PromptStereo: Zero-Shot Stereo Matching via Structure and Motion Prompts

Supplementary Material

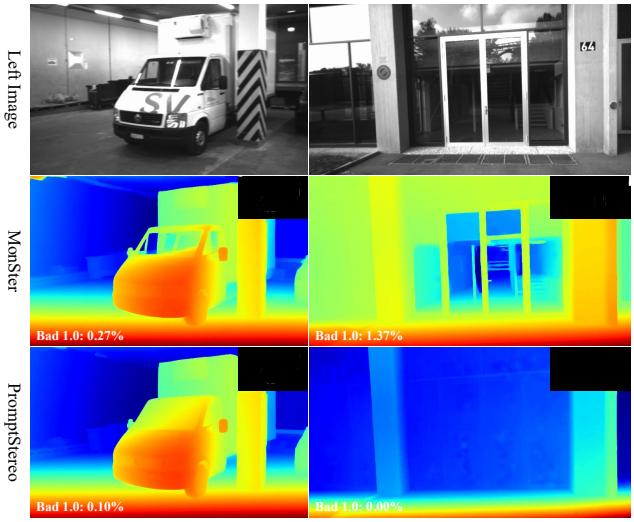

Figure 6. Visualization of ETH3D (unlimited training sets). The Bad 1.0 metric map is in the upper right corner.

Model	KITTI 2012 Bad 3.0	KITTI 2015 Bad 3.0	Midd-T (H) Bad 2.0	Midd-2021 Bad 2.0	ETH3D Bad 1.0
MonSter	**3.27**	3.72	3.75	8.46	1.02
Ours w/o FSD	3.34	3.50	2.41	3.97	**0.71**
Ours w/ FSD	3.33	**3.40**	**2.21**	**2.78**	0.79

Table 7. Ablation study of whether to use FSD [45].

6. Additional Experiment

In this section, we provide additional comparisons with several methods, including implementation details and method-specific training strategies. We compare our PromptStereo with three methods: MonSter [14], FoundationStereo [45], and Stereo Anywhere [2].

6.1. MonSter

In the main text, we evaluate MonSter [14] with a re-trained Scene Flow checkpoint. Although the official repository provides a Scene Flow checkpoint, our evaluation reveals inconsistent results. The public checkpoint shows unrealistically strong performance. After consulting the authors, we have been informed that the public Scene Flow checkpoint is not trained solely on Scene Flow. They uploaded the wrong checkpoint, and they plan to correct it in the future. Therefore, we re-train MonSter using the official code.

As shown in Fig. 6, we further visualize results on ETH3D. Although it does not provide ground truth for glass surfaces, the visualization clearly shows that our PromptStereo correctly infers these regions even on grayscale images. At the same time, MonSter fails to handle such challenging transparent areas.

Besides, we investigate the influence of training sets. Since MonSter's mixed training sets do not include FSD [45], we remove FSD and add TartanAir, following MonSter's configuration. As shown in Tab. 7, excluding FSD leads to only a marginal decrease in accuracy (except ETH3D), and our PromptStereo still surpasses MonSter. This further demonstrates the effectiveness and universality of PromptStereo.

6.2. FoundationStereo

In the main text, we report FoundationStereo's results, but do not include it in our comparison. This is due to two main practical constraints. First, FoundationStereo [45] requires a tremendous training configuration (a batch size of 128 on 32 A100 GPUs), whereas other methods only use a batch size of 8 on 4 GPUs with 24 GB of memory. Second, only the inference code is publicly available; the training code has not been released, making its results difficult to reproduce. For these reasons, we exclude FoundationStereo from direct comparisons in the main text.

However, our PromptStereo still surpasses FoundationStereo on Middlebury 2021 and Booster. As shown in Fig. 7, PromptStereo achieves slightly better accuracy on Middlebury 2021. Interestingly, FoundationStereo produces extreme and unstable disparity estimates along the image borders, which in turn shift the overall color visualization, whereas PromptStereo maintains well-behaved predictions and fine color visualization. Moreover, as shown in Fig. 8, FoundationStereo remains poor performance on reflective and transparent surfaces, whereas PromptStereo can provide reasonable predictions even on large mirror surfaces. This robustness to transparent regions is a key advantage of PromptStereo over FoundationStereo.

6.3. Stereo Anywhere

Stereo Anywhere is also a special case. Its main innovations lie in its normal volume construction, volume truncation, and volume augmentation. Beyond these design aspects, our inspection of the released code shows two additional factors that may affect the fairness of comparison. First, its data augmentation differs from RAFT-Stereo's standard augmentation, which includes more than a dozen transformations (ChannelDropout, ChannelShuffle, MotionBlur, ImageCompression, GaussNoise, etc). Second, Stereo Anywhere is not trained from scratch; instead, it initializes training from a RAFT-Stereo checkpoint. To make the comparison fair, we re-train PromptStereo on Scene Flow from a PromptStereo Scene Flow checkpoint and apply the same data augmentation as Stereo Anywhere.

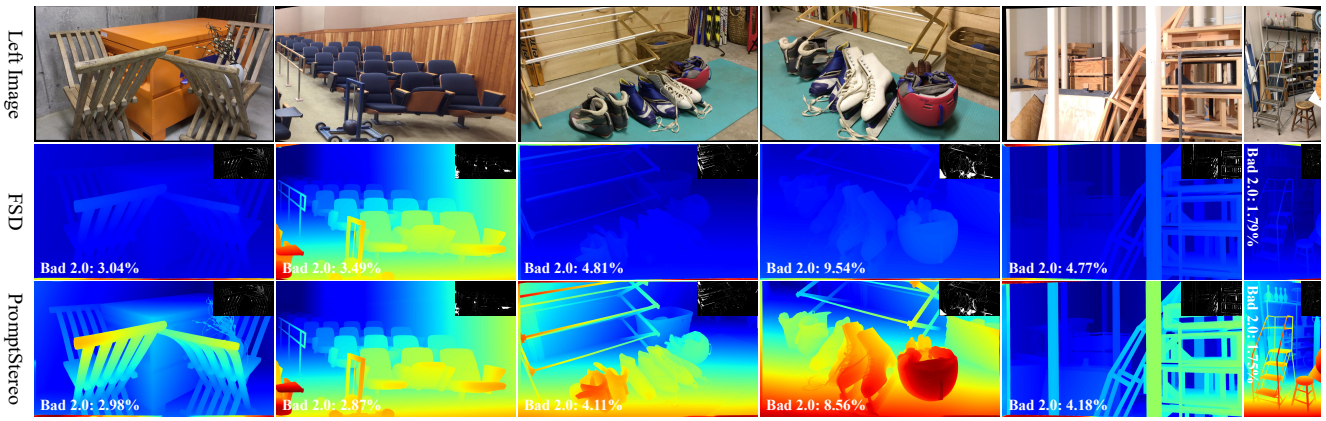

Figure 7. Visualization of Middlebury 2021 (unlimited training sets). The Bad 2.0 metric map is in the upper right corner.

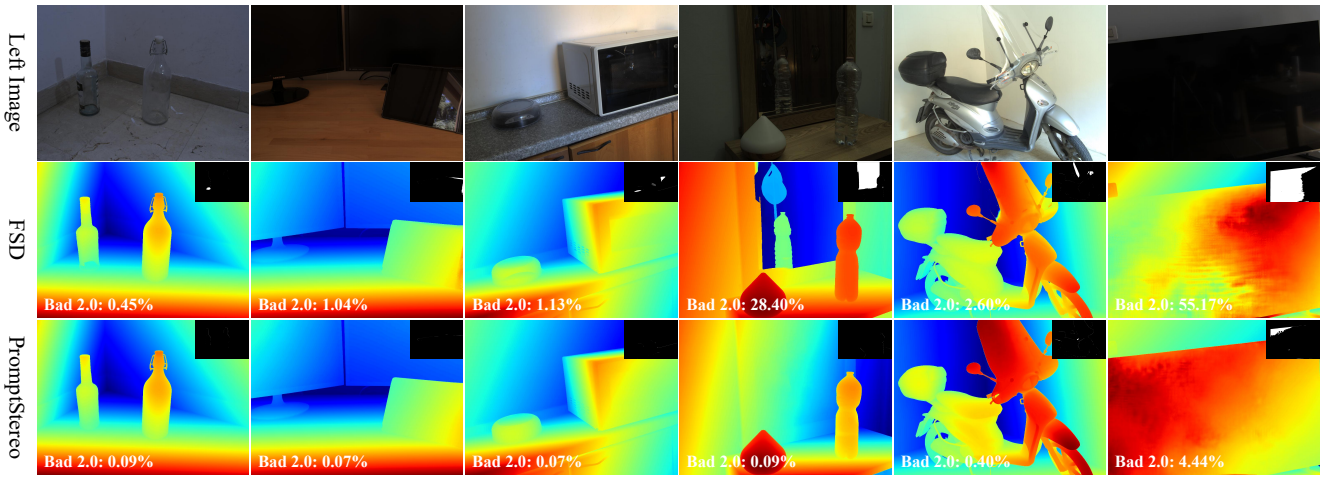

Figure 8. Visualization of Booster (unlimited training sets). The Bad 2.0 metric map is in the upper right corner.

Model	KITTI 2012 Bad 3.0	KITTI 2015 Bad 3.0	Midd-T (H) Bad 2.0	Midd-2021 Bad 2.0	ETH3D Bad 1.0
Stereo Anywhere	3.90	**3.93**	4.49	5.18	1.43
Ours	3.77	4.59	**3.76**	**4.84**	1.30
Ours w/ STA	**3.33**	4.12	4.03	5.06	**1.26**

Table 8. Ablation study of special training strategy.

As shown in Tab. 8, the re-trained PromptStereo model achieves improvements on KITTI and ETH3D, with notable gains on KITTI. Performance on Middlebury decreases slightly, yet both versions of PromptStereo still outperform Stereo Anywhere. We also measure inference time on Scene Flow: Stereo Anywhere requires 0.65 seconds per sample, while PromptStereo needs only 0.36 seconds, representing a substantial efficiency advantage.

6.4. Benchmark Results

To verify that our model is also effective for fine-tuning, we fine-tune it and submit the results to KITTI. As shown in Tab. 9, our model outperforms MonSter, and Fig. 9 also indicates that our model ranks 1st on the KITTI 2012 reflective leaderboard.

KITTI 15	D1-bg Noc	D1-all Noc	D1-bg All	D1-all All
MonSter	1.05	1.33	1.13	**1.41**
Ours	**1.04**	**1.32**	**1.12**	**1.41**
KITTI 12	Out-2 Noc	Out-2 All	Out-3 Noc	Out-3 All
MonSter	1.36	1.75	**0.84**	**1.09**
Ours	**1.32**	**1.69**	**0.84**	**1.09**
KITTI 12 Ref.	Out-2 Noc	Out-2 All	Out-3 Noc	Out-3 All
MonSter	5.66	6.81	2.75	3.38
Ours	**5.11**	**6.17**	**2.54**	**3.08**

Table 9. KITTI leaderboard results.

Table Reflective | Error threshold 3 pixels | Evaluation area All pixels

	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime
1	PromptStereo			2.54 %	3.08 %	0.7 px	0.7 px	0.21 s

Figure 9. We rank 1st on the KITTI 2012 reflective leaderboard.

7. Discuss with DEFOM-Stereo

During the rebuttal and meta-review, PromptStereo is required to clarify its differences with DEFOM-Stereo, which we will explain clearly in this section. We also hope that readers truly understand where the core of PromptStereo lies.

First, PromptStereo does not merely perform a one-time initialization of ViT features using a pre-trained DPT. The PRU, based on the DPT decoder, directly uses the multi-scale features extracted by MonSter for iterative updates, treating DPT as an integral part of the iterative mechanism. In contrast, DEFOM-Stereo still employs GRU for iterations, with its core innovation of obtaining an initial modulated depth map using a depth foundation model and performing scale and delta updates. Our PromptStereo is fundamentally distinct from this. The core contribution of this paper is to propose a new design direction for the iterative update of stereo matching methods: replacing the long-standing default GRU iteration paradigm with a DPT-based architecture.