Object-Driven Multi-Layer Scene Decomposition From a Single Image

Helisa Dhamo [1] helisa.dhamo@tum.de
Nassir Navab [1] nassir.navab@tum.de
Federico Tombari [1,2] tombari@in.tum.de

[1] Technische Universität München [2] Google

Abstract

We present a method that tackles the challenge of predicting color and depth behind the visible content of an image. Our approach aims at building up a Layered Depth Image (LDI) from a single RGB input, which is an efficient representation that arranges the scene in layers, including originally occluded regions. Unlike previous work, we enable an adaptive scheme for the number of layers and incorporate semantic encoding for better hallucination of partly occluded objects. Additionally, our approach is object-driven, which especially boosts the accuracy for the occluded intermediate objects. The framework consists of two steps. First, we individually complete each object in terms of color and depth, while estimating the scene layout. Second, we rebuild the scene based on the regressed layers and enforce the recomposed image to resemble the structure of the original input. The learned representation enables various applications, such as 3D photography and diminished reality, all from a single RGB image.

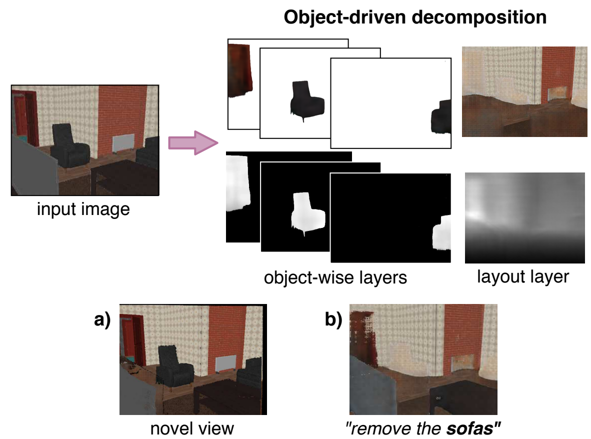

Figure 1. **Overview of our method and its applications.** *(Top):* Given a single color image, we infer a layered representation that consists of a set of RGB and depth images for every object in the scene, as well as the scene layout. *(Bottom):* Illustration of two applications, **a)** view synthesis and **b)** object removal.

1. Introduction

Completing a scene beyond the partial occlusion of its components is a strongly desired property for many computer vision applications. For instance, in robotic manipulation, the ability to see the full target object despite the presence of occluding elements can lead to a more successful and precise grasping. In the autonomous driving context the estimation of the full profile and location of potential obstacles occluded by the car in front of us would prove useful to increase the robustness of the trajectory planning and safety control.

Scene completion beyond occlusion is important not just to improve higher-level perception systems, but also to enhance the fruition of captured visual data. 3D photography uses image content behind occlusion to enhance the user experience while looking at a photo by synthesizing novel unseen views. When changing the vantage point the picture was originally taken from, the visual content around object borders gets dis-occluded, thus enabling the image fruition to become more immersive. The combination of 3D photography with a virtual reality (VR) display, such as a Head Mounted Display (HMD), holds the potential to generate a visually effective application.

Layered Depth Image (LDI), pioneered by Shade *et al.* [35], is a data representation distinctively suitable for the aforementioned applications. To augment a single view into a 3D photo, a single depth image is not enough, since it is not designed to store visual and geometric information beyond that of the visible object parts in the scene. On the other hand, having a fully completed 3D model of the scene is often an unnecessary complication, since most of the information present in such model would never be used if the novel vantage points are either nearby the original one and/or small in number. It is worth noting that generating such completed 3D scenes typically comes with high computational and memory cost [51, 11, 37, 44].

Therefore, a layered structure of the original view represents an interesting trade-off between simplicity in the representation and capacity of storing all necessary information for these application. Recent works generate such a data representation either from multi-view [14], or stereo [50] input, with some variations in the representation. More adventurous approaches [6, 41] aim to learn an LDI from a single color image. The motivation is providing a method that does not rely on the availability of appropriate photo pairs/sets, so that consumers can reconstruct a 3D photo out of any image, casually at hand. Both Dhamo *et al*. [6] and Tulsiani *et al*. [41] report results with LDIs having two layers only (background and foreground).

Intuitively, we are able to guess the complete appearance of the partially occluded objects we see, using the color information from the visible parts, together with some object specific characteristics. Here, we motivate a similar feature learning process in our proposed framework. Given the accessibility of state-of-the-art object detectors *e.g*. Mask R-CNN [12], we assume that predicted instance masks (partial visibility map) and class categories are available. The proposed approach is based on object-wise RGB-D completion followed by a re-composition branch which we call *minimum depth pooling*, that inspects the reconstruction of the original image from the layered representation. This improves the depth prediction accuracy, in that it aligns the visible parts to obtain a global consistency. The proposed method uses the predicted mask probabilities as an attention guiding for the appearance of every object, while the class category predictions aim to induce priors for the object completion.

Our work aims to bridge current limitations in LDI prediction from a single image, and relies on four main contributions. First, we propose a flexible extension for a multiple layer representation, where unlike [41], the number of layers is not pre-defined in the form of CNN architecture branches. Second, we leverage predictions of semantic identities to perform object-oriented completions, which are not considered in the generation procedure of previous methods [6, 41]. Third, we propose a re-composition loss that is specific to our task. As a fourth contribution, we generate two datasets suitable for learning layered representations, which we will publicly release for further research.

We show that our results outperform state-of-the-art methods in LDI prediction and view synthesis. In addition, along with view synthesis, our object-level separation enables a new application, the removal of particular target objects, in a diminished reality scenario, Fig. 1.

2. Related Work

Recent developments in computer vision, extensively target the inference of 3D content from monocular 2D images, either in 2.5D (depth map, normal), object/scene 3D models and layered representations.

Depth prediction from single view is widely tackled with CNNs, excluding the first few works that consisted of hand-engineered features [33, 34] and data-driven methods [22, 20]. Eigen et al. [8] propose a multi-scale CNN architecture. Roy and Todorovic [32] propose regression forests with a shallow CNN on each node. Deeper fully convolutional architectures [23, 21] were later proposed based on ResNet [13] and DenseNet [15] respectively. Commonly, Conditional Random Fields (CRFs) [24, 30, 43, 47, 29] are used to enforce geometrical constraints. Other works exploit semantics to further boost depth performance [26, 18].

CNNs have been also applied to 3D inference from a single color image. A family of methods restrict the output to a single object 3D model [4, 9, 46, 45]. In contrast, Tulsiani *et al*. [40] infer a factored 3D scene model composed of a layout and a set of object shapes. None of these methods predicts texture behind occlusion, which is subject of our approach. Other methods exploit more extended inputs to predict 3D scene representations, such as a panorama image [51], RGB-D [11] or a depth map [37, 44].

Layered scene representations come in a diversity of contexts, such as depth ordering of semantic maps [48, 39, 16] and color images [7], motion analysis and optical flow [42, 38], stereo reconstruction [3], scene decomposition in depth surfaces [28] and planes [27]. Our focus is on the Layered Depth Images (LDI) introduced by Shade *et al*. [35], which refer to a single view representation of a scene that contains multiple layers of RGB-D information. This can be used for efficient image-based rendering on view perturbations, to deal with information holes on dis-occlusion. Hedman *et al*. [14] use such a representation for 3D photo capturing from multi-view inputs. Zhou *et al*. [50] infer a similar representation from stereo input, that decomposes an image into sweep planes with fixed depth. Very recent works, [6, 41] propose LDI prediction from a single RGB image, which has the practical advantage of enabling the 3D enhancement of any photo, even if additional views or depth maps are not available. Dhamo *et al*. [6] automatically generate ground truth data from large-scale datasets that contain trajectory poses, by warping multiple frames to a target view, to populate the second layer of the target LDI. Then, LDIs are learned in a supervised way, formulated as depth prediction and RGB-D inpainting. Tulsiani *et al*. [41] instead, overpass the data limitations by proposing a view synthesis supervision, where the LDIs are learned in a self-supervised way.

3. Method

In this Section we detail the proposed multi-layer scene decomposition method. Section 3.1 presents the rendering procedure we employ to generate ground truth data. Then, the following Sections describe the learning model, which consists of three components: object completion, layout prediction, and image re-composition. These three stages are reported in Fig 2, that sketches the overall architec-

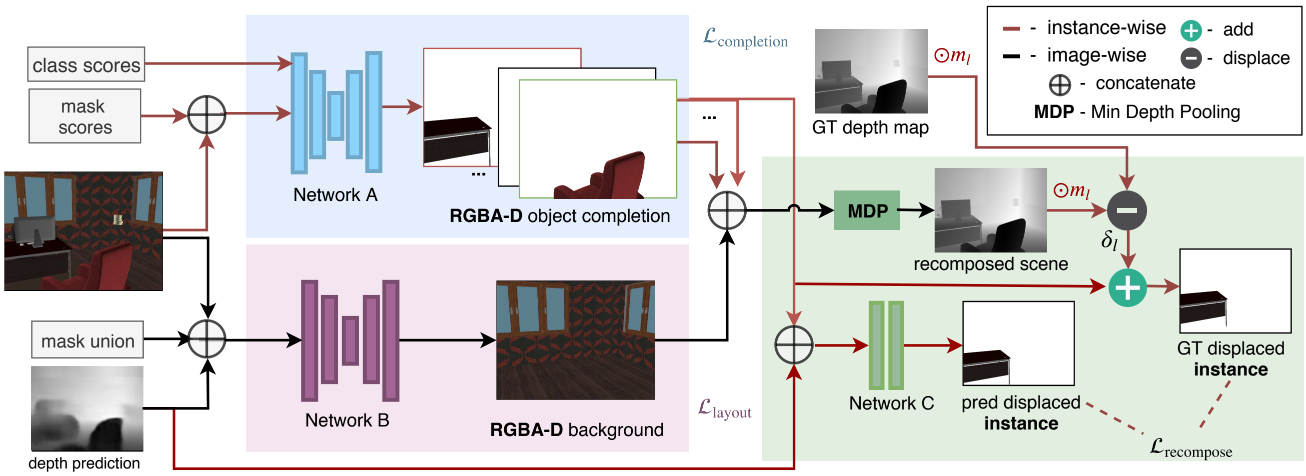

Figure 2. **Proposed scene layering framework.** *Left:* While *Network A* (*top*) completes the occluded parts for each detected object instance, to a RGBA-D representation, *Network B* (*bottom*) predicts RGBA and depth images for the empty scene. *Right:* The outputs are concatenated and fed to the Minimum Depth Pooling (MDP) layer, that recomposes the scene. Instance-wise, the displacement of the recomposed first layer depth from the ground truth depth is used in the re-composition loss to supervise *Network C* and give the final result.

ture of our model. Since scenes come with varying levels of complexity, assuming a fixed, pre-determined number of layers to represent them tends to limit the flexibility and, as such, the performance of image decomposition approaches. We introduce an adaptive model where the number of layers is dependent on the number of the detected object instances in the current scene.

3.1. Data generation

LDI prediction from a single image is quite a novel task, therefore large-scale datasets suitable for deep learning are not available. To overpass this limitation, one could either formulate an indirect supervision [41], or investigate ways to generate ground truth layered image representations [6]. In this work we explore the latter, to investigate the potential advantage of a rich supervision. Our goal of object-oriented layer inference implies the need for additional ground truth, such as RGBA-D representations of every object and layout of the scene, which we acquire automatically from existing datasets. Unlike Dhamo *et al*. [6], we employ a mesh-based rendering approach. The advantage is that the 3D mesh captures all the available information in the scene, while an image-based approach [6] only captures information which is present in the set of consecutive image frames to be warped. For every frame, we render the visible instances separately, similarly to [7]. In addition to the color images and the visibility masks, we extract depth maps and object categories for every instance. For this purpose, we utilize instance annotations, which are available in the 3D meshes, to separately extract the vertices that belong to each visible object in every view frame. Structural elements, identified by their semantic category (floor, walls, ceiling, window) are grouped together in the layout layer. We make sure that instances that were not originally visible in a certain view, are not included in its layered representation. For example, we do not want the object from another room (behind the enclosing wall of the currently visible room) to form part in the compositional layers of that view. The advantage of the proposed semantic-aware rendering with respect to [41, 6] is that it enables learning of class specific features, which might turn helpful in regressing plausible objects in the novel LDI layers.

Here, we work with two different datasets, namely SunCG [37] and Stanford 2D-3D [1]. Both datasets contain scene meshes together with 2D modalities (color, depth, instances, semantics). The latter suffers from a typical real-world mesh nuisance, namely the presence of holes and missing surface parts, which is critical in our task as it leads to incomplete object renderings behind occlusion. However, we observed in Stanford 2D-3D [1] fewer such holes compared to other state-of-the-art large-scale real datasets. Through a post-processing step, we select a subset of the rendered layered images, *i.e*. all those where the amount of overlap between layers is beyond a threshold, to ensure that there is enough novel information on dis-occlusion. We use SunCG [37] for a fair comparison and ablation, given pixel-perfect ground truth, while Standford 2D-3D [1] demonstrates applicability in a real world setup.

The generated object and layout layers can be easily arranged in an LDI representation, using the depth maps to sort the layers at every pixel location.

3.2. Object Completion

The goal of the object completion branch (Network A, Fig. 2) is to learn a mapping from an occluded object x_c to the RGB-D representation of it in full visibility y. We use a helping mask for each object, as an intuitive prior for ambiguous problems [9, 7, 6]. In addition, we incorporate semantic classes so to encourage the model to learn class specific properties. Therefore, Network A receives the input RGB image, the predicted mask and class scores of an object, and predicts a five-channel output – *i.e.*, the completed RGBA-D representation of that object. The algorithm is applied instance-wise, for each available mask. The architecture details are provided in Section 4. For this task, we found it more adequate to feed full images in Network A instead of object-focused regions of interest (RoIs). Although RoI cropping is more efficient, it limits the generation to a fixed resolution, which mostly affects the texture details of big objects. In addition, it weakens depth perception, as it reduces global context and hinders the understanding for object scaling and extent.

During training, for each ground truth instance mask we determine the respective prediction from Mask R-CNN [12]. The match is defined as the highest intersection-over-union (IoU) between the predicted and ground truth masks. To avoid wrong correspondences, we discard matches that have an $IoU < 0.3$. Utilizing these valid ground truth - prediction pairs, we learn how to complete the occluded part of the objects in the image. We use an L1 loss on the RGBA-D predictions $\hat{y}$, which is weighted by a relevance map γ

$$\mathcal{L}_{\text{completion}} = ||\gamma(y - \hat{y})||_1. \tag{1}$$

The need of γ arises from the fact that the different regions in the image have different relevance for the object completion problem. We want to set a higher influence of the loss in pixels close to the object appearance. First, we set $\gamma = 0.7$ in the *visible area* of the object in the original image including *a close neighbourhood* (by applying a 31×31 dilation in the ground truth mask), $\gamma = 1.5$ in the *occluded regions* of that object, and $\gamma = 0.2$ otherwise. During inference, each mask and category prediction provided by Mask R-CNN [12] is fed to the object-completion network together with the input color image.

Object completion from a single image is a challenging problem, in that predicting the visible part only might represent already a relatively good solution and the occluded parts are not properly learned. The guiding masks are not pixel perfect, therefore the model has to learn to differentiate between the different object instances along the edges, concurrently with the original goal of learning plausible object completions. Therefore, as additional pre-training stage to our learning strategy, we aim to encode common object properties to help our object completion network generate plausible outputs. In this pre-training stage, we train Network A as an auto-encoder (*i.e.*, in an unsupervised way) on single RGBA-D object representations

$$\mathcal{L}_{\text{auto}} = ||x - \hat{x}||_1, \tag{2}$$

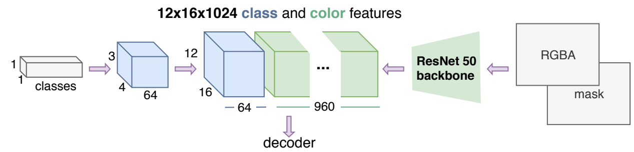

Figure 3. **Overview of the proposed object completion encoder architecture.** Class probabilities branch *(left)* and image branch *(right)* are concatenated along channels in the bottleneck layer.

where x is the ground truth RGBA-D map and $\hat{x}$ the auto-encoded output. For this unsupervised learning part, we use those object instances that do not touch the image borders to guarantee visibility on the whole object appearance. Then, we freeze the decoder (including the bottleneck layer) and train the encoder for object completion using the supervised strategy described in eq. (1). Intuitively, this encourages the occluded objects to share the same latent space as their respective RGBA-D representation, in full visibility.

3.3. Layout Prediction

The layout branch (Network B in Fig. 2) is designed to find a mapping between the input RGB image x_c and the corresponding RGB-D scene layout y, *i.e.* the object-free representation of the scene.

We employ a fully-convolutional network with skip connections as in [31], whose details are provided in Section 4. We observed considerable improvement on the depth layout prediction, when a standard depth map is regressed beforehand and provided as prior to the layout network. In contrast, such input hindered the optimization performance on the object branch, which we relate with the blurred and inaccurate object edges on such depth maps. Hence, our model receives an RGBA image x_c, a depth map $\hat{x}_d$ predicted via a CNN (in our experiments we use [23]), as well as the union of all the predicted instance masks used in Section 3.2. The mask union is used to give the model a hint on where the instance-free regions are located, so that it could exploit them to extrapolate layout features. Unlike the background inpainting in [6], we do not mask out the non-structural regions of the image. We wish to point out that, since the masks are simply predictions and not ground truth annotations, they are noisy and can mask out useful content from the image.

With the goal of predicting visually appealing layouts, we propose to carry out this task by means of an adversarial approach [10]. In addition, we want to encourage edge consistency in our generations, as the room contour is an interesting property of the layout. A similar motivation has been explored in [49], where occlusion boundaries are predicted as an intermediate step to improve a depth completion task. Unlike [49], instead of explicitly generating and supervising edges, we incorporate the perceptual loss $\mathcal{L}_p$ from [19]. Both the generated and ground truth layouts are fed in a

VGG-16 [36] network, pre-trained on ImageNet [5]. Then, in addition to the standard reconstruction loss $\mathcal{L}_r$, we want to exploit the $\mathcal{L}_1$ distance between the respective feature maps at a certain layer of the VGG-16 network. We choose to extract the features from the first VGG block, as we observed that it captures edges as desired. Then, the complete optimization problem becomes

$$\mathcal{L}_{\text{layout}} = \lambda_r \mathcal{L}_r + \lambda_p \mathcal{L}_p + \min_G \max_D (\mathcal{L}_{\text{a}}) \tag{3}$$

$$\mathcal{L}_{\text{r}} = ||y_c - \hat{y}_{\text{c}}||_1 + ||y_d - \hat{y}_d||_1 \tag{4}$$

$$\mathcal{L}_{\text{p}} = ||\phi(y_{\text{c}}) - \phi(\hat{y}_{\text{c}})||_1 \tag{5}$$

$$\mathcal{L}_{\text{a}} = \mathbb{E}_{x_c, y_c}[logD(x_c, y_c)] + \mathbb{E}_{x_c}[log(1 - D(x_c, G(x_c)))], \tag{6}$$

where y_c, y_d denote output color and depth respectively, and ϕ is the output feature map of the first VGG-16 block. Our layout prediction shares the definition and motivations of the one proposed in [40], however they differ in two aspects. First, we provide a background mask for attention guiding, and second, we regress an additional texture component.

A full ablation on the improvement of our design choices is provided as supplementary material.

3.4. Image Re-composition

An important requirement for our method is to regress layers that are correctly sorted in depth. This means, *e.g.* in the case of a scene with a foreground table in front of a background wall, that the distance between the viewpoint and the wall should not be smaller than the distance between the same viewpoint and the table. To implicitly and globally enforce the depth consistency of all regressed scene parts and layout, we propose an additional component in our model that we dub *minimum depth pooling* (MDP). This concatenates all the predicted layers (including objects and layout) and, for every pixel, extracts the RGBA-D from the layer with the lowest depth. The result of MDP is in the best case identical to the original input and the corresponding visible depth, together with an index map i_{map} that is the argmin of depth. This *image re-composition* strategy enforces the predicted multi-layer representation to coherently encode the structure of the original input image.

Since depth predictors learn a global understanding for depth, we use a standard predicted depth map $\hat{x}_d$ as a prior for our layer sorting problem. Foe each layer l, we only keep the region in $\hat{x}_d$ in which l is the front structure, using a binary mask m_l, which is one if $i_{map} = l$, and zero otherwise. We incorporate a re-composition block (*Network C*), that receives the predicted instance depth $\hat{y}_{d,l}$, concatenated with the masked $\hat{x}_d$ from [22]. Then we learn a set of depth displacements δ_l, as the distance between the mean ground truth $y_{d,l}$ and the mean predicted $\hat{y}_{d,l}$ layer depth (over m_l)

$$\delta_l = \frac{\sum m_l \odot y_{d,l}}{\sum m_l} - \frac{\sum m_l \odot \hat{y}_{d,l}}{\sum m_l}, \tag{7}$$

$$\mathcal{L}_{\text{recompose}} = ||y_{\delta,l} - \hat{y}_{d,l}||_1 \tag{8}$$

where $\odot$ is the element-wise multiplication and $y_{\delta,l} = \hat{y}_{d,l} + \delta_l$ is the displaced depth for instance l.

We show in experiments that the proposed MDP-driven re-composition loss improves not only the visible region of the objects, but also the occluded parts, given that it allows the whole layer to shift towards the right direction.

4. Implementation details

In this section, we provide a more detailed description regarding our architecture choices and implementation.

Network A The object completion network receives two inputs, the first one being an RGBA image concatenated with mask confidences, and the second a vector of class scores, both predicted from Mask R-CNN [12], Fig. 3. The images are fed into a ResNet-50 [13] backbone, with the original fully connected layer removed. We append one more convolution layer with $\text{out}_{\text{channels}} = 960$. The second path consists of two deconvolution layers of 64 channels applied consecutively on the class probabilities, which is a feature vector whose size equals the number of classes. Both branch outputs are concatenated along the channels, followed by *layer normalization* [2]. The network decoder consists of five up-projection layers from [23]. The architectures of the auto-encoder and the object completion network are identical, however, since the input modalities are partially different, we learn the encoder from scratch instead of fine-tuning the autoencoder weights.

Network B The layout generator has a U-Net [31] structure, similar to [17]. The generator G is an architecture with skip connections consisting of seven convolutions and deconvolutions, with a stride of two. The number of filters starts at 64 and is doubled after each convolution. Similarly, the deconvolutions halve the number of feature channels. Encoder and decoder outputs with the same resolution are concatenated. The discriminator D consists of 6 convolutions followed by a fully connected layer. Here, the output feature maps contain 64, 128, 256, 512, 512 and 512 channels. All layers in G and D are followed by batch normalization and leaky ReLU. The loss weights are $\lambda_r = 100$ and $\lambda_p = 25$.

Network C The re-composition block is composed of three 3×3 convolutions, each followed by ReLU.

Zero-padding around borders LDI representations are mostly intended for applications that involve a viewpoint change. As a side effect, the novel image contains empty regions on the borders, when the content was not visible in the original view. Therefore, we add zero padding to our input images before feeding them to the framework. Then, during inference, the the original view is spanned by predicting the originally padded surroundings. For the experiments of

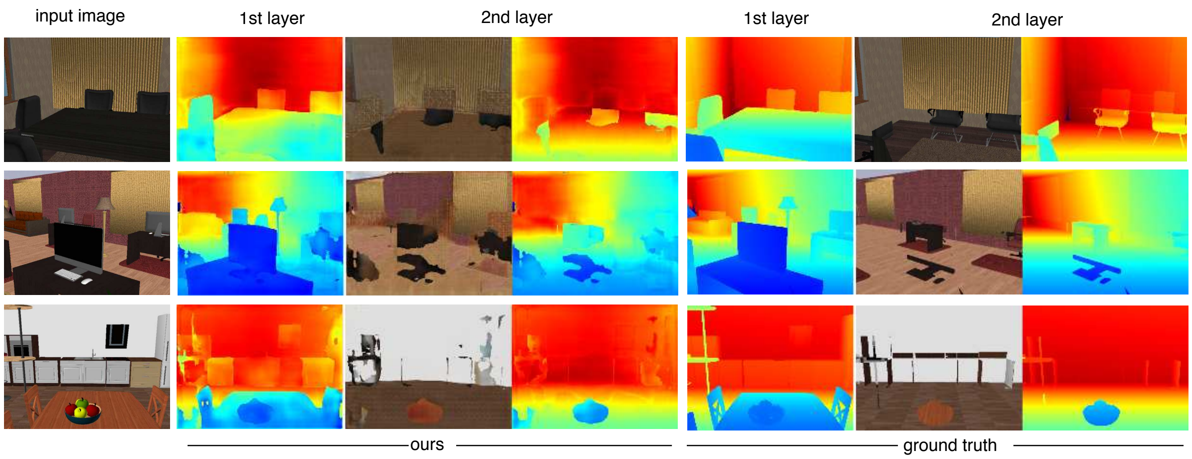

Figure 4. **LDI prediction results on SunCG.** *Left:* The input color image. *Center:* Our predictions for the first two layers, obtained after sorting the object-wise layers. *Right:* Ground truth, as extracted from the mesh-based rendering.

Method	1st layer depth		2nd layer depth		2nd layer RGB	
	MPE	RMSE	MPE	RMSE	MPE	RMSE
Tulsiani *et al.* [41]	1.174	1.687	1.582	2.873	72.70	91.51
Dhamo *et al.* [6]	0.511	0.832	1.139	1.848	48.57	76.98
Ours, baseline (w/o class scores)	0.551	0.879	0.687	1.120	43.97	66.51
+ class scores	0.508	0.793	0.700	1.090	44.50	65.70
+ $\mathcal{L}_p$	0.496	0.800	0.657	1.095	43.92	66.48
+ $\mathcal{L}_{\text{recompose}}$	**0.473**	**0.767**	**0.641**	**1.071**	**43.12**	**65.66**

Table 1. **Evaluation of LDI prediction on SunCG** for the first two layers of depth and the 2nd layer of RGB. We outperform the baselines. The errors are measured for color range $0-255$ and depth in meters.

this paper, we use padding bands of 16 pixels on the top and bottom and 12 pixels on the left and right borders.

5. Experiments

In this section, we present qualitative and quantitative evaluations of our method on two public benchmark datasets: SunCG [37] and Stanford 2D-3D [1]. We merge the output objects into a layered representation, in accordance with the original LDI idea [35] used in related works [6, 41]. In each pixel, the first layer represents the first visible point along the ray-line, the second layer relates to the next visible surface point and so on. The merging is done by an extended version of MDP, which sorts the depth of the object-wise layers, instead of simply returning the minimum. In these experiments, we use Mask R-CNN [12] predictions for the mask and class scores as input to our framework. For Stanford 2D-3D, we employ a network trained on the MS-COCO dataset [25]. For our experiments on SunCG, we finetune the network pre-trained on MS-COCO, using the NYU 40 class categories. As for the input depth predictions, we use the model from Laina *et al.* [23], respectively trained on SunCG and Stanford 2D-3D. We chose [12] and [23] as common baselines with available code. We also make this choice for fairness of comparison, since [6] also uses [23] to predict the first layer depth. However, our method is expected to work with various such models. We evaluate our results on two metrics, Mean Pixel Error (MPE) and Root Mean Square Error (RMSE). The measurements for each layer are done separately, since the difficulty is expected to depend on the layer index.

5.1. Layered representation

SunCG We compare the proposed method against state-of-the-art work in LDI prediction from a single image on the SunCG dataset. The comparison with Dhamo *et al.* [6] offers insights on the importance of assuming more than one level of occlusion in the scene. In contrast to their hard foreground/background separation, our representation supports more than one level of occlusion (Fig. 4, second row, desk). On the other end, the comparison with Tulsiani *et al.* [41] confirms the performance gain from a rich supervision. Since current work only evaluates on a two-layer LDI, we utilize our first two layers for the purpose of this experiment. Results on all layers are reported on the supplement. We use the same train and test splits in all these experiments, with 11k images on the training set and 2k on the test set. We report results in Table 1. We clearly outper-

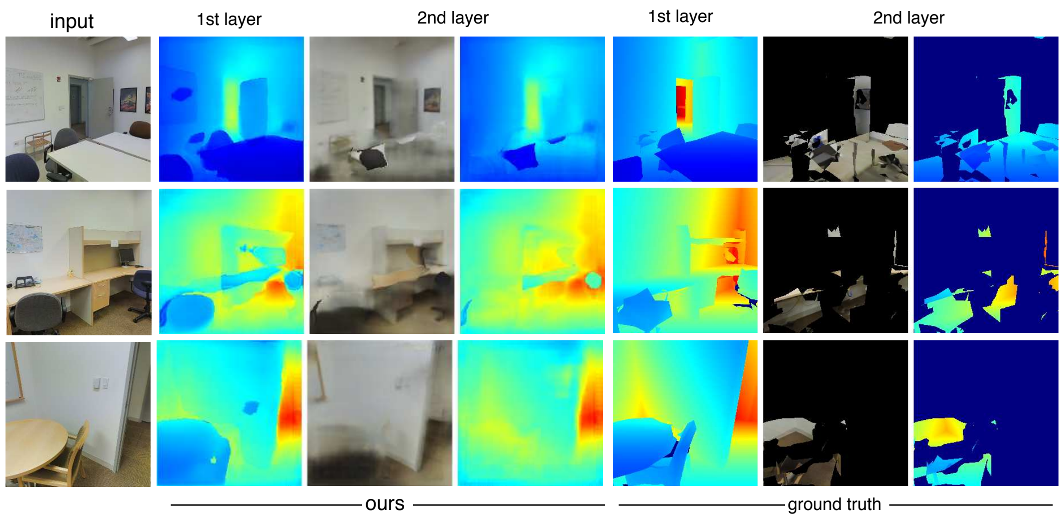

Figure 5. **LDI prediction results on Stanford 2D-3D.** *Left:* The input color image. *Center:* Our predictions for the first two layers, obtained after sorting the object-wise layers. *Right:* Ground truth, as extracted from the mesh-based rendering. Black in the color images and dark blue in the depth maps indicates information holes.

Method	1st layer depth		2nd layer depth		2nd layer RGB	
	MPE	RMSE	MPE	RMSE	MPE	RMSE
Tulsiani *et al.* [41]	0.805	1.088	0.954	1.230	57.42	72.65
Dhamo *et al.* [6]	**0.456**	**0.676**	0.830	1.193	42.92	55.87
Ours w/o $\mathcal{L}_{\text{recompose}}$	0.509	0.764	0.692	0.993	42.57	55.07
Ours	0.469	0.695	**0.688**	**0.987**	**42.45**	**54.92**

Table 2. **Evaluation of LDI prediction on Stanford 2D-3D.** LDI predictions for the first two layers of depth and 2nd layer of RGB. The errors are measured for color range $0-255$ and depth in meters.

Method	SSIM ↑	MPE ↓	RMSE ↓
Tulsiani *et al.* [41]	0.33	71.36	87.09
Dhamo *et al.* [6]	0.56	29.01	49.89
Ours	**0.65**	**18.19**	**34.71**

Table 3. **View synthesis on SunCG.** The synthesized **color** images are evaluated in terms of SSIM, MPE and RMSE, in range 0-255.

form [41] and [6] in all metrics, both for color and depth.

Additionally, we wish to point out a qualitative difference between our depth predictions and [41, 6]. Our method learns the depths instance-wise, therefore it overcomes the common problem of many CNN depth predictors represented by smeared object boundaries (more on the supplement). Sharp object edges are an attractive characteristic in view synthesis, as opposed to smooth edges, which in turn, lead to undesired loss of information during warping. Visual comparisons against these methods are provided in the supplementary material. Still referring to the results of Table 1, one can observe an improvement from adding the class category component, the perceptual loss $\mathcal{L}_p$ as well as our re-composition block, especially for depth.

Stanford 2D-3D Given the data limitations, we extracted 14k images with considerable ground truth coverage, from which 13k constructs the train set and 1k is kept for the test set. We follow one of the cross-validation splits suggested in [1] (area 1,2,3,4,6 vs. area 5a,b). We used the networks pre-trained on SunCG and fine tuned on the Stanford 2D-3D dataset. For this transfer, we convert the MS-COCO classes to NYU 40 to match the categories in our learned SunCG models. We had to disable the GAN loss for the Stanford 2D-3D training, since the network was predicting sparse layouts, trying to mimic a property of the real data. Table 2 reports the LDI prediction results. Also here, our method outperforms the baselines for the second layer prediction, which is the main focus of the works. On the first layer, Dhamo *et al*. results slightly superior, as our problem formulation is more sensitive to holes in the ground truth data and missed detections. However, the results in the synthetic domain encourage further improvement as more complete real datasets become available. In this experiment, we trained Tulsiani *et al.* on rendered images, as Stanford 2D-3D does not provide raw sequential camera trajectories with high overlap. To ensure that the rendering artifacts do not hinder the consistency between the views (needed for learning through view synthesis), we use rendered images for both the source and the target view (also on test time).

As shown in Fig. 5, the ground truth for the second layer

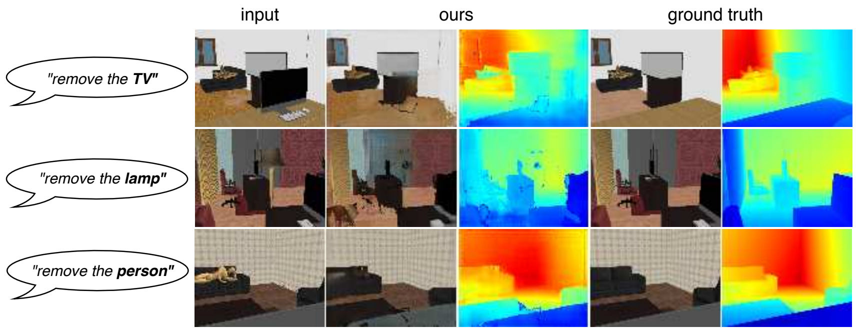

Figure 6. **Illustration of object removal results.** The category labels of the left indicate which object should be removed from the original image. We compare our predicted synthesized images (center) against the ground truth (right).

appears sparse, although as mentioned in Section 3.1 we automatically select images where information behind occlusion is available. We only consider the subset of available ground truth pixels for the error measurements.

5.2. View synthesis

Generating novel views is a direct application of a LDI representation. Therefore, we perform comparisons in this task. To generate data for this experiment, during the mesh-based rendering, we perturb the camera poses to obtain target frames. For the comparison, we use the same data splits as in the previous experiment. We utilize the layered representation learned from [41, 6] and our proposed method, and apply image-based rendering to synthesize the target views. We employ a simple rendering approach - basically, the first layer is warped first, while the following layers fill the holes left by the first rendering in a sequential manner. The resulting color images are then compared against the ground truth target views. Quantitative results are shown in Table 3. One can clearly observe a better performance of our method, such as alignment with the target, as witnessed by the more accurate color and depth in occluded regions. Fig. 7 illustrates how our method preserves the shapes of the front objects while rendering.

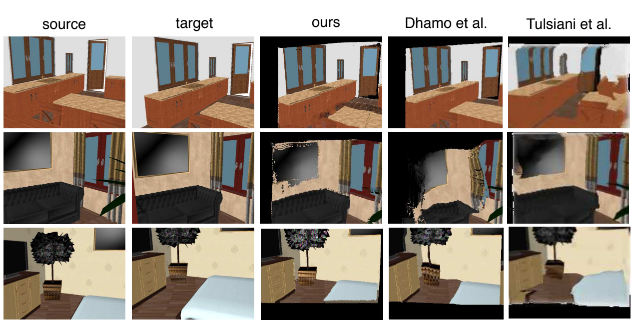

Figure 7. **View synthesis examples.** *Left:* Source image, *i.e.* the input to the proposed method as well as the target image, to be compared with the predictions. *Right:* Predicted novel views, using the LDIs from the proposed method, [6] and [41].

5.3. Object removal

Further, we illustrate the application of our method in diminished reality, *i.e.* removal of specific objects from the scene. We take the layered representation as regressed from our framework, where each layer consist of an object or layout. Then, we pass an object category, which we want to be removed from the original image. In this case, we assume fixed input commands of the form ”remove **class**”, which satisfies the scope of this work. However, combinations with more advanced natural language processing algorithms would be interesting to explore. Fig. 6 shows a few examples of this application. We observe that our method predicts plausible shapes for partly occluded objects, even when considerable fractions of them are missing.

6. Conclusions

We addressed the timely problem of inferring a layered representation of a scene, where only a single image is known. The proposed model enables a flexible number of output layers, *i.e.* adapts to the scene complexity. We have shown that the method outperforms previous works, especially targeting the occluded regions. Semantic information was incorporated, which has shown to improve the object completion performance. There is still a number of challenges that need to be addressed for further improvement, such as making the textures crisp and realistic, combining the advantages of global and local context with a view to efficiency, or exploiting spatial relations between objects.

Acknowledgement

This research has been supported by DFG funds under the project #381855581. We gratefully acknowledge NVIDIA with the donation of a Titan XP GPU used for this research. We would like to thank Nikolas Brasch, Fabian Manhardt and Manuel Nickel for the valuable suggestions and proofreading.

References

[1] Iro Armeni, Sasha Sax, Amir Roshan Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv:1702.01105*, 2017.

[2] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[3] Simon Baker, Richard Szeliski, and P. Anandan. A layered approach to stereo reconstruction. *CVPR*, 1998.

[4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kehui Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

[6] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 2019.

[7] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, 2018.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[9] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[11] Ruiqi Guo, Chuhang Zou, and Derek Hoiem. Predicting complete 3d models of indoor scenes. *arXiv:1504.02437*, 2015.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *ICCV*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[14] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3d photography. *ACM Trans. Graph.*, 2017.

[15] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[16] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. *ICCV*, 2013.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *ICCV*, 2017.

[18] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson W. H. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *ECCV*, 2018.

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[20] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from videos using nonparametric sampling. In *Dense Image Correspondences for Computer Vision*. 2016.

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017.

[22] Janusz Konrad, Meng Wang, and Prakash Ishwar. 2d-to-3d image conversion by learning depth from examples. In *CVPR Workshops*, 2012.

[23] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[24] Bo Li, Chunhua Shen, Yuchao Dai, A. van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[26] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. *CVPR*, 2010.

[27] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *CVPR*, 2019.

[28] Chen Liu, Pushmeet Kohli, and Yasutaka Fukurawa. Layered scene decomposition via the occlusion-crf. In *CVPR*, 2016.

[29] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. *CVPR*, 2015.

[30] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.

[32] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. *CVPR*, 2016.

[33] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *NIPS*, 2006.

[34] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 2009.

[35] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. SIGGRAPH '98, 1998.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[37] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *CVPR*, 2017.

[38] Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered segmentation and optical flow estimation over time. *CVPR*, 2012.

[39] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. *CVPR*, 2014.

[40] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018.

[41] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *ECCV*, 2018.

[42] John Y. A. Wang and Edward H. Adelson. Representing moving images with layers. *IEEE transactions on image processing*, 1994.

[43] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L. Yuille. Towards unified depth and semantic prediction from a single image. *CVPR*, 2015.

[44] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Adversarial semantic scene completion from a single depth image. *3DV*, 2018.

[45] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *NIPS*, 2017.

[46] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *NIPS*, 2016.

[47] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *CVPR*, 2017.

[48] Yi Yang, Sam Hallman, Deva Ramanan, and Charless C. Fowlkes. Layered object models for image segmentation. *PAMI*, 2012.

[49] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. *CVPR*, 2018.

[50] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph.*, 2018.

[51] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. *arXiv:1803.08999*, 2018.