Supplementary Material for Mind The Edge: Refining Depth Edges in Sparsely-Supervised Monocular Depth Estimation

Lior Talker[1] Aviad Cohen[1] Erez Yosef[1,2] Alexandra Dana[1] Michael Dinerstein[1]
[1]Samsung Israel R&D Center, Tel Aviv, Israel [2]Tel Aviv University, Israel
{lior.talker,aviad.cohen,alex.dana,m.dinerstein}@samsung.com erez.yo@gmail.com

In the following section we provide additional details, examples and results for the experiments described in the main paper.

1. The Synscapes dataset

To simulate a real oudoor dataset with LIDAR supervision, we sample pixels from the dense depth GT in a typical LIDAR pattern, similarly to the KITTI dataset, with 64 vertical beams and an horizontal beam density of 0.09 degrees between beams. Since this is a naïve approach we follow the LIDAR density, presented in Fig. 3 of the main paper, and randomly remove LIDAR samples such that the resulting distribution is similar to KITTI's. Note, however, that the true LIDAR spatial distribution in KITTI is more complex than our simulation since large continuous areas lack LIDAR samples in KITTI, in contrast to our simulation. We hypothesize that a more realistic simultion would result in worse depth edges of the baseline, increasing the depth edges performance gap from our method.

In Fig. 12 we present two images from the Synscapes dataset alongside the depth prediciton of the baseline and our method (with Packnet-SAN). It can be seen that although the Packnet-SAN baseline is has mostly accurate edges, our method still outperforms it in some parts of the image. The quantitative results are presented in the main paper (Tab. 3) and in the bottom of Fig. 1, which demonstrate a small, but consistent improvement in the depth edges quality. Importantly, as argued in the main paper, the per-pixel metrics, e.g., ARE, of our method is significantly better, which suggests that when dense depth is available, the improvement in the edge quality is translated into an improvement in the per-pixel depth metrics.

2. Simultaneous Training on Source and Target

In Fig. 3, Tab. 1 and Tab. 2 we report an experiment where we trained Packnet-SAN simoultaneously on the source (GTA-PreSIL) and the target (KITTI or DDAD) with the depth loss. Both in KITTI (denoted as Packnet-SAN (K+G)) and in DDAD (denoted as Packnet-SAN (D+G)) the edge metric (AUC) is similar or slightly better than the baseline trained on KITTI or DDAD alone. However, the depth metrics (ARE) are significantly worse, often twice as worse. We conclude that our method is a significantly better alternative than simoultaneous training. Furthermore, we note that simoultaneous training is also more time consuming since for each target dataset the source has to be trained as well. In addition, training on GTA-PreSIL, followed by training on KITTI does not yield good performance, where the edge and ARE metrics are slightly worse than the Packnet-SAN baseline.

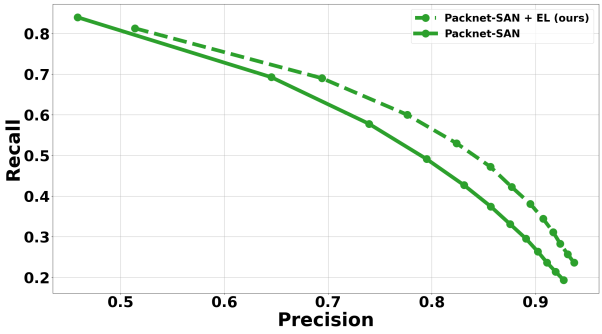

Figure 1. Precision and recall of the depth edges of the baseline vs. our method for Packnet-SAN on the Synscapes dataset.

3. Using RGB Edges Instead of Depth Edges

A potential simple alternative for the use of the depth edges, obtained from the DEE network, is to use image edges, obtained using an edge detector running directly on the RGB images. We used Canny edge detector (empirically set thresholds to 120 and 160) to extract multiscale edges and normals, which were used to train Packnet on KITTI with edge loss, instead of using our DEE output. The quantitative results using Packnet are 4.61% ARE (worse than baseline and ours) and 49.28% (42.09%) AUC (slightly better than the baseline but significantly worse than ours) on the KDE dataset. These results emphasize the importance of the DEE

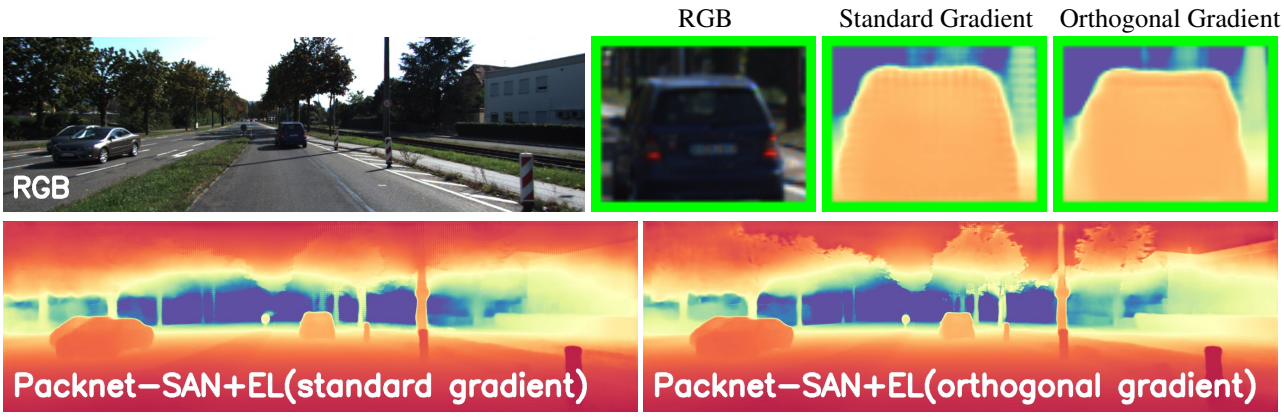

Figure 2. The standard spatial image gradient (named 'standard') and our proposed orthogonal to the edges spatial gradient (named 'orthogonal'). See Sec. 3.2 in the main paper for details.

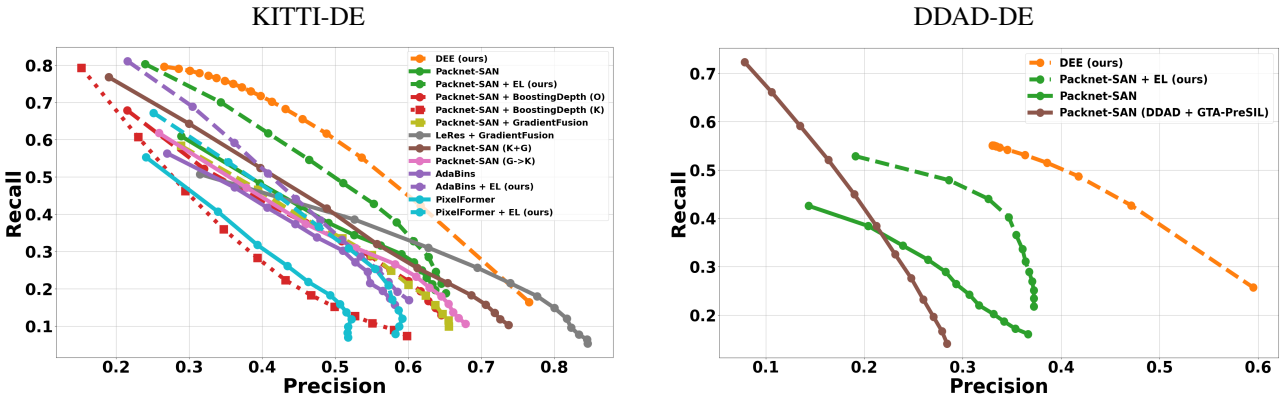

Figure 3. **Precision and recall of the depth edges on the KITTI-DE and DDAD-DE evaluation sets**. Each of the points on the graphs that correspond to an MDE method is generated with different parameters of the Canny edge detector. Each of the points on the graphs that correspond to the DEE method is generated by thresholding the depth edge probability in the range $(0, 1)$.

network, as edges from the RGB are highly noisy and their correlation to the depth edges is not high enough (see false edges in Fig. 4).

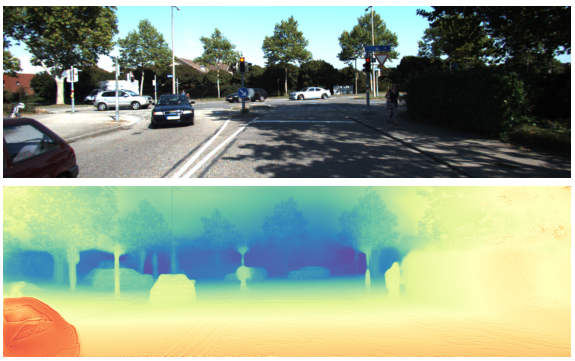

Figure 4. Training on RGB edges instead of depth edges.

4. Virtual Human Insertion for Data Aug.

We demonstrate another potential use for our method - inserting virtual humans to automotive images, potentially for data augmentation. In Fig. 5 we present a virtual human inserted to the scene, where in our method the occlusion is significantly more realistic than the baseline.

Figure 5. Virtual human insertion. Left: Packnet-SAN (baseline), Right: Packnet-SAN + EL (ours).

Method	KITTI-DE				KITTI test		
	AUC (edges) ↑	ORD ↓	ARE ↓	$\delta < 1.25$ ↑	ORD ↓	ARE ↓	$\delta < 1.25$ ↑
Packnet-SAN	47.56% (39.40%)	7.68%	3.45%	98.66%	12.40%	6.17%	95.39%
Packnet-SAN (K+G)	53.55% (41.83%)	8.81%	6.01%	95.83%	11.78%	8.90%	91.04%
Packnet-SAN (G → K)	45.56% (35.39%)	8.52%	5.26%	96.39%	11.29%	8.02%	92.28%
Packnet-SAN + BoostingDepth (O)	46.04% (37.07%)	10.35%	9.32%	88.90%	12.63%	11.10%	86.41%
Packnet-SAN + BoostingDepth (K)	36.19% (31.27%)	9.47%	7.24%	93.62%	11.45%	8.33%	91.99%
Packnet-SAN + GradientFusion	44.51% (34.10%)	9.18%	5.93%	95.66%	11.15%	7.18%	94.17%
LeRes + GradientFusion	44.59% (34.53%)	12.02%	17.26%	71.26%	12.80%	15.76%	76.12%
Packnet-SAN + EL (ours)	**61.87% (49.02%)**	7.75%	3.61%	98.53%	12.48%	6.50%	95.06%
AdaBins	41.23% (34.11%)	7.69%	3.14%	98.78%	10.14%	6.28%	95.85%
AdaBins + EL (ours)	**53.47% (44.00%)**	7.64%	3.11%	98.79%	10.13%	6.21%	95.87%
PixelFormer	32.79% (26.44%)	7.47%	3.00%	98.79%	7.56%	5.45%	96.98%
PixelFormer + EL (ours)	**46.23% (35.33%)**	7.53%	2.94%	98.80%	7.58%	5.59%	96.72%

Table 1. **Results on the KITTI dataset.** The AUC is given for the range where at least one MDE method has valid measurement: [0.12,0.65]. In parentheses we also report the AUC of the full [0,1] range. In BoostingDepth, O is for the original training (dense data) by the authors, and K is for our training (KITTI data). K+G and G → K stand for simultaneous training on KITTI and GTA-PreSIL, and training on GTA-PreSIL followed by training on KITTI.

Method	DDAD-DE				DDAD test		
	AUC (edges) ↑	ORD ↓	ARE ↓	$\delta < 1.25$ ↑	ORD ↓	ARE ↓	$\delta < 1.25$ ↑
Packnet-SAN	31.52% (23.32%)	8.03%	8.89%	91.62%	8.95%	9.49%	90.7%
Packnet-SAN (D + G)	29.89% (25.49%)	10.37%	14.09%	83.14%	12.29%	16.74%	78.09%
Packnet-SAN + EL (ours)	**48.32% (32.29%)**	8.38%	8.99%	91.44%	9.43%	10.0%	89.5%

Table 2. **Results on the DDAD dataset.** The AUC is given for the range where at least one MDE method has valid measurement: [0.14,0.37]. In parentheses we also report the AUC of the full [0,1] range. D+G stands for simultaneous training on DDAD and GTA-PreSIL.

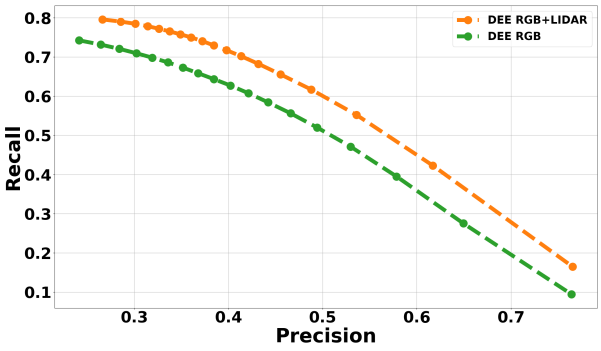

Figure 6. Precision and recall of the DEE model with RGB only and RGB+LIDAR inputs, inffered on the KITTI-DE dataset.

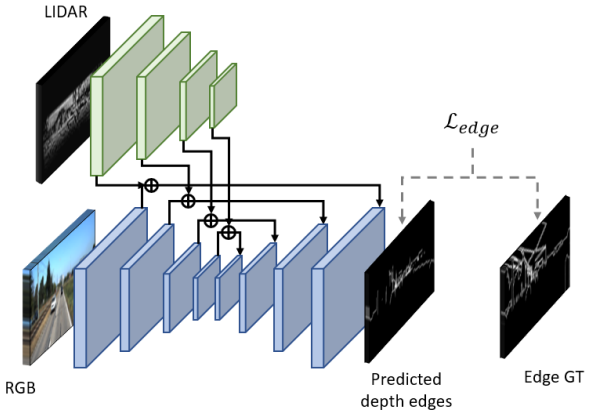

Figure 7. The architecture of the DEE model.

5. Comparison to BoostingDepth [4] details

To adjust BoostingDepth to Packnet-SAN, we computed Packnet-SAN's receptive field both theoretically and empirically and obtained a recpetive field of 1028 pixels. We utilize it as required by their method. To train the depth merger on KITTI data (called 'Packnet-SAN + BoostingDepth (K) in this paper), we generate training data as explained in BoostingDepth's Github repository, where the

Figure 8. Examples of depth predictions of Packnet-SAN and Packnet-SAN + EL (ours) of images from the DDAD-DE dataset - Part I.

low-resolution patches are taken to be in the Receptive Field (RF) size of Packnet-SAN (1028^2) and the high-resolution paches are taken to be the entire image (1280^2). Note that the original weights of the depth merger were trained with the IBims-1 [3] and the Middlebury [5] datasets, and used the depth maps of e.g., MIDAS. In this case the ratio between the low-resolution patches and the high-resolution patches, 384^2 and 672^2 is much larger than in our case, which may contribute to the difference in performance depicted between the depth mergers. This difference, which is an unchangeable constant, is a limitation of BoostingDepth which relies on the RF of the MDE and the size of the image. Furthermore, we hypothesize that the KITTI dataset does not contain sufficient depth information near depth

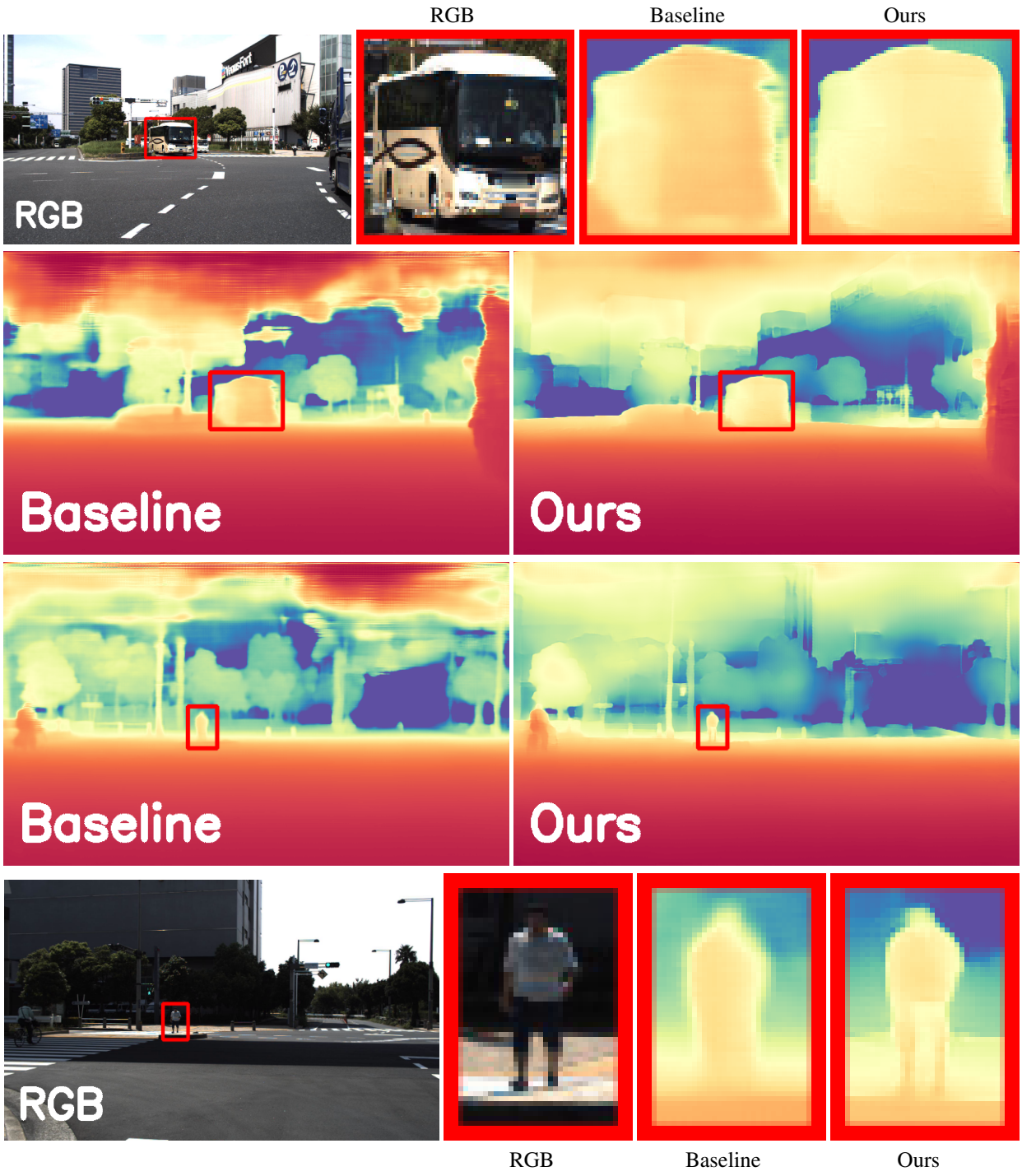

Figure 9. Examples of depth predictions of Packnet-SAN and Packnet-SAN + EL (ours) of images from the DDAD-DE dataset - Part II.

edges (see Fig. 3 in main paper) due to the LIDAR sparsity, which results in the high resolution depth predictions of Packnet-SAN to have inaccurate edges, differently from the original depth merger trained on dense datasets. The qualitative results are presented in Fig. 10.

6. Additional Details

6.1. KDE and DDE Datasets Annotation Details

As discussed in the paper, we manually annotate 102 images of KITTI (i.e., KDE dataset) and 50 images of DDAD (i.e.,

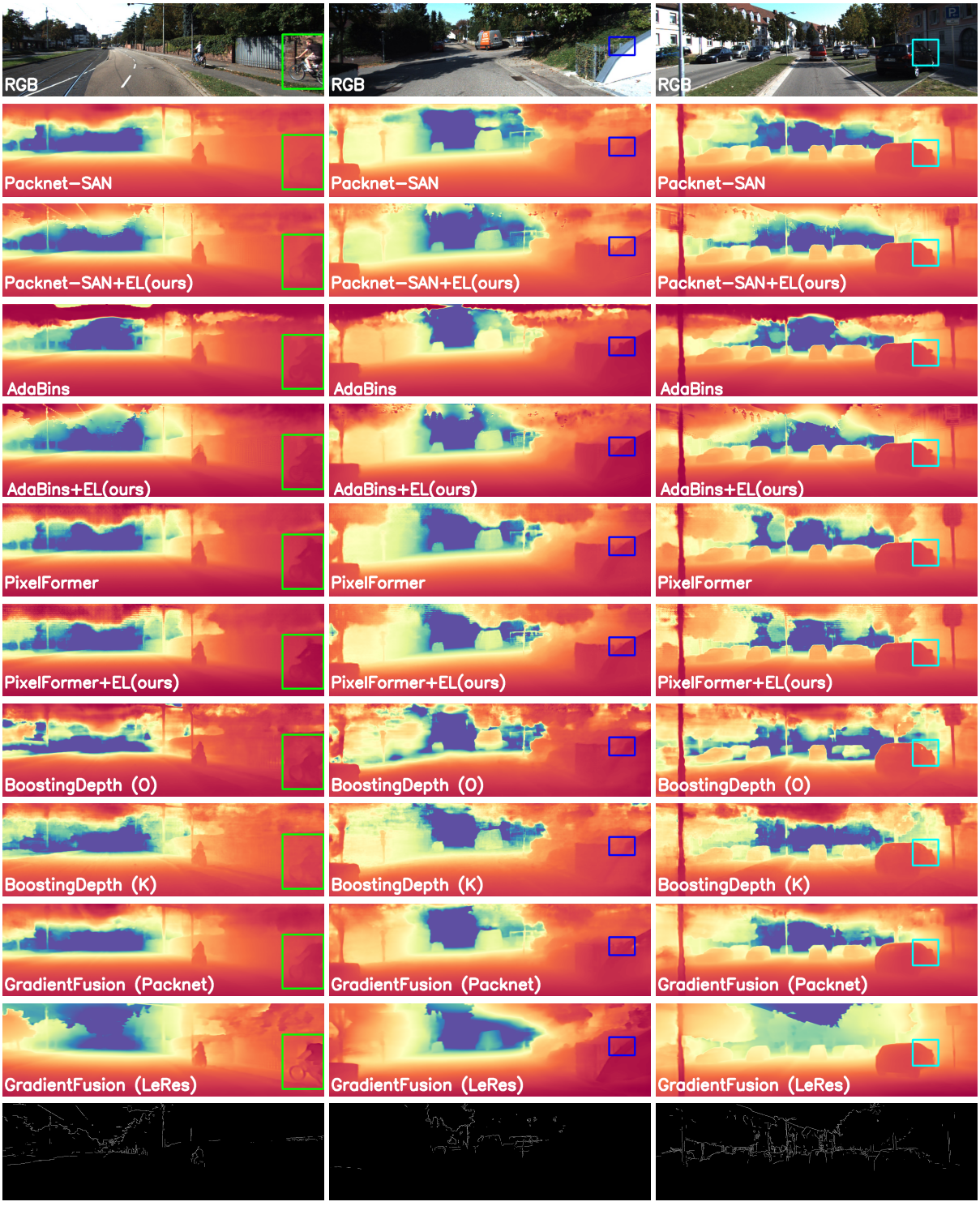

Figure 10. Full Images (zoom-ins in Figure 11) of depth predictions of Packnet-SAN, AdaBins, PixelFormer (both baseline and ours) and BoostingDepth and GradientFusion with the original depth merger and the version we trained on KITTI. The last row depicts the result of the DEE network on those images, which are taken from the Eigen KITTI testset.

DDE dataset). To ease the annotation process, we start from edge maps of panoptic (semantic + instance) segmentation

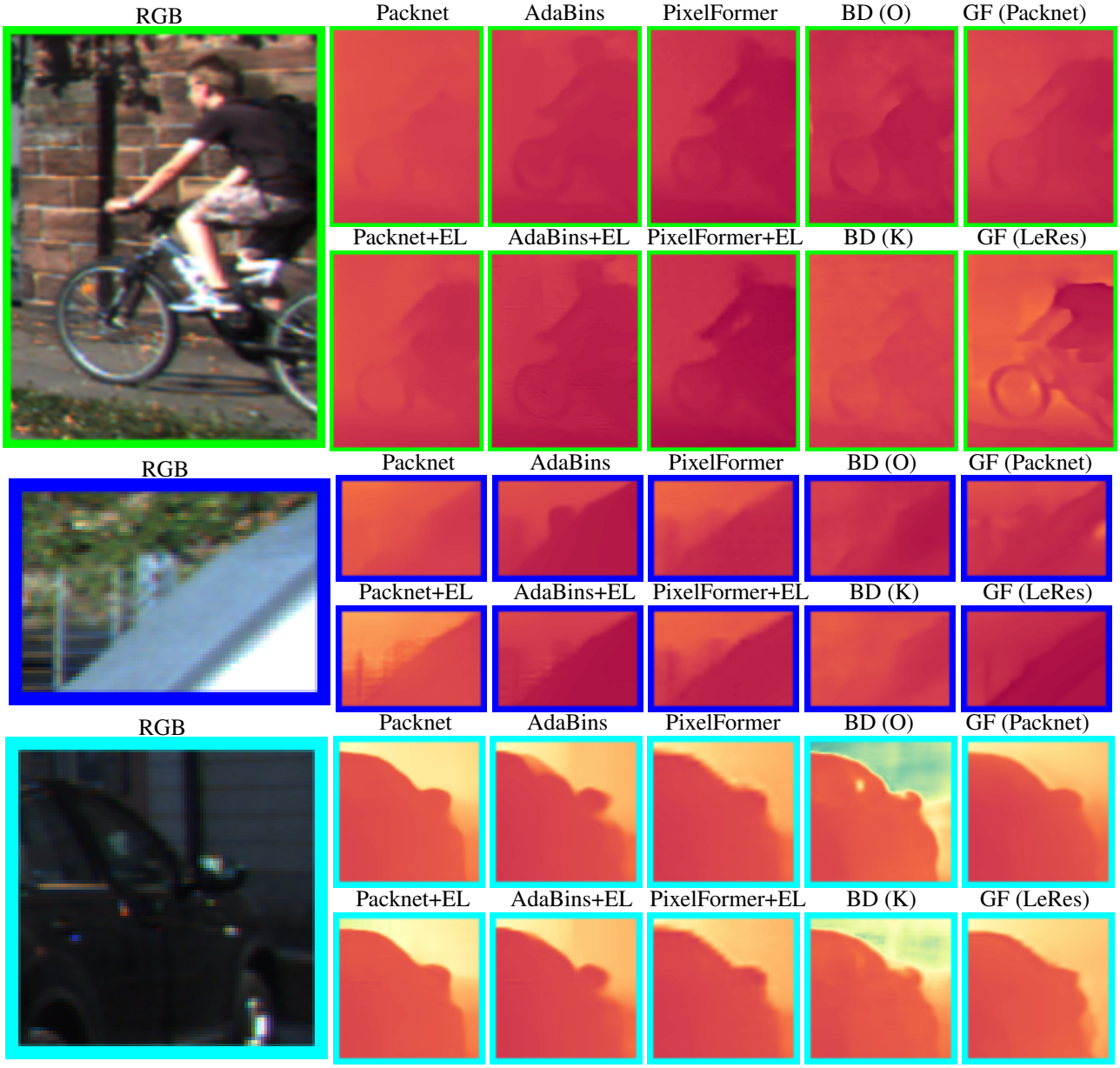

Figure 11. Zoom-ins of the depth predictions from Fig. 10. The bottom three zoom-in rows, from top to bottom, correspond to the three columns on top, from left to right. BD and GF corresponds to BoostingDepth and GradientFusion, respectively.

of those images, which yields an edge map which contains most of the depth edges in the scene (see Fig. 13), but typically has the following shortcomings: (i) Some classes, e.g., building, do not have instance segmentation so potential depth edges between different instances do not exist. (ii) Some edges between different classes, e.g., (bottom of) car and road or road and sidewalk, do not necessarily reflect discontinuities in depth. In the first case, we add the relevant depth edges (see green examples in Fig. 13), and in the second case we remove the relevant edges (see red examples in Fig. 13).

The annotators' guideline for adding or removing edges from the initial edge map from the panoptic segmentation is 4 meters. That is, a depth edge should be annotated if between the two sides of the depth edge there exists a depth discontinuity of at least 4 meters (this is estimated by the annotator). We also note that on average the number of annotated depth edges occupy $\sim 2\%$ of the image ($\sim 10K$ pixels).

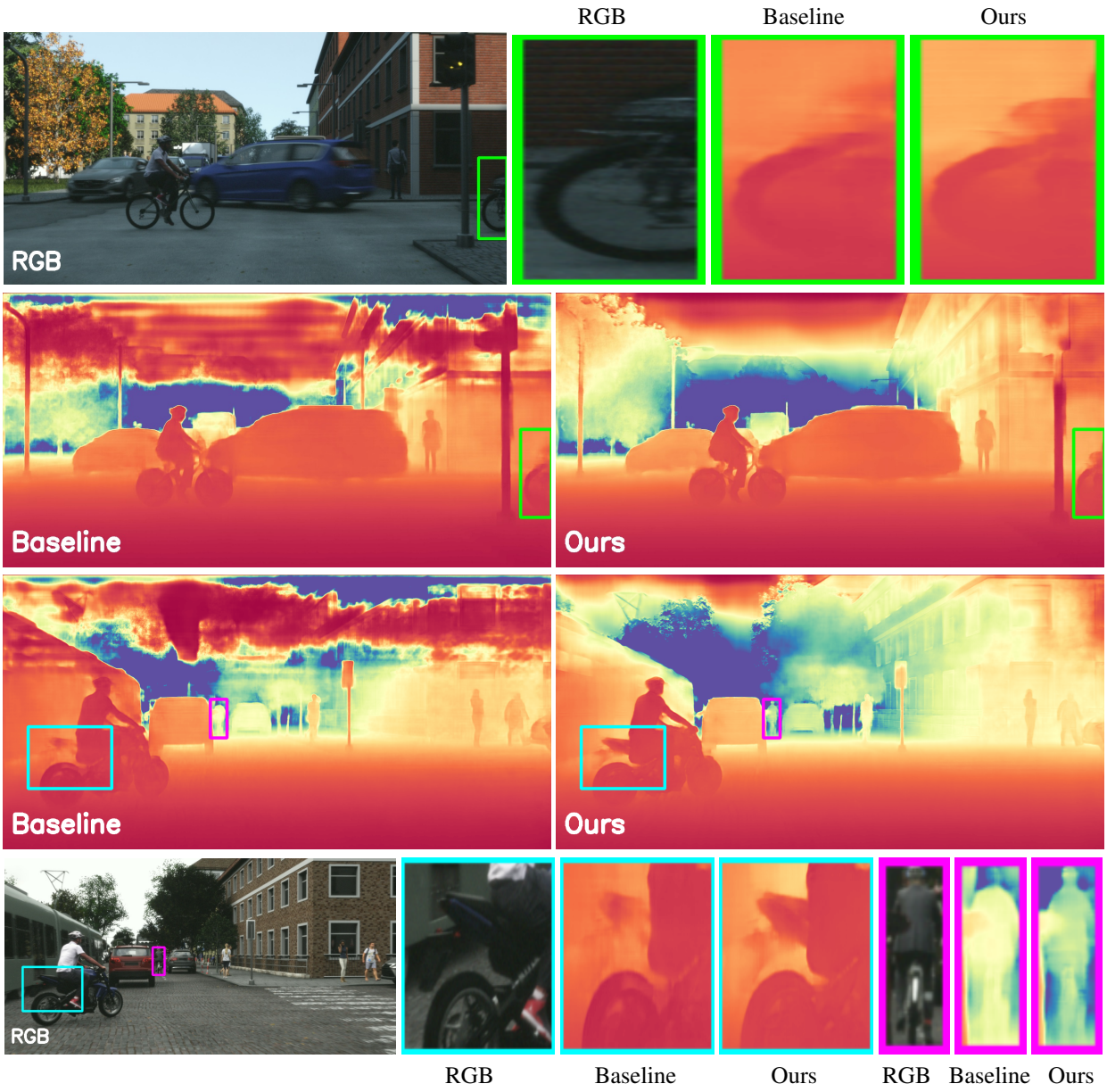

Figure 12. Examples of depth predictions of Packnet-SAN and Packnet-SAN + EL (ours) of images from the Synscapes dataset.

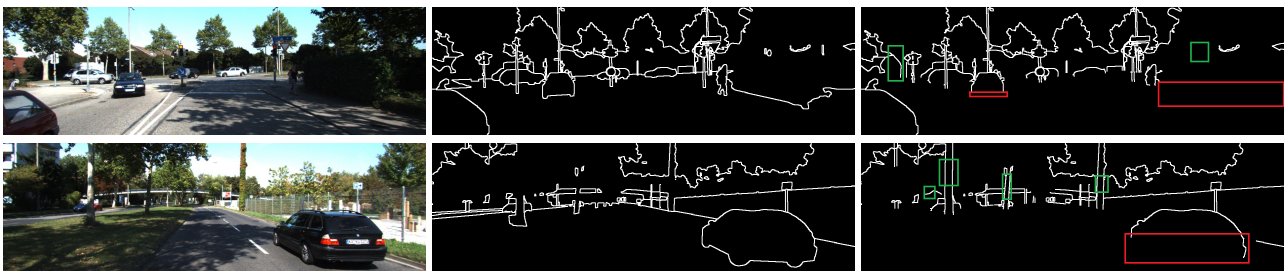

Figure 13. Examples of the annotation process (the edges are dilated for visualization purposes). (a) The RGB image. (b) The edges of the instance segmentation GT. (c) The depth edges GT, where green and red rectangles exemplifies edge addition and deletion, respectively.

6.2. RGB and RGB+LIDAR in the DEE network

As we argue in the main paper that the performance of the DEE network is significantly better when the input is RGB and LIDAR, in constrast to RGB only. In Fig. 6 we present the depth edges precision-recall graph for the DEE network on the KITTI-DE dataset, where the performance gap in favor of the RGB+LIDAR input is clearly present. Moreover, examples of the output of the DEE network is presented in the bottom row of Fig. 10.

6.3. Depth Edge Loss

As we argue in the main paper (L370), using the standard spatial image gradient often yield undesired artifacts. See the stripes over the car silhouette for a depiction of this phenomenon in Fig. 2. The quantitative performance on the KITTI-DE dataset is also somewhat lower: ARE: 4.03% and AUC (edges): 59.3% (48.13%).

6.4. Implementation Details

The DEE network follows the architecture of the U-net like PackNet-SAN [1] (Fig. 7) which was used as an MDE network. One of the beneficial properties of PackNet-SAN is its sparse encoder for the sparse LIDAR signal, which uses only sparse layers (e.g., sparse convolutions), which is suitable for our case of depth edges estimation. The DEE network is trained for five epochs on the GTA-PreSIL dataset [2]. For training the MDE using our edge loss, we set $\alpha = 0.1$ and $\alpha = 1.0$ for Packnet-SAN and AdaBins, respectively. Also, we note that AdaBins is trained only with the largest scale for both baseline and our method as in the original training.

7. Additional Examples for KITTI and DDAD

We present additional qualitative results for KITTI in Fig. 10 and for DDAD in Fig. 8 and Fig. 9.

References

[1] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. Sparse auxiliary networks for unified monocular depth prediction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11078–11088, 2021. 9

[2] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2522–2529. IEEE, 2019. 9

[3] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Korner. Evaluation of cnn-based single-image depth estimation methods. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 4

[4] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021. 3

[5] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 4