

Zero-Shot Monocular Scene Flow Estimation in the Wild

Supplementary Material

In this supplementary material, we present additional technical details and evaluations pertaining to our model architecture, dataset, and methodology. Please check our supplementary video for a high-level overview and abundant qualitative results with comparisons.

A. Additional Experimental Setting

Datasets. For evaluation on the Spring Dataset, we use a subset of the training set {0005, 0009, 0013, 0017, 0023, 0037, 0044}. For VKITTI2, we designate Scene18 as the test sequence due to its overlap with KITTI Scene Flow training data, preventing data contamination. VKITTI2 is created to reproduce several scenes of KITTI, thus we consider methods trained on VKITTI2 to be in-domain with KITTI experiments.

Implementation Details. All experiments are trained with 8 NVIDIA A100 GPUs for 50 epochs, which take 12 hours. We accumulate gradients every 2 steps, and clip gradients by norm value 0.5. We have 8 DDP processes with a batch size of 4, thus our effective batch size is $4 \times 8 \times 2 = 64$.

We select a learning rate of $1e^{-6}$ for the encoder-decoder, $1e^{-6}$ for the pointmap heads, and $1e^{-4}$ for the offset heads. For the first 8 epochs, all learning rates start from 0.1 of above values, and gradually warm up. Note that, differently from DUST3R [8] or MAST3R [4], our heads do not predict confidence maps as we found empirically that they made our training unstable.

Training Datasets. At each epoch, we re-sample $1e^4$ examples from the whole Data Recipe for training by balancing different datasets considering diversity and statistics (Tab. 2). 10000 samples for each epoch is made of: (SHIFT:2024, DynamicReplica:1942, VKITTI2:958, MOVIF:3560, PointOdyssey:1400, Spring:116). We resize and crop all dataset samples into 288×512 resolution. For stereo cameras, we ignore right camera frames.

Resolution/Scale Alignment. During evaluation, we resize inputs to be width-512 and resize outputs later. If scale alignment is required, we compute median scale between the predicted and ground-truth pointmap following DUST3R [8], and scale both the pointmap and scene flow.

Baselines. DUST3R/MASt3R predict a pointmap for the second frame in the first frame’s coordinate system. However, for scene flow, we need geometry for the second frame in the second frame’s coordinate system. As such, for the Spring and VKITTI2 datasets, we use the ground truth camera pose to transform the pointmaps into the second frame’s

coordinate system such that we can create depth maps for the second frame. For KITTI, we use DUST3R’s [8] pose estimation method to infer camera pose.

Resolution/Scale Alignment. For fairness, most methods resize inputs to be width-512 during inference. Self-Mono-SF requires 256×832 , while OpticalExpansion [9] requires 384×1280 . For DUST3R / MAST3R, we multiply pointmaps by estimating scaling factor. For Self-Mono-SF and DepthAnythingV2-metric, the scale factor is computed for each depth estimation/groundtruth pair.

Evaluation Metrics.

EPE: $\|\widehat{sf} - sf\|_2$ averaged over each pixel, where, \widehat{sf} and sf denote the estimated and ground truth scene flows respectively.

AccS: Percentage of points where $EPE < 0.05$ or relative error $< 5\%$.

AccR: Percentage of points where $EPE < 0.1$ or relative error $< 10\%$.

Out: Percentage of points where $EPE > 0.3$ or relative error $> 10\%$.

AbsRel: Absolute relative error $|d^* - d|/d$.

δ_1 : Percentage of $\max(d^*/d, d/d^*) < 1.25$.

B. Additional experiments

B.1. Does Joint Estimation Help?

Key to our method’s success is our design that forces the model to predict geometry and scene flow jointly. To validate this statement we present an ablation study in Tab. 1, which shows different pretraining strategies.

Our analysis allows us to make a few observations. First, we compare the performance of our method when we train the offset head from scratch, and when we initialize it with either DUST3R or MAST3R’s pretrained model. Initializing the offset head results in significantly lower (better) EPE and AbsR-r numbers in Tab. 1. From them, the importance of high-quality 3D priors for the offset head is therefore clear, which also confirms the entanglement of depth and motion predictions. Second, our approach outperforms the combination of MAST3R, a state-of-the-art depth estimation algorithm, and RAFT, a state-of-the-art optical flow algorithm—and the algorithm we use to generate the pseudo ground truth we train on. Our approach combining MAST3R prior with the offset head achieves optimal results across both scene flow (EPE: 0.452, AccR: 0.443) and depth metrics (AbsR-r: 0.111, $\delta_1 - r$: 0.927).

3D Prior	Scene Flow Estimation				Depth Estimation			
	EPE↓	AccS↑	AccR↑	Out↓	AbsR-r↓	δ_1 -r↑	AbsR-m↓	δ_1 -m↑
Ours (scratch)	1.071	0.420	<u>0.442</u>	0.957	0.353	0.433	0.547	0.049
Ours (DUST3R [8])	<u>0.588</u>	0.378	0.408	<u>0.896</u>	0.121	0.887	0.233	0.423
MASt3R [4]	3.708	0.264	0.267	0.999	0.108	0.929	0.245	0.288
Ours (MASt3R [4])	0.452	<u>0.398</u>	0.443	0.873	<u>0.111</u>	<u>0.927</u>	<u>0.236</u>	<u>0.345</u>

Table 1. **Ablation over Joint Estimation Pipelines.** Verify the effect of 3D Prior and Offset Heads on KITTI. In the table ‘Ours (w)’ is our model initialized with the weights of method ‘w’.

Data	Scene Flow Estimation				Depth Estimation			
	EPE↓	AccS↑	AccR↑	Out↓	AbsR-r↓	δ_1 -r↑	AbsR-m↓	δ_1 -m↑
<i>exclude</i>	0.641	0.392	0.431	0.899	0.116	0.922	0.256	0.237
<i>specific</i>	0.569	0.317	0.393	0.911	0.090	0.925	0.580	0.095
<i>all</i>	0.452	0.398	0.443	0.873	0.111	0.927	0.236	0.345

Table 2. **Ablation over Data Recipe** on KITTI. *all*: using all datasets for training. *exclude*: using all datasets, except for train set of VKITTI2. *specific*: only using train set of VKITTI2.

B.2. Data Recipe is Key for Generalization

In Tab. 2, we explore how the datasets used in training affect the performance on KITTI. We compare our method performance with three training strategies: training on all the datasets (all), training on all but VKITTI2 (exclude), and training only on VKITTI2 (specific). Note that none of these strategies train on KITTI. Unsurprisingly, training with all datasets yields the best overall performance for both scene flow and depth. However, when omitting VKITTI2, which is the only driving dataset use, we maintain robust performance (EPE: 0.641, AbsR-m: 0.256), which demonstrates the generalization capabilities gained with diverse data exposure.

B.3. Reference coordinate frame

DUST3R and MASt3R make predictions of pointmaps from pairs of images, where both sets of points are predicted in the coordinate frame of the first camera C_1 . As scene flow estimation does not typically assume camera poses, it is defined as the (field of) vector from a 3D point in the first camera frame C_1 at time t_1 to a 3D point in the second camera frame C_2 at time t_2 . This difference of output reference frames—either always predicting in C_1 or from C_1 into C_2 —raises the question of whether the reference coordinate frame is important for prediction accuracy. This is especially true given that we fine tune the DUST3R network that operates solely in C_1 .

We ran an experiment to change the reference scene flow coordinate frame for our output scene flow: either 1) from C_1 into C_2 , 2) from C_1 to C_1 (sometimes called *rectified scene flow*), or 3) within a world coordinate frame. Scene flow training data labels are defined from C_1 to C_2 as in 1), so to produce the vector from C_1 to C_1 we must reproject the 3D end point of the vector into C_1 ’s reference frame using

Table 3. **Changing reference coordinate frame for the end point of scene flow.** For synthetic data with known camera poses from which to accurately translate coordinate frames for training data and evaluation, there is no clear gain from any of the reference frames (VKITTIv2, Spring). For real-world KITTI data, camera poses may be in error, and we see large increases in scene flow error when attempting to train and evaluate on scene flow data in a different reference frame.

Reference frame	Scene Flow Estimation				Depth Estimation			
	EPE↓	AccS↑	AccR↑	Out↓	AbsR-r↓	δ_1 -r↑	AbsR-m↓	δ_1 -m↑
VKITTI2 Dataset [1]								
Known camera poses								
World	0.154	0.744	0.830	0.511	0.140	0.856	0.222	0.682
Camera 1	0.202	0.808	0.886	0.449	0.139	0.858	0.210	0.711
Camera 2	0.190	0.780	0.876	0.484	0.137	0.859	0.236	0.634
Spring Dataset [6]								
Known camera poses								
World	0.010	0.986	0.998	0.812	0.265	0.600	1.132	0.043
Camera 1	0.012	0.990	1.000	0.811	0.272	0.594	0.716	0.103
Camera 2	0.013	0.989	0.999	0.813	0.280	0.597	0.679	0.119
KITTI [7]								
Estimated camera poses								
World	2.637	0.268	0.271	0.995	0.110	0.928	0.212	0.546
Camera 1	4.202	0.251	0.253	0.992	0.108	0.928	0.214	0.519
Camera 2	0.452	0.398	0.443	0.873	0.111	0.927	0.236	0.345

known ground truth camera poses. When computing metrics like end point error or accuracy against ground truth scene flow from C_1 to C_2 , this transformation is also necessary. For synthetic datasets like Spring and VKITTI2, the transformation is simple as camera poses are known. For VKITTI2, the camera poses must be estimated and may be in error.

Table 3 shows that the choice of coordinate frame is not significantly important when evaluated on Spring and VKITTI2, with any reference frame scoring approximately similarly. This suggests that the pretrained backbone can be successfully fine tuned with scene flow data and our training scheme for any coordinate frame. For VKITTI2, we observe large drops in performance as the reference frame changes. We attribute this to errors in the estimated camera poses, which affect training and metric evaluation.

B.4. Additional quantitative evaluation

We show our method’s superiority with comparison with different checkpoints of previous monocular scene flow methods in Tab. 4.

B.5. Additional Qualitative Results

We show additional qualitative comparisons for KITTI (Fig. 1). Overall, our approach is robust and performs consistently better across different scenes.

References

- [1] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 2, 3
- [2] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 4
- [3] Junhwa Hur and Stefan Roth. Self-supervised multi-frame

KITTI [7] (Real)								
Method	Scene Flow Estimation				Depth Estimation			
	EPE↓	AccS↑	AccR↑	Out↓	AbsR-r↓	δ_1 -r↑	AbsR-m↓	δ_1 -m↑
<i>Ours</i>	0.452	0.398	0.443	0.873	0.111	0.927	0.236	0.345
Self-Mono-SF [2]-kitti-train	0.454	0.345	0.435	0.853	0.100	0.905	0.105	0.879
Self-Mono-SF [2]-kitti-eigen	0.517	0.392	0.457	0.870	0.085	0.929	0.089	0.917
OpticalExpansion [9]-kitti-train	2.703	0.357	0.366	0.970	0.132	0.871	0.529	0.172
OpticalExpansion [9]-kitti-trainval	2.682	0.359	0.367	0.967	0.132	0.871	0.529	0.172
<i>Ours-exclude</i>	0.641	0.392	0.431	0.899	0.116	0.922	0.256	0.237
OpticalExpansion [9]-driving	2.537	0.348	0.363	0.968	0.132	0.871	0.529	0.172
OpticalExpansion [9]-robust	2.702	0.359	0.366	0.969	0.132	0.871	0.529	0.172
VKITTI2 [1] (Real)								
Method	Scene Flow Estimation				Depth Estimation			
	EPE↓	AccS↑	AccR↑	Out↓	AbsR-r↓	δ_1 -r↑	AbsR-m↓	δ_1 -m↑
<i>Ours</i>	0.190	0.780	0.876	0.484	0.137	0.859	0.236	0.634
Self-Mono-SF [2]-kitti-train	0.294	0.694	0.751	0.589	0.244	0.568	0.224	0.589
Self-Mono-SF [2]-kitti-eigen	0.373	0.688	0.738	0.666	0.200	0.668	0.189	0.669
OpticalExpansion [9]-kitti-train	1.879	0.653	0.663	0.947	0.194	0.727	0.203	0.746
OpticalExpansion [9]-kitti-trainval	1.845	0.655	0.664	0.946	0.194	0.727	0.203	0.746
<i>Ours-exclude</i>	0.409	0.727	0.772	0.696	0.144	0.836	0.168	0.778
OpticalExpansion [9]-driving	1.809	0.654	0.662	0.946	0.194	0.727	0.203	0.746
OpticalExpansion [9]-robust	1.915	0.654	0.662	0.958	0.194	0.727	0.203	0.746
Spring [6] (Synthetic)								
Method	Scene Flow Estimation				Depth Estimation			
	EPE↓	AccS↑	AccR↑	Out↓	AbsR-r↓	δ_1 -r↑	AbsR-m↓	δ_1 -m↑
<i>Ours</i>	0.013	0.989	0.999	0.813	0.280	0.597	0.679	0.119
<i>Ours-exclude</i>	0.014	0.992	1.000	0.787	0.294	0.599	0.682	0.130
Self-Mono-SF [2]-kitti-train	1.005	0.251	0.328	0.880	0.501	0.353	0.784	0.044
Self-Mono-SF [2]-kitti-eigen	0.717	0.272	0.365	0.895	0.532	0.344	0.772	0.050
OpticalExpansion [9]-driving	0.029	0.873	0.971	0.805	0.430	0.468	0.804	0.004
OpticalExpansion [9]-kitti-train	0.038	0.849	0.935	0.799	0.430	0.468	0.804	0.004
OpticalExpansion [9]-kitti-trainval	0.027	0.916	0.967	0.800	0.430	0.468	0.804	0.004
OpticalExpansion [9]-robust	0.016	0.969	0.995	0.801	0.430	0.468	0.804	0.004

Table 4. **Comparison with Monocular Scene Flow Methods.**

<i>Self-Mono-SF</i> [2]-kitti-train:	Trained on KITTI, train split;
<i>Self-Mono-SF</i> [2]-kitti-eigen:	Trained on KITTI, eigen split;
<i>OpticalExpansion</i> [9]-driving:	Trained on Driving [5];
<i>OpticalExpansion</i> [9]-kitti-train:	Trained on Driving [5], and then finetune on KITTI, train split;
<i>OpticalExpansion</i> [9]-kitti-trainval:	Trained on Driving [5], and then finetune on KITTI, trainval split;
<i>OpticalExpansion</i> [9]-robust:	Trained for the Robust Vision Challenge.

monocular scene flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[4] Vincent Leroy, Yann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 1, 2, 4

[5] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, op-

tical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[6] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nali-vayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

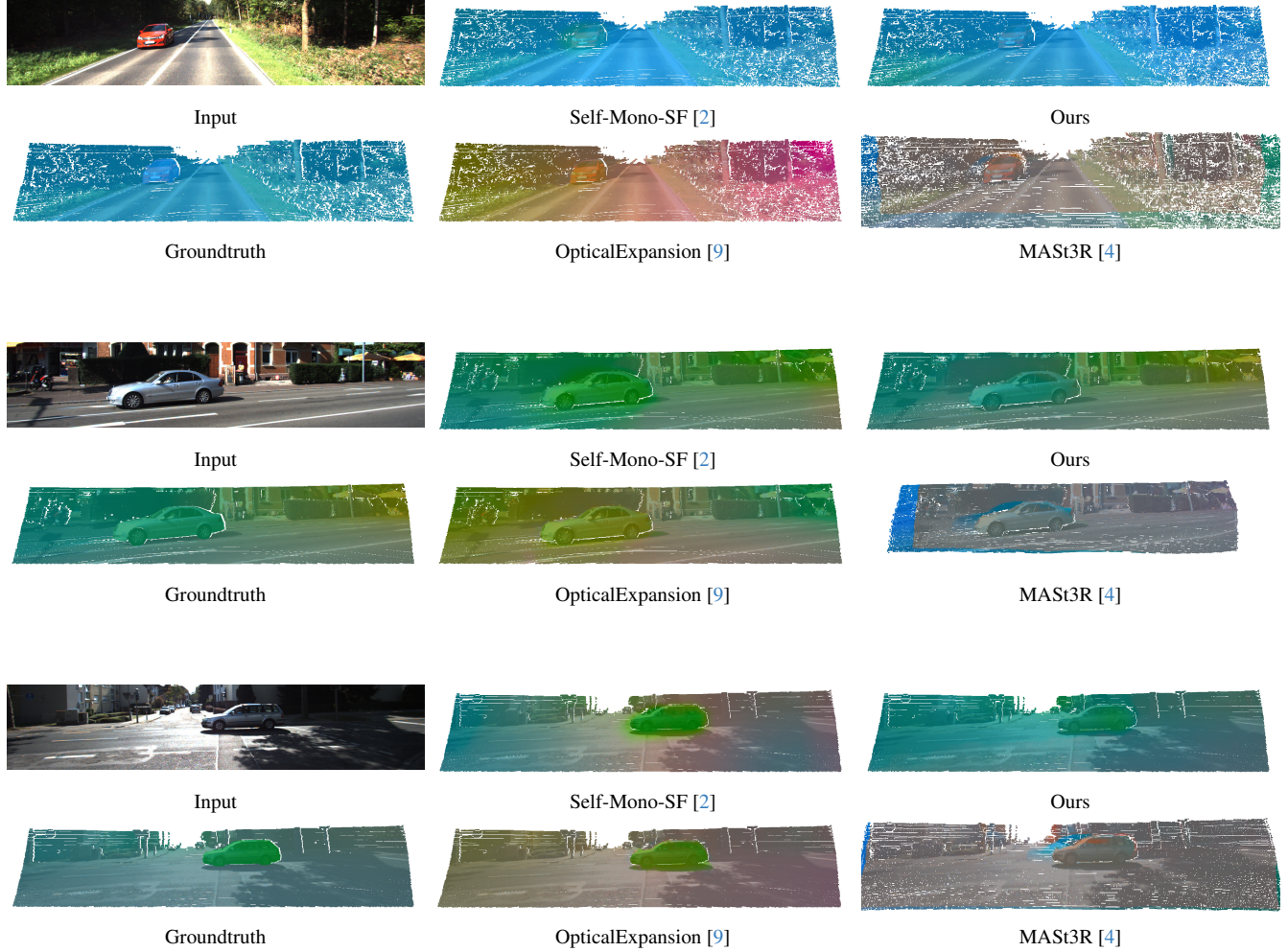


Figure 1. **Additional Qualitative Results on KITTI [7]**, with scene flow represented by CIE-LAB [3] colorwheel overlaid on corresponding 3D structure estimated for first input image. Ours method shows the highest similarity to the groundtruth.

- [7] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3, 4
- [8] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2
- [9] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3, 4