DeepContext: Context-Encoding Neural Pathways for 3D Holistic Scene Understanding

Yinda Zhang[1] Mingru Bai[1] Pushmeet Kohli[2,5] Shahram Izadi[3,5] Jianxiong Xiao[1,4]

[1]Princeton University [2]DeepMind [3]PerceptiveIO [4]AutoX [5]Microsoft Research

Abstract

3D context has been shown to be extremely important for scene understanding, yet very little research has been done on integrating context information with deep neural network architectures. This paper presents an approach to embed 3D context into the topology of a neural network trained to perform holistic scene understanding. Given a depth image depicting a 3D scene, our network aligns the observed scene with a predefined 3D scene template, and then reasons about the existence and location of each object within the scene template. In doing so, our model recognizes multiple objects in a single forward pass of a 3D convolutional neural network, capturing both global scene and local object information simultaneously. To create training data for this 3D network, we generate partially synthetic depth images which are rendered by replacing real objects with a repository of CAD models of the same object category[1]. Extensive experiments demonstrate the effectiveness of our algorithm compared to the state of the art.

1. Introduction

Understanding indoor scene in 3D space is critically useful in many applications, such as indoor robotics, augmented reality. To support this task, the goal of this paper is to recognize the category and the 3D location of furniture from a single depth image.

Context has been successfully used to handle this challenging problem in many previous works. Particularly, holistic scene context models, which integrate both the bottom up local evidence and the top down scene context, have achieved superior performance [6, 23, 24, 48, 49]. However, they suffer from a severe drawback that the bottom up and top down stages are run separately. The bottom up stage using only the local evidence needs to generate a large quantity of noisy hypotheses to ensure a high recall, and the top down inference usually requires combinatorial algorithms, such as belief propagation or MCMC, which are computationally expensive in a noisy solution space. Therefore, the whole combined system can hardly achieve a reasonably optimal solution efficiently and robustly.

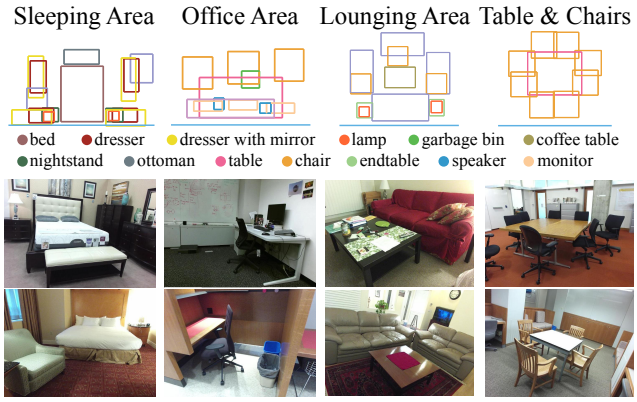

Figure 1. **Example of canonical scene templates (top view) and the natural images they represent.** We learn four scene templates from SUN-RGBD[32]. Each scene template encodes the canonical layout of a functional area.

Inspired by the success of deep learning, we propose a 3D deep convolutional neural network architecture that jointly leverages local appearance and global scene context efficiently for 3D scene understanding.

Designing a deep learning architecture to encode context for scene understanding is challenging. Unlike an object whose location and size can be represented with a fixed number of parameters, a scene could involve unknown number of objects and thus requires variable dimensionality to represent, which is hard to incorporate with convolutional neural network with a fixed architecture. Also, although holistic scene models allow flexible context, they require common knowledge to manually predefine relationship between objects, e.g. the relative distance between bed and nightstands. As a result, the model may unnecessarily encode weak context, ignore important context, or measure context in an over simplified way.

To solve these issues, we propose and learn a scene representation encoded in scene templates. A scene template contains a super set of objects with strong contextual correlation that could possibly appear in a scene with relatively constrained furniture arrangements. It allows a prediction

[1]Code and dataset are available at http://deepcontext.cs.princeton.edu. Part of this work is done when Yinda Zhang was an intern at Microsoft Research, Jianxiong Xiao was at Princeton University, Pushmeet Kohli and Shahram Izadi were at Microsoft Research.

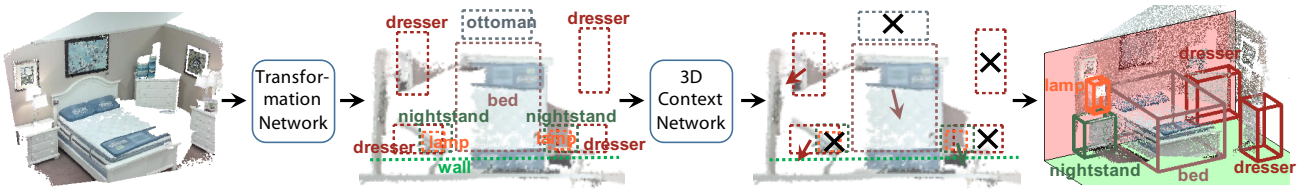

Figure 2. **Our deep 3D scene understanding pipeline.** Given a 3D volumetric input derived from a depth image, we first aligns the scene template with the input data. Given the initial alignment, our 3D context network estimates the existence of an object and adjusts the object location based on local object features and holistic scene feature, to produce the final 3D scene understanding result.

of "not present" for the involved objects so that a variety of scenes can be represented with a fixed dimensionality. A scene can be considered as a scene template with a subset of objects activated. Scene template also learns to only consider objects with strong context, and we argue that contextless objects, such as a chair can be arbitrarily placed, should be detected by a local appearance based object detector.

Each template represents a functional sub-region of an indoor scene, predefined with canonical furniture arrangements and estimated 3D anchor positions of possible objects with respect to the reference frame of the template. We incorporate these template anchors as priors in the neural architecture by designing a transformation network that aligns the input 3D scene (corresponding to the observed depth image) with the template (i.e. the canonical furniture arrangement in 3D space). The aligned 3D scene is then fed into a 3D context neural network that determines the existence and location of each object in the scene template. This 3D context neural network contains a holistic scene pathway and an object pathway using 3D Region Of Interest (ROI) pooling in order to classify object existence and regress object location respectively. Our model learns to leverage both global and local information from two pathways, and can recognize multiple objects in a single forward pass of a 3D neural network. It is noted that we do not manually define the contextual relationships between objects, but allow the network to automatically learn context in arbitrary format across all objects.

Data is yet another challenging problem for training our network. Holistic scene understanding requires the 3D ConvNet to have sufficient model capacity, which needs to be trained with a massive amount of data. However, existing RGB-D datasets for scene understanding are all small. To overcome this limitation, we synthesize training data from existing RGB-D datasets by replacing objects in a scene with those from a repository of CAD models from the same object category, and render them in place to generate partially synthesized depth images. Our synthetic data exhibits a variety of different local object appearances, while still keeping the indoor furniture arrangements and clutter as shown in the real scenes. In experiments, we use these synthetic data to pretrain and then finetune our network on a small amount of real data, whereas the same network directly trained on real data can not converge.

The contributions of this paper are mainly three aspects. 1) We propose a scene template representation that enables the use of a deep learning approach for scene understanding and learning context. The scene template only encodes objects with strong context, and provides a fixed dimension of representation for a family of scenes. 2) We propose a 3D context neural network that learns scene context automatically. It leverages both global context and local appearance, and detects all objects in context efficiently in a single forward pass of the network. 3) We propose a hybrid data augmentation method, which generates depth images keeping indoor furniture arrangements from real scenes but containing synthetic objects with different appearance.

Related Work The role of context has been studied extensively in computer vision [1, 3, 4, 5, 8, 10, 11, 13, 18, 19, 20, 21, 25, 27, 29, 30, 36, 37, 38, 40, 41, 42, 43, 44]. While most existing research is limited to 2D, there are some works on modeling context for total scene understanding from RGB-D images [15, 23, 31, 39, 48]. In term of methodology, most of such approaches take object detection as the input and incorporate context models during a post-processing. We aim to integrate context more tightly with deep neural network for object detection.

There are some efforts incorporating holistic context model for scene understanding, which is closely related to our work. Scene context is usually manually defined as a unary term on a single object, pairwise term between a pair of objects to satisfy certain functionality [23, 46], or a more complicated hierarchy architecture [6, 24, 49]. The learned context models are usually applied on a large set of object hypotheses generated using local evidence, e.g. line segments [49] or cuboid [23], by energy minimization. Therefore high order context might be ignored or infeasible to optimize. Context can be also represented in a non-parametric way [48], which potentially enables high order context but is more computationally expensive to infer during the testing time. In contrast, our 3D context network does not require any heuristic intervene on the context and learns context automatically. We also require no object hypothesis generation, which is essential in making our method more computationally efficient.

Deep learning has been applied to 3D data, but most of

these works focus on modeling objects [45] and object detection [26, 34]. Recently, some successes have been made on applying deep learning for inverse graphics [16, 17]. Our approach goes one step further to embrace the full complexity of real-world scenes to perform holistic scene understanding. Related to our transformation network, Spatial Transformation Networks [14] can learn the transformation of an input data to a canonical alignment in an unsupervised fashion. However, unlike MNIST digits (which were considered in [14]) or an individual object where an alignment to a canonical viewpoint is quite natural, it is not clear what transforms are needed to reach a canonical configuration for a 3D scene. We define the desired alignment in template coordinates and use supervised training by employing the ground truth alignments available from our training data.

While many works have considered rendering synthetic data for training (a.k.a, graphics for vision, or synthesis for analysis), these efforts mostly focus on object rendering, either in color [22, 35] or depth [33]. There is also work rendering synthetic data from CAD model of complicated scenes for scene understanding [12, 47]. However, the generated depth is overly clean, and the scene layouts generated by either by algorithm or human artists are not guaranteed to be correct. In contrast, we utilize both the CAD models and real depth maps to generate more natural data with appropriate context and real-world clutter.

2. Algorithm Overview

Our approach works by first automatically constructing a set of scene templates from the training data (see Section 3.1). Rather than a holistic model for everything in the scene, each scene template only represents objects with context in a sub-area of a scene performing particular functionality. Each template defines a distribution of possible layouts of one or more instances of different object categories in a fixed dimensionality.

Given a depth map of a scene as input[2], we convert it into a 3D volumetric representation of the scene and feed it into the neural network. The neural network first infers the scene template that is suitable to represent the scene, or leaves it to a local appearance based object detector if none of the predefined scene templates is satisfied. If a scene template is chosen, the transformation network estimates the rotation and translation that aligns the scene to the inferred scene template. With this initial alignment, the 3D context network extracts both the global scene feature encoding scene context and the local object features pooled for each anchor object defined in the template, as shown in Fig.2. These features are concatenated together to predict the existence of each anchor object in the template and an offset to adjust its bounding box for a better object fit. The final result is an understanding of the scene with a 3D location and category for each object in the scene, as well as room layout elements including wall, floor, and ceiling, which are represented as objects in the network.

[2]Note that while all the figures in the paper contain color, our system relies only on depth as input without using any color information.

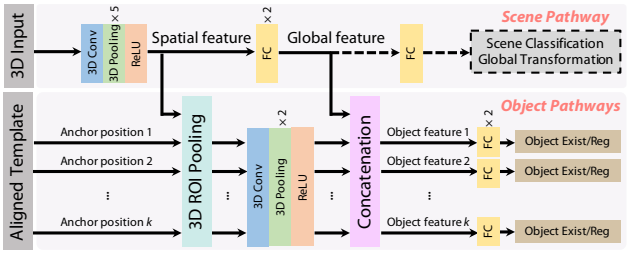

Figure 3. **3D context network.** The network consists of two pathways. The scene pathway takes the whole scene as input and extracts spatial feature and global feature. The object pathway pools local feature from the spatial feature. The network learns to use both the local and global features to perform object detection, including wall, ceiling, and floor.

3. Learning Scene Template

Objects with context in a functional area are usually at relatively fixed locations. For example, a sleeping area is usually composed of a bed with one or two nightstands on the side, with optional lamps on the top. Object detection is likely to succeed by searching around these canonical locations. We learn the categories of object instances and their canonical sizes and locations in the template, from the training data. Examples of each template can be seen in Fig. 1.

3.1. Data-driven Template Definition

We learn to create scene templates using the SUN-RGBD dataset consisting of 10,335 RGB-D images with 3D object bounding box annotations. These RGB-D images are mostly captured from household environments with strong context. As a first experiment of combining 3D deep learning with context, we choose four scene templates: sleeping area, office area, lounging area, and table & chair set, because they represent commonly seen indoor environments with relatively larger numbers of images provided in SUN-RGBD. Our approach can be extended to other functional areas given sufficient training data. For SUN-RGBD, 75% of the images from household scene categories can be described, fully or partially, using these four scene templates.

Our goal is to learn layouts of the scene, such that each template summarizes the bounding box location and category of all objects appearing in the training set. To enable the learning of the template, we select the images that contain a single functional area, and label them with the scene type they belong to. Other images containing arbitrary objects or multiple scene templates are not used in learning scene templates. The ground truth scene categories are used not only for learning the aforementioned templates, but also for learning the scene template classification, the transfor-

mation networks, and the 3D context networks in the following sections.

To obtain the anchor positions (i.e. common locations) for each object type in a template, we take all 3D scenes belonging to this scene template and align them with respect to the center and orientation of a major object[3]. After that, we run k-means clustering for each object type and use the top k cluster centroids as the anchor positions and size, where k is user-defined. We also include room layout elements, including wall, floor, ceiling, which are all represented as regular objects with predefined thickness. Each scene template has tens of object anchors in total for various object categories (Fig. 1).

3.2. Generating Template-Based Ground Truth

To train a 3D context network using scene templates, we need to convert the original ground truth data from SUN RGB-D dataset to a template representation. Specifically, we need to associate each annotated object in the original ground truth with one of the objects defined in the scene template. Similar to above, we first align the training images with their corresponding scene templates using the center and rotation of the major object. For the rest of the objects, we run a bipartite matching between the dataset annotation and the template anchors, using the difference of center location and size as the distance, while ensuring that the objects of the same category are matched.

4. 3D Scene Parsing Network

Given a depth image as input, we first convert it into a 3D volumetric representation, using the Truncated Signed Distance Function (TSDF) [34, 28]. We use a $128 \times 128 \times 64$ grid for the TSDF to include a whole scene, with a voxel unit size of 0.05 meters and a truncation value of 0.15 meters. This TSDF representation is fed into the 3D neural network such that the model runs naturally in 3D space and directly produces output in 3D.

4.1. Scene Template Classification Network

We first train a neural network to estimate the scene template category for the input scene (Fig. 3, Scene pathway). The TSDF representation of the input scene is firstly fed into 3 layers of 3D convolution + 3D pooling + ReLU, and converted to a spatial feature map. After passing through two fully connected layers, the 3D spatial feature is converted to a global feature vector that encodes the information from the whole scene. The global feature is used for scene template classification with a classic softmax layer. During testing, we choose the scene template with the highest score for the input scene if the confidence is high enough (> 0.95). Otherwise, we do not run our method because none of the scene templates fits the input scene. Such scenes are passed to a local appearance based object detector for object detection. In practice, the four scene templates can match with more than half of the images in the SUN-RGBD dataset captured from various of indoor environments.

[3]We manually choose bed for sleeping area, desk for office area, sofa for lounging area, and table for table&chair set as the major objects.

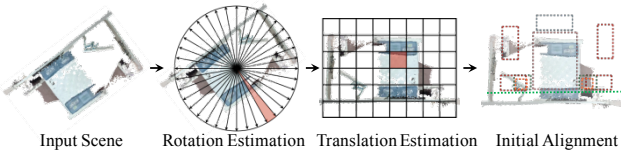

Figure 4. **Transformation estimation.** Our transformation network first produces global rotation and then translation to align the input scene with its scene template in 3D space. Both the rotation and translation are estimated as classification problems.

4.2. Transformation Network

Given the scene template category, our method estimates a global transformation consisting of a 3D rotation and translation that aligns the point cloud of the input scene to the target predefined scene-template (Fig. 4). This is essentially a transformation that aligns the major object in the input scene with that from the scene template. This makes the result of this stage invariant to rotations in the input, and the wall and bounding box of objects are globally aligned to three main directions. The next part of our architecture, the 3D context network, relies on this alignment to obtain the object orientation and the location to pool feature based on 3D object anchor locations from the scene template.

We first estimate the rotation. We assume that the gravity direction is given, e.g. from an accelerometer. In our case, this gravity direction is provided by the SUN RGB-D dataset used in our experiments. Therefore, we only need to estimate the yaw, which rotates the input point cloud in horizontal plane to the scene template viewpoint shown in Fig.1. We divide the 360-degree range of rotation into 36 bins and cast this problem into a classification task (Fig. 4). We train a 3D ConvNet using the same architecture as the scene template classification network introduced in Sec. 4.1 except generating a 36 channel output for softmax. During training, we align each training input scene to the center of the point cloud and add noise for rotations (+/- 10 degrees) and translations ($1/6$ of the range of the point cloud).

For translation, we apply the same network architecture to identify the translation after applying the predicted rotation. The goal is to predict the 3D offset between the centers of the major objects of the input point cloud and its corresponding scene template. To achieve this goal, we discretize the 3D translation space into a grid of 0.5m^3 resolution with dimensions of $[-2.5, 2.5] \times [-2.5, 2.5] \times [-1.5, 1]$, and formulate this task again as a 726-way classification problem (Fig. 4). We tried direct regression with various loss functions, but it did not work as well as classification. We also tried an ICP-based approach, however it could not produce good results.

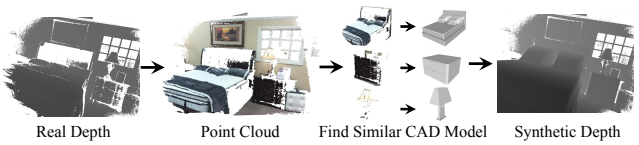

Figure 5. **Hybrid data synthesis.** We first search for similar CAD model for each object. Then, we randomly choose models from good matches, and replace the points in annotated bounding box with the rendered CAD model.

4.3. 3D Context Network

We now describe the context neural network for indoor scene parsing using scene templates. For each scene template defined in the previous section, a separate prediction network is trained. As shown in Fig. 3, the network has two pathways. The global scene pathway, given a 3D volumetric input in a coordinate system that is aligned with the template, produces both a spatial feature that preserves the spatial structure in the input data and a global feature for the whole scene. For the object pathway, we take the spatial feature map from the scene pathway as input, and pool the local 3D Region Of Interest (ROI) based on the 3D scene template for the specific object. The 3D ROI pooling is a max pooling at $6 \times 6 \times 6$ resolution, inspired by the 2D ROI pooling from [9]. The 3D pooled features are then passed through 2 layers of 3D convolution + 3D pooling + ReLU, and then concatenated with the global feature vector from the scene pathway. After two more fully connected layers, the network predicts the existence of the object (a binary classification task) as well as the offset of the 3D object bounding box (3D location and size) related to the anchor locations learned in Sec. 3.1 (a regression task using L1-smooth loss [34]). Including the global scene feature vector in the object feature vector provides holistic context information to help identify if the object exists and its location.

4.4. Training Schema

Our 3D scene parsing network contains a series of components with a large number of parameters. We perform careful training strategy to avoid bad local optima. We first train the scene pathway alone to perform a 4-way scene classification task. After this training converges, we finetune the classification network to estimate the transformation for each individual scene template. An alternative approach is to jointly train a network for classification and transformation, however this does not perform well in practice. The object pathway is then enabled, and the two pathways are jointly finetuned to perform object detection. We found that this form of pretraining, from easy to hard task, is crucial in our experiments. Otherwise, training the four networks independently from scratch cannot produce meaningful models.

5. Synthesizing Hybrid Data for Pre-training

In contrast to existing deep architectures for 3D [34, 45], our model takes the whole scene with multiple objects as input. As such, during training, it needs to model the different variations in the scene layout. We found the RGB-D images from the existing SUN RGB-D [32] dataset are far from sufficient. Furthermore, capturing and annotating RGB-D images on the scale of ImageNet [7] was impractical. To overcome the data deficiency problem, we increase the size of the training data by replacing the annotated objects from SUN RGB-D with CAD models of same category from ShapeNetCore dataset [2] (Fig. 5). This allows us to generate context-valid scenes, as the context still comes from a real environment, while changing the shapes of the objects. By replacing the annotated objects while keeping the full complexity of the areas outside the annotated bounding boxes, we could generate more realistic hybrid data partially maintaining sensor noise. This is in contrast to images generated from purely synthetic models which do not contain clutter caused by the presence of small objects.

To search for similar CAD models for annotated objects in RGB-D images, we need to define the distance between a CAD model $\mathcal{M}$, and the 3D point cloud $\mathcal{P}$ representing the object. In order to get a symmetric definition, we first put the model in the annotated 3D box, scale it to fit, render $\mathcal{M}$ with the camera parameter of the depth image, and convert the rendered depth image to a point cloud $\mathcal{V}$. This is to mimic the partial view due to self occlusion. Then, we define the distance between $\mathcal{P}$ and $\mathcal{S}$ as:

$$D(\mathcal{P},\mathcal{S}) = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\Big(\min_{q\in\mathcal{V}} d(p,q)\Big) + \frac{1}{|\mathcal{V}|}\sum_{p\in\mathcal{V}}\Big(\min_{q\in\mathcal{P}} d(p,q)\Big),$$

where $d(p,q)$ is the distance between two 3D points p and q. After acquiring a short list of similar CAD models for each object, we randomly choose one and render the depth image with the original annotation as training data.

We generate a hybrid training set that is 1,000 times bigger than the original RGB-D training set. For both of the pathways in the 3D context network, we have to train the models on this large hybrid dataset first, followed by finetuning on the real depth maps. Otherwise, the training cannot converge.

6. Experiments

We use the SUN RGB-D dataset [32] because they provide high quality 3D bounding box annotations of objects. As described in Section 3.1, we manually select images that can be perfectly represented by one of the scene templates, and choose 1,863 RGB-D images from SUN RGB-D. We use 1,502 depth images to learn scene templates and train the 3D scene parsing network, and the remaining 361 images for testing. We also evaluate our model for object detection on a testing set containing images that cannot be per-

fectly represented, e.g. containing arbitrary objects or multiple scene templates, to demonstrate that our scene templates have a good generalization capability and a high impact on real scenes in the wild.

Our model uses the half data type, which represents a floating point number by 2 bytes, to reduce the memory usage. We train the model with a mini-batch of 24 depth images requiring 10GB, which nearly fills a typical 12GB GPU. However, this mini-batch size was too small to obtain reliable gradients for optimization. Therefore, we accumulate the gradients over four iterations of forward and backward without weight update, and only update the weights once afterwards. Using this approach, the effective mini-batch size is 24×4 or 96.

6.1. 3D object detection.

Our model recognizes major objects in a scene, which can be evaluated by 3D object detection. Qualitative parsing results are shown in Fig. 8. Our model finds most of the objects correctly and produces decent scene parsing results for challenging cases, e.g. heavy occlusion and missing depth. 3D context enables long range regression when initial alignment is far from correct, as shown in the 5th row. The last row shows a failure case, where our model recognizes it as a sleeping area misled by the futon with blankets. Therefore, our model overlooks the coffee table, but still predicts the wall and floor correctly and find a proper place to sleep.

Table 1 shows quantitative comparison to the local appearance based 3D object detector Deep Sliding Shape (DSS) [34] and also the cloud of gradient feature based context model from Ren *et al*. (COG) [31]. Our average precision (3rd row) is comparable to state-of-the-art, but only takes about 0.5 seconds to process an image for all object categories, which is about 40 times faster than DSS which takes 20 seconds per image.

Context complements local evidence. Fig. 6 shows some qualitative comparisons between our context model and the local object detector DSS [34]. We can see that our context model works significantly better in detecting objects with missing depth (the monitor in 1st and 3rd examples) and heavy occlusion (the nightstand in 2nd example). 3D context also helps to remove objects in incorrect arrangements, such as the table on top of another table, and the nightstand at the tail of the bed or in office, as shown in the result of DSS. Comparatively, DSS works better for objects that are not constrained, e.g. chairs on the right of 3rd example.

We integrate the result from DSS and our context model. The combined result achieves significantly better performance than each of the models individually, increasing the mean average precision from the 43.76% for DSS standalone to 50.50%. This significant improvement demonstrates that our context model provides complementary information with a local appearance based object detector.

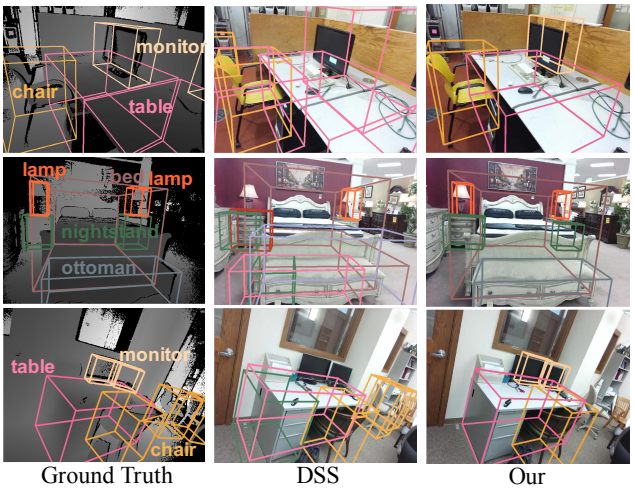

Figure 6. **Comparison between our context model and the local object detector DSS [34].** Our context model works well for objects with missing depth (monitors in 1st, 3rd row), heavy occlusion (nightstand in 2nd row), and prevents detections with wrong arrangement (wrong table and nightstand in DSS result).

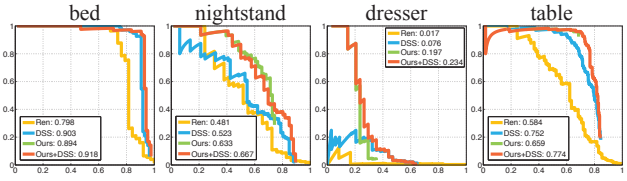

Figure 7. **Precision recall curves for some object categories.** We compare our algorithm with the 3D object detector DSS [34] and the cloud of gradient feature based context model Ren *et al*. [31].

Fig. 7 shows the Precision-Recall (PR) curves for some of the object categories. We can clearly see that our (green) recalls are not as high as DSS (blue) that runs in a sliding window fashion to exhaustively cover the search space. This is because our model only detects objects within the context. However, our algorithm maintains a very high precision, which applies to a broader range of working situations, with slightly lower recall. Nevertheless, combining the result of our method and DSS (red) obtains the best performance in terms of both precision and recall.

Generalization to imperfect scene template images. Our method can work not only on perfect scene template images, but also images in the wild. Thanks to the template classification and alignment component, our method can find the right place in the input scene to apply the context model. To evaluate, we randomly pick 2,000 images that are not used for training from the SUN-RGBD dataset. This uniformly sampled testing set reflects the scene distribution from the dataset, and contains many images that cannot be perfectly represented by any of the scene templates. We test DSS on this test set and achieve 26.80% mAP (Ta-

	bed	night-stand	dresser	coffee table	mirror dresser	end table	lamp	monitor	ottoman	sofa	chair	table
COG [31]	79.8	48.1	1.70	-	-	-	-	-	-	55.8	72.9	58.4
DSS [34]	90.3	52.3	7.60	52.7	4.40	13.3	40.2	15.0	23.7	71.3	79.1	75.2
Ours	89.4	63.3	19.7	40.5	16.8	27.9	41.6	18.2	13.3	50.3	44.5	65.9
Ours + DSS	**91.8**	**66.7**	**23.4**	50.1	**10.0**	**35.3**	**53.6**	**23.2**	**31.5**	62.8	**80.2**	**77.4**
GT Align	92.4	64.4	19.7	49.3	23.4	25.0	31.4	16.0	15.8	63.6	46.1	70.4
GT Align+Scene	94.1	66.3	19.4	48.9	23.4	21.7	31.4	16.1	15.8	74.6	50.2	74.0
DSS, Full	75.7	30.0	7.14	19.5	0.64	11.7	20.9	1.80	8.49	51.7	52.9	41.1
Ours, Full	**75.8**	**44.1**	**15.7**	**25.8**	**4.99**	**12.4**	**22.4**	**3.47**	**10.7**	49.0	**53.2**	30.5

Table 1. **Average precision for 3D object detection.** We (row 3) achieve comparable performance with DSS [34] (row 2). Combining two methods (row 4) achieves significantly better performance, which shows our model learns context complementary to local appearance. Our model can further achieve better performance with better alignment and scene classification. The last row shows our superior performance on extended testing set where images might not be perfectly represented by any single scene template.

Layout Estimation (Mean/Median)	Sleeping Area	Office Area	Lounging Area	Table &Chair
Ceiling Initial	0.57/0.56	-	-	0.84/0.71
Ceiling Estimate	0.45/0.40	-	-	0.72/0.44
Floor Initial	0.30/0.25	0.28/0.24	0.25/0.23	0.22/0.20
Floor Estimate	0.10/0.09	0.09/0.06	0.22/0.16	0.08/0.05
Wall Initial	0.40/0.30	0.70/0.60	-	-
Wall Estimate	0.22/0.08	0.60/0.21	-	-

Table 2. **Error (in meter) for room layout estimation.** Our network reduces the layout error upon initialization from the transformation network. Note that for some scene categories, the ceiling and wall may not be visible from the images and therefore there are no annotations (marked with "-").

Method	Sym.	Sleeping Area	Office Area	Lounging Area	Table &Chair
ICP	No	75.6%	69.2%	58.5%	38.1%
ICP	Yes	96.3%	89.0%	92.5%	75.3%
Network	No	92.7%	87.9%	71.7%	44.3%
Network	Yes	100.0%	93.4%	94.3%	73.2%

(a) Rotation Estimation Accuracy↑

Method	Rot.	Sleeping Area	Office Area	Lounging Area	Table &Chair
ICP	-	0.473	0.627	1.019	0.558
Network	GT	0.278	0.246	0.336	0.346
Network	Est	0.306	0.278	0.606	0.332

(b) Translation Error (in meters) ↓

Table 3. **Evaluation of the transformation networks.** Our transformation network outperforms direct point cloud matching in the accuracies of both rotation and translation.

ble 1, the 2nd last row), which is similar to the performance reported in [34]. We further run our method on testing images with the template classification confidence higher than 0.95, which ends up choosing 1,260 images. We combine our result with DSS, and the performance is shown in the last row of Table 1. As can be seen, our model successfully wins in 10 out of 12 categories, and improves the mAP to 29.00%. This improvement shows that our model can be applied to a variety of indoor scenes. It is also extremely effective in improving the scene understanding result in the aligned sub-area.

6.2. Room Layout and Total Scene Understanding

Layout estimation. As part of our model, we can estimate the existence and location of the ceiling, floor, and the wall directly behind the camera view. Table 2 shows quantitative evaluation. We can see that the 3D context network can successfully reduce the error and predict a more accurate room layout. Note that for some scene categories, the ceiling and wall are usually not visible from the images. These cases are marked as "-".

Scene understanding. We use the metrics proposed in [32] to evaluate total 3D Scene Understanding accuracy. These metrics favor algorithms producing correct detections for all categories and accurate estimation of the free space. We compare our model with Ren *et al.* (COG) [31]. For geometry precision (P_g), geometry recall (R_g), and semantic recall (R_r), we achieve 71.02%, 54.43%, and 52.96%, which all clearly outperform 66.93%, 50.59%, and 47.99% from COG. Note that our algorithm uses only the depth map as input, while COG uses both color and depth.

6.3. System Component Analysis

Our 3D context network relies on the initial alignment produced by scene template classification and transformation estimation model. We also investigate how these factors affect our performance.

Transformation Prediction. Table 3 reports the evaluation of template alignment. For rotation, we show the percentage of data within a 10 degree range to the ground truth. For translation, we show the distance between the estimated translation and the ground truth.

For rotation, since some scenes (especially for lounging area and table&chair set) are symmetric with respect to the horizontal plane, a correct estimation of the main direction would be enough for our purposes. Therefore, we report the accuracies both with and without symmetry [Sym.].

To compare with our neural network-based approach, we design an ICP approach based on point cloud alignment as a baseline. Given a point cloud from a testing depth map, we align it with the point cloud of each image in the training set,

Figure 8. **Visualization of the qualitative results on the testset.**

by exhaustively searching for the best rotation and translation, using the measurement in Section 5. We choose the alignment with the best aligned training depth map as our transformation. We can see that our neural network based approach significantly outperforms this baseline.

To see how sensitive our model is to the initial alignment, we evaluate our model with the ground truth alignment, and the result is shown in Table 1 [GT Align]. We can see that the 9 out of 12 categories are improved in terms of AP, compared to that with estimated transformation, and the overall mAP improves 2.19%.

Template Classification. The accuracy of the scene template classification is 89.5%. In addition to the ground truth transformation, we test our model with truth template category. This further improves the mAP by 1.52%.

7. Conclusion

We propose a 3D ConvNet architecture that directly encodes context and local evidence leveraging scene template. The template is learned from training data to represent the functional area with relatively strong context evidence. We show that context model provides complementary information with a local object detector, which can be easily integrate. Our system has a fairly high coverage on real datasets, and achieves the state of the art performance for 3D object detection on the SUN-RGBD dataset.

References

[1] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 2013. 2

[2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. In *arXiv*, 2015. 5

[3] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010. 2

[4] M. J. Choi, A. Torralba, and A. S. Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 2012. 2

[5] M. J. Choi, A. Torralba, and A. S. Willsky. A tree-based context model for object recognition. *PAMI*, 2012. 2

[6] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3D geometric phrases. In *CVPR*, 2013. 1, 2

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[8] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *IJCV*, 2011. 2

[9] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 5

[10] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *ICCV*, 2005. 2

[11] F. Han and S.-C. Zhu. Bottom-up/top-down image parsing with attribute grammar. *PAMI*, 2009. 2

[12] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. *arXiv*, 2015. 3

[13] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, 2008. 2

[14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 3

[15] H. Jiang and J. Xiao. A linear approach to matching cuboids in RGBD images. In *CVPR*, 2013. 2

[16] T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, and V. Mansinghka. Picture: A probabilistic programming language for scene perception. In *CVPR*, 2015. 3

[17] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015. 3

[18] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*. 2010. 2

[19] C. Li, A. Kowdle, A. Saxena, and T. Chen. Towards holistic scene understanding: Feedback enabled cascaded classification models. *PAMI*, 2012. 2

[20] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *CVPR*, 2009. 2

[21] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010. 2

[22] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via cnn image purification. *ACM Transactions on Graphics (TOG)*, 2015. 3

[23] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3D object detection with rgbd cameras. In *ICCV*, 2013. 1, 2

[24] T. Liu, S. Chaudhuri, V. G. Kim, Q. Huang, N. J. Mitra, and T. Funkhouser. Creating consistent scene graphs using a probabilistic grammar. *ACM Transactions on Graphics (TOG)*, 33(6):211, 2014. 1, 2

[25] V. K. Mansinghka, T. D. Kulkarni, Y. N. Perov, and J. B. Tenenbaum. Approximate bayesian image interpretation using generative probabilistic graphics programs. In *NIPS*, 2013. 2

[26] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. 2015. 3

[27] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014. 2

[28] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 4

[29] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 2

[30] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 2

[31] Z. Ren and E. B. Sudderth. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *CVPR*, 2016. 2, 6, 7

[32] S. Song, S. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 1, 5, 7

[33] S. Song and J. Xiao. Sliding Shapes for 3D object detection in RGB-D images. In *ECCV*, 2014. 3

[34] S. Song and J. Xiao. Deep Sliding Shapes for amodal 3D object detection in RGB-D images. In *CVPR*, 2016. 3, 4, 5, 6, 7

[35] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2015. 3

[36] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *NIPS*, 2005. 2

[37] E. B. Sudderth and M. I. Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In *NIPS*, 2008. 2

[38] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, 2005. 2

[39] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Depth from familiar objects: A hierarchical model for 3D scenes. In *CVPR*, 2006. 2

[40] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky. Describing visual scenes using transformed objects and parts. *IJCV*, 2008. 2

[41] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 2011. 2

[42] Z. Tu. Auto-context and its application to high-level vision tasks. In *CVPR*, 2008. 2

[43] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 2005. 2

[44] T. Wu and S.-C. Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *IJCV*, 2011. 2

[45] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3D ShapeNets for 2.5D object recognition and Next-Best-View prediction. *ArXiv e-prints*, 2014. 3, 5

[46] L. F. Yu, S. K. Yeung, C. K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher. Make it home: automatic optimization of furniture arrangement. 2011. 2

[47] Y. Zhang, M. Bai, P. Kohli, S. Izadi, and J. Xiao. Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding. *arxiv*, 2016. 3

[48] Y. Zhang, S. Song, P. Tan, and J. Xiao. PanoContext: A whole-room 3D context model for panoramic scene understanding. In *ECCV*, 2014. 1, 2

[49] Y. Zhao and S.-C. Zhu. Integrating function, geometry, appearance for scene parsing. 1, 2