

# SharpNet: Fast and Accurate Recovery of Occluding Contours in Monocular Depth Estimation

Michaël Ramamonjisoa<sup>1</sup>      Vincent Lepetit<sup>1</sup>

<sup>1</sup>LIGM (UMR 8049), École des Ponts, UPE

{michael.ramamonjisoa,vincent.lepetit}@enpc.fr

## Abstract

We introduce SharpNet, a method that predicts an accurate depth map given a single input color image, with a particular attention to the reconstruction of occluding contours: Occluding contours are an important cue for object recognition, and for realistic integration of virtual objects in Augmented Reality, but they are also notoriously difficult to reconstruct accurately. For example, they are a challenge for stereo-based reconstruction methods, as points around an occluding contour are only visible in one of the two views. Inspired by recent methods that introduce normal estimation to improve depth prediction, we introduce novel terms to constrain normals, depth and occluding contours predictions. Since ground truth depth is difficult to obtain with pixel-perfect accuracy along occluding contours, we use synthetic images for training, followed by fine-tuning on real data. We demonstrate our approach on the challenging NYUv2-Depth dataset, and show that our method outperforms the state-of-the-art along occluding contours, while performing on par with the best recent methods for the rest of the images. Its accuracy along the occluding contours is actually better than the “ground truth” acquired by a depth camera based on structured light. We show this by introducing a new benchmark based on NYUv2-Depth for evaluating occluding contours in monocular reconstruction, which is our second contribution.

## 1. Introduction

Monocular depth estimation is a very ill-posed yet highly desirable task for applications such as robotics, augmented or mixed reality, autonomous driving, and scene understanding in general. Recently, many methods have been proposed to solve this problem using Deep Learning approaches, either relying on supervised learning [6, 5, 18, 8] or on self-supervised learning [10, 33, 22], and these methods often yield very impressive results.

Despite recent advances in monocular depth estimation,

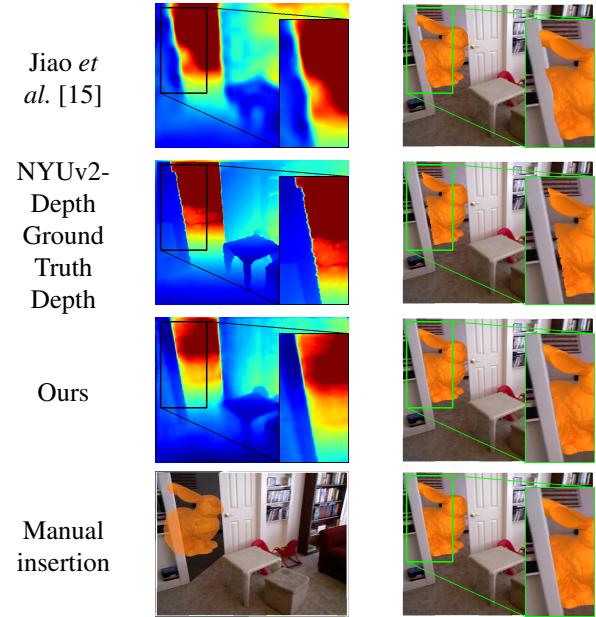


Figure 1: Our SharpNet method shows significant improvement over state-of-the-art monocular depth estimation methods in terms of occluding contours accuracy, and even produces sharper edges along these contours than structured-light depth cameras. In this figure we augment an RGB image from NYUv2 [25] with a virtual Stanford rabbit using different depth maps for occlusion-aware integration. The first three rows show the depth map used for occlusion-aware insertion (left) and resulting augmentation (right). An error of only a few pixels along occluding contours can significantly degrade the realism of the integration, comparatively to manual insertion (last row) using a binary mask.

occluding contours remain difficult to reconstruct correctly from depth as shown in Fig. 1, while they are still an important cue for object recognition, and for augmented reality or path planning, for example. This has several causes:

First, the depth annotations of training images are likely to be inaccurate along the occluding contours, if the depth annotations are obtained with a stereo reconstruction method or a structured light camera. This is for example the case for the NYUv2-Depth dataset [25], which is an important benchmark used by many recent works for evaluation. This is because on one or both sides of the occluding contours lie 3D points that are visible in only one image, which challenges the 3D reconstruction [27]. Structured light cameras essentially rely on stereo reconstruction, where one image is replaced by a known pattern [11], and therefore suffer from the same problem. Secondly, occluding contours, despite their importance, represent a small part of the images, and may not influence the loss function used during training if they are not handled with special care.

In this paper, we show that it is possible to learn to reconstruct occluding contours more accurately by adding a simple term that constrains the depth predictions together with the occluding contours during learning. This approach is inspired by recent works that predict the depths and normals for an input image, and enforce constraints between them [30, 23, 35]. A similar constraint between depth and occluding contours can be introduced, and we show that this results in better reconstructions along the occluding contours, without degrading the accuracy of the rest of the reconstruction.

Specifically, we train a network to predict depths, normals, and occluding contours for an input image, by minimizing a loss function that integrates constraints between the depths and the occluding contours, and also between depths and normals. We show that these two constraints can be integrated in a very similar way with simple terms in the loss function. At run-time, we can predict only the depth values, making our method suitable for real-time applications since it runs at 150 fps on  $640 \times 480$  images.

We show that each aspect of our training procedure improves the depth output. In particular, our experiments show that the constraint between depths and occluding contours is important, and that the improvement is not only due to multi-task learning. Learning to predict the normals in addition to the depths and the occluding contours helps the convergence of training towards good depth predictions.

We demonstrate our approach on the NYUv2-Depth dataset, in order to compare it against previous methods. Since we need training data with pixel perfect depth annotation along the occluding contours, we use synthetic images to pretrain the network before fine-tuning on NYUv2-Depth. We simply use the object instance boundaries given by the synthetic dataset as training annotations of the occluding contours. However, we only use the depth ground truth as training data when finetuning on the NYUv2-Depth dataset.

A proper evaluation of the accuracy of the occluding

contours is difficult. Since the “ground truth” depth data is typically noisy along occluding contours, as in NYUv2-Depth, an evaluation based on this data would not be representative of the actual quality. Even with better depth data, identifying occluding contours automatically as ground truth depth discontinuities would be sensitive to the parameters used by the identification method [1] (see Fig. 4).

We therefore decided to annotate manually the occluding contours in a subset of 100 images randomly sampled from the NYUv2-Depth validation set, which we call the NYUv2-OC dataset. Our annotations and our code for the evaluation of the occluding contours are publicly available for comparison. We evaluate our method on this data in terms of 2D localization, in addition to evaluating depth estimation on the NYUv2-Depth validation set using more standard depth estimation metrics [6, 5, 18]. Our experiments show that while achieving competitive results on the NYUv2-Depth benchmark by placing second on all of them, we outperform all previous methods in terms of occluding contours 2D localization, especially the current leading method on monocular depth estimation [15].

## 2. Related Work

Monocular depth estimation from images made significant progress recently. In the following, we mainly discuss the most recent methods and popular techniques that help monocular depth estimation: Learning from synthetic data and using normals for learning to predict depths.

### 2.1. Supervised and Self-Supervised Monocular Depth Estimation

With the development of large datasets of images annotated with depth data [25, 9, 26, 39, 3, 20], many supervised methods have been proposed. Eigen *et al.* [6, 5] used multi-scale depth estimation to capture global and local information to help depth prediction. Given the remarkable performances they achieved on both popular benchmarks NYUv2-Depth [25] and KITTI [9], more work extended this multi-scale approach [19, 34]. Previous work also considered ordinal depth classification [8] or pair-wise depth-map comparisons [2] to add local and non-local constraints. Our approach relies on a simpler mono-scale architecture, making it efficient at run-time. Our constraints between depths, normals, and occluding contours guide learning towards good depth prediction for the whole image.

Laina *et al.* [18] exploited the power of deep residual neural networks [12] and showed that using the more appropriate BerHu [21, 40] reconstruction loss yields better performances. However, their end results are quite smooth around occluding contours, making their method inappropriate for realistic occlusion-aware augmented reality.

Jiao *et al.* [15] noticed that the depth distribution of the NYUv2 dataset is heavy-tailed. The authors therefore pro-

posed an attention-driven loss for the network supervision, and paired the depth estimation task with semantic segmentation to improve performances on the dataset. However, while they currently achieve the best performance on the NYUv2-Depth dataset, their approach suffers from a bias towards high-depth areas such as windows, corridors or mirrors. While this translates into a significant decrease of the final error, it also produces blurry depth maps, as one can see in Fig. 1. By contrast, our reconstructions tend to be much sharper along the occluding boundaries as desired, and our method is much faster, making it suitable for real-time applications.

Self-supervised learning methods have also become popular for monocular reconstruction, as they exploit the consistency between multiple views [10, 33, 22, 36, 37, 28]. While such approach is very appealing, it does not yet reach the accuracy of supervised methods in general, and it should be preferred only when no annotated data are available for supervised learning.

## 2.2. Edge- and Occlusion-Aware Depth Estimation

Wang *et al.* [30] introduced their SURGE method to improve scene reconstruction on planar and edge regions by learning to jointly predict depth and normal maps, as well as edges and planar regions. They then refine the depth prediction by solving an optimization problem using a Dense Conditional Random Field (DCRF). While their method yields appealing reconstruction results on planar regions, it still underperforms state-of-the-art methods on global metrics, and the use of DCRF makes it unsuited for real-time applications. Furthermore, SURGE [30] is evaluated on the reconstruction quality around edges using standard depth error metrics, but not on the 2D localization of their occluding contours.

Many self-supervised methods [36, 35, 37, 28, 10] have incorporated edge- or occlusion-aware geometry constraints which exist when working with stereo pairs or sequences of images as provided in the very popular KITTI depth estimation benchmark [9]. However, although these methods can perform monocular depth estimation at test time, they require multiple calibrated views at training time. They are therefore unable to work on monocular RGB-D datasets such as NYUv2-Depth [25] or SUN-RGBD [26].

[31, 14] worked on occlusion-aware depth estimation to improve reconstruction for augmented reality applications. While achieving spectacular results, they however require one or multiple light-field images, which are more costly to obtain than ubiquitous RGB images.

Conscious of the lack of evaluation metrics and benchmarks for quality of edge and planes reconstruction from monocular depth estimates, Koch *et al.* [16] introduced the iBims-v1 dataset, a high quality benchmark of 100 RGB images with their associated depth map. This work tack-

les the low quality of depth maps of other RGB-D datasets such as [26] and [25], and introduces annotations and metrics for occluding contours and planarity of planar regions. Our evaluation method of occluding contours reconstruction quality is based on their work.

## 3. Method

As shown in Fig. 2, we train a network  $f(\mathbf{I}; \Theta)$  to predict, for a training color image  $\mathbf{I}$ , a depth map  $\hat{\mathbf{D}}$ , a map of occluding contours probabilities  $\hat{\mathbf{C}}$ , and a map  $\hat{\mathbf{N}}$  of surface normals. Although we focus on high quality depth-maps prediction, our occluding contours and normals map can also be used for other applications. Our approach generalizes well to various indoor datasets in terms of geometry estimation as can be seen in Fig. 3.

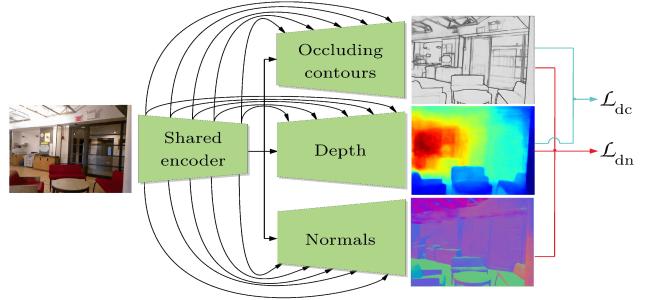


Figure 2: The architecture of our “U-net”-shape [24] multi-task encoder-decoder network. We use a single ResNet50 encoder which learns an intermediate representation that is shared by all decoders. With this setting, the representation generalizes better for all tasks. We use skip connections between features of the encoder and of the decoder at corresponding scales.

### 3.1. Training Overview

We first train  $f$  on the synthetic dataset PBRS [39], which provides the ground truth for the depth map  $\mathbf{D}$ , the normals map  $\mathbf{N}$ , and the binary map of object instance contours  $\mathbf{C}$  for each training image  $\mathbf{I}$ . Since *occluding* contours are not directly provided in the PBRS dataset, we choose to use the *object instance* contours  $\mathbf{C}$  as a proxy. We argue that on a macroscopic scale, a large proportion of occluding contours in an image are due to objects occluding one another. However, we show that we can also enable our network to learn internal occluding contours within objects even without “pure” occluding contours supervision. Indeed, we make use of constraints on depth map and occluding contour predictions  $\hat{\mathbf{D}}$  and  $\hat{\mathbf{C}}$  respectively (see Section. 3.4 for more details) to enforce the contour estimation task to also predict intra-object occluding boundaries.

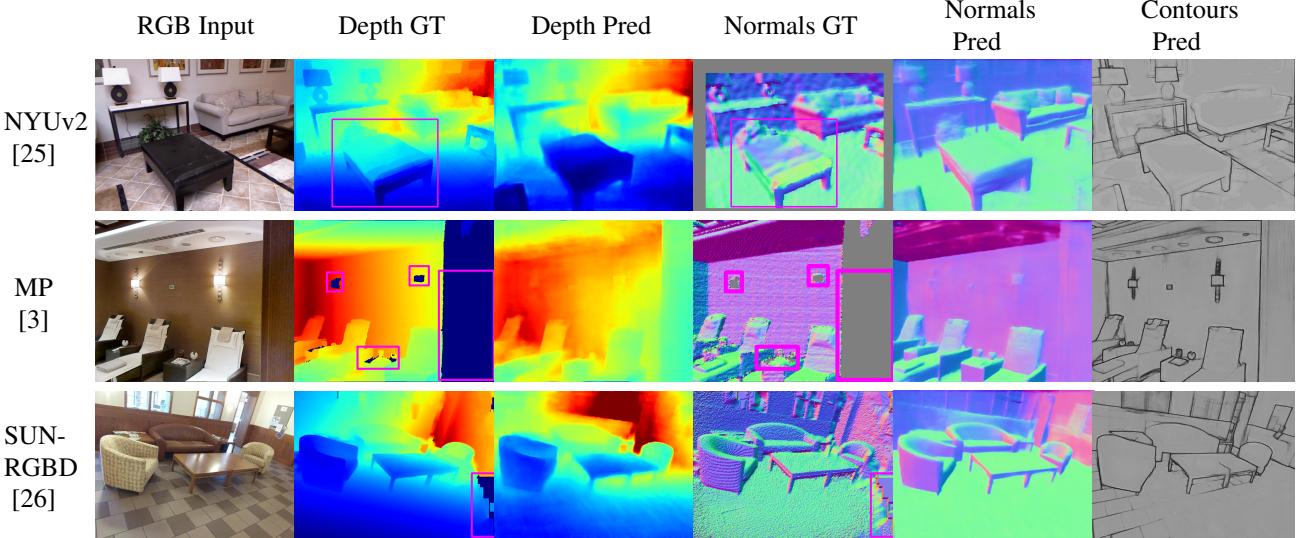


Figure 3: Several predictions on single RGB images from multiple real RGB-D datasets. “MP” stands for Matterport3D, “GT” stands for ground truth and “Pred” for prediction. We highlight areas where we successfully reconstructed geometry while Kinect depth maps were inaccurate (the chair should be closer than the lamp in first image). Ground truth normals are computed using code from [25] for NYUv2 and [20] for SUN-RGBD. Normal maps are already provided in Matterport3D.

We then finetune  $f$  on the NYUv2-Depth dataset without direct supervision on the occluding contours or normals ( $\mathcal{L}_c$  and  $\mathcal{L}_n$  described below): Even though [17] and [25] produce ground truth normals map with different estimation methods operating on the Kinect-v1 depth maps, their output results are generally noisy. Occluding contours are not given in the original NYUv2-Depth dataset. Although one could automatically extract them using edge detectors [1, 4] on depth maps, such extraction is very sensitive to the detector’s parameters (see Figure 4). Instead, we introduce consensus terms that explicitly constrain the predicted contours, normals and depth maps together ( $\mathcal{L}_{dc}$  and  $\mathcal{L}_{dn}$  described below) at training time.

At test-time, we can choose to use only the depth stream of  $f$  if we are not interested in the normals nor the boundaries, making inference very fast.

### 3.2. Loss Function

We estimate the parameters  $\Theta$  of network  $f$  by minimizing the following loss function over all the training images:

$$\begin{aligned} \mathcal{L} = & \lambda_d \mathcal{L}_d(\mathbf{D}, \widehat{\mathbf{D}}) + \lambda_c \mathcal{L}_c(\mathbf{C}, \widehat{\mathbf{C}}) + \lambda_n \mathcal{L}_n(\mathbf{N}, \widehat{\mathbf{N}}) + \\ & \mathcal{L}_{dc}(\widehat{\mathbf{D}}, \widehat{\mathbf{C}}) + \mathcal{L}_{dn}(\widehat{\mathbf{D}}, \widehat{\mathbf{N}}), \end{aligned} \quad (1)$$

where

- $\mathcal{L}_d$ ,  $\mathcal{L}_c$ , and  $\mathcal{L}_n$  are supervision terms for the depth, the occluding contours, and the normals respectively. We adjust weights  $\lambda_d$ ,  $\lambda_c$ , and  $\lambda_n$  during training so that

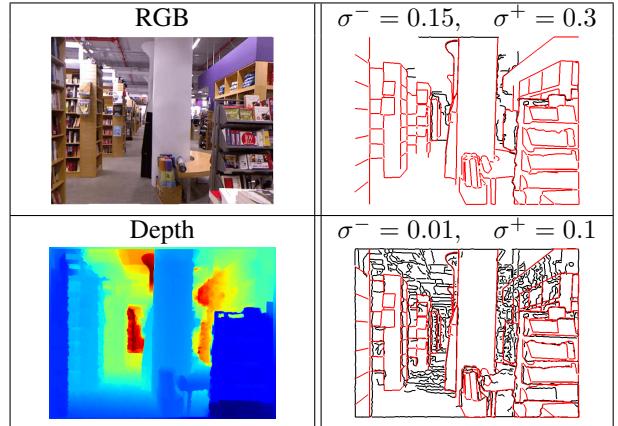


Figure 4: A RGB-D sample of NYUv2-Depth for which we manually annotated occluding contours in NYUv2-OC, (in red lines). We show in black the edges detected on ground truth Kinect-v1 depth map using different Canny detector parameters ( $\sigma^-$  and  $\sigma^+$  denote low and high threshold respectively). Highly permissive detectors often yield many spurious contours, whereas restrictive ones miss many true contours. Automatic occluding contours extraction from Kinect depth maps is therefore unreliable for extraction of ground truth occluding contours, motivating our manually annotated NYUv2-OC dataset.

we focus first on learning local geometry (normals and

boundaries) then on depth. See Section 4.1 for more details.

- $\mathcal{L}_{dc}$  and  $\mathcal{L}_{dn}$  introduce constraints between the predicted depth map and the predicted contours, and between the predicted depth map and the predicted normals respectively.

We detail these losses below. All losses are computed using only valid pixel locations. The PBRS synthetic dataset provides such a mask. When finetuning on NYUv2-Depth, we mask out the white pixels on the images border.

### 3.3. Supervision Terms $\mathcal{L}_d$ , $\mathcal{L}_c$ , and $\mathcal{L}_n$

The supervision terms on the predicted depth and normal maps are drawn from previous works on monocular depth prediction. For our term on occluding contours prediction, we rely on previous work for edge prediction.

**Depth prediction loss  $\mathcal{L}_d$ .** As in recent works, our loss on depth prediction applies to log-distances. We use the BerHu loss function [21, 40], as it was shown in [18] to result in faster converging and better solutions:

$$\begin{aligned} \mathcal{L}_d(\mathbf{D}, \widehat{\mathbf{D}}) &= \frac{1}{N} \sum_i \text{BerHu}(\log(\widehat{D}_i) - \log(D_i)) \\ &\quad + \frac{1}{N} \sum_i \|\nabla \log(\widehat{D}_i) - \nabla \log(D_i)\|^2. \end{aligned} \quad (2)$$

The sum is over all the  $N$  valid pixel locations. The BerHu (also known as reverse Huber) function is defined as a  $L_2$  loss for large deviations, and a  $L_1$  loss for small ones. As in [18], we take the  $c$  parameter of the BerHu function as  $c = \frac{1}{5} \max_i(|\log(\widehat{D}_i) - \log(D_i)|)$ .

**Occluding contours prediction loss  $\mathcal{L}_c$ .** We use the recent attention loss from [29], which was developed for 2d edge detection, to learn to predict the occluding contours. This attention loss helps dealing with the imbalance of edge pixels compared to non-edge pixels:

$$\text{AL}(\hat{p}, p) = \begin{cases} -\alpha\beta^{(1-\hat{p})^\gamma} \log(\hat{p}) & \text{if } p = 1 \\ -(1-\alpha)\beta^{\hat{p}^\gamma} \log(1-\hat{p}) & \text{else} \end{cases} \quad (3)$$

where  $(\beta, \gamma)$  are hyper-parameters which we set to the authors values (4, 0.5), and  $\alpha$  is computed image per image as the proportion of contour pixels. We use this pixel-wise attention loss to define the occluding contours prediction loss:

$$\mathcal{L}_c(\mathbf{C}, \widehat{\mathbf{C}}) = \frac{1}{N} \sum_i \text{AL}(\widehat{C}_i, C_i). \quad (4)$$

As mentioned above, this loss is disabled when finetuning on the NYUv2-Depth dataset.

**Normals prediction loss  $\mathcal{L}_n$ .** For normals prediction, we use a common method introduced by Eigen *et al.* [5] which is to minimize, for all valid pixels  $i$ , the angle between the predicted normals  $\widehat{\mathbf{N}}_i$  and their ground truth counterpart  $\mathbf{N}_i$ . This angle minimization is performed by maximizing their dot-product. We therefore used the following loss:

$$\mathcal{L}_n(\mathbf{N}, \widehat{\mathbf{N}}) = \frac{1}{N} \sum_i \left( 1 - \frac{\langle \widehat{\mathbf{N}}_i, \mathbf{N}_i \rangle}{\|\widehat{\mathbf{N}}_i\| \|\mathbf{N}_i\|} \right). \quad (5)$$

This loss slightly differs from the one of [5] as we limit it to positive values. As mentioned earlier, this loss is disabled when finetuning on the NYUv2-Depth dataset.

### 3.4. Consensus Terms $\mathcal{L}_{dc}$ and $\mathcal{L}_{dn}$

**Depth-contours consensus term.** In order to force the network to predict sharp depth edges at occluding contours where strong depth discontinuities occur, we propose the following loss between the predicted occluding contours probability map  $\widehat{\mathbf{C}}$  and the predicted depth map  $\widehat{\mathbf{D}}$ :

$$\begin{aligned} \mathcal{L}_{dc}(\widehat{\mathbf{D}}, \widehat{\mathbf{C}}) &= -\frac{1}{N} \sum_i \log(\widehat{C}_i) \cdot \|\nabla(\widehat{D}_i)\|^2 \|\Delta(\widehat{D}_i)\| \\ &\quad + \mu \left( \|\widehat{\mathbf{C}}\| - \frac{1}{N} \sum_i \log(1 - \widehat{C}_i) \cdot e^{-\|\Delta(\widehat{D}_i)\|} \right). \end{aligned} \quad (6)$$

This encourages the network to associate pixels with large depth gradients with occluding contours: High-gradient areas will lead to a large loss unless the occluding contour probability is close to one. [10, 13] also used this type of edge-aware gradient-loss, although they used it to impose consensus between photometric gradients and depth gradients. However, relying on photometric gradients can be dangerous: textured areas can exhibit strong image gradients without strong depth gradients.

**Depth-normals consensus loss.** Depth and normals are two highly correlated entities. Thus, to impose geometric consistency during prediction between the normal and depth predictions  $\widehat{\mathbf{D}}$  and  $\widehat{\mathbf{N}}$ , we use the following loss:

$$\mathcal{L}_{dn}(\widehat{\mathbf{D}}, \widehat{\mathbf{N}}) = \frac{1}{N} \sum_i \left( 1 - \frac{\langle \widehat{\mathbf{u}}_i, \widehat{\mathbf{n}}_i \rangle}{\|\widehat{\mathbf{u}}_i\| \|\widehat{\mathbf{n}}_i\|} \right), \quad (7)$$

where  $\widehat{\mathbf{n}}_i = (\widehat{n}_x^i, \widehat{n}_y^i)^T$  is extracted from the 3D vector  $\widehat{\mathbf{N}}_i = (\widehat{n}_x^i, \widehat{n}_y^i, \widehat{n}_z^i)^T$ , and  $\widehat{\mathbf{u}}_i = (\partial_x \widehat{D}_i, \partial_y \widehat{D}_i)$  is computed as the 2D gradient of the depth map estimate using finite differences. This term enforces consistency between the normals and depth predictions in a similar fashion as in [30, 35, 7]. However, our formulation of depth-normals consensus is much simpler than those proposed in previous

works as they express their constraint in 3D world coordinates, thus requiring the camera calibration matrix. Instead, we only assume that orthographic projection holds, which is a good first order assumption [32].

Imposing this constraint during finetuning allows us to constrain normals, and depth, even when the ground truth normals  $\mathbf{N}$  are not available (or accurate enough for our application).

## 4. Experiments

We evaluate our method and compare it to previous work using standard metrics, as well as the depth boundary edge (DBE) accuracy metric introduced by Koch et al. [16] (see following Section 4.2 and Eq. (8) for more details). We show that our method achieves the best trade-off between global reconstruction error and DBE.

### 4.1. Implementation Details

We implement our work in Pytorch and make our pre-trained weight, training and evaluation code publicly available.<sup>1</sup> Both training and evaluation are done on a single high-end NVIDIA GTX 1080 Ti GPU.

**Datasets.** We first train our network on the synthetic PBRS [39] dataset, using depth and normals maps annotations, along with object instance boundaries maps which we use as a proxy to occluding contours annotations. We split the PBRS dataset in training/validation/test sets using a 80%/10%/10% ratio. We then finetune our network on the NYUv2-Depth training set using only depth data. Finally, we use the NYUv2-Depth validation set for depth evaluation and our new NYUv2-OC for occluding contours accuracy evaluation.

**Training.** Training a multi-task network requires some caution: Since several loss terms are involved, and in particular one for each task, one should pay special attention to any suboptimal solution for one task due to ‘over-learning’ another. To monitor each task individually, we monitor each individual loss along with the global training loss and make sure that all of them decrease during training. When setting all loss coefficients equal to one, we noticed that the normals loss  $\mathcal{L}_{normals}$  decreased faster than others. Similarly, we found that learning boundaries was much faster than learning depth. As [38], we also argue that this is because local features such as contours or local planes, *i.e.* where normals are constant, are easier to learn since they appear in almost all training examples. Training depth, however, requires the network to exploit context data such as room layout in order to regress a globally consistent depth map.

---

<sup>1</sup> [www.github.com/MichaelRamamonjisoa/SharpNet](http://www.github.com/MichaelRamamonjisoa/SharpNet)

Based on those observations, we chose to learn the easier tasks first, then use them as guidance to the more complex task of depth estimation through our novel consensus loss terms of Eqs. (7) and (6). For finetuning on real data with the NYUv2 dataset, we first disabled the consensus terms and froze the contours and normals decoders in order to first bridge the depth domain gap between PBRS and NYUv2. After convergence, we finetuned the network again with consensus terms back on, which helped enhancing predictions by ensuring consistency between geometric entities. We found that it was necessary to freeze the normals and contours decoders during finetuning to prevent their predictions  $\widehat{\mathbf{C}}$  and  $\widehat{\mathbf{N}}$  from degrading until being unable to play their geometry guidance role. We argue that this is due to (1) a larger synthetic-to-real domain gap for depth than for contours and normals, and (2) noisy depth ground truth with some inaccuracies along occluding contours and crease along walls. We therefore relied on the ResNet50 encoder to learn a representation which produces geometrically consistent predictions  $\widehat{\mathbf{C}}$ ,  $\widehat{\mathbf{N}}$  and  $\widehat{\mathbf{D}}$ .

### 4.2. Evaluation Method

We evaluate our method on the benchmark dataset NYUv2 Depth [25]. The most common metrics are: Thresholded accuracies ( $\delta_1, \delta_2, \delta_3$ ), linear and logarithmic Root Mean Squared Error  $RMSE_{lin}$  and  $RMSE_{log}$ , Absolute Relative difference  $rel$ , and logarithmic error  $\log_{10}$ .

**NYUv2-Depth benchmark evaluation.** We have run a comparative study between our method and previous ones, summarized in Table 1. Since authors evaluating on the NYUv2-Depth benchmark often apply different evaluation methods, fair comparison is difficult to perform. For instance, [34] and [8] evaluate on crops with regions provided by Eigen *et al.* [5]. Some authors also clip resulting depth-maps to the valid depth sensor range [0.7m; 10m]. Most importantly, not all the authors make their prediction and/or evaluation code publicly available. The authors of [15] kindly shared their predictions on the NYUv2-Depth dataset with us, and the following evaluation of their method was obtained based on the depth map predictions they provided us with. All other mentioned methods have released their predictions online.

Fair comparison is ensured by performing evaluation of each method solely using its associated depth map predictions and one single evaluation code.

**Occluding contours location accuracy.** To evaluate occluding contours location accuracy, we follow the work of Koch *et al.* [16] as they proposed an experimental method for such evaluation. Since it is fundamental to examine whether predicted depths maps are able to represent all occluding contours as depth discontinuities in an accurate

Method	Evaluated on full NYUv2-Depth							Evaluated on our NYUv2-OC			
	Accuracy $\uparrow$ ( $\delta_i = 1.25^{\circ}$ )			Error $\downarrow$				$\epsilon_{DBE}^{acc} \downarrow$ (px) $\{\sigma^-, \sigma^+\}$			
	$\delta_1$	$\delta_2$	$\delta_3$	rel	$\log_{10}$	RMSE (lin)	RMSE (log)	{0.1, 0.2}	{0.01, 0.1}	{0.005, 0.06}	{0.03, 0.05}
Eigen <i>et al.</i> [5] (VGG)	0.766	0.949	0.988	0.195	0.068	0.660	0.217	2.895	3.065	3.199	3.203
Eigen <i>et al.</i> [5] (AlexNet)	0.690*	0.911*	0.977*	0.250*	0.082*	0.755*	0.259*	2.840	3.029	3.202	3.242
Laina <i>et al.</i> [18]	0.818	0.955	0.988	0.170	0.059	0.602	0.200	3.901	4.033	4.116	4.133
Fu <i>et al.</i> [8]	0.850	0.957	0.985	0.150	0.052	0.578	0.194	3.714	3.754	4.040	4.062
Jiao <i>et al.</i> [15]	<b>0.909</b>	<b>0.981</b>	<b>0.995</b>	<b>0.133</b>	<b>0.042</b>	<b>0.401</b>	<b>0.146</b>	6.389*	4.073*	4.179*	4.190*
Ours	<u>0.888</u>	<u>0.979</u>	<u>0.995</u>	<u>0.139</u>	<u>0.047</u>	<u>0.495</u>	<u>0.157</u>	<b>2.272</b>	<b>2.629</b>	<b>3.066</b>	<b>3.152</b>

Table 1: Our final evaluation results. Bold and underlined results indicate first and second place respectively. Asterisks indicate the last place. Numerical results might vary from the original papers, as we evaluated all methods with the same code, using only the authors depth map predictions. Results are evaluated in the center crop proposed by [5] and clipped depth predictions to range [0.7m, 10m].

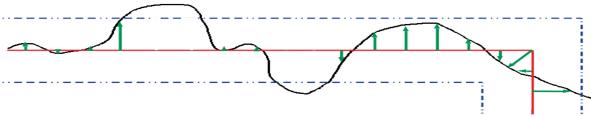


Figure 5: The truncated chamfer distance is computed as the sum Euclidean distances  $E_i$  (in green) between the detected edge  $\hat{Y}_i$  (in black) and the ground truth edge  $Y_i$  (in red). The  $E_i$  above 10 pixels (above the blue dashed line) are ignored.

way, they analyzed occluding contours accuracy performances by detecting edges in predicted and ground truth depth maps and comparing those edges.

Since acquired depth maps in the NYUv2-Depth dataset are especially noisy around occluding boundaries, we manually annotated a subset of the dataset with occluding contours, building our NYUv2-OC dataset, which we used for evaluation. Several samples of our NYU-OC dataset are shown in Fig. 4 and Fig. 7. In order to evaluate the predicted depth maps’  $D$  quality in terms of occluding contours reconstruction, binary edges  $\hat{Y}$  are first extracted from  $\hat{D}$  with a Canny detector.<sup>2</sup> They are then compared to the ground truth annotated binary edges  $Y$  from our NYU-OC dataset by measuring the a *Truncated Chamfer Distance* (TCD). Specifically, for each pixel  $\hat{Y}_i$  of  $\hat{Y}$  we compute its euclidean distance  $E_i$  to the closest edge pixel  $\hat{Y}_j = 1$ . If the distance between  $\hat{Y}_i$  and  $\hat{Y}_j$  is bigger than 10 pixels we set  $e_i$  to 0 in order to evaluate predicted edges only around the ground truth edges as seen in Fig. 5. This is done efficiently using *Euclidean Distance Transform* on  $Y$ . The depth boundary edge (DBE) accuracy is then computed as the mean TCD over detected edges  $\hat{Y}_i = 1$ :

$$\epsilon_{DBE}^{acc} = \frac{1}{\sum_i \hat{Y}_i} \sum_i E_i \cdot \hat{Y}_i, \quad (8)$$

<sup>2</sup>Edges are extracted from depth maps with normalized dynamic range.

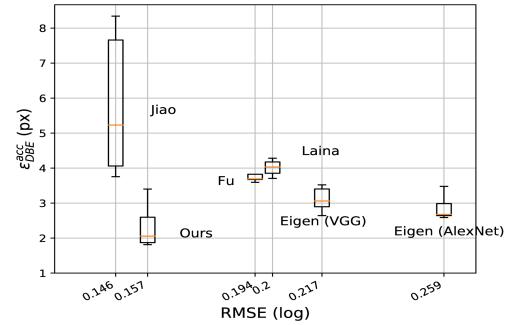


Figure 6: Our method outperforms state-of-the-art in terms of trade-off between global depth reconstruction error and occluding boundary accuracy.

We compare our method against state-of-the-art depth estimation methods using this metric and different Canny parameters. Evaluation results are shown in Table 1: We outperform all state-of-the-art methods on occluding contours accuracy, while being a competitive second best on standard depth estimation evaluation metrics.

Since the detected edges in  $\hat{Y}$  are highly sensitive to the edge detector’s parameters (see Fig.4), we evaluate the DBE accuracy  $\epsilon_{DBE}^{acc}$  using many random combinations of threshold parameters  $\sigma^+$  and  $\sigma^-$  of the Canny edge detector. The results are shown in Fig. 6.

### 4.3. Ablation Study

To prove the impact of our geometry consensus terms, we performed an ablation study to analyze the contribution of training with synthetic and real data, as well as our novel geometry consensus terms. Evaluation of different models on our NYUv2-OC dataset are shown in Table 2, confirming their contribution to both improved depth reconstruction results over the whole NYUv2-Depth dataset and occluding contours accuracy.

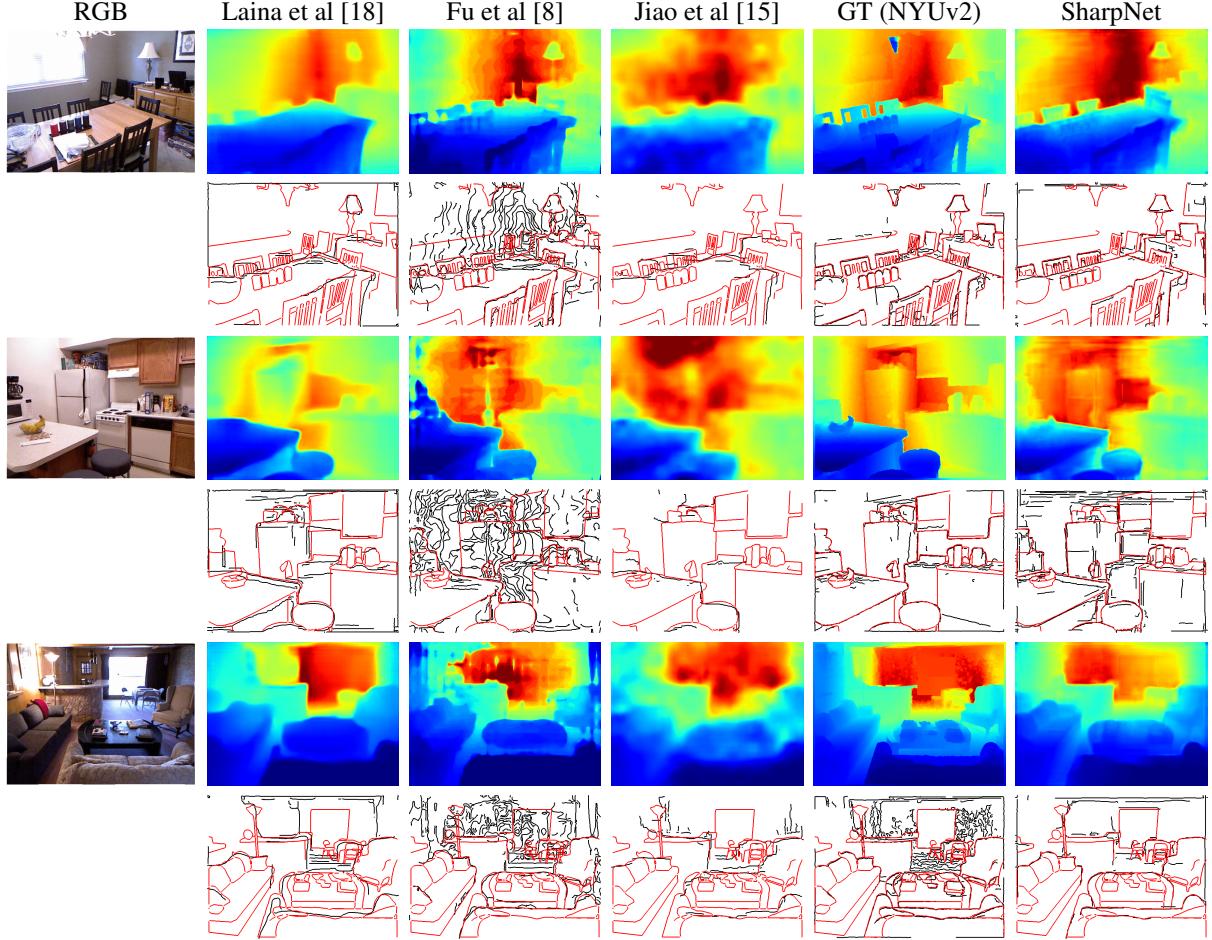


Figure 7: Several examples of images from our NYUv2-OC dataset and their associated depth map estimate for different methods. The second row for each image shows the in black the detected edges on those estimates using a Canny edge detector (in black) with  $\sigma^- = 0.03$  and  $\sigma^+ = 0.05$ , overlaid on our manually annotated ground truth in red. Our SharpNet method not only creates sharper occluding contours, leading to less spurious and erroneous contours than with [8] the Kinect-v1 depth-map; it also leads to much better located edges than other methods.

Method	Training Dataset	$RMSE_{log}$	$\epsilon_{DBE}^{acc} \downarrow (px) \{ \sigma^-, \sigma^+ \}$			
			$\{0.1, 0.2\} \{0.01, 0.1\} \{0.005, 0.06\} \{0.03, 0.05\}$	$\{0.1, 0.2\} \{0.01, 0.1\} \{0.005, 0.06\} \{0.03, 0.05\}$	$\{0.1, 0.2\} \{0.01, 0.1\} \{0.005, 0.06\} \{0.03, 0.05\}$	$\{0.1, 0.2\} \{0.01, 0.1\} \{0.005, 0.06\} \{0.03, 0.05\}$
w/o consensus	PBRS	0.304*	2.321	2.751*	3.298*	3.380*
w/o consensus	PBRS	0.262	<b>2.046</b>	<b>2.332</b>	<b>2.574</b>	<b>2.645</b>
w/o consensus	NYUv2	<u>0.163</u>	2.600*	2.638	3.127	3.182
w/ consensus	PBRS + NYUv2	<u>0.157</u>	<u>2.272</u>	<u>2.629</u>	<u>3.066</u>	<u>3.152</u>

Table 2: Our added geometry consensus terms brings a significant performance boost by guiding the depth towards learning accurate occluding contours and it also helps keeping a good trade-off between occluding contours accuracy and depth reconstruction during the necessary fine-tuning on real RGB-D data.  $RMSE_{log}$  is computed over the full NYUv2-Depth dataset. Notations of Table. 1 are used here.

## 5. Conclusion

In this paper, we show that our SharpNet method is able to achieve competitive depth reconstruction from a single RGB image with particular attention to occluding contours thanks to geometry consensus terms introduced during multi-task training. Our high-quality depth estimation which yields high accuracy occluding contours reconstruction allows for realistic integration of virtual objects in real-time augmented reality as we achieve 150 fps inference speed. We show the superiority of our SharpNet over state-of-the-art by introducing a first version of our new NYUv2-OC occluding contours dataset, which we plan to extend in future work. As by-products of our approach, high-quality normals and contours predictions can also be a useful representation for other computer vision tasks.

## References

- [1] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.
- [2] Y. Cao, T. Zhao, K. Xian, C. Shen, and Z. Cao. Monocular Depth Estimation with Augmented Ordinal Depth Relationships. *IEEE Transactions on Image Processing*, 2018.
- [3] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.
- [4] P. Dollár and C. L. Zitnick. Fast Edge Detection Using Structured Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, August 2015.
- [5] D. Eigen and R. Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. In *International Conference on Computer Vision*, pages 2650–2658, 2015.
- [6] 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.
- [7] X. Fei, A. Wang, and S. Soatto. Geo-Supervised Visual Depth Prediction. *IEEE Robotics and Automation Letters*, 4:1661–1668, 2018.
- [8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2018.
- [9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research*, 2013.
- [10] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In *Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced Computer Vision With Microsoft Kinect Sensor: A Review. *IEEE Transactions on Cybernetics*, 2013.
- [12] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [13] P. Heise, S. Klose, B. Jensen, and A. Knoll. PM-Huber: PatchMatch with Huber Regularization for Stereo Matching. In *IEEE International Conference on Computer Vision*, pages 2360–2367, 2013.
- [14] X. Jiang, M. L. Pendu, and C. Guillemot. Depth Estimation with Occlusion Handling from a Sparse Set of Light Field Views. *IEEE International Conference on Image Processing*, pages 634–638, 2018.
- [15] J. Jiao, Y. Cao, Y. Song, and R. W. H. Lau. Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss. In *European Conference on Computer Vision*, 2018.
- [16] T. Koch, L. Liebel, F. Fraundorfer, and M. Körner. Evaluation of CNN-Based Single-Image Depth Estimation Methods. In *European Conference on Computer Vision*, 2018.
- [17] L. Ladicky, J. Shi, and M. Pollefeys. Pulling Things Out of Perspective. In *Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014.
- [18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper Depth Prediction with Fully Convolutional Residual Networks. In *International Conference on 3D Vision*, pages 239–248, 2016.
- [19] J. Li, R. Klein, and A. Yao. A Two-Streamed Network for Estimating Fine-Scaled Depth Maps from Single RGB Images. In *International Conference on Computer Vision*, pages 3392–3400, 2017.
- [20] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? 2017.
- [21] B. Owen. A Robust Hybrid of Lasso and Ridge Regression. *Contemp. Math.*, 443, 01 2007.
- [22] M. Poggi, F. Tosi, and S. Mattoccia. Learning Monocular Depth Estimation with Unsupervised Trinocular Assumptions. In *International Conference on 3D Vision*, pages 324–333, 2018.
- [23] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric Neural Network for Joint Depth and Surface Normal Estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.
- [24] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, abs/1505.04597, 2015.
- [25] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision*, 2012.
- [26] S. Song, S. P. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition*, pages 567–576, June 2015.
- [27] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.
- [28] Q. Teng, Y. Chen, and C. Huang. Occlusion-Aware Unsupervised Learning of Monocular Depth, Optical Flow and Camera Pose with Geometric Constraints. *Future Internet*, 10:92, 09 2018.
- [29] G. Wang, X. Liang, and F. W. B. Li. DOOBNet: Deep Object Occlusion Boundary Detection from an Image. *CoRR*, abs/1806.03772, 2018.
- [30] P. Wang, X. Shen, B. Russell, S. Cohen, B. Price, and A. L. Yuille. SURGE: Surface Regularized Geometry Estimation from a Single Image. In *Advances in Neural Information Processing Systems*, pages 172–180, 2016.
- [31] T. Wang, A. A. Efros, and R. Ramamoorthi. Depth Estimation with Occlusion Modeling Using Light-Field Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2170–2181, Nov 2016.
- [32] Z. Wu and L. Li. A Line-Integration Based Method for Depth Recovery from Surface Normals. *Computer Vision, Graphics, and Image Processing*, 43(1):53–66, 1988.
- [33] J. Xie, R. Girshick, and A. Farhadi. Deep3D: Fully Automatic 2D-To-3d Video Conversion with Deep Convolutional Neural Networks. In *European Conference on Computer Vision*, pages 842–857, 2016.

- [34] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-Scale Continuous CRFs as Sequential Deep Networks for Monocular Depth Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2017.
- [35] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 225–234, 2018.
- [36] Z. Yang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised Learning of Geometry From Videos With Edge-Aware Depth-Normal Consistency. In *AAAI*, 2018.
- [37] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.
- [38] Y. Zhang and T. Funkhouser. Deep depth completion of a single rgbd image. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [39] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-Based Rendering for Indoor Scene Understanding Using Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.
- [40] L. Zwald and S. Lambert-Lacroix. The Berhu Penalty and the Grouped Effect.