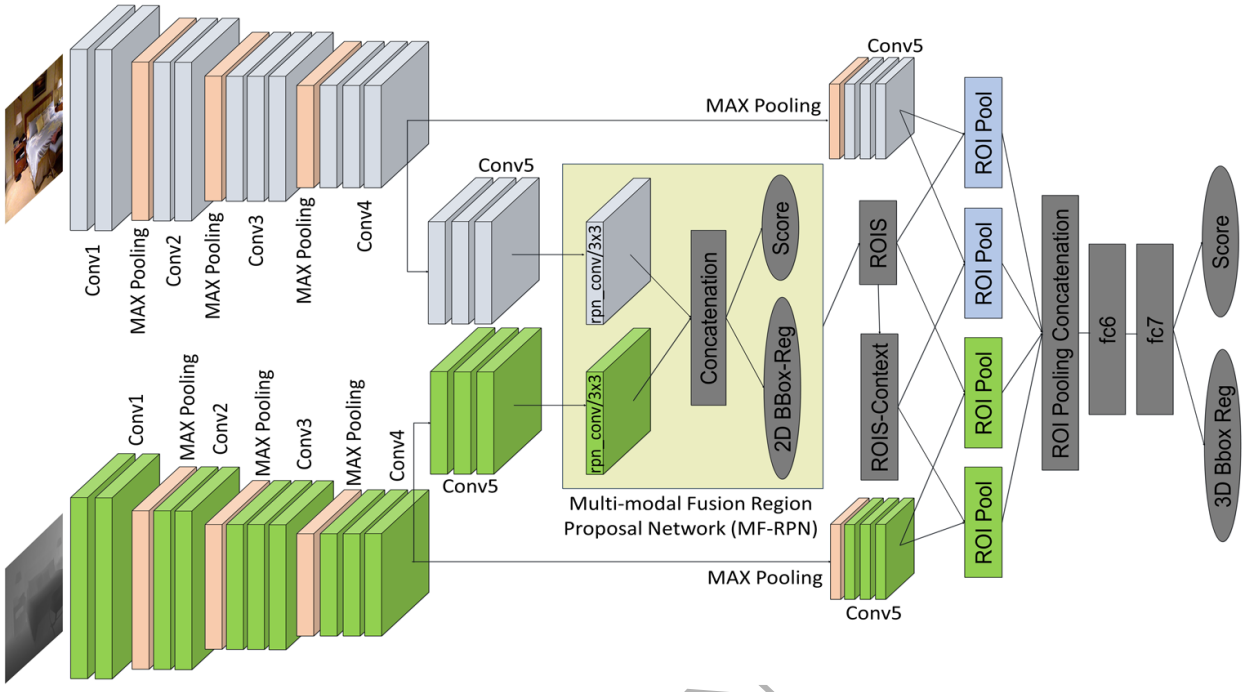

Figure 1: **Architectural diagram.** Given RGB-D image, we first generate 2D object proposals from the Multi-modal Fusion Region Proposal Networks (MF-RPN). The MF-RPN takes input from the unsampled branch of *conv5* from both color and depth stream, while the RoI pooling layers take input from the downsample branch of *conv5*. We also use context embedding technique which is nothing but 1.5 times bigger region than the corresponding ROIS. Each 2D bounding box is then scaled down and the 3D bounding boxes initialized from the depth data within the scaled downed region for 3D bounding box regression.

- As objects are in different size and shapes in 3D space, the centroid of 3D bounding box targets on xz plane is normalized by the diagonal of the objects length and width. We do not use Manhattan world assumption like some 3D detection system do [44, 43] and do not use 3D CAD models for orientation estimation.
- We conduct extensive quantitative experiments on the two challenging datasets: NYUv2 and SUN RGB-D dataset to understand the strengths and limitations of our design approach.

2. Related Works

Object detection has long been a challenging problem in computer vision. With the rapid success of deep convolutional neural networks, recent years have witnessed a revolutionize progress in object detection techniques. The state-of-the-art e.g., R-CNN [12], Fast R-CNN [11], Faster R-CNN [36], YOLO [34, 35], SSD [25], Mask-RCNN [17], and Focal Loss [24] are the most successful methods in terms of speed and accuracy of detecting objects from images. Nevertheless, those approaches were designed for predicting 2D rectangular bounding boxes around the visible part of objects in the given images. However, very few successful works have been conducted in 3D object detection in terms of speed and accuracy. Following, we briefly review the existing object detection algorithms in RGB-D images. Based on 2D and 3D, we can divide those into two sections:

2.1. 2D Object Detection in RGB-D Images

Earlier, RGB-D object recognition and detection work relied on channel specific hand engineered feature descriptors by treating depth as an extra channel. In [22], Lai *et al.* first introduced RGB-D object dataset

and proposed an object recognition and detection framework where object recognition was performed by using a combination of several hand-crafted features (*i.e.*, SIFT descriptors, texton histograms, and color histograms) together with spin images and shape features representation. And object detection was performed by computing a variant of a histogram of oriented gradients (HOG) over both RGB and depth images and normalized depth histograms. On the other hand, learning-based feature descriptors have opened a new perspective on feature extraction techniques. Blum *et al.* [2] explored a feature learning based approach from RGB-D data where a convolutional K-means descriptor can learn features automatically from the neighborhood of detected interest points with SURF features. Bo *et al.* [3] presented a hierarchical matching pursuit (HMP) method based on sparse coding to learn new feature representations in an unsupervised way from RGB-D images. The success of deep convolutional neural networks (CNNs) for 2D object detection on RGB images has more recently led the researchers to exploit the use of CNNs on RGB-D images. Gupta *et al.* [14] first proposed a depth R-CNN based object detector using deep CNNs on RGB-D images. Unlike R-CNN which used color images as input, they treated depth as additional streams of input by encoding it into three channels as a combination of height above ground, horizontal disparity and angle with gravity (HHA). Though object detection on RGB images has accelerated a great with the initialization of pre-trained model learned on large-scale image dataset (*i.e.*, ImageNet [7]), however, there is no such large-scale dataset for depth modality, which could be used as a pre-trained model. Gupta *et al.* employ the ImageNet pre-trained model to initialize both color and depth parameters in [14] and later they proposed a cross model supervision transfer technique in [15], where they exploit learned representations from a largely labeled modality as a supervisory signal for training representations for unlabeled paired depth modality.

2.2. 3D Object Detection in RGB-D Images

More recent approaches that have aimed to detect 3D objects in RGB-D images exploit 3D point cloud with voxel grid representation. For instance, [43] trained exemplar SVM classifier on hand-crafted features computed on 3D point cloud by rendering a collection of 3D CAD models into synthetic depth maps. During testing, they slide a 3D detection window in the 3D space to match the exemplar shape and each window. Furthermore, they also use depth segmentation to improve the performances. Gupta *et al.* extended depth R-CNN [14] in [13] for generating 3D bounding boxes by aligning 3D CAD models to 3D points projected back from 2D recognition results using segmentation mask with modified iterative closest point (ICP) algorithm. Inspired by [43], Liu *et al.* [26] also adopt 3D sliding window strategy in the 3D point cloud to get scores for all exemplar-SVMs. The 3D bounding boxes were projected into 2D bounding boxes on both channels and then the cross-modal features were extracted by feeding it into pre-trained R-CNNs and bimodal deep Boltzmann Machine, respectively. Finally, the exemplar-SVMs are used for detection. However, the uses of many exemplar classifier and hand-crafted features make the algorithm computationally intensive. Similar to [43], Ren *et al.* in [37] design a cloud of oriented gradients (COG) descriptor for 3D object detection in a point cloud and they further extended their work in [38] by using latent support surfaces. Although Ren *et al.* archived significant gain on mAP, their model takes 10-30min processing time for each image while testing. [43] extended in Deep Sliding Shapes [44], where Song and Xiao inspired by the faster R-CNN [36] presented a 3D ConvNet based approach that converts a point cloud of an entire scene into a 3D volumetric grid and outputs 3D bounding boxes. Their approach has two modules: a 3D Region Proposal Network (3D RPN) for generating 3D region proposals; and a joint Object Recognition Network (3D ORN) for extracting geometric features in 3D and color features in 2D. Nevertheless, the computations of the two modules have done separately and do not share convolutional layers. Moreover, operating 3D ConvNets on the 3D voxel grids in the large 3D search space are usually computationally expensive.

The recent methods that exploiting 3D object detection from the 2D point of view has gained attention to the researcher. [21] proposes a 2D driven 3D object detection method to reduce the search space for objects in 3D, which uses a multistage pipeline to perform 2D object detection, 3D object orientation regression and object refinement based on context information. However, they use hand-crafted features (*i.e.,* histograms for the coordinates of the 3D points) to train a multilayer perceptron (MLP) network to regress the 3D bounding boxes. [8] also propose a 3D object detection technique by inferring 3D bounding boxes from 2D. A limitation of their work is that it is not fully integrated architecture as there are two separate computations: first, they externally compute 2D bounding box proposals along with their segmentation

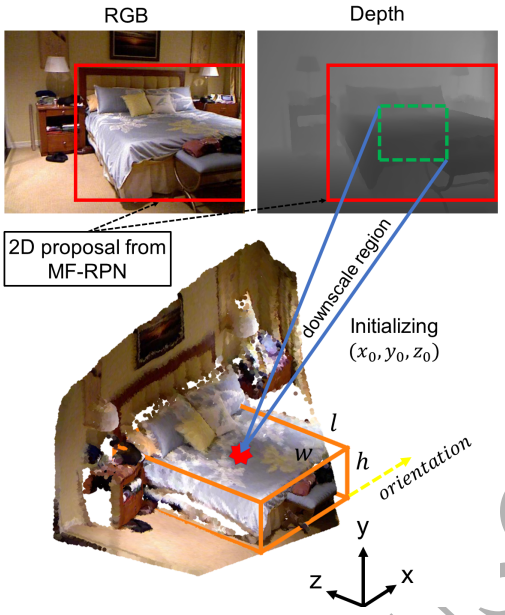

Figure 2: **3D bounding box generation.** Given a 2D bounding box from MF-RPN (shown in the red rectangular region), we first reduced it by a scale factor (shown in the green dotted the rectangular region on the depth image). Then the centroid (x_0, y_0, z_0) of the 3D bounding box is computed as the 3D point projected from the center point of the 2D box in conjunction with the median of the scale down region's depth crop. (l, w, h) represents length, width, and height that initialized from the averaged class-wise box dimensions calculated from the training set and the yellow dotted line determines the orientation angle of the 3D box.

mask by using extended multiscale combinatorial grouping (MCG) algorithm [14], then they use those 2D proposals and corresponding segmentation mask information to compute 3D boxes for classification and 3D bounding box regression. Nevertheless, computing 2D proposals and segmentation mask from the offline algorithm (*i.e.,* MCG [14]) requires quite high computation cost and are not feasible to the automated system. In addition, they require to provide object segmentation information in both training and testing phases to predict 3D bounding boxes.

In this paper, our goal is to design an efficient 3D object detection framework that can eliminate the use of high computation point cloud processing and the need of segmentation mask information to predict the 3D bounding boxes for objects in 3D space. We follow the 3D object detection from the 2D point of view concept and design an integrated end-to-end architecture instead of piecewise computation. In particular, our object detection network can be seen as a simplification of the recent state-of-the-art 3D object detection networks [44, 8] in RGB-D images.

3. Proposed 3D Object Detection Framework

Given an RGB-D data as input, our goal is to classify and localize 3D objects in its full extent in 3D space. Figure 1 present architectural diagram of our proposed 3D object detection framework consisting of three modules: multi-modal deep feature learning that extracts multi-modal features from RGB-D data, multi-modal fusion region proposal networks (MF-RPN) that predict 2D region proposals, and 3D object detector that outputs 3D bounding boxes for the objects. We will describe each module in the following subsections.

3.1. Multi-modal Deep Feature Learning

As shown in Figure 1, our network architecture has two streams of deep convolutional networks, that takes a pair of color and depth images to leverage shared feature maps. The color feature has been widely used for

many CNNs. Since depth provides supplementary information about the object's geometrical shape which is invariant to illuminations and color variation, combining color images to the depth data can significantly improve the performances of the object recognition and detection system [39, 14]. The multi-modal fashion makes the model robust and carries more discriminative feature than conventional unimodal structure.

In this paper, we use VGG16 [41] network in both streams as a backbone CNNs. Original VGG16 [41] network encompasses five hierarchies of convolutional blocks: *conv1*, *conv2*, *conv3*, *conv4*, and *conv5*. Note that each block followed by a *pooling* layer that downsampling the spatial dimensions of the feature maps by a factor of 2. Consequently, the 16× downsampling of the original VGG16 [41] network often miss the small object proposals due to the low-resolution feature maps on *conv5*. For that reason, we add another branch of *conv5* with 8× downsampling by removing the pooling layer after *conv4* of the VGG16 [41] network to provide high-resolution feature maps to the Multimodal Fusion Region Proposal Networks (MF-RPN) as shown in Figure 1. This higher resolution feature maps of *conv5* layer facilitate the MF-RPN aims to predict at relatively small objects as well as big objects too. The 16× downsample version of VGG16 [41] network directly goes to the 3D object detectors through the RoI pooling layers. Both MF-RPN and 3D detection networks share the weights from *conv1* to *conv4* that enable the shared weight policy.

3.2. Multimodal Fusion Region Proposal Network (MF-RPN)

Region proposal generation is a crucial part of any object detection pipeline [36, 44, 11]. Searching proposals in 3D space for 3D objects are computationally expensive due to the addition of one more dimension significantly enlarges the search space. Moreover, different object category has different object size in 3D space, which make the sliding anchor box strategy more complex. Hence, starting from 2D rather than from the 3D is computationally efficient and plausible [13]. Having observed this fact, instead of exhaustive search in the 3D space, we start finding 2D proposals first on RGB-Depth pair, then convert those to 3D proposals in the next phase. Similar to Faster-RCNN [36], we adopt the same sliding window class agnostic RPN with a little modification in order to learn from both color and depth features as a multimodal fashion. We named it Multi-modal Fusion Region Proposal Network (MF-RPN). In general, RPN is a small subnetwork, which slides a 3×3 window over the whole image to evaluate a region correspond to an object or not object. In the original RPN design, a 3×3 convolutional layer is fed into two siblings of 1×1 convolutions to perform an object/non-object binary classification and bounding box regression, respectively. In our RPN design, we integrated the depth information to the region proposal networks to learn multimodal features. Accordingly, we attach one 3×3 convolutional layer on the top of high-resolution branches of the *conv5* block in both RGB-Depth streams, which are then merged together to the next layer followed by two siblings of 1×1 convolutions for classification and bounding box regression. The binary classification and bounding box regression targets are carried out by a set of *anchor* boxes with pre-defined scales and aspect ratios in order to cover different shapes of the objects. We assign training labels to the anchors based on their Intersection-over-Union (IoU) with the ground truth boxes and do not add any extra rules in addition to those in [36]. An anchor box is assigned to a positive label if it has highest IoU for a given ground truth box or has IoU over 0.7 with any ground truth boxes, and an anchor box is assigned to a negative label if it has IoU lower than 0.3 for all ground truth boxes. For bounding box regression, the bounding box regression target is computed as $t^* = (t_x^*,\ t_y^*,\ t_w^*,\ t_h^*)$ by following [36]:

$$\begin{aligned} t_x^* &= (x^* - x_a)/w_a, \quad & t_y^* &= (y^* - y_a)/h_a, \\ t_w^* &= \log(\frac{w^*}{w_a}), & t_h^* &= \log(\frac{h^*}{h_a}) \end{aligned} \tag{1}$$

where, x, y, w, and h represents the 2D bounding box's center coordinate and its width and height. The variables (x^*, y^*, w^*, h^*) and (x_a, y_a, w_a, h_a) denotes the ground-truth box and anchor box, respectively. Following the multi-task loss in [36], the loss function is defined for each anchor as:

$$L\left(p, p^*, t, t^*\right) = L_{cls}\left(p, p^*\right) + \lambda p^* L_{reg}(t, t^*) \tag{2}$$

where, L_{cls} is for objectness score, p defines the predicted probability of an anchor being an object and p^* denotes the ground-truth (1 if an anchor is positive, and 0 otherwise). The second term L_{reg} is a smooth L_1 loss for 2D box regression.

3.3. 3D Object Detector

3.3.1. 3D Proposal Generation

Thus far, the 2D region proposals are determined by the MF-RPN that most likely contain an object. Our next step is to estimate the 3D bounding boxes from the predicted 2D proposals along with depth data. Inspired by the strategy of inferring 3D bounding boxes from 2D ones described in [8] where a 3D bounding box is initialized from the instance segmentation information, we simply use 2D bounding box proposals information to initialize the 3D boxes without the need of any pre-computed segmentation proposals which requires extra offline computation and not suitable for automated system. For every 2D proposal, we first scale down the 2D bounding box by a scale factor and then we crop the depth values within the scaled down region to compute the median. Finally, the 3D box centroid is calculated as the 3D point projected from the center point of the 2D box together with the median depth values. The scale down region of the 2D proposals mostly covers the center area of the object which can initialize the 3D box centroid correctly. The 3D box size is the averaged class-wise box dimensions calculated from the training set. Figure 2 shows an illustration of generating a 3D bounding box from a 2D object proposal.

A 3D bounding box is a seven-elements vector $(x_0, y_0, z_0, l, w, h, \theta)$ where (x_0, y_0, z_0) is the centroid under camera coordinate system, (l, w, h) are the three dimensions, and θ represents the orientation angle, an angle between principal axis and orientation vector of the 3D box under tilt coordinate system. We do not use any *Manhattan-world* assumption. The orientation vector is the vector perpendicular to the longest edge of the 3D box in the xz plane. The center of the 3D bounding box is estimated as 3D points projected from the scaled down region's depth crop. $[x_0, y_0]$ is computed as the 3D projection of the center point of 2D proposals and z_0 is simply the median depth value of the pixels inside the scale down region of the 2D proposal's depth crop described in Eq. 3:

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = R_{tilt} * \begin{pmatrix} z_{median} * (c_x - o_x)/f_x \\ z_{median} * (c_y - o_y)/f_y \\ z_{median} \end{pmatrix} \tag{3}$$

where, R_{tilt} is the transformation matrix between the tilt and camera coordinate system, (c_x, c_y) is the center of 2D bounding box proposals from MF-RPN, and the camera intrinsic parameters (o_x, o_y) and (f_x, f_y) is the principal point and focal length, respectively. The dimensions (l, w, h) are initialized from averaged class-wise box dimensions estimated from the training set as base 3D box size and the orientation angle θ is initialized by setting it to zero.

3.3.2. 3D Bounding Box Regression

In this step, we need to refine the 3D box proposals that best fit the object being detected. In particular, the refinement process is performed by regressing the 3D bounding boxes to a target vector $t \in \mathbb{R}^7$ [12, 36, 11, 5, 49], $t = (t_x, t_y, t_z, t_l, t_w, t_h, t_\theta)$ and computed as Eq. 4:

$$\begin{aligned} t_x &= \frac{x^{gt} - x}{\sqrt{l^2 + w^2}}, \quad t_y = \frac{y^{gt} - y}{h}, \quad t_z = \frac{z^{gt} - z}{\sqrt{l^2 + w^2}}, \\ t_l &= \log\left(\frac{l^{gt}}{l}\right), \quad t_w = \log\left(\frac{w^{gt}}{w}\right), \quad t_h = \log\left(\frac{h^{gt}}{h}\right), \\ t_\theta &= \theta^{gt} * \pi/180 \end{aligned} \tag{4}$$

where, x^{gt} and x represent ground-truth box and predicted box, respectively. In the same way, we calculate for y, z, l, w, h and θ. Note that here, t_x and t_z are homogeneously normalized by the diagonal length of the 3D proposal box estimated from length and width instead of normalized by box dimensions as in Eq. 1 due to the varying size of the object length and width, which differs from the [44, 8].

3.3.3. Multi-task Loss Function

The positive and negative RoIs are determined by the IoU overlap of 2D proposals with the ground truth boxes. A 3D proposal is assumed to be positive if its corresponding 2D proposal IoU overlap with a ground

truth box is at least 0.5, and negative otherwise. During training, each RoI is labeled with a ground-truth class p^* and corresponding ground-truth 3D bounding-box regression target t^*. We use a multi-task loss function L on each labeled RoI to jointly predict object classes and oriented 3D bounding boxes:

$$L\left(p, p^*, t, t^*\right) = L_{cls}\left(p, p^*\right) + \lambda\left[p^* > 0\right] L_{3d\_reg}(t, t^*) \tag{5}$$

where, p is the predicted probability of the object class and t is the predicted 3D bounding box regression offset with respect to ground-truth. Like MF-RPN, the classification loss L_{cls} uses cross-entropy and the 3D bounding box regression loss L_{3d_reg} uses the smooth l_1 loss as defined in [11]. λ is the balancing parameter and we use $\lambda = 1$ in our experiment. During inference, we apply Non-Maximum Suppression (NMS) on the 2D detected boxes only. We do not use 3D NMS because objects occupy different space on the 3D ground plane.

4. Experimental Validations

In this section, we perform extensive experiments to validate the performance of our proposed 3D object detection method.

4.1. Dataset

We evaluate our 3D object detection framework on the two standard datasets: first, NYUv2 dataset [40] with improved annotations provided by [8], and second, SUN RGB-D dataset [42]. The Standard NYUv2 dataset comprises in total 1449 densely labeled pairs of aligned RGB and depth images captured by Microsoft Kinect v1, where 795 images provided for training and 654 images to test. On the other hand, SUN RGB-D dataset contains 10335 RGB-D scene images, in which 5285 images for training and 5050 images for testing. We employ raw depth images to compute the 3D bounding boxes center as described in Figure 2, and zero filled depth images as input to the CNNs to train the networks.

4.2. Evaluation Metric

In order to evaluate the detection performance, we follow the conventional 3D volume Intersection over Union (IoU) metric defined in [43]. A detected bounding box is considered to be true positive if the IoU with the ground truth is greater than 0.25. We train our model for 19-class object detection task on NYUv2 dataset namely *bathtub, bed, bookshelf, box, chair, counter, desk, door, dresser, garbage bin, lamp, monitor, night stand, pillow, sink, sofa, table, television, and toilet* and 10-class object detection task on SUN RGB-D dataset namely *bathtub, bed, bookshelf, chair, desk, dresser, night stand, sofa, table, and toilet* by following [44] and [37], respectively. Finally, after calculating Average Precision (AP) for each class, we calculate mean Average Precision (mAP) for performance evaluation.

4.3. Experimental Setup

For all training, we initialize convolutional layers of the both RGB and depth streams with a pre-trained VGG16 network [41] trained on ImageNet [7]. We also carry out data augmentation technique by adding flipped RGB and depth images to the training set. We train the MF-RPN to produce 2D boxes that most likely contain an object and then estimates 3D bounding boxes from the scale down region of the 2D detected boxes. As context has shown performance gain in many object detection tasks [1, 4], we use context embedding together with the detected 2D regions. The context region is 1.5 times larger than the 2D region proposals. The RoI pooling layer extract features from the 2D region proposals and their context regions, which are then stacked together to the next layer for multi-task learning. We adopt the pragmatic 4-step alternating training described in [36] to learn the shared features, where we train the network on stage 1 and 3 for 80,000 iterations and on stage 2 and 4 for 40,000 iterations with a base learning rate 0.001 and reduce it by a factor 10 after 60,000 iterations and 30,000 iterations, respectively. A momentum 0.9, weight decay 0.0005 are used in all experiments. For simplicity, we choose the balancing parameter λ to 1 in the loss function. We also use batch normalization layers after each convolutional layer. During training, we fixed the input images shorter side to 427 pixels which are passed through both region proposal networks and object recognition networks. Our 3D detection system is implemented on the open source Caffe [19] library with an NVIDIA TITAN X GPU.

Table 1: Performance Comparisons for 19-Classes 3D Object Detection on NYUv2 dataset.

Method	mAP	Time
DSS[44]	36.3	19.55s
Deng[8]	40.9	0.74s
Ours	**43.1**	**0.30s**

Table 2: Performance Comparisons for 10-Classes 3D Object Detection on SUNRGBD dataset.

Method	mAP	Time
COG [37]	47.63	10-30min
LSS [38]	51.0	10-30min
DSS [44]	42.1	19.55s
2D-driven [21]	45.12	4.15s
Ours	45.7	**0.30s**

4.4. Comparing with State-of-the-art Methods

In this subsection, we conducted several experiments to evaluate the performance of the proposed 3D object detection framework on NYUv2 [40] and SUN RGB-D [42] dataset and compared the results with the state-of-the-art methods.

NYUv2. Table 1 shows the performance of our 3D object detector on the NYUv2 test set. Here, we compare with the previous state-of-the-art methods DSS [44] and Deng [8] on a 19-class object detection task. The results clearly show that we outperform the state-of-the-art methods by a large margin in terms of speed and accuracy. In particular, the detection results boost to mAP 43.1, which is 6.8% improvement over DSS [44] and 2.2% improvement over [8]. Moreover, our method significantly enhances the speed of the testing time for each image, which is 65× and 2.5× faster than DSS [44] and [8], respectively. Our method achieves best results on 19-class objects on the NYUv2 dataset. We found that the experimental results significantly outperform the category: box, counter, desk, lamp, monitor, night stand, table and television over both DSS [44] and [8] especially in small objects. In DSS [44], 3D anchor boxes slide over the points clouds that could be sparse or empty due to the noisy and incomplete nature of the depth data from the Kinect, which led the 3D anchor boxes missed the sparse object in the point cloud. [8] uses segmentation mask information of an object instance to initialize the 3D box, which results in providing segment pixel information for each object instances in training/testing time. In contrast, we simply use 2D proposals generated from the MF-RPN and initialize 3D bounding boxes form the reduces region's depth values of the 2D bounding boxes. Both MF-RPN and 3D object detector learned from color and depth features that neither depends on points cloud or object segment information. From the experiment, we found the scale factor of 0.3 performs (described in Sec 4.5) better since the reduced region mostly covers the center area of the object which can initialize the 3D box centroid correctly.

SUN RGB-D. Table 2 shows the quantitative results of our proposed method on the SUN RGB-D dataset. Following COG [37], we compare a 10-class object detection task against four state-of-the-art methods: COG [37], LSS [38], DSS [44], and 2D-driven [21]. We report the results and runtime directly from [37], [38], [44], and [21]. Like NYUv2 dataset, here we present the result of our technique using scale factor 0.3. As seen in the table, our method is 13.8× faster than 2D-driven [21], 65× faster than DSS [44], and three orders of magnitude faster than COG [37] and LSS [38], while attaining a comparable detection performance which is 0.58% and 3.6% improvement over [21] and [44], respectively. When compared to COG [37] and LSS [38], our method does not search 3D bounding boxes exhaustively in point clouds which requires extra-large computation time. We computed 3D bounding boxes directly from the detected 2D boxes. The performance of our 3D detector is dependent on the 2D detection. Sometimes, if the 2D detector misses objects due to dark lighting or occlusion, the corresponding 3D object will also not detect. Moreover, as reported in [8], directly applying the 2D rule on SUN RGB-D with 2D annotation provided by [42], causes a serious problem due to the 2D annotation problems in that dataset (more described in [8]). However, we still manage to achieve comparable performance in terms of speed and accuracy.

It is worth noting that our method achieves that result without starting from point cloud processing or using any segmentation proposals that require extra high computation. In addition, our approach mitigates

Table 3: Ablation study of the Scale factor effect on NYUv2 dataset.

Scale factor																				mAP
congruent	46.8	81.6	41.7	10.8	48.2	52.0	35.5	10.4	43.7	34.5	36.7	4.6	60.2	43.7	51.8	60.2	43.2	4.2	76.5	41.4
0.6	47.4	81.0	40.6	11.3	50.6	49.7	36.0	9.8	39.2	36.0	36.7	4.9	60.7	45.3	53.5	61.9	46.3	7.8	77.9	41.9
0.5	47.4	83.0	41.8	11.2	51.5	49.9	37.0	3.9	41.5	37.4	38.8	4.9	63.9	43.5	56.2	59.2	44.7	11.3	75.8	42.3
0.4	42.3	80.4	44.5	11.0	48.8	52.6	35.0	9.7	42.8	35.9	38.1	8.4	63.2	44.7	56.5	58.1	42.7	14.8	78.5	42.5
0.3	44.6	82.2	40.0	11.3	49.0	53.3	37.0	10.3	42.2	33.4	35.8	14.0	64.7	46.0	56.4	57.9	45.3	19.2	75.5	**43.1**
0.2	41.5	80.1	36.6	11.1	47.3	52.3	36.3	6.8	41.1	38.6	36.5	6.9	65.2	44.5	56.1	58.9	45.9	14.9	75.2	41.9

Table 4: Ablation study of the different convolutional features on NYUv2 dataset. "RGB": use color image only as input. "D": normalized depth image. "Ct": context information.

Data																				mAP
RGB	20.1	59.6	20.7	9.6	28.9	41.5	21.6	1.4	31.0	27.6	22.0	1.1	34.3	29.2	48.3	40.0	27.5	5.2	62.8	28.0
RGB+D	44.2	77.2	32.8	5.9	44.1	48.9	31.7	5.1	32.6	31.7	35.5	1.7	53.9	38.0	49.8	52.8	41.2	17.7	77.2	38.0
RGB+D+Ct	44.6	82.2	40.0	11.3	49.0	53.3	37.0	10.3	42.2	33.4	35.8	14.0	64.7	46.0	56.4	57.9	45.3	19.2	75.5	**43.1**

Table 5: Ablation study of the effect of RoI Pooling layer connected to *conv5* with/without max-pooling layer on NYUv2 dataset.

conv5 layer																				mAP
without max-pool	52.7	81.7	43.0	11.2	46.9	49.4	34.3	2.4	39.9	35.2	33.9	6.1	58.8	45.5	52.2	59.2	43.6	14.1	79.1	41.5
with max-pool	44.6	82.2	40.0	11.3	49.0	53.3	37.0	10.3	42.2	33.4	35.8	14.0	64.7	46.0	56.4	57.9	45.3	19.2	75.5	**43.1**

the problem of extensive computation on 3D search space on point cloud and does not require any extra offline processing of 2D segment proposal generation. In Figure 3, we visualize some representative true positive detection results on NYUv2 test data with a class score greater than a threshold (0.7). We also show some error detection results on NYUv2 data in Figure 4 where errors are divided into four types: *wrong categories*, *orientation error*, *inaccurate locations* and *wrong box size*.

4.5. Ablation Study

To understand the importance of scale down of 2D boxes, we conduct several experiments on the NYUv2 dataset with the different scale factors. Table 3 shows the experimental results, which proved that the 3D detection accuracy is gradually increasing as we reduce the 2D bounding box size towards its center. In particular, the mAP is 41.4% with no reduction (congruent) and it dramatically increases to 41.9% when we set the scale factor to 0.6. The accuracy is significantly improved to mAP 43.1% when we use scale factor 0.3. However, the improvement suddenly starts dropping at 0.2 scale factor, since at this point the scaled down box area becomes too small to preserve the box centers depth information. This result demonstrates that scaled down region's depth information is a crucial point of 3D bounding box initialization. In addition, the experiment suggests a scale factor of 0.4 to 0.3 performs better detection result.

We also study the importance of the different features combination in the input on the NYUv2 dataset using scale factor 0.3 in Table 4. When only RGB images used as convolutional features, we achieve only 28% of mAP. Adding depth images to the RGB improve the result to 38% mAP. Further adding the Contextual information to the RoI pooling layer boosted the result by 5.1%. In our experiment, we used 1.5 times larger 2D region proposals as context region.

To understand the effect of the different feature map size of *conv5* layers, we experiment with RoI pooling layer connected to *conv5* with/without max-pooling layer on NYUv2 dataset with a scale factor 0.3 in Table 5. When RoI pooling layer connects to *conv5* with a max-pooling layer, we achieve 43.1% mAP. On the other hand, if we attach the RoI pooling layer to higher resolution version of *conv5* without a max-pooling layer, the detection performance dramatically degrades by 1.6%. As the RoI pooling layer is resolution sensitive, we found RoI pooling layer connected to lower feature map size of *conv5* perform slightly better than RoI pooling layer connected to higher feature map size of *conv5* layers.

5. Conclusion

3D object detection in RGB-D images is a very challenging task due to the scarcity of large-scale annotated training data. To confront that challenge, in this work, we present a fast and efficient algorithm for 3D object detection in indoor scenarios. We make mainly two contributions. First, we design a multimodal fusion region proposal networks (MF-RPN) to predict 2D object proposals and second, we propose an efficient technique to generate 3D proposals from the 2D object proposals by scaling down the 2D bounding boxes. Unlike many state-of-the-art works, our method does not use multiple stage point cloud processing or use any pre-computed segmentation information. Given a pair of color and depth, our system can predict multiple 3D objects in its full extent. When compared with the state-of-the-art 3D object detectors, our method achieves faster and better detection performance by a significant margin on the challenging NYUv2 dataset while still performing comparably on the large-scale SUN RGB-D dataset. In future, we will extend and verify the effectiveness of the proposed method on tasks, such as real-time 3D object detection and tracking.

Acknowledgements

This work was supported by National Key R&D Program of China [grant number 2017YFB1002203]; the National Natural Science Foundation of China [grant number 61671426, 61731022, 61471150, 61572077]; the Instrument Developing Project of the Chinese Academy of Sciences [grant number YZ201670]; and the Beijing Natural Science Foundation [grant number 4182071]. Thanks to CAS-TWAS Presidents Fellowship for supporting MMR as a doctoral student [fellowship number 2015CTF075].

References

References

[1] S. Bell, C. L. Zitnick, K. Bala, R. B. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2874–2883. doi:10.1109/CVPR.2016.314.

[2] M. Blum, J. T. Springenberg, J. Wülfing, M. A. Riedmiller, A learned feature descriptor for object recognition in RGB-D data, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 1298–1303. doi:10.1109/ICRA.2012.6225188.

[3] L. Bo, X. Ren, D. Fox, Unsupervised feature learning for RGB-D based object recognition, in: The 13th International Symposium on Experimental Robotics (ISER), 2012, pp. 387–402. doi:10.1007/978-3-319-00065-7_27.

[4] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, R. Urtasun, 3d object proposals for accurate object class detection, in: Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), 2015, pp. 424–432.

[5] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3d object detection network for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6526–6534. doi:10.1109/CVPR.2017.691.

[6] Y. Chen, D. Pan, Y. Pan, S. Liu, A. Gu, M. Wang, Indoor scene understanding via monocular RGB-D images, Information Sciences 320 (2015) 361–371. doi:10.1016/j.ins.2015.03.023.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, F. Li, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255. doi:10.1109/CVPRW.2009.5206848.

[8] Z. Deng, L. J. Latecki, Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 398–406. doi:10.1109/CVPR.2017.50.

[9] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645. doi:10.1109/TPAMI.2009.167.

[10] S. Fidler, S. J. Dickinson, R. Urtasun, 3d object detection and viewpoint estimation with a deformable 3d cuboid model, in: Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), 2012, pp. 611–619.

[11] R. B. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448. doi:10.1109/ICCV.2015.169.

[12] R. B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587. doi:10.1109/CVPR.2014.81.

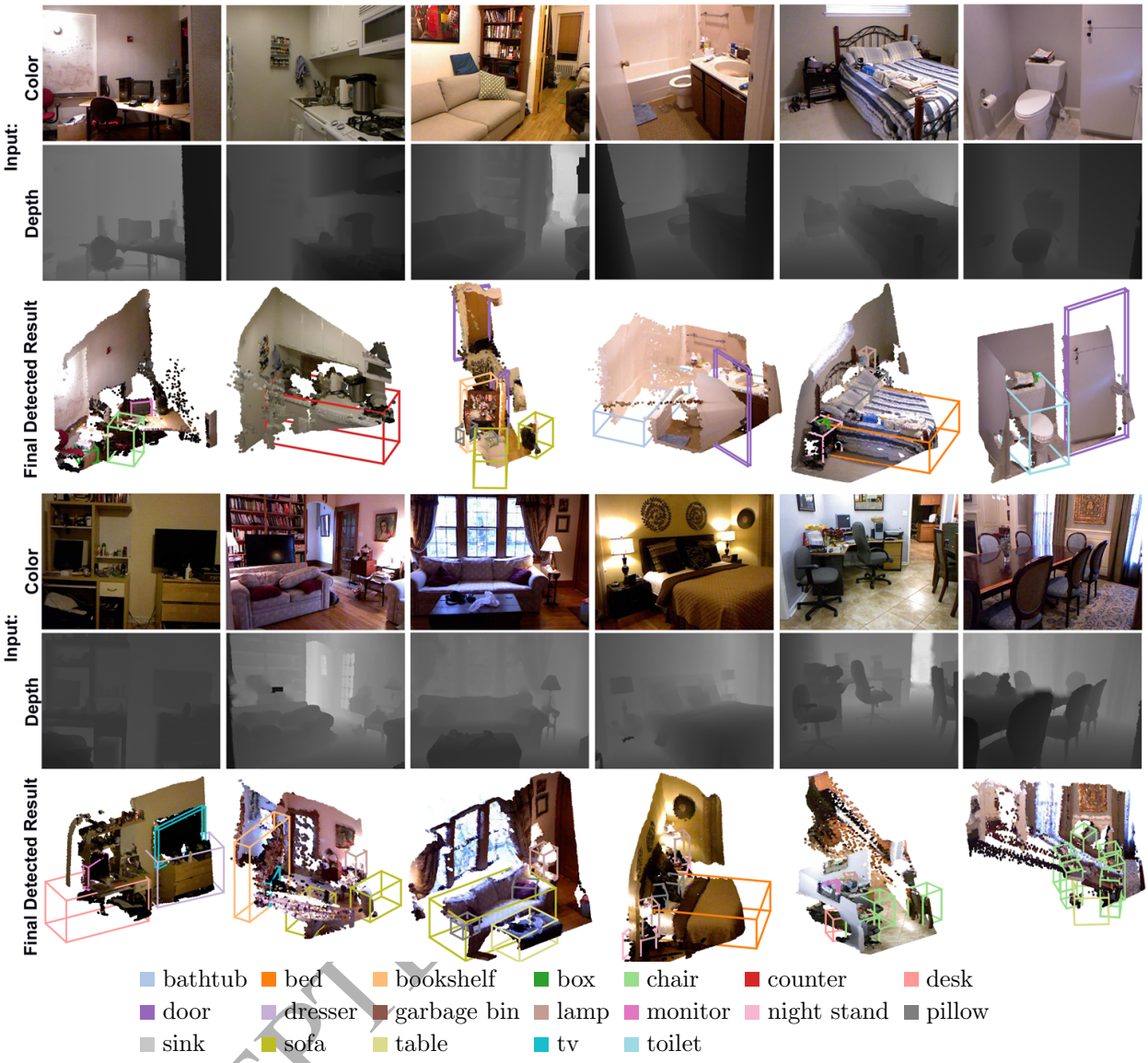

Figure 3: Sample examples of 3D object detection results from our final model on NYUv2 test set. Our model is able to detect multiple sizes, scales and orientations objects. Detections with confidence score > 0.7 are visualized.

[13] S. Gupta, P. A. Arbeláez, R. B. Girshick, J. Malik, Aligning 3d models to RGB-D images of cluttered scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4731–4740. doi:10.1109/CVPR.2015.7299105.

[14] S. Gupta, R. B. Girshick, P. A. Arbeláez, J. Malik, Learning rich features from RGB-D images for object detection and segmentation, in: Proceedings of the 13th European Conference on Computer Vision (ECCV), 2014, pp. 345–360. doi:10.1007/978-3-319-10584-0_23.

[15] S. Gupta, J. Hoffman, J. Malik, Cross modal distillation for supervision transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2827–2836. doi:10.1109/CVPR.2016.309.

[16] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with microsoft kinect sensor: A review, IEEE Transactions on Cybernetics 43 (5) (2013) 1318–1334. doi:10.1109/TCYB.2013.2265378.

[17] K. He, G. Gkioxari, P. Dollár, R. B. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322.

[18] D. Hoiem, A. A. Efros, M. Hebert, Putting objects in perspective, International Journal of Computer Vision 80 (1) (2008) 3–15. doi:10.1007/s11263-008-0137-5.

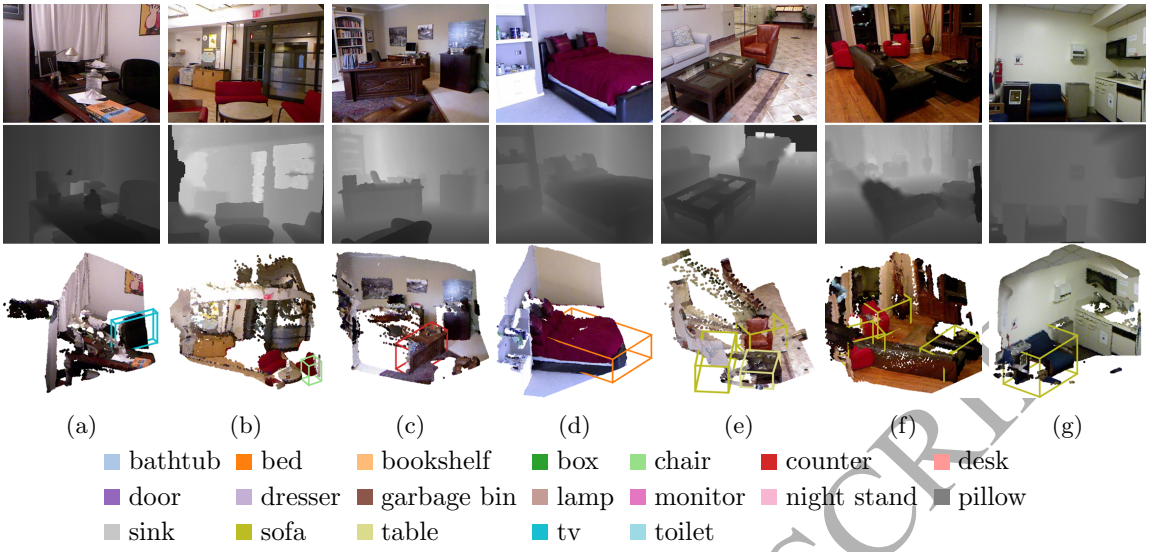

Figure 4: Examples of Error Detection on NYUv2 test set. We show four types of error detection. Wrong categories: (a) chair detected as a television, (b) sofa detected as a chair, (c) desk detected as a counter. Orientation error: (d) wrong orientation of the bed, (e) wrong orientation of the sofa. Inaccurate locations: (f) inaccurate location of the sofa. Wrong box size: (f) and (g) wrong box size is placed on the sofa.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia (MM), 2014, pp. 675–678. doi:10.1145/2647868.2654889.

[20] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90. doi:10.1145/3065386.

[21] J. Lahoud, B. Ghanem, 2d-driven 3d object detection in rgb-d images, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4632–4640. doi:10.1109/ICCV.2017.495.

[22] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view RGB-D object dataset, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 1817–1824. doi:10.1109/ICRA.2011.5980382.

[23] B. Leng, Y. Liu, K. Yu, X. Zhang, Z. Xiong, 3d object understanding with 3d convolutional neural networks, Information Sciences 366 (2016) 188–201. doi:10.1016/j.ins.2015.08.007.

[24] T. Lin, P. Goyal, R. B. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007. doi:10.1109/ICCV.2017.324.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, A. C. Berg, SSD: single shot multibox detector, in: Proceedings of the 14th European Conference on Computer Vision ECCV, 2016, pp. 21–37. doi:10.1007/978-3-319-46448-0_2.

[26] W. Liu, R. Ji, S. Li, Towards 3d object detection with bimodal deep boltzmann machines over RGBD imagery, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3013–3021. doi:10.1109/CVPR.2015.7298920.

[27] T. Malisiewicz, A. Gupta, A. A. Efros, Ensemble of exemplar-svms for object detection and beyond, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 89–96. doi:10.1109/ICCV.2011.6126229.

[28] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 922–928. doi:10.1109/IROS.2015.7353481.

[29] N. Payet, S. Todorovic, From contours to 3d object detection and pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 983–990. doi:10.1109/ICCV.2011.6126342.

[30] C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85. doi:10.1109/CVPR.2017.16.

[31] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L. J. Guibas, Volumetric and multi-view cnns for object classification on 3d data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5648–5656. doi:10.1109/CVPR.2016.609.

[32] C. R. Qi, L. Yi, H. Su, L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS), 2017, pp. 5105–5114.

[33] M. M. Rahman, Y. Tan, J. Xue, K. Lu, RGB-D object recognition with multimodal deep convolutional neural networks, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 991–996. doi:10.1109/ICME.2017.8019538.

[34] J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.

[35] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525. doi:10.1109/CVPR.2017.690.

[36] S. Ren, K. He, R. B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 39 (6) (2017) 1137–1149. doi:10.1109/TPAMI.2016.2577031.

[37] Z. Ren, E. B. Sudderth, Three-dimensional object detection and layout prediction using clouds of oriented gradients, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1525–1533. doi:10.1109/CVPR.2016.169.

[38] Z. Ren, E. B. Sudderth, 3d object detection with latent support surfaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[39] L. Shao, Z. Cai, L. Liu, K. Lu, Performance evaluation of deep feature learning for RGB-D image/video classification, Information Sciences 385 (2017) 266–283. doi:10.1016/j.ins.2017.01.013.

[40] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, in: Proceedings of the 12th European Conference on Computer Vision (ECCV), 2012, pp. 746–760. doi:10.1007/978-3-642-33715-4_54.

[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556.

[42] S. Song, S. P. Lichtenberg, J. Xiao, SUN RGB-D: A RGB-D scene understanding benchmark suite, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 567–576. doi:10.1109/CVPR.2015.7298655.

[43] S. Song, J. Xiao, Sliding shapes for 3d object detection in depth images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 634–651. doi:10.1007/978-3-319-10599-4_41.

[44] S. Song, J. Xiao, Deep sliding shapes for amodal 3d object detection in RGB-D images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 808–816. doi:10.1109/CVPR.2016.94.

[45] H. Su, S. Maji, E. Kalogerakis, E. G. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 945–953. doi:10.1109/ICCV.2015.114.

[46] D. Tao, J. Cheng, X. Lin, J. Yu, Local structure preserving discriminative projections for RGB-D sensor-based scene classification, Information Sciences 320 (2015) 383–394. doi:10.1016/j.ins.2015.03.031.

[47] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920. doi:10.1109/CVPR.2015.7298801.

[48] Y. Xia, L. Zhang, W. Xu, Z. Shan, Y. Liu, Recognizing multi-view objects with occlusions using a deep architecture, Information Sciences 320 (2015) 333–345. doi:10.1016/j.ins.2015.01.038.

[49] Y. Zhou, O. Tuzel, Voxelnet: End-to-end learning for point cloud based 3d object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.