Supplemental Materials on

Depth Map Decomposition for Monocular Depth Estimation

Jinyoung Jun[1][0000−0002−8256−6846], Jae-Han Lee[2][0000−0002−3674−4023], Chul Lee[3][0000−0001−9329−7365], and Chang-Su Kim[1][0000−0002−4276−1831]

[1] School of Electrical Engineering, Korea University, Seoul, Korea
[2] Gauss Labs Inc.
[3] Department of Multimedia Engineering, Dongguk University, Seoul, Korea
jyjun@mcl.korea.ac.kr, jaehan.lee@gausslabs.ai,
chullee@dongguk.edu, changsukim@korea.ac.kr

S1. Detailed Network Architecture

We adopt EfficientNet-B5 [47] as the backbone for the shared encoder, which processes a $3 \times 384 \times 512$ RGB image to yield a $2048 \times 12 \times 16$ high-level feature. Fig. S1 shows the network structures of G-Net, N-Net, and M-Net. Note that EfficientNet-B5 has six reduction levels. We use the output features of the second to fifth reduction levels for the skip connection using the concatenation. The detailed structure of the channel attention layer [15] is shown in Fig. S2.

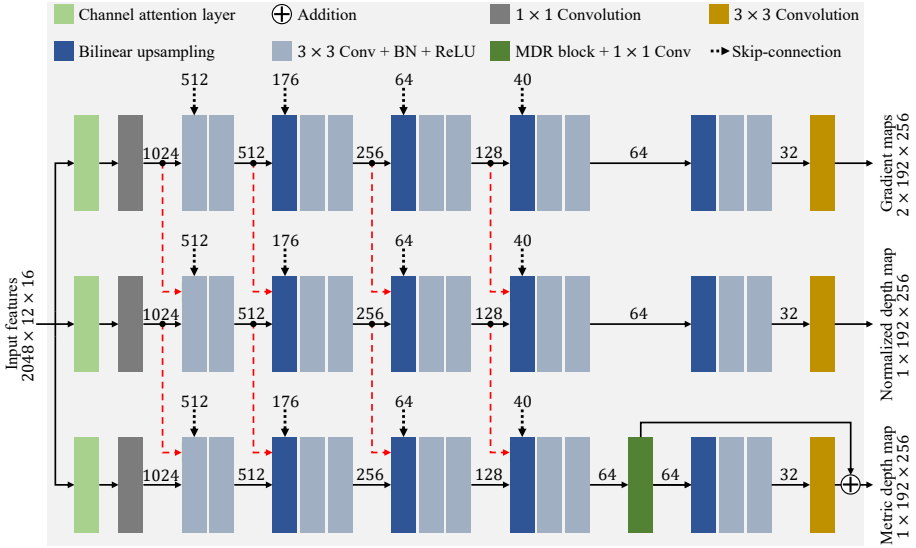

Fig. S1: Network structures of G-Net, N-Net, and M-Net. Each number denotes the number of channels of a feature tensor.

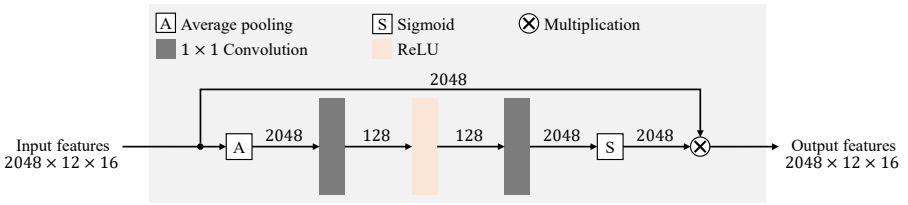

Fig. S2: Detailed structure of the channel attention layer. Each number denotes the number of channels of a feature tensor.

S2. Encoder Backbones

We test the proposed algorithm using different encoder backbones on the NYUv2 dataset [45]. Table S1 compares the proposed algorithm with the conventional algorithms: Lee and Kim [28], and Bhat *et al.* [1]. It is observed that the proposed algorithm performs better than the conventional algorithms when using the same backbone network. Also, the proposed algorithm outperforms [28] meaningfully and comparable to [1] when using only 17K training images.

Table S1: Depth estimation results on NYUv2 using various encoder backbone.

Method	Encoder backbone	#	RMSE	REL	log 10	δ_1	δ_2	δ_3
Lee and Kim [28]	PNASNet-5 [36]	58K	0.430	0.119	0.050	0.870	0.974	0.993
Proposed	PNASNet-5 [36]	17K	**0.403**	**0.109**	**0.048**	**0.882**	**0.983**	**0.997**
Bhat *et al.* [1]	EfficientNet-B5 [47]	50K	0.364	0.103	0.044	0.903	0.984	**0.997**
Proposed	EfficientNet-B5 [47]	17K	0.370	0.103	0.045	0.903	**0.986**	**0.997**
Proposed	EfficientNet-B5 [47]	51K	**0.362**	**0.100**	**0.043**	**0.907**	**0.986**	**0.997**

S3. Loss Functions

The weights for $\mathcal{L}_G$ and $\mathcal{L}_N$ are set to 1, and those for $\mathcal{L}_{M_x}$, $\mathcal{L}_{M_y}$, $\mathcal{L}_{N_x}$ and $\mathcal{L}_{N_y}$ are set to 0.2, as in [33, 51]. Also, the weights for $\mathcal{L}_{\mu_M}$, $\mathcal{L}_M$, and $\mathcal{L}_{\log_M}$ are 1, 0.5, 0.5, respectively.

Note that we use $\mathcal{L}_{\log_M}$ to alleviate the problem that $\mathcal{L}_M$ is less effective for distant depths. Table S2 compares the performances with and without $\mathcal{L}_{\log_M}$ for pixels whose ground-truth depths are farther or nearer than 5m. The pixels farther than 5m account for 5.17% of all pixels. We see that the performances improve in most metrics when $\mathcal{L}_{\log_M}$ is applied in addition to $\mathcal{L}_M$.

Table S2: Efficacy of $\mathcal{L}_{\log_M}$ according to the ground-truth depth range.

Distance	Setting	RMSE	REL	log 10	δ_1	δ_2	δ_3
> 5m	$\mathcal{L}_M$	0.755	0.096	0.046	0.879	0.978	**0.999**
	$\mathcal{L}_{\log_M}$	0.749	**0.095**	**0.045**	0.883	**0.979**	**0.999**
	$\mathcal{L}_M + \mathcal{L}_{\log_M}$	**0.743**	**0.095**	**0.045**	**0.888**	**0.979**	0.998
≤ 5m	$\mathcal{L}_M$	0.301	**0.098**	0.043	0.911	0.988	**0.998**
	$\mathcal{L}_{\log_M}$	0.298	0.099	**0.042**	**0.914**	0.988	**0.998**
	$\mathcal{L}_M + \mathcal{L}_{\log_M}$	**0.296**	**0.098**	**0.042**	**0.914**	**0.989**	**0.998**

S4. Efficacy of μ_M

To evaluate the efficacy of μ_M, we define the depth mean error as

$$|\mu(\hat{M}) - \mu(M)| \tag{S1}$$

where M and $\hat{M}$ denote the ground-truth and predicted depth maps, respectively. Also, $\mu(\cdot)$ denote the mean of valid pixel depths in a depth map. We use $\mu(\cdot)$ rather than μ_M since the ground truth depth map in NYUv2 is not dense.

Instead of MDR, we train an alternative regressor using the output of the shared encoder after average pooling followed by fully connected layers to predict μ_M. By comparing the first and the second rows in Table S3, we see that without MDR, the explicit regression of μ_M rather degrades the performances. MDR is also used to regress the scale feature σ_M explicitly as well, but scaling by σ_M is less reliable than shifting by μ_M. Hence, it degrades the performances in the last row in Table S3.

Table S3: Depth mean errors on the NYUv2 dataset with and without MDR and regression of μ_M.

MDR	μ_M	RMSE	REL	log 10	δ_1	δ_2	δ_3	Mean error
✓	-	0.381	0.108	0.046	0.894	0.984	**0.997**	0.168
-	✓	0.393	0.114	0.048	0.884	0.982	0.996	0.173
✓	✓	**0.370**	**0.103**	**0.045**	**0.903**	**0.986**	**0.997**	**0.155**

MDR	σ_M	RMSE	REL	log 10	δ_1	δ_2	δ_3	Mean error
✓	✓	0.385	0.112	0.047	0.889	0.982	**0.997**	0.169

Next, we define the depth distribution error as

$$|\sigma(\hat{M}) - \sigma(M)| \tag{S2}$$

where $\sigma(\cdot)$ denote the standard deviation of valid pixel depths in a depth map. Table S4 compares the depth distribution errors of MDR* and MDR, which denote the proposed algorithm with μ_M deactivated and activated, respectively, as in Table 5 in the main paper. Again, MDR performs better than MDR*.

Table S4: Depth distribution errors on the NYUv2 dataset.

MDR*	MDR
0.144	**0.134**

Fig. S3 shows qualitative results, together with the standard deviations σ of the corresponding depth maps. We see that, by predicting μ_M through the MDR block, we can estimate the overall depth scales more reliably.

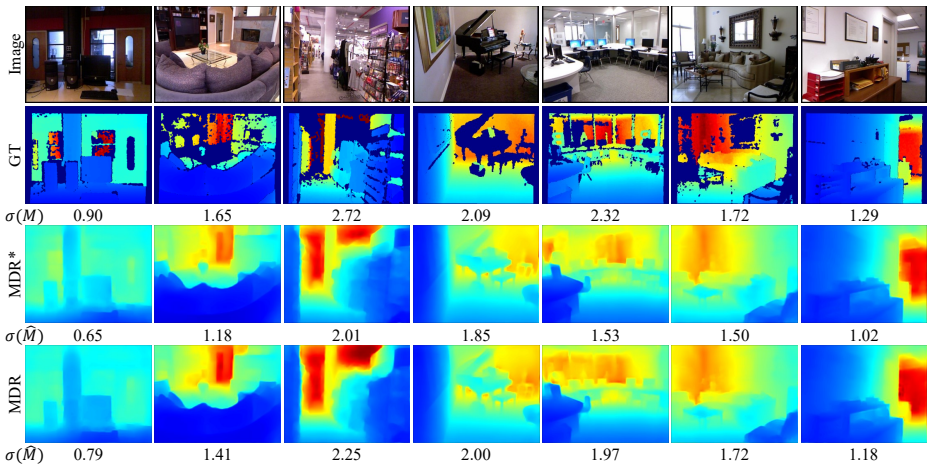

Fig. S3: Qualitative comparison of metric depth maps, estimated by MDR* and MDR, respectively.

S5. Efficacy of G-Net

To demonstrate the efficacy of G-Net, we train the proposed algorithm using N-Net and M-Net only, excluding G-Net. This combination is denoted by M+N, while the default combination by M+N+G. Note that M+N and M+N+G in Table S5 are identical to the second and third rows of Table 3 in the main paper, respectively. In the green boxes in Fig. S4, M+N+G provides sharper edges and more consistent depths on planar regions than M+N does, indicating G-Net helps to improve the performances. Fig. S5 compares metric depth maps. Note that M+N+G reduces estimation errors.

Table S5: Depth estimation results on NYUv2 according to the use of G-Net.

Setting	RMSE	REL	log 10	δ_1	δ_2	δ_3	Kendall's τ	WHDR(%)
M+N	0.389	0.111	**0.047**	**0.888**	**0.982**	0.996	0.814	14.19
M+N+G	**0.387**	**0.109**	**0.047**	**0.888**	**0.982**	**0.997**	**0.817**	**14.01**

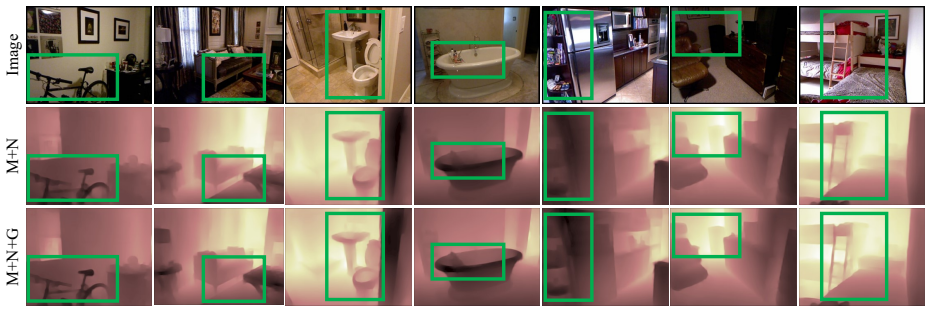

Fig. S4: Normalized depth maps of the two combinations M+N and M+N+G. The differences can be more easily observed within the green boxes.

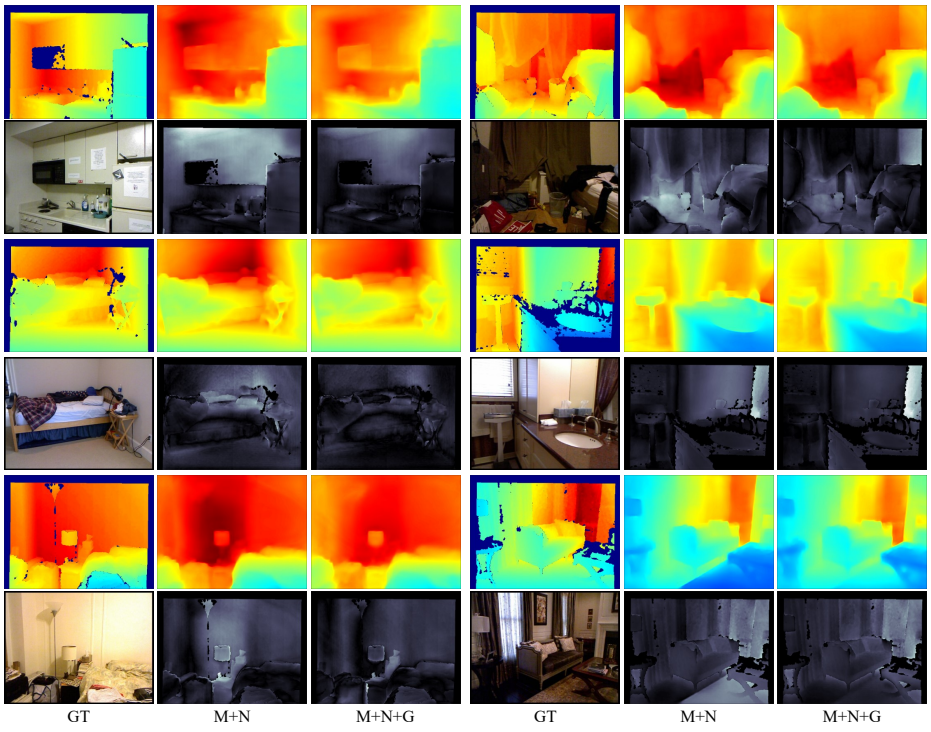

Fig. S5: Metric depth maps of M+N and M+N+G. The error maps are also provided, in which brighter pixels correspond to larger errors.

S6. More Qualitative Results

Fig. S6 and Fig. S7 compare the proposed algorithm with the conventional algorithms [1, 42] on the NYUv2 dataset.

Fig. S8 and Fig. S9 compare the proposed algorithm with the baseline network on the NYUv2 dataset, when the 795 NYU training images and the DIML-Indoor [23] dataset are used for the training, respectively.

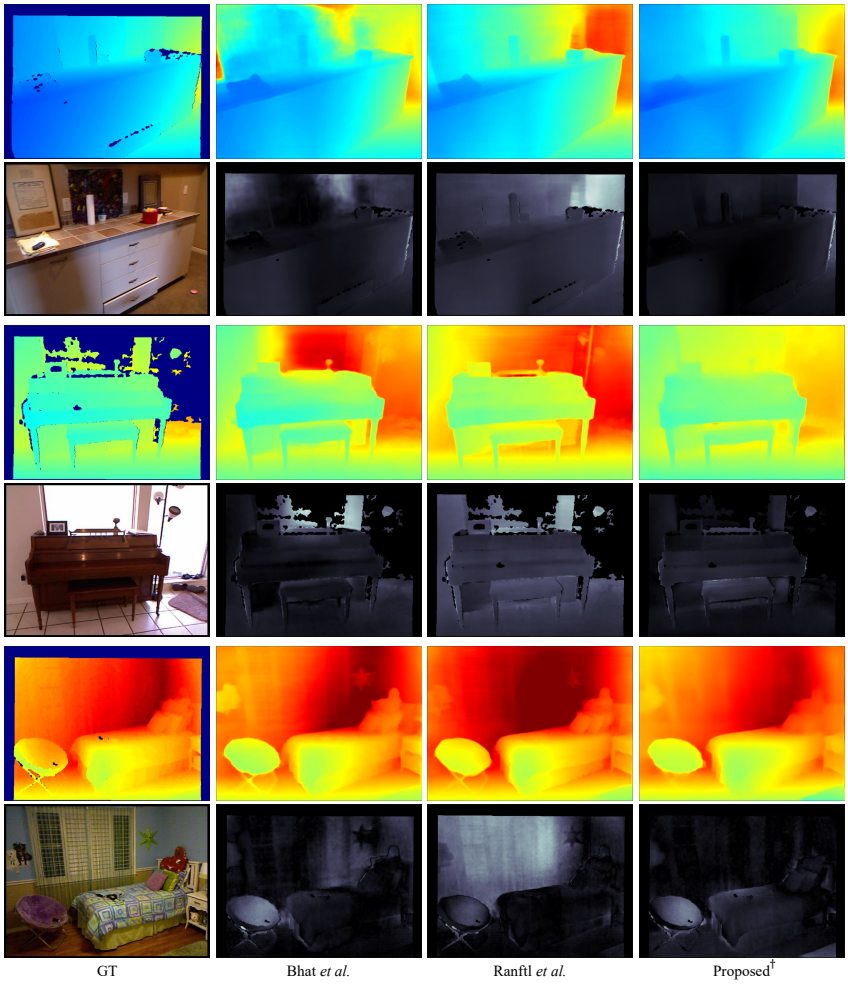

Fig. S6: Qualitative comparison of the proposed algorithm with Bhat *et al.* [1] and Ranftl *et al.* [42] on the NYUv2 dataset. The error maps are also provided, in which brighter pixels correspond to larger errors.

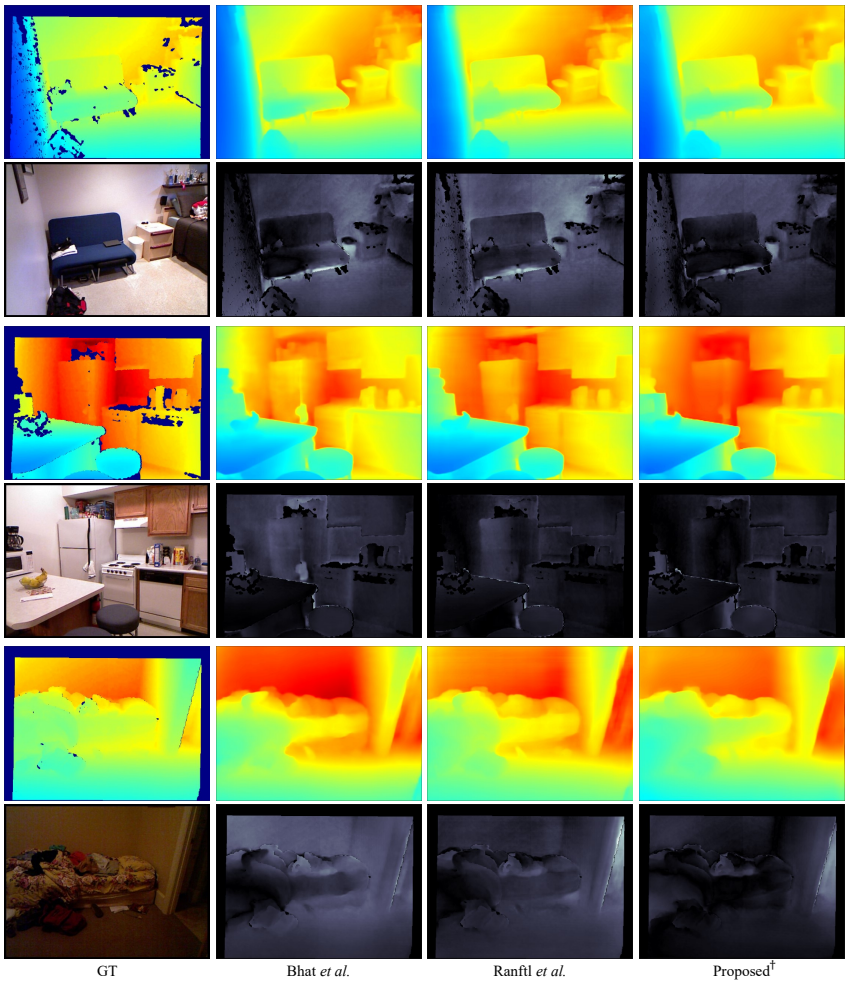

Fig. S7: Qualitative comparison of the proposed algorithm with Bhat *et al.* [1] and Ranftl *et al.* [42] on the NYUv2 dataset. The error maps are also provided, in which brighter pixels correspond to larger errors.

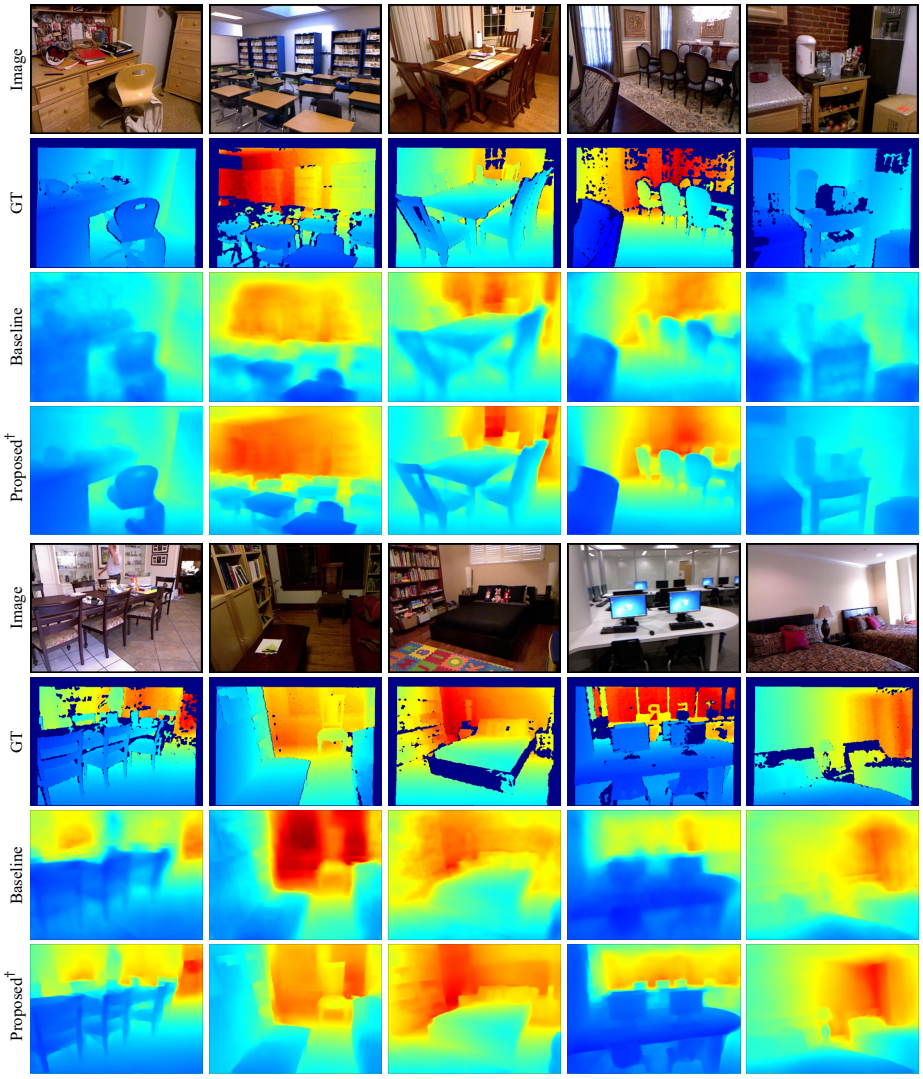

Fig. S8: Qualitative comparison of the proposed algorithm with the baseline using the 795 NYUv2 training images.

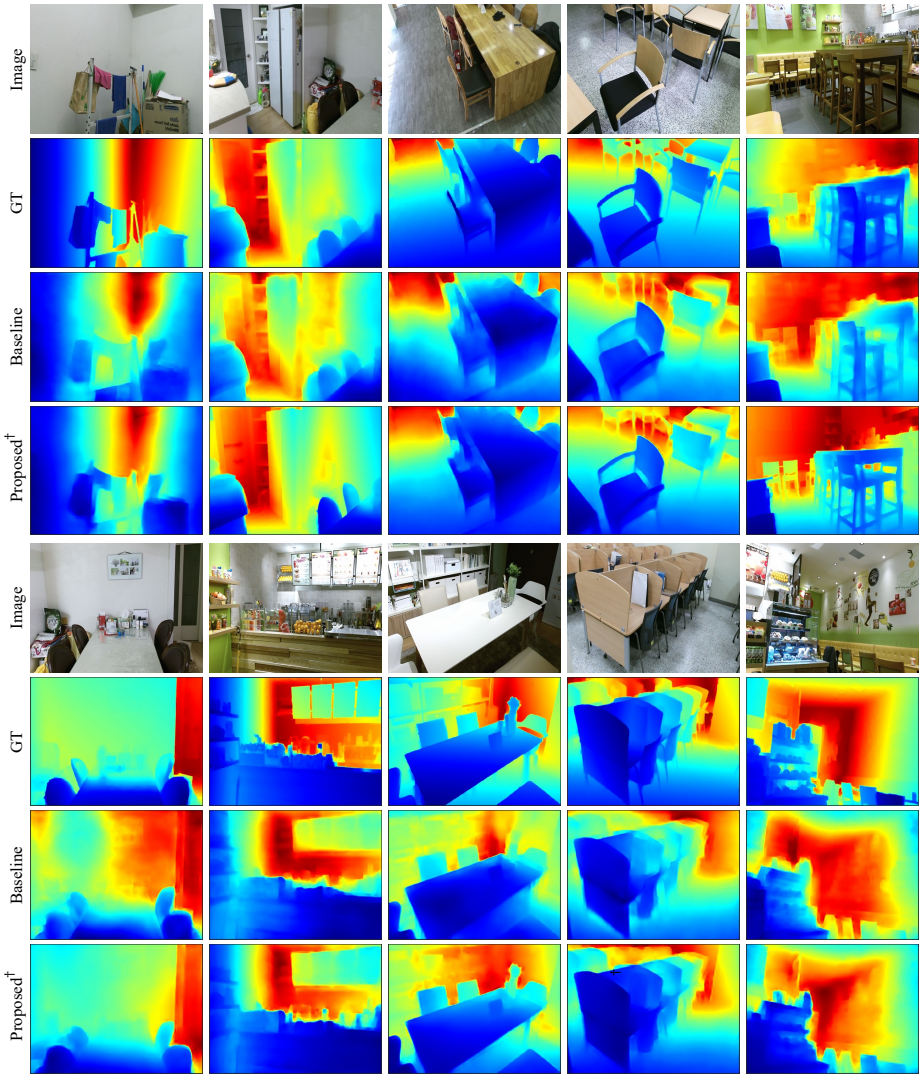

Fig. S9: Qualitative comparison of the proposed algorithm with the baseline using the DIML-Indoor dataset.