上海科技大学
ShanghaiTech University

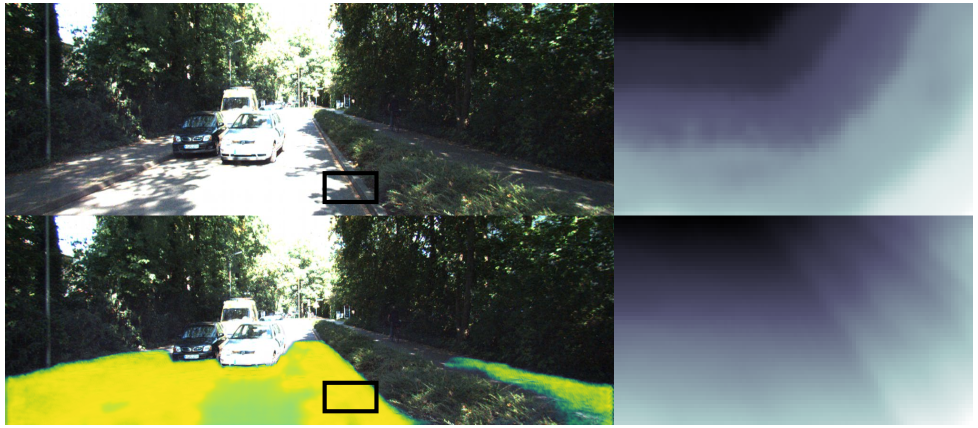
SVIP LAB
ShanghaiTech Vision and Intelligent Perception Lab

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

THU-PM-078

PlaneDepth: Self-supervised Depth Estimation via Orthogonal Planes

Authors: Ruoyu Wang, Zehao Yu, Shenghua Gao

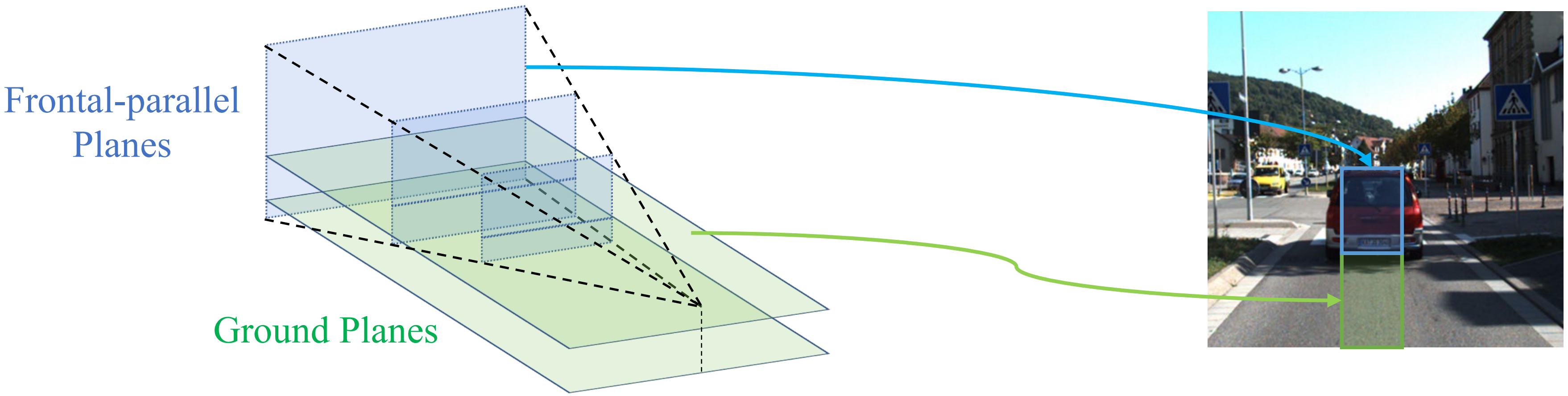


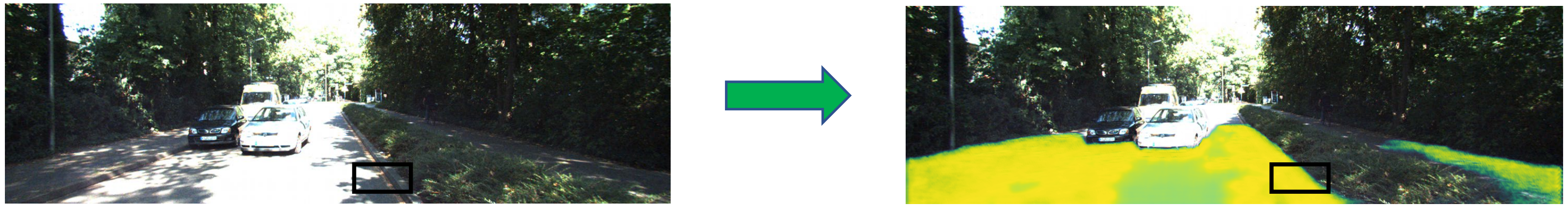
Email: wangry3@shanghaitech.edu.cn

Github: https://github.com/svip-lab/PlaneDepth

Preview

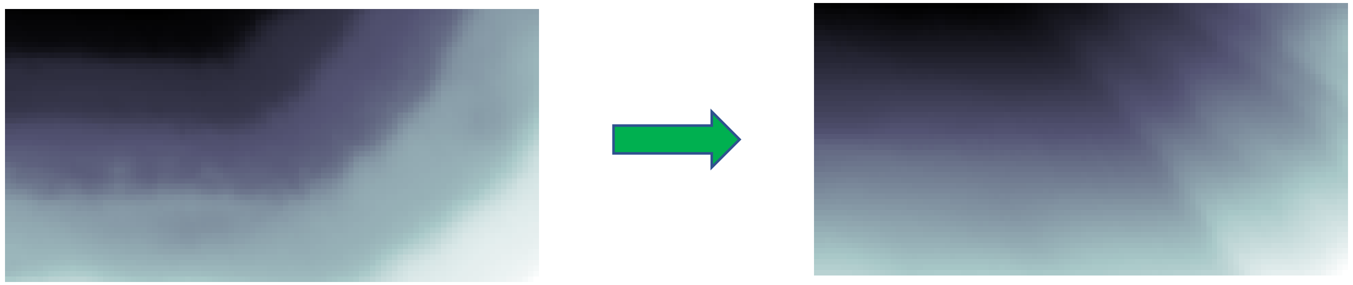
Preview - Orthogonal Planes



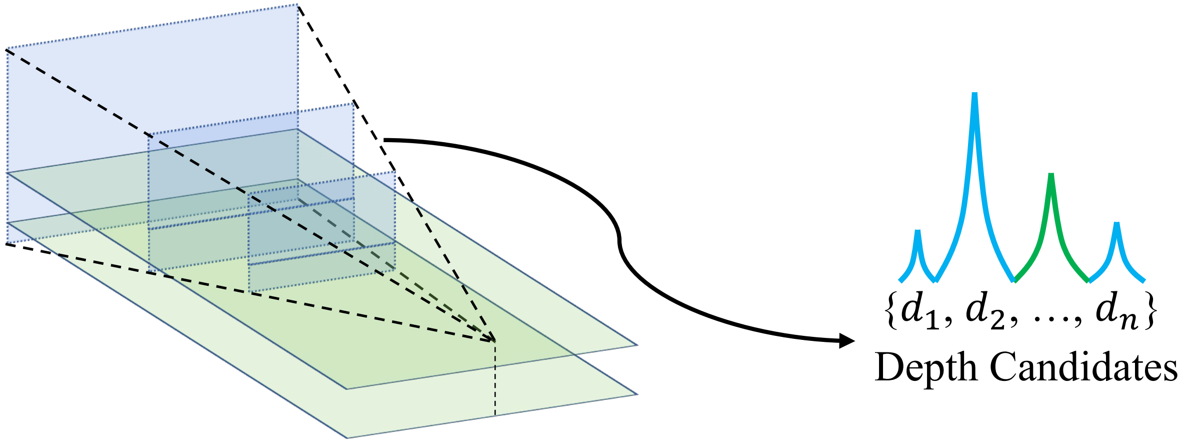


Frontal-parallel Planes

Orthogonal Planes

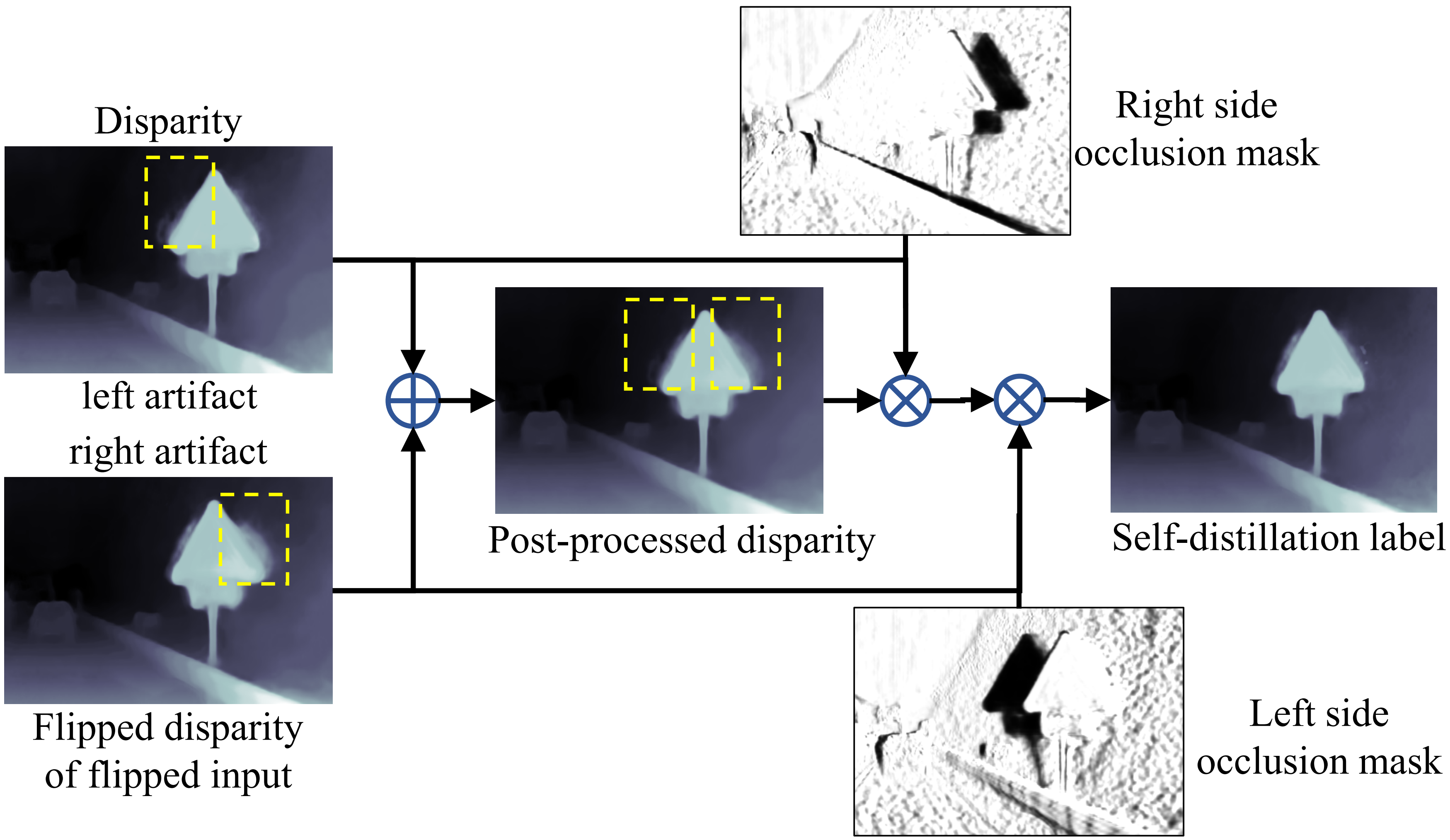


Preview - Mixture Laplace Loss



$$p(d) = \sum \frac{\hat{\pi}_i e^{\frac{|d - d_i|}{\hat{\sigma}_i}}}{2\hat{\sigma}_i}$$

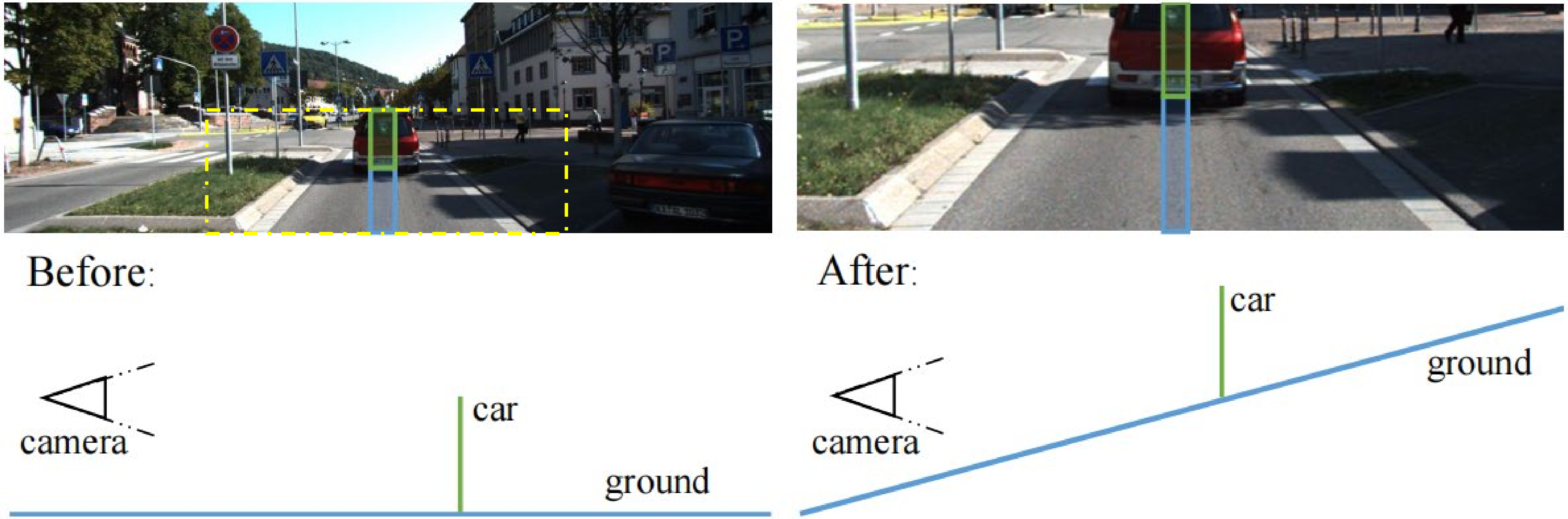
Preview - Self-distillation



Details

Details - Ground

- The widely used resizing cropping augmentation cause ground slopes.
  - We assume that the camera intrinsics remain consistent.



Details - Ground

- Rectify the ground planes to ensure that they are always parallel to the ground.

Plane Definition:

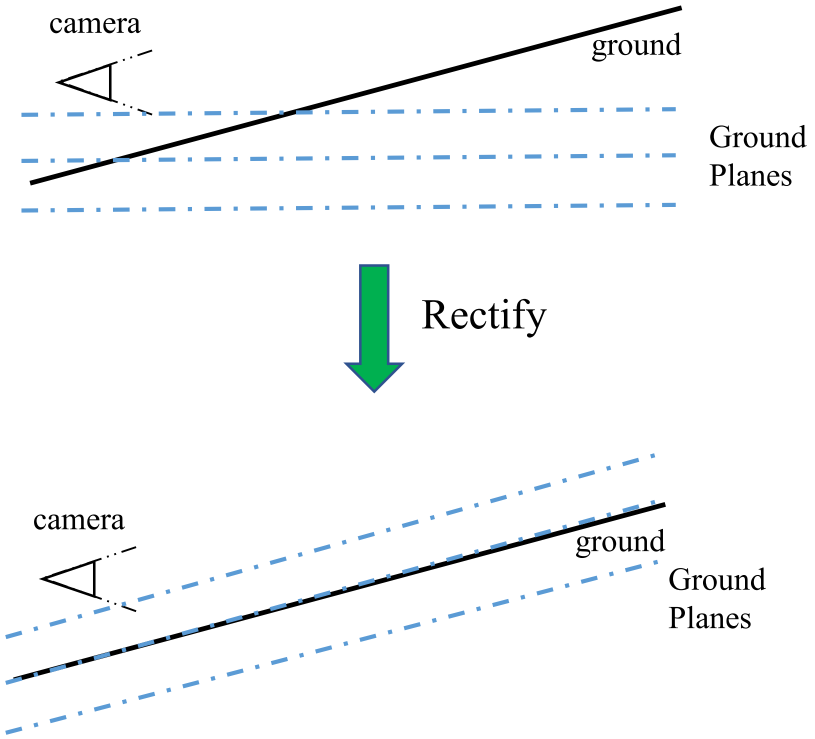
$$\mathbf{n}_i^T \mathbf{w} - \delta_i = 0, \quad i = 0, 1, \dots, N-1$$

Rectifying Matrix:

$$\mathbf{R_C} = \begin{bmatrix} 1 & & \frac{c_x - p_x}{f_x} \\ & 1 & \frac{c_y - p_y}{f_y} \\ & & f_s \end{bmatrix}$$

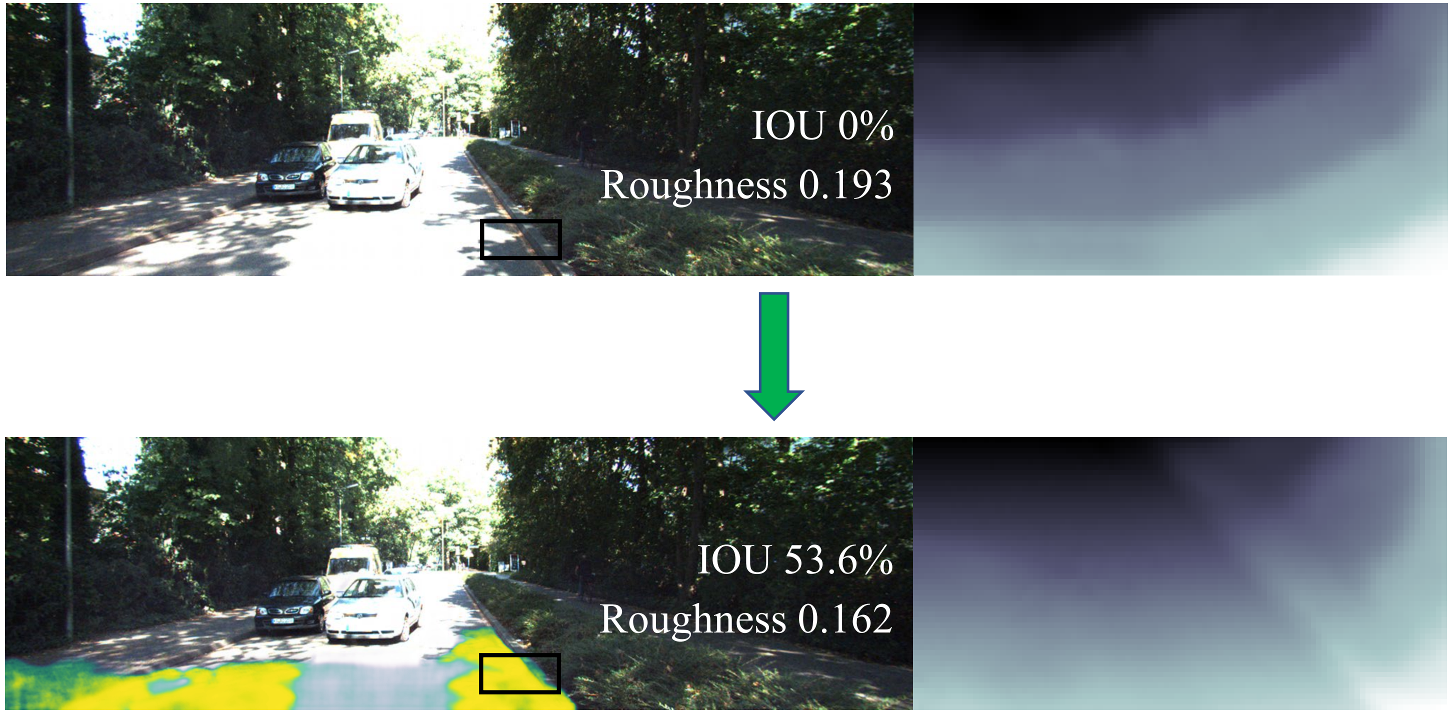
Rectified Parameters:

$$\tilde{\mathbf{n}} = \frac{\mathbf{R_C}^{-T}}{||\mathbf{R_C}^{-T}\mathbf{n}||}\mathbf{n} \qquad \tilde{\delta} = \frac{\delta}{||\mathbf{R_C}^{-T}\mathbf{n}||}$$

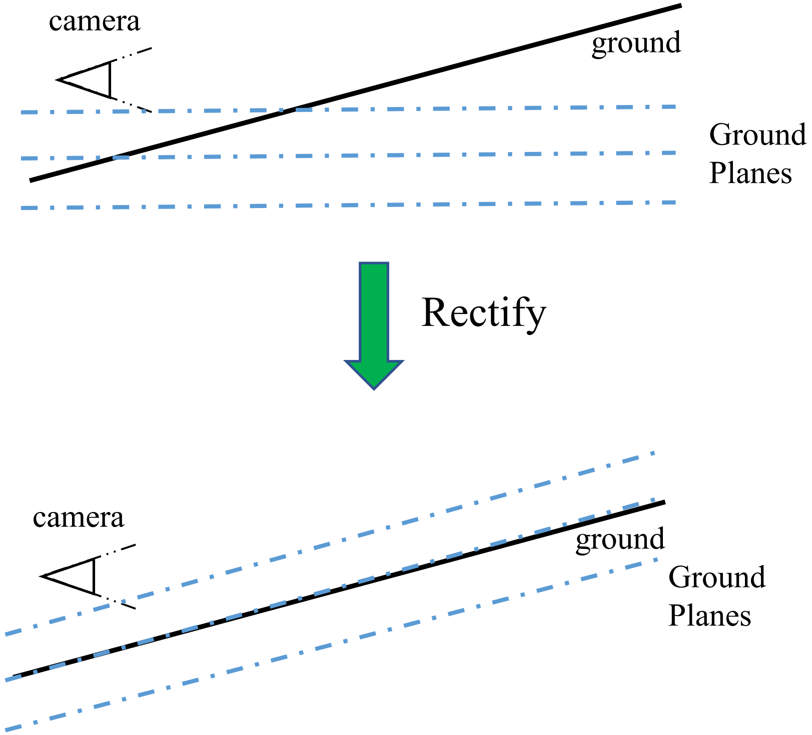


Details - Ground

- Rectify the ground planes to ensure that they are always parallel to the ground.

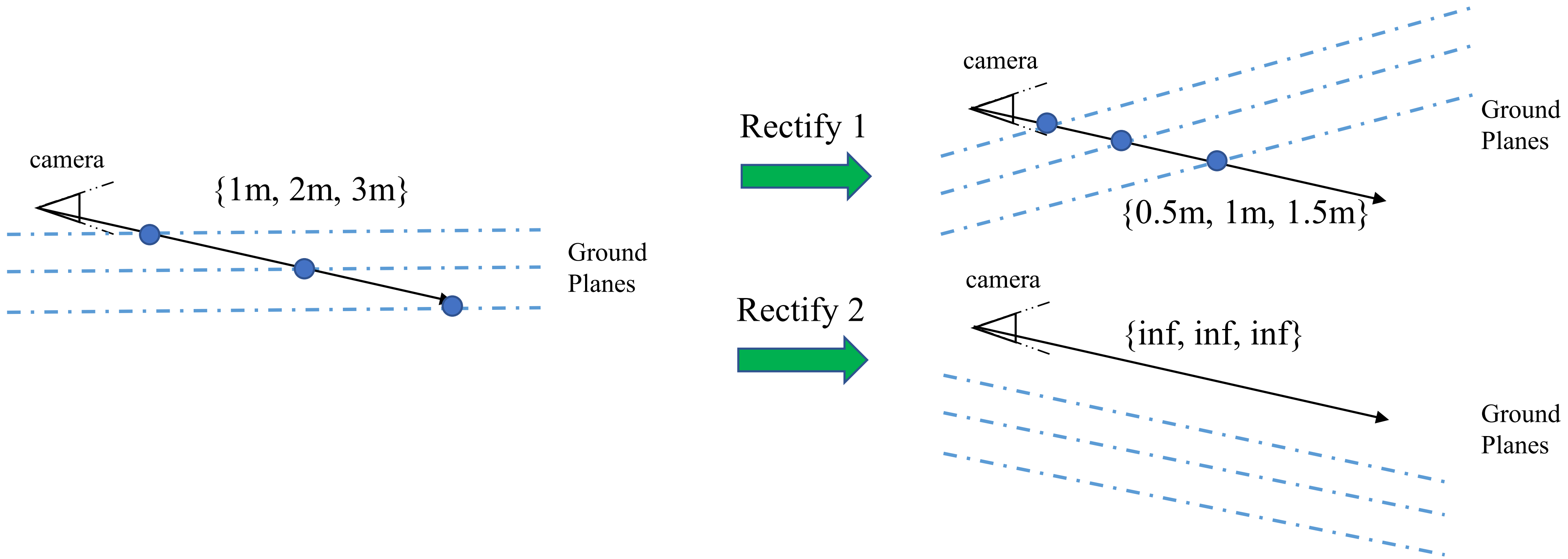


Still not good



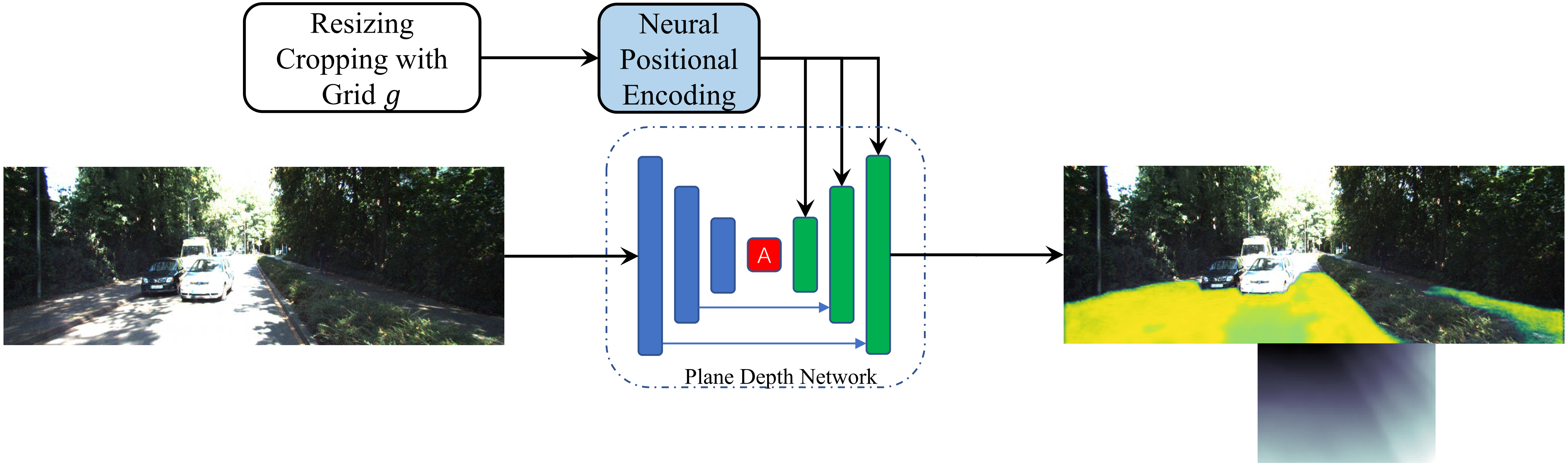
Details - Ground

- Ground depth candidates of each pixel vary due to rectification.
  - Make it hard to learn



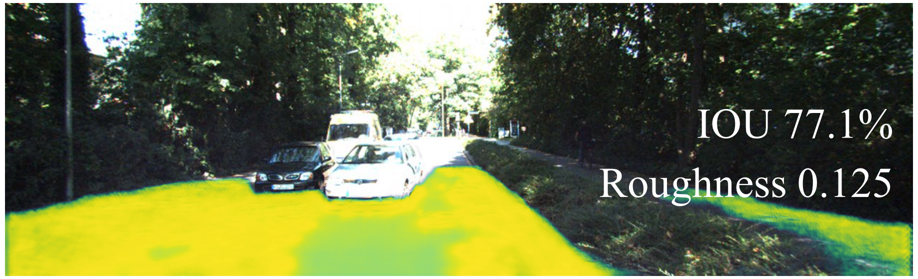
Details - Ground

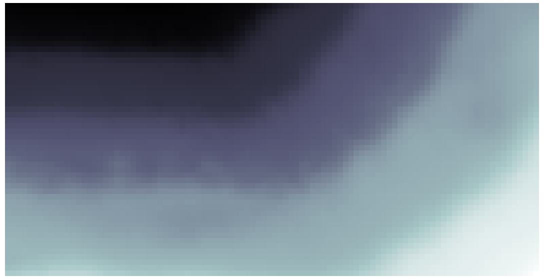
- Input the parameters of resizing cropping during training.
  - Tell the CNN how the rectification is.

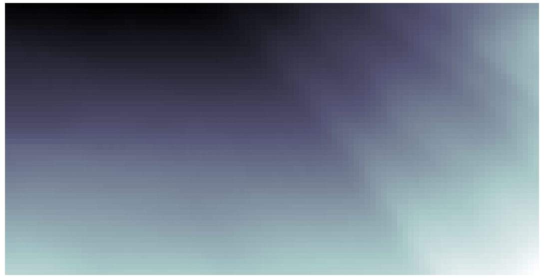


Details - Ground

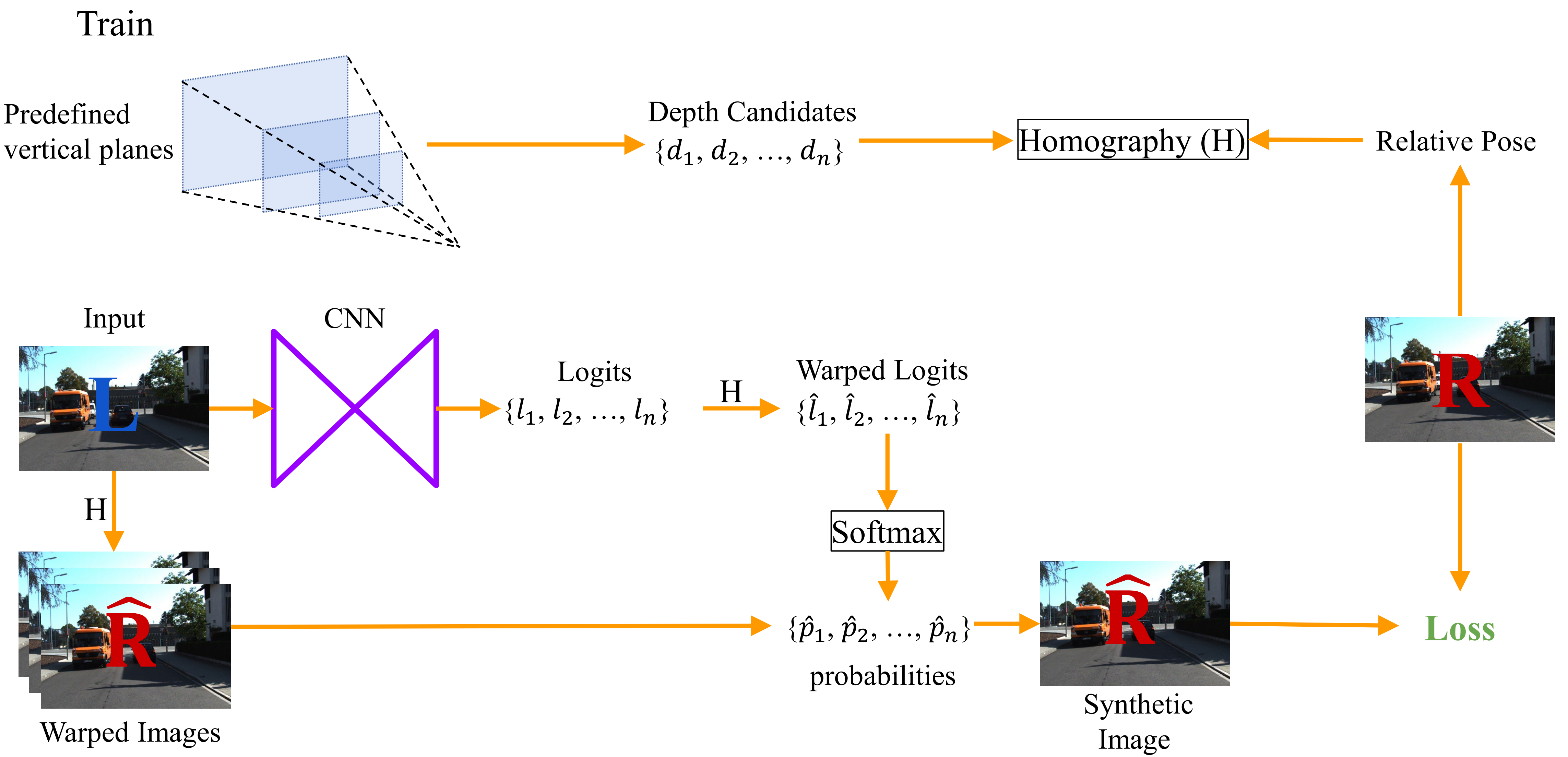
- Input the parameters of resizing cropping during training.
  - Tell the CNN how the rectification is.







Details - Mixture Laplace Loss



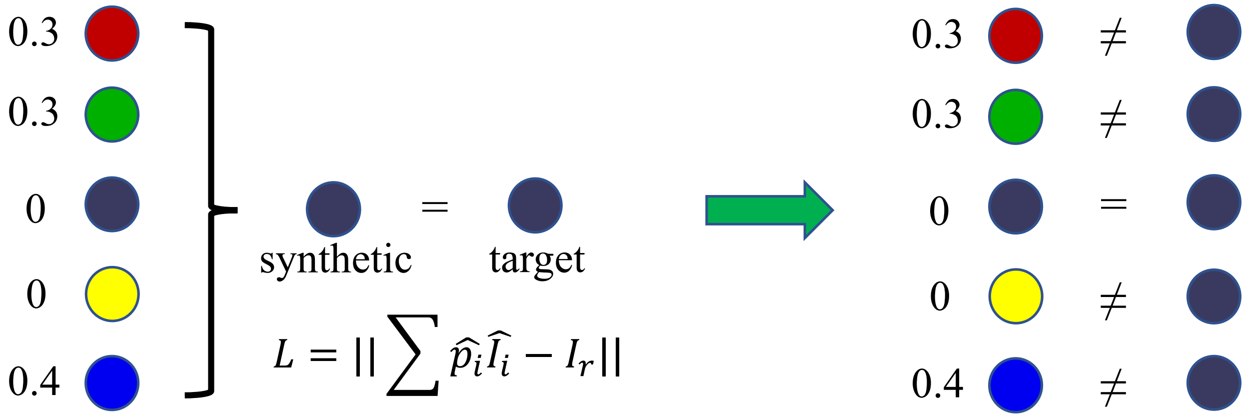
Forget About the LiDAR: Self-Supervised Depth Estimators with MED Probability Volumes [Gonzalez et al. 2020]

Details - Mixture Laplace Loss

- It is non-trivial to compute L1 loss on the synthetic image.

Details - Mixture Laplace Loss

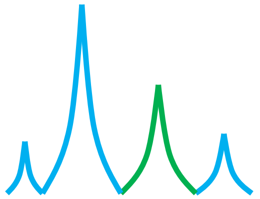
- Compute photometric error on each warped plane before composing.



$$L = \left|\left|\sum \hat{p}_i \hat{I}_i - I_r\right|\right|$$

$$L = \sum \hat{p}_i \left|\left|\hat{I}_i - I_r\right|\right|$$

- Mixture Laplace Distribution



$\{d_1, d_2, \ldots, d_n\}$

Depth Candidates

$$L = -\log \sum \frac{\hat{\pi}_i e^{\frac{-||\hat{I}_i - I_r||}{\hat{\sigma}_i}}}{2\hat{\sigma}_i}$$

$$p_i = \sum \frac{\hat{\pi}_j e^{\frac{|d_i - d_j|}{\hat{\sigma}_j}}}{2\hat{\sigma}_j}$$

Post-process and Self-distillation

- Self-distillation [Gonzalez et al. 2020]:
  - Solve occlusion effect

L

R

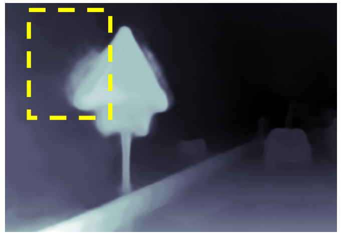
Input

Loss

Occlusion only occurs on the left side of objects in the left view.

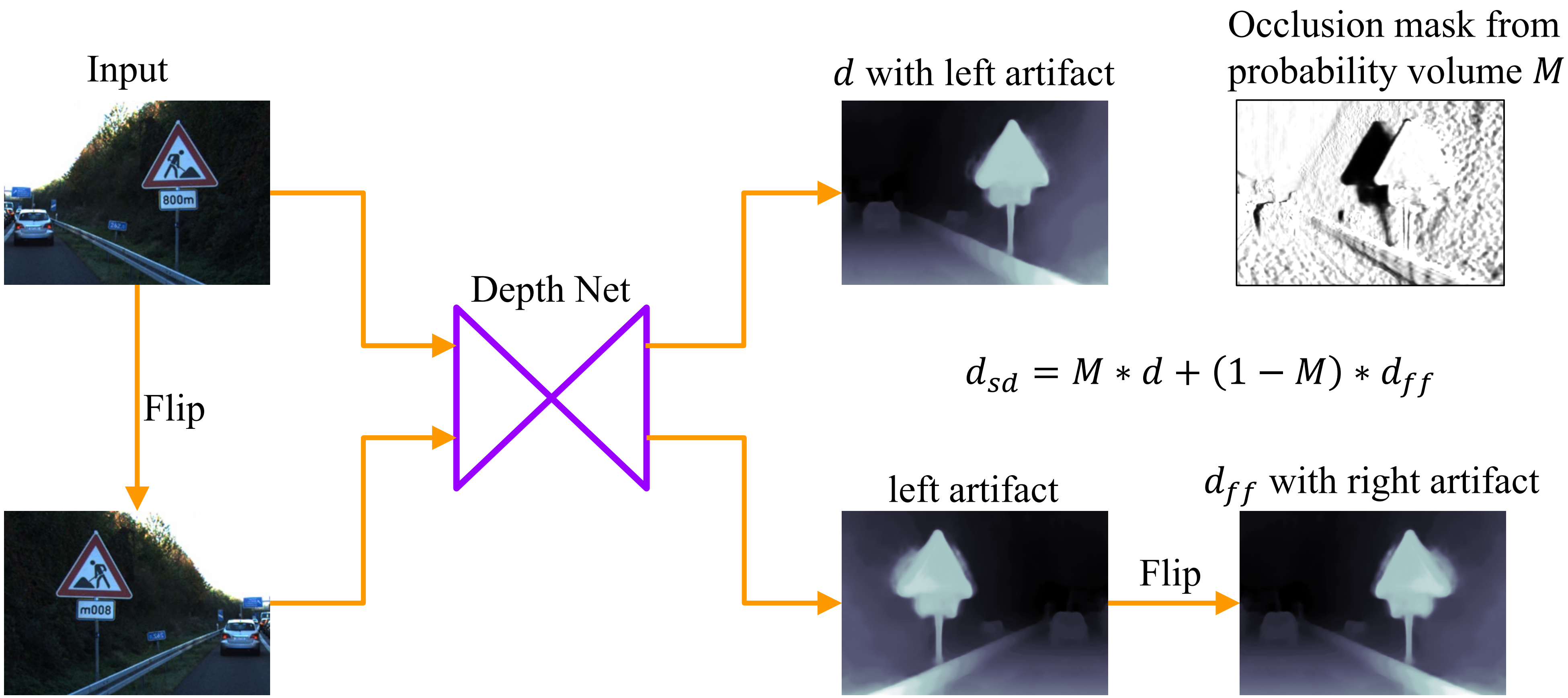
Self-distillation [Gonzalez et al. 2020]

- If we only input left views and find matchings in the right view, all artifacts caused by occlusion will appear at left of objects.

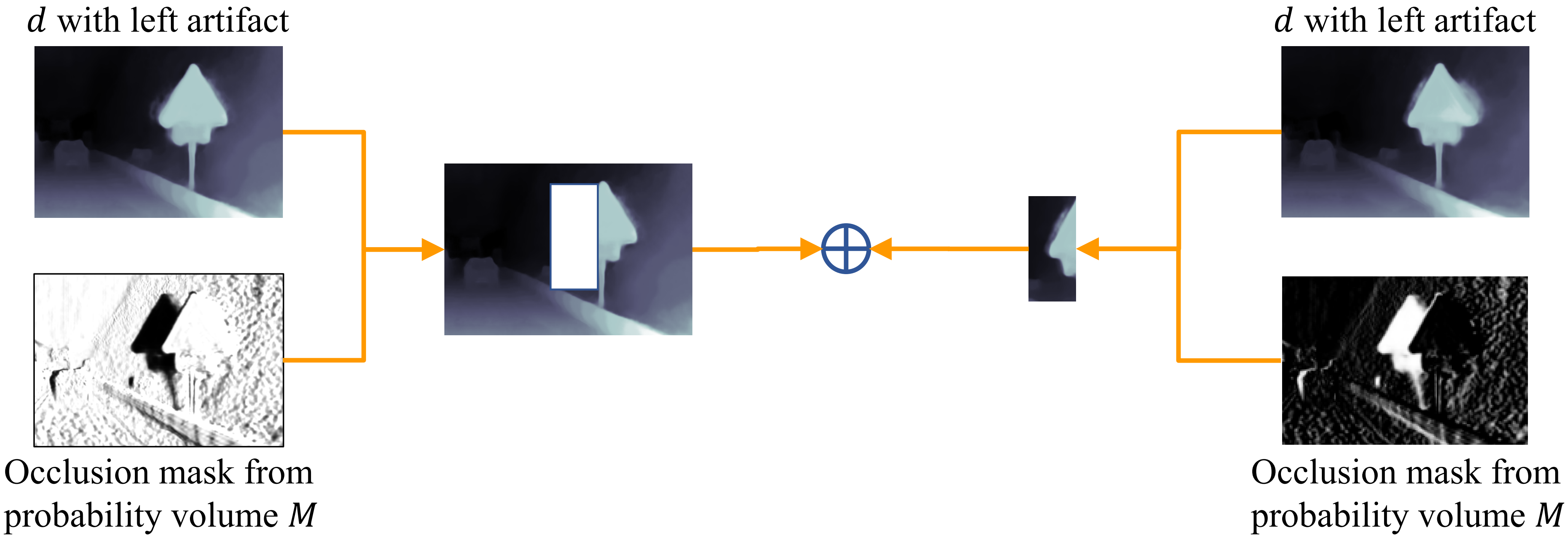


Self-distillation [Gonzalez et al. 2020]

- If we only input left views and find matchings in the right view, all artifacts caused by occlusion will appear at left of objects.



Self-distillation [Gonzalez et al. 2020]

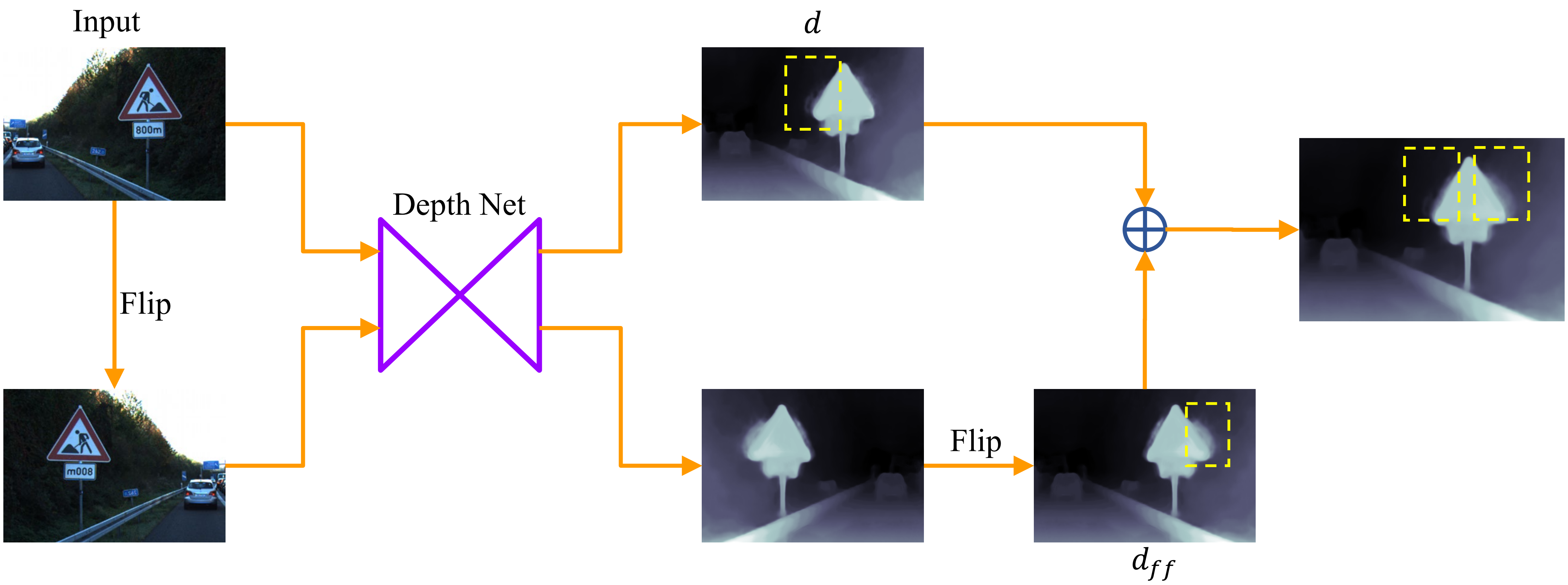


$$d_{sd} = M * d + (1 - M) * d_{ff}$$

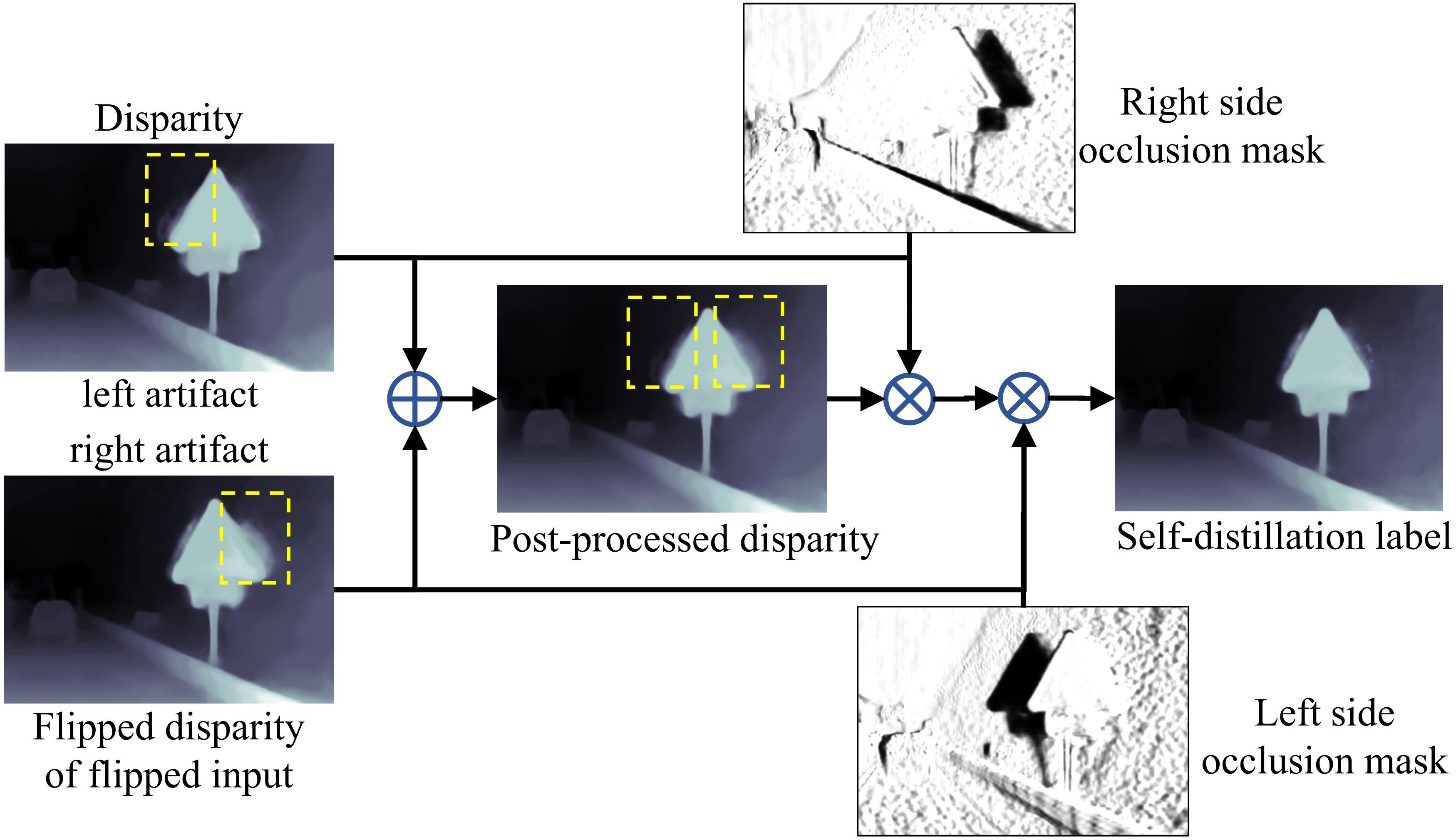
Post-process and Self-distillation

- Post-process [Godard et al. 2017]:
  - Improve results

$$d_{\text{pp}} = \frac{1}{2}(d + d_{\text{ff}})$$



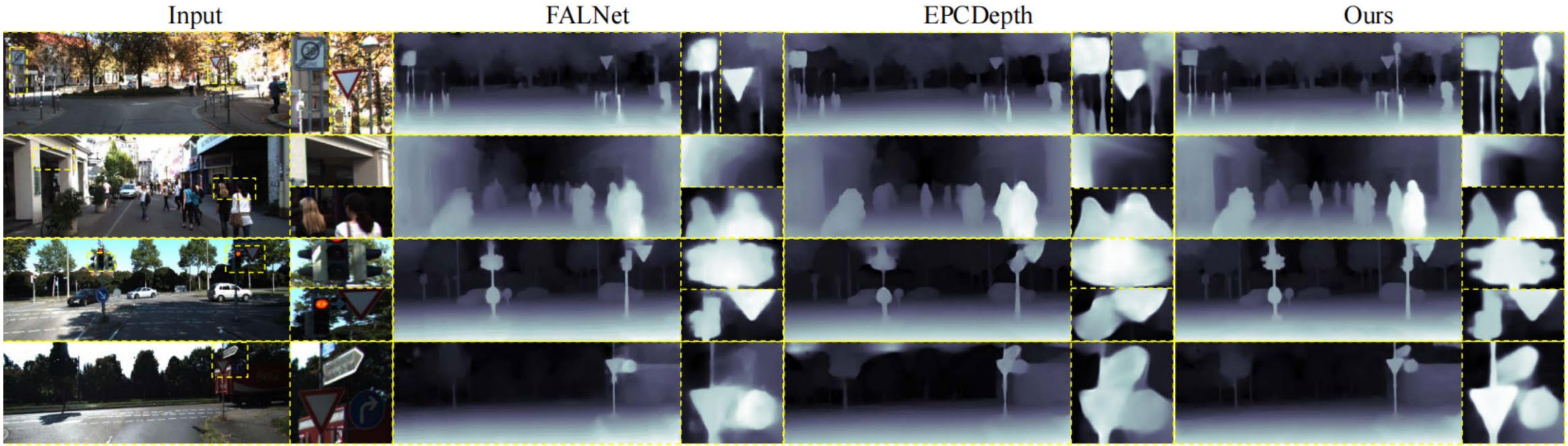
Post-process and Self-distillation



Results

Methods	PP	Network	Resolution	Train	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	A1↑	A2↑	A3↑
Raw Eigen Test Set [8]											
Monodepth2 [12]	✓	ResNet-50	1024×320	S	0.097	0.793	4.533	0.181	0.896	0.965	0.983
DepthHint [40]	✓	ResNet-50	1024×320	S	0.096	0.710	4.393	0.185	0.890	0.962	0.981
EPCDepth [29]	✓	ResNet-50	1024×320	S	0.091	0.646	4.207	0.176	0.901	0.966	0.983
OCFDNet [48]	✓	ResNet-50-A	1280×384	S	0.090	0.564	4.007	0.172	0.903	0.967	0.984
FALNet [14]		FALNet	1280×384	S	0.097	0.590	3.991	0.177	0.893	0.966	0.984
FALNet [14]	✓	FALNet	1280×384	S	0.093	0.564	**3.973**	0.174	0.898	0.967	**0.985**
PLADENet [13]		PLADENet	1280×384	S	0.092	0.626	4.046	0.175	0.896	0.965	0.984
PLADENet [13]	✓	PLADENet	1280×384	S	0.089	0.590	4.008	0.172	0.900	0.967	**0.985**
Ours		ResNet-50-A	1280×384	S	0.085	0.563	4.023	0.171	0.910	**0.968**	0.984
Ours	✓	ResNet-50-A	1280×384	S	**0.084**	**0.549**	3.981	**0.169**	**0.911**	**0.968**	0.984
Ours†	✓	ResNet-50-A	1280×384	S	0.083	0.533	3.919	0.167	0.913	0.969	0.985
Monodepth2 [12]	✓	ResNet-18	1024×320	MS	0.104	0.775	4.562	0.191	0.878	0.959	0.981
DepthHint [40]	✓	ResNet-18	1024×320	MS	0.098	0.702	4.398	0.183	0.887	**0.963**	**0.983**
FeatureNet [36]		ResNet-50	1024×320	MS	0.099	0.697	4.427	0.184	0.889	**0.963**	0.982
Ours*		ResNet-50-A	1280×384	MS	0.092	0.601	4.188	0.184	0.893	0.961	0.981
Ours*	✓	ResNet-50-A	1280×384	MS	**0.090**	**0.584**	**4.130**	**0.182**	**0.896**	0.962	0.981
Improved Eigen Test Set [39]											
Monodepth2 [12]	✓	ResNet-50	1024×320	S	0.077	0.455	3.489	0.119	0.933	0.988	0.996
DepthHint [40]	✓	ResNet-50	1024×320	S	0.074	0.364	3.202	0.114	0.936	0.989	0.997
OCFDNet [48]	✓	ResNet-50-A	1280×384	S	0.070	0.262	2.786	0.103	0.951	0.993	0.998
FALNet [14]	✓	FALNet	1280×384	S	0.071	0.281	2.912	0.108	0.943	0.991	0.998
PLADENet [13]	✓	PLADENet	1280×384	S	0.066	0.272	2.918	0.104	0.945	0.992	0.998
Ours		ResNet-50-A	1280×384	S	**0.063**	0.245	2.718	0.096	0.959	0.994	0.998
Ours	✓	ResNet-50-A	1280×384	S	**0.063**	**0.236**	**2.674**	**0.095**	**0.960**	**0.994**	**0.999**
Ours†	✓	ResNet-50-A	1280×384	S	0.062	0.227	2.609	0.093	0.961	0.995	0.999

Results



Thanks!

Email: wangry3@shanghaitech.edu.cn

Github: https://github.com/svip-lab/PlaneDepth