The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

3D Box Proposals from a Single Monocular Image of an Indoor Scene

Wei Zhuo,[1,4] **Mathieu Salzmann,**[2] **Xuming He,**[3] **Miaomiao Liu**[1,4]
[1]Australian National University, [2]EPFL, [3]ShanghaiTech University, [4]Data61,CSIRO, Canberra, Australia
wei.zhuo@anu.edu.au, mathieu.salzmann@epfl.ch, hexm@shanghaitech.edu.cn, miaomiao.liu@data61.csiro.au

Abstract

Modern object detection methods typically rely on bounding box proposals as input. While initially popularized in the 2D case, this idea has received increasing attention for 3D bounding boxes. Nevertheless, existing 3D box proposal techniques all assume having access to depth as input, which is unfortunately not always available in practice. In this paper, we therefore introduce an approach to generating 3D box proposals from a single monocular RGB image. To this end, we develop an integrated, fully differentiable framework that inherently predicts a depth map, extracts a 3D volumetric scene representation and generates 3D object proposals. At the core of our approach lies a novel residual, differentiable truncated signed distance function module, which, accounting for the relatively low accuracy of the predicted depth map, extracts a 3D volumetric representation of the scene. Our experiments on the standard NYUv2 dataset demonstrate that our framework lets us generate high-quality 3D box proposals and that it outperforms the two-stage technique consisting of successively performing state-of-the-art depth prediction and depth-based 3D proposal generation.

Introduction

In the context of 2D scene understanding, generating class-independent object proposals, such as bounding boxes, has proven key to the success of modern object detectors; it has led not only to faster runtimes but also to more accurate detections (Ren et al. 2015). Reasoning in 2D, however, only provides a limited description of the scene. A 3D interpretation would be highly beneficial for many tasks, such as autonomous navigation, robotics manipulation and Augmented Reality.

In recent years, several works have attempted to provide such a 3D interpretation by going beyond 2D bounding boxes. In particular, several methods have been proposed to model 3D objects with the 2D coordinates of the eight vertices of their 3D bounding box (Dwibedi et al. 2016; Hedau, Hoiem, and Forsyth 2010; Payet and Todorovic 2011). While this indeed better captures the shape of the object, e.g., by better adapting to orientations not parallel the image axes, it still does not provide a 3D interpretation; each 3D bounding box can only be recovered up to scale. By contrast, (Fidler, Dickinson, and Urtasun 2012; Chen et al. 2016a) truly reason in 3D. These works, however, have tackled the class-specific scenario in outdoor scenes, and would thus not generalize well to more cluttered environments, such as indoor scenes, and to arbitrary objects.

To the best of our knowledge, all the methods that consider the problem of generating class-independent object proposals (Chen et al. 2016b; Song and Xiao 2016) assume the availability of depth information. In particular, (Song and Xiao 2016) achieved state-of-the-art results in indoor scenes by encoding a 3D scene with a truncated signed distance function (TSDF) and developing a region proposal network (RPN) based on 3D convolutions to generate proposals. In practice, however, depth is not always available, in which case these methods are inapplicable.

In this paper, we therefore introduce an approach to generating class-independent 3D box proposals from a single monocular RGB image. Based on the recent progress in monocular depth estimation (Eigen, Puhrsch, and Fergus 2014; Eigen and Fergus 2015), the most straightforward way to doing so would be to rely on a state-of-the-art method to predict depth, followed by the state-of-the-art depth-based proposal generation technique of (Song and Xiao 2016). Here, however, we show that we can significantly outperform this two-stage approach by developing an integrated, fully-differentiable framework that can be trained in an end-to-end manner.

More specifically, we first propose a differentiable TSDF (DTSDF) module that can be appended to a depth-prediction network and produces an approximate TSDF-based representation. The quality of the resulting 3D representation, however, is limited by the accuracy of the predicted depth map and by our approximation of the TSDF, even when training the network end-to-end. To overcome this, we therefore introduce a residual version of our DTSDF module, which allows us to compensate for the depth inaccuracies and thus generate high-quality 3D box proposals.

We demonstrate the effectiveness of our method on the standard NYUv2 dataset. Our experiments evidence the benefits of the different components of our integrated framework. Furthermore, they show that our approach significantly outperforms the two-stage approach consisting of successive depth prediction and box proposal generation.

Related Work

Nowadays, the majority of object detection methods rely on generating class-independent object proposals. This trend was popularized in the 2D scenario, where diverse proposal mechanisms have been developed (Uijlings et al. 2013; Cheng et al. 2014; Pinheiro, Collobert, and Dollár 2015; Hayder, He, and Salzmann 2016; Pont-Tuset et al. 2017), and has proven highly beneficial for both accuracy and runtime (Ren et al. 2015; Girshick 2015). In particular, in the current Deep Learning era, the region proposal network of (Ren et al. 2015), which shares feature computation across the different regions, has been adopted by several approaches (Song and Xiao 2016; Dai, He, and Sun 2016; Hayder, He, and Salzmann 2017; He et al. 2017).

With the growing popularity of 3D scene understanding, it therefore seems natural that several works have turned to the problem of generating 3D box proposals. To this end, most approaches exploit the availability of depth sensors. In this setting, the main trend consists of fitting a 3D model to the given point cloud inside a 2D bounding box (Chen et al. 2016b; Lin, Fidler, and Urtasun 2013; Gupta et al. 2015; Deng and Latecki 2017), which typically leads to class-dependent methods. By contrast, the work of (Song and Xiao 2016) directly infers 3D boxes by learning shape features in a volumetric scene representation. While inspired by (Song and Xiao 2016), here, we introduce an approach that generates class-independent 3D box proposals directly from a single monocular image.

A few methods have nonetheless also attempted to reason in 3D from monocular input (Fidler, Dickinson, and Urtasun 2012; Chen et al. 2016a). These works, however, focus on the class-specific scenario, and have only tackled cases where a small number of classes are present in the scene. As such, they would not generalize to cluttered indoor scenes, and, more importantly, to the problem of class-independent object proposal generation that we tackle here.

In short, to the best of our knowledge, our approach constitutes the first attempt at generating class-independent 3D box proposals from a single monocular image. To this end, we leverage the recent progress on monocular depth estimation (Karsch, Liu, and Kang 2012; Liu, Salzmann, and He 2014; Ladicky, Shi, and Pollefeys 2014; Eigen, Puhrsch, and Fergus 2014; Zhuo et al. 2015; Eigen and Fergus 2015; Liu et al. 2016; Laina et al. 2016) and develop a fully-differentiable residual module able to generate a volumetric representation of a cluttered indoor scene from an input image.

While, in a different line of research, a few attempts have been made to learn a mapping from 2D images and 3D object representations (Wu et al. 2016; Girdhar et al. 2016; Zhou et al. 2017), these methods were developed for object-centric images, and are therefore not applicable to the complex indoor scene images that we deal with in this paper.

Methodology

We aim to generate 3D object proposals from a single monocular image. To this end, we design a multi-task deep network that predicts a depth map, extracts a volumetric representation of the scene and generates 3D object proposals in the form of cuboids. The corresponding three subnetworks are shown in Figure 1. All three of them are differentiable, and the entire network can thus be trained in an end-to-end fashion. In the remainder of this section, we introduce our volumetric representation prediction network and the object proposal subnetwork, and then discuss our overall training strategy.

Volumetric Representation Prediction Network

To build a 3D volumetric scene representation, we first estimate a pixel-wise depth map from the input image and then compute an approximate TSDF from the corresponding point cloud. Below, we introduce our approach to addressing these two steps. Importantly, to be able to integrate the resulting modules in a complete multi-task network, we design them so that they are fully differentiable.

Depth Estimation In this work, we adopt the VGG-based depth estimation network of (Eigen and Fergus 2015) as our depth prediction network. In particular, we utilize the first two scales of the network of (Eigen and Fergus 2015), which yields an output of size 55×74. We then upsample the resulting depth map to the full image size using bilinear interpolation, which can be cast as a convolution.

Differentiable TSDF Given the depth map predicted by the depth network discussed above, we rely on a TSDF, introduced by (Newcombe et al. 2011), as our volumetric scene representation. In the accurate TSDF representation, the 3D space is divided into equally-spaced voxels. Each voxel is assigned a value encoding the distance of the voxel center to the closest surface point, derived from the depth map. Unfortunately, computing such an accurate TSDF is not differentiable with respect to the input due to the use of a nearest neighbor search procedure. To address this, and thus be able to exploit this for end-to-end training, we propose to make use of a differentiable, projective TSDF approximation. In particular, instead of looking for the nearest surface point in the entire 3D space, we only perform the search along the visual ray of each voxel; we further introduce a soft truncation function to compute the final representation, which makes the entire process differentiable.

More formally, we divide the 3D space into $L \times H \times W$ equally-spaced cells of equal volume. Let us then denote by $G \in \mathbb{R}^{L \times H \times W}$ the 3D grid whose nodes encode the x, y, and z coordinates of the corresponding cell centers in the 3D world referential. Our goal now is to compute a TSDF value for each 3D point on G based on a projective approximation.

Under a perspective camera model, the image location $\mathbf{q} = [x_c, y_c]^T \in \mathbb{R}^2$ obtained by projection of a 3D point $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^3$ can be expressed as

$$h(\mathbf{p}) = \lfloor \pi(KR^{-1}\mathbf{p}) \rfloor \,, \tag{1}$$

where π is the perspective projection function, that is, $\pi([x, y, z] = [x/z, y/z])$, $\lfloor\rfloor$ is the *floor* operator, and K and R are the matrix of camera intrinsic parameters and the camera rotation matrix, respectively. The latter only specifies the tilt angle, allowing us to align the scene according to the gravity direction. Furthermore, given a depth image

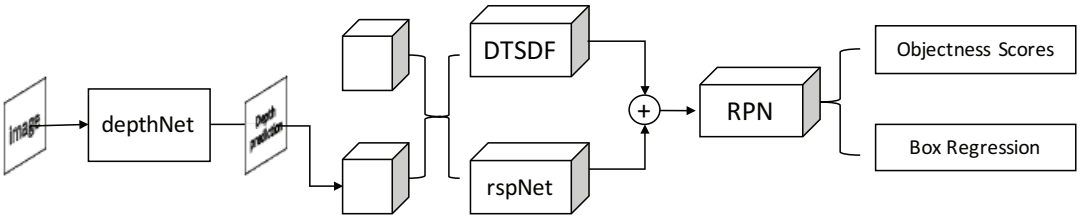

Figure 1: Our 3D object proposal framework. Our model consists of three parts integrated in a single architecture: a depth estimation network (DepthNet); a residual module to convert the predicted depth into a volumetric representation; a region proposal network (RPN). The middle part, which constitutes our key contribution, consists of a differentiable TSDF (DTSDF) encoding and of a residual-side-path network (rspNet) accounting for the predicted depth inaccuracies. These two subnetworks take two 3D grids as input, which correspond to a voxelization of the scene 3D volume and the projection of the depth map in this voxelization (see text for more detail). Ultimately, our model outputs the coordinates of 3D candidate boxes and corresponding objectness scores.

$D \in \mathbb{R}^{M \times N}$, the 3D point $\hat{\mathbf{p}} = [\hat{x}, \hat{y}, \hat{z}]^T$ at image location $\mathbf{q}$ can be obtained

$$\hat{\mathbf{p}} = c \cdot D(q), c = RK^{-1}[\mathbf{q}^T, 1]^T \ . \tag{2}$$

At each grid location $\mathbf{p}$ on G, the projective TSDF value can thus be obtained by projecting $\mathbf{p}$ and computing the distance between $\mathbf{p}$ the corresponding depth map point $\hat{\mathbf{p}}$.

More precisely, here, we consider the distances in the three directions x, y and z separately. Let $g(\mathbf{p}, \hat{\mathbf{p}}) = \mathbf{p} - \hat{\mathbf{p}}$, and $\psi = \|g(\mathbf{p}, \hat{\mathbf{p}})\|_2$ is the distance. The difference in z direction can be computed as $n(\hat{\mathbf{p}}) = [0, 0, 1](R^{-1}\mathbf{p} - \hat{\mathbf{p}})$. Then, we define the truncated signed distance functions in x, y and z as

$$f(\mathbf{p}, \hat{\mathbf{p}}) = \begin{cases} \mathbf{1}\mu \cdot s(n(\hat{\mathbf{p}})) & \text{if } \psi/\delta \geq 1 \\ \min(g(\mathbf{p}, \hat{\mathbf{p}}), \mathbf{1}\mu) \cdot s(n(\hat{\mathbf{p}})) & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{1} \in \mathbb{R}^3$ is a vector of ones, and $\min(\cdot)$ computes the element-wise minimum; $\delta = 0.1$, $\mu = 0.05$ and $s(x) = \tanh(k \cdot x)$ (with $k = 10$ in our experiments) truncates the distance to a signed constant. In words, $\hat{\mathbf{p}}$ encodes the surface points, and the sign of the distance then indicates whether the cell falls behind the surface, that is the cell is invisible (positive distance), or if it is visible (negative distance). We further assign zero values to the grid locations of G whose 2D projections fall out of the image range, which we refer to as invalid grid regions afterwards.

With this definition, the TSDF value at each location $\mathbf{p}$ on G is differentiable with respect to $\hat{\mathbf{p}}$. Therefore, following the chain rule, since the values of $\mathbf{p}$, K, R, and therefore $\mathbf{q}$ are fixed, the TSDF values are differentiable with respect to the depth prediction D. This will then allow us to employ this volumetric representation in an end-to-end learning framework.

Residual Network for Volumetric Representation The DTSDF described above relies on the distance along the visual ray and, as mentioned before, is only an approximation to the true TSDF; the true nearest-neighbor to a 3D point might not be on its visual ray. We have found, however, that, in practice, the true neighbor is usually not far away from this approximation. Motivated by this observation, and to further account for the inaccuracies in the estimated depth map, we introduce a residual path to improve the DSTDF.

Specifically, the input to our residual-side-path network (rspNet) consists of a 3D grid similar to G defined above. However, instead of only encoding the 3D position of the corresponding cell center at each location, we further append to this position the coordinates of the corresponding surface point $\hat{\mathbf{p}}$ along the visual ray. This 3D grid with 6 channels then acts as input to an encoder-decoder network with the following structure

$$\begin{aligned} &input \rightarrow conv3d \rightarrow relu \rightarrow pool3d \rightarrow \\ &\rightarrow conv3d \rightarrow relu \rightarrow pool3d \rightarrow deconv3d \rightarrow \\ &\rightarrow relu \rightarrow deconv3d \rightarrow relu \rightarrow output \end{aligned} \tag{4}$$

where conv3d, deconv3d, pool3d represent convolution, deconvolution, pooling operations on a 3D grid, respectively. The parameters defining the layers of our rspNet are given in Table 1. Note that all convs/deconvs here contain no bias, so as to guarantee zero values in invalid grid regions.

Intuitively, the given input information allows the rspNet to compute the distance between the cell center and the corresponding surface point as in the DTSDF. However, by performing convolutions, it is also able to compensate for errors by looking at larger regions in the reconstructed volume and capture the necessary information for prediction.

Altogether, our approach to volumetric representation prediction lets us effectively leverage the strengths of the explicit DTSDF computation and of the learning-based rspNet. While the explicit computation is less flexible, it provides with a reliable approximation of the true TSDF. By contrast, access to limited data might make it hard to use the rspNet on its own, but it provides more flexibility to compensate for the DTSDF and the depth prediction mistakes. The resulting volumetric representation is then used as input to the region proposal network described below.

3D Object Proposal Generation

3D Object Proposal Network We follow the recent trend in generating object proposals consisting of sharing feature computation, thus speeding up runtime (Ren et al. 2015; Song and Xiao 2016). In particular, since we work in 3D, we

	conv1	pool1	conv2	pool2	deconv1	deconv2
kernel	[3,3,3]	[2,2,2]	[3,3,3]	[2,2,2]	[3,3,3]	[3,3,3]
channels	3	3	3	3	3	3
stride	1	2	1	2	2	2

Table 1: Parameters of our residual-side-path network. The kernel size is represented in the order [width, length, height]. In the first convolution, we convert from the original 6 channels (3D cell center + 3D nearest point position along the visual ray) to 3. In the remaining layers, we maintain the number of channels to 3 to match the volumetric representation of the DTSDF and keep the memory requirement relatively low. For each layer, the strides are the same for all three dimensions. We do not use any biases in our layers.

adopt the multi-scale 3D region proposal network (RPN) of the Deep Sliding Shapes (DSS) method of (Song and Xiao 2016). This network relies on a volumetric representation as input, and is thus well suited to be appended the framework discussed above. As output, it produces another volume at a lower resolution. To each voxel of the output volume are associated J anchors ($J = 19$ in our work). These anchors represent potential 3D object bounding boxes of different sizes and aspect ratios, and were defined according to the statistics of the training data. The RPN then predicts a probability for each anchor to correspond to an actual object at each cell location. Furthermore, it also regresses a 6D vector encoding the center position and the three side lengths of the corresponding 3D bounding box. Instead of pooling the features for each anchor separately, all the anchors of each voxel share their features, but have different classifiers/regressors.

Extended Anchors In (Song and Xiao 2016), runtime and accuracy were improved by removing empty anchors based on the input depth map. Here, however, we do not have access to the ground-truth depth maps, and our depth predictions are imperfect. In practice, we found that too many boxes were removed by this procedure. Furthermore, we also observed that our depth prediction often essentially differed from the true one by a single scale factor. Motivated by this, we therefore propose to enlarge the anchor pool by scaling the depth maps.

The range of the scale factors between predicted and true depth maps on the training set was found to be [0.8, 1.2]. We therefore scale the depth prediction with a global scale in this range with a stride 0.05. Assuming that all resulting scaled maps are valid ones, we only remove the anchors that do not contain any points of the scaled depth maps. Since this procedure can quickly lead to a huge number of anchors to consider, thus increasing runtime, we performed 3D non-maximum suppression to remove anchors with a large overlap. We further limited the number of valid anchors to 15,000 for each anchor type. Specifically, we keep all the non-empty anchors in the original, unscaled depth maps, and add anchors from the scaled depth maps by scoring them according to the corresponding inverse absolute scale difference with 1, e.g., $1/|1.2 - 1|$.

As evidenced by our experiments, the quality of the anchors is important to our results; our extended anchors allow us to obtain dense supervision in a huge 3D space during training, and, at test time, it prevents high-scoring proposals from not being considered because they have been removed from the candidate pool. Nevertheless, as evidenced by our ablation study, the extended anchors are not the only key to the success of our approach; it rather helps boosting the effectiveness of our residual volumetric prediction module.

Multi-task Loss and Network Training

We now turn to the problem of training our model. Assuming that we have access to ground-truth depth maps during training, we propose to define a multi-task loss consisting of two parts. The first one measures depth prediction error, and the second encodes errors on the generated proposal themselves. Specifically, we define our loss as

$$\mathcal{L} = \lambda \mathcal{L}_{depth}(D, D^*) + \sum_{i=1}^{N} \mathcal{L}_{rpn}(p_i, p_i^*, t_i, t_i^*) \ , \qquad (5)$$

where $\mathcal{L}_{depth}$ is the depth loss between the predicted depth map D and the ground-truth one D^*, and $\mathcal{L}_{rpn}$ is the object proposal loss comparing the predicted class probability p_i and regressed box parameters t_i with the ground-truth ones p_i^* and t_i^* for each of the N candidate anchors.

For the depth loss, we adopt the same loss function as in (Eigen, Puhrsch, and Fergus 2014). In practice, as evidenced by our experiments, we have found the depth loss to be important, as it ensures that the input to the DTSDF and to the rspNet remains meaningful for volumetric representation prediction.

The object proposal loss consists of two parts: a softmax loss for classification of object vs non-object and a smooth ℓ_1 loss on the regression variables. The 3D regression loss is a direct extension of the one commonly used in 2D (Ren and Sudderth 2016; Girshick 2015). Assuming that the objects lie on the ground, a 3D bounding box can be defined by 7 parameters, $[X, Y, Z, L, H, W, \theta]$, where the first three are the coordinates of the box center, the following three are the side lengths in the three directions, and θ is the orientation. In this paper, we approximate the orientation of each object by the global orientation of scene, which can be estimated from the predicted depth (Uijlings et al. 2013). We are therefore left with 6 parameters to estimate.

Let us denote by $[X^*, Y^*, Z^*, L^*, H^*, W^*]$, $[X, Y, Z, L, H, W]$, and $[X_a, Y_a, Z_a, L_a, H_a, W_a]$ the parameters of a ground-truth box, a predicted one, and an anchor, respectively. To keep the magnitudes of these different values more comparable, we make use of relative

values defined as

$$t_x = \frac{(X - X_a)}{W_a}, t_y = \frac{(Y - Y_a)}{H_a}, t_z = \frac{(Z - Z_a)}{L_a}$$
$$t_w = \log\left(\frac{W}{W_a}\right),\ t_h = \log\left(\frac{H}{H_a}\right),\ t_l = \log\left(\frac{L}{L_a}\right)$$
$$t^*_x = \frac{(X^* - X_a)}{W_a}, t^*_y = \frac{(Y^* - Y_a)}{H_a}, t^*_z = \frac{(Z^* - Z_a)}{L_a}$$
$$t^*_w = \log\left(\frac{W^*}{W_a}\right),\ t^*_h = \log\left(\frac{H^*}{H_a}\right),\ t^*_l = \log\left(\frac{L^*}{L_a}\right)\ , \tag{6}$$

where $[t^*_x, t^*_y, t^*_z, t^*_w, t^*_h, t^*_l]$ denote ground-truth values and $[t_x, t_y, t_z, t_w, t_h, t_l]$ predicted ones.

Altogether, our object proposal loss can thus be written as

$$\mathcal{L}_{rpn}(p, p^*, t, t^*) = \mathcal{L}_{cls}(p, p^*) + \gamma \sum_{i \in s} r(t^*_i - t_i)$$

where

$$L_{cls}(p, p^*) = -p^* log(p) - (1 - p^*) log(1 - p) \tag{7}$$

and

$$r(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

is the smooth ℓ_1 loss.

In practice, we first generate depth maps using the depth network. We then remove the empty anchors, following the procedure used to generate the extended anchors discussed above, based on the depth predictions. At test time, we similarly remove the empty boxes based on the predicted depth maps, and further perform 3D non-maximum suppression according to the predicted probabilities p, with a threshold of 0.35 on the volumetric IoU. Furthermore, we only keep the top K proposals ($K = 150$ in our work) among the anchors of each of the 19 categories.

Experiments

We evaluate our model on the NYUv2 dataset (Silberman et al. 2012), which consists of RGB images with their corresponding depth maps. The ground-truth 3D bounding boxes are provided with the SUN RGB-D dataset (Song, Lichtenberg, and Xiao 2015). NYUv2 contains 795 training images and 654 test images. In our experiments, we use only RGB images as input, while ground-truth depth maps are employed for supervision during training. For the reconstruction volume, we adopt the range and resolution used in (Song and Xiao 2016), that is, [-2.6, 2.6] meters horizontally, [-1.5,1] meters vertically, and [0.4, 5.6] meters in depth. Each cell has side lengths of 0.025 meters. Altogether this yields a volumetric representation of size $208 \times 100 \times 208$.

We implemented our model in tensorflow, and trained it on two NVIDIA Tesla P100, each with 16GB memory. We mini-batches containing one image each iteration. For each batch, we sampled negative anchors in a ratio of 1.2 w.r.t. the positive anchors, with the pos/neg labels assigned according to the rules of (Song and Xiao 2016). We set the initial learning rate to 0.001 and decreased it at a rate of 0.5 every 2 epochs. The final network was selected using a validation set of 50 images, which was taken out of the standard training set. In our experiments, we trained our model for at most 15 epochs. Training our deep network takes roughly 10 hours, and inference takes 2.72s per image on average.

Baselines

As mentioned before, this work constitutes the first attempt to tackle the problem of generating class-independent 3D box proposals from a single monocular image. Therefore, we developed our own baselines by making use of the state-of-the-art monocular depth estimation network of (Eigen and Fergus 2015) (with the all three scales, compared to two scales in our framework), followed by the state-of-the-art depth-based 3D proposal generation method of (Song and Xiao 2016). We further also designed a baseline inspired by the effective faster R-CNN framework of (Ren et al. 2015). In practice, we trained the baselines using mini-batched of the same size as ours, and an initial learning rate of 0.001. We discuss these baselines in more detail below.

Est-DSS The original DSS framework (Song and Xiao 2016) consists of a class-independent multi-scale region proposal network trained on the accurate TSDF representations of input depth maps. To work in our monocular setting, we replace the ground-truth depth maps with those predicted by the three scale model of (Eigen, Puhrsch, and Fergus 2014). DSS relies on the accurate TSDF, computed from the true nearest neighbor of each voxel. To better understand the accuracy loss incurred by relying on the projective TSDF in our model, we further developed another baseline, named **Est-DSS-Approx**, where we replaced the accurate TSDF with our approximate one within the DSS framework.

Est-Faster-RCNN Another approach consists of directly predicting 3D bounding boxes from the 2D image. To develop a baselines that works in this setting, we made use of the state-of-the-art faster R-CNN 2D detection framework of (Ren et al. 2015). More precisely, we modified this framework to regress 3D coordinates from 2D anchors. To nonetheless exploit depth information, we extracted HHA features (Gupta et al. 2014) from the depth predictions. These features, in conjunction with color images, acted as input to train the proposal generation part of the faster R-CNN. Specifically, we placed 12 different 2D anchors at each node on the image grid. For each anchor, we then regress the depth of its 3D box center, the coordinates of the 2D projection of the box center, and its height, width and length in 3D space, relative to the anchor center and size, similarly to Eq. 6. As in our framework, the orientation of each box was estimated according to the global orientation of the scene.

Evaluation Metrics

We evaluate the accuracy of 3D object proposals by calculating their recall, according to a volume overlap with ground-truth larger than 0.25, and their average box overlap (ABO) with the ground-truth. Note that, in NYUv2, the ground-truth consists of 3D box parameters in the world referential

methods	bathtub	bed	bookshelf	box	chair	counter	desk	door	dresser	bin	lamp
Est-Faster-RCNN	33.3	84.5	59.8	9.9	82.1	84.6	82.9	4.9	76.4	45.8	34.5
Est-DSS-Approx	70.8	94.8	44.8	10.6	83.8	92.3	79.4	4.9	83.6	30.5	27.2
Est-DSS	75.0	95.5	49.4	10.6	85.3	84.6	84.4	11.8	90.9	37.3	23.6
Ours	70.8	93.5	34.5	**15.6**	**87.4**	**92.3**	82.9	5.9	83.6	37.3	20.0
	monitor	pillow	nightstand	sink	sofa	table	tv	toilet	Recall	ABO	#Box
Est-Faster-RCNN	8.3	27.3	70.8	53.2	81.2	80.4	24.2	93.3	62.3	0.319	2000
Est-DSS-Approx	4.2	17.9	81.3	71.4	90.6	89.7	21.2	96.7	63.6	0.346	2000
Est-DSS	0.00	24.1	79.2	72.7	92.0	91.8	27.3	96.7	66.1	0.348	2000
Ours	**25.0**	**48.3**	**81.3**	**85.7**	89.9	89.7	**33.3**	93.3	**69.3**	**0.364**	2000

Table 2: Comparison of our model with the baselines on NYUv2. We show the class-wise recalls and overall recall and ABO of the 2000 top scored 3D windows on test set. Note that our model outperforms the two-stage baselines and the faster R-CNN one in both overall recall and ABO, thus showing the benefits of having an end-to-end learning framework.

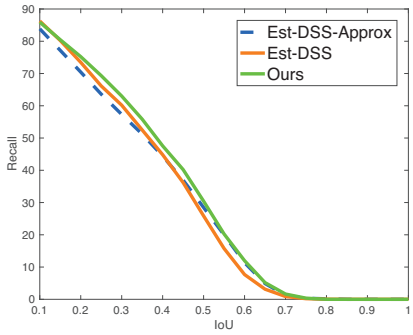

Figure 2: Recall as a function of the IoU threshold. Note that our approach outperforms the two-stage baselines, or performs on par with them, across the whole range of IoU thresholds, while the best performing baseline varies for low and high IoUs. This evidences the stability of our approach.

with a tilt rotation calculated from the ground-truth depth. Since, in our monocular setting, we cannot have access to the ground-truth tilt rotation, we estimate it using the initial depth estimates obtained from the network of (Eigen, Puhrsch, and Fergus 2014).

Experimental Results

We now present our results on the NYUv2 dataset. In Table 2, we first compare our complete model with the three baselines introduced above. For all methods, we selected the 2000 3D bounding boxes with highest score to calculate the recall and ABO. Our model significantly outperforms the baselines in terms of both recall and ABO, thus showing the importance of end-to-end training and the effectiveness of our residual volumetric prediction approach at compensating the errors in depth prediction and due to the TSDF approximation. Even though we tackle the problem of generating class-independent 3D object proposals, we also report the recall for each of the 19 object categories present in the dataset. Note that our model more effectively handles classes of small objects, such as sink, monitor and pillow, which are typically more challenging to detect. This, we believe, demonstrates that, while generic, our end-to-end training mechanism allows us to learn effective class-specific representations automatically. A qualitative comparison of our model with the best-performing Est-DSS baseline on a few images is provided in Figure 3.

methods	recall	ABO	#Box
DSS	84.9	0.461	2000
3D Selective Search	74.2	0.409	2000
Ours	69.3	0.364	2000

Table 3: Comparison to depth-based models on NYUv2. We compare our model with methods based on ground-truth depth. Note that the gap between our model and these depth-based ones is relatively small, despite the fact that we rely only on a monocular image as input.

Note also that exploiting volumetric representations, as done by Est-DSS, Est-DSS-Approx and our approach, seems to be more effective than direct regression to 3D as in our Est-Faster-RCNN baseline, which yields the least accurate proposals. We believe that this is due to the fact that the volumetric representations is better suited to capture the shape of 3D objects and their distances in 3D space. Finally, by comparing Est-DSS-Approx and Est-DSS, we can see that, as expected, the projective TSDF approximation yields worse results. Note, however, that our residual framework manages to compensate for this loss of accuracy, as better evidenced in the ablation study below.

In Figure 2, we plot the recall as a function of the IoU threshold for our method and the two-stage baselines. Note that we generally outperform, or perform on par with, the baselines on the whole range of IoU values. Note also that the best performing baseline differs for low and high IoU thresholds. This, we believe, further demonstrates the stability of our model.

Comparison to Depth-based Models In Table 3, we compare our results with those of depth-based models, i.e., DSS (Song and Xiao 2016) and 3D selective search (Uijlings et al. 2013), whose results were obtained using the implementation by (Song and Xiao 2016). While our model, which relies only on an RGB image as input, yields slightly worse results that these methods that exploit ground-truth depth, the gap is remarkably small; e.g., we achieve only 4.9% lower recall than 3D selective search. Considering the

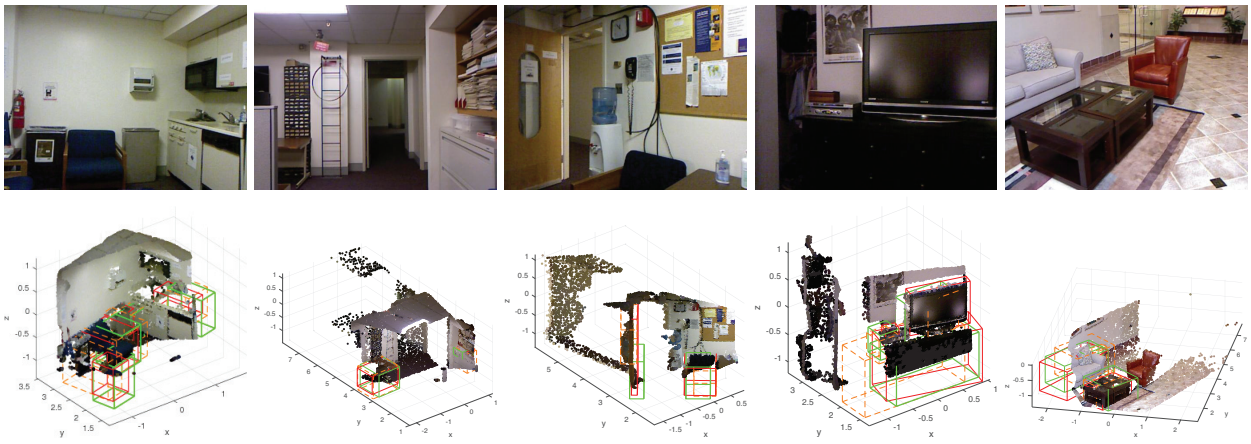

Figure 3: Qualitative comparison of our model with Est-DSS. We show the proposals with highest IoU returned by our model and by the baseline. The results of our model are shown in green, those of the baseline in dashed orange and the ground-truth boxes in red. Note that our proposals better match the ground-truth ones.

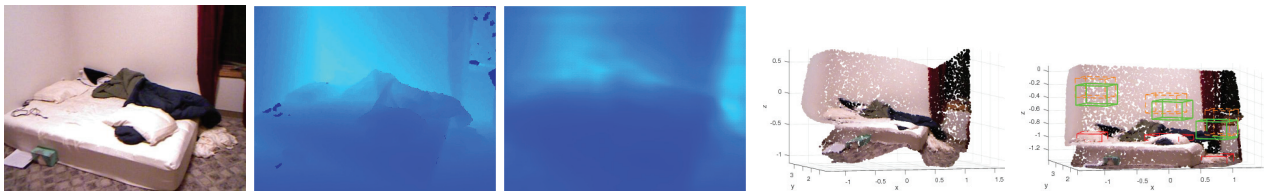

Figure 4: A failure example. Due to the similar appearance of the foreground (pillow) with the background (bed), the depth cannot be predicted very accurately. Inaccurate depth estimation leads to the failure of our method as well as the baseline method. The results of our model are shown in green, those of the baseline in dashed orange and the ground-truth boxes in red.

strong ambiguities of depth estimation from a monocular image, we believe that this small gap shows the effectiveness of our monocular 3D box proposal model.

Ablation Study In addition to the previous baseline comparison, we perform a comprehensive analysis of the impact of the different components of our approach. In particular, we evaluate the importance of (i) making our framework end-to-end trainable; (ii) relying on the accurate TSDF compared to the approximate one; (iii) our residual network for volumetric prediction; (iv) our extended anchors; (v) the use of the depth loss as intermediate supervision.

The results of this ablation study are provided in Table 4. In short, we can see that (i) our residual volumetric prediction is able to compensate for the loss in accuracy incurred by the use of the projective TSDF; (ii) Our extended anchors help further boost the accuracy of our model, while they only have little effect when used in conjunction with the two-stage baseline; (iii) depth supervision is important for our model, as it ensures that the input to our volumetric prediction remains meaningful. In our experiments, we set the weight of the depth loss λ to 1. We found, however, that our results were robust to this value, as long as it is sufficiently large. For example, with $\lambda = 10$, our model still outperforms the baseline with a recall of 67.4 and an ABO of 0.359. Interestingly, we have observed that our depth estimates, while not globally more accurate than the initial ones, better separate the foreground objects from the background, which seems natural since we aim to generate proposals for the foreground objects. Altogether, we believe that this ablation study clearly evidences the strengths of the different components of our approach.

Generalization of our Model To demonstrate the generality of our approach, we make use of the SUN-RGBD dataset (Song, Lichtenberg, and Xiao 2015) to test our model and the Est-DSS baseline. The SUN-RGBD dataset consists of 5050 test images, including some from the NYUv2 dataset. For this evaluation to be more meaningful, we do not fine-tune the models, and explicitly exclude the NYUv2 images from the test set, thus leaving us with 4395 images. In practice, since the image size of SUN-RGBD changes, we simply resize them all to the size of the NYUv2 images, i.e., [427,561]. Furthermore, to adjust the camera intrinsic matrix to the new image size, we multiply the focal lengths and the principle point by the ratio of the NYUv2 image size to the original SUN-RGBD size of each input image.

The results of this experiment are provided in Table 5. Our model achieves a performance similar to that on the NYUv2 dataset. Furthermore, it outperforms the baseline in both recall and ABO, thus showing the ability of our model to generalize to new data.

End-to-End Trainable	Acc.TSDF	Residual Path	Ext. Anchors	Depth Loss	Recall	ABO
✗	✗	✗	✗	✗	63.6	0.346
✗	✓	✗	✗	✗	66.1	0.348
✗	✓	✗	✓	✗	65.7	0.356
✓	✗	✓	✓	✗	54.7	0.298
✓	✗	✓	✗	✓	66.3	0.360
✓	✗	✗	✓	✓	67.4	0.356
✓	✗	✓	✓	✓	69.3	0.364

Table 4: Ablation study on NYUv2. We evaluate the influence of different components of our framework. Note that our residual volumetric prediction module is able to effectively compensate for our use of an approximate TSDF representation, and can be further improved by the use of our extended anchors, which, by contrast, have only little effect on the two-stage baseline.

methods	recall	ABO	#Box
Est-DSS	64.1	0.328	2000
Ours	67.9	0.353	2000

Table 5: Generality study on SUN-RGBD excluding NYUv2: We evaluate our model and a baseline model, which are both trained on NYUv2, on a subset of SUN-RGBD excluding those from NYUv2. Here it shows that our model remains more effective than the baseline on this data, thus evidencing the generality of our approach.

For completeness, we also report the results of our approach on the complete SUN-RGBD test set, including the images from NYUv2. This corresponds to a recall of 68.1%, and an ABO of 0.354. Both of which are slightly higher than our results on the previous SUN-RGBD subset.

Conclusion

We have introduced an end-to-end method to generate class-independent 3D object proposals from a single monocular image. To the best of our knowledge, this constitutes the first attempt to work in this challenging setting for complex indoor scenes. Our experiments have demonstrated that our residual, fully-differentiable TSDF module produces an effective volumetric representation to generate box proposals, thus outperforming the two-stage approach based on the standard, non-differentiable TSDF. We have found that depth supervision was beneficial to our model. Importantly, however, our model does not require accurate depth on all parts of the image. In particular, the accuracy of the background depth is unimportant since we focus on foreground objects. In the future, we therefore plan to modify the depth loss to focus more strongly on the foreground objects.

Acknowledgement

The first author is supported by the Chinese Scholarship Council and CSIRO-Data61. The authors would like to thank CSIRO, for providing the GPU cluster used for all experiments in this paper. This project was also partially supported by the Program of Shanghai Subject Chief Scientist (A type) (No.15XD1502900).

References

Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; and Urtasun, R. 2016a. Monocular 3d object detection for autonomous driving. In *CVPR*.

Chen, X.; Kundu, K.; Zhu, Y.; Ma, H.; Fidler, S.; and Urtasun, R. 2016b. 3d object proposals using stereo imagery for accurate object class detection. *arXiv preprint arXiv:1608.07711*.

Cheng, M.-M.; Zhang, Z.; Lin, W.-Y.; and Torr, P. 2014. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*.

Dai, J.; He, K.; and Sun, J. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*.

Deng, Z., and Latecki, L. J. 2017. Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images. In *CVPR*.

Dwibedi, D.; Malisiewicz, T.; Badrinarayanan, V.; and Rabinovich, A. 2016. Deep cuboid detection: Beyond 2d bounding boxes. *arXiv preprint arXiv:1611.10010*.

Eigen, D., and Fergus, R. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*.

Eigen, D.; Puhrsch, C.; and Fergus, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*.

Fidler, S.; Dickinson, S.; and Urtasun, R. 2012. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*.

Girdhar, R.; Fouhey, D. F.; Rodriguez, M.; and Gupta, A. 2016. Learning a predictable and generative vector representation for objects. In *ECCV*.

Girshick, R. 2015. Fast r-cnn. In *ICCV*.

Gupta, S.; Girshick, R.; Arbeláez, P.; and Malik, J. 2014. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*.

Gupta, S.; Arbeláez, P.; Girshick, R.; and Malik, J. 2015. Aligning 3d models to rgb-d images of cluttered scenes. In *CVPR*.

Hayder, Z.; He, X.; and Salzmann, M. 2016. Learning to co-generate object proposals with a deep structured network. In *CVPR*.

Hayder, Z.; He, X.; and Salzmann, M. 2017. Boundary-aware instance segmentation. In *CVPR*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. *ICCV*.

Hedau, V.; Hoiem, D.; and Forsyth, D. 2010. Thinking inside the box: Using appearance models and context based on room geometry. *ECCV*.

Karsch, K.; Liu, C.; and Kang, S. B. 2012. Depth extraction from video using non-parametric sampling. In *ECCV*.

Ladicky, L.; Shi, J.; and Pollefeys, M. 2014. Pulling things out of perspective. In *CVPR*.

Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; and Navab, N. 2016. Deeper depth prediction with fully convolutional residual networks. In *3DV*.

Lin, D.; Fidler, S.; and Urtasun, R. 2013. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*.

Liu, F.; Shen, C.; Lin, G.; and Reid, I. 2016. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI* 38(10).

Liu, M.; Salzmann, M.; and He, X. 2014. Discrete-continuous depth estimation from a single image. In *CVPR*.

Newcombe, R. A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A. J.; Kohi, P.; Shotton, J.; Hodges, S.; and Fitzgibbon, A. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*.

Payet, N., and Todorovic, S. 2011. From contours to 3d object detection and pose estimation. In *ICCV*.

Pinheiro, P. O.; Collobert, R.; and Dollár, P. 2015. Learning to segment object candidates. In *NIPS*.

Pont-Tuset, J.; Arbelaez, P.; Barron, J. T.; Marques, F.; and Malik, J. 2017. Multiscale combinatorial grouping for image segmentation and object proposal generation. *TPAMI* 39(1).

Ren, Z., and Sudderth, E. B. 2016. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor segmentation and support inference from rgbd images. *ECCV*.

Song, S., and Xiao, J. 2016. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*.

Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*.

Uijlings, J. R.; van de Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective search for object recognition. *IJCV* 104(2).

Wu, J.; Xue, T.; Lim, J. J.; Tian, Y.; Tenenbaum, J. B.; Torralba, A.; and Freeman, W. T. 2016. Single image 3d interpreter network. In *ECCV*.

Zhou, T.; Brown, M.; Snavely, N.; and Lowe, D. G. 2017. Unsupervised learning of depth and ego-motion from video. In *CVPR*.

Zhuo, W.; Salzmann, M.; He, X.; and Liu, M. 2015. Indoor scene structure analysis for single image depth estimation. In *CVPR*.