Photo-realistic Neural Domain Randomization – Supplementary Material –

Sergey Zakharov[1], Rareș Ambruș[1], Vitor Guizilini[1], Wadim Kehl[2], and Adrien Gaidon[1]

[1] Toyota Research Institute, Los Altos, CA
[2] Woven Planet, Tokyo, Japan

1 RenderNet

Our RenderNet is a UNet-based encoder-decoder CNN. It takes a 15D input (see Fig. 1 of the main submission) consisting of concatenated scene coordinates in camera space X (3D), surface normals map N (3D), albedo A (3D), roughness R (1D), specularity S (1D), light direction map L_{dir} (3D), and light distance map L_{dist} (1D). Its inference results in four 3D maps: $D_{dir}, D_{ind}, G_{dir}$, and G_{ind} being the diffuse and glossy BSDF outputs for direct and indirect lighting, respectively. Predicted outputs are then used to form a final rendering.

To train it, we used the Adam optimizer [12] with a learning rate of $1e^{-4}$. Each of the four RenderNet outputs is supervised with respective ground truth images using an L1 loss.

2 CBOD

Our CBOD detector consists of two modules: the correspondence module and the pose estimation module. This section provides their detailed description. Our detector largely follows [20, 11, 16, 13, 9].

Correspondence Module. Our correspondence module is a ResNet12-based encoder-decoder CNN with four decoder heads to regress the ID mask and three channels of the dense 2D-3D correspondence map (U, V, W) from a 320×240×3 RGB image. However, we would like to note that any other backbone architecture could be used without any need to change the rest of the pipeline. The decoders upsample features up to their original size using a stack of bilinear interpolations followed by convolutional layers.

Correspondence heads regress tensors of size $H \times W \times C$, where C is the discretization density of the correspondence map, which equals to 256 in our case. Each channel stores the probability values for the class corresponding to a specific channel number. Once regressed, we compose single channel tensors (U, V, W) storing the class ID with maximal probability using the argmax operation. Defining correspondence estimation as a classification problem allows us to significantly decrease the output solution space, which subsequently results in a

better quality of 2D-3D matches and faster convergence. Resulting U, V, and W channels are used to form a 2D NOCS map encoding normalized object's coordinates in RGB. Visualizing each color component in 3D allows us to restore a partial object's geometry as can be seen in Fig. 1 and establish 2D-3D correspondences needed for pose estimation.

Similarly, the ID mask head outputs a $H \times W \times O$ tensor with O corresponding to the number of objects in the dataset plus one additional class for background. We apply an argmax operation to the output tensor and identify detected classes. Then, we use detected classes to retrieve class-specific channels, apply Otsu's method [15] to retrieve detected masks, and finally identify connected mask regions. This strategy proved to be more robust when compared to directly using argmax-based masks to define regions. Resulting mask regions are then used by the pose module.

The final loss function for *RenderNet* is defined as the sum of four losses:

$$\mathcal{L} = \mathcal{L}_m + \mathcal{L}_u + \mathcal{L}_v + \mathcal{L}_w, \tag{1}$$

where $\mathcal{L}_m$ is the mask loss, and $\mathcal{L}_u$, $\mathcal{L}_v$, and $\mathcal{L}_w$ are the losses responsible for the separate correspondence map channels U, V, and W. All losses are defined as multi-class cross-entropy functions.

Pose Module. Given detected regions and estimated 2D-3D correspondences, we use a Perspective-n-Point (PnP) [18] solver that estimates the camera pose given correspondences and intrinsic parameters of the camera. Random sample consensus or RANSAC [3] is used in conjunction with PnP to make predictions more robust to outliers. We use a standard PnP-RANSAC implementation Perspective-n-Point (PnP) provided in the OpenCV function `solvePnPRansac`. We set the number of RANSAC iterations to 300 and reprojection error threshold to 1.

Training Details. To train CBOD, we used the ADAM optimizer [12] with a learning rate of $5e^{-4}$ and weight decay of $4e^{-5}$. As opposed to [8, 14, 20], we do not use pretrained models and do not freeze the first layers of the network to have a fair comparison between different types of data.

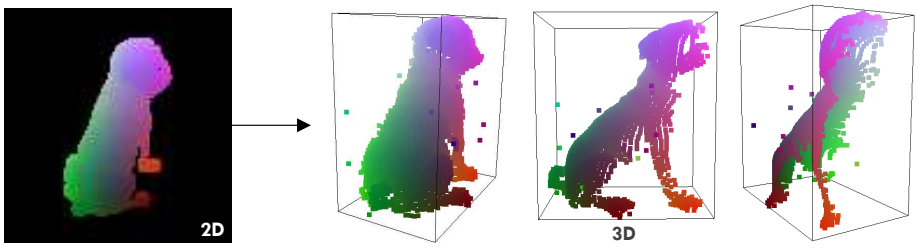

Fig. 1: 2D-3D correspondences. We recover partial geometry from a regressed 2D NOCS map and establish 2D-3D correspondences.

3 Baselines

Recall the two benchmarks *HB Dynamic Lighting Benchmark* where we train on HB_5 and *HB-LM Cross Domain Adaptation Benchmark* where we train on HB_2. For both benchmarks the training set consists of 272 frames, from which we generate an extended training set of 1088 with *BlenderProc* [1] using the posed CAD models and with randomized materials, as described in the main text. Using the synthetically generated images along with the corresponding real images (paired or unpaired), we train the GAN baselines as described below. A preprocessing step involves converting the synthetic images to grayscale, to avoid the complication of having the same synthetic object with multiple random materials being mapped to the same real object. Qualitative results for the *HB Dynamic Lighting Benchmark* are shown in Fig. 3 while Fig. 4 shows qualitative results for the *HB-LM Cross Domain Adaptation Benchmark*. Additionally, in Fig. 5 we show qualitative results when running the baselines on the additional 1000 frames generated with BlenderProc containing randomized camera viewpoints and the same objects as HB_2 but with randomized poses.

3.1 Paired image translation

We compare PNDR against two paired image translation GAN [5] based baselines: pix2pix [10] and pix2pixHD [19]. To train these methods we use synthetic and real image pairs. Although this setting is unrealistic in practice, we leverage this information from the HB dataset and train paired image-to-image translation, which we regard as an upper bound for in-domain performance.

pix2pix [10]: we use the official implementation [10, 21] and train for 200 epochs with the Adam optimizer [12] and with a starting learning rate of $2e^{-4}$ and with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is kept fixed for the first 100 epochs and decayed to 0 during the next 100 epochs. The input images are resized to 286×286 and a random crop of 256×256 pixels is selected.

pix2pixHD [19]: we use the official implementation and train for 200 epochs the Adam optimizer [12] and with a starting learning rate of $2e^{-4}$ and with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is kept fixed for the first 100 epochs

Fig. 2: Sampling novel light and materials. *PNDR* generates photo-realistic renderings by sampling materials and light positions.

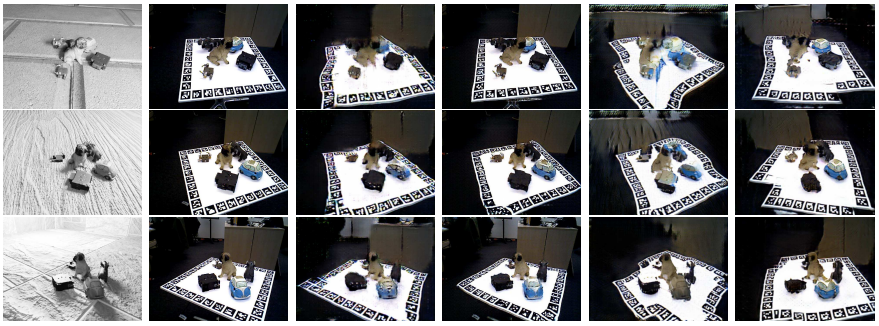

Fig. 3: Qualitative results on HB5. We show the results of the paired and upaired image translation baselines on the HB5 scene used in the HB Dynamic Lighting Benchmark. Top row: images from the HB_5 training set; last two rows: images from the HB_5 test set. The real image is used as pair target when training [10, 19] while [21, 17] use an unordered set of real images as targets.

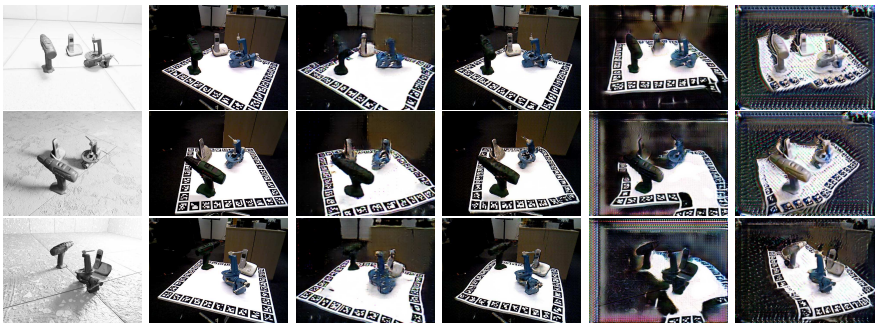

Fig. 4: Qualitative results on HB2. We show the results of the paired and upaired image translation baselines on the HB5 scene used in the HB-LM Cross Domain Adaptation Benchmark. Top row: images from the HB_2 training set; last two rows: images from the HB_2 test set. The real image is used as pair target when training [10, 19] while [21, 17] use an unordered set of real images as targets.

and decayed to 0 during the next 100 epochs. We train on the original image resolution of 640×480 without any resizing and without any cropping.

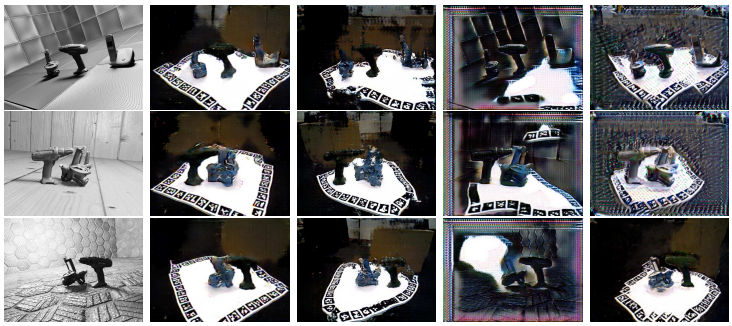

Fig. 5: Qualitative results on synthetic images containing the HB2 objects rendered with random poses. This is a completely synthetic dataset (i.e., no real image counterparts) and all GAN baseline methods struggle to generalize to generalize to this setting.

3.2 Unpaired image translation

We also compare PNDR against the unpaired image translation baselines cycleGAN [21] and CUT [17]. As before, we first convert the synthetic images to grayscale to simplify learning. Nevertheless, we empirically observed poor translation results, and particularly the GANs failed to maintain the shape of the objects, which is crucial to our downstream task, i.e., correspondence-based 3D object detection. To improve the performance of these baselines, we leverage the ground truth object masks, and added an L1 loss between the input grayscale synthetic image and the image outputted by the generator, which we first convert to grayscale as well. The L1 loss is applied only on the pixels falling in the ground truth object mask. Although unrealistic in practice, this additional L1 loss helped the unpaired image translation method maintain the shape of the objects in the generated images.

cycleGAN [21]: we use the official implementation [10, 21] and train for 200 epochs with the Adam optimizer [12] and with a starting learning rate of $2e^{-4}$ and with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate decay and input image resolution are the same as when training *pix2pix*, the only difference being the L1 loss between the generated image and the input synthetic image.

CUT [17]: we use the official implementation [17] and train for 400 epochs with the Adam optimizer [12] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The starting learning is $2e^{-4}$ and is kept fixed for 200 epochs after which it is decayed linearly to 0 over the next 200 epochs. The input images are resized to 286×286 and a

random crop of size 256×256 is selected during training. We apply the same L1 loss between the generated output image and the input synthetic image on the pixels contained in the ground truth object mask.

4 Monocular Depth Estimation

4.1 Implementation details

monodepth2 [4]: We use the ResNet18 [7] encoder-decoder architecture, as described in [4], *without* ImageNet pretraining. We train for 20 epochs with the Adam optimizer [12] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate starts at $1.5e^{-4}$ as is decayed by 20% every 5 epochs and the input images are resized to 256×320. We use standard color jittering and no cropping.
packnet-sfm [6]: we use the PackNet architecture from the official implementation. We train for 20 epochs with the Adam optimizer [12] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate starts at $2e^{-4}$ as is decayed by 20% every 5 epochs and the input images are resized to 256×320. We use standard color jittering and no cropping.

4.2 Losses

We follow the standard monocular depth estimation losses [2] defined as follows:

$$AbsRel = \frac{1}{N} \sum_{d \in D^*} \frac{|d - d^*|}{d^*} \tag{2}$$

$$RMSE = \sqrt{\frac{1}{|D^*|} \sum_{d \in D^*} |d - d^*|} \tag{3}$$

$$\delta_1 = \% \text{ of d s.t. } \max\left(\frac{d}{d^*}, \frac{d^*}{d}\right) < 1.25 \tag{4}$$

where d^* and d represents respectively ground-truth and corresponding predicted depth values, with D^* being the set containing all valid ground-truth depth pixels.

5 Other Downstream Tasks

We extend the CBOD evaluation to support these tasks on the HB dynamic lighting benchmark. In particular, we use CBOD's multi-label object masks to compute 2D bounding boxes. We use a standard F1 metric, which is defined as a weighted average of precision and recall, to evaluate performance of 2D object detection. We consider detections to be correct when the intersection over union (IoU) between predicted and ground truth bounding boxes is >= 0.5. We also evaluate the quality of estimated multi-label object masks using the IoU metric.

Table 1: 2D detection and instance segmentation.

Train	Method	HB5		HB10	
		F1	**mIoU**	**F1**	**mIoU**
Real		0.92	0.95	0.21	0.46
CAD	RayTraced - 1088	0.54	0.74	0.07	0.34
	PNDR - 1088	0.61	0.83	0.22	0.51

Our results on the new tasks are summarized in Table 1 and they are consistent with our observations on the tasks of 6D object detection and monocular depth estimation (cf. Tables 1, 2, and 5 in the main paper). For a constant light scene (i.e. HB5), PNDR improves over training on domain-randomized ray tracing images, closing the gap towards real data training. Additionally, PNDR achieves much better generalization to a more difficult dynamic lighting scene (i.e. HB10), improving even over the baseline trained directly on real data.

References

1. Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H.: Blenderproc. arXiv preprint arXiv:1911.01911 (2019)
2. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS (2014)
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
4. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3828–3838 (2019)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
6. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2485–2494 (2020)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
8. Hinterstoisser, S., Lepetit, V., Wohlhart, P., Konolige, K.: On pre-trained image features and synthetic images for deep learning (2017)
9. Hodan, T., Barath, D., Matas, J.: Epos: Estimating 6d pose of objects with symmetries. In: CVPR. pp. 11703–11712 (2020)
10. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
11. Jafari, O.H., Mustikovela, S.K., Pertsch, K., Brachmann, E., Rother, C.: ipose: instance-aware 6d pose estimation of partly occluded objects. In: ACCV. pp. 477–492. Springer (2018)
12. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: ICCV. pp. 7678–7687 (2019)
14. Manhardt, F., Kehl, W., Navab, N., Tombari, F.: Deep model-based 6d pose refinement in rgb. In: ECCV (2018)
15. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1), 62–66 (1979)
16. Park, K., Patten, T., Vincze, M.: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In: ICCV. pp. 7668–7677 (2019)
17. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision. pp. 319–345. Springer (2020)
18. Quan, L., Lan, Z.: Linear n-point camera pose determination. IEEE Transactions on pattern analysis and machine intelligence **21**(8), 774–780 (1999)
19. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)

20. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: ICCV (2019)
21. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)