Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries

Junjie Hu[1,2], Mete Ozay[1], Yan Zhang[1,2], Takayuki Okatani[1,2]

[1] Graduate School of Information Sciences, Tohoku University, Japan
[2] Center for Advanced Intelligence project, RIKEN, Japan
{junjie.hu, mozay, zhang, okatani}@vision.is.tohoku.ac.jp

Abstract. We revisit the problem of estimating depth of a scene from its single RGB image. Despite the recent success of deep learning based methods, we show that there is still room for improvement in two aspects by training a deep network consisting of two sub-networks; a base network for providing an initial depth estimate, and a refinement network for refining it. First, spatial resolution of the estimated depth maps can be improved using skip connections among the sub-networks which are trained in a sequential fashion. Second, we can improve estimation accuracy of boundaries of objects in scenes by employing the proposed loss functions using depth gradients. Experimental results show that the proposed network and methods improve depth estimation performance of baseline networks, particularly for reconstruction of small objects and refinement of distortion of edges, and outperform the state-of-the-art methods on benchmark datasets.

Keywords: single image depth estimation, fully convolutional network, skip connection, natural range image statistics, gradient-based loss

1 Introduction

In recent years, convolutional neural networks (CNNs) have been applied to all sorts of computer vision tasks, for most of which we have seen great successes. The "formula" for success has been to find a suitable use and design of CNNs for a target task, provided that a sufficient number of training samples are available. This formula can be broken down to the following three factors: i) architectural design of CNNs, ii) choice of a loss function, and iii) a training method.

The importance of (i) is needless to say. Although the building blocks are the same for different tasks, architectural design of a network tends to determine success or failure, which has almost an infinite number of degrees of freedom. The researchers have also paid attention to (ii). Taking image restoration tasks for instance, such as image denoising, super-resolution, inpainting etc., perceptual loss [1, 2], contextual loss [2] and Wasserstein loss [3] are proposed. For (iii), although the use of SGD and its variants in an end-to-end fashion have been the most preferred, researchers invented different methods for different tasks to ease difficulties with training deep networks, such as addition of a loss to intermediate layers [4], step-by-step training from smaller to larger

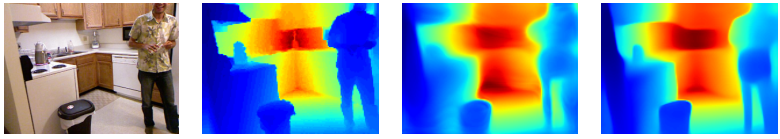
networks [5]; besides, the use of adversarial training now seems to be a standard method particularly when the goal is to generate realistic images [6, 7].

Since the first work by Eigen et al. [8], there have been a number of studies [9, 10, 11, 12, 13, 14] to use CNNs to estimate the dense depth map of a scene from its single image. In this paper, we aim to improve the accuracy level of the previous methods by reconsidering the problem from the aforementioned three points of view. Although attaining maximum accuracy may be a common goal of previous studies, we point out that there is room for improvement in the following two aspects; a) spatial resolution of estimated depth maps, and b) accuracy of object boundaries recovered in them. It should be noted that these two are not exclusive to each other.

Depth maps estimated by previous methods tend to have (sometimes much) lower resolution than input images. Its cause can be traced back to a series of downsampling operations performed in CNNs. Downsampling (or pooling) is a component of CNNs that seems to be indispensable for many tasks ranging from object category classification to key point detection. Semantic segmentation, which is one of these tasks, have some in common with depth estimation. A large amount of studies have been conducted for semantic segmentation, in which the primary concern is to overcome the loss of resolution due to downsampling. Researchers have proposed several approaches, i.e., skip connection [15, 16, 17], dilated convolution [18, 19, 20, 21], pyramid pooling [22], and learned deconvolution [23, 24]. Among these, skip connections have been most universally employed and proven to be effective not just for semantic segmentation but for many other tasks, such as key point detection, optical flow estimation, etc. For depth estimation, researchers have proposed a multi-scale approach of using two CNNs for coarse reconstruction and fine scale refinement [8], or an improvement over standard up-convolution (i.e., upsampling followed by convolution) named up-projection [13]. However, it is not clear how effective skip connection is for depth estimation.

In this paper, we show that the employment of skip connection is indeed effective for depth estimation, but the improvement is achieved by an implementation different from the CNNs employed in the aforementioned tasks. To be specific, we use two CNNs: a base network for estimating a depth map with a certain resolution, and a refinement network for refining it to obtain more accurate depth map with finer details. We connect the outputs of the lower layers of the base network to the input of the refinement network. The connections may be regarded as skip connections, if we regard the two networks as a single network.

The other aspect which we consider for improvement is with respect to estimation accuracy of object boundaries in depth maps. In the past, the statistics of range images of natural scenes (called depth maps in this paper) have been studied [25, 26, 27]. In [25], the authors analyzed distribution of three-dimensional points using co-occurrence statistics, and two- and three-dimensional joint distributions of Haar filter reactions. The results indicate that the range images have much simpler structures than optical images, which is also known as the "random collage model", that is, the world can be broken down into piecewise smooth regions that depend little on each other and sharp discontinuities in between them. The sharp discontinuities typically emerge at the occluding boundaries of objects in scenes, which form step edges in depth maps. This structure is a key property of depth maps (of natural scenes). However, previous methods often



(a) (b) (c) (d)

Fig. 1. Effects of loss functions on depth estimation. (a) Input image. (b) Ground truth depth map. (c) Prediction of a CNN trained with the loss of difference in depth. (ℓ_1 norm is used.) (d) Prediction of an identical CNN trained with a combined loss of differences in depth and gradient.

fail to recover such edges correctly; the recovered edges tend to be spatially distorted or sometimes blurry even though they are in fact very sharp.

In this work, we point out that this is partly attributable to the loss function used to train CNNs, as shown in Fig. 1. Many studies employ the sum of differences in depth between an estimated depth map and its ground truth for their loss function (with different norms, i.e., ℓ_2 loss [8, 10, 28], ℓ_1 loss [29, 30], and a robustified $berhu$ loss [13]). We show that this loss is insensitive to errors emerging at step edges, such as shift in their positions and difference between sharp and blurry edges. We then propose to use two additional loss functions both based on gradients, difference in gradients (l_{grad}) and difference in normals to scene surfaces (l_{normal}) between an estimated map and its ground truth. We describe that the three loss functions are complementary with each other and show that their combination contributes to improve accuracy of depth map estimation. We show experimental results that confirms effectiveness of our approach.

We also found that careful consideration is necessary for the factor (iii) for successful applications of CNNs mentioned at the beginning, i.e., how to train networks. As mentioned above, our model consists of a base network and a refinement network. For the base network, we use a modified version of the CNN proposed by Laina et al. [13], where the ResNet [31] with 50 layers used in their original model is replaced by DenseNet [32] with 169 layers for its efficiency. It is possible to regard these two networks as a single integrated network and train it in an end-to-end fashion. However, this method did not work well in our experiments, probably due to the increased number of parameters and lack of a correspondingly large number of training samples. We found that a sequential training method provides satisfactory results, in which the base network is first trained until convergence, and then the refinement network is trained while the parameters of the base network are fixed.

2 Proposed Method

This section presents our method. We will describe the network architecture, loss function, and training strategy in this order.

2.1 Network Architecture

Our network is a hybrid of the multi-scale approach of Eigen et al. [8] and the approach of Laina et al. that uses novel layers performing upsampling (plus convolution) on top of

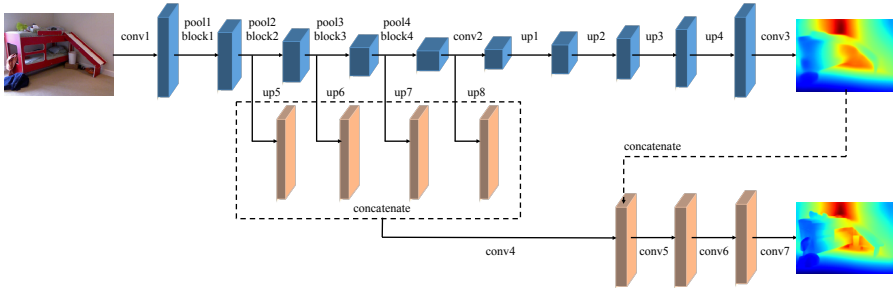


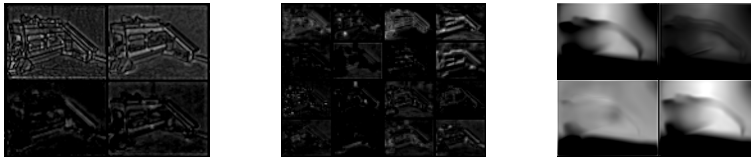
Fig. 2. A diagram of the proposed network architecture. The base network is marked by blue and the refinement network is marked by orange. The ♯ used in "conv♯" denotes the index of the corresponding convolutional layer, and so on. "block" denotes the DenseNet block consisting of multiple convolutional layers. "up" denotes a up-projection layer introduced in [13]. Batch normalization and ReLU nonlinearity are applied to the output of each convolutional layer except conv3 and conv7.

a ResNet-based fully convolutional network (FCN) [13]. Following Eigen et al. [8], we consider a model consisting of two sub-networks, i.e., a base network and a refinement network, as shown in Fig. 2. The base network is built upon the network proposed by Laina et al. [13]. Given a two-dimensional RGB image of a scene, the base network outputs an initial estimate of its depth map. The refinement network refines this estimate into a more accurate estimate with finer details.

The base network is a moderately revised version of the network of Laina et al. [13]. To be specific, we make two modifications. One is that we use a DenseNet with 169 layers instead of a ResNet with 50 layers in the model. We use a DenseNet due to its better balance of model size and performance; it tends to be smaller and achieve same or better level accuracy in many tasks compared to a ResNet. The network of Laina et al. has 63.6M parameters, whereas our base network has 38.9M parameters. The second modification is employment of a gradient-based loss (i.e., l_{grad} of (2)), which contributes to a certain amount of improvement achieved by our model, as will be explained later.

Our proposed base network differs from the model of Eigen et al. [8], where their global coarse scale network, corresponding to the base network in our model, outputs only a depth map with very coarse resolution. Owing to the use of the powerful DenseNet/ResNet along with the up-projection operation that improves up-convolution [13], our base network can provide estimates at the state-of-the-art level of accuracy. Thus, our method aims to improve already fairly accurate estimates to achieve even a higher level of accuracy by using the refinement network.

Despite its relatively good performance, the outputs of the base network tend to lose spatial resolution. We use the refinement network to recover the resolution, where the outputs of the lower layers of the base network are utilized. Fig. 7 shows examples of low layer outputs (block1 and block2) and high layer outputs (up4) of the base network. It is observed that the lower layer outputs maintain fine details of object shapes, whereas



(a) (b) (c)

Fig. 3. Visualization of outputs of different layers of the base network for the input image shown in Fig.2. Selected channels of (a) block1, (b) block2, and (c) up4.

Table 1. Sizes of output features, and input/output channels of each layer used in our model, when it is configured for input/output sizes of samples belonging to the NYU-Depth V2 dataset. Left: the base network. Right: the refinement network.

Layer	Output Size	In/C	Out/C
conv1	114×152	3	64
pool1, block1	57×76	64	128
pool2, block2	28×38	128	256
pool3, block3	14×19	256	640
pool4, block4	7×9	640	1664
conv2	7×9	1664	832
up1	14×19	832	416
up2	28×38	416	208
up3	57×76	208	104
up4	114×152	104	52
conv3	114×152	52	1

Layer	Output Size	In/C	Out/C
up5	114×152	128	16
up6	114×152	256	16
up7	114×152	640	16
up8	114×152	1664	16
conv4	114×152	64	64
conv5	114×152	65	65
conv6	114×152	65	65
conv7	114×152	65	1

their shapes are distorted and blurry in the higher layer outputs despite the employment of the up-projection improving up-convolution [13].

The refinement network takes two sources of inputs. One is the final output of the base network (the initial estimate of the dense map), and the other is a set of the layer outputs of four DenseNet blocks, as shown in Fig.2. The outputs of the four DenseNet blocks are upsampled by $\times 2$, 4, 8, and 16, respectively, so as to have the same size as the final output. Their outputs are of 16 channels.They are concatenated in their channels, and inputted to the first layer of the refinement network. The output of the base network is inputted to the second layer by channel-wise concatenating it with the output of the first layer. The refinement network has only 2.5M parameters, and thus the entire model has 41.5M parameters in total, which is still lower than 63.6M parameters of the model proposed in [13].

2.2 Loss Functions for the Proposed Two Sub-networks

Let l_{depth} be the sum of the ℓ_1 norm $\|\cdot\|_1$ over all pixels ($i = 1, 2, \ldots, n$) of the difference between the i^{th} depth estimate d_i and its ground truth g_i by

$$l_{\text{depth}} = \frac{1}{n}\sum_{i=1}^{n} \|d_i - g_i\|_1. \tag{1}$$

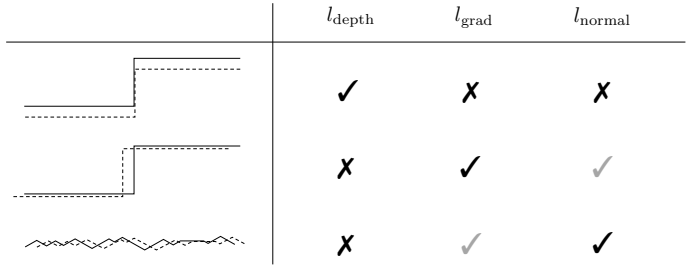


Fig. 4. The three loss functions have orthogonal sensitivities to different types of errors of estimated depth maps. The solid and dotted lines depicted in the first column indicate two depth maps under comparison, where they are represented as one-dimensional depth images for the sake of explanation, and the vertical axis is depth and the horizontal axis is, say, the x axis of the images.

This loss (or its ℓ_2 norm or robustified version) is widely employed in previous studies. However, we point out that for a step edge structure of depth, while this loss is sensitive to overall shifts in depth direction, it is comparatively insensitive to shifts in xy directions, as shown in the top and second rows of Fig. 4. It is similarly insensitive also to edges becoming blurry.

The statistics of natural range images indicate that natural scenes consist of a lot of such step edge structures [25], which can easily be confirmed from examples of ground truth depth maps in various datasets. We think that this insensitivity to small errors around edges must be a major reason for the phenomenon that the edges in depth maps estimated by CNNs trained using this loss, tend to be distorted or blurry.

Thus, it is necessary to penalize such errors around edges more. For this purpose, we consider the following loss function of the gradients of depth:

$$l_{\text{grad}} = \frac{1}{n}\sum_{i=1}^{n}\left(\|\nabla_x(d_i) - \nabla_x(g_i)\|_1 + \|\nabla_y(d_i) - \nabla_y(g_i)\|_1\right), \tag{2}$$

where $\nabla_x(d_i)$ is the spatial derivative computed at the i^{th} pixel of $d(x, y)$ with respect to x, and so on. This loss is sensitive to the shift of edges in xy directions, as shown in Fig. 4. Note that the proposed two loss functions l_{depth} and l_{grad} work in a complementary manner for different types of errors. Thus, we use the (weighted) sum of l_{depth} and l_{grad} to train our networks. It should be noted that l_{depth} is anisotropic. A more general loss that is isotropic is

$$\tilde{l}_{\text{grad}} = \frac{1}{n}\sum_{i=1}^{n}\left(\left(\nabla_x(d_i) - \nabla_x(g_i)\right)^2 + \left(\nabla_y(d_i) - \nabla_y(g_i)\right)^2\right)^{1/2}. \tag{3}$$

We found in our experiments that the anisotropic l_{grad} yields better results with smaller computational cost than $\hat{l}_{\text{grad}}$. Therefore, we use the sum of the two loss functions to train the base network by

$$L_{\text{base}} = l_{\text{depth}} + \lambda l_{\text{grad}}, \tag{4}$$

where $\lambda \in \mathbb{R}$ is a loss weighting coefficient.

Depth maps of natural scenes can roughly be modeled by a limited number of smooth surfaces and step edges in between them, according to the statistics of natural range images [25]. For instance, depth will often be discontinuous at the boundary of an object. Errors around such strong edges are well penalized by l_{grad}. However, since depth differences at such occluding boundaries of objects can sometimes be very large, we must choose a modest (i.e., not very large) weight $\lambda > 0$ on l_{grad}. (Note that l_{grad} is not upper bounded.) Then, the term $\lambda l_{\mathrm{grad}}$ cannot penalize small structural errors such as those of high-frequency undulation of a surface, as shown in the bottom row of Fig. 4.

To deal with such small depth structures and further improve fine details of depth maps, we consider yet another loss for training of the refinement network, which measures accuracy of the normal to the surface of an estimated depth map with respect to its ground truth. Denoting the surface normal of an estimated depth map and its ground truth by $n_i^d \equiv [-\nabla_x(d_i), -\nabla_y(d_i), 1]^\top$ and $n_i^g \equiv [-\nabla_x(g_i), -\nabla_y(g_i), 1]^\top$ respectively, we define the following loss measuring the difference between the two normals by

$$l_{\mathrm{normal}} = \frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{\langle n_i^d, n_i^g\rangle}{\sqrt{\langle n_i^d, n_i^d\rangle}\sqrt{\langle n_i^g, n_i^g\rangle}}\right), \tag{5}$$

where $\langle\cdot,\cdot\rangle$ indicates the inner product of vectors. Although this loss is also computed from depth gradients, it measures the angle between two surface normals, and thus is sensitive to small depth structures illustrated in the bottom row of Fig. 4. Thus, we can say that l_{normal} is also complementary with the other two losses. Finally, we define the loss for the refinement net by

$$L_{\mathrm{refine}} = l_{\mathrm{depth}} + \lambda l_{\mathrm{grad}} + \mu l_{\mathrm{normal}}, \tag{6}$$

where $\mu \in \mathbb{R}$ is a weighting coefficient.

2.3 Training Methods

As we mentioned earlier, we train the base network and the refinement network individually in a sequential fashion. We first train the base network until convergence is achieved using the loss L_{base}. Next, freezing all the parameters of the base network, we train the refinement network with the loss L_{refine} until convergence.

In the previous studies [8, 13, 14, 29, 33], it is shown that data argumentation helps improve accuracy as well as avoid over-fitting. We employ the following data augmentation methods, which are individually applied to each sample (an RGB image and the corresponding depth map) in an online manner:

- Scale: For a random number $s \in [1, 1.5]$, the RGB image and the true depth map are scaled up by s, and depth values are divided by s.
- Flip: The RGB and the depth image are both horizontally flipped with 0.5 probability.
- Rotation: The RGB and the depth image are both rotated by a random degree $r \in [-5, 5]$.

– Color Jitter: Brightness, contrast, and saturation values of the RGB image are randomly scaled by $c \in [0.6, 1.4]$.

3 Experimental Results

3.1 Accuracy Measures for Depth Estimation

We evaluate our model on the indoor NYU-Depth V2 and the outdoor KITTI datasets. Denoting the total number of (valid) pixels used in all evaluated images by T, we use the following accuracy measures that are commonly employed in previous studies:

– Root mean squared error (RMS): $\sqrt{\frac{1}{T}\sum_{i=1}^{T}(d_i - g_i)^2}$.

– Mean relative error (REL): $\frac{1}{T}\sum_{i=1}^{T}\frac{\|d_i - g_i\|_1}{g_i}$.

– Thresholded accuracy: Percentage of d_i, such that $\max\left(\frac{d_i}{g_i}, \frac{g_i}{d_i}\right) = \delta <$ threshold.

These popular measures enable us to compare methods from multiple aspects to evaluate depth accuracy. However, a combination of these measures still has limitation. We argue that these measures are not good at detecting spatial distortion of object edges, because of the same reason as we discussed in Sec. 2.2. For instance, the method of Ma and Karaman [29], which leverages known depths at a few scene points to improve depth estimation, does outperform others that do not use such additional information, if we use the above measures. However, the outputs of their method tend to have spatially distorted or blurry object edges; local structures are often missing. The same tendency can be observed for others, particularly recent ones; estimated depth maps showing low RMS errors tend to have apparent errors of this type.

Edge accuracy: In order to more properly evaluate accuracy of estimated depth maps, we propose an additional measure to gauge positional errors of edges which will be overlooked by the above measures. For this purpose, we apply the Sobel operator [34] to both of the estimated and the true depth maps, and then apply a threshold to them to identify pixels which satisfy $\sqrt{f_x(i)^2 + f_y(i)^2} >$ (threshold) by 'pixels on edges', where f_x and f_y are 3×3 horizontal and vertical Sobel operators, respectively. Assuming those of the true depth map to be true, we measure precision, recall and F1 score for those of the estimated map. We used three different thresholds: 0.25, 0.5 and 1 for the NYU-Depth V2; and 1, 5, and 10 for the KITTI.

As explained earlier, we train the two networks sequentially. The DenseNet built in the base network is initialized by a model pretained with the ImageNet dataset [35]. The other layers in the base network and all the layers of the refinement network are randomly initialized. To train both networks, we use Adam optimizer with learning rate of 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 0.0001. The weight λ of l_{grad} in L_{base} and L_{refine} is set as $\lambda = 10$ for the NYU-Depth V2 dataset and $\lambda = 1$ for the KITTI, where differences in the range of depths of the two datasets are taken into account. The weight μ of l_{normal} in L_{refine} is set to 1 throughout all the experiments. We conducted all the experiments using PyTorch [36] with batch size of 16.

Table 2. Comparisons of different methods including our base network and refinement network with different loss functions on the NYU-Depth V2 dataset. The methods marked by * use partially known depths, and those with ** employ joint task learning.

Method	RMS	REL	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [8]	0.907	0.215	0.611	0.887	0.971
Li et al. [10]	0.821	0.232	0.621	0.886	0.968
Liu et al. [9]	0.824	0.230	0.614	0.883	0.971
Chakrabarti et al. [38]	0.620	0.149	0.806	0.958	0.987
Cao et al. [11]	0.819	0.232	0.646	0.892	0.968
Li et al. [39]	0.635	0.143	0.788	0.958	0.991
Xu et al. [14]	0.586	**0.121**	0.811	0.954	0.987
Ma and Karaman [29]	(-)	0.143	0.810	0.959	0.989
Laina et al. [13]	0.573	0.127	0.811	0.953	0.988
Base net w/ l_{depth}	0.596	0.139	0.817	0.958	0.989
Base net w/ $l_{\text{depth}} + \lambda l_{\text{grad}}$	0.581	0.133	0.832	0.964	0.990
Refinement net w/ $l_{\text{depth}} + \lambda l_{\text{grad}}$	0.578	0.132	0.834	0.966	0.990
Refinement net w/ $l_{\text{depth}} + \lambda l_{\text{grad}} + \mu l_{\text{normal}}$	**0.568**	0.132	**0.838**	**0.966**	**0.990**
Chen et al. [12]*	1.100	0.340	-	-	-
Ma and Karaman [29]*	(-)	0.044	0.971	0.994	0.998
Ladicky et al. [40]**	-	-	0.542	0.829	0.941
Eigen and Fergus [33]**	0.641	0.158	0.769	0.950	0.988
Wang et al. [41]**	0.745	0.220	0.605	0.890	0.970
Dharmasiri et al. [42]**	0.624	0.156	0.776	0.953	0.989

3.2 The NYU-Depth V2 Dataset

This dataset consists of a variety of indoor scenes, and is the most widely used for the task of single view depth prediction [37]. We follow the same procedure as the one employed in the previous studies [8, 13, 29]. We use the official splits for 464 scenes, i.e., 249 scenes for training and 215 scenes for testing. For training, we use the official toolbox to extract RGB images and depth maps from the raw data, and fill in missing pixels in the depth maps to generate ground truths. This results in approximately 50K unique pairs of an image and a depth map, each of which contains 640×480 pixels. We then reduce the sizes of images and depth maps. We downsample images to 320×240 pixels using bilinear interpolation, and then crop their central parts to obtain images with 304×228 pixels. These images are inputted to our network. The depth maps are downsampled to 114×152 to fit the size of output. For testing, following the previous studies, we use the same small subset of 654 samples. We train the base network for 10 epochs and the refinement network for 60 epochs.

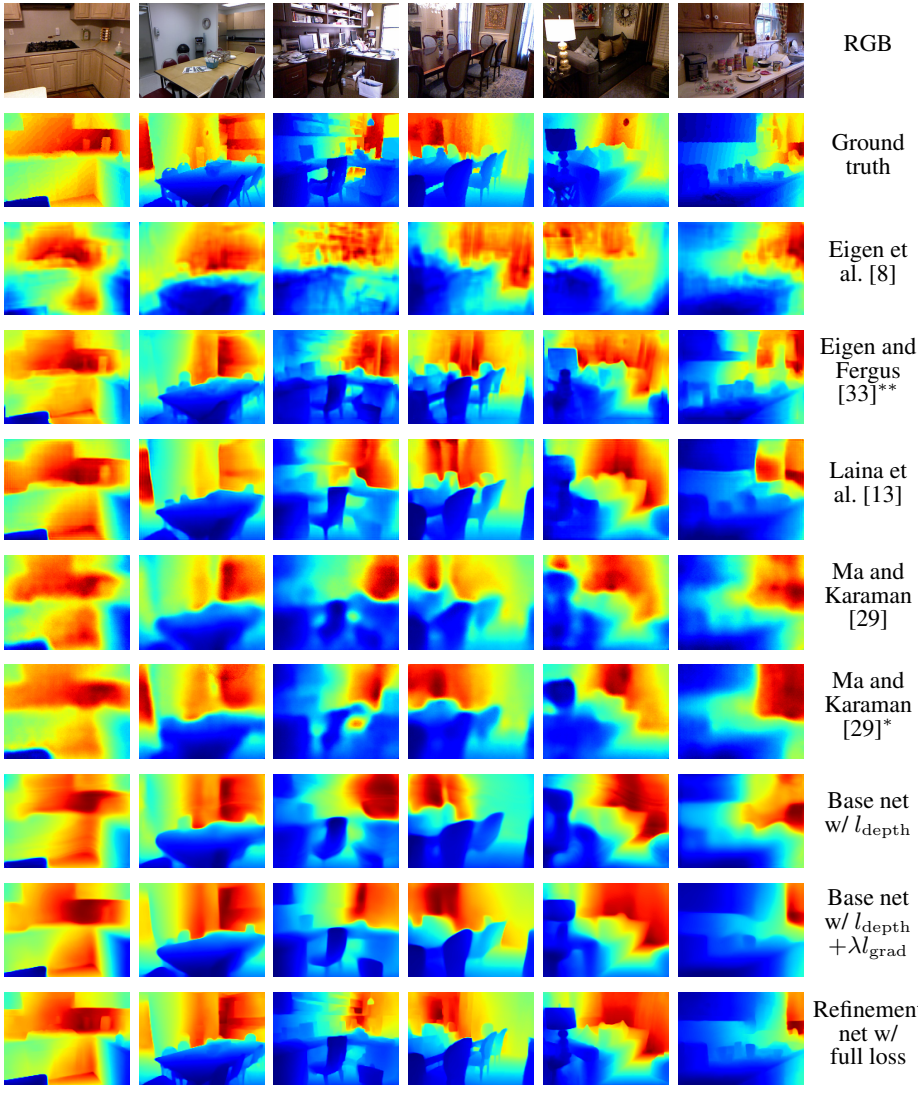


Fig. 5. Results of different methods for six images. From the first to the last row; input RGB images, ground truth depth map, Eigen et al. [8], Eigen and Fergus [33], Laina et al. [13], Ma and Karaman [29] with and without known depths at specified 200 pixels, our base network trained with l_{depth} and $l_{\mathrm{depth}}+\lambda l_{\mathrm{grad}}$, and our refinement network trained with the full loss function. The methods marked by * use partially known depths, and those with ** employ joint task learning.

Table 2[1] shows the results of our method together with those of existing methods on the three basic measures. For the sake of references, the table includes the methods that

[1] RMS values given in [29] are omitted here (displayed as (-)), because we found an error in the authors' code, and believe that the numbers reported in their paper are miscalculated.

Table 3. Edge accuracy for the NYU-Depth V2. See text for details. The method marked by ** employs joint task learning.

		Precision	Recall	F1 score
>0.25	Eigen and Fergus [33]**	0.550	0.486	0.505
	Laina et al. [13]	0.601	0.595	0.592
	Base net	0.725	0.549	0.615
	Refinement net	0.757	0.611	0.669
>0.5	Eigen and Fergus [33]**	0.589	0.461	0.504
	Laina et al. [13]	0.584	0.510	0.537
	Base net	0.719	0.486	0.572
	Refinement net	0.748	0.576	0.645
>1	Eigen and Fergus [33]**	0.732	0.489	0.575
	Laina et al. [13]	0.663	0.523	0.577
	Base net	0.776	0.521	0.616
	Refinement net	0.789	0.600	0.675

use additional information other than the input RGB images; those with * use relative depth between pairs or partially known depths [12, 29], and those with ** employ joint task learning [10, 33, 40, 41, 42]. For the method of Ma and Karaman [29], we show two results that are obtained from single RGB images alone and with partially known depths (200 pixels). The methods denoted without a superscript [8, 9, 11, 13, 14, 38, 39] and ours should be able to be compared in an equal condition.

It is first observed from Table 2 that our base network with l_{depth} yields a result similar to that of Laina et al. [13], and Ma and Karaman [29]. This is no wonder since the base network is similar to their networks, and l_{depth} is the same or similar to their loss functions. Note also that the employment of a DenseNet in our base network does not contribute to improvement of accuracy by itself. It is then observed that the addition of l_{grad} to the loss provides non-negligible accuracy improvements in the three measures. As a result, this configuration (i.e., the base network trained with L_{base}) already performs better than any of the existing methods for all $\delta(\delta < 1.25^k, k = 1, 2, 3)$. It is seen that the use of the refinement network with the same loss L_{base} further improves estimation accuracy, but to only a moderate degree. The refinement network provides maximum improvement when we include l_{normal} in the loss, which outperforms existing methods in RMS and all δ.

The improvements attained by the refinement network might not appear very large while using the classical measures shown in Table 2. However, while visually comparing their estimated depth maps side by side, it can be observed that the refinement network does provide significant improvements especially in fine details of objects in scenes. Fig. 5 shows the depth maps estimated by different methods[2] including our method in different configurations. It is confirmed that objects in the scenes

[2] For Eigen et al. [8] and Eigen and Fergus [33], we show the results that are made publicly available by the authors. For Laina et al. [13] and Ma et al. [29], we use the authors' code to obtain the results and show them here.

Table 4. Comparisons of accuracy of different methods on the KITTI dataset. We used the same 28 scenes utilized by Eigen et al. [8] for evaluation. The methods marked by * use the stereo pair of images instead of ground truth depth maps to train CNNs to perform single view depth estimation. All the values below are those reported by the authors.

Method	RMS	REL	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [8]	7.156	0.190	0.692	0.899	0.967
Liu et al. [28]	7.046	0.217	0.656	0.881	0.968
Liu et al. [45]	6.427	0.203	0.684	0.894	0.965
Garg et al. [46]*	5.104	0.169	0.740	0.904	0.962
Godard et al. [43]*	4.935	0.114	0.861	0.949	0.976
Kuznietsov et al. [47]*	4.627	-	0.856	0.960	0.986
Base net	4.706	0.110	0.869	0.967	0.988
Refinement net	**4.604**	**0.107**	**0.875**	**0.970**	**0.990**

gradually recover finer details in the order of the base network with l_{depth}, that with $l_{\mathrm{depth}} + \lambda l_{\mathrm{grad}}$, and the refinement network with the full loss $l_{\mathrm{depth}} + \lambda l_{\mathrm{grad}} + \mu l_{\mathrm{normal}}$; the differences in between the three are not small. Compared with existing methods, we can conclude that, overall, the refinement net trained with the full loss more correctly recovers object edges and small structures, such as bottles in a kitchen and lamp shades on a desk. This can also be confirmed by the new measure we introduced above, as shown in Table 3. The improvements of the refinement network are seen particularly in recall (and thus F1 score).

3.3 The KITTI Dataset

This dataset, collected by car-mounted cameras and a LIDAR sensor, was also widely used as a benchmark in previous studies. It contains 61 scenes belonging to the "city", "residential", and "road" categories. Eigen et al. [8] used 28 scenes for testing, and 28 scenes selected from the remaining scenes for training. Other study [43] used the same 28 scenes for testing, and all the remaining 33 scenes for training. We use the official KITTI *depth prediction* dataset which was made publicly available recently to provide a benchmark of depth map prediction task [44]. Although the official split of scenes is provided for training and testing, we use the same split as suggested in [43] to make a fair comparison with previous studies. The resulting training set consists of 23,158 pairs of images and depth maps.

As the dataset only provides sparse depth maps, we use the depth completion toolbox of the NYU-Depth V2 to interpolate pixels with missing depth. We resize the original images of size 1224×368 pixels to 756×228 pixels, which are used for inputs to our model. Following Eigen et al. [8], we discard the upper 1/3 of the images for training and testing (i.e., evaluation), making the output depth maps from our model have the size of 378×76 pixels. We train the base network for 10 epochs and the refinement network for 50 epochs.

We compare our method with existing methods which train CNNs in a supervised fashion as employed in our method, i.e., Eigen et al. [8], Liu et al. [28], and Liu et

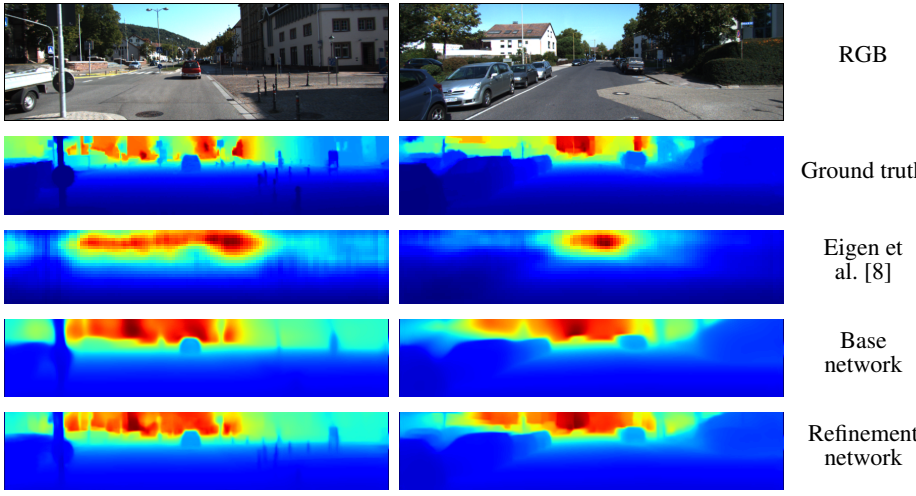


Fig. 6. Results obtained by different methods for two input images. From the first to the last row, the input RGB images, ground truth, the estimated depth maps of Eigen et al. [8], our base network, and refinement network, respectively.

Table 5. Edge accuracy for the KITTI dataset.

		Precision	Recall	F1 score
>1	Base net	0.870	0.809	0.835
	Refinement net	0.873	0.818	0.842
>5	Base net	0.780	0.687	0.727
	Refinement net	0.805	0.731	0.763
>10	Base net	0.770	0.613	0.679
	Refinement net	0.789	0.681	0.723

al. [45]. We also compare our method with recent "unsupervised" approaches [43, 46] which train CNNs using stereo pairs of images instead of ground truth depth maps, and another approach [47] that trains CNNs using both ground truth depths and stereo pairs of images. Table 4 shows their performances. It is seen that our method outperforms existing methods for all the accuracy measures. It is also observed that the refinement network contributes to improve accuracy from the estimated depth map of the base network. Fig. 6 shows results for two images. The same observations can be made as obtained in the results for the NYU-Depth V2. First, there are blurry and distorted object edges in the estimates of the base network, whereas they become sharper and less distorted in the estimates of the refinement network. Second, small objects, such as the poles of road signs and the poles along the sidewalk, which are not correctly recovered in the estimates of the base network, are recovered in the estimates of the refinement network. This is quantitatively confirmed by the proposed edge accuracy measure, as shown in Table 5.

Additional implementation details, qualitative results with visualization of learned features and predicted depths, results on the large indoor ScanNet dataset, and analysis of learning curves are given in the supplemental material. The code will be made publicly available.

4 Conclusion

We have presented a method for single view depth estimation using CNNs. It is based on the standard approach in which a CNN is trained in a supervised fashion using pairs of an input image and a ground truth depth map of a scene. We have pointed out that despite a number of studies pursuing this approach, there is still room for improvement in two aspects, which can be achieved by our method.

One is with room for increasing spatial resolution of estimated depth maps. Our model consists of two sub-networks named the base network and the refinement network; the former provides an initial estimate of a depth map, which is inputted to the latter to obtain a more accurate estimate. To recover loss of resolution due to a series of downsampling necessarily performed in the base network, we employ skip connections that transfer the lower layer outputs of the base network to the input layer of the refinement network. The two networks are independently trained in a sequential fashion.

The other aspect is with respect to estimation accuracy of boundaries of objects in scenes. The previous methods fail to correctly estimate their positions and shapes. We explain that this failure may be attributable to the loss function employed by the previous methods. The analyses of natural range image statistics indicate that the world can be decomposed into smooth surfaces and sharp discontinuities in between them; the latter corresponds to object boundaries. Then, this makes it important to be able to accurately reconstruct those discontinuities, which appear as step edges in depth maps, and to deal with them appropriately during training of CNNs. We have made simple analysis of how different loss functions affect measurement of estimation errors around step edges. Based on it, we argue that the loss of difference in depth is insensitive to positional shift and blurring of the edges, whereas the loss of difference in gradients tends to be sensitive to them. We further employ an additional loss of difference in surface normals, which is expected to be sensitive to small structures that tend to be neglected by the above two losses. We then propose to use a combined loss of the three loss functions.

Finally, we presented experimental results on the NYU-Depth V2 and KITTI datasets. There is an issue of existing measures for estimation accuracy, which is that they are also based on difference in depth, and thus fail to correctly measure reconstruction errors of step edges. Our method does improve overall accuracy in such traditional measures, but the margins are not very large. For more proper evaluation, we presented a simple measure for reconstruction accuracy of step edges, and then showed through experimental results that the proposed method outperforms the previous methods with a larger margin in this measure. This agrees well with visual comparisons between the results of the proposed method and those of the previous ones.

References

1. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (ECCV), Springer (2016) 694–711
2. Yeh, R., Chen, C., Lim, T.Y., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arXiv preprint arXiv:1607.07539 (2016)
3. Tartavel, G., Peyré, G., Gousseau, Y.: Wasserstein loss for image synthesis and restoration. SIAM Journal on Imaging Sciences **9**(4) (2016) 1726–1755
4. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision (ICCV). (2015) 1395–1403
5. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems (NIPS). (2015) 91–99
6. Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2017) 4
7. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
8. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems (NIPS). (2014) 2366–2374
9. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 5162–5170
10. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 1119–1127
11. Cao, Y., Wu, Z., Shen, C.: Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Transactions on Circuits and Systems for Video Technology (2017)
12. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in Neural Information Processing Systems (NIPS). (2016) 730–738
13. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 239–248
14. Xu, D., Ricci, E., Ouyang, W., Wang, X., Sebe, N.: Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 161–169
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). (2015) 3431–3440
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
17. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
18. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)

19. Liang-Chieh, C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Representations (ICLR). (2015)
20. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence (PAMI) (2017)
21. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
22. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). (2017) 2881–2890
23. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2015) 1520–1528
24. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence (PAMI) **39**(12) (2017) 2481–2495
25. Huang, J., Lee, A.B., Mumford, D.: Statistics of range images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2000) 324–331
26. Lee, A.B., Pedersen, K.S., Mumford, D.: The nonlinear statistics of high-contrast patches in natural images. International Journal of Computer Vision (IJCV) **54**(1-3) (2003) 83–103
27. Kalkan, S., Worgotter, F., Kruger, N.: Statistical analysis of local 3d structure in 2d images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (June 2006) 1114–1121
28. Liu, F., Shen, C., Lin, G., Reid, I.: Learning depth from single monocular images using deep convolutional neural fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **38**(10) (2016) 2024–2039
29. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. arXiv preprint arXiv:1709.07492 (2017)
30. Park, H., Lee, K.M.: Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 4623–4631
31. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). (2016) 770–778
32. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. arXiv preprint arXiv:1608.06993 (2016)
33. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015) 2650–2658
34. Sobel, I., Feldman, G.: A 3x3 isotropic gradient operator for image processing. a talk at the Stanford Artificial Project in (1968) 271–272
35. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2009) 248–255
36. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
37. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision (ECCV). (2012)

38. Chakrabarti, A., Shao, J., Shakhnarovich, G.: Depth from a single image by harmonizing overcomplete local network predictions. In: Advances in Neural Information Processing Systems (NIPS). (2016) 2658–2666
39. Li, J., Klein, R., Yao, A.: A two-streamed network for estimating fine-scaled depth maps from single rgb images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 3372–3380
40. Ladicky, L., Shi, J., Pollefeys, M.: Pulling things out of perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014) 89–96
41. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Towards unified depth and semantic prediction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 2800–2809
42. Dharmasiri, T., Spek, A., Drummond, T.: Joint prediction of depths, normals and surface curvature from rgb images using cnns. CoRR **abs/1706.07593** (2017)
43. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2017) 7
44. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: International Conference on 3D Vision (3DV). (2017)
45. Liu, F., Lin, G., Shen, C.: Discriminative training of deep fully connected continuous crfs with task-specific loss. IEEE Transactions on Image Processing **26**(5) (2017) 2127–2136
46. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision (ECCV), Springer (2016) 740–756
47. Kuznietsov, Y., Stückler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 6647–6655
48. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)

Supplemental material

In this supplemental material, we provide implementation details, and additional experimental analyses and results.

In Section A, hyperparameters of the DenseNet used in our experiments are given. Visualization of the learned features is given in Section B. We analyzed the learning curves in Section C. Additional results for depth estimation using the ScanNet dataset are provided in Section D.

A Hyperparameters of the DenseNet used in the Experiments

The layers from conv1 to block4 used in our proposed architecture (see Fig. 2 of the main text) are original layers of the DenseNet-169. The dense block, which follows a convolutional and pooling layer (transition layer) in the DenseNet, is denoted by block(i) with pool(i+1) in the proposed network architecture (see Fig. 2 of the main text). Additional implementation details for kernels used at layers from the 2^{nd} (conv2) to the 7^{th} (conv7) convolution layer are given in Table 6.

Table 6. Hyperparameters of kernels used at layers from the 2^{nd} (conv2) to the 7^{th} (conv7) convolution layer of the DenseNet.

conv	kernel size	stride	padding
conv2	1×1	1	0
conv3	3×3	1	1
conv4	5×5	1	2
conv5	5×5	1	2
conv6	5×5	1	2
conv7	5×5	1	2

B Visualization of Learned Features

We show some features computed at middle layers, and predicted depth map of the base and refinement network. More precisely, we show features computed at the output of the up4 layer for the base network, and those computed at the conv6 layer for the refinement network. Fig. 7 shows results of four images belonging to the NYU-Depth V2 dataset. As observed in the figure, the refinement network can extract features with more detailed local information and less structural distortion. Therefore, the refinement network efficiently improved the estimations obtained from the base network.

C Analysis of Learning Curves

In this section, we analyze the effect of change of loss function. We compare the loss curves of different loss functions, i.e. l_{depth}, L_{base}, L_{refine} in Fig. 8. In the results, each

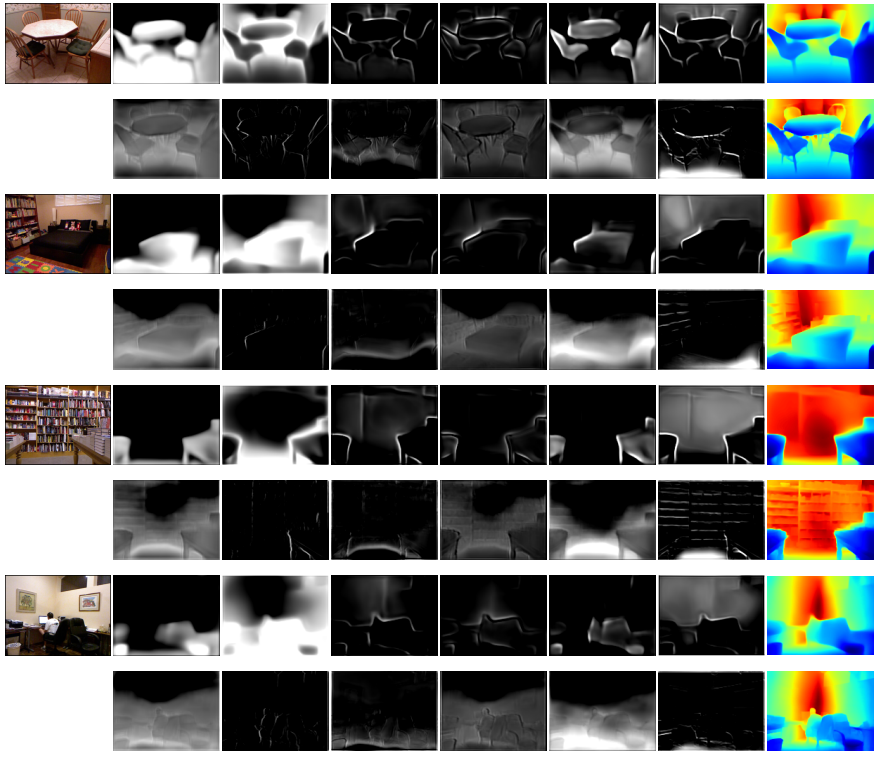
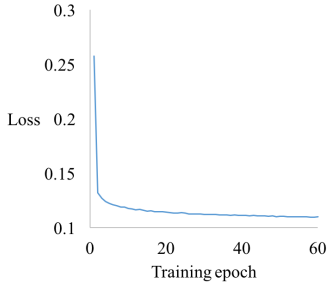


Fig. 7. From left to right: RGB image, selected output features (6 channels), and predicted depth map. For each image, the corresponding result obtained from the base network is shown in the upper row, and that of the refinement network is shown in the lower row.

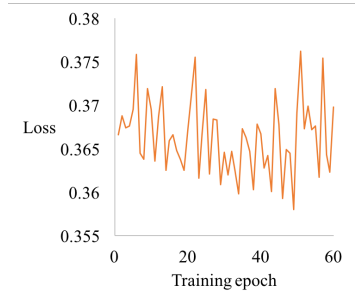
loss function has a similar loss curve for the training dataset. However, it is observed that over-fitting occurs when we train the network with l_{depth} alone, as shown in Fig. 8(b). When we employ a different loss L_{base}, and train the same network under the same condition, over-fitting is well mitigated, as shown in Fig. 8(d). In addition, training the network with the full loss function L_{refine}, provided the best generalization performance (see Fig. 8(f)).

D Depth Estimation Results for the ScanNet Dataset

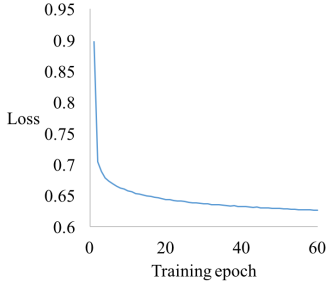
In this section, we show the results for depth estimation using the ScanNet dataset [48], which is a very large indoor dataset consisting of 2.5 million RGB-D images of 1513 scenes. In the experiments, we used a subset that consists of 17,269 samples for training, 2713 samples for validation, and 5554 samples for testing. Compared with the NYU-Depth V2, this dataset is more difficult to train because of larger number of scenes and less number of training samples. We used a pretrained model on the NYU-Depth V2 for a base network to avoid over-fitting. Depth prediction results and edge accuracy are



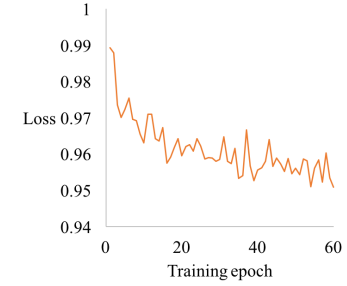
(a) Loss curve for the training dataset (l_{depth}).



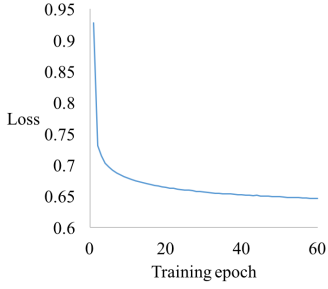
(b) Loss curve for the test dataset (l_{depth}).



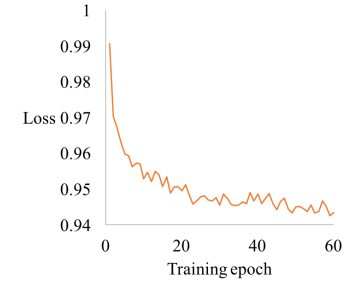
(c) Loss curve for the training dataset (L_{base}).



(d) Loss curve for the test dataset (L_{base}).



(e) Loss curve for the training dataset (L_{refine}).



(f) Loss curve for the test dataset (L_{refine}).

Fig. 8. Loss curve of the refinement network trained with different loss functions (l_{depth}, L_{base}, L_{refine}) using the NYU-Depth V2. λ is set to 10, μ is set to 1 for all the experiments.

shown in Table 7 and Table 8, respectively. Visual results are given for some sample images in Fig. 9.

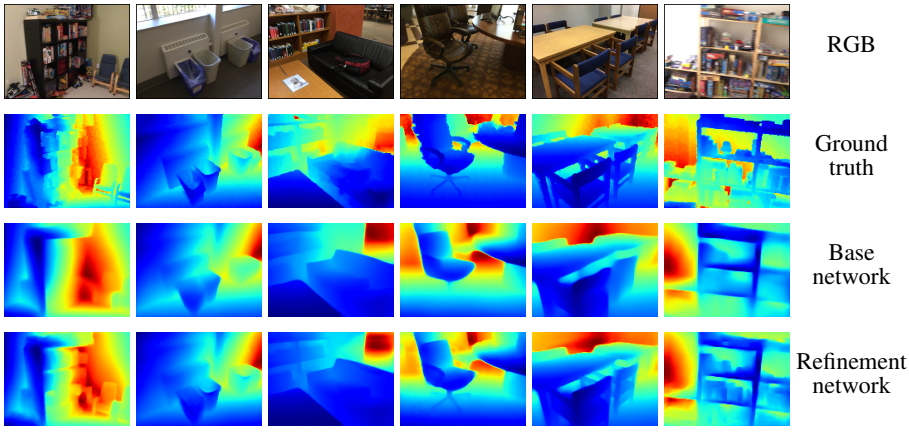


Fig. 9. Visual comparison of predicted depth maps. From the first row to the last row: Inputs of RGB images, ground truth, results obtained using our base network, and our refinement network.

Table 7. Depth prediction results for the ScanNet dataset.

Method	RMS	REL	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Base net	0.409	0.142	0.808	0.950	0.986
Refinement net	**0.404**	**0.141**	**0.810**	**0.952**	**0.987**

Table 8. Edge accuracy for the ScanNet dataset.

		Precision	Recall	F1 score
>0.25	Base net	0.778	0.478	0.577
	Refinement net	0.786	0.506	0.599
>0.5	Base net	0.824	0.499	0.607
	Refinement net	0.825	0.530	0.631
>1	Base net	0.883	0.587	0.691
	Refinement net	0.886	0.600	0.702