Learning Stereo from Single Images Supplementary Material

Anonymous ECCV submission

Paper ID 2683

In this document we provide supplementary descriptions and results.

1 KITTI Test Server Evaluation

Figs. 1 and 2 show the complete set of available qualitative results from the held-out 2015 test set from the KITTI online server. We show the official benchmark model of (**GANet official**)[1] [15], a GANet retrained by us using only Sceneflow [9] (**Sceneflow GANet**), and GANet trained only with our MfS data (**MfS GANet (Ours)**). Note that *only* **GANet official** uses KITTI data to finetune, as is apparent in the overly smooth predictions on transparent surfaces like car windows (see rows 6 and 7 in Fig. 1). In Fig. 2 (also rows 6 and 7), can see poor quality predictions in the right of the sky region for the finetuned model, whereas our MfS trained model produces much more sensible predictions. Our results are generally more faithful to the boundaries of the color input image and have fewer artefacts than the alternative methods despite using no LiDAR or synthetic data during training. Quantitative results are presented in Table 1.

Table 1: KITTI 2015 Benchmark. All trained by us on GANet [15], without any finetuning with KITTI LiDAR data. Our model does better than the sceneflow trained alternative. Row three gives the GANet scores (after KITTI LiDAR finetuning) on the same benchmark as reported in their paper [15], and row four gives the current best GANet score on the KITTI benchmark leaderboard. Our scores are not competitive with these scores from models which have had domain-specific KITTI LiDAR finetuning, but our qualitative results are more faithful (Figs. 1 and 2).

Training data	KITTI Finetune	D1-bg Noc	D1-fg Noc	D1-all Noc	D1-bg All	D1-fg All	D1-all All
Sceneflow GANet		3.60	16.56	5.74	3.86	17.21	6.08
MfS GANet (Ours)		**2.96**	**15.09**	**4.97**	**3.13**	**15.57**	**5.20**
GANet [15] (in paper)	✓		3.39	1.84		3.91	2.03
GANet Official [15]	✓	1.34	3.11	1.63	1.48	3.46	1.81

[1] `http://www.cvlibs.net/datasets/kitti/eval_scene_flow_detail.php?benchmark=stereo&result=ccb2b24d3e08ec968368f85a4eeab8b668e70b8c`

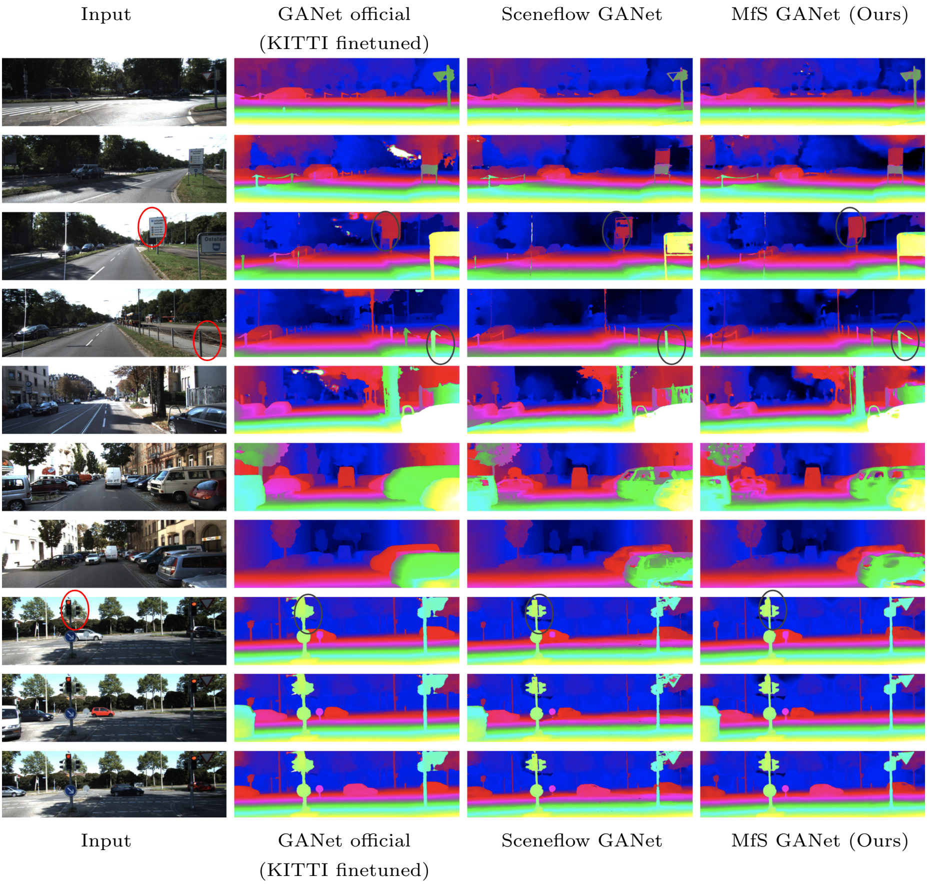

Fig. 1: KITTI 2015 benchmark qualitative comparison. On the held-out KITTI 2015 benchmark images our predictions MfS GANet (Ours), produced without any KITTI finetuning, are qualitatively more faithful to the scene geometry than both the Sceneflow-trained model and the official KITTI-finetund GANet [15]. Visualisations here are generated by the KITTI upload server. [5].

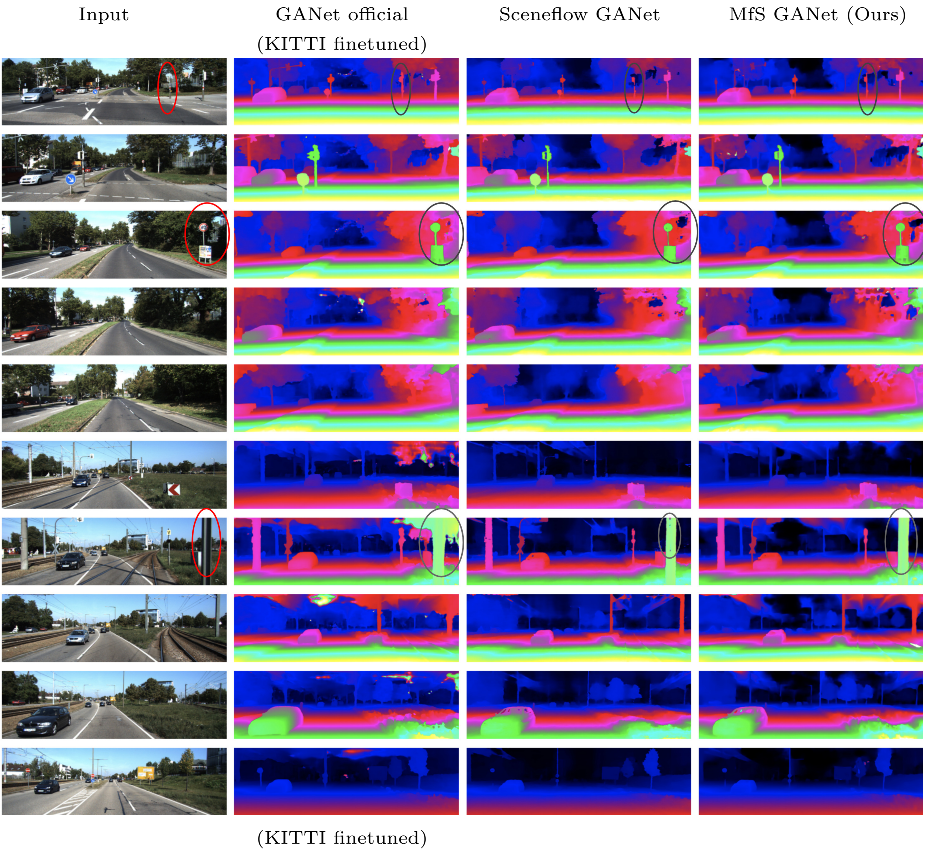

Fig. 2: KITTI 2015 benchmark qualitative comparison – continued. See Fig. 1 caption for details.

2 Image Synthesis Visualisation

When converting predicted depth Z to disparity D we can vary the scaling factor s; $\tilde{D} = \frac{s\, Z_{\max}}{Z}$. In Fig. 3 we illustrate the resulting $\tilde{I}_r$ for different values of s. As s becomes larger, we see that it has the effect of changing the relative camera viewpoint of the synthesized right image e.g. we observe the front of the bus occluding the trash can on the sidewalk.

(a) Input (left) image (b) Mono estimation [11]

(c) $\tilde{I}_r(s = 20)$ (d) $\tilde{I}_r(s = 50)$ (e) $\tilde{I}_r(s = 100)$

(f) $\tilde{I}_r(s = 150)$ (g) $\tilde{I}_r(s = 200)$ (h) $\tilde{I}_r(s = 225)$

Fig. 3: Visualization of different scaling factors s**.** Here we show the effect of varying the scaling factor s (c - h) when constructing the synthesized right image $\tilde{I}_r$ from the input left image (a) and its predicted depth (b).

3 Baseline Algorithm Details

Here we describe the baseline image synthesis approaches we compared to in Table 1 in the main paper. We show qualitative results from these approaches here in Fig. 7 and in Fig. 6 in the main paper.

Affine Warps — For the affine warps baseline, similar to [2], we warp each I_l with a top shift s_{top} and a bottom shift s_{bottom}, with 50% probability $s_{\text{top}} \sim \text{Unif}\,[0, d_{\max}]$ and $s_{\text{bottom}} \sim \text{Unif}\,[0, s_{\text{top}}]$; and with 50% probability $s_{\text{bottom}} \sim$

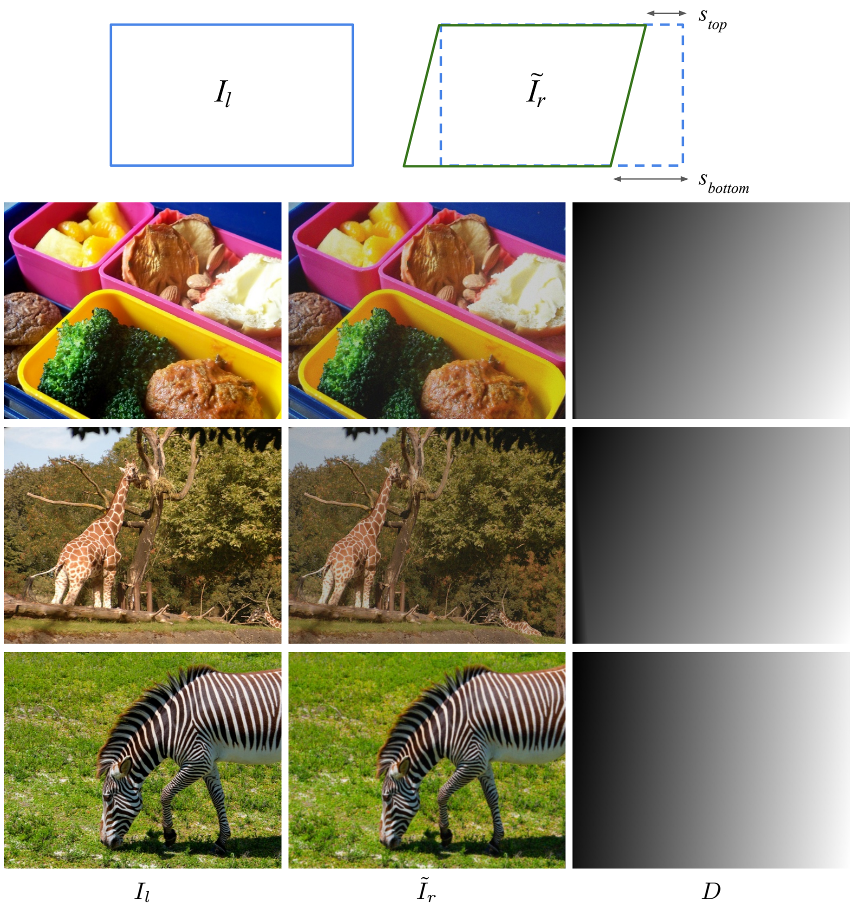

Fig. 4: Affine warp baseline. The diagram at top shows how $s_{\text{top}}, s_{\text{bottom}}$ relate to the warp applied to $\tilde{I}_r$. Below, we show examples of the resulting affine warped images.

$\text{Unif}\,[0, d_{\max}]$ and $s_{\text{top}} \sim \text{Unif}\,[0, s_{\text{bottom}}]$. Following the warping, I_l and $\tilde{I}_r$ are both cropped to avoid black borders, and the corresponding D is adjusted appropriately to account for this cropping. Details and example training images are shown in Fig. 4.

Random Pasted Shapes — For the random pasted shapes baseline we paste 'foreground' shapes onto a 'background' image in a similar manner to [8]. We start with affine warped images, but with $d_{\max} = 50$ to help to ensure that foreground 'objects' move with greater disparity than the 'background' image. We then sample num_patches $\sim \text{Unif}[0, 10]$ and for each patch we load a random image from the training set and mask out a region. The masked region is pasted on top of I_l, and a translated version of the masked region is placed on $\tilde{I}_r$. Masks are generated with equal probability from rectangles, partial ellipses, polygons

Fig. 5: Random pasted shapes baseline. We show examples of images warped using this baseline.

and thin objects. With the exception of rectangles, masks are generated with OpenCV's drawing functions. Examples of random pasted shapes training tuples are shown in Fig. 5. Note that the disparity of the background image is less visible compared with the disparity maps from the Affine Warp baseline, due to the reduced $d_{\max}$ used here.

Rectangles are axis-aligned, with x-axis and y-axis bounds sampled from $\text{Unif}[0.1W, 0.9W]$ and $\text{Unif}[0.1H, 0.9H]$ respectively.

Partial ellipses have a center (x, y), where $x \sim \text{Unif}[0.1W, 0.9W]$ and $y \sim \text{Unif}[0.1H, 0.9H]$. With 75% probability the ellipse is a full ellipse, with start_angle = 0 and end_angle = 360. Otherwise, the ellipse is partial, with the angular bounds sampled from $\text{Unif}[0, 360]$. The rotation angle of the ellipse is sampled from $\text{Unif}[0, 360]$, and the axes sizes are sampled from $\text{Unif}[0.1W, 0.9W]$.

Polygons are generated using the approach described in [10], with a number of sides sampled from $\text{Unif}[3, 20]$, spikyness = 0.8, irregularity = 0.5, and aveRadius $\sim \text{Unif}[0.01W, 0.3W]$.

Thin objects are full ellipses with with minor axis $\sim \text{Unif}[0.001H, 0.025H]$ and major axis $\sim \text{Unif}[0.1W, 0.5W]$.

I_l $\tilde{I}_r$ D

Fig. 6: Random superpixels baseline. Synthesized stereo pairs using superpixel warping.

Random Superpixels — For this baseline, we segment I_l into superpixels using [3, 12], with parameters scale $\sim$ Unif$[50, 200]$, $\sigma \sim$ Unif$[0, 1]$, and min_size $\sim$ Unif$[75, 275]$. We initialise D as a plane, where pixel i has disparity value d_i defined by $d_i = ax + by + c$ with $a \sim$ Unif$[-0.025, 0.025]$, $b \sim$ Unif$[0.3, 0.4]$, and $c \sim$ Unif$[15, 20]$. Superpixels s_j in I_l are chosen with probability 0.6 to act as foreground objects. We set the disparity values of these foreground superpixels as $d_j = \Sigma_{i \in j} \frac{d_i}{n} + x_j$, where $x_j \sim$ Unif$[0, 64]$ and n is the number of pixels in superpixel s_j. Finally we clip the values of D to lie between 0 and $d_{\max}$. Using D, we then forward warp I_l to generate our right image $\tilde{I}_r$, handing occlusions and collisions in the way described in Section 3.2 of the main paper. Examples of this training data are shown in Fig. 6.

4 Evaluation Details

Here we provide additional details for the experiments in main paper.

4.1 Evaluation Image Resolution

Due to architecture constraints, we make predictions at slightly different resolutions for each network; see Table 2 for details. Note that all predictions are resized to the native resolution of the ground truth for evaluation. When training a given architecture with different datasets we use the same resolution for all datasets for fair comparison.

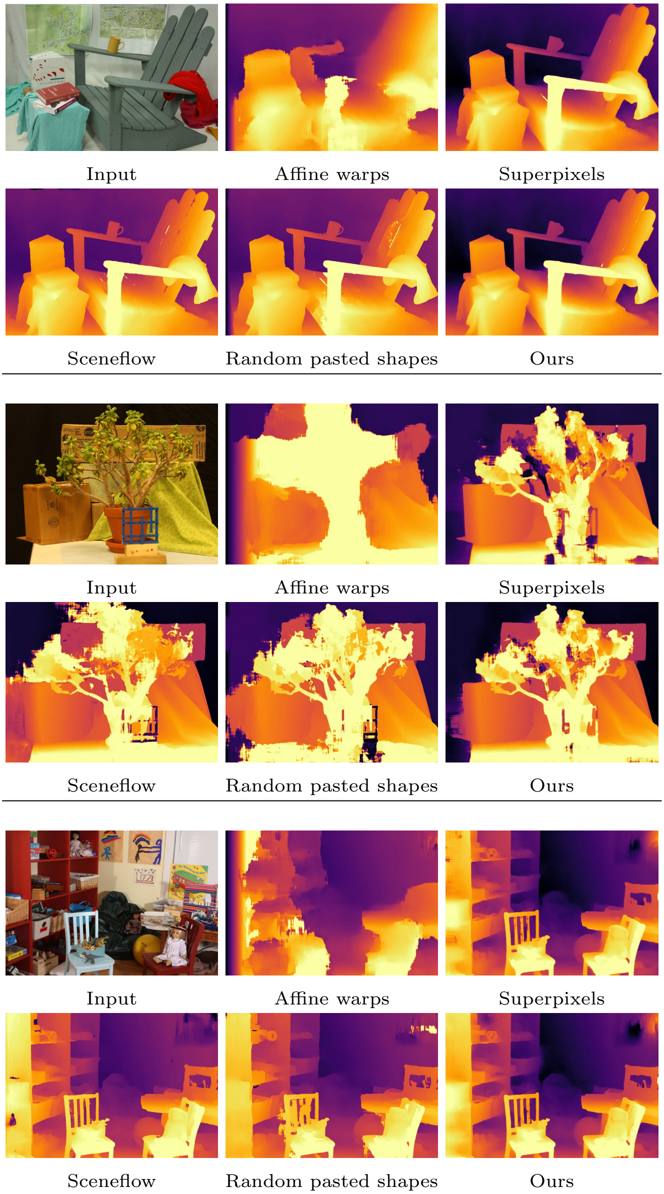

Fig. 7: Additional results comparing our method to the baseline algorithms.

Table 2: Evaluation image resolution. Here we show the prediction resolutions for each architecture. Note that all predictions are resized to the native resolution of the ground truth for evaluation.

Architecture	KITTI '12	KITTI '15	Middlebury	ETH3D
iResnet [7]	1280×384	1280×384	1280×768	768×448
PSM [1]	1280×384	1280×384	1280×768	768×448
GANet [15]	1248×384	1248×384	1248×768	768×432

4.2 KITTI Evaluation Splits

KITTI 2012 results in the paper are reported on the 40 images from the validation split of [1]. The image indices used by both us and [1] are `[3, 6, 20, 26, 38, 41, 43, 44, 49, 60, 67, 70, 81, 84, 89, 97, 109, 119, 122, 123, 129, 130, 132, 134, 141, 144, 152, 158, 165, 171, 174, 179, 184, 186]`. The KITTI 2015 indices are `[1, 3, 6, 20, 26, 35, 38, 41, 43, 44, 49, 60, 67, 70, 81, 84, 89, 97, 109, 119, 122, 123, 129, 130, 132, 134, 141, 144, 152, 158, 159, 165, 171, 174, 179, 182, 184, 186, 187, 196]`.

5 Additional KITTI Results

In Table 3 we present additional metrics for the KITTI 2012 experiments from Table 1 in the main paper where we compare different methods for generating training data. We also show additional metrics for KITTI 2015. As in the main paper, our method brings consistent improvement over the baselines.

Table 3: KITTI 2012 and 2015 Results. Additional metrics for PSMNet's [1] trained with different stereo data.

Synthesis approach	Training data	EPE Noc	<3px Noc	EPE All	<3px All
KITTI 2012					
Affine warps	MfS	3.04	12.72	3.74	14.78
Random pasted shapes	MfS	2.70	9.62	3.38	11.21
Random superpixels	MfS	1.15	4.97	1.33	5.90
Synthetic	Sceneflow	0.95	4.77	1.03	5.51
Ours	MfS	**0.77**	**3.58**	**0.91**	**4.42**
KITTI 2015					
Affine warps	MfS	2.05	13.67	2.33	14.94
Random pasted shapes	MfS	1.29	6.33	1.53	7.63
Random superpixels	MfS	1.13	5.11	1.15	5.38
Synthetic	Sceneflow	1.18	5.54	1.19	5.73
Ours	MfS	**1.06**	**4.80**	**1.07**	**4.92**

6 Recovering from Monocular Depth Errors

Fig. 7 in the main paper shows that our trained stereo networks can overcome some of the errors of monocular depth estimation. In Fig. 8 here, we observe the same result across three different monocular depth networks: MiDaS [11], Megadepth [6], and Monodepth2 [4]. In each case, problems present in monocular depths, such as missing objects and uneven ground planes do not transfer to our eventual stereo predictions. We note here that although Monodepth2 [4] was trained using monocular videos from KITTI, our resulting stereo network has never seen images from this dataset.

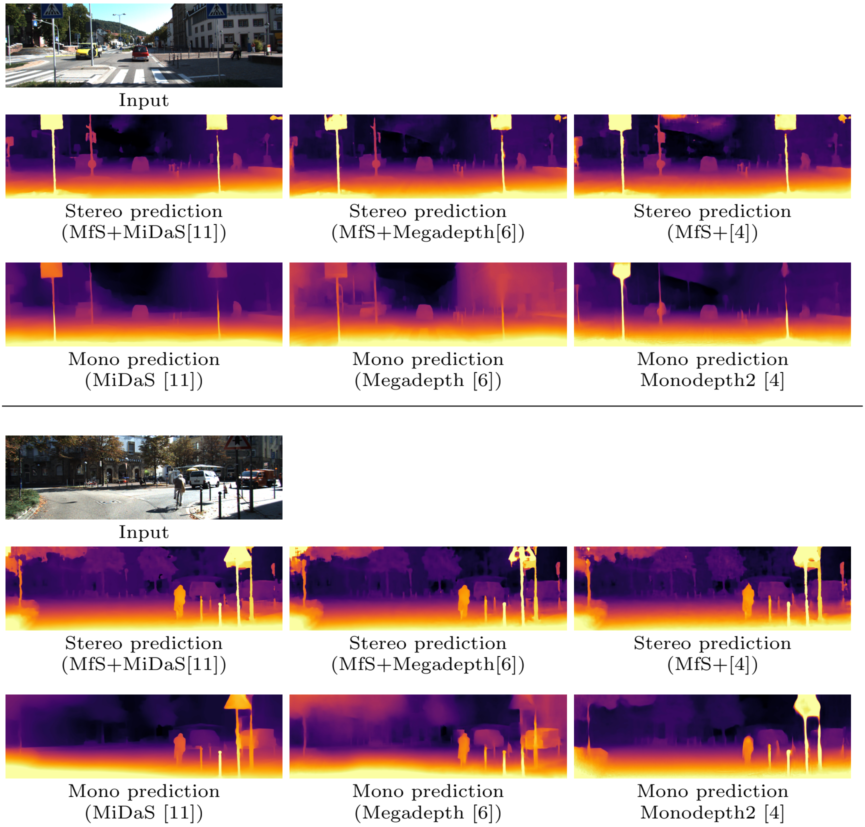

Fig. 8: We can recover from errors in monocular depth estimation.

7 Ablation

Table 4 shows the full set of ablation results for our method, again justifying our design decisions (see Table 4 in the main paper). We show some qualitative comparisons for this experiment in Fig. 9.

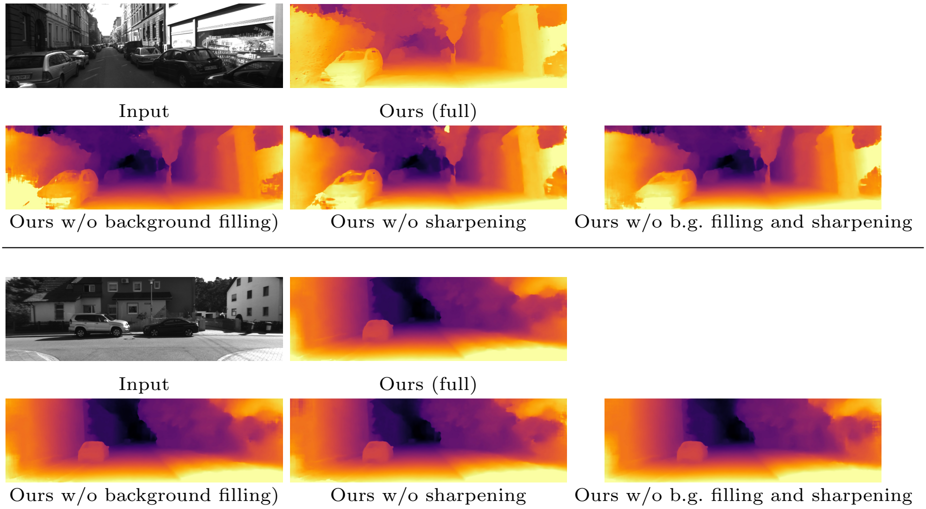

Fig. 9: Qualitative Ablation Results. Here we see stereo predictions for models trained with different parts of our synthesis pipeline disabled. A quantitative comparison is available in Table 4.

8 Training for Longer

For all the stereo results in the main paper we trained for 175k steps. In Table 5, we present results for training both PSMNet and GANet for and additional 175k and 150k steps respectively. We can see that longer training benefits both our MfS and Sceneflow trained stereo models. However, we still outperform the Sceneflow trained models when we increase the number of iterations.[2]

9 Additional Qualitative Results

In Figs. 10 and 11 we present additional comparisons using PSMNet to Sceneflow training versus our MfS dataset for both KITTI 2012 and KITTI 2015. In Figs. 12

[2] Note that row 1 in Table 6 in the main paper has incorrect numbers for KITTI — these should be the same as row 4 in Table 2 in the main paper.

Table 4: Ablation results. By including all parts of our synthesis pipeline when creating training we achieve the best results overall (bottom row).

		KITTI '12		KITTI '15		Middlebury		ETH3D	
Sharpening	Background filling	EPE	<3px	EPE	<3px	EPE	<2px	EPE	<1px
✗	✗	1.03	5.22	1.09	5.37	8.15	31.89	**0.51**	9.59
✗	✓	**0.88**	4.88	**1.07**	5.14	7.44	29.17	0.52	9.44
✓	✗	1.06	4.90	1.08	4.98	7.20	27.66	0.57	9.03
✓	✓	0.91	**4.43**	**1.07**	**4.92**	**6.34**	**27.33**	0.52	**8.78**

Table 5: Training for longer. Stereo models trained with our MfS dataset still outperform Sceneflow models even with more training steps.

Architecture	Training data	Steps	KITTI '12 EPE	KITTI '12 <3px	KITTI '15 EPE	KITTI '15 <3px	Middlebury EPE	Middlebury <2px	ETH3D EPE	ETH3D <1px
PSMNet [1]	Sceneflow	175k	1.03	5.51	1.19	5.73	9.45	36.09	0.67	14.66
		350k	1.01	5.31	1.15	5.62	8.54	34.04	0.68	11.51
PSMNet [1]	MfS	175k	**0.91**	4.43	1.07	4.92	6.34	27.33	0.52	8.78
		350k	1.02	**4.36**	**1.04**	**4.56**	**6.26**	**25.38**	**0.44**	**7.35**
GANet [15]	Sceneflow	175k	1.00	5.45	1.21	6.11	10.94	32.57	0.49	9.97
		325k	0.96	5.24	1.14	5.43	9.81	32.20	0.48	9.45
GANet [15]	MfS	175k	**0.81**	4.32	1.04	4.66	5.54	24.75	0.44	7.73
		325k	0.83	**4.27**	**1.03**	**4.61**	**5.29**	**23.79**	**0.41**	**6.45**

and 13 we compare using Flickr1024 [13], a dataset stereo images collected online. For Flickr1024, we should results using our method using depth generated by MiDaS [11] or MegaDepth [6]. It is worth remembering that MegaDepth [6] does not use any synthetic or ground truth depth at training time but it still can be used to train a stereo model that produces high quality predictions. We reached out to the authors of the ReDWeb dataset [14] so we could evaluate on it, but unfortunately the original input stereo frames are not available. In all cases we see that our fully automatic data generation pipeline results in high quality stereo predictions without directly requiring any synthetic training data which is time consuming to create.

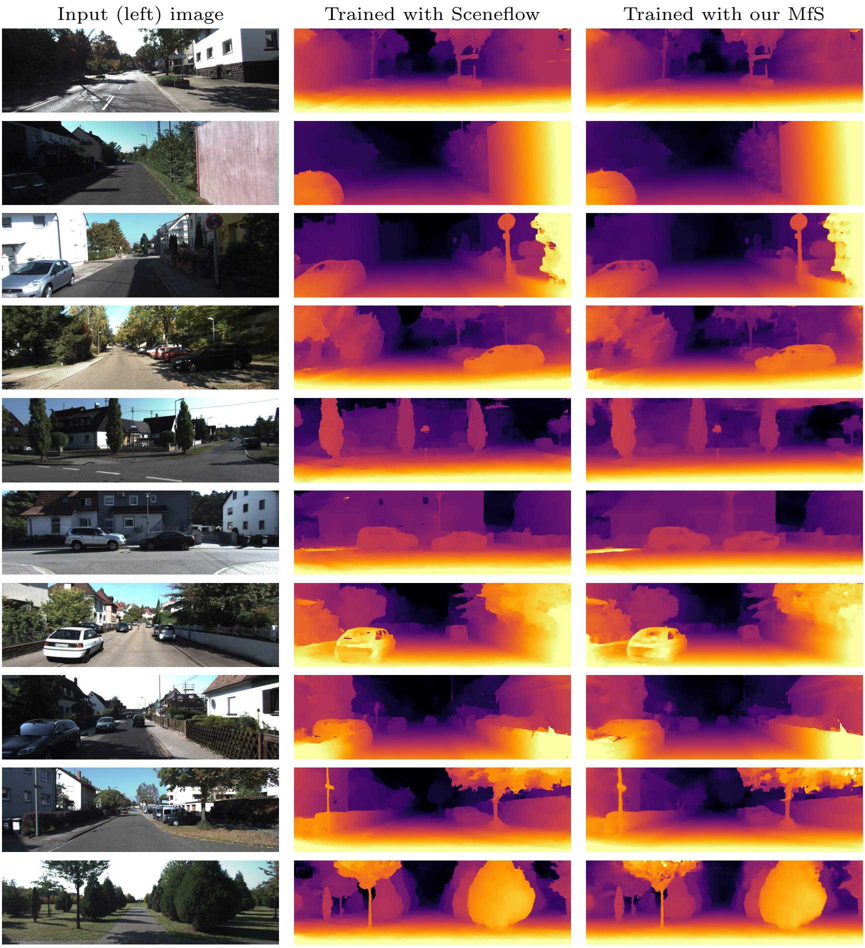

Fig. 10: Additional KITTI 2012 qualitative results.

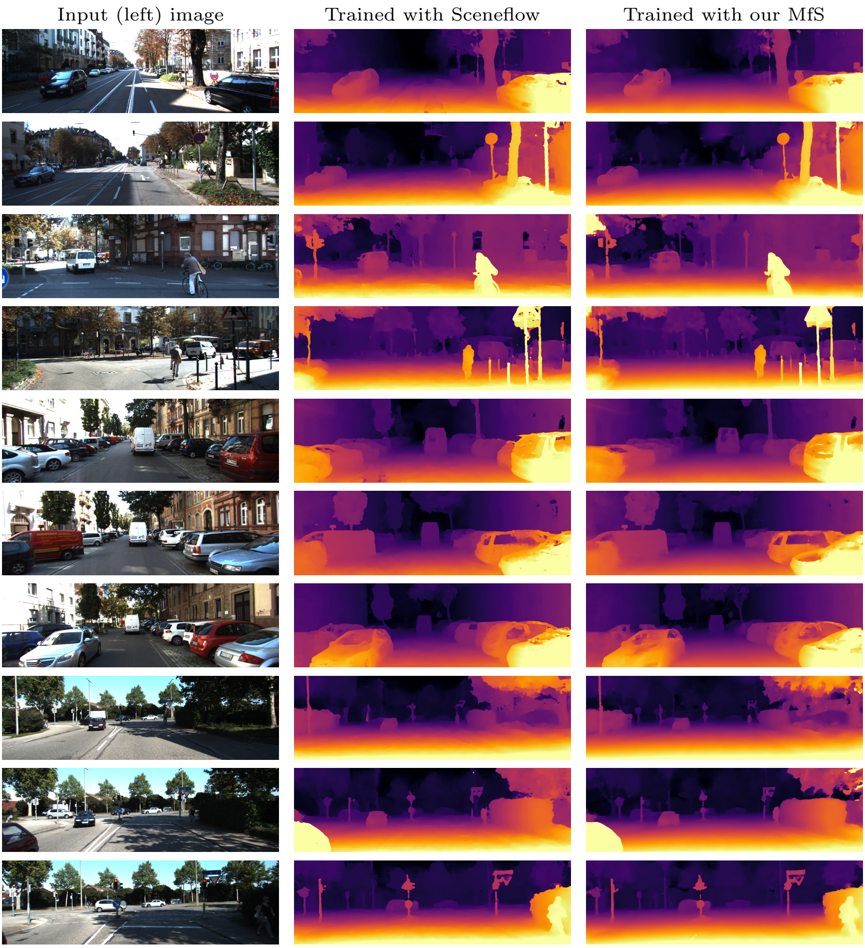

Fig. 11: Additional KITTI 2015 qualitative results.

Fig. 12: Additional Flickr1024 [13] qualitative results.

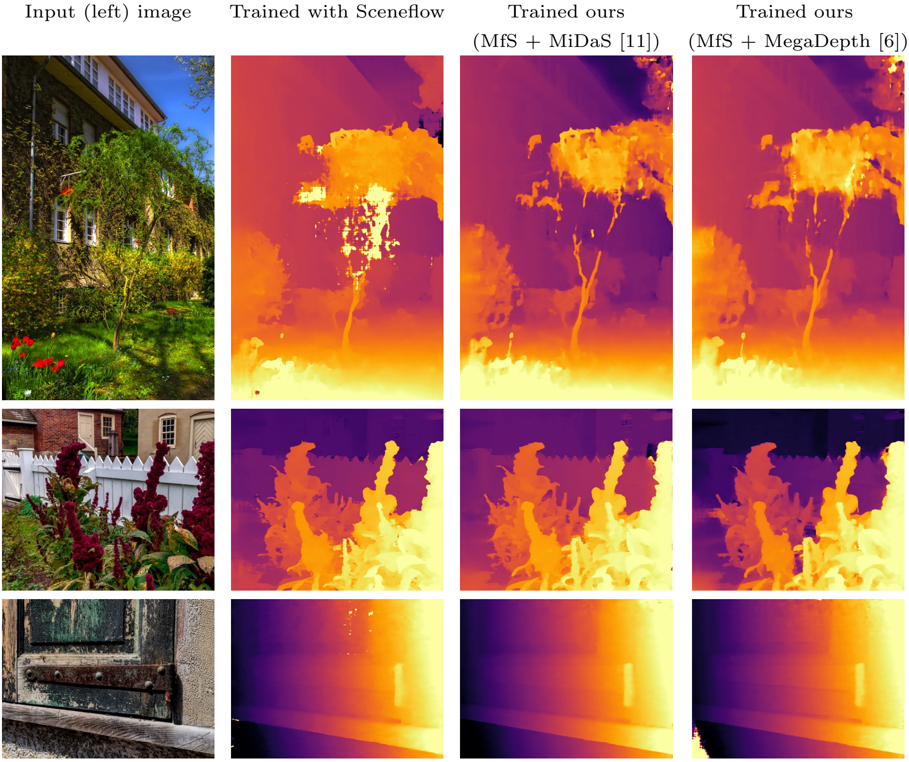

Fig. 13: Additional Flickr1024 [13] qualitative results.

References

1. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: CVPR (2018)
2. DeTone, D., Malisiewicz, T., Rabinovich, A.: SuperPoint: Self-supervised interest point detection and description. In: CVPR Deep Learning for Visual SLAM Workshop (2018)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)
4. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019)
5. KITTI: Kitti stereo evaluation 2015 server. `http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo`, accessed: 2020-03-10
6. Li, Z., Snavely, N.: MegaDepth: Learning single-view depth prediction from internet photos. In: CVPR (2018)
7. Liang, Z., Feng, Y., Guo, Y., Liu, H., Chen, W., Qiao, L., Zhou, L., Zhang, J.: Learning for disparity estimation through feature constancy. In: CVPR (2018)
8. Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., Brox, T.: What makes good synthetic training data for learning disparity and optical flow estimation? IJCV (2018)
9. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR (2016)
10. Ounsworth, M.: Algorithm to generate a random 2D polygon. `https://stackoverflow.com/a/25276331/279858`, accessed: 2020-03-10
11. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv:1907.01341 (2019)
12. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: scikit-image: image processing in python. PeerJ (2014)
13. Wang, Y., Wang, L., Yang, J., An, W., Guo, Y.: Flickr1024: A large-scale dataset for stereo image super-resolution. In: ICCV Workshops (2019)
14. Xian, K., Shen, C., Cao, Z., Lu, H., Xiao, Y., Li, R., Luo, Z.: Monocular relative depth perception with web stereo data supervision. In: CVPR (2018)
15. Zhang, F., Prisacariu, V., Yang, R., Torr, P.H.: GA-Net: Guided aggregation net for end-to-end stereo matching. In: CVPR (2019)