Supplementary Material

Aurélien Cecille[1,2], Stefan Duffner[2], Franck Davoine[2], Thibault Neveu[1], and Rémi Agier[1]

[1] Visual Behavior, Lyon, France
[2] INSA Lyon, CNRS, Ecole Centrale de Lyon, Universite Claude Bernard Lyon 1, Université Lumière Lyon 2, LIRIS, UMR5205, 69621 Villeurbanne, France

1 Baseline Details

We compare the GroCo method with a baseline that was made to eliminate confounding factors such as model architectures or training pipelines. As such, the baseline uses the same encoder, decoder, photometric loss and smoothness loss. The difference is that the baseline model does not use the ground prior, so in order to learn the scale the prediction we use a slightly modified version of losses presented in [5], dividing the translation scale loss $\mathcal{L}_{pose}$ by the amplitude of the scaled translation to not penalize larger translations more than smaller ones:

$$\mathcal{L}_{pose} = \left| \frac{P_{scaled} - P_{pred}}{P_{scaled}} \right| \tag{1}$$

The gradient being computed on P_{pred} only.

We also use Algorithm 1 to compute the scale, because we found that it performed better than the method of [5] as seen in Tab. 1.

1.1 Scale Computation

Algorithm 1: Scale Computation

Data: depth, ground, $\mathrm{d}_{max} = 15$, margin $= 0.1$
row_q3 $\leftarrow$ **third_quartile**(depth, axis='width');
ratio $\leftarrow \frac{\text{ground}}{\text{depth}}$;
ground_mask $\leftarrow$ (ground $< \mathrm{d}_{max}$) $\wedge$ ($\frac{|\text{row\_q3} - \text{depth}|}{\text{row\_q3}} <$ margin);
depth_scale $\leftarrow$ **median**(ratio[ground_mask]);
Result: depth_scale

In supervised settings, the scale is usually computed by taking the ratio between the median of the ground truth lidar depth and the median of the predicted depth. In Algorithm 1 we adopt a similar idea but replace the ground truth by the theoretical ground plane depth. To make it work, we segment the region where this prior should be compared to the predicted depth. To do this,

Fig. 1: Examples of ground masks using Algorithm 1 with a 10% threshold and a maximum distance of 15m.

we use an algorithm that is robust to changes of camera view points, observing that in driving scenarios the ground points are usually the one that are the furthest away on each row of the image. Using this assumption we segment pixels with depths close to the 3rd quartile on each row of the up-to-scale depth, thereby removing potential outliers like potential pothole. Fig. 1 also shows that we keep only the pixels up to a certain distance since they are more likely the one respecting the flat ground hypothesis.
Note that when we use roll augmentation, images are inversely rotated before the algorithm to keep the horizon line horizontal.

Tab. 1 shows that this method outperforms the one of [5] on the improved KITTI Eigen test set [4] with the up-to-scale prediction of the Monodepth2 model [2]. We compare the standard deviation σ between the scale estimated by the ground truth and the one estimated by the competing scale algorithms .
In order to consider overestimation in the same way as underestimation the metric $\mathcal{M}$ is computed as follows for each image:

$$\mathcal{M}(scale_{pred}, scale_{gt}) = \max\left(\frac{scale_{pred}}{scale_{gt}}, \frac{scale_{gt}}{scale_{pred}}\right) - 1 \tag{2}$$

Table 1: Comparison of the standard deviation σ between the scale recovery algorithms and the ground truth. Evaluated on the improved eigen test split, using the default Monodepth2 [2] model.

Method	σ
Scale Recovery [5]	0.1022
Algorithm 1	0.0547

2 Additional Visualizations

2.1 Videos

We provide two videos on sequences of the DDAD validation dataset, with both the GroCo model trained on KITTI [1] (in zero-shot on DDAD) and the one trained on DDAD [3]. These videos show a bird's eye view (BEV) of the image that is projected thanks to the depth estimation.

2.2 Height Images

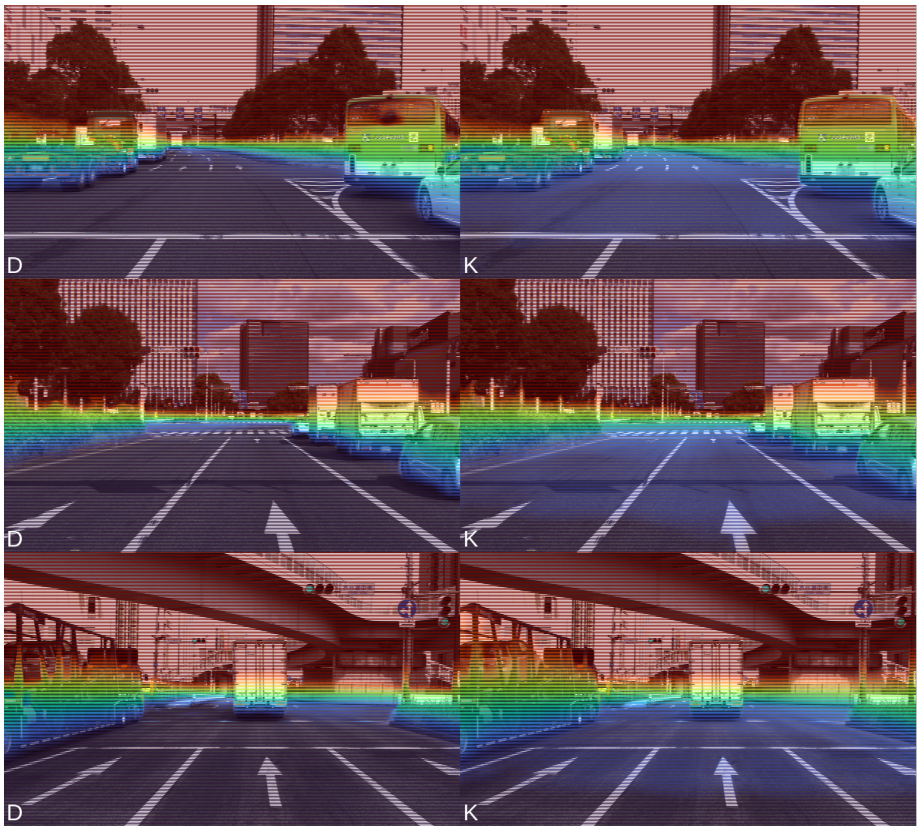

Fig. 2: Pixels heights up to 3m computed from GroCo's predicted depth. Images with (D) are trained on DDAD while the ones with (K) are trained on KITTI.

References

1. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
2. Godard, C., Aodha, O.M., Firman, M., Brostow, G.: Digging into self-supervised monocular depth estimation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3827–3837. IEEE. https://doi.org/10.1109/ICCV.2019.00393, https://ieeexplore.ieee.org/document/9009796/
3. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3D packing for self-supervised monocular depth estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2482–2491. IEEE. https://doi.org/10.1109/CVPR42600.2020.00256, https://ieeexplore.ieee.org/document/9156708/
4. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: International Conference on 3D Vision (3DV) (2017)
5. Wagstaff, B., Kelly, J.: Self-supervised scale recovery for monocular depth and egomotion estimation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2620–2627. https://doi.org/10.1109/IROS51168.2021.9635938, https://ieeexplore.ieee.org/document/9635938, ISSN: 2153-0866