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

This paper considers the problem of single image depth estimation. The employment of convolutional neural networks (CNNs) has recently brought about significant advancements in the research of this problem. However, most existing methods suffer from loss of spatial resolution in the estimated depth maps; a typical symptom is distorted and blurry reconstruction of object boundaries. In this paper, toward more accurate estimation with a focus on depth maps with higher spatial resolution, we propose two improvements to existing approaches. One is about the strategy of fusing features extracted at different scales, for which we propose an improved network architecture consisting of four modules: an encoder, decoder, multi-scale feature fusion module, and refinement module. The other is about loss functions for measuring inference errors used in training. We show that three loss terms, which measure errors in depth, gradients and surface normals, respectively, contribute to improvement of accuracy in a complementary fashion. Experimental results show that these two improvements enable to attain higher accuracy than the current state-of-the-arts, which is given by finer resolution reconstruction, for example, with small objects and object boundaries.

1. Introduction

The problem of estimating the depth map of a scene from its single image has attracted a lot of attention in the field of computer vision, since depth maps have a lot of applications, such as augmented reality [22], human computer interaction [33], human activity recognition [12], scene recognition [41], and segmentation [27, 32]. The recent employment of convolutional neural networks (CNNs) has accelerated the research of the problem [9, 25, 23, 2, 4]. In early studies, depth maps can be estimated only with much lower resolution than input images, which is mostly at-
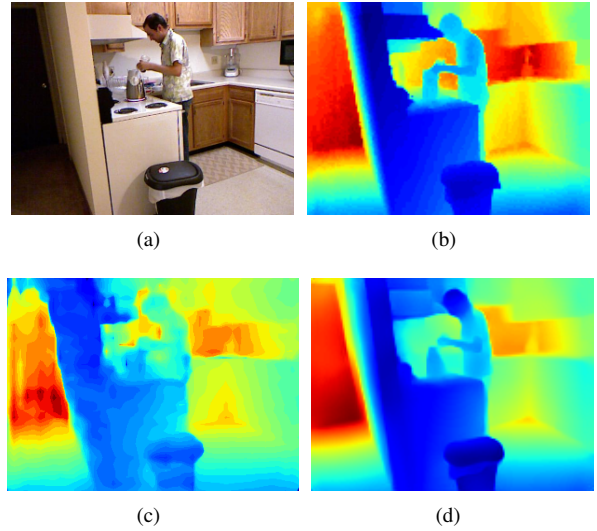

Figure 1. Example comparison of estimated depth maps; (a) RGB input, (b) ground truth depth, (c) the current state-of-the-art [10], and (d) our method.

tributable to a series of downsampling operations performed in CNNs. To recover the degraded spatial resolution, a number of studies have been conducted so far. First, a new upsampling method called up-projection was proposed in [16]. Integration of CRFs into CNNs for depth refinement, along with its end-to-end training method, was proposed in [40]. Several works applied joint multi-task learning [8, 6], which provides a good amount of improvement of estimated depth maps. Recently, dilated convolution was employed in [10], updating the state-of-the-art accuracy.

In this paper, we argue that there is room for further improvements despite these previous efforts. In the depth maps estimated by the previous methods, including the one that has achieved the state-of-the-art accuracy, we can observe distortions in object shapes, missing small objects, and mosaic patterns, as shown in Fig.1. Depth maps with high spatial resolution, e.g., precise object boundaries, are

especially important for some applications such as object recognition [35] and depth-aware image re-rendering and editing [24].

The features learned in different layers of CNNs for depth estimation, which represent information at different scales, should contain different depth cues. For example, the lower layer features may have information on the details of object shapes, while the higher layer features represent global depth information without details of object shapes. Therefore, the former and latter should be used in a complementary fashion to estimate more accurate depth maps. Toward this end, we propose a feature fusion module that up-scales lower layer features to the same (high) level of resolution using skip connections, and then learns how to fuse the up-scaled different scale features with a convolutional layer. We also employ a decoder module to decode the high level feature extracted by the encoder. Then, the outputs of the two modules are integrated and further refined to provide final estimation.

To summarize, we propose an improved architectural design that consists of four modules, namely, i) an encoder (E) used for multi-scale feature extraction, ii) a decoder (D) used for feature decoding, iii) a multi-scale feature fusion (MFF) module to fuse features of different resolutions extracted by E, and iv) a refinement module (R) to refine and integrate features obtained from D and MFF for final prediction. The proposed framework is shown in Fig. 2. Note that the employment of symmetric skip connections was widely used in previous studies on many tasks, it needs pay attention to fix the channels of the same scale features between the encoder and decoder, while the proposed feature fusion module is more flexible to use and it can be used with different backbone networks employed for the decoder-encoder part. In our experiments, we show results obtained with several backbone networks, such as ResNet [13], DenseNet [11], and SENet [14]. Our method is trained in an end to end fashion without employing any post-processing refinement.

We also propose an improved loss function for training the CNNs. It is motivated by the previous studies on statistical properties of range images of *natural scenes* [15, 1, 19]. In [15], 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). As mentioned above, however, previous methods often fail to recover such edges correctly.

We argue that distorted and blurry edges emerged in the depth maps estimated by previous methods are mostly attributable to the loss function employed by them. Many previous studies use 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 [9, 23, 26], ℓ_1 loss [28, 30], and a robust $berhu$ loss [16]). We point out that this type of losses 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, 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 particularly around object edges. We show experimental results that confirm effectiveness of our approach.

2. Related Work

Skip connections Skip connections have been widely used for many purposes. In particular, they are employed in encoder-decoder networks to recover the degraded resolution due to downsampling, as in U-Net [7]. In that case, skip connections are symmetrically inserted between the same level of layers in the encoder and decoder. This design is applied to image restoration [29] , semantic segmentation [18], and others. In this paper, we use skip connections in a different way. Instead of symmetric application between the encoder and decoder, we directly merge the up-scaled feature maps from the different layers of the encoder to the final decoder output. This is done by the proposed multi-scale feature fusion module.

Gradient-based loss Several existing studies also employ functions of depth gradients for losses [8, 36]. Most of previous methods are built upon the framework of multi-task learning, such as [8, 38] where semantic segmentation, surface normals etc. are simultaneously learned. On the other hand, we are interested in the following question: how far can we go in the minimal case where only ground truth depths are available for training? This question is important, because RGBD cameras have advanced greatly, making them easier to use, whereas segmentation labels are costly to obtain. Therefore, we revisit this minimal formulation. We analyze how the three different losses (i.e., l_{depth}, l_{grad}, and l_{normal}) can contribute to the improvement of accuracy. There are other works that use gradient-based losses including DeMon [36], which consider different problems from single image depth estimation.
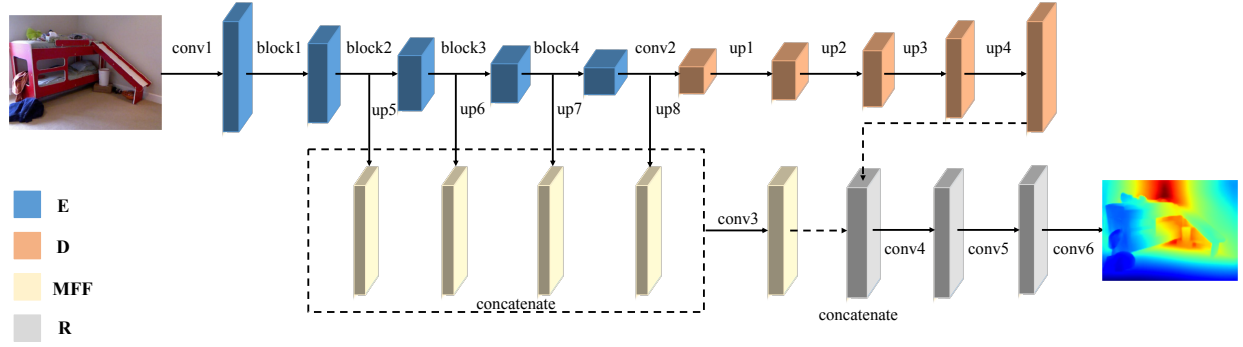
Figure 2. A diagram of the proposed network architecture. Given an input image, the encoder (E) extracts multi-scale features (1/4, 1/8, 1/16, and 1/32). The decoder (D) converts the last 1/32 scale feature to get a 1/2 scale feature. Each of the multi-scale features is up-scaled to 1/2 scale, and fused by the multi-scale feature fusion module (MFF). The outputs of D and MFF and are refined by the refinement module (R) to obtain the final depth map. Each box named "block$n$" denotes a block of multiple convolutional layers, such as residual block of ResNet; each box named "up$n$" denotes a up-projection layer introduced in [16]. Batch normalization and ReLU nonlinearity are applied to the output of each convolutional layer except conv6.
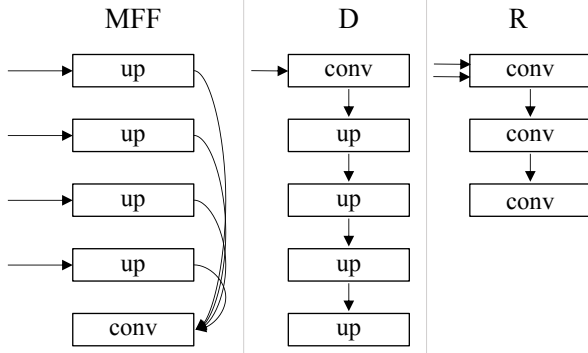

Figure 3. Diagrams of MFF, D and R. We employ the same up-sampling strategy (up-projection) used in [16].

3. Proposed Method

3.1. Improved Network Design

The proposed network architecture consists of four modules: an encoder (E), a decoder (D), a multi-scale feature fusion module (MFF), and a refinement module (R), as shown in Fig. 2. The encoder extracts features at multiple scales: 1/4, 1/8, 1/16, and 1/32. The decoder employs four up-projection modules [16] to gradually up-scale the final feature from the encoder while decreasing the number of channels. Similar encoder-decoder networks were employed in [16, 28], but their outputs tend to lose spatial resolution and not to preserve object shapes.

The MFF module integrates four different scale features from the encoder using up-projection and channel-wise concatenation. To be specific, the outputs of the four encoder blocks (each having 16 channels) are upsampled by ×2, 4, 8, and 16, respectively, so as to have the same size as the fi-

Table 1. Sizes of output features, and input/output channels of each layer when using a ResNet-50 as encoder.

Layer	Output Size	Input/C	Output/C
conv1	114×152	3	64
block1	57×76	64	256
block2	29×38	256	512
block3	15×19	512	1020
block4	8×10	1020	2040
conv2	8×10	2040	1020
up1	15×19	1020	512
up2	29×38	512	256
up3	57×76	256	128
up4	114×152	128	64
up5	114×152	256	16
up6	114×152	512	16
up7	114×152	1020	16
up8	114×152	2040	16
conv3	114×152	64	64
conv4	114×152	128	128
conv5	114×152	128	128
conv6	114×152	128	1

nal output. This upsampling is done in a channel-wise manner. Then they are concatenated and further transformed by a convolutional layer to obtain the output of the MFF module, which has 64 channels. The main purpose of the MFF module is to merge different information at multiple scales into one. It is seen in Fig. 4 that the lower layer outputs of the encoder retain information with finer spatial resolution, which should be utilized to restore the fine details lost due to the multiple applications of downsampling.

The feature from the decoder and fused multi-scale features from MFF are then concatenated in their channels and

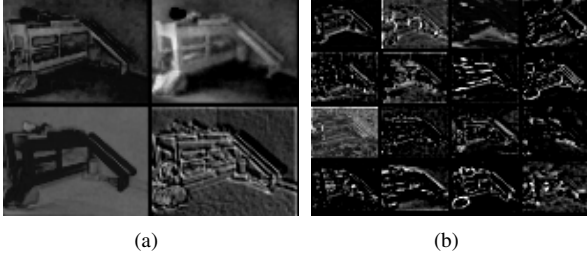|	l_{depth}	l_{grad}	l_{normal}
	✓	✗	✗
	✗	✓	✓
	✗	✓	✓

Figure 4. Visualization of outputs of different layers of the encoder network for the input image shown in Fig. 2. Selected channels of (a) block1, and (b) block2.

Figure 5. 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 by 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.

fed to the refinement module having three convolutional layers to give the final prediction. MFF and the refinement module only add a small number of parameters as compared with the encoder-decoder part. For instance, if we use the ResNet-50 as the encoder, then the encoder-decoder part has 63.6M parameters, while the feature fusion and refinement modules have only 4M parameters.

3.2. Loss Functions

Most of the previous studies employ the sum of the difference between the depth estimate d_i and its ground truth g_i for a loss:

$$l_1 = \frac{1}{n}\sum_{i=1}^{n} e_i, \qquad (1)$$

where $e_i = \|d_i - g_i\|_1$. Some use its ℓ_2 norm or robustified version. There are two issues with this type of losses. One is that a unit depth difference (e.g., 1cm) has an equal contribution to the loss between distant and nearby points in a scene. It should be more on nearby points and less on distant points. This is pointed out in [21], where a depth-balanced Euclidean loss is employed. We propose a simpler remedy here, which is to use the logarithm of depth errors as

$$l_{\text{depth}} = \frac{1}{n}\sum_{i=1}^{n} F(e_i), \qquad (2)$$

where

$$F(x) = \ln(x + \alpha), \qquad (3)$$

where $\alpha(> 0)$ is a parameter we set.

The other issue, which is the main concern here, is that for a step edge structure of depth, while the above conventional loss is sensitive to shifts in depth direction, it is comparatively insensitive to shifts in x and y directions, as illustrated in the top and second rows of Fig. 5. It is similarly insensitive also to distortion and blur of edges.

The statistics of natural range images indicate that natural scenes consist of a lot of such step edge structures [15], 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. Thus, we consider the following loss function of the gradients of depth;

$$l_{\text{grad}} = \frac{1}{n}\sum_{i=1}^{n} (F(\nabla_x(e_i)) + F(\nabla_y(e_i))), \qquad (4)$$

where $\nabla_x(e_i)$ is the spatial derivative of e_i computed at the ith pixel with respect to x, and so on. This loss is sensitive to the shift of edges in x and y directions, as shown in Fig. 5. 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.

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 [15]. 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 λl_{grad} cannot penalize small structural errors such as those of high-frequency undulation of a surface, as shown in the bottom row of Fig. 5.

To deal with such small depth structures and further improve fine details of depth maps, we consider yet another loss for training, 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

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

where $\langle \cdot, \cdot \rangle$ denotes 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. 5. Thus, we can say that l_{normal} is also complementary with the other two losses. Finally, we define the loss by

$$L = l_{\text{depth}} + \lambda l_{\text{grad}} + \mu l_{\text{normal}}, \quad (6)$$

where $\lambda, \mu \in \mathbb{R}$ are weighting coefficients.

3.3. Accuracy Measures for Depth Estimation

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 the previous studies:

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

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

- Mean $\log 10$ error ($\log 10$): $\frac{1}{T} \sum_{i=1}^{T} \| \log_{10} d_i - \log_{10} g_i \|_1$.

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

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 before. For instance, the method of Ma and Karaman [28], which leverages known depths at a few scene points to improve depth estimation, does outperform other methods 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 the recent ones; estimated depth maps showing small 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 that is sensitive to positional errors of edges which will be overlooked by the above measures. For this purpose, we apply the Sobel operator [17] 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. We used three different thresholds: 0.25, 0.5, and 1. Assuming those of the true depth map to be true, we measure precision (P), recall (R) and F1 score for those of the estimated map.

4. Experiments

4.1. Implementation Details

We use the NYU-Depth V2 dataset [34] which consists of a variety of indoor scenes, and is the most widely used for the task of single view depth prediction. In the previous studies [9, 16, 28, 8, 40], it is shown that data argumentation helps to improve accuracy as well as to 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:

- 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 $c \in [0.6, 1.4]$.

We follow the same procedure as the one employed in the previous studies. We use the official splits for 464 scenes, i.e., 249 scenes for training and 215 scenes for testing. Following previous methods, we downsample images from original size (640×480) to 320×240 pixels using bilinear interpolation, and then crop their central parts to obtain images with 304×228 pixels. For training, 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 proposed network for 20 epochs. The encoder module in the network is initialized by a model pretrained with the ImageNet dataset [5]. The other layers in the network are randomly initialized. We use Adam optimizer with an initial learning rate of 0.0001, and reduce it to 10% for every 5 epochs. We set $\beta_1 = 0.9$, $\beta_2 = 0.999$, and use weight decay of 0.0001. The weights λ of l_{grad} and μ of l_{normal} are set as $\lambda = 1$ and $\mu = 1$, and α in the mapping function is set to 0.5 in all the experiments. We conducted all the experiments using PyTorch [31] with batch size of 8.

Table 2. Comparisons of different methods on the NYU-Depth V2 dataset. The methods marked by [*] use partially known depths, and those with [**] employ joint task learning.

Method	RMS	REL	log 10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [9]	0.907	0.215	-	0.611	0.887	0.971
Liu et al. [25]	0.824	0.230	0.095	0.614	0.883	0.971
Chakrabarti et al. [3]	0.620	0.149	-	0.806	0.958	0.987
Cao et al. [2]	0.819	0.232	0.091	0.646	0.892	0.968
Li et al. [24]	0.635	0.143	0.063	0.788	0.958	0.991
Ma and Karaman [28]	(-)	0.143	-	0.810	0.959	0.989
Laina et al. [16]	0.573	0.127	0.055	0.811	0.953	0.988
Xu et al. [40]	0.586	0.121	0.052	0.811	0.954	0.987
Lee et al. [21]	0.572	0.139	-	0.815	0.963	0.991
Fu et al. [10]	**0.509**	**0.115**	0.051	0.828	0.965	0.992
Qi et al. [32]	0.569	0.128	0.057	0.834	0.960	0.990
Ours (ResNet-50)	0.555	0.126	0.054	0.843	0.968	0.991
Ours (DenseNet-161)	0.544	0.123	0.053	0.855	0.972	0.993
Ours (SENet-154)	0.530	**0.115**	**0.050**	**0.866**	**0.975**	**0.993**
Ma and Karaman [28]*	(-)	0.044	-	0.971	0.994	0.998
Li et al. [23]	0.821	0.232	0.094	0.621	0.886	0.968
Eigen and Fergus [8]**	0.641	0.158	-	0.769	0.950	0.988
Dharmasiri et al. [6]**	0.624	0.156	-	0.776	0.953	0.989
Xu et al. [39]**	0.582	0.120	0.055	0.817	0.954	0.987

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

Threshold	Method	P	R	F1
>0.25	Eigen and Fergus [8]**	0.544	0.481	0.500
	Laina et al [16]	0.489	0.435	0.454
	Xu et al. [40]	0.516	0.400	0.436
	Fu et al. [10]	0.320	0.583	0.402
	Ours (SENet-154)	0.644	0.508	0.562
>0.5	Eigen and Fergus [8]**	0.587	0.456	0.501
	Laina et al [16]	0.536	0.422	0.463
	Xu et al. [40]	0.600	0.366	0.439
	Fu et al. [10]	0.316	0.473	0.412
	Ours (SENet-154)	0.668	0.505	0.568
>1	Eigen and Fergus [8]**	0.733	0.488	0.574
	Laina et al [16]	0.670	0.479	0.548
	Xu et al. [40]	0.794	0.407	0.525
	Fu et al. [10]	0.483	0.512	0.485
	Ours (SENet-154)	0.759	0.540	0.623

4.2. Performance Comparison

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 use additional information other than the input RGB images; those with [*] use relative depth between pairs or partially known depths [4, 28], and those with [**] employ joint task learning [20, 37, 23, 8, 6, 39]. For the method of Ma and Karaman [28], 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 [9, 16, 40, 25, 2, 3, 24, 21, 32, 10] and ours should be able to be compared in an equal condition.

Our method achieved the best performance on REL, $\log 10$, and outperformed previous methods for $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$ with a large margin. It is observed that our method provided the second best performance for RMS, while the method of [10] performed a little better on this measure. However, they used 120K training samples, and we used less number of training samples (50K samples). In addition, our method outperformed the methods proposed in [16, 28], which employed encoder-decoder networks with ResNet-50.

Figure 6 shows the depth maps estimated by different methods[2], including joint multi-task learning method [8], encoder-decoder network [16], CRF-based refinement method [40], dilated ordinary regression network [10] and our method. We show them in the ascending order of quality in traditional measures. It is seen that the methods of [16, 40] suffer from heavy distortion of shapes, and the method of [10] produces sharp discontinuities of shapes of

---

[1]RMS values given in [28] 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.

[2]For [8], we show the results that are made publicly available by the authors. For [16, 40, 10], we use the authors' code to obtain the results and show them here.
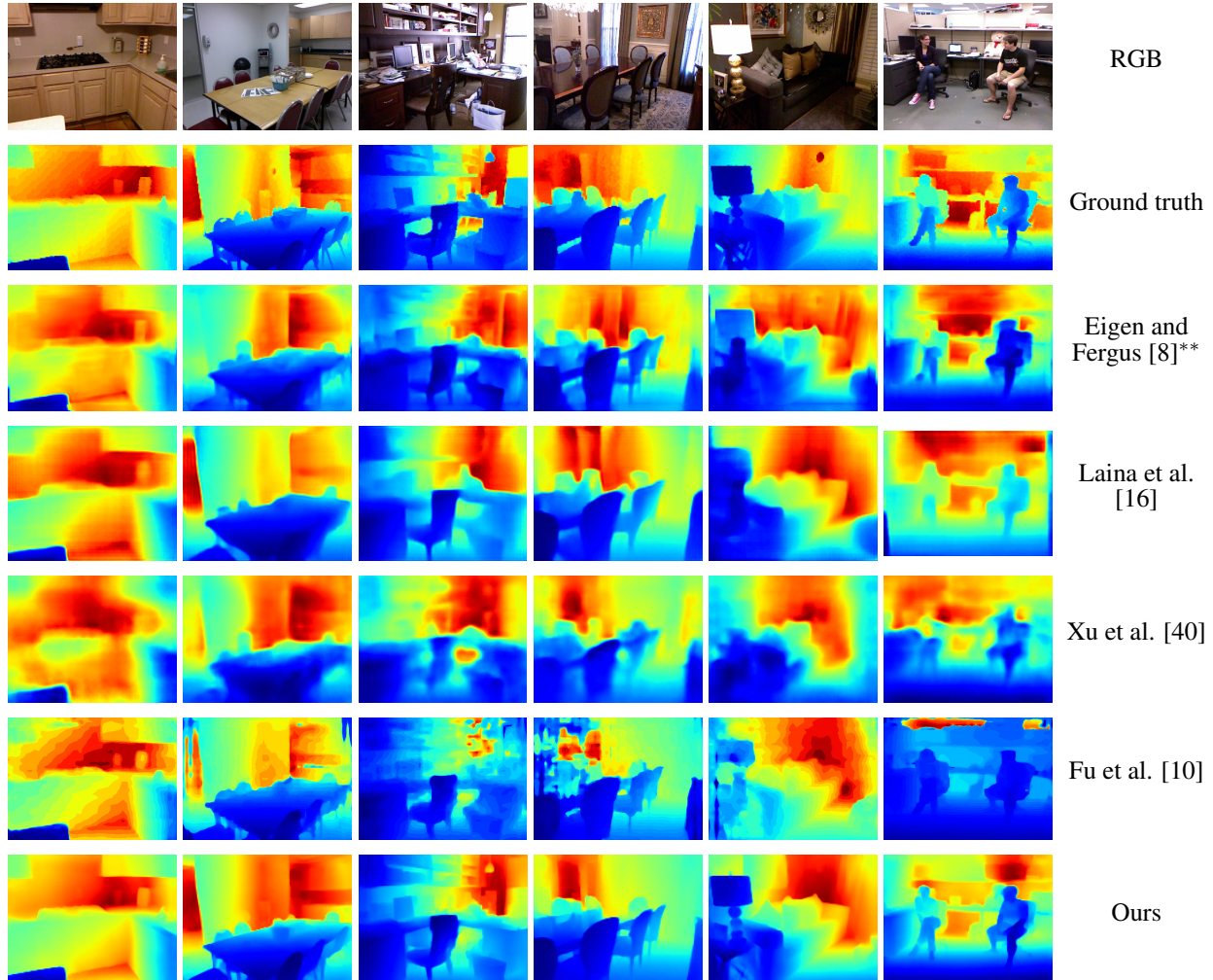
Figure 6. Results of different methods for six images. From the first to the last row; input RGB images, ground truth depth map, a multi-task learning method [8], encoder-decoder network [16], CRF-based method [40], dilated ordinary regression network [10], and our proposed network trained with the full loss function. We show them in the ascending order of quality in traditional measures.

objects, and mosaic effect is even observed. Although the method of [8] shows relatively clear image boundaries, they produced many inaccurate weak edges which can be observed in the far side of each room. Our method shows significant performance by correctly recovering edges of objects and small structures, such as bottles in a kitchen and lamp shades on a desk.

We also observe a relationship between our proposed measure of edge accuracy and visual quality of estimated depth maps. For instance, our proposed method outperforms all the other methods given in Table 3, and provides depth maps with the finest details in Fig.6. A similar relationship is also observed for the method of [8] which shows the best scores among the previous works in Table 3, and provides better visual results compared to these related work in Fig.6.

Table 4. Results of our method that is built on ResNet-50 trained with different loss functions on the NYU-Depth V2 dataset. For edge accuracy, we report the results for >0.5.

	RMS	REL	$\delta < 1.25$	F1
w/ l_{depth}	0.580	0.133	0.830	0.525
w/ $l_{\text{depth}} + \lambda l_{\text{grad}}$	0.563	0.128	0.841	0.543
w/ full loss	**0.555**	**0.126**	**0.843**	**0.548**

4.3. Ablation Study

In order to compare and analyze the performance of the proposed loss functions, we train our model with different losses while using ResNet-50 as an encoder. In visual comparison of their estimated depth maps side by side, as shown in Figure 7 with several samples, it can be observed that the results trained with l_{depth} are more distorted and blurry.
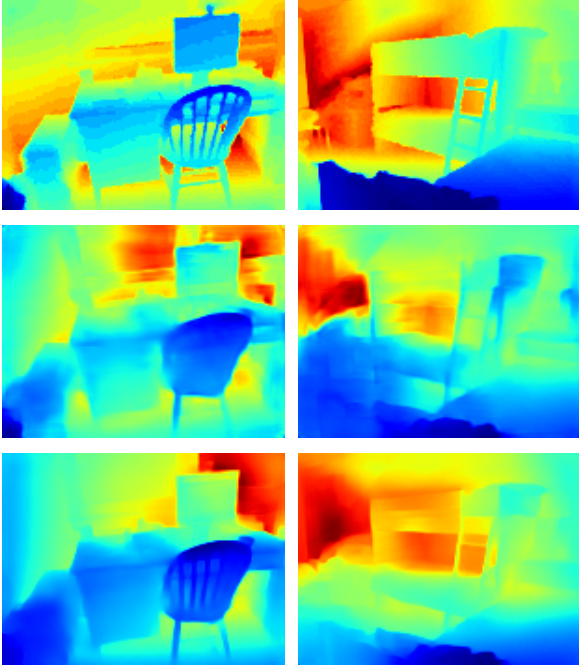
Figure 7. Visual comparison of our method trained using ResNet-50 with different loss functions on the NYU-Depth V2 dataset. From top to bottom: Ground Truth, trained with l_{depth} and the full loss, respectively.

Moreover, the proposed loss function provides significant improvement especially for estimation of fine details and clear boundaries of objects in scenes. The numerical results given in Table 4, show that finer details of objects in the scenes are gradually recovered using a model with l_{depth}, $l_{\mathrm{depth}} + \lambda l_{\mathrm{grad}}$ and the full loss $l_{\mathrm{depth}} + \lambda l_{\mathrm{grad}} + \mu l_{\mathrm{normal}}$, respectively.

5. Conclusion

We have presented two improvements to existing methods for single image depth estimation. One is the improved design of network architecture. It consists of four modules: an encoder, a decoder, a multi-scale feature fusion module and a refinement module. Any base network can be used for the encoder, such as ResNet, DenseNet, and SENet. The overall network is trained in an end to end fashion without any post-processing refinement.

The previous methods fail to correctly estimate boundaries of objects in scenes. We explain that this failure may be attributable to the loss functions employed by the previous methods. The analyses of natural range image statistics indicate that the real world scene 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 a 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 dataset. We observed that existing measures, which make use of difference in depth, fail to correctly measure reconstruction error of step edges for evaluation of estimation accuracy. For more proper evaluation, we presented a simple measure of reconstruction accuracy of step edges. Our method outperforms previous methods using both traditional measures and the proposed measure, especially providing remarkable improvement on reconstruction of small objects and estimation of object boundaries. This agrees well with visual comparisons between the results of the proposed method and those of the previous ones.

References

[1] L. A. B, P. K. S, and M. David. The nonlinear statistics of high-contrast patches in natural images. *IJCV*, 54(1-3):83–103, 2003.

[2] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[3] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, pages 2658–2666, 2016.

[4] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *NIPS*, pages 730–738, 2016.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[6] T. Dharmasiri, A. Spek, and T. Drummond. Joint prediction of depths, normals and surface curvature from rgb images using cnns. *CoRR*, abs/1706.07593, 2017.

[7] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, pages 179–187. Springer, 2016.

[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV*, pages 2650–2658, 2015.

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[10] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[11] H. Gao, L. Zhuang, W. K. Q, and van der Maaten Laurens. Densely connected convolutional networks. *CVPR*, 2017.

[12] R. K. Gupta, A. Y. S. Chia, and D. Rajan. Human activities recognition using depth images. In *ACM Multimedia*, 2013.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[14] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[15] J. Huang, A. B. Lee, and D. Mumford. Statistics of range images. In *CVPR*, volume 1, pages 324–331, 2000.

[16] L. Iro, R. Christian, B. Vasileios, T. Federico, and N. Nassir. Deeper depth prediction with fully convolutional residual networks. In *3DV*, pages 239–248, 2016.

[17] S. Irwin and F. Gary. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.

[18] J. Jiang, L. Zheng, F. Luo, and Z. Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv preprint arXiv:1806.01054*, 2018.

[19] S. Kalkan, F. Wörgötter, and N. Krüger. Statistical analysis of local 3d structure in 2d images. In *CVPR*, volume 1, pages 1114–1121, June 2006.

[20] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, pages 89–96, 2014.

[21] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim. Single-image depth estimation based on fourier domain analysis. In *CVPR*, pages 330–339, 2018.

[22] W. Lee, N. Park, and W. Woo. Depth-assisted real-time 3d object detection for augmented reality. *ICAT*, 2:126–132, 2011.

[23] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. *CVPR*, pages 1119–1127, 2015.

[24] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *CVPR*, pages 3372–3380, 2017.

[25] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, pages 5162–5170, 2015.

[26] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *PAMI*, 38(10):2024–2039, 2016.

[27] Y. Lu, J. Zhou, J. Wang, J. Chen, K. Smith, C. Wilder, and S. Wang. Curve-structure segmentation from depth maps: A cnn-based approach and its application to exploring cultural heritage objects. *AAAI*, 2018.

[28] F. Ma and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. *ICRA*, 2018.

[29] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, 2016.

[30] H. Park and K. M. Lee. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. *ICCV*, pages 4623–4631, 2017.

[31] A. Paszke, S. Gross, and A. Lerer. Automatic differentiation in pytorch. 2017.

[32] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, pages 283–291, 2018.

[33] Z. Ren, J. Meng, and J. Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. *2011 8th International Conference on Information, Communications and Signal Processing*, pages 1–5, 2011.

[34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[35] X. Song, L. Herranz, and S. Jiang. Depth cnns for rgb-d scene recognition: Learning from scratch better than transferring from rgb-cnns. In *AAAI*, pages 4271–4277, 2017.

[36] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, volume 5, page 6, 2017.

[37] P. Wang, X. Shen, Z. L. Lin, S. Cohen, B. L. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, pages 2800–2809, 2015.

[38] P. Wang, X. Shen, B. Russell, S. Cohen, B. L. Price, and A. L. Yuille. Surge: Surface regularized geometry estimation from a single image. In *NIPS*, pages 172–180, 2016.

[39] D. Xu, W. Ouyang, X. Wang, and N. Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. *CVPR*, 2018.

[40] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *CVPR*, pages 161–169, 2017.

[41] S. Zia, B. Yüksel, D. Yüret, and Y. Yemez. Rgb-d object recognition using deep convolutional neural networks. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 887–894. IEEE, 2017.