LightedDepth: Video Depth Estimation in light of Limited Inference View Angles === Supplementary Material ===

Shengjie Zhu and Xiaoming Liu
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824
zhusheng@msu.edu, liuxm@cse.msu.edu

1. Additional Implementation Details

We randomly select $N_k = 10,000$ pixels from RAFT [11] predicted flowmap. The recording vector size B is set to be 100. The max scale s_{max} for KITTI and NYUv2 datasets are 3 meter and 1 meter individually. The weight parameter λ between epipolar constraint $h_e(\cdot)$ and $h_c(\cdot)$ is set to 0.3 in NYUv2. The minimum camera translation k_s for KITTI and NYUv2 is 0.1 meter and 0.05 meter separately. In the ScanNet experiment, we take BTS [6] trained on NYUv2 as the monocular-depth initialization.

2. Derivation in Details

2.1. Pixel-wise scale estimation

The projection process in the main paper Eqn. 4 is:

$$d'\mathbf{q} = d'\begin{bmatrix} q^x & q^y & 1 \end{bmatrix}^\intercal = d\mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{p} + s\mathbf{K}\bar{\mathbf{t}}, \quad (1)$$

where $\mathbf{p}$ and $\mathbf{q}$ are homogeneous 2D pixel coordinates at frame $\mathbf{I}_m$ and $\mathbf{I}_n$, given by the optical flow prediction $\mathbf{O}$. $\mathbf{K}$ is a 3×3 camera intrinsic matrix. The pixel-wise scale to be computed is denoted as s. $\mathbf{R}$ and $\bar{\mathbf{t}}$ are rotation matrix and normalized translation vector. Denote $\mathbf{M} = \begin{bmatrix}\mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3\end{bmatrix}^\intercal = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$ and $\begin{bmatrix} x & y & z \end{bmatrix}^\intercal = \mathbf{K}\bar{\mathbf{t}}$. By expanding Eqn. 1:

$$d' \cdot \begin{bmatrix} q^x \\ q^y \\ 1 \end{bmatrix} = d \cdot \begin{bmatrix} \mathbf{m}_1^\intercal \\ \mathbf{m}_2^\intercal \\ \mathbf{m}_3^\intercal \end{bmatrix} \mathbf{p} + s \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (2)$$

As mentioned in the main paper Sec. 3.2 and main paper Fig. 3, the point pair $\mathbf{p}$ and $\mathbf{q}$ is given by the optical flow, not complying with the projection geometry, making depth and scale in horizontal and vertical direction follow different relationships. As a result, we represent the horizontal and vertical depth d^x and d^y by the scale s^* separately as:

$$\begin{cases} d' \cdot q^x &= d^x \cdot \mathbf{m}_1^\intercal\mathbf{p} + s \cdot x \\ d' &= d^x \cdot \mathbf{m}_3^\intercal\mathbf{p} + s \cdot z \end{cases}, \quad \begin{cases} d' \cdot q^y &= d^y \cdot \mathbf{m}_2^\intercal\mathbf{p} + s \cdot y \\ d' &= d^y \cdot \mathbf{m}_3^\intercal\mathbf{p} + s \cdot z \end{cases}. \quad (3)$$

Solving Eqn. 3, we have:

$$d^x(s) = s\frac{x - q^x \cdot z}{q^x\mathbf{m_3}^\intercal\mathbf{p} - \mathbf{m_1}^\intercal\mathbf{p}}, \quad d^y(s) = s\frac{y - q^y \cdot z}{q^y\mathbf{m_3}^\intercal\mathbf{p} - \mathbf{m_2}^\intercal\mathbf{p}}. \quad (4)$$

Then, we compute the pixel-wise scale s by minimizing the quadratic loss between the depthmap $d \in \mathbf{D}$ and scale-induced horizontal & vertical depth d^x & d^y:

$$L(s) = \left(d^x(s) - d\right)^2 + \left(d^y(s) - d\right)^2. \quad (5)$$

The global optimal solution of quadratic loss $L(s^*)$ is achieved when its gradient is zero:

$$\frac{\partial L(s)}{\partial s} = 2\left(d^x(s) - d + d^y(s) - d\right) = 0. \quad (6)$$

By injecting Eqn. 4 into Eqn. 6, we solve s as:

$$s = \frac{d}{\frac{1}{2}\left(\frac{x - q^x \cdot z}{q^x\mathbf{m}_3^\intercal\mathbf{p} - \mathbf{m}_1^\intercal\mathbf{p}} + \frac{y - q^y \cdot z}{q^y\mathbf{m}_3^\intercal\mathbf{p} - \mathbf{m}_2^\intercal\mathbf{p}}\right)}. \quad (7)$$

The $m = \log(s) - \log(d)$ in main paper Eqn. 7 is:

$$\begin{aligned} m &= \log(s) - \log(d) \\ &= -\log\frac{1}{2}\left(\frac{x - q^x \cdot z}{q^x\mathbf{m}_3^\intercal\mathbf{p} - \mathbf{m}_1^\intercal\mathbf{p}} + \frac{y - q^y \cdot z}{q^y\mathbf{m}_3^\intercal\mathbf{p} - \mathbf{m}_2^\intercal\mathbf{p}}\right) \end{aligned} \quad (8)$$

2.2. Proof of the Scale & Depth Learning Equality

Theorem. *Considering the optimal scale s^* as the average of pixel-wise scale s in log space, the learning loss of the optical scale L_{s^*} is **upper-bounded** by the learning loss of the video depth L_d and a noise term contributed by the normalized pose $\overline{\mathbf{P}}$ and optical flow $\mathbf{O}$ estimate. Given a robust normalized pose and optical flow estimate, scale learning grounds down to video depth learning.*

Proof. Define the optimal scale s^* as the average of pixel-wise scale s in log space:

$$\log(s^*) = \frac{1}{n}\sum_{i=1}^{n}\log(s_i) = \frac{1}{n}\sum_{i=1}^{n}\left(\log(d_i) + m_i\right). \quad (9)$$

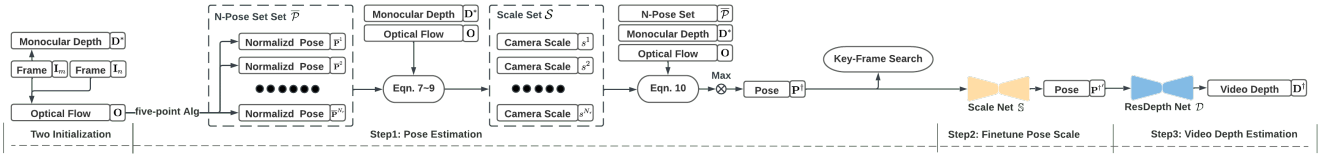


Figure 1. We illustrate the empirical validation experiment of the learning equality between camera scale and video depth. In specific, we extend the original framework in the main paper Fig. 2 with an additional "ScaleNet" $\mathbb{S}$ before the video depth estimation, which further finetunes the camera scale s^+ of the optimized pose $\mathbf{P}^\dagger$ from Sec. 3.2. Row 3 in the main paper Tab. 5 records the pose and depth performance from this modified framework. The "ScaleNet" $\mathbb{S}$ adopts same architecture with "ResDepth Net" $\mathbb{D}$.

where m_i is given in Eqn. 8. Choose a robust l_1 loss for s^*:

$$
\begin{aligned}
L_{s^*} &= \| \log(\tilde{s}) - \log(s^*) \| = \| \log(\tilde{s}) - \frac{1}{n}\sum_{i=1}^{n} \log(s_i) \| \\
&= \| \frac{1}{n}\sum_{i=1}^{n} (\log(\tilde{d}_i) + \tilde{m}_i) - \frac{1}{n}\sum_{i=1}^{n} (\log(d_i) + m_i) \| \\
&\leq \| \frac{1}{n}\sum_{i=1}^{n} \Big(\log(\tilde{d}_i) - \log(d_i) \Big) \| + \| \frac{1}{n}\sum_{i=1}^{n} (\tilde{m}_i - m_i) \| \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \| \log(\tilde{d}_i) - \log(d_i) \| + \| \frac{1}{n}\sum_{i=1}^{n} (\tilde{m}_i - m_i) \| \\
&= L_d + \varepsilon(\overline{\mathbf{P}}, \mathbf{O}).
\end{aligned}
\tag{10}
$$

Here, $\tilde{s}$ and $\tilde{m}_i$ is given by groundtruth labels. L_d is a robust l_1 video depth loss. And $\varepsilon(\overline{\mathbf{P}}, \mathbf{O})$ is an error term that only depends on predicted normalized pose $\overline{\mathbf{P}}$, and flow $\mathbf{O}$. □

Learning scale, as minimizing loss L_{s^*}, can be achieved through minimizing its upper bound loss L_d via learning video depth. Except for the empirical experiment on the main paper Tab. 5, we were not able to measure how large the error term $\varepsilon(\overline{\mathbf{P}}, \mathbf{O})$ is due to the lack of a dataset providing pose and flow groundtruth simultaneously.

One may question whether our definition of the optimal scale s^* is suitable to depict the learning process of a deep scale estimator. Here, we emphasize that most prior works [5, 7, 8, 15] which estimate pose by regression adopt a Fully Convolutional Network, *e.g.* Monodepth2's [5] pose estimator predicts the pose as the average of a 2×2 pose map under a $320 \times 1,024$ input image resolution. Following this, we define a generalized form of wide-adopted pose estimation convention, suitable for the analysis.

3. Additional Ablations

3.1. Compare Projection Flow and Optical Flow

We conduct an empirical experiment to validate our motivation introduced in the main paper introduction section. Specifically, we compare the performance between 3D projection flow and 2D optical flow. The former is computed by the optimized camera pose and depthmap, while the latter is from regression. In Tab. 1, the 2D optical flow (RAFT [11]) demonstrates a clear advantage over the projection flow (DeepV2D and Ours), empirically supporting our motivation. To exclude the influence of moving foreground objects, we additionally report their performance after excluding pedestrians, cyclists, and cars. The semantics label is provided by KITTI15 Semantic Dataset [4]. Metric "epe" and "a1" [11] stand for "endpoint error" and "outlier percentage". Both the lower, the better.

Mehod	All		Background	
	F1-epe	F1-a1	F1-epe	F1-a1
RAFT [11]	**1.284**	**4.539**	**1.238**	**4.759**
DeepV2D [10]	9.957	22.610	2.180	9.789
Ours	**9.321**	**20.723**	**1.631**	**7.692**

Table 1. **Flow Performance Comparison on KITTI FLOW15 Dataset [4].** RAFT [11] computes flow via regression while DeepV2D and our method compute flow via combining predicted depthmap and pose. To facilitate comparison with depth methods, we adopt Garg crop [3]. [Key: **Best**, **Second Best**, All: evaluated over the entire image, Background: evaluated only on background semantic categories.]

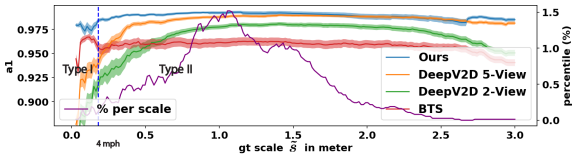
3.2. Scale & Depth Learning Equality

We detail the empirical ablation experiment of the scale and video depth learning equality in Fig. 1. The Fig. 1 and the main paper Fig. 2 correspond to the framework adopted in main paper Tab. 5 row 3 & 2. In Fig. 1, we augment the optimized scale $s^\dagger$ with an additional ScaleNet $\mathbb{S}$ with identical architecture to ResDepth Net (video depth network) $\mathbb{D}$.

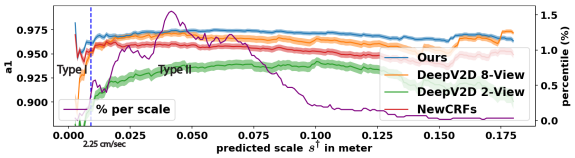
3.3. Depth Performance *w.r.t* Camera Scale

In Fig. 2, we compare video depth performance among ours, DeepV2D's, and our mono-depth initialization, along different camera scales.

As described in the main paper Sec. 3.1, prior works [10, 13] only adopt a cost volume based decoder designed for multi-view stereo to learn the depth prior. We consider such a decoder is insufficient for depth learning. In the left-end of both subfigures (a) and (b), *i.e.*, Type I area, the camera translation movement is insufficient. Due to a near static camera movement, the video depth estimation de-generates



(a) Metric $a1$ *w.r.t.* camera scale on KITTI dataset



(b) Metric $a1$ *w.r.t.* camera scale on NYUv2 dataset

Figure 2. Subfigures (a) and (b) plot the error curves and standard deviations of $a1$ metric ($\delta < 1.25$) against consecutive-frame camera scale on KITTI and NYUv2. Both subfigures report the performance after applying the median scaling. Since NYUv2 does not provide groundtruth poses, we use predicted scale $s^{\dagger}$ as a replacement. To help analysis, we divide the plot into camera translation insufficient cases (Type I) and sufficient cases (Type II). We mark the camera translation speed threshold between Type I and II below the blue dashed line. The right axes (in purple) plots the percentage of the total frame number *w.r.t.* camera scale.

to a monocular depth estimation. In this case, the monocular work significantly outperforms the multi-view video work, indicating an insufficient depth prior knowledge learning.

We consider the monocular depth network and video depth network learning different types of depth prior, with the latter focusing on prior learning in the presence of limited image correspondence. As proof, our method still outperforms multi-view DeepV2D when the camera movement is sufficient. Meanwhile, even around near-static frames, our method still outperforms mono-depth, due to the proposed key-frame search strategy.

3.4. Indoor and Outdoor Camera Trajectory

In the main paper Fig. 5, we colorize the odometry using a measurement m ranging between 0 and 1. The measurement decibels the relative contribution from the rotation and translation movement to an optical flow. Suppose the camera movement $\mathbf{P} = \begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$ is decomposed into a pure rotation movement $\mathbf{P}_r = \begin{bmatrix}\mathbf{R} & \mathbf{0}\end{bmatrix}$ and a pure translation movement $\mathbf{P}_t = \begin{bmatrix}\mathbf{I} & \mathbf{R}^{\intercal}\mathbf{t}\end{bmatrix}$ where $\mathbf{P} = \mathbf{P}_t\mathbf{P}_r$. Given a pixel $\mathbf{p}_i$ at frame $\mathbf{I}_m$, each optical flow $\mathbf{o}_i$ can be decomposed into a rotation concluded flow $\mathbf{o}_i^r = f(\mathbf{P}_r, \mathbf{p}_i) - \mathbf{p}_i$ and a translation concluded flow $\mathbf{o}_i^t = \mathbf{o}_i - \mathbf{o}_i^r$. Function $f(\cdot)$ is the camera projection function, forming a homography under a pure rotation movement $\mathbf{P}_r$, becoming independent of depth. Our measurement is defined as:

$$\text{measurement} = \frac{\sum_i^N \left(\|\mathbf{o}_i^r\|^2 > \|\mathbf{o}_i^t\|^2\right)}{N} \tag{11}$$

The total pixel number is N. As the measurement m approaches 1, the camera rotation contributes most of the scene motion, in against an epipolar constraint based pose estimation algorithm.

4. Additional Quantitative Comparisons

Evaluate Video Depth on EigenSfM split. SfMR [12] suggests eigen split [1] includes near-static images, which is not suitable for SfM methods. As a result, they additionally evaluate on a subset of eigen split whose camera scale is larger than 0.5 meter, *i.e.*, gt scale $\tilde{s} > 0.5$ in Fig 2 (a), around a speed > 10 mph. Still, our method outperforms SfMR [12] with a clear margin, as in Tab. 2.

Evaluation on Odometry Sequence 09 and 10. We further compare with [12] on odometry sequences 09 and 10. Note, due to car translation movement in sequence 09 and 10 is more evident and steady than eigen split, our method achieves a decent performance of 0.994 on a1 metric. The result is shown in Tab. 2.

Performance at Noisy Flow and Mono-Depth. In Tab. 3, we report depth and pose performance using a lower-performed flow model selected from one of the baselines in RAFT (Tab. 1 entry C+T). We use it to analyze performance under noisy inputs. From Tab. 3, we see flow performance affects pose, which in turn lowers depth performance. However, it can be alleviated with a filtering strategy. Even with noisy flow, we still maintain the SoTA depth performance. We also ablate performance under noisy mono-depth input in the main paper Tab. 1, where MonoDepth2 is a noisy lightweight monocular estimator.

Qualitative Comparisons. Please refer to Figs. 3 & 4 & 5 & 6 for more visual comparisons.

Limitations our method favors sufficient translation, more suitable for outdoor environments. Meanwhile, performance can be improved if equipped with an additional bundle-adjustment module from multi-view methods.

References

[1] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 3

[2] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 4

[3] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 2

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 2

[5] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 2, 4

Method	Frame	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
DORN [2]	1	0.067	0.295	2.929	0.108	0.949	0.988	0.995
BTS [6]	1	0.055	0.234	2.859	0.091	0.963	0.993	0.998
TwoView [12]	2	0.034	**0.103**	1.919	**0.057**	**0.989**	**0.998**	**0.999**
DeepV2D [10]	2	0.050	0.212	2.483	0.089	0.973	0.992	0.997
	5	**0.028**	0.118	**1.731**	0.062	**0.989**	0.996	0.998
EvidentDepth (Ours)	2	**0.022**	**0.065**	**1.543**	**0.043**	**0.994**	**0.999**	**1.000**
Monodepth2-sup [5]	1	0.094	0.484	3.614	0.148	0.893	0.975	0.994
BTS [6]	1	0.079	0.297	2.760	0.114	0.932	0.989	0.997
TwoView [12]	2	**0.044**	**0.162**	**2.216**	**0.079**	**0.977**	**0.995**	**0.998**
EvidentDepth (Ours)	2	**0.024**	**0.055**	**1.324**	**0.043**	**0.994**	**0.999**	**1.000**

Table 2. **KITTI Monocular Video Depth Evaluation.** Upper table evaluates on EigenSfM split [12]. The lower table evaluates on odometry sequence 09 & 10. We evaluate under Garg crop capped at 80 meters with semi-dense groundtruth after applying median scaling. [Key: **Best**, **Second Best**, Frame: the number of consecutive frames required in inference.]

Depth	Flow	Filter	KITTI Depth					KITTI Flow15		Seq 09		Seq 10	
			AbsRel	SqRel	RMSE	RMSE log	$\delta < 1.25$	F1-epe	F1-all	r_{err}	t_{err}	r_{err}	t_{err}
BTS [6]	PWC [9]	x	0.039	0.123	1.999	0.066	0.981	10.35	33.7	1.82	0.46	1.58	0.53
		✓	0.031	0.109	1.896	0.058	0.986			1.32	0.30	1.40	0.39
	RAFT [11]	x	0.028	0.093	1.695	0.052	0.990	5.04	17.4	1.08	0.26	1.29	0.36

Table 3. **Depth and Pose Performance with sub-optimal flow model.** We ablate depth and pose performance under lower performed flow model PWC. Flow metrics are defined in RAFT. Filter uses a flow outlier removal strategy introduced in [14].

[6] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 1, 4, 5, 6

[7] Sudeep Pillai, Rareş Ambruş, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In *ICRA*, 2019. 2

[8] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *CVPR*, 2019. 2

[9] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *ECCV*, 2018. 4

[10] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *ICLR*, 2020. 2, 4

[11] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1, 2, 4

[12] Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, Kaihao Zhang, Nikolai Smolyanskiy, and Hongdong Li. Deep two-view structure-from-motion revisited. *CVPR*, 2021. 3, 4

[13] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *ECCV*, 2020. 2

[14] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, 2020. 4

[15] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018. 2

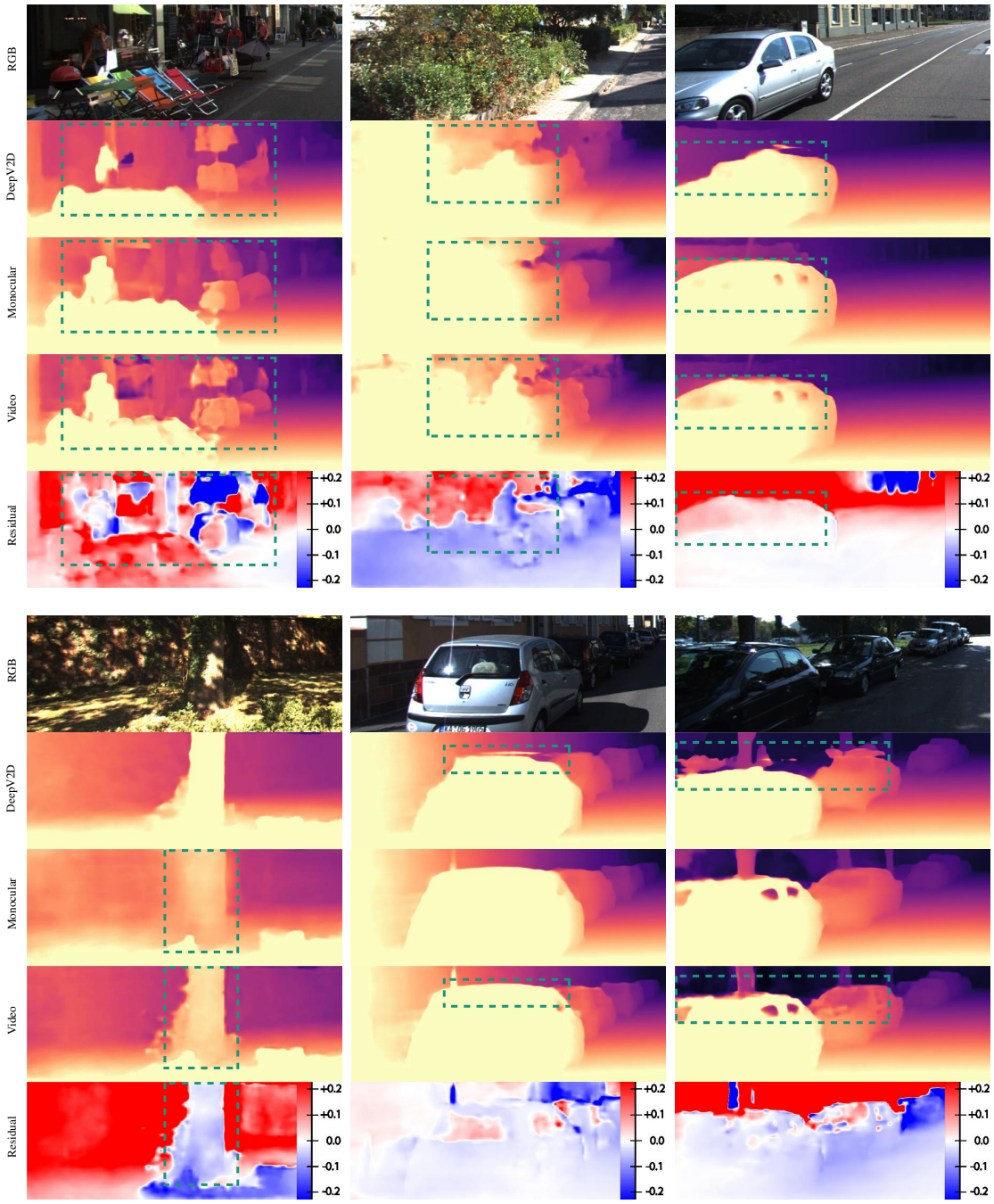


Figure 3. Residual depth is measured *in meters* as $\mathbf{D}^* \cdot (\exp(\Delta\mathbf{D}) - 1)$. We use Green boxes to highlight our improvement over 5-view DeepV2D and monocular depth input BTS [6]. We avoid artifacts in DeepV2D arising from moving foreground objects and near-static camera movement due to our formulation of video depth estimation as log space residual estimation.

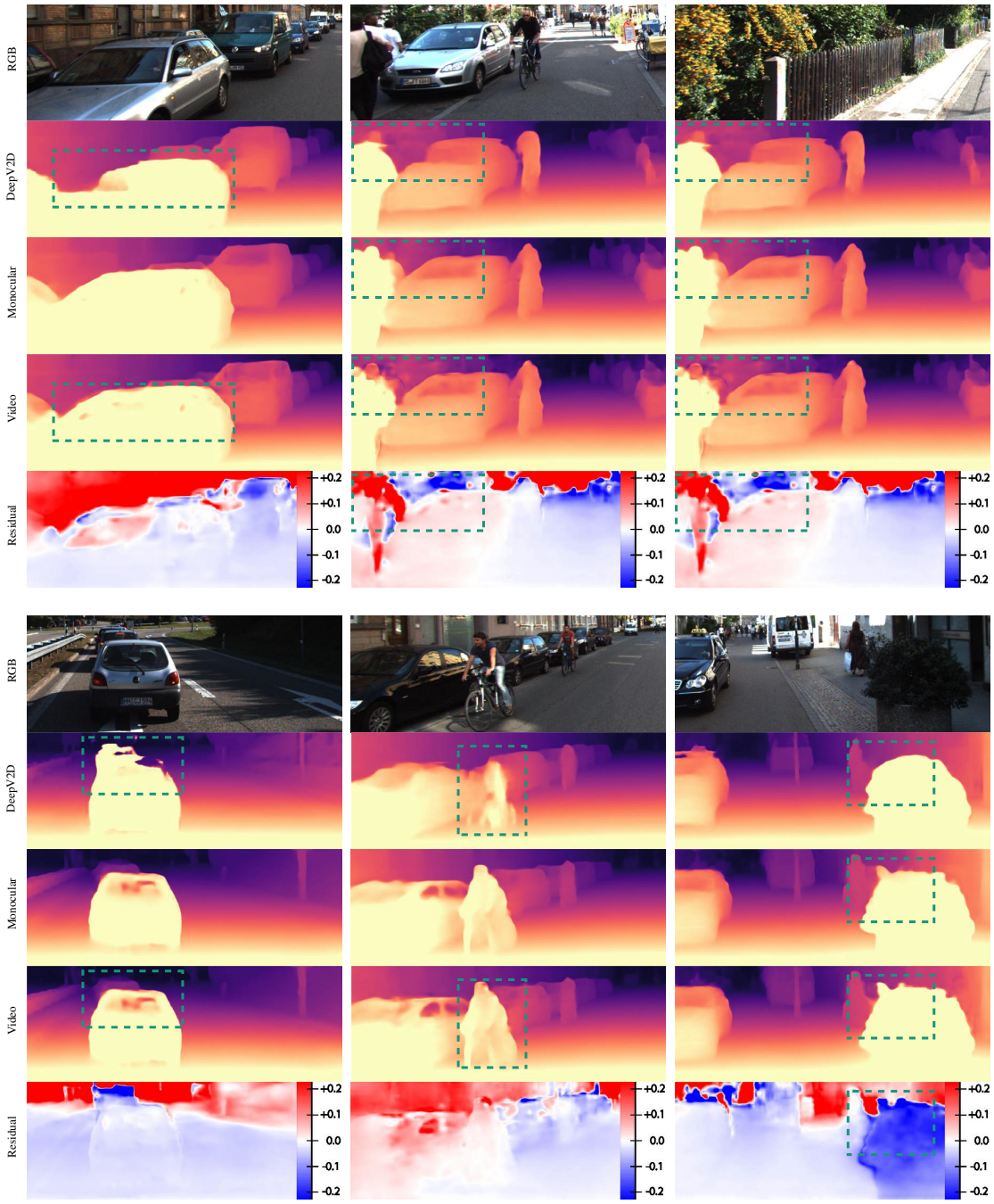


Figure 4. Continued with Fig. 3. Residual depth is measured in meters as $\mathbf{D}^* \cdot (\exp(\Delta\mathbf{D}) - 1)$. We use Green boxes to highlight our improvement over 5-view DeepV2D and monocular depth input BTS [6]. We avoid artifacts in DeepV2D arising from moving foreground objects and near-static camera movement due to our formulation of video depth estimation as log space residual estimation.

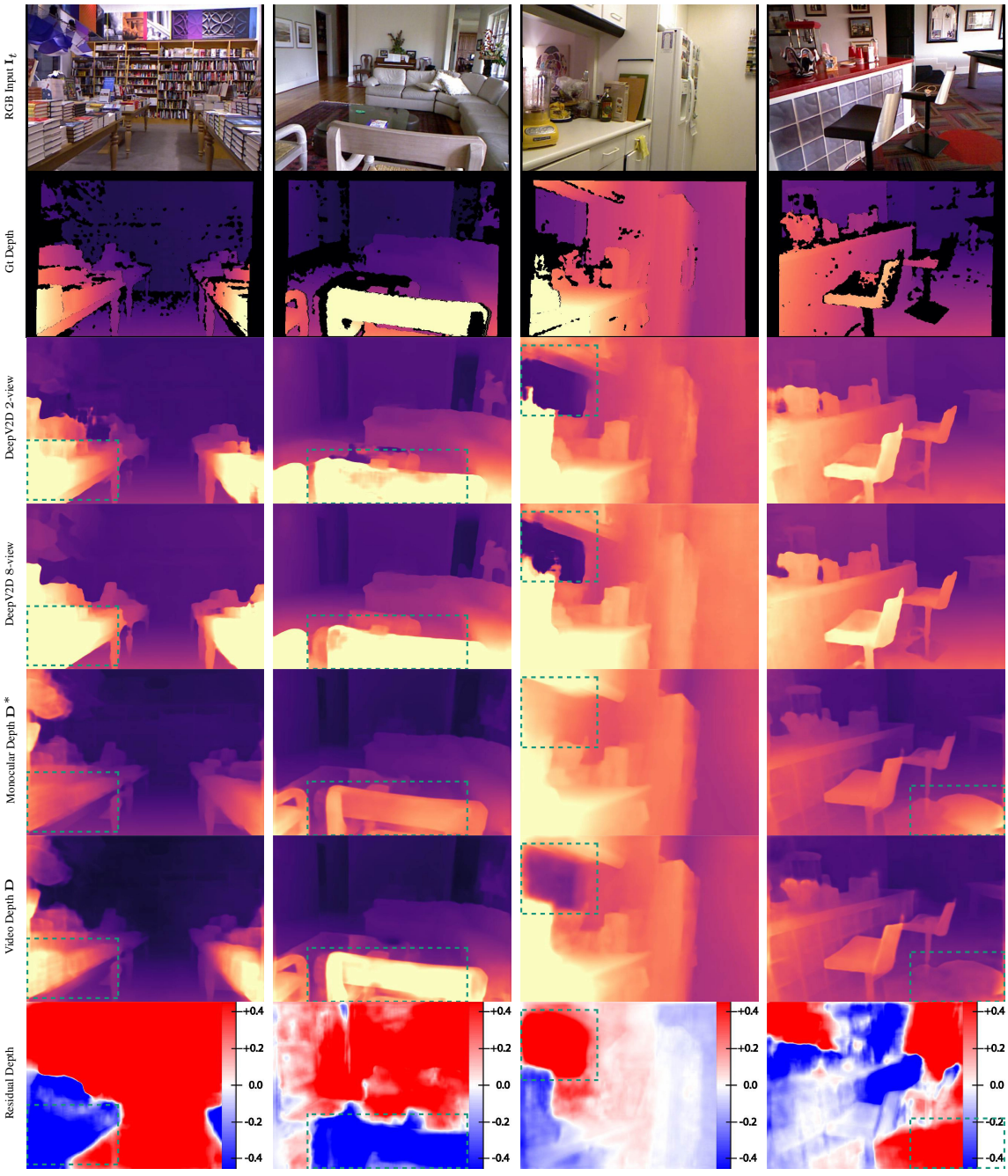


Figure 5. Residual depth is measured in meters as $\mathbf{D}^* \cdot (\exp(\Delta\mathbf{D}) - 1)$. We use Green boxes to highlight our improvement over 8-view DeepV2D, 2-view DeepV2D and monocular depth initialization.

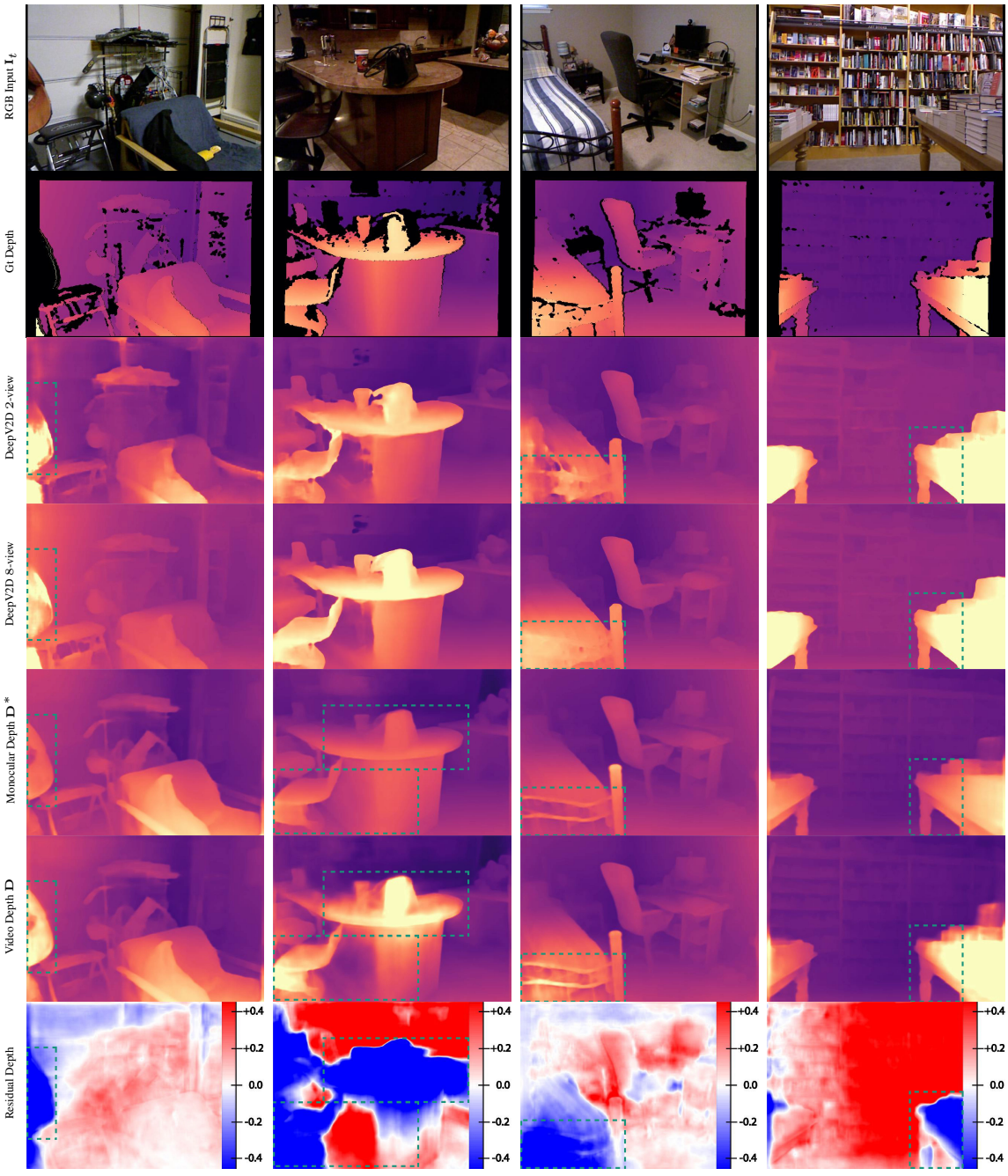


Figure 6. Continued with Fig. 5. Residual depth is measured in meters as $\mathbf{D}^* \cdot (\exp(\Delta\mathbf{D}) - 1)$. We use Green boxes to highlight our improvement over 8-view DeepV2D, 2-view DeepV2D and monocular depth initialization.