S2R-DepthNet: Learning a Generalizable Depth-specific Structural Representation
Supplementary Material

Xiaotian Chen[1*] Yuwang Wang[2†] Xuejin Chen[1] Wenjun Zeng[2]
[1] University of Science and Technology of China [2] Microsoft Research Asia
ustcxt@mail.ustc.edu.cn {yuwwan,wezeng}@microsoft.com xjchen99@ustc.edu.cn

1. Implementation details of encoder $\mathcal{E}_s$

We adopt an image translation framework [7] to train the encoder $\mathcal{E}_s$ of the STE module, as shown in Figure 1. To make the $\mathcal{E}_s$ generalizable to various style images, we choose the Painter By Numbers (PBN) dataset [1] with a large style variation as the target domain for image translation and a synthetic dataset as the source domain. Given a source domain image x^s and a target domain image x^t, we first use a shared $\mathcal{E}_s$ to extract the structure code for both the source and target domains denoted as c_a^s and c_a^t respectively. Then the domain specific style encoders $\mathcal{E}_{style}^s$ and $\mathcal{E}_{style}^t$ generate style codes c_b^s and c_b^t for the source and target domains respectively. The encoded structure code and style code are complementary for each domain. Combining these two codes, the original images from source and target domains can be restored by decoders G_s and G_t respectively. In addition, we also combine c_a^s extracted from the source domain dataset with a randomly sampled style latent code c_b^t from the prior distribution $q(c_b^t) \sim \mathcal{N}(0, I)$. We use G_t to produce the final output image $x_{s\to t}$. Similarly, we also combine c_a^t extracted from the target domain dataset with a randomly sampled style latent code c_b^s from the prior distribution $q(c_b^s) \sim \mathcal{N}(0, I)$. G_s is used to produce the final output image $x_{t\to s}$.

Given an image sampled from the data distribution, because the two parts are complementary, we decode them back to the original image by minimizing

$$\mathcal{L}_{recon}^{x^s} = \mathbb{E}_{x^s\sim p(x^s)}[||\mathcal{G}_s(\mathcal{E}_s(x^s), \mathcal{E}_{style}^s(x^s)) - x^s||_1]. \tag{1}$$

After obtaining the final image $x_{s\to t}$ through cross-domain translation, we input it to the shared structure encoder and the specific style encoder, so that the obtained latent code can also be reconstructed by minimizing

$$\mathcal{L}_{recon}^{c_a^s} = \mathbb{E}_{c_a^s\sim p(c_a^s), c_b^t\sim q(c_b^t)}[||\mathcal{E}_s(\mathcal{G}_t(c_a^s, c_b^t)) - c_a^s||_1], \tag{2}$$

and

$$\mathcal{L}_{recon}^{c_b^t} = \mathbb{E}_{c_a^s\sim p(c_a^s), c_b^t\sim q(c_b^t)}[||\mathcal{E}_{style}^t(\mathcal{G}_t(c_a^s, c_b^t)) - c_b^t||_1], \tag{3}$$

where $q(c_b^t)$ is the prior $N(0, I)$. We also use the adversarial loss to match the distribution of the translated images to the PBN data distribution as

$$\begin{aligned}\mathcal{L}_{adv}^{x^t} = &\mathbb{E}_{c_a^s\sim p(c_a^s), c_b^t\sim q(c_b^t)}[log(1 - D_t(x_{s\to t}))]+ \\ &\mathbb{E}_{x_t\sim p(x_t)}[logD_t(x_t)],\end{aligned} \tag{4}$$

where D_t is a discriminator that tries to distinguish translated images from painting images. For the other branch, we follow similar pipeline to design the losses. By minimizing these loss functions, the structure code and style code of the image can be effectively disentangled. The total training objective is:

$$\begin{aligned}\min_{(\mathcal{E}_s,\mathcal{G}_s,\mathcal{G}_t,D_s,D_t)} \max_{(D_s,D_t)} \mathcal{L}_{total} = \\ \mathcal{L}_{adv}^{x^s} + \mathcal{L}_{adv}^{x^t} + \lambda_1(L_{recon}^{x^s} + L_{recon}^{x^t})+ \\ \lambda_2(\mathcal{L}_{recon}^{c_b^t} + L_{recon}^{c_a^s})+ \\ \lambda_3(\mathcal{L}_{recon}^{c_b^s} + L_{recon}^{c_a^t}),\end{aligned} \tag{5}$$

where $\lambda_1, \lambda_2, \lambda_3$ are weights for the three losses respectively.

2. Visualize structure maps of images with different styles but the same structure code

We show the structure maps of images with different styles but the same structure codes in Figure 2. The structure maps are generated by feeding images of different

*This work was done when Xiaotian Chen was an intern at Microsoft Research Asia.
†Corresponding author.
[1]https://www.kaggle.com/c/painter-by-numbers

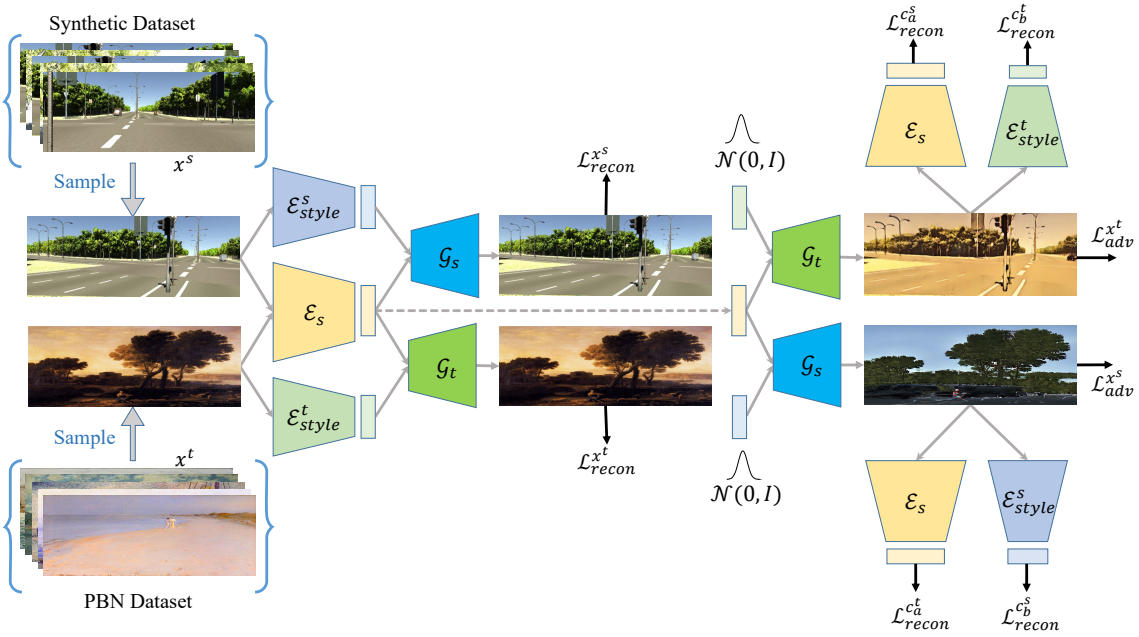

Figure 1. An overview of training process the encoder of STE module.

styles but same structure code into the STE module. It can be seen from Figure 2 that although the image styles are different, the generated structure maps are very similar, which further proves that the STE module can ignore the style information and only extract the structure information.

3. Performance on NYU Depth v2 for semi-supervised setting

We show the performance of our method under the semi-supervised setting on NYU Depth v2 in Table 1. It can be seen from Table 1 that even though our method only uses 500 labeled real data (0.42% of the total dataset) to fine-tune the domain generalization model, our method outperforms semi-supervised methods [14] under the same settings, and even outperforms some fully supervised methods [10, 2]. It is worth noting that Laina *et al*. [9] use more than 120k data to train the depth predictor, and we still surpass their approach on the RMSE.

4. Datasets and Evaluating Metrics

In the following section, we will introduce these datasets and evaluating metrics used in our experiments in detail.

vKITTI [4] is a photo-realistic synthetic dataset, that contains 21260 image-depth paired generated from five different virtual worlds in diverse urban settings and weather conditions. The image resolution of this dataset is 375×1242. To train our network, we follow prior works [15, 13], to randomly select 20760 image-depth pairs as our train datasets.

Table 1. Performance on NYU Depth v2 for semi-supervised setting with best results marked in bold.

Method	Abs Rel	RMSE	log 10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Li *et al*. [10]	0.232	0.821	0.094	0.621	0.886	0.968
Eigen *et al*. [2]	0.215	0.907	-	0.611	0.887	0.971
Laina *et al*. [9]	0.127	0.573	0.055	0.811	0.953	0.988
Zhao *et al*. [14]	0.186	0.710	-	0.712	0.917	0.977
Ours	**0.168**	**0.544**	**0.069**	**0.764**	**0.945**	**0.984**

We downsample all the images to 192×640 and data augmentation is conducted including random horizontal flipping with a probability of 0.5, rotation with degrees in $[-5°, 5°]$, and brightness adjustment. Because the ground truth of KITTI and vKITTI are significantly different, the maximum depth of the vKITTI dataset is $655.35m$, and the maximum depth value of KITTI is $80m$. In order to reduce the influence of ground truth differences, the depth value of vKITTI is usually clipped to $80m$ [15, 8].

SUNCG [12] is an indoor synthetic dataset, which contains 45622 3D houses with various room types. The image size is 480×640. Following previous studies [15], we chose the camera locations, poses, and parameters based on the distribution of real NYU Depth v2 dataset [11] and retained valid depth maps using the same criteria as Zheng *et al*. [15]. 130190 image-depth pairs are downsampled to 192×256 and used for training.

KITTI [5] is an outdoor real dataset, which is built for various computer vision tasks for autonomous driving. The images and depth maps are captured for outdoor scenes through a LiDAR sensor deployed on a driving vehicle. The

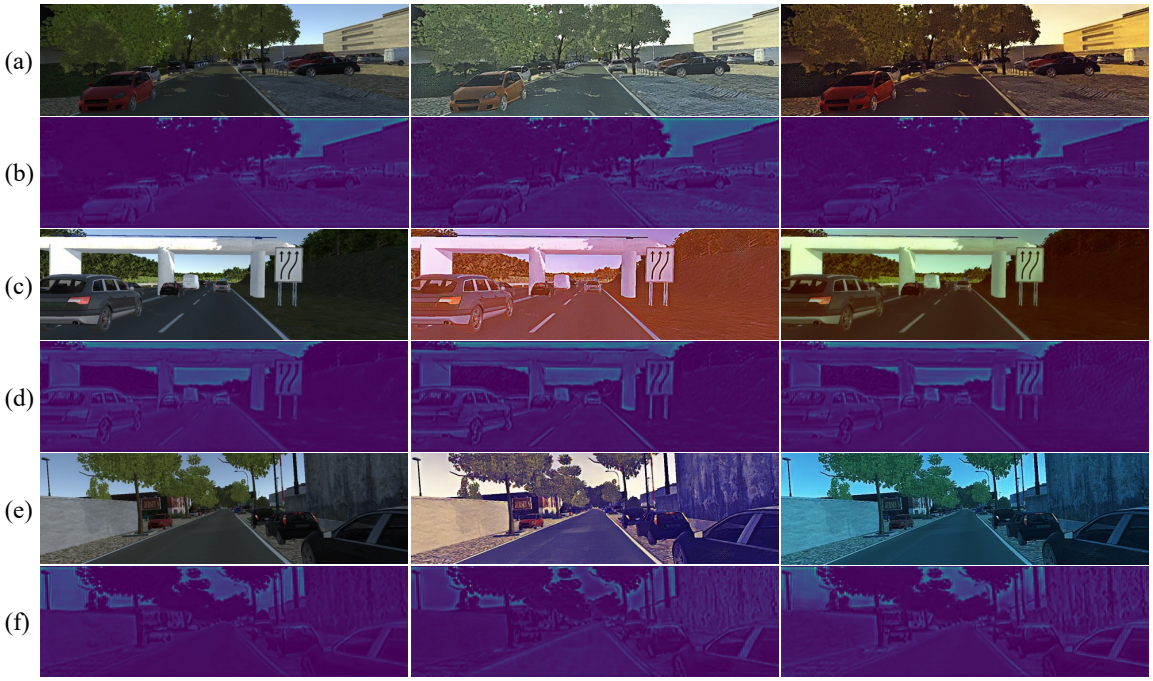

Figure 2. Visualize structure maps of different styles and the same structure code. (a), (c) and (e) represent the images of different styles under the same structure code. (b), (d) and (f) are the generated structure maps.

original image resolution is 375×1241.

NYU Depth v2 [11] is a real indoor dataset, which contains 464 video sequences of indoor scenes captured with Microsoft Kinect. The dataset is wildly used to evaluate monocular depth estimation tasks for indoor scenes. Following previous work [15, 8, 6, 1], we use the official 654 aligned image-depth pairs for evaluation. The image resolution is 480×640.

Evaluating Metrics To quantitatively evaluate the proposed approach, we follow previous work [2, 3, 9, 16]. The evaluation metrics include root mean squared error (RMSE), mean relative error (REL), Mean $\log 10$ error ($\log 10$), root mean squared error in $\log$ space ($\text{RMSE}_{\log}$), and squared relative error (Squa-Rel), defined as:

- RMSE: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d}_i)^2}$.
- REL: $\frac{1}{N}\sum_{i=1}^{N}\frac{|d_i - \hat{d}_i|}{\hat{d}_i}$.
- Mean $\log 10$ error ($\log 10$): $\frac{1}{N}\sum_{i=1}^{N}|\log_{10} d_i - \log_{10}\hat{d}_i|$.
- $\text{RMSE}_{\log}$: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log d_i - \log \hat{d}_i)^2}$.
- Squa-Rel: $\frac{1}{N}\sum_{i=1}^{N}\frac{|d_i - \hat{d}_i|^2}{\hat{d}_i}$.
- Accuracy with threshold t: Percentage of pixels whose depth d_i satisfies $\max\left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}\right) = \delta < t$, where $t \in [1.25, 1.25^2, 1.25^3]$ respectively.

d_i and $\hat{d}_i$ are the predicted depth and ground-truth depth at pixel i respectively. N denotes the number of valid pixels in the ground-truth depth map.

References

[1] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In *IJCAI*, 2019.

[2] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

[3] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[4] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.

[5] Andreas Geiger, Philip Lenz, Stiller Christoph, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[6] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *WACV*, 2019.

[7] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[8] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In *CVPR*, 2018.

[9] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[10] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *CVPR*, 2015.

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

[12] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

[13] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptatizhaoon for monocular depth estimation. In *CVPR*, 2019.

[14] Yunhan Zhao, Shu Kong, Daeyun Shin, and Charless Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *CVPR*, 2020.

[15] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *ECCV*, 2018.

[16] Tinghui Zhou, Brown Matthew, Snavely Noah, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.