- Supplementary Material - Guiding Monocular Depth Estimation Using Depth-Attention Volume

Lam Huynh[1], Phong Nguyen-Ha[1], Jiri Matas[2], Esa Rahtu[3], and Janne Heikkilä[1]

[1] Center for Machine Vision and Signal Analysis, University of Oulu, Finland
[2] Center for Machine Perception, Czech Technical University, Czech Republic
[3] Computer Vision Group, Tampere University, Finland

In this document, we present additional qualitative results on NYU-Depth-v2 and SUN-RGBD datasets in Section 1 and 2, respectively. Section 3 provides extensive analysis for planarity error and non-local embedding space selection strategy. Details of the network architecture are described in Section 4 while Section 5 gives the definitions of the evaluation metrics. Besides, we attach a video demo of our monocular depth estimation model for a random indoor scene in the supplementary material.

1 Additional qualitative results on NYU-Depth-v2

This section provides further results and analysis on NYU-Depth-v2 dataset.

1.1 Depth map with and without $\mathcal{L}_{attention}$

As shown in Figure 1 and 2, the model with full loss significantly improves depth map quality at boundaries and detailed areas.

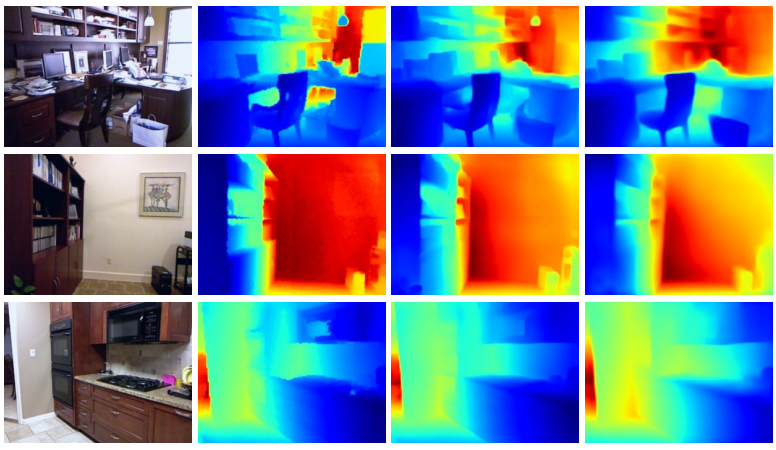

Fig. 1. Predicted depth maps from our model train with and without the $\mathcal{L}_{attention}$ term.

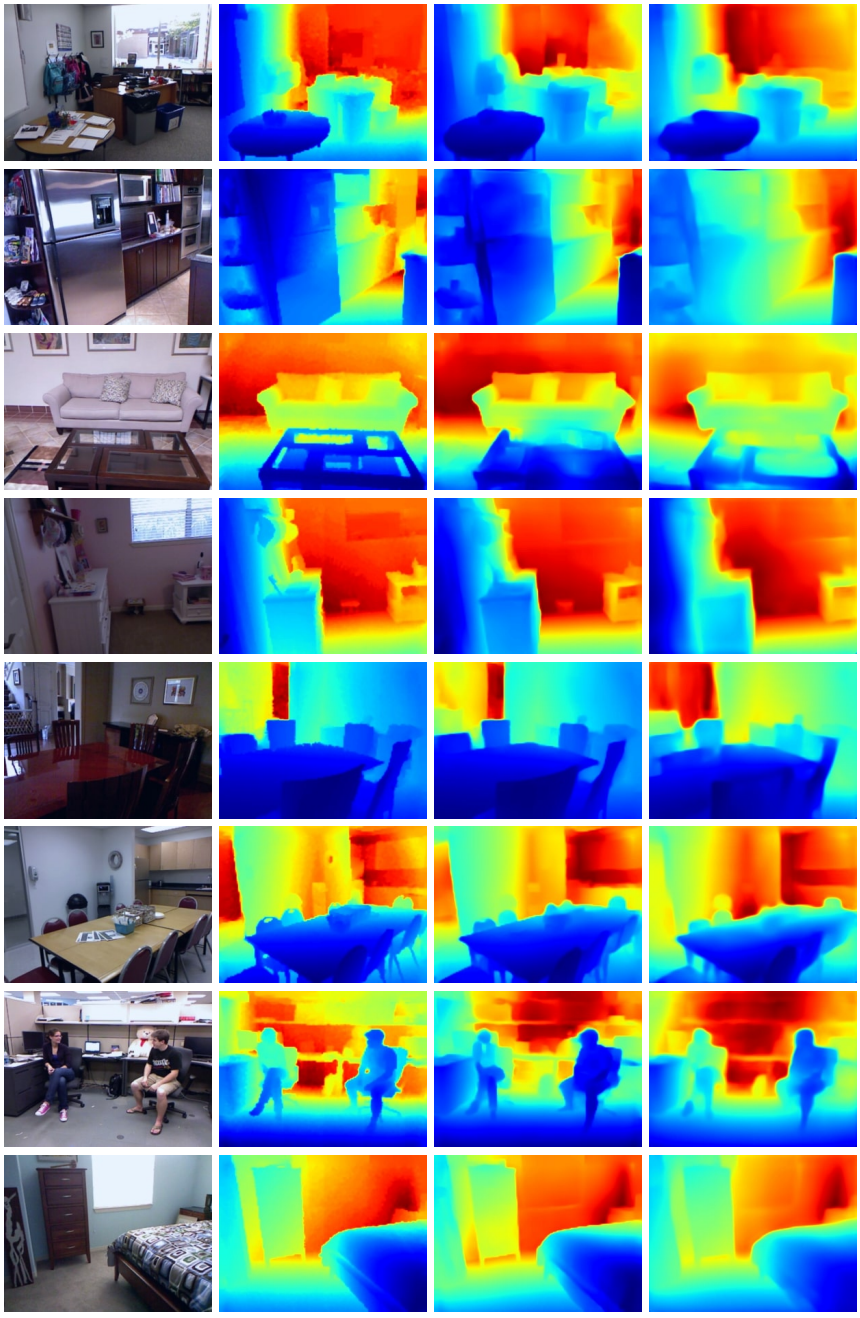

Fig. 2. Predicted depth maps from our model train with and without the $\mathcal{L}_{attention}$ term.

1.2 Point cloud reconstructions

We further examine the accuracy of the predicted depth maps by reconstructing the point clouds of three arbitrary views in the NYU-Depth-v2 test set. The back-projected 3D points are shown in Figure 3. The results near the walls, floors and ceilings are virtually linear and close to the ground truths.

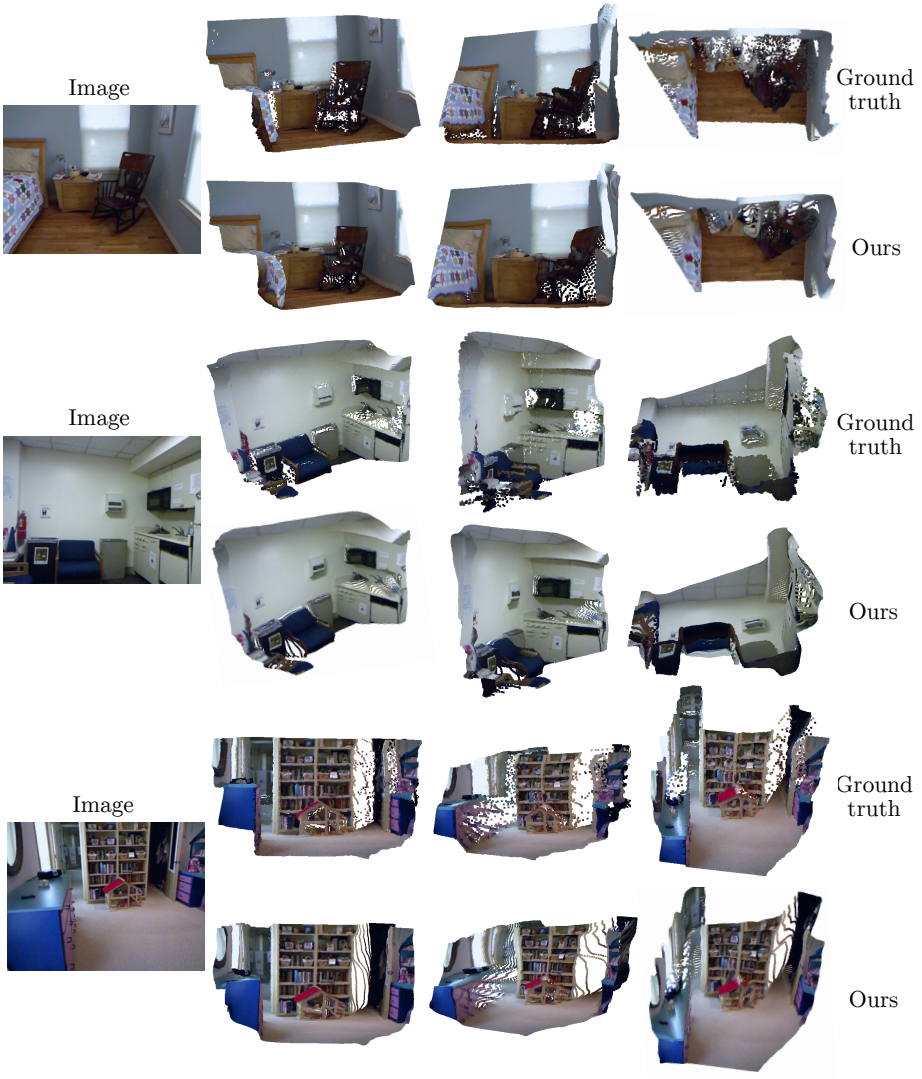

Fig. 3. Reconstructed point clouds for a set of randomly selected examples from NYU-Depth-v2. The images from the point clouds are captured in different camera poses to provide an overview of the 3D scenes.

2 Additional qualitative results of cross-dataset evaluation on SUN-RGBD

In this section, we use our pretrained model on NYU-Depth-v2 [17] to estimate depth values from SUN-RGBD images [8,19,20]. Dissecting the predicted depth maps, reconstruction point clouds and attention maps demonstrates the generalization ability of our proposed method.

2.1 Predicted depth maps

As shown in Figure 4, our model provides reasonable depth maps for SUN-RGBD examples although it has not been trained on this dataset. The geometry layout of the scene is retained, even in difficult scenarios (e.g. images in row (5) and (6) in Figure 4).

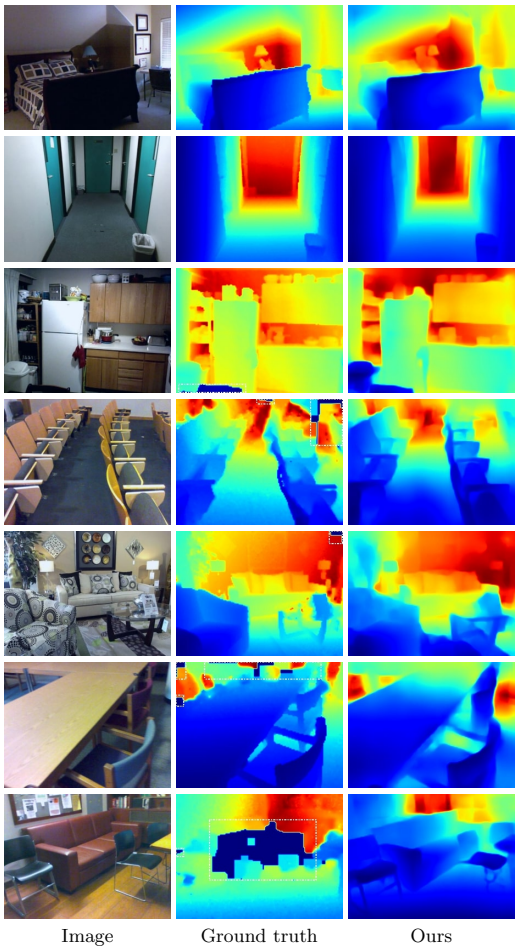

Fig. 4. Ramdomly examples from the SUN-RGBD test set. Areas in the white boxes show missing or incorrect depth values from the ground truth data.

2.2 Analyzing the predicted attention maps

As depicted in Figure 5, the proposed network learns to pay attention on planar-areas. At the green query point in the first image, the network concentrates on table surfaces as indicated by the warm color in its attention map. At the magenta query point, the model shifts its attention to the wall in the background.

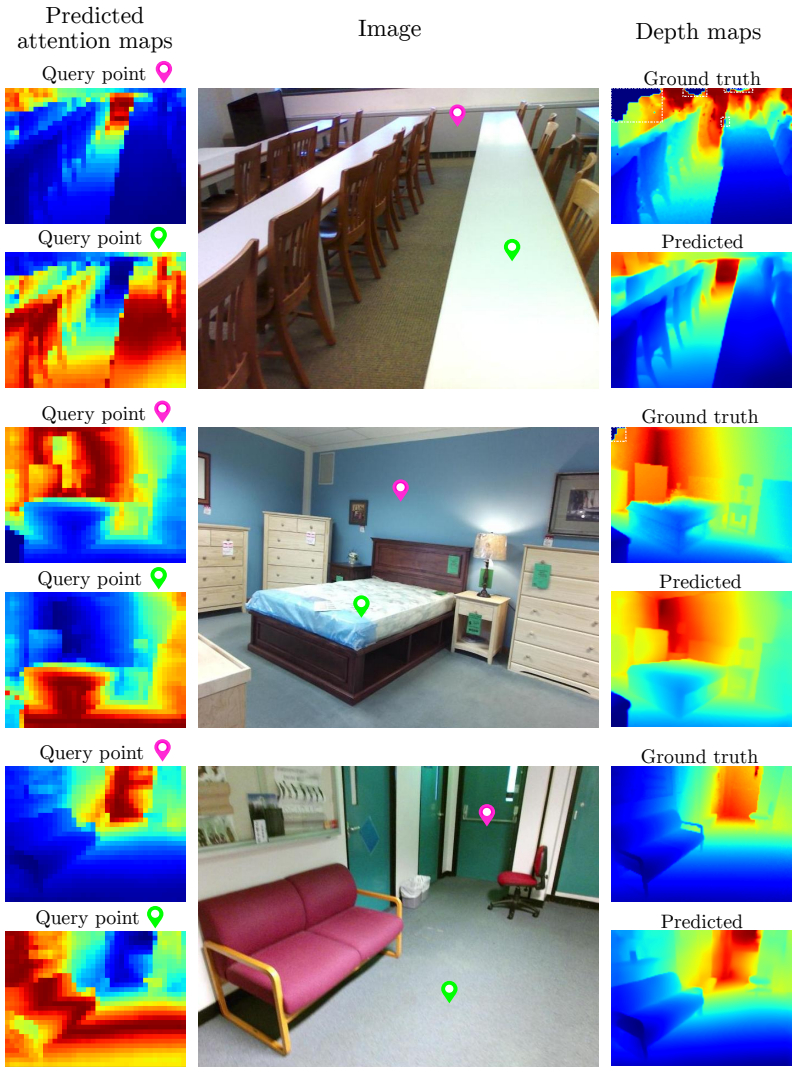

Fig. 5. Estimated attention and depth maps of our model train on NYU-Depth-v2 and test on SUN-RGBD. Left column presents predicted attention maps at indicated query points, while right column shows the predicted and ground truth depth maps. The input images are displayed in the middle.

2.3 Point cloud reconstruction

Figure 6 illustrates re-projected point clouds for SUN-RGBD examples produced by a model that is trained on NYU-Depth-v2. The produced point clouds are relatively close to the ground truth despite the fact that the model was trained on a different dataset.

Fig. 6. Reconstructed point clouds for randomly selected samples from the SUN-RGBD test set. The images of the point clouds are captured from different camera poses to provide an overview of the 3D scenes. The estimated depth maps are obtains from the model train with NYU-Depth-v2.

Table 1. iBims-1 benchmark. Metrics with ↓ mean lower is better and ↑ mean higher is better. Methods indicated with † and ‡ are trained using the AlexNet [10] or VGG [18], respectively.

Method	REL↓	log10↓	RMS↓	δ_1↑	δ_2↑	δ_3↑	ϵ^{plan}↓	ϵ^{orie}↓	ϵ^{acc}↓	ϵ^{comp}↓	ϵ^{0}↑	ϵ^{-}↓	ϵ^{+}↓
Eigen'14 [5]	0.32	0.17	1.55	0.36	0.65	0.84	7.70	24.91	9.97	9.99	70.37	27.42	2.22
Eigen'15 [4] †	0.30	0.15	1.38	0.40	0.73	0.88	7.52	21.50	4.66	8.68	77.48	18.93	3.59
Eigen'15 [4] ‡	0.25	0.13	1.26	0.47	0.78	0.93	**5.97**	17.65	4.05	8.01	79.88	18.72	1.41
Laina'16 [11]	0.26	0.13	1.20	0.50	0.78	0.91	6.46	19.13	6.19	9.17	81.02	17.01	1.97
Liu'15 [14]	0.30	0.13	1.26	0.48	0.78	0.91	8.45	28.69	**2.42**	**7.11**	79.70	14.16	6.14
Li'17 [12]	**0.22**	0.11	1.09	0.58	**0.85**	**0.94**	7.82	22.20	3.90	8.17	83.71	13.20	3.09
Liu'18 [13]	0.29	0.17	1.45	0.41	0.70	0.86	7.26	**17.24**	4.84	8.86	71.24	28.36	**0.40**
Ramam.'19 [15]	0.26	0.11	1.07	**0.59**	0.84	**0.94**	9.95	25.67	3.52	7.61	84.03	9.48	6.49
Ours	0.24	**0.10**	**1.06**	**0.59**	0.84	**0.94**	7.21	18.45	3.46	7.43	**84.36**	**6.84**	6.27

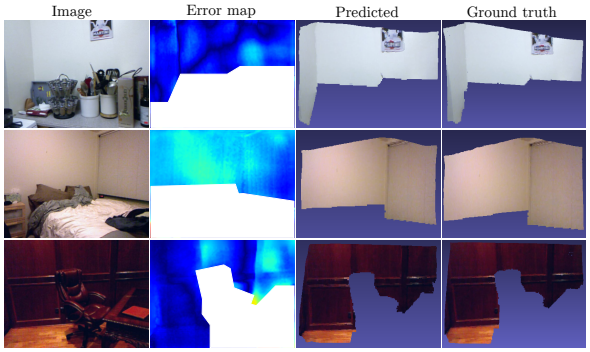

Fig. 7. Visualization of pixel around the planar areas. The second column shows the error map, while the third and forth column present the predicted and ground truth point cloud.

3 Additional analysis

3.1 Planarity error

Table 1 compares our model with monocular depth estimation methods that officially provides by the iBims-1 benchmark [9]. The results indicate that we outperform the recent methods [13,15] in most of the metrics (including plane related ones). Interestingly, the studies from Li et al. [12] and Liu et al. [14] although yield unfavourable results on NYU-Depth-v2 [17] seem generalize well on the iBims-1. Besides, we show qualitative results of our method around planar areas in Figure 7.

3.2 Non-local embedding space selection strategy

We empirically found that training the depth attention module using the cross-modulation in two embedding spaces yields superior to using a single embedding with double the number of features as shown in Table 2.

Table 2. Performance of our model using different types of embedding space.

Embedding space	REL↓	RMS↓	δ_1↑	δ_2↑	δ_3↑
Single embedding	0.115	0.432	0.868	0.975	0.994
Cross-modulation	**0.108**	**0.412**	**0.882**	**0.980**	**0.996**

4 Network architecture

This section gives complementary details regarding the network architecture and training process. The general structure of our network encompasses an encoder, a non-local depth attention module and a decoder. We construct the encoder by removing the average pooling and fully connected layer from the DRN-D-22 variation of the dilated residual networks [22,23]. Table 3 shows the detailed structure of our encoder where ***conv*** represents 2D convolutional layer with specific kernel-size (***k***), stride (***s***), and dilation (***d***). ***bn*** stands for batch normalization. **CH** is the number of output channels and ***RES*** is the spatial resolution of the output feature maps. ***basic-block*** represents the basic residual block with corresponding dilation.

As explained in the manuscript, we split the training scheme into three parts. It is worth to mention that during the first training phase, we initialize the encoder with pre-trained weights on the ImageNet [3]. Our experiments confirm that using the pretraining model improves accuracy and speed of convergence. The second and third training stages follow the procedure described in the main paper.

Table 3. Detail structure of the encoder.

Encoder

Input	**Operations**	***k***	***s***	***d***	**CH**	**RES**	**Output**
image	conv+bn+relu	7	1	1	16	228×304	layer0
layer0	conv+bn+relu	3	1	1	16	228×304	layer1
layer1	conv+bn+relu	3	2	1	32	114×152	layer2
layer2	basic-block	-	-	1	64	57×76	d-res-1a
d-res-1a	conv+bn	1	2	1	64	57×76	d-res-1b
d-res-1b	basic-block	-	-	1	64	57×76	layer3
layer3	basic-block	-	-	1	128	29×38	d-res-2a
d-res-2a	conv+bn	1	2	1	128	29×38	d-res-2b
d-res-2b	basic-block	-	-	1	128	29×38	layer4
layer4	basic-block	-	-	2	256	29×38	d-res-3a
d-res-3a	conv+bn	1	1	1	256	29×38	d-res-3b
d-res-3b	basic-block	-	-	2	256	29×38	layer5
layer5	basic-block	-	-	4	512	29×38	d-res-4a
d-res-4a	conv+bn	1	1	1	512	29×38	d-res-4b
d-res-4b	basic-block	-	-	4	512	29×38	layer6
layer6	conv+bn+relu	3	1	2	512	29×38	layer7
layer7	conv+bn+relu	3	1	1	512	29×38	layer8-X

The non-local depth attention module is the central component of our network with the detailed structure provided in Table 4. In that, **green, blue, orange** indicate the green, blue, and orange embedding spaces mentioned in the

manuscript. "$\odot$" denotes element-wise multiplication, "$\oplus$" indicates element-wise sum, and "$\otimes$" is the outer product. Layers denoted with ‡‡ imply reshaping and permuting the tensor to match the required shape for operation. Note that *green-1bn, green-1γ, green-1β* and *blue-1bn, blue-1γ, blue-1β* are generated at the same time as indicated by the dashed line.

Table 4. Internal structure of the non-local depth attention module.

Non-local depth attention module

Input	**Operations**	***k***	***s***	***d***	**CH**	**RES**	**Output**
layer8-X	conv	1	1	1	256	29×38	orange
layer8-X	conv	1	1	1	1024	29×38	green-1
green-1	bn	-	-	-	1024	29×38	green-1bn
green-1	conv	1	1	1	1024	29×38	green-1γ
green-1	conv	1	1	1	1024	29×38	green-1β
green-1bn, blue-1γ	$\odot$	-	-	-	1024	29×38	green-1bn-γ
green-1bn-γ, blue-1β	$\oplus$	-	-	-	1024	29×38	green-1-denorm
green-1-denorm	relu+conv	1	1	1	1024	29×38	green-2
layer8-X	conv	1	1	1	1024	29×38	blue-1
blue-1	bn	-	-	-	1024	29×38	blue-1bn
blue-1	conv	1	1	1	1024	29×38	blue-1γ
blue-1	conv	1	1	1	1024	29×38	blue-1β
blue-1bn, green-1γ	$\odot$	-	-	-	1024	29×38	blue-1bn-γ
blue-1bn-γ, green-1β	$\oplus$	-	-	-	1024	29×38	blue-1-denorm
blue-1-denorm	relu+conv	1	1	1	1024	29×38	blue-2
green-2‡‡, blue-2‡‡	$\otimes$	-	-	-	1	1102×1102	dav-1
dav-1	sigmoid	-	-	-	1	1102×1102	dav-2
dav-2, orange	$\otimes$	-	-	-	256	29×38	dav-3‡‡
dav-3	conv+bn	1	1	1	512	29×38	dav-4
dav-4, layer8-X	$\oplus$	-	-	-	512	29×38	layer8-Y

Unlike previous studies [4,11,6,16,24,15,7,1,21], we implement a straightforward decoder with two bilinear upsamplings follow by 2D convolutional layers and batch-normalizations. Finally, the upsampled feature maps are refined to produce the final depth map using two 2D convolutional layers. Table 5 provides a detailed structure of our decoder.

Table 5. Internal structure of the decoder where **bilinear** represents bilinear upsampling layers.

Decoder

Input	**Operations**	***k***	***s***	***d***	**CH**	**RES**	**Output**
layer8-Y	bilinear+conv+bn	3	1	1	256	57×76	up-conv-1
up-conv-1	bilinear+conv+bn	3	1	1	128	114×152	up-conv-2
up-conv-2	conv+bn+relu	5	1	1	64	114×152	refine-1
refine-1	conv	5	1	1	1	114×152	depth

5 Definitions of the evaluation metrics

All pixels in the predicted and ground truth depth maps with depth values in the range [0.0, 10.0] are considered valid and used to calculate the errors. We evaluate the performance for our model and for baselines on the NYU-Depth-v2 [17] using the following metrics:

- Mean absolute relative error (REL):

$$\frac{1}{T}\sum_{i=1}^{T}\frac{|\hat{d}_i - d|}{d} \tag{1}$$

- Root mean square error (RMS):

$$\sqrt{\frac{1}{T}\sum_{i=1}^{T}(\hat{d}_i - d_i)^2} \tag{2}$$

- Thresholded accuracy (δ_i):

$$max(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}) = \delta^i < 1.25^i \quad (i = 1, 2, 3) \tag{3}$$

where T is the number of valid pixel, $\hat{d}_i$ indicates the predicted depth value at pixel i, and d_i is the ground truth depth at pixel i. Lower REL and RMS values indicate better results, while the higher δ_1, δ_2 and δ_3, the better. In addition to the mean absolute relative error (REL), we assess model performance on ScanNet [2] and SUN-RGBD [19,8,20] using:

- Mean relative square error (sqREL):

$$\frac{1}{T}\sum_{i=1}^{T}\frac{(\hat{d}_i - d_i)^2}{d_i^2} \tag{4}$$

- Mean absolute error of the inverse depth (iMAE):

$$\frac{1}{T}\sum_{i=1}^{T}|\hat{p}_i - p_i| \tag{5}$$

- Root mean square error of the inverse depth (iRMSE):

$$\sqrt{\frac{1}{T}\sum_{i=1}^{T}(\hat{p}_i - p_i)^2} \tag{6}$$

- Scale-invariant mean square error (SI) [5]:

$$\frac{1}{2T}\sum_{i=1}^{T}\left[\log\hat{d}_i - \log d_i + \frac{1}{T}\sum_{j=1}^{T}(\log d_j - \log\hat{d}_j)\right]^2 \tag{7}$$

where $\hat{p}_i$ and p_i are the inverse value of the predicted and ground truth depth at pixel i, respectively. For the iBims-1 benchmark [9], besides the mentioned metrics, we evaluate model performance using:

- Root mean square error in logarithm space (log10):

$$\sqrt{\frac{1}{T}\sum_{i=1}^{T}(\log \hat{d}_i - \log d_i)^2} \tag{8}$$

- Flatness of the predicted 3D planes, which measures by the standard deviation of average distance between the predicted 3D points with its corresponding 3D plane (ϵ^{plan}):

$$\mathbb{V}\left[\sum_{\mathbf{P}_{k;i,j}\in P_k} d(\pi_k, \mathbf{P}_{k;i,j})\right] \tag{9}$$

- Orientation of the predicted 3D planes, which measures by angle between predicted and ground truth normal vectors (ϵ^{orie}):

$$acos(n_k, \hat{n}_k) \tag{10}$$

  where $\pi_k = (n_k, o_k)$ is the predicted plane with normal vector n and offset o, $\mathbf{P}_{k;i,j}$ is the 3D point with respect to plane k^{th}, and P_k indicates the annotated planes.
- Accuracy of depth boundary, which measures by multiplying the predicted edge map with a pre-defined distance map. (ϵ^{acc}):

$$\frac{1}{\sum_i \sum_j \hat{y}_{i,j}} \sum_i \sum_j e_{i,j} \cdot \hat{y}_{i,j} \tag{11}$$

- Completeness of depth boundary, which measures by multiplying the ground truth edge map with a predicted distance map. (ϵ^{comp}):

$$\frac{1}{\sum_i \sum_j y_{i,j}} \sum_i \sum_j \hat{e}_{i,j} \cdot y_{i,j} \tag{12}$$

  where y and $\hat{y}$ are the predicted and ground truth binary edge maps. e and $\hat{e}$ are the pre-defined distance maps which are calculated using the Euclidean distance transform.
- Directed depth errors ($\epsilon^0, \epsilon^-, \epsilon^+$) are measured based on a reference plane that located at 3 meters distance. The ϵ^0 is the percentage of predicted 3D points that lie in the reference plane. On the other hand, ϵ^- and ϵ^+ are the propositions of 3D points that lie in front or behind the reference plane.

References

1. Chen, X., Chen, X., Zha, Z.J.: Structure-aware residual pyramid network for monocular depth estimation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. pp. 694–700. AAAI Press (2019)
2. 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. pp. 5828–5839 (2017)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
4. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision. pp. 2650–2658 (2015)
5. 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. pp. 2366–2374 (2014)
6. Hao, Z., Li, Y., You, S., Lu, F.: Detail preserving depth estimation from a single image using attention guided networks. In: 2018 International Conference on 3D Vision (3DV). pp. 304–313. IEEE (2018)
7. Hu, J., Ozay, M., Zhang, Y., Okatani, T.: Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In: IEEE Winter Conf. on Applications of Computer Vision (WACV) (2019)
8. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3d object dataset: Putting the kinect to work. In: Consumer depth cameras for computer vision, pp. 141–165. Springer (2013)
9. Koch, T., Liebel, L., Fraundorfer, F., Körner, M.: Evaluation of cnn-based single-image depth estimation methods. In: Leal-Taix, L., Roth, S. (eds.) European Conference on Computer Vision Workshop (ECCV-WS). pp. 331–348. Springer International Publishing (2018)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
11. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV). pp. 239–248. IEEE (2016)
12. 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 International Conference on Computer Vision. pp. 3372–3380 (2017)
13. Liu, C., Yang, J., Ceylan, D., Yumer, E., Furukawa, Y.: Planenet: Piece-wise planar reconstruction from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2579–2588 (2018)
14. 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. pp. 5162–5170 (2015)
15. Ramamonjisoa, M., Lepetit, V.: Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. The IEEE International Conference on Computer Vision (ICCV) Workshops (2019)
16. Ren, H., El-khamy, M., Lee, J.: Deep robust single image depth estimation neural network using scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 37–45 (2019)

17. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision. pp. 746–760. Springer (2012)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)
20. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1625–1632 (2013)
21. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
22. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR) (2016)
23. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 472–480 (2017)
24. Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J.: Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4106–4115 (2019)