Decoder Modulation for Indoor Depth Completion

Dmitry Senushkin
d.senushkin@partner.samsung.com
Ilia Belikov
ilia.belikov@samsung.com
Anton Konushin
a.konushin@samsung.com

Samsung AI Center,
Moscow, Russia

Abstract

Accurate depth map estimation is an essential step in scene spatial mapping for AR applications and 3D modeling. Current depth sensors provide time-synchronized depth and color images in real-time, but have limited range and suffer from missing and erroneous depth values on transparent or glossy surfaces. We investigate the task of depth completion that aims at improving the accuracy of depth measurements and recovering the missing depth values using additional information from corresponding color images. Surprisingly, we find that a simple baseline model based on modern encoder-decoder architecture for semantic segmentation achieves state-of-the-art accuracy on standard depth completion benchmarks. Then, we show that the accuracy can be further improved by taking into account a mask of missing depth values. The main contributions of our work are two-fold. First, we propose a modified decoder architecture, where features from raw depth and color are modulated by features from the mask via Spatially-Adaptive Denormalization (SPADE)[25]. Second, we introduce a new loss function for depth estimation based on direct comparison of log depth prediction with ground truth values. The resulting model outperforms current state-of-the-art by a large margin on the challenging Matterport3D [1] dataset. The source code and the trained models are available at https://github.com/saic-vul/saic_depth_completion.

1 Introduction

In recent years, depth sensors have become an essential component of many devices, from self-driving cars to smartphones. However, the quality of modern depth sensors is still far from perfect. LiDaR systems provide accurate yet spatially sparse measurements while being quite expensive. Commodity-grade depth sensors, based on an active stereo with structured light (*e.g.* Microsoft Kinect), or Time-of-Flight (*e.g.* Microsoft Kinect Azure, and depth sensors in many smartphones), provide relatively dense estimations yet less accurate and within limited distance range. The depth completion methods aim at filling the missing values and correcting erroneous measurements using additional RGB images. With the rapid growth of the self-driving car industry, research on depth completion is mostly focused on processing LiDaR data in outdoor scenarios ([6, 31, 32]). Such methods can not be straightforwardly transferred to data from commodity-grade depth sensors.

Inspired by these observations, we develop a new approach to solving depth completion problem. We use a simple baseline model based on encoder-decoder architecture for semantic segmentation. Surprisingly, such a model is enough to achieve state-of-the-art accuracy on Matterport3D[3] depth completion benchmark. Then, we show that the accuracy can be further improved by taking into account a mask of missing depth values. We propose a modified decoder architecture, where features from raw depth and color are modulated by features from the mask via Spatially-Adaptive Denor-malization (SPADE)[25]. We also introduce a new loss function for depth estimation based on direct comparison of log depth prediction with ground truth values. Our model sets new state-of-the-art on Matterport3D depth completion dataset.

2 Related Work

In this section, we review works on several topics that are related to depth processing, or inspired our work. Namely, we cover depth estimation and semantic segmentation (as the most worked-out case of dense image regression).

Depth Completion. The pioneer works on depth completion adopted complicated heuristic algorithms for processing raw sensor data. These algorithms were based on compressive sensing theory [10] or used combined wavelet-contourlet dictionary [20]. Uhrig *et al.* [32] was the first to develop a successful learnable depth completion method based a convolution operation on sparse input. The boundaries for learnable methods were pushed further by image guidance ([6], [33], [37], [29]). Tang *et al.* [31] proposed an approach of training content-dependent and spatially-variant kernels for processing sparse depth features. Li *et al.* [17] suggested a multi-scale guided cascade hourglass architecture for depth completion. Chen *et al.* [5] came up with 2D-3D fusion pipeline based on continuous convolution. Apart from utilizing images, some of recently proposed methods take clues from surface normals ([26], [11], [35], [38]) and / or object boundaries ([11], [38]). However, the majority of the aforementioned works focus on LiDaR-based depth completion in outdoor scenarios and report results on the well-known KITTI benchmark [32]. There are only a few works on processing non-LiDaR indoor data obtained with Kinect sensors. Recently, Zhang *et al.* [38] introduced a large scale RGBD dataset for indoor depth completion, and Huang *et al.* [11] was the first to outperform original results on this dataset. These results were achieved by a complicated multi-stage method that relies on resource-exhausting preprocessing. In this paper, we propose a novel depth completion method that surpasses previous state-of-the-art while being light-weight and straightforward.

Depth Estimation and Dense Labelling. In a broad sense, depth completion is a dense labelling problem. Therefore, techniques that appeared to be effective for other dense labelling tasks might be useful for depth completion as well. Encoder-decoder architectures with skip connections originally developed for semantic segmentation [27] proved to be capable of solving a wide range of tasks. Another considerable approach is feature pyramid pooling [4] [39]. At the same time, light-weight networks such as[24] capable of running on a device in real time have broadened horizons for deep learning-driven applications.

Another dense labelling problem related to depth completion is single-view depth estimation. Generally speaking, while filling gaps in a depth map, a depth completion method

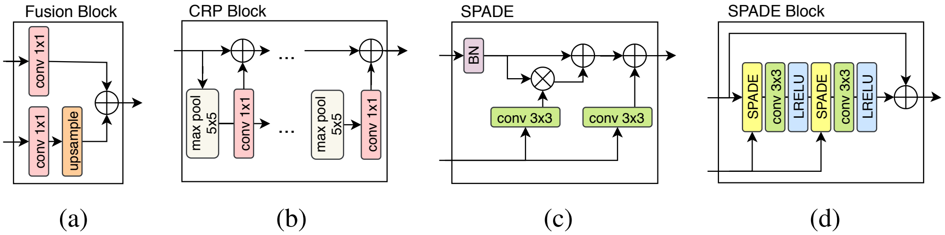

Figure 1: Basic block for our network: a) Fusion block [23], b) CRP block [23], c) SPADE [25] d) SPADE residual block [25], ReLU activation function has been replaced by Leaky ReLU.

actually solves depth estimation problem. Deep learning-based methods of depth estimation have evolved in recent years [2, 9, 19, 21]. By now, they have reached the accuracy of the depth sensors yet being able to run in real-time [34] or even on-a-chip [1]. However, in general, the acquisition of accurate ground-truth depth maps is impossible due to certain limitations of existing depth sensors. To overcome the aforementioned difficulties, different approaches focusing on data acquisition, data refinement and usage of alternative data sources, were proposed as well [18], [15].

3 Proposed Method

In this paper, we present a novel approach to solving the depth completion task. We introduce a simple yet efficient baseline for this task based on the light-weight network for semantic segmentation – *Light-Weight Refine Net* [23]. We also propose the decoder modulation process based on mask of missing areas that highlights the areas where the depth should be inpainted.

3.1 Baseline

The majority of modern depth completion methods rely on custom networks. These networks are difficult to train from scratch as the training process might be unstable. Furthermore, they have issues with overfitting that appears to be a serious problem considering the lack of depth completion datasets. A standard approach to address these difficulties is the usage of pre-trained backbones. In this study, we show that a network based on efficient baseline (*Light-Weight Refine Net*, LRN) [23] can be successfully adapted for depth completion task.

Light-Weight Refine Net is the encoder-decoder architecture that relies on pre-trained backbone and the efficient and light-weight decoder. In our experiments, we opted for EfficientNet [30] as a backbone. LRN decoder is built using chained residual pooling blocks (CRP) and fusion blocks (see Figure 1(b) and (a) respectively).

The simplest way to adapt this model for depth completion is to add a convolutional layer (commonly called a stem layer), so the network could accept 4-channel RGBD or 5-channel RGBD + mask input. With any of these modifications, the described model surpasses the current state-of-the-art method on Matterport3D dataset.

Despite the results being satisfactory in terms of accuracy metrics, output depth maps appear to be blurry. Since depth-sensing applications may require precise boundaries, we developed the decoder modulation branch that addresses this issue.

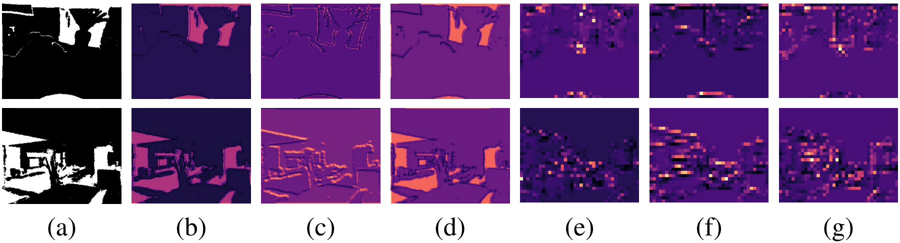

Figure 2: **Mask features**. (a) input mask , (b)-(d) high resolution features ($\frac{H}{2} \times \frac{W}{2}$), (e)-(f) mid resolution features ($\frac{H}{8} \times \frac{W}{8}$). Large values are highlighted. Features of filled regions tend to be small and constant while for unfilled areas features might take values in a wide range. One can also notice large activation values marking the boundaries of objects that might also be helpful for depth inpainting.

3.2 Decoder modulation

The existing methods are not capable of producing sharp-edged, non-blurry depth inpainting. We attribute this feature to the usage of the same processing strategy for areas with valid depth measurements and areas without valid depth measurements. Ideally, a depth completion network should act like an auto-encoder inside valid areas and like a depth estimation model inside areas with missing or erroneous values. We propose an architecture which in theory is able to select the proper operation mode for valid and invalid areas.

The model should operate in the inpainting mode either for areas with missing values or incorrect measurements. Based on the data coming directly from a depth sensor, we can determine unfilled areas, while the information about incorrect measurements is not available in the general case. Thus, we use the mask of missing values as a lower bound estimate of the areas where the network should perform inpainting.

To distinguish between depth auto-encoder or depth estimation operating mode, we suggest applying Spatially-Adaptive Denormalization (SPADE)[25] based on a mask to the features extracted from RGBD image. For a given image partition, SPADE learns a dense affinity transformation for batch normalization [12] statistics for each image area. SPADE is supposed to learn different transforms for filled and unfilled areas of a depth map. In support of this hypothesis, we have observed that mask features differ for filled and unfilled areas (Figure 2). Following our concept, the proper operating mode could be chosen considering these features.

The high-level network design is shown in Figure 3. Decoder modulation branch of our network consists of a simple mask encoder composed from convolutions with leaky ReLU activations. This mask encoder is connected through bilinear upsampling to SPADE blocks where the modulation of the decoder features is performed. The design of these blocks (Figure 1(d)) is derived from the original paper by Park *et al.* [25]. Note that all our models use an RBGD image and a mask as input and do not require pre-computed surface normals, boundaries, semantic maps and other supplementary information.

3.3 Loss function

Standard regression losses such as l_1 or l_2 applied in a real domain often lead to smoothed depth maps as they penalize all errors equally. For depth completion, an accurate estimate of the distance to a close object might be more important than to a distant one, so logarithmic

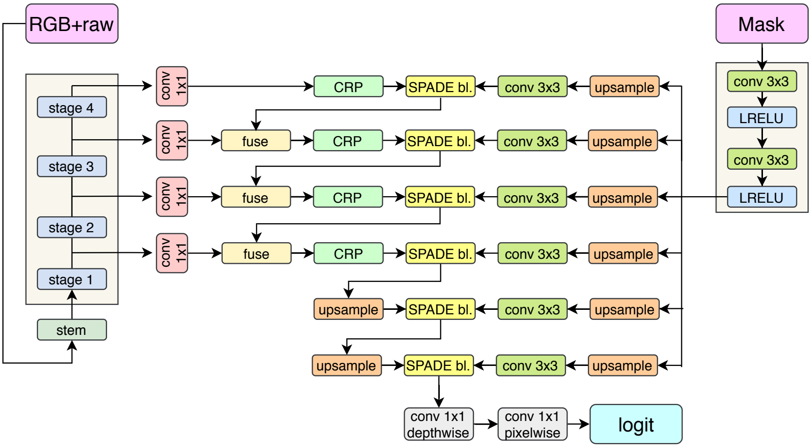

Figure 3: Network architecture

domain seems to be a natural choice. Following Eigen *et al.* [8], we calculate losses in log scale. However, we predict depth in log scale directly. Eventually, we use the following loss function:

$$\mathcal{L}(d_i, d_i^*) = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} \left| \log d_i - d_i^* \right|, \tag{1}$$

where $d_i^* = f_\theta(x_i)$ – predicted logits, f_θ – network function and d_i is a ground truth. In general, minimization of this loss may be reformulated as maximization of likelihood with specific depth distribution. See appendix 5 for details.

4 Experimental results

In this section, we present the results that prove the effectiveness of our method in different set-ups. We estimate the contribution of each component of the proposed method in ablation studies. In addition, we investigate the generalizing ability and robustness of the proposed method.

4.1 Datasets

Matterport3D [38]. This large-scale indoor dataset consists of 110,000 RGB images of resolution 320×256 covering 90 scenes. For each RGB image, there is a time-synchronized raw depth map received via Matterport camera setup with Microsoft Kinect on board. Apart from that, ground truth depth maps are rendered from reconstructed meshes. Since both raw and ground truth depth maps are available, Matterport3D can be used to train a depth completion model.

NYUv2 [22]. NYUv2 is an indoor dataset which consists of two parts. NYUv2 Depth is manually annotated for depth estimation and depth completion. It contains only 1449 RGB images, of which 795 comprise the training set and 654 are left for validation. This amount of data is insufficient for training a depth completion models. At the same time, NYUv2 Raw,

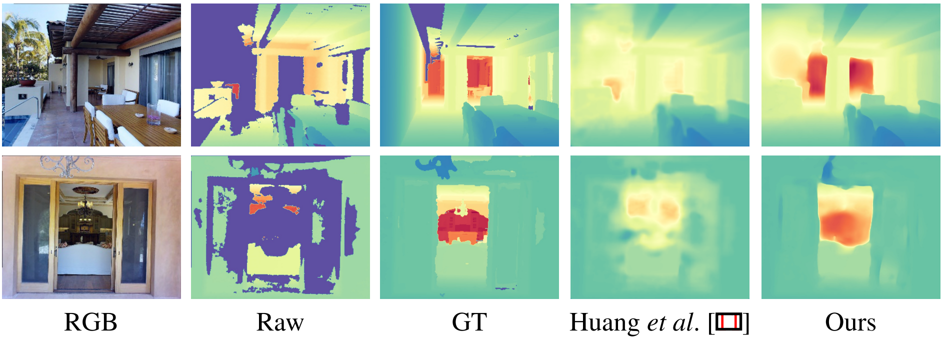

Figure 4: Visual results on Matterport3D TEST compared with the previous SoTA [11]. Mask modulation prevents simple interpolation when crossing the border with a large difference in depth and saves information from raw depth with almost no changes.

which is not annotated, contains more than 300,000 training RGB images accompanied by raw depth maps from Microsoft Kinect sensor. For this dataset, neither ground truth dense depth maps nor reconstructed meshes are available, therefore it is inapplicable for training a depth completion model. However, it can be used to investigate the behaviour of different depth completion methods.

4.2 Evaluation metrics

Following the standard evaluation protocol for indoor depth completion, we use root mean squared error (RMSE), mean absolute error (MAE), δ_i and SSIM. The δ_i denotes the percentage of predicted pixels where the relative error is less than a threshold i. Specifically, i is chosen to be equal to 1.05, 1.10, 1.25, 1.25^2 and 1.25^3 separately for evaluation. Here, the larger is i, the more sensitive is the δ_i metric. Larger values of δ_i reflect a more accurate prediction. RMSE and MAE directly measure absolute depth accuracy. RMSE is more sensitive than MAE and is chosen to be the main metric for ranking models. In general, our test pipeline is similar to Huang *et al.* [11] [1].

4.3 Experimental setup

We used Adam optimization algorithm [13] with initial learning rate set to 10^{-4}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and without weight decay. The *EfficientNet* backbone [30] was initialized with weights pre-trained on ImageNet [7]. After that, all the models were trained end-to-end for 200 epochs with batch size of 32. All RGBD inputs were resized to 320×256 and normalized by mean and standard deviation. For RGB, we used pre-computed ImageNet statistics. For depth, we calculated statistics of the Matterport3D training subset. In addition, we encoded missing areas with 1's, and filled areas with 2's on mask. This manner would facilitate the training of biases in the mask encoder.

We implemented all models in Python 3.7 using PyTorch 1.4 library. We used EfficientNet from `Segmentation Models Pytorch` [36]. For experiments, a single Nvidia Tesla P40 GPU was used.

[1] The evaluation code is available on official page https://github.com/patrickwu2/Depth-Completion. For fair comparison, we opted for the evaluation procedure based on official code.

model	**rmse**	**mae**	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$	**SSIM**
Bilateral	1.978	0.774	0.385	0.497	0.613	0.689	0.730	0.507
MRF	1.675	0.618	0.506	0.556	0.651	0.780	0.856	0.692
FCN	1.262	0.517	0.397	0.527	0.681	0.808	0.868	0.605
Huang *et al.* [11]	1.092	0.342	0.661	0.750	0.850	0.911	0.936	0.799
Zhang *et al.* [38]	1.316	0.461	0.657	0.708	0.781	0.851	0.888	0.762
inverted Huber loss								
LRN-b4	1.059	0.299	0.716	0.802	**0.889**	**0.931**	0.948	**0.843**
DM-LRN-b4	1.040	0.296	0.721	0.808	0.888	0.929	0.947	**0.843**
log depth prediction l_1 loss								
LRN-b4	1.052	0.304	0.720	0.810	0.887	0.929	0.948	0.840
LRN-b4 + mask	1.036	0.304	0.726	0.810	0.885	0.930	**0.949**	0.838
DM-LRN-b4	**1.001**	**0.289**	**0.739**	**0.815**	**0.889**	0.930	0.948	0.842

Table 1: **Matterport TEST**. Our best model with decoder modulation (DM prefix) outperforms current state-of-the-art. For FCN, MRF, Bilateral metrics are adopted from Huang *et al.* [11].

4.4 Performance and comparison

We conducted numerous experiments on Matterport3D dataset, using standard train/test split proposed by Zhang *et al.* [38]. In addition, we divided training data into train and validation subsets. According to Table 1, our best model noticeably outperforms previous state-of-the-art on Matterport3D.

Figure 4 represents visual results on Matterport3D test set. As one can observe, our model stores information from raw depth with minimal changes where the depth value is available. Region-mask-based affine feature transformations in the decoder prevent object boundaries from blurring.

4.5 Ablation study

To investigate the impact of each proposed component on the final performance, we conduct ablation studies on the Matterport3D test dataset. The quantitative comparisons are summarized in Table 1.

Loss function. Firstly, we investigate how the choice of the loss function affects the performance. To give an alternative for the proposed log l_1 loss, we train our model with the inverted Huber loss proven to be effective in solving depth estimation task [24]. According to Table 1, opting for the proposed loss function leads to an improvement in terms of RMSE, $\delta_{1.05}$ and $\delta_{1.10}$ that are the most sensitive to minor deviations. At the same time, using Huber loss helps to achieve better results according to the values of $\delta_{1.25}, \delta_{1.25^2}$ and SSIM. In general, the proposed loss function puts an emphasis on details while Huber loss yields less accurate estimates but reduces the number of outliers.

Decoder modulation. Secondly, we studied different strategies of masks utilization. We tested a decoder modulation branch against a more straightforward approach such as appending mask to RGBD inputs, or even not using the mask at all. In total, we trained three models: original LRN taking RGBD as an input (LRN-b4), LRN taking the concatenated RGBD +

model	**rmse**	**mae**	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$	**SSIM**
LRN-b0	1.110	0.327	0.718	0.800	0.881	0.927	0.946	0.834
LRN-b1	1.072	0.310	0.731	**0.816**	0.887	0.930	0.947	0.840
LRN-b2	1.069	0.313	0.730	0.809	0.885	0.928	0.947	0.838
LRN-b3	1.021	0.299	0.735	0.813	0.884	0.929	0.948	0.840
LRN-b4	1.052	0.304	0.720	0.810	0.887	0.929	0.948	0.840
DM-LRN-b0	1.077	0.309	0.722	0.804	0.884	0.928	0.947	0.837
DM-LRN-b1	1.035	0.301	0.730	0.808	0.886	0.928	0.947	0.839
DM-LRN-b2	1.017	0.293	0.726	0.809	**0.889**	**0.931**	**0.949**	0.841
DM-LRN-b3	1.040	0.302	0.731	0.813	0.888	**0.931**	**0.949**	0.840
DM-LRN-b4	**1.001**	**0.289**	**0.739**	0.815	**0.889**	0.930	0.948	**0.842**

Table 2: **Matterport TEST**. Different EfficientNet backbones. B3 configuration demonstrates an unexpected behavior.

mask as an input (LRN-b4+mask), and LRN with decoder modulation branch (DM-LRN-b4). According to the values of RMSE, our model performs better than the others. Moreover, the model with concatenated RGBD and mask inputs appeared to perform almost as good as the model without the mask, demonstrating that usage of such a simple masks utilization strategy is insufficient.

Backbone. Moreover, we conducted a series of experiments with different EfficientNet backbones. Our method outperforms the standard approach for all the backbones except b3 (which we believe to be an outlier). For large EfficientNet configurations (b5 and upper), as well as for LRN and DM-LRN, training process appeared to be unstable (models diverge). All results are presented in Table 2.

4.6 Generalization ability and robustness

Commodity-grade depth sensor output depends massively on lightning conditions, occlusions, presence of reflecting or transparent surfaces, etc. Accordingly, a good depth completion method should be able to process data with various level of errors. Besides accuracy, robustness and ability for generalization are major requirements for a depth completion method.

NYUv2 Depth. To study the generalization ability of our model, we evaluated it on original NYUv2 Depth test subset following standard protocol [31], so we could compare it with existing depth estimation methods. As shown in Table 3, our model demonstrates competitive results without being trained on this dataset. However, NYUv2 Depth raw images have significant artificial outliers at the border, namely a frame about 10 pixels wide at the edges. The errors of our model are mainly related to these areas (see Figure 6).

NYUv2 Raw. Next, we investigate robustness to the different types of noise. In this experiment, we evaluate our method only against current state-of-the-art depth completion method [11]. We use NYUv2 Raw subset in order to avoid artifacts at the edges and to obtain statistically significant results. Eventually, we evaluate our method on all samples from the scenes present in the original NYUv2 Depth test subset (6911 samples in total).

model	**rmse**	**mae**	$\delta_{1.05}$	$\delta_{1.10}$	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$	**SSIM**
Depth estimation (trained on NYUv2)								
Saxena *et al.* [28]	1.214	–	–	–	0.447	0.745	0.897	–
Eigen *et al.* [8]	0.641	–	–	–	0.769	0.750	0.988	–
Laina *et al.* [14]	0.573	–	–	–	0.811	0.953	0.988	–
Lee *et al.* [16]	0.392	–	–	–	0.885	0.978	0.994	–
Depth completion (trained on Matterport3D)								
Huang *et al.* [11]	0.425	0.203	0.647	0.781	0.920	0.982	0.995	0.659
DM-LRN-b4	0.649	0.239	0.731	0.796	0.883	0.943	0.971	0.710

Table 3: **NYUv2 Depth TEST**. Our model pretrained on Matterport3D in completion regime compared with recent depth estimation models trained on NYUv2. Due to the artifacts at the edges our model is inferior to models that perform simple interpolation. Nevertheless, it surpasses them in terms of sensitive metrics $\delta_{1.05,1.10}$.

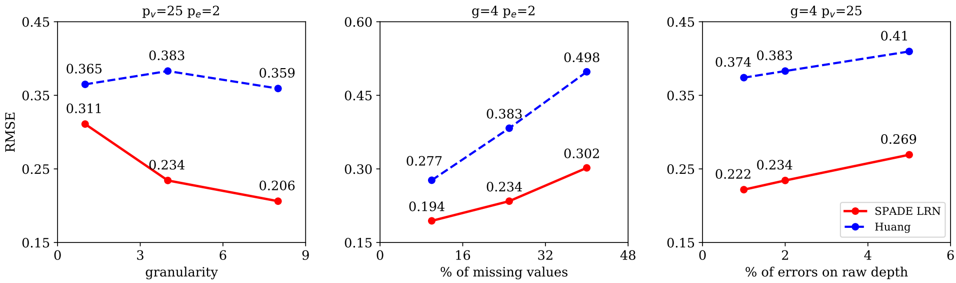

Figure 5: **NYUv2 Raw TEST**. Analysis of the model behavior with respect to granularity, fullness and errors on raw depth.

Since there is no ground truth depth in NYUv2 Raw dataset, we use raw sensor data instead. We pass a corrupted raw depth as an input, while original raw depth serves as a target (see Figure 7). The process of spoiling the sensor data is controlled by three parameters: granularity g, void percentile p_v, and error percentile p_e. We use the granularity parameter to adjust the form of the holes.

We initialize an auxiliary image with random pixel values sampled from uniform distribution and apply Gaussian filter with $\sigma = g$. Pixels that fall into the percentile p_v are marked as invalid on source depth map. We also add a number of erroneous values to the input depth image by scaling depth in some areas. Following previous pattern, we create a mask using percentile p_e and granularity g. For masked region, a random value is sampled from $\{0.5m, 2m\}$, where m is the average depth across the masked depth region. Then all pixels within the area are set to the selected value.

Fig. 5 shows how the accuracy depends on the value of each parameter. The higher the granularity of the areas with missing depth is (with the total amount of missing values being fixed), the better is the performance of our model in terms of RMSE. With an increase in the total amount of missing values, RMSE of the proposed method increases twice as slow compared to the previous state-of-the-art. Both methods demonstrate similar behaviour in other aspects, although the proposed method surpasses previous state-of-the-art with any granularity, fullness and percentage of errors.

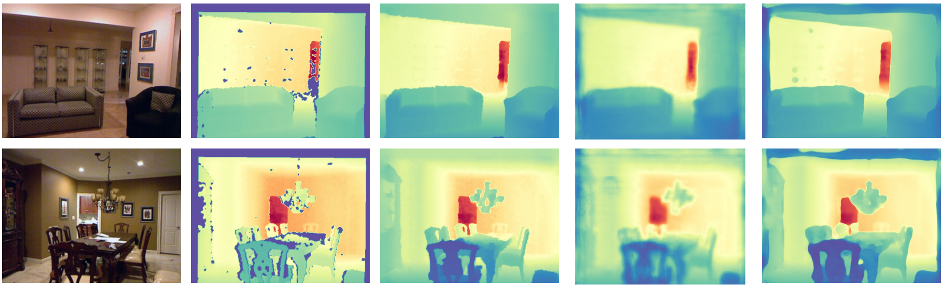
RGB Raw GT Huang *et al.* [□] Ours

Figure 6: Visual results on NYUv2 Depth official test compared to the previous SOTA [□]. Our model produces more precise and less blurry prediction but suffers from artifacts at the edges.

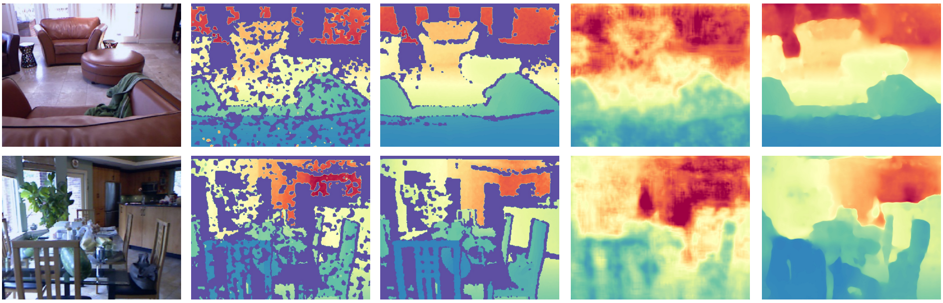
RGB Raw GT Huang *et al.* [□] Ours

Figure 7: Visual results on NYUv2 Raw unofficial test compared to the previous SOTA [□]. Our model produces more precise and less blurry prediction.

5 Conclusion

We propose a novel approach to depth completion. We build our model based on the Light-Refinement Network architecture with an original decoder modulation branch. In this branch, we use SPADE blocks to switch between depth estimation and depth auto-encoding operating modes. We trained the proposed network on Matterport3D and achieved new state-of-the-art results for this dataset. We also evaluated our method on NYUv2, where it demonstrated strong generalizing ability on real data and robustness to different types of noise.

Appendix

Loss function

We construct the loss function based on the principle of maximum likelihood. Namely, we assume that, given RGBD image x, the corresponding target depth d has an exponential distribution:

$$p(d|x,\theta) \propto \exp\left[-\left|\log\frac{d}{e^{f_\theta(x)}}\right|\right] = \begin{cases} \frac{d}{e^{f_\theta(x)}}, & d < e^{f_\theta(x)} \\ \frac{e^{f_\theta(x)}}{d}, & d > e^{f_\theta(x)} \end{cases} \tag{2}$$

where f_θ is neural network function. The data $\{(x_i, d_i), i \in \mathcal{O}\}$ is assumed to be drawn independently from the distribution. $\mathcal{O}$ – index set of pixels with existing target depth. Then, the negative log likelihood function is given by

$$-\log p(\mathbf{d}|\mathbf{x}, \theta) \propto \sum_{i \in \mathcal{O}} \left| \log \frac{d_i}{e^{f_\theta(x_i)}} \right| = \sum_{i \in \mathcal{O}} \left| \log d_i - f_\theta(x_i) \right| \tag{3}$$

If we use 3 as loss function, different pixels would affect the final result differently. Aiming to equalize this impact, we additionally use cardinality of set $\mathcal{O}$ as normalization factor. The task of maximizing likelihood 3 is reduced to the task of minimizing empirical risk with loss function:

$$\mathcal{L}(d_i, d_i^*) = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} \left| \log d_i - \hat{d}_i^* \right|, \tag{4}$$

where $\hat{d}_i^* = f_\theta(x_i)$ – predicted logits.

As can be seen from the formula 2, by minimizing loss function 4 we optimize relative δ metrics, unlike most existing regression approaches based on Euclidean or Manhattan distances in actual domain.

References

[1] Ambarella cvflow technology overview. https://www.ambarella.com/technology/technology-overview. Accessed: 2018-10-30.

[2] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *arXiv preprint arXiv:1811.06152*, 2018.

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. URL http://arxiv.org/abs/1606.00915.

[5] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[6] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. *ArXiv*, abs/1911.05377, 2019.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014.

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[10] S. Hawe, M. Kleinsteuber, and K. Diepold. Dense disparity maps from sparse disparity measurements. In *2011 International Conference on Computer Vision*, pages 2126–2133, 2011.

[11] Yu-Kai Huang, Tsung-Han Wu, Yueh-Cheng Liu, and Winston H. Hsu. Indoor depth completion with boundary consistency and self-attention. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct 2019. doi: 10.1109/iccvw.2019.00137. URL http://dx.doi.org/10.1109/ICCVW.2019.00137.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICMLâĂŹ15, page 448âĂŞ456. JMLR.org, 2015.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[14] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016. URL http://arxiv.org/abs/1606.00373.

[15] Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *CoRR*, abs/1907.01341, 2019. URL http://arxiv.org/abs/1907.01341.

[16] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *CoRR*, abs/1907.10326, 2019. URL http://arxiv.org/abs/1907.10326.

[17] Ang Li, Zejian Yuan, Yonggen Ling, Wanchao Chi, shenghao zhang, and Chong Zhang. A multi-scale guided cascade hourglass network for depth completion. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[18] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CoRR*, abs/1804.00607, 2018. URL http://arxiv.org/abs/1804.00607.

[19] Zhengfa Liang, Yiliu Feng, YGHLW Chen, and LQLZJ Zhang. Learning for disparity estimation through feature constancy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2811–2820, 2018.

[20] L. Liu, S. H. Chan, and T. Q. Nguyen. Depth reconstruction from sparse samples: Representation, algorithm, and sampling. *IEEE Transactions on Image Processing*, 24 (6):1983–1996, 2015.

[21] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[22] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[23] Vladimir Nekrasov, Chunhua Shen, and Ian D. Reid. Light-weight refinenet for real-time semantic segmentation. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 125. BMVA Press, 2018. URL http://bmvc2018.org/contents/papers/0494.pdf.

[24] Vladimir Nekrasov, Thanuja Dharmasiri, Andrew Spek, Tom Drummond, Chunhua Shen, and Ian Reid. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. pages 7101–7107, 05 2019. doi: 10.1109/ICRA.2019.8794220.

[25] T. Park, M. Liu, T. Wang, and J. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2332–2341, 2019.

[26] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[27] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a. (available on arXiv:1505.04597 [cs.CV]).

[28] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5): 824–840, May 2009. doi: 10.1109/TPAMI.2008.132.

[29] S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor. Dfusenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 13–20, 2019.

[30] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California,

USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/tan19a.html.

[31] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *ArXiv*, abs/1908.01238, 2019.

[32] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. *2017 International Conference on 3D Vision (3DV)*, pages 11–20, 2017.

[33] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6, 2019.

[34] Wofk, Diana and Ma, Fangchang and Yang, Tien-Ju and Karaman, Sertac and Sze, Vivienne. FastDepth: Fast Monocular Depth Estimation on Embedded Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[35] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[36] Pavel Yakubovskiy. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2020.

[37] Yanchao Yang, Alex Wong, and Stefano Soatto. Dense depth posterior (ddp) from single image and sparse range. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[38] Yinda Zhang and Thomas A. Funkhouser. Deep depth completion of a single RGB-D image. *CoRR*, abs/1803.09326, 2018. URL http://arxiv.org/abs/1803.09326.

[39] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016. URL http://arxiv.org/abs/1612.01105.