Counterfactual Depth from a Single RGB Image

Theerasit Issaranon Chuhang Zou David Forsyth
University of Illinois at Urbana-Champaign
{issaran1, czou4, daf}@illinois.edu

Abstract

We describe a method that predicts, from a single RGB image, a depth map that describes the scene when a masked object is removed – we call this "counterfactual depth" that models hidden scene geometry together with the observations. Our method works for the same reason that scene completion works: the spatial structure of objects is simple. But we offer a much higher resolution representation of space than current scene completion methods, as we operate at pixel-level precision and do not rely on a voxel representation. Furthermore, we do not require RGBD inputs.

Our method uses a standard encoder-decoder architecture, and with a decoder modified to accept an object mask. We describe a small evaluation dataset that we have collected, which allows inference about what factors affect reconstruction most strongly. Using this dataset, we show that our depth predictions for masked objects are better than other baselines.

1. Introduction

People regularly reason about free space they cannot see. For example, you might reach to grasp a cup, and your fingers will fold around the back of the cup, confident that there is room. As another example, you might put a mug down on your desk behind the laptop, even though you cannot see there. While your model of this invisible space might not be precise, you have it and use it every day. When you do so, you are using "counterfactual depth" — the depth you would see if an object had been removed. This paper shows how to predict counterfactual depth from images.

This ability to "see behind" is reproduced in scene completion methods, which seek to complete voxel maps to account for the back of objects, and to infer invisible free space. But these methods produce limited resolution models of space, and require depth measurements to do so on another hand. Besides, stereo pairs provide less help to infer scene geometry behind objects, since the larger unknown depth region can't be fully observed by small changes in camera position. While there are excellent methods for inferring depth from a single image, the resulting depth maps represent only the free space to the nearest object.

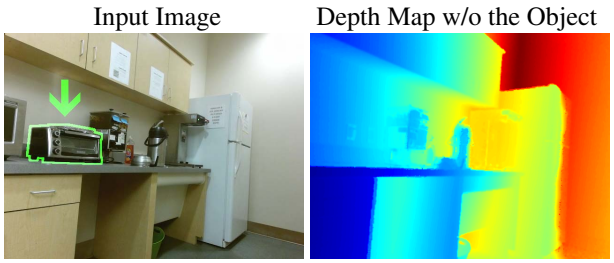

Figure 1. **Illustration.** Given an image of a scene (**left**), our goal is to predict the depth map for that scene *with the object removed* (**right**): *e.g.* the image depth without the microwave (outlined in green). Our system predicts depth directly from a single RGB image, offering a representation of the free space behind an object, even though it cannot see what lies there. These predictions are possible because indoor depth maps have quite strongly correlated spatial structure. Best viewed in color.

In this paper, we describe a system that can accept an image and an object mask, and produce a depth map for the scene where the masked object has been removed (Figure 1): e.g. if you mask a cup in an image of a cup on a table, our system will show you the depth behind the cup. Our method works for the same reason that scene completion works. Indoor scenes are very highly structured, and it is quite easy to come up with very good estimates of depth in unknown regions. However, image details are important: we show that our method easily outperforms Poisson smoothing of the depth map. Furthermore, our method easily outperforms the natural baseline of inpainting the image and recovering depth from the result, because inpainting often produces unnatural pixel fields.

Our approach is closely related to scene completion [41, 13], and works for the same reason that scene completion works. Scene geometries have quite simple spatially consistent structure. However, our method differs in important ways. We do not require additional depth information, and predict on RGB image only. Our system learns from images and depth maps (which are easy to acquire at a large scale), rather than from polyhedral 3D models of scenes.

Rather than actively reconstructing the entire scene at limited resolution (voxels), our method is *passive*: with no object mask, our method reports a depth map for the image; provided with a mask, it reconstructs the depth map of the image with that object removed. This deferred computation allows us to produce representations with smoothed output and much higher resolution than voxels can support. Our approach differs from the layered scene decomposition [26] and depth hole filling [1, 28] which all rely largely on the quality of input depth to perceive the hidden geometry.

Our **contributions**: 1) We describe a system that learns, from data, to reconstruct the depth that would be observed if an object or multiple objects were removed from a scene. 2) For images where an object is removed, quantitative evaluations demonstrate that our method outperforms strong natural baselines (depth hole filling, image inpainting and then depth prediction). 3) We introduce a carefully designed test set taken from real scenes that allows experiments investigating what scene and object properties tend to result in accurate reconstructions.

2. Related Work

Single image depth estimation is now well established. Early approaches use biased models (e.g. boxes for rooms [16]) or aggressive smoothing (e.g. [19]). Markov random field (MRF) [37] and Conditional random field (CRF) [30] can be applied to regress image depth against monocular images. More recent approaches use deep neural networks with multi-scale predictions [11, 12], large-scale datasets [25, 2] and user interactions [36]. Stereo provides strong cues for unsupervised learning [14, 45] or semi-supervised learning with LiDAR [23]. Other approaches use sparse depth samples [31] or variational models [20]. Laina *et al*. [24] propose a fully convolution approach with an encoder-decoder structure, and utilize per-pixel reverse Huber loss for better predictions. Chen *et al*. [9] propose to learn from pixel pairs of relative depth, which is further improved with supervisions of surface normal [10]. Our approach regresses on both depth and surface normal predictions. Different from Chen *et al*., we preprocess the ground truth surface normal with weighted quantized vectorization to ensure a smooth prediction. Moreover, we show in experiment that, in our task, angular-based surface normal loss can help improve performance (while Chen *et al*. found that this is less effective).

Depth completion helps predict the 3D layout of a scene and the objects in a novel view. The completion can be performed on point clouds [8], RGBD sensors [42, 38, 6, 44, 29], raw depth scans [34, 13, 41] or semantic segmentations [1]. The predictions can be represented as dense depth maps [44, 29, 6], 3D meshes [34, 8], or voxels [13, 41]. Our "conterfactual depth prediction" task is challenging, because we only condition on a single RGB input and a 2D object mask only, and predict the dense depth map of the scene with the object removed – we predict the depth that can be seen and the depth that we cannot see.

We also investigate the natural baseline of removing objects from the scene – **image inpainting**. We can apply existing single image depth estimation approaches on the inpainted images, and obtain the predicted depth map with the objects removed. Image inpainting can be achieved by smoothing from unmasked neighbors [35, 7, 4], patch-based approaches [5, 15], planar structure guidance [17] or convolution neural networks [18, 43, 27, 33]. We use the method by Iizuka *et al*. [18], which is one of the state-of-the-art for high resolution predictions with source code available, as our image inpainting baseline.

3. Approach

Assume a single RGB image I is given. Now, for *any* object mask M_{object} that identifies an object in the scene, write $\mathcal{M}$ for the set of pixels lying on the object. We would like to predict the depth for the scene *with that object removed* (Figure 2). We write d for the depth field; d_{behind} for the depth predicted for pixels in $\mathcal{M}$ (i.e. the depth behind the object in the mask); and d_{observe} for the depth predicted for pixels out of $\mathcal{M}$. For example, if the scene had a cup on a desk, and the mask lay on the cup, then d_{behind} would be the desk behind the cup, d_{observe} would be the rest of the desk, and d_{behind} should be predictable because of the spatial coherence of objects.

3.1. Network architecture

Figure 2 gives an overview of our network. We choose to modify the depth predictor by Laina *et al*. [24], because it is fully convolutional, and can model the dense spatial relationship between d_{behind} and d_{observe}. The encoder-decoder strategy of that method allows coarse-to-fine corrections of d_{behind}. Our network's input RGB image size is $228\times304\times3$ (height $\times$ width $\times$ dimension) and the output depth map is $128\times160\times1$. The encoder is based on Resnet-50, with the fully-connected layers and the top pooling layers removed. The bottleneck feature space is $8\times10\times1024$. The decoder consists four up-projection blocks and a 3×3 convolution layer afterwards. We use the object mask M_{object} to guide the prediction by concatenating M_{object} to each of the input feature layers of the up-projection block. M_{object} is 0 for pixels on the object to be removed and is otherwise 1 for non-removed area. The bottleneck forces the decoder to capture long scale order in depth fields; the mask then informs the decoder where it should ignore image features and extrapolate depth. Extrapolation is helped by having an image feature encoded, because the features give some information about the likely depth behavior at the boundary of the mask, so the decoder can extrapolate into the masked region using both depth prior statistics and feature infor-

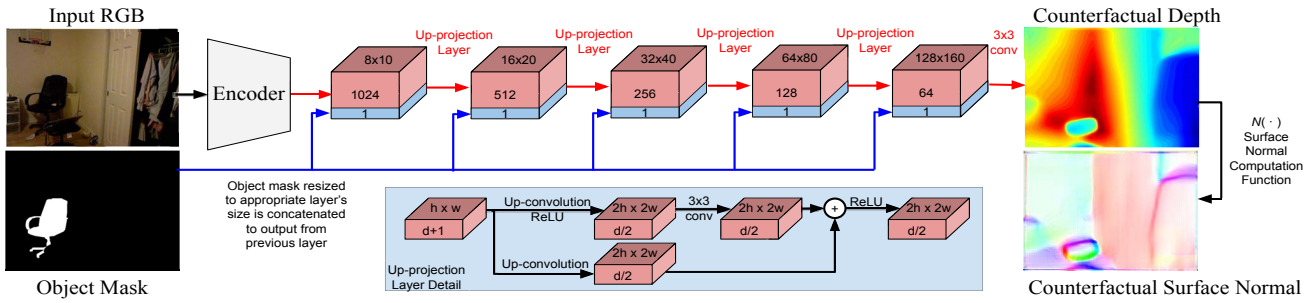

Figure 2. **Network architecture.** Our network takes as input a single RGB image and a 2D object mask. The network follows an encoder and decoder strategy. The final output is the predicted depth of the scene with the object removed: we predict the depth of layouts behind the chair, and the depth of other non-removed objects, e.g. the small table in front of the chair. We also show the surface normal derived from our predicted depth for better illustration. Best viewed in color.

mation to guide the extrapolation. This comes at the cost of training difficulty. The decoder has a strictly more difficult task than Laina *et al.*'s decoder, because it must be willing to extrapolate into any masked region supplied at run time. We also experienced with concatenating the object mask with the input RGB image as input, but observed performance degrades.

3.2. Network loss

Given a predicted image depth $\hat{d}$, and a ground truth depth d, the overall network loss for each image I is:

$$L(d,\hat{d}) = w_1 L_{\text{surface}}(d,\hat{d}) + w_2 L_{\text{avg}}(d,\hat{d}) + w_3 \text{berHu}(d,\hat{d}) \quad (1)$$

$L(d,\hat{d})$ is the weighted summation of the surface normal loss L_{surface}, the average image depth difference L_{avg} and the pixel-wise reverse Huber (berHu) loss [32].

Surface normal loss with weighted smoothed ground truth. Much of the world is made of large polygons [8, 17], so that we can expect strong spatial correlations in surface normal. One can obtain small depth errors with large surface normal errors, which suggests controlling surface normal error directly. We use a loss that encourages normals derived from the predicted depth to be accurate:

$$L_{\text{surface}}(d,\hat{d}) = -\frac{\sum_{p\in I} c_p \log\left(N(d_p)\cdot N'(\hat{d_p})\right)}{Q} \quad (2)$$

L_{surface} penalizes the average pixel-wise negative log likelihood of the angular distance between the predicted surface normal and the ground truth. p denotes a pixel in I positioned at (x,y). Q denotes the total number of pixels in I, and c_p is the pixel-wise weight that we will explain later. $N'(\cdot)$ denotes the surface normal computation which is the first-order derivatives of predicted depth.

However, computing ground truth normals $N(\cdot)$ requires care. For two adjacent pixels with only a few millimeters apart, a small error in measurement can still produce a steep change in normal direction. We apply a window-based gradient smoothing method, given known camera focal length f_x and f_y in x and y dimension respectively, computing gradients $n_p = (n_{p_x}, n_{p_y}, n_{p_z})$ at pixel p based on the neighboring pixels: $n_{p_x} = f_x \frac{1}{8}\sum_i \frac{d(x+i,y)-d(x-i,y)}{2i}$, $i \in \{1,2,\ldots,8\}$. We compute n_{p_y} in the same way, set $n_{p_z} = 1$ and normalize n_p to unit 1.

We then smooth the normal spatially, using a procedure to retain sharp normal discontinuities. We quantize each ground truth normal into discrete bins. We divide the hemisphere of the normal space (assuming all pointing towards the viewpoint) into equally spanned bins of 16 latitudes and 4 azimuths. Then, we score the confidence of each bin belonging to the pixel's normal based on the weighted average angular distance to the pixel's 8×8 neighbors: $c_b = \frac{1}{64}\sum_q \left(\max\left(n_q \cdot n_b, 0\right)\right)^\beta$. q denotes a pixel in neighborhood, n_b denotes candidate bin b's normal. We set $\beta = 8$ to model a smooth decrease of the angle between two normal vectors going further apart. Finally, we assign the highest score to c_p and its normal to n_p. The advantage of the weighting strategy is that for a flat ground truth region, most of the processed ground truth normal will be in the same bin, so we will recover a constant plane. Similarly, at a normal discontinuity (e.g. a ridge), one normal will dominate on one side and the other will dominate on the other, so the ridge will not be smoothed (see Figure 3). We show in experiments (Sec. 5.2) that training with L_{surface} helps boost our performance. It's worth noting that our approach is faster than plane fitting [39], and is more accurate than simple partial derivatives (please find more detailed comparison in supplemental material). This is crucial since we need to re-compute surface normal for each training sample as required by the data augmentation in Sec. 3.3.

Depth prediction loss. We penalize the average ℓ_2 depth difference compared to the ground truth: $L_{\text{avg}} = \left(\frac{\sum_p d_p - \sum_p \hat{d_P}}{Q}\right)^2$. We use reverse Huber loss $\text{berHu}(d,\hat{d})$ to penalize the per-pixel prediction error, which has shown

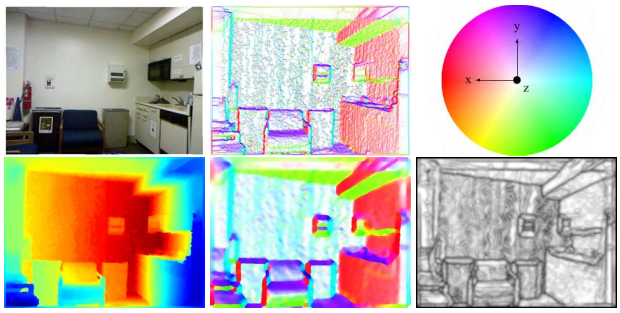

Figure 3. **Surface normal derived from depth v.s. our weighted quantized smoothed normal**. We show: RGB image (top left) and the ground truth depth (bottom left), ground truth surface normal which is the first-order derivatives of the ground truth depth (middle top) and our weighted quantized smooth ground truth normal (middel down). Top right is the normal direction field. Note that lighter pixel indicates that the surface normal is pointing closer to the z direction that points towards us. Bottom right is the confidence map that encourages higher confidence (lighter pixel) for planes than boundaries. Best viewed in color.

superiority in single image depth estimation [24]. We set the cut-off rate $c = 0.2 \max_p(|d_p - \hat{d}_p|)$ for each batch.

3.3. Implementation details

In inference, for each input image I and the object mask M_{object}, we first perform the largest center crop with the same aspect ratio as the network input size, then resize I to fit the network input size. The output depth map is then resized back to the same scale as the original cropped image by bilinear interpolation.

Mask dropout. Initial experiments indicated that depth regressions against images tend to have quite localized support, likely because very high spatial correlations in real images mean that large-scale support is superfluous. But a network that predicts depth in locations where there are no known pixel values needs to have spatial support on very long scales (so that a location where pixel values aren't known can draw from locations where the pixels are known). To achieve this, we randomly flip each pixel value in the object mask with a chance of 10%, meaning a mask dropout rate of 0.1. This forces the network to be able to use nearby pixels to predict depths. We mask out the flipped pixels when computing the loss to avoid error backpropagation. We show in experiments (Sec. 5.2) that training with mask dropout helps stabilize our performance.

Data Augmentation. During training, we perform random cropping instead of center cropping to increase the training samples. The window size varies between the fraction $\alpha = [\frac{2}{3}, 1]$ of the size of the largest center crop. We perform the same cropping for the ground truth depth map d. Note that a smaller cropping is equivalent to a closer view of the object, resulting in a smaller distance to the camera. We thus divide each pixel value d with α in order to preserve the depth scales across different crops of the same image. We also update each crop's normal given the re-scaled depth, using the weighted quantized smoothed normal computation as described in Sec. 3.2. Moreover, we perform random rotation on the image plane ranges in $[-5, 5]$ degrees, random horizontal flipping and image color changes with each of the RGB channel being multiplied by the weight ranges in $[0.8, 1.2]$ independently. Each augmentation parameter is uniformly and randomly sampled from the defined range.

4. Dataset

4.1. Training

To train our method, we need triples of ground truth: RGB image, object mask, depth with masked object being removed. Such datasets do not exist, and are difficult to make on a large scale. Instead, we make the ground truth tuples by rendering a synthetic dataset. However, a rendered dataset may not properly represent texture or illumination. We thus combine the data with the standard NYUd v2 [39] real dataset (where we have only empty object masks). Training samples are selected uniformly across each training set (synthetic or real), with a 50% probability of choosing one or another. We apply mask dropout on all object masks.

Synthetic: AI2-THOR [22] is an indoor virtual environment that supports physical simulation of objects in the scene. We modified the default simulation setting to be able to remove every object in the scene, rather than pickupable objects only. AI2-THOR has 120 predefined scenes from four categories of rooms: kitchen, living room, bedroom and bathroom. In each scene, we place an agent at a random location for 100 times. The height of agent is sampled under the normal distribution with mean of 1.0m and a standard derivation (std) of 0.1m. The agent looks at the scene with a randomly sampled altitude, which is normally distributed with a mean of 0^o (looking at horizon) and a std of 10^o. At each view, we generate the ground truth depth map with one of the objects removed. For each type of room, we use 27 scenes for training and withhold three scenes for testing. This creates 47k 640×480 image-depth pairs of synthetic samples. Each rendered depth map ranges up to 5 meters.

Real: NYUd v2 [39] is one of the widely used RGBD dataset with real indoor scenes. We use the official train and test split in our experiment.

4.2. Testing

Synthetic. We use the test split of AI2-THOR to compare with other baselines. We obtain 1162 test samples with depth changes of least 0.25m per pixel after the object is removed. Slight changes in depth can hardly be examined the performance.

Figure 4. **Image samples from the dataset we use**. Left to right: NYUd v2 [39] (real dataset), AI2-THOR [22] (synthetic dataset), our collected dataset (real dataset). AI2-THOR and our collected dataset has ground truth depth with object removed. Best viewed in color.

Factor	variables
shape complexity	simple (e.g. box), complex (e.g. chair)
shape rarity	common (e.g. box), rare (e.g. doll)
number of objects close by	0, 1, 2
object behind	wall, empty space, other objects
distance to the camera	1.5m, 2.0m

Table 1. Factors and variables used to construct our dataset.

Real. We have collected a small but carefully structured RGBD dataset for evaluation using Kinect v2, as shown in Figure 4. Our dataset contains both RGB images and the depth maps before and after the removal of objects. For each image, we carefully label a 2D tight object mask around the object to be removed. Our images are collected so as to investigate five factors that might affect the prediction error (Table 1): (1) the complexity of the object; (2) the rarity of the object in the training set; (3) number of other non-removed objects close by with similar depth; (4) the object location; (5) the distance between the object and the camera. The first two factors focus on the object itself and the latter three focus on the spatial relationship between the object and the scene. This results in $2 \times 2 \times 3 \times 3 \times 2 = 72$ testing cases. Please find more detailed dataset configurations in supplemental material.

5. Experiments

Experimental setup. We implement our network using MatConvNet and train it on a single NVIDIA Titan X GPU. We use the weights of pretrained ResNet-50 on ImageNet to initialize the the encoder, then train the whole network end-to-end. We use ADAM [21] to update network parameters with a batch size of 32 and an initial learning rate of 0.01. The learning rate is then halved after every 5 epochs and the whole training procedure takes around 20 epochs to converge. In our experiment, we set the term weights in Eq. 1 as: $w_1 = 1, w_2 = 0.5, w_3 = 1$.

Baselines. To demonstrate the effectiveness of our approach, we compare with three classes of natural baselines: (1) **"Do nothing"**. We simply ignore the mask and apply our approach to estimate image depth. In this case we're predicting image depth *with* the object. (2) **Depth inpainting**. We use the object mask to remove the object from our predicted depth map, then fill in the hole using three different methods. For the first method, we apply Poisson editing [35] to interpolate the missing depth based on neighboring depth values. For the second method, we apply a vanilla auto-encoder. The auto-encoder gets as input the concatenation of the depth map and the object mask, and predicts the scene depth with the object removed. The encoder (decoder) consists five convolution layers with kernel size of 3×3, with max pooling (scale factor 2) and ReLU in between, resulting in the same 8×10 bottleneck feature size as ours. We train the auto-encoder with the same setting as our approach. For the third method, we compare to the state-of-the-art depth hole filling approach DepthComp by Atapour *et al*. [1]. DepthComp requires additional input of semantic segmentation maps. We use the outputs from SegNet [3] trained on SUNRGBD [40] to run the experiment. (3) **Image inpainting**. Given the object mask, we inpaint the RGB image using the method by Iizuka *et al*. [18], then predict depth from the inpainted one using our approach.

For fair comparison, we use our network with no object mask to produce the initial depth map for all baselines. We evaluate the performance of our approach and all the baselines using the following standard single image depth estimation **evaluation metrics**:

- **rms**: root mean squared error: $\sqrt{\frac{1}{Q}\sum_p (d_p - \hat{d}_p)^2}$
- **mae**: mean absolute error: $\frac{1}{Q}\sum_p |d_p - \hat{d}_p|$
- **rel**: mean absolute relative error: $\frac{1}{Q}\sum_p \frac{|d_p - \hat{d}_p|}{\hat{d}_p}$
- $\boldsymbol{\delta_i}$: percentage of pixels where the ratio (or its reciprocal) between the prediction and the label is within a threshold, 1.25, to the power i: $\frac{1}{Q}\sum_p \mathbf{1}[\max(\frac{d_p}{\hat{d}_p}, \frac{\hat{d}_p}{d_p}) < 1.25^i]$. We set $i = \{1, 2, 3\}$.

Note that rms, mae, and rel are error metrics (the lower the better) and δ_i measures accuracy (the higher the better). For detailed analysis, we calculate the average pixel performance using the metrics on the entire image (all pixels), the region inside the mask (interior), and the region outside the mask (exterior). Performance on the entire image naturally shows the ability of predicting image depth with an object removed; performance on the interior region demonstrates the ability to predict the scene depth behind the object; and performance on the exterior region demonstrates the ability of predicting the depth of non-removed area.

5.1. Qualitative results

Depth with an object removed. We show in Figure 5 our qualitative performance compared with other baselines on NYUd v2 dataset. NYUd v2 does not have ground truth depth with the object removed, so we could only compare qualitatively. We use the ground truth 2D segmentation in NYUd v2 as the input object mask. Our approach is able to produce well-behaved depth behind the object and the depth of non-removed area, along with a good normal estimates for the hidden geometry. Note that depth predictions

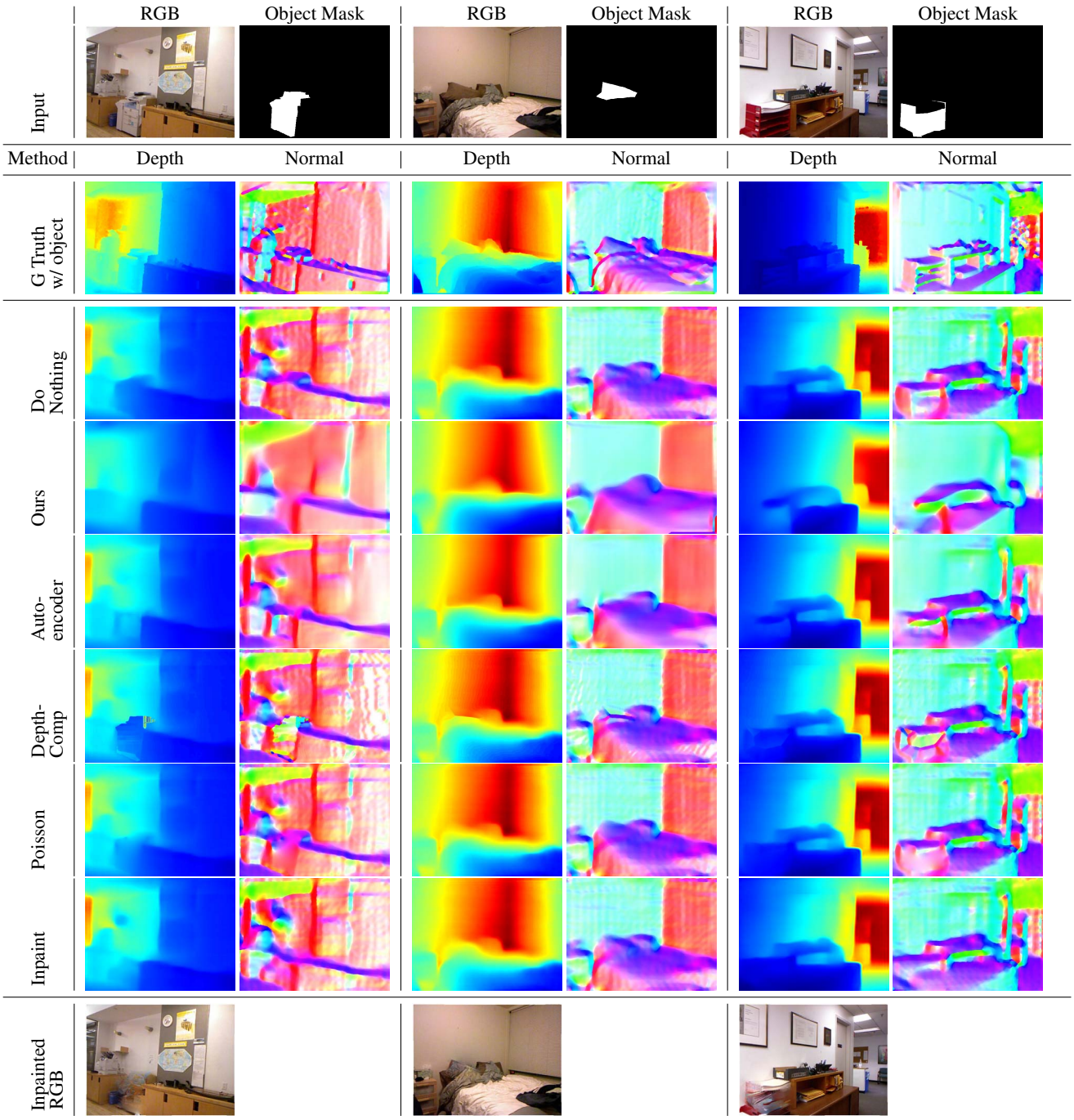

Figure 5. Qualitative results of depth estimation with the object *removed* on the NYUd v2 dataset [39]. We compare our approach to several baselines. We show in the second row the ground truth scene depth with all the object *non-removed*. For image inpainting baseline we also show the inpainted RGB image for analysis. The surface normal is derived from the predicted depth. Our method is able to estimate the hidden geometry behind the cupboard when the printer is removed (column 1); the space on top of the bed when the pillow is removed (column 2); and the space below the ream of paper when the shelves but not that paper are removed (column 3). Best viewed in color.

by the inpainting baseline are mangled by inpainting errors. Poisson smoothing produces somewhat better estimates, but fails in the obvious way when one side of the background is closer than the other (first column). We show in Figure 6 more qualitative results on our collected real dataset and the synthetic AI2-THOR dataset.

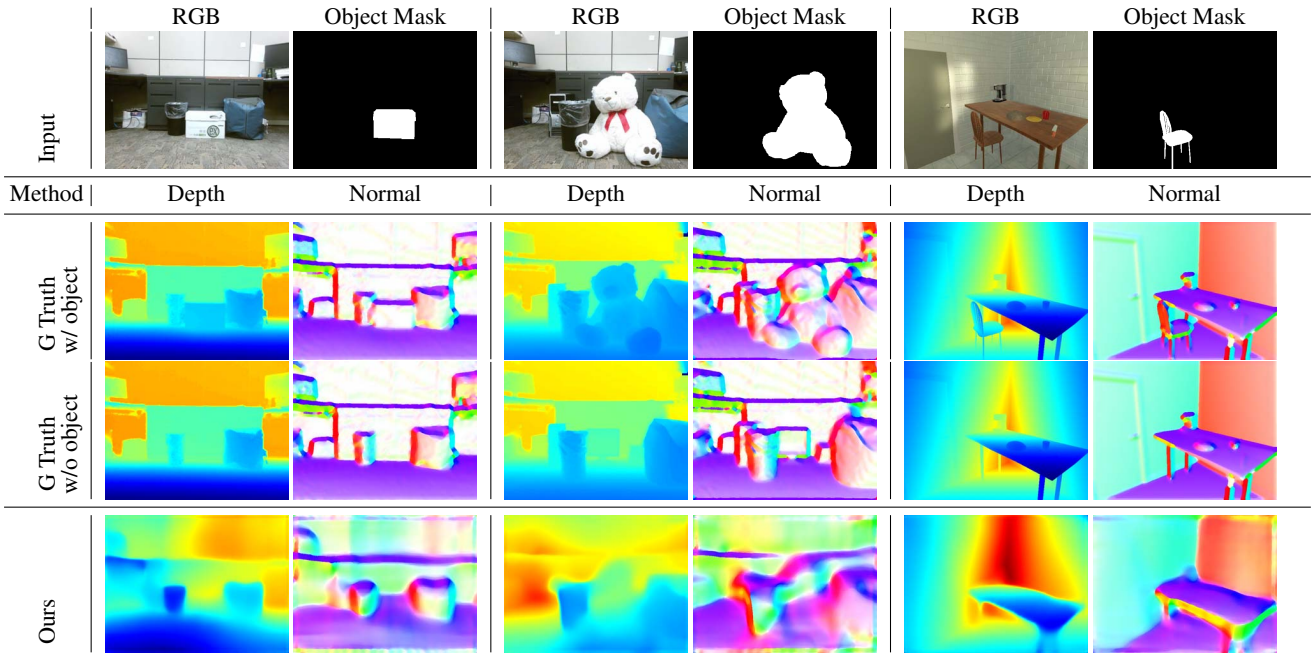

Figure 6. Qualitative results of depth estimation with the object *removed* on our collected real dataset (column 1,2) and the synthetic AI2-THOR testset (column 3). Both two datasets have the ground truth depth with the object removed shown in the third row. Note that our method is able to predict the gap between the bin and the bag behind the center box is (column 1); the gap between the bin and the bag behind the fluffy bear (column 2); and one of the table's leg that occluded by the removed chair (column 3). Our method has no explicit object model or semantics, and so is not puzzled by stuffed toys. Please refer to the supplemental material for comparisons with other baselines on the two datasets. Best viewed in color.

Method	All Pixels						Interior						Exterior					
	rms	mae	rel	δ_1	δ_2	δ_3	rms	mae	rel	δ_1	δ_2	δ_3	rms	mae	rel	δ_1	δ_2	δ_3
Do nothing	.548	.364	.158	75.6	92.5	98.0	.667	.498	.158	68.6	92.3	98.8	**.539**	.357	**.156**	**76.4**	93.0	**98.2**
Poisson	.548	.363	.158	75.9	92.6	97.9	.691	.492	.156	72.2	92.6	97.3	*	*	*	*	*	*
DepthComp	.546	.361	.158	76.0	92.7	98.1	.684	.490	.157	71.6	92.6	97.9	*	*	*	*	*	*
Inpaint	.582	.386	.165	73.8	91.3	97.6	.665	.479	.152	73.9	92.8	98.5	.577	.381	.164	74.2	91.5	97.8
Auto-encoder	.578	.390	.163	73.6	91.6	98.0	.602	.441	.139	77.1	**95.3**	**99.7**	.577	.388	.163	73.7	91.7	98.0
Ours	**.542**	**.359**	**.157**	**76.3**	92.9	**98.2**	.592	.423	.138	78.9	**95.3**	99.4	**.539**	**.356**	**.156**	**76.4**	93.0	**98.2**
Ours w/o mask dropout	.542	.364	.162	75.0	**93.5**	97.8	**.569**	**.407**	**.133**	**80.2**	**95.3**	99.1	.540	.363	.162	75.1	**93.7**	97.8
Ours w/o norm	.629	.430	.187	70.1	89.5	96.1	.678	.490	.158	73.9	92.4	97.4	.627	.428	.186	70.2	89.6	96.1

Table 2. Depth estimation performance with object *removed* compared with other baselines on the synthetic AI2-THOR test set. We evaluate average pixel performance on all image pixels (All Pixels), pixels inside the object mask (Interior) and pixels outside the object mask (Exterior). All baselines get initial depths (without object remove) from our method with the object masked out. The "*" in exterior columns means that the method does not produce pixels in this region. We also show in the last two rows the ablation study of our network without mask dropout and without our surface normal loss. **Bold** shows the best score in each column.

Method	All Pixels						Interior						Exterior					
	rms	mae	rel	δ_1	δ_2	δ_3	rms	mae	rel	δ_1	δ_2	δ_3	rms	mae	rel	δ_1	δ_2	δ_3
Do Nothing	.447	.368	.207	67.0	90.6	99.6	.600	.513	.267	35.8	67.6	97.0	**.430**	**.355**	.201	**69.9**	92.7	99.8
Poisson	.427	.352	.198	69.6	92.8	99.8	.394	.320	.168	66.8	93.9	99.9	*	*	*	*	*	*
DepthComp	.438	.360	.203	68.0	91.6	99.7	.513	.424	.225	47.9	79.7	98.8	*	*	*	*	*	*
Inpaint	.538	.434	.258	60.2	86.4	98.9	.526	.445	.235	52.3	92.6	99.8	.539	.433	.260	60.9	85.9	98.8
Auto-encoder	.431	.360	.192	66.0	**95.0**	**100.**	.353	.290	.153	70.5	97.7	**100.**	.437	.366	.196	65.5	**94.7**	**100.**
Ours	**.425**	**.349**	.198	**70.6**	93.0	99.8	**.310**	**.247**	**.133**	**81.9**	**99.6**	**100.**	.435	.359	.204	69.5	92.4	99.8
Ours w/o mask dropout	.762	.612	.272	38.9	71.3	90.1	.517	.416	.203	51.3	87.5	99.3	.781	.630	.279	37.7	69.7	89.3
Ours w/o norm	.455	.364	**.188**	66.7	93.6	99.3	.393	.310	.160	68.8	96.0	99.8	.460	.369	**.191**	66.5	93.4	99.2

Table 3. Depth estimation performance with object *removed* compared with other baselines on our collected evaluation dataset. We evaluate average pixel performance on all image pixels (All Pixels), pixels inside the object mask (Interior) and pixels outside the object mask (Exterior). All baselines get initial depths (without object removed) from our method with the object masked out.

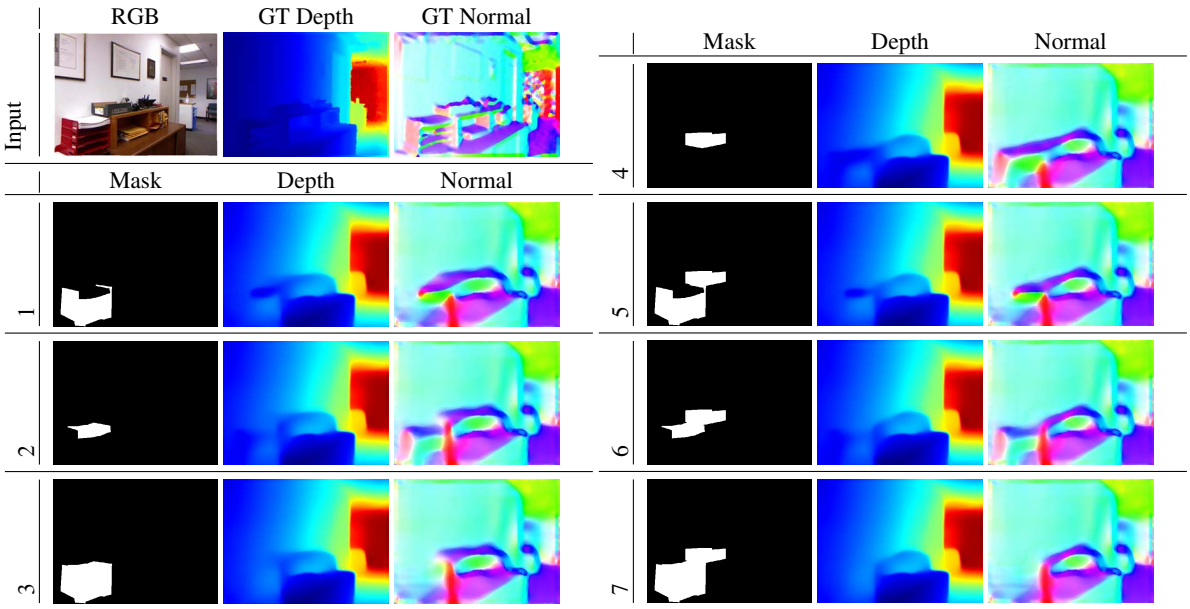

Figure 7. Qualitative results of depth estimation with multiple objects *removed* on the NYUd v2 dataset [39]. In the first row, we show from left to right the inout RGB image, ground truth depth with all objects and the derived surface normal. In each following example we show from left to right the input object mask, our predicted depth with the object(s) removed and the derived surface normal. We show seven different input object masks as different combinations of three objects: a bookshelf, a ream of paper on the bookshelf, and the box beside the bookshelf. Our network is able to remove object(s) within the supplied mask and retain other objects in the scene.

Depth with *multiple* objects removed. One important benefit of using object mask as input is that we can arbitrarily remove any number of objects from the scene and predict the depth without these objects. Figure 7 demonstrates the ability of our network to estimate scene depth with different combinations of objects removed from the same scene. Our approach is also able to produce consistent predictions for non-removed area (e.g. layouts, counter) in the same scene.

5.2. Quantitative results

We show in Table 2 our quantitative comparison on the test set of the synthetic AI2-THOR dataset. Table 3 reports the performance on our collected real dataset. Poisson and DepthComp do not perturb depth outside the object mask region, hence, their exterior region is equal to "Do nothing". We report their error metrics in exterior as *. Our method outperforms all baselines on most metrics. Inpainting method does not work; Poisson and DepthComp have trouble removing an object. Auto-encoder and ours produce comparatively good interior (ours still slightly better) depth, but Auto-encoder produces worse depth estimates of exterior region. Note that for some measurements the depth prediction performance inside the object masked could be better than the prediction on the whole image scale. We believe that it's uncommon that objects mask other clutter, so the masked scene tends to be walls, floors, etc., where depth has simpler statistics and is easier to predict.

Ablation study. We show in Table 2 and Table 3 the performance gains by training with our smoothed ground truth normal loss (ours v.s. ours w/o normal) and the mask dropout data augmentation (ours v.s. ours w/o mask).

Factors that affect error. We investigate how properties of test data affect the error of the method, by regressing error against the attributes of the test images (Sec. 4.2) and looking for significant predictors. We use both individual terms and pairwise interactions, and apply an ANOVA. Please find detailed analysis in supplemental material.

Single image depth *with* the object. For images where no object is removed, our approach is able to predict scene depth that is of comparable quality to that of state-of-the-art single image depth estimation methods. Please find detailed evaluations in supplemental material.

6. Conclusion

We have introduced a new task – estimating the hidden geometry behind the object. Our method takes as input a single RGB image and an object mask, and predicts a depth map that describes the scene when the object is removed. We show, both qualitatively and quantitatively, that our approach is able to predict depth behind objects better than other baselines, and is flexible in removing multiple objects. Our approach can be further utilized for applications like object insertion and manipulation in a single RGB image.

Acknowledgements

This research is supported in part by ONR MURI grant N00014-16-1-2007.

References

[1] A. Atapour-Abarghouei and T. P. Breckon. Depthcomp: real-time depth image completion based on prior semantic scene segmentation. 2017. 2, 5

[2] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2810, 2018. 2

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 5

[4] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001. 2

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24, 2009. 2

[6] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. 2

[7] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2001. 2

[8] A.-L. Chauve, P. Labatut, and J.-P. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1261–1268. IEEE, 2010. 2, 3

[9] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 2

[10] W. Chen, D. Xiang, and J. Deng. Surface normals in the wild. In *Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy*, pages 22–29, 2017. 2

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 2

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 2

[13] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. 1, 2

[14] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 2

[15] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007. 2

[16] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *European Conference on Computer Vision*, pages 224–237. Springer, 2010. 2

[17] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf. Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)*, 33(4):129, 2014. 2, 3

[18] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 2, 5

[19] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014. 2

[20] Y. Kim, H. Jung, D. Min, and K. Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE Transactions on Image Processing*, 27(8):4131–4144, 2018. 2

[21] D. Kinga and J. B. Adam. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5, 2015. 5

[22] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 4, 5

[23] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017. 2

[24] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 2, 4

[25] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2

[26] C. Liu, P. Kohli, and Y. Furukawa. Layered scene decomposition via the occlusion-crf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–173, 2016. 2

[27] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018. 2

[28] J. Liu, X. Gong, and J. Liu. Guided inpainting and filtering for kinect depth maps. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 2055–2058. IEEE, 2012. 2

[29] M. Liu, X. He, and M. Salzmann. Building scene models by completing and hallucinating depth and semantics. In *European Conference on Computer Vision*, pages 258–274. Springer, 2016. 2

[30] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 716–723, 2014. 2

[31] F. Mal and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018. 2

[32] A. B. Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007. 3

[33] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 2

[34] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, number CONF, 2005. 2

[35] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003. 2, 5

[36] D. Ron, K. Duan, C. Ma, N. Xu, S. Wang, S. Hanumante, and D. Sagar. Monocular depth estimation via deep structured models with ordinal constraints. In *2018 International Conference on 3D Vision (3DV)*, pages 570–577. IEEE, 2018. 2

[37] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006. 2

[38] J. Shen and S.-C. S. Cheung. Layer depth denoising and completion for structured-light rgb-d cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1187–1194, 2013. 2

[39] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 3, 4, 5, 6, 8

[40] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 5

[41] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–198. IEEE, 2017. 1, 2

[42] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 2

[43] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017. 2

[44] Y. Zhang and T. Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018. 2

[45] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 2