3D-PL: Domain Adaptive Depth Estimation with 3D-aware Pseudo-Labeling *Supplementary Materials*

Yu-Ting Yen[1,2], Chia-Ni Lu[1], Wei-Chen Chiu[1], Yi-Hsuan Tsai[2]

[1]National Chiao Tung University, Taiwan [2]Phiar Technologies

1 Stereo Setting

In this section, we provide the details for objective and the overall training pipeline in the stereo-pair setting as introduced in Section 3.3 of the main paper.

1.1 Objectives

When stereo pairs are available, we utilize the geometry constraints to have self-supervised stereo supervisions as GASDA [16] using the geometry consistency loss. The stereo pairs contain the left image x_r, which is also used in the single-image setting, and the corresponding right image. Here we denote the left and right real images as x^{left}, x^{right} (we ignore the notation r in the real image like x_r^{left} for simplicity) and their depth prediction $\tilde{y}^{left} = F(x^{left}), \tilde{y}^{right} = F(x^{right})$. The geometry consistency loss in GASDA [16] is a reconstruction penalty between the real left image x^{left} and the warped left image $\tilde{x}^{left}$.

$$\mathcal{L}_{tgc}^{left}(F) = \eta\frac{1 - SSIM(x^{left}, \tilde{x}^{left})}{2} + \mu||x^{left} - \tilde{x}^{left}||, \tag{1}$$

where η and μ are set as 0.85 and 0.15 respectively following [16]. The warped left image $\tilde{x}^{left}$ is obtained from the disparity a and the right image x^{right} with bilinear sampling [5] following [4]:

$$\tilde{x}^{left} = x^{right} - a, \tag{2}$$

Since we know the camera parameters when collecting the stereo images, we can convert the depth prediction $\tilde{y}^{left}$ of left image to the disparity a through:

$$a = \frac{b \cdot f}{\tilde{y}^{left}}, \tag{3}$$

where b is the baseline distance between the two cameras and f is the focal length, both parameters are known in the stereo-pair setting. In addition to reconstructing $\tilde{x}^{left}$ from the right image x^{right}, we also warp $\tilde{y}^{right}$ and x^{left} to get $\tilde{x}^{right}$ in our experiments using a similar process and loss $\mathcal{L}_{tgc}^{right}$. Finally, our geometry consistency loss is $\mathcal{L}_{tgc} = \mathcal{L}_{tgc}^{left} + \mathcal{L}_{tgc}^{right}$.

1.2 Overall Training Pipeline

In our stereo-pair setting, there are also two training stages for training a preliminary depth model F and applying the proposed pseudo-labeling techniques through this preliminary model.

Training a preliminary depth model F. In the stereo-pair setting, we follow the single-image setting to use Eq.(6) in the main paper and train a preliminary depth model F for 20 epochs and further train another 10 epochs with adding $\mathcal{L}_{tgc}$:

$$\mathcal{L}_{base}^{stereo} = \lambda_{task}\mathcal{L}_{task}^{s} + \lambda_{sm}\mathcal{L}_{sm} + \lambda_{tgc}\mathcal{L}_{tgc}, \tag{4}$$

where λ_{task}, λ_{sm}, and λ_{tgc} are set as 100, 0.1, and 50 respectively. Here we do not include the $\mathcal{L}_{task}^{s \to r}$ loss to make the model training more focused on the real-domain data. In the second stage, the overall loss is defined as Eq.(10) in the main paper.

2 Sensitivity Analysis

In this section, we analyze the impact of different parameters, such as threshold or the weights in the loss. All experiments are performed in the single-image setting.

2.1 Threshold τ

Table 1 shows our results under different threshold τ as defined in Eq.(4) of the main paper, which controls the range of pseudo-label. The higher threshold means more pseudo-labels are chosen but may not be accurate, while the lower one can obtain more precise pseudo-labels but the amount is less. As shown in Table 1, our method performs robustly under a reasonable range of τ (e.g., 0.3 to 1 meter).

Threshold	Error Metrics (lower, better)			
	Abs Rel	Sq Rel	RMSE	RMSE log
$\tau = 0.1$	**0.161**	1.048	4.497	**0.239**
$\tau = 0.3$	0.162	**1.045**	4.476	**0.239**
$\underline{\tau = 0.5}$	0.162	1.049	**4.463**	**0.239**
$\tau = 1$	**0.161**	1.053	**4.463**	**0.239**
$\tau = 2$	**0.161**	1.060	4.468	**0.239**
$\tau = 3$	0.162	1.066	4.473	**0.239**

Table 1: Our results of different thresholds τ. The unit of τ is meter. Underline denotes our final setting.

2.2 Proportion of Pseudo-label Loss α

Table 2 shows the experiments of using different weight proportion between the pseudo-label loss on real data and the task loss on synthetic data, where α is defined in Eq.(9) of the main paper. With increasing the weight of pseudo-label loss, e.g., $\alpha = 0.3$ to 0.7, the performance is gradually improved, which shows the benefits of our proposed pseudo-labeling strategy. However, the performance drops when α becomes too large, which indicates the importance of having the accurate supervisions from the synthetic data to stabilize model training.

α	$1-\alpha$	Error Metrics (lower, better)			
		Abs Rel	Sq Rel	RMSE	RMSE log
0.3	0.7	**0.161**	1.051	4.520	0.241
0.5	0.5	**0.161**	1.053	4.490	0.240
$\underline{0.7}$	$\underline{0.3}$	0.162	1.049	**4.463**	**0.239**
0.9	0.1	0.164	**1.043**	4.508	0.240
1	0	0.191	1.175	4.472	0.253

Table 2: Our results of using different proportions α between the pseudo-label loss (α) and the task loss ($1-\alpha$). Underline denotes our final setting.

2.3 Weighted Terms λ_{cons} and λ_{comp}

Table 3 shows the results of different values of weighted terms (λ_{cons}, λ_{comp}) between 2D-based and 3D-aware pseudo-label loss($\mathcal{L}^{cons}_{pseudo}$, $\mathcal{L}^{comp}_{pseudo}$), defined in Eq.(9) of the main paper. As shown in Table 3, our method performs robustly under a reasonable range of λ_{cons} and λ_{comp} if they do not become too large. We also note that, since the 2D position projected from 3D has a little scale shift to the original 2D pixel on the image plane, there exists scale difference between $\mathcal{L}^{cons}_{pseudo}(\approx 10^{-3})$ and $\mathcal{L}^{comp}_{pseudo}(\approx 10^{-2})$. Thus, we use 10 times λ_{cons} than λ_{comp} as our final setting.

λ_{cons}	λ_{comp}	Error Metrics (lower, better)			
		Abs Rel	Sq Rel	RMSE	RMSE log
0.1	0.01	**0.162**	1.062	4.711	0.247
0.1	0.1	0.163	**1.045**	4.483	0.240
$\underline{1}$	$\underline{0.1}$	**0.162**	1.049	**4.463**	**0.239**
1	1	0.169	1.097	**4.463**	0.243
10	1	0.168	1.102	4.485	0.243
10	10	0.171	1.138	4.504	0.244

Table 3: Our results of using different values of weighted term ($\lambda_{cons}, \lambda_{comp}$) between 2D and 3D pseudo-label loss. Underline denotes our final setting.

3 More Experiments

In this section, we provide experiments for showing the effectiveness of 3D completion model G_{com}, the comparison with 2D depth completion model, the design choices of the depth estimation loss $\mathcal{L}_{task}$, and the model complexity.

3.1 Effectiveness of G_{com}

We verify whether the 3D completion model G_{com} is well trained. To this end, we simply take one sequence "0018" out of Virtual KITTI dataset as the testing set while the remaining is the training set, and then use the same training procedure stated in the main paper to train our 3D completion model G_{com}. During evaluation, we first project the 2D ground truth depth y_s in testing set to 3D point clouds and uniformly sample them to have sparse point cloud $\hat{p}^s_{sparse}$, and then we take $\hat{p}^s_{sparse}$ as the input of G_{com} to obtain the result of completion $\tilde{p}^s_{dense}$. Finally, we project $\tilde{p}^s_{dense}$ to the 2D depth map $\tilde{y}^s_{dense}$ and measure the depth accuracy with its original ground truth y_s. Table 4 shows that G_{com} has the ability to produce precise and reasonable 3D completion results.

Method	Accuracy Metrics (higher, better)		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
3D completion model G_{com}	0.976	0.991	0.995

Table 4: Performance of the completion model G_{com} trained on the synthetic dataset.

We also provide details for network architecture and sampling strategy of our completion model G_{com}.

Network Architecture of G_{com}. 3D completion model G_{com} is modified from PCN [15]. We follows [14] to adjust the PCN network, including the encoder and the decoder. The encoder of our completion model G_{com} is simplified to one PointNet [11] layer. Our decoder only uses the second stage of point generation in PCN, and we take our sparse point cloud as the "Coarse Output" in PCN. The whole network architecture will be made available to the public.

Sampling Strategy. We "uniformly" sample 3D point cloud into 30720 (25% of pixel number in an image) sparse points as the input to the 3D completion model G_{com}. The 3D point cloud before sampling is projected from 2D depth map through the projection mechanism introduced in Section 4.1. During the pre-training process of G_{com}, we sample the point cloud projected from synthetic ground truth depth y_s to sparse point cloud $\hat{p}^s_{sparse}$. In 3D-aware pseudo-labeling generation, we project 2D pseudo-labels to 3D as point clouds $\hat{p}_{cons}$ and sample $\hat{p}_{cons}$ to sparse point cloud $\hat{p}_{sparse}$.

3.2 Comparison with 2D Depth Completion Model

Since there exists 2D depth completion methods which are also able to complete depth values directly on 2D depth-maps/image [1, 7, 9], we compare our 3D point cloud completion model G_{com} with a 2D depth completion model [1] to validate the necessity of our 3D-aware approach. We apply a recent 2D depth completion model [1] to the sparse depth map sampled from our confident area with two types of training setting. One is pre-trained model provided by the author, and the other one is the model trained from scratch on vKITTI [2] with the same setting as our completion model G_{com}.

Note that our 3D completion model G_{com} is only trained on vKITTI [2] and the 2D depth completion model provided by the author is pre-trained on KITTI [3], so the 2D depth completion model accesses more information from the real domain. We replace our 3D completion model G_{com} with the 2D depth completion model [1] to generate pseudo-labels for training depth prediction model F. As shown in Table 5, even "+ 2D depth completion [1](pre-trained by authors)" is pre-trained on KITTI supervisedly (i.e., using the ground truth depths for training), our proposed 3D-aware approach (i.e., "+ 3D-aware completion label $\hat{y}_{cons}$") provides better performance in all metrics. In addition, we re-train the 2D depth completion model [1] with the same training setting as ours (i.e., trained on vKITTI), and our proposed 3D-aware approach reaches 29% lower error on the "Sq Rel" metric. This shows that our 3D completion model G_{com}, which explicitly considers the 3D structural information, is able to produce more reliable pseudo-labels than the 2D depth completion models.

Method	Abs Rel	Sq Rel	RMSE	RMSE log
+ 2D depth completion [1] (pre-trained by authors)	**0.164**	1.068	4.746	0.247
+ 2D depth completion [1] (trained on vKITTI)	0.186	1.476	6.125	0.310
+ 3D-aware completion label $\hat{y}_{cons}$ (Ours)	**0.164**	**1.054**	**4.473**	**0.239**

Table 5: Training depth prediction model F by using the pseudo-labels generated from different completion models. Note that "+ 2D depth completion [1] (pre-trained by authors)" is pre-trained on KITTI [3], in which the 2D depth completion model [1] has the supervision on depth directly from the real domain, while our model is trained on vKITTI.

3.3 Design Choices of Depth Estimation Loss $\mathcal{L}_{task}$

As stated in Section 2.2, when training with pseudo-labels, it is important to have accurate supervisions on the depth estimation loss $\mathcal{L}_{task}$ to stabilize model training. In Eq.(9) of the main paper, we retain $\mathcal{L}^{s}_{task}$ as the depth estimation to make the model training more focused on the real-domain data. While there exists another option $\mathcal{L}^{s\to r}_{task}$ for the depth estimation loss, as $\mathcal{L}^{s\to r}_{task}$ considers real-stylized images, images produced by style transfer may not align with their original depth ground truths well. Table 6 shows the experiments of adopting different options for the depth estimation loss in Eq.(9) of the main paper, which demonstrates that using $\mathcal{L}^{s}_{task}$ instead of $\mathcal{L}^{s\to r}_{task}$ has lower errors.

Method	Abs Rel	Sq Rel	RMSE	RMSE log
using both $\mathcal{L}^{s}_{task}$ and $\mathcal{L}^{s\to r}_{task}$	**0.160**	1.074	4.611	0.246
using $\mathcal{L}^{s\to r}_{task}$ only	0.161	1.090	4.635	0.247
using $\mathcal{L}^{s}_{task}$ only	0.162	**1.049**	**4.463**	**0.239**

Table 6: Different options for the depth estimation loss $\mathcal{L}_{task}$ in Eq.(9) of the main paper.

3.4 Model Complexity

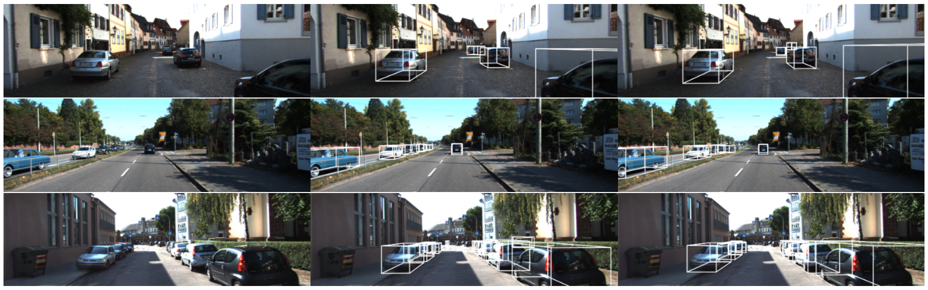
We analyze the model complexity by computing the number of parameters and the training/testing time for our models. We have depth prediction model F and the completion model G_{com}. The completion model G_{com} only contains 1.324M parameters, which is much smaller than the depth model F (54.565M). The training time for depth model F and completion model G_{com} are 80 and 21 hours. During testing, only the depth model F is required, where it does not introduce additional overheads compared to normal model inference (0.014 seconds for a 192×640 image as used in [16,17]).

3.5 Application on 3D Object Detection

To show the effectiveness of our depth result, we apply the final depth prediction to the 3D object detection task. We adopt Pseudo-LiDAR [13] to convert our generated depth map to pseudo LiDAR, and take the pseudo LiDAR as the input to the 3D object detection model. We show example results in Figure 1 compared to the ground truths. We also follow [13] to evaluate the result on the validation set of KITTI object detection benchmark for the "car" category. With the IoU threshold at 0.7, the average precision for the 3D object box detection (AP_{3D}) is 15.8%, 12.3%, and 11.2% for easy, moderate, and hard cases, respectively.

4 Details for 2D/3D Projection

We provide the implementation details for the projection procedure between 2D and 3D, including the projection mechanism and some discussions.



Input image Ground truth 3D-PL + Pseudo-LiDAR [13]

Fig. 1: 3D object detection results using 3D-PL.

4.1 Projection Mechanism

Projection from 2D to 3D. We aim to reconstruct each point (x_i, y_i, z_i) in the 3D space from the 2D image pixel (u_i, v_i) with its depth value d_i based on the standard pinhole camera model. We assume the size of image is $H \times W$ and the pixel positions on the original image plane are $\{(u_i, v_i)\}_{i=1}^{H\times W}$, where each pixel (u_i, v_i) has the corresponding depth value d_i. Then, we project the point from 2D to 3D through $project_{2D\rightarrow 3D}$ to obtain 3D point (x_i, y_i, z_i) in the 3D point cloud $\hat{y}_{cons}$:

$$x_i = \frac{d_i^*(u_i - o_x)}{f}, y_i = \frac{d_i^*(v_i - o_y)}{f}, z_i = d_i^*, \tag{5}$$

where f is the focal length, o_x and o_y are the 2D position of camera center, $d_i^* = d_i + \varepsilon$, ε is a shift to convert relative depth value d_i to the absolute depth value from the camera center. Please note that, a single image has infinite possible 3D reconstruction depending on different camera parameters. Since our objective of the 3D completion model is to learn the structure and the depth relationship in the 3D space, we do not need to restore exactly the same setting as the image being captured in the real world. On the other hand, as we cannot know the camera parameters of the real data, we hence set up reasonable projection parameters on our own and use the same setting in training the 3D completion model and finding 3D-aware pseudo-labels. In experiments, we adopt the same focal length f as virtual KITTI [2] and set ε as 40. Normally, ε is set equal to the focal length f, but such setting would lead to large values for x_i and y_i coordinates as indicated in Eq. (5). We therefore in experiments adopt the normalized depth values and fix the focal length to seek for a suitable shift ε, which gives a reasonable scale of 3D coordinates and still maintains the relationship between depth values.

Projection from 3D to 2D. After the 3D completion process, we obtain $\tilde{p}_{dense} = (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$, and then we project each point back to the original 2D plane as $(\tilde{u}_i, \tilde{v}_i)$ with the updated depth value $\tilde{d}_i = \tilde{z}_i - \varepsilon$ (we ignore the ε for simplicity in the main paper) by the inverse operation of Eq. (5):

$$\tilde{u_i} = \frac{\tilde{x_i} \cdot f}{\tilde{z_i}} + o_x, \tilde{v_i} = \frac{\tilde{y_i} \cdot f}{\tilde{z_i}} + o_y, \tilde{d_i} = \tilde{z_i} - \varepsilon, \tag{6}$$

where $(\tilde{u}_i, \tilde{v}_i)$ are rounded to integers. Since all the 3D points are generated through the completion model, the position $(\tilde{u}_i, \tilde{v}_i)$ projected from point cloud may be duplicated (i.e., two projected points happen to overlap in the 2D plane) or out of the original image plane (i.e., $(\tilde{u}_i, \tilde{v}_i) < 0$ or $(\tilde{u}_i, \tilde{v}_i) > (H, W)$). For duplicated points, we choose the minimum depth value among all duplicated points as the final depth. For those positions out of the original image plane, we view them as failing projection and do not take them as the pseudo-label $\hat{y}_{comp}$.

4.2 Discussions of 2D/3D Pseudo-label

We discuss whether $\hat{y}_{comp}$ is complementary to the original pseudo label $\hat{y}_{cons}$. We observe that in Figure 3 of the main paper, for some regions that look similar, the values of the same pixel between $\hat{y}_{cons}$ and $\hat{y}_{comp}$ are very close but have a little scale shift ($< 10^{-1}$). For the areas that appear different (e.g., bottom-left area), $\hat{y}_{comp}$ has nearer depth values than original $\hat{y}_{cons}$, in which nearer depth values are more reasonable for the object and grass in the bottom-left corner of the image. It shows that the completion model refers to the 3D structural information to produce better results.

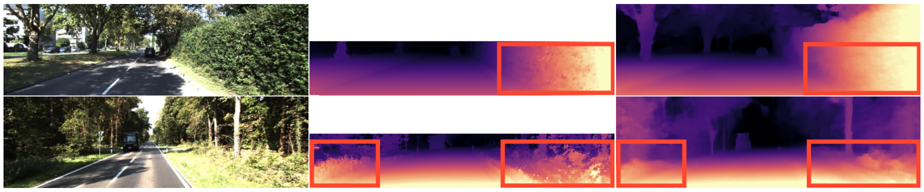


Fig. 2: 3D-PL produces better results in overall structure and shape of objects, but may lose some details for the objects with complicated textures such as grass and plants.

5 Limitations

Figure 2 shows one example of the limitation in our 3D-PL with the stereo-pair setting. Since 3D-PL focuses on the structural information, it can perform well on the overall structure, e.g., the shape of cars and the hard objects such as road signs or traffic lights. However, for the object that has complicated textures like grass, 3D-PL produces smoother results but loses the details of the plant.

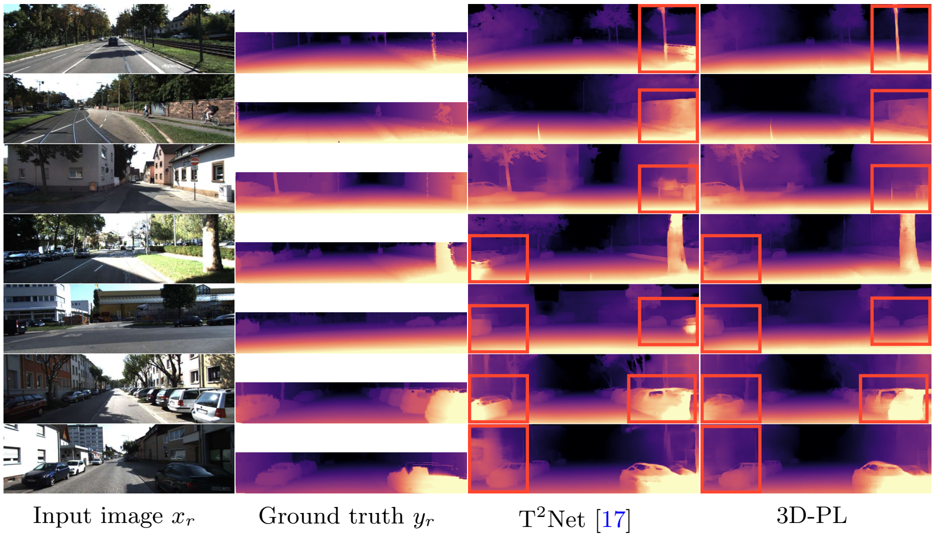


Fig. 3: More qualitative results on KITTI [3] in the single-image setting.

6 More qualitative results

We provide more qualitative results for different settings. Figure 3 and Figure 4 are results for KITTI [3] in the single-image and stereo-pair settings, respectively. Figure 5 and Figure 6 are results for KITTI stereo 2015 [8] in single-image and stereo-pair settings, respectively. Figure 7 presents results for make3D [12] in the single-image setting.

References

1. Eldesokey, A., Felsberg, M., Holmquist, K., Persson, M.: Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 5
2. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 5, 7
3. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 5, 9, 10
4. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 1
5. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in Neural Information Processing Systems (NeurIPS) **28** (2015) 1
6. Lopez-Rodriguez, A., Mikolajczyk, K.: Desc: Domain adaptation for depth estimation via semantic consistency. ArXiv:2009.01579 (2020) 10, 11
7. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In: IEEE International Conference on Robotics and Automation (ICRA) (2019) 5

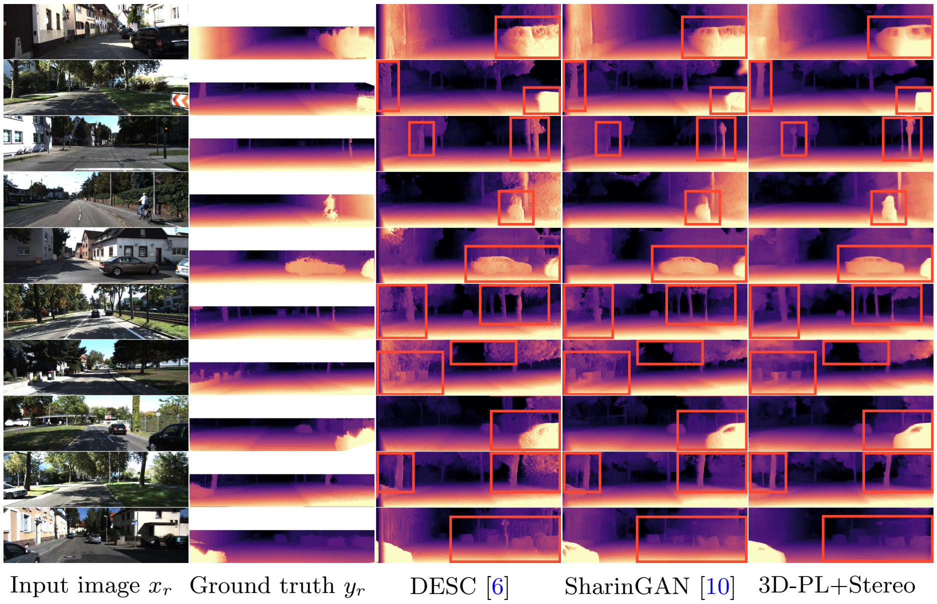


Fig. 4: More qualitative results on KITTI [3] with having stereo pairs during training.

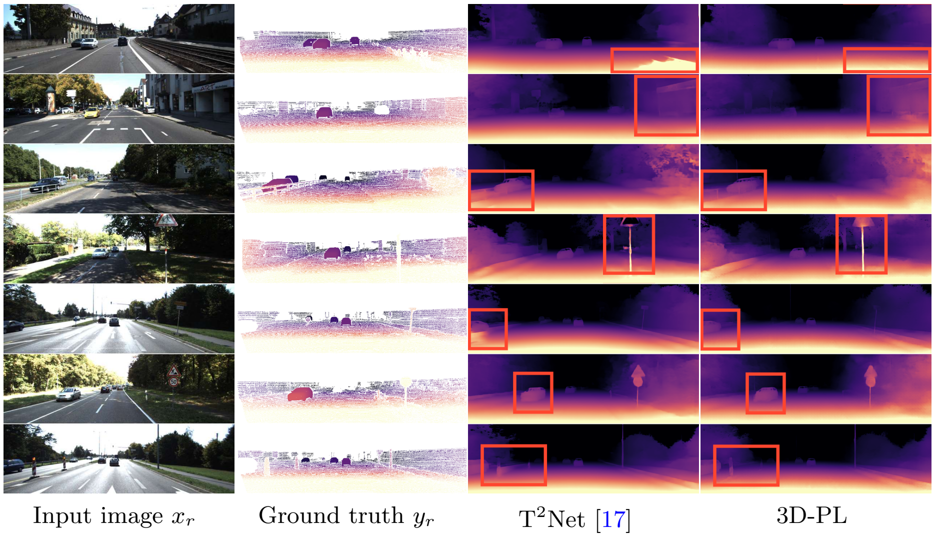


Fig. 5: More qualitative results on KITTI stereo 2015 [8] in the single-image setting.

8. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 9, 10, 11

9. Park, J., Joo, K., Hu, Z., Liu, C.K., So Kweon, I.: Non-local spatial propagation network for depth completion. In: European Conference on Computer Vision

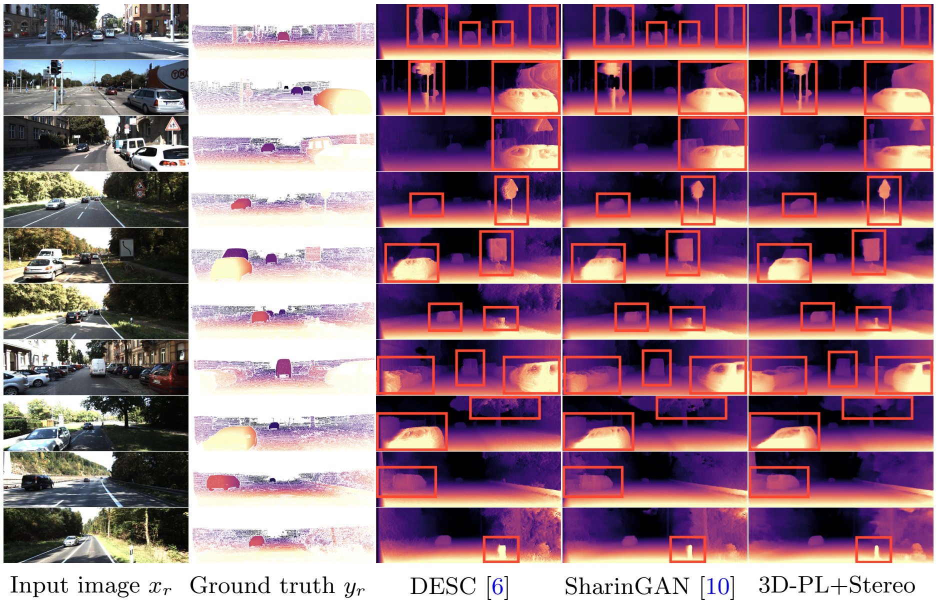


Fig. 6: More qualitative results on KITTI stereo 2015 [8] with having stereo pairs during training.

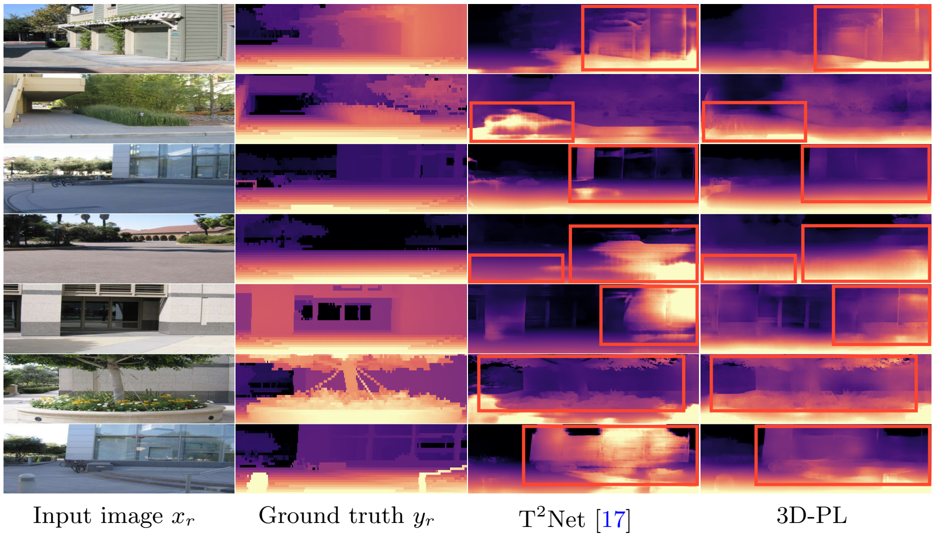


Fig. 7: More qualitative results on Make3D [12] in the single-image setting.

(ECCV) (2020) 5

10. PNVR, K., Zhou, H., Jacobs, D.: Sharingan: Combining synthetic and real data for unsupervised geometry estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 10, 11
11. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and

Pattern Recognition (CVPR) (2017) 4

12. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2008) 9, 11
13. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8445–8453 (2019) 6
14. Xiang, R., Zheng, F., Su, H., Zhang, Z.: 3ddepthnet: Point cloud guided depth completion network for sparse depth and single color image. ArXiv:2003.09175 (2020) 4
15. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: International Conference on 3D Vision (3DV) (2018) 4
16. Zhao, S., Fu, H., Gong, M., Tao, D.: Geometry-aware symmetric domain adaptation for monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 6
17. Zheng, C., Cham, T.J., Cai, J.: T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In: European Conference on Computer Vision (ECCV) (2018) 6, 9, 10, 11