NeRF-Supervised Deep Stereo – Supplementary Material

Fabio Tosi[1] Alessio Tonioni[2] Daniele De Gregorio[3] Matteo Poggi[1]
[1]University of Bologna [2]Google Inc. [3]Eyecan.ai
{fabio.tosi5, m.poggi}@unibo.it alessiot@google.com daniele.degregorio@eyecan.ai

This document reports additional material concerning our work "NeRF-Supervised Deep Stereo".

1. Additional Implementation Details

Depth Label Scaling. We found through empirical observation that depth labels generated by Instant-NGP [6] and converted into disparities often have absolute mean errors of about 0.5-1.5 disparities when considering an image resolution of 2Mpx. This was confirmed through manual verification and comparison of the generated disparity maps with those obtained by both SGM and an *oracle* stereo network, which was RAFT-Stereo trained on both synthetic and real data annotated with ground-truth disparities. This misalignment can result in sub-optimal performance for stereo networks trained on this data. The root cause of this effect in the original Instant-NGP code is unclear, but it can be compensated. As a pre-processing step, we adjust the rendered disparity maps generated by Instant-NGP by fitting a scale-shift pair of values for each triplet. This correction is performed in a self-supervised manner by optimizing a scale-shift pair of values for each triplet. The values are initialized to (1,0) and optimized through stochastic gradient descent to minimize the minimum reprojection error between the images of the considered triplet-stereo pair according to the rescaled disparity. No external labels or cues are used during the optimization process. This pre-processing step, consisting of only 10 optimization steps, takes less than a second per rendered disparity and generally results in scale and shift values close to (1,0), such as ~ 1.01 and $1e^{-6}$. Although we cannot speculate on the causes of the initial alignment, this simple pre-processing allows for solving it, as highlighted by the high accuracy of the stereo networks we are able to train on our data.

Deep Stereo Testing. At test time, we evaluate the accuracy of all the trained models using stereo images at their native resolution, with the exception of PSMNet and CFNet when evaluated on the Midd-A and Midd-T datasets at full resolution (F). Due to high memory requirements, in such cases, we feed both networks with stereo pairs at half of the original resolution. The predicted disparity maps are then upsampled using nearest neighbor interpolation and evaluated at the same spatial resolution as the available ground-truth. We adopt the same protocol when testing models of our competitors, as shown in Table 6 of the main paper, except for [9]. For this model, we follow the same image rescaling procedure suggested by the authors that produces the best results, with resolutions of 1280×768 for Middlebury, 1280×384 for KITTI, and 768×448 for ETH3D. Changing this size results in lower accuracy compared to the results reported in the main paper. For example, if the Midd-A images are resized to 2560×1536, the MfS-RAFT-Stereo model's error increases to 23.95% at full resolution, compared to the 19.79% reported in Table 5 (G) of the main paper.

2. Collected Dataset

We provide an overview of the scenes in our dataset. Fig. 1 shows examples taken from 60 out of the 270 sequences collected for our experiments. We want to emphasize the diversity of the contexts depicted, including indoor and outdoor environments, objects, plants, urban elements, and more.

Figures 2, 3 and 4 show, for a total of 10 scenes, a subset of 15 images from different viewpoints captured during our acquisition campaign. Respectively, Fig. 2 shows four scenes featuring very thin objects, over which the synergy between the triplet photometric loss $\mathcal{L}_{3\rho}$ and the depth labels rendered by NeRF allows to provide a strong supervision for training a stereo network without the need for additional sensors to obtain ground-truth annotations. Fig. 3 displays a few small-scale scenes collected by focusing on one or multiple small objects. Lastly, Figure 4 exhibits four examples of scenes collected in a full 360° manner, as evidenced by the images that capture a wheelbarrow and a fountain from various viewpoints all around them.

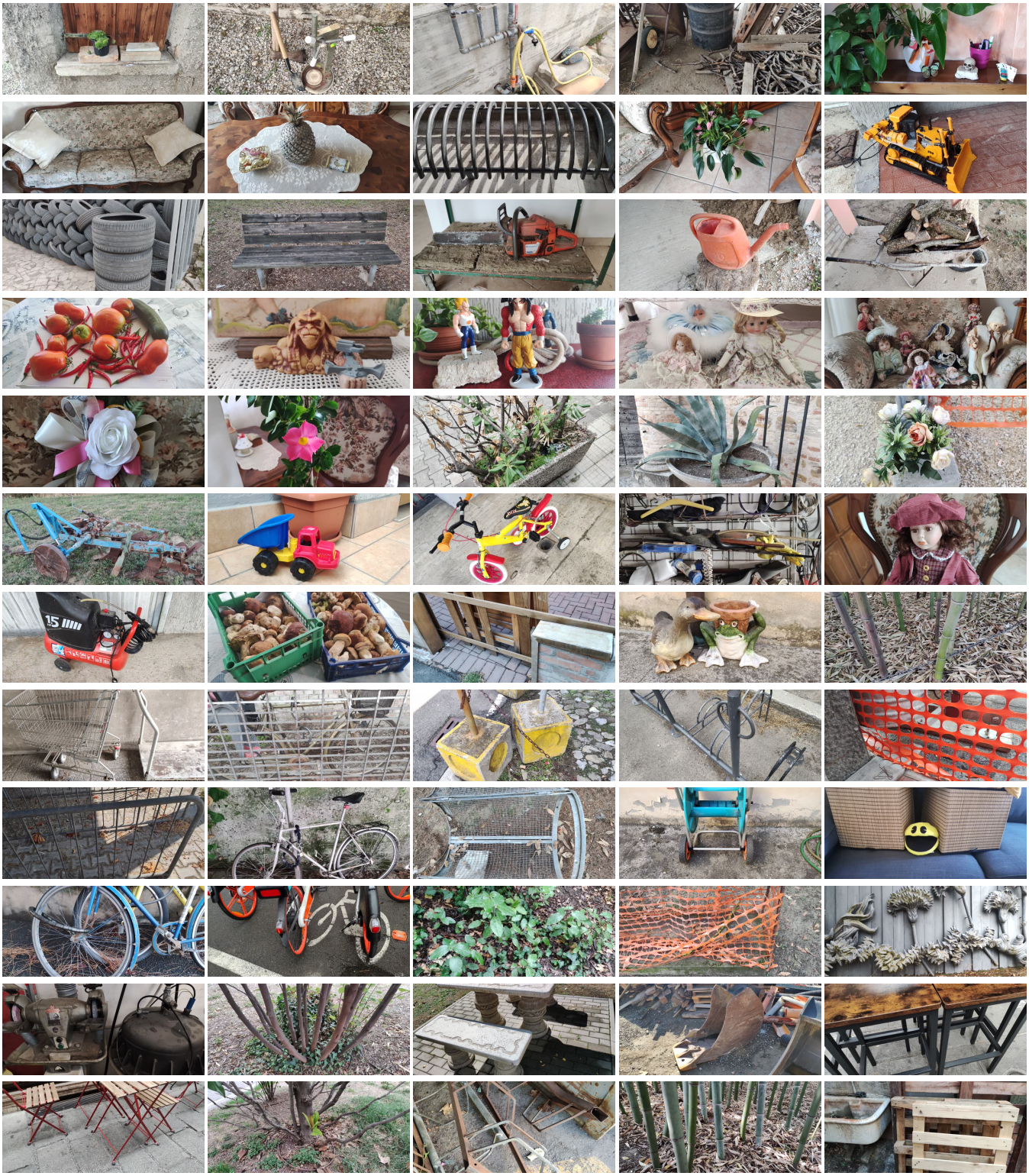

Figure 1. **Examples of scenes in our dataset.** We report single samples from 60 of the scenes composing our datasets.

Figure 2. **Examples of multiple viewpoints collected for single scenes.** We show 15 images collected from different viewpoints for 4 of the scenes in our dataset.

Figure 3. **Examples of multiple viewpoints collected for single scenes.** We show 15 images collected from different viewpoints for 4 of the scene in our dataset.

Figure 4. **Examples of multiple viewpoints collected for single scenes (360° acquisition).** We show 15 images collected from different viewpoints for 4 of the scene in our dataset.

3. Rendered Training Data

Now, we present a few examples of rendered image triplets and their corresponding depth and ambient occlusion maps, with reference to the center view. Figures 5 and 6 demonstrate some examples highlighting the impact of various baselines on the rendered views, as well as the quality of the rendered depth map and how the ambient occlusion map can effectively detect most depth outliers. On the leftmost and rightmost columns, we display the rendered left and right images in a triplet, respectively. From top to bottom, we illustrate the effect of different virtual baselines on the rendered images. In particular, we can observe how some details in the rendered frames align with the red, green and blue lines when shifting between small, medium, and large baselines – i.e., 0.1, 0.3, and 0.5 units.

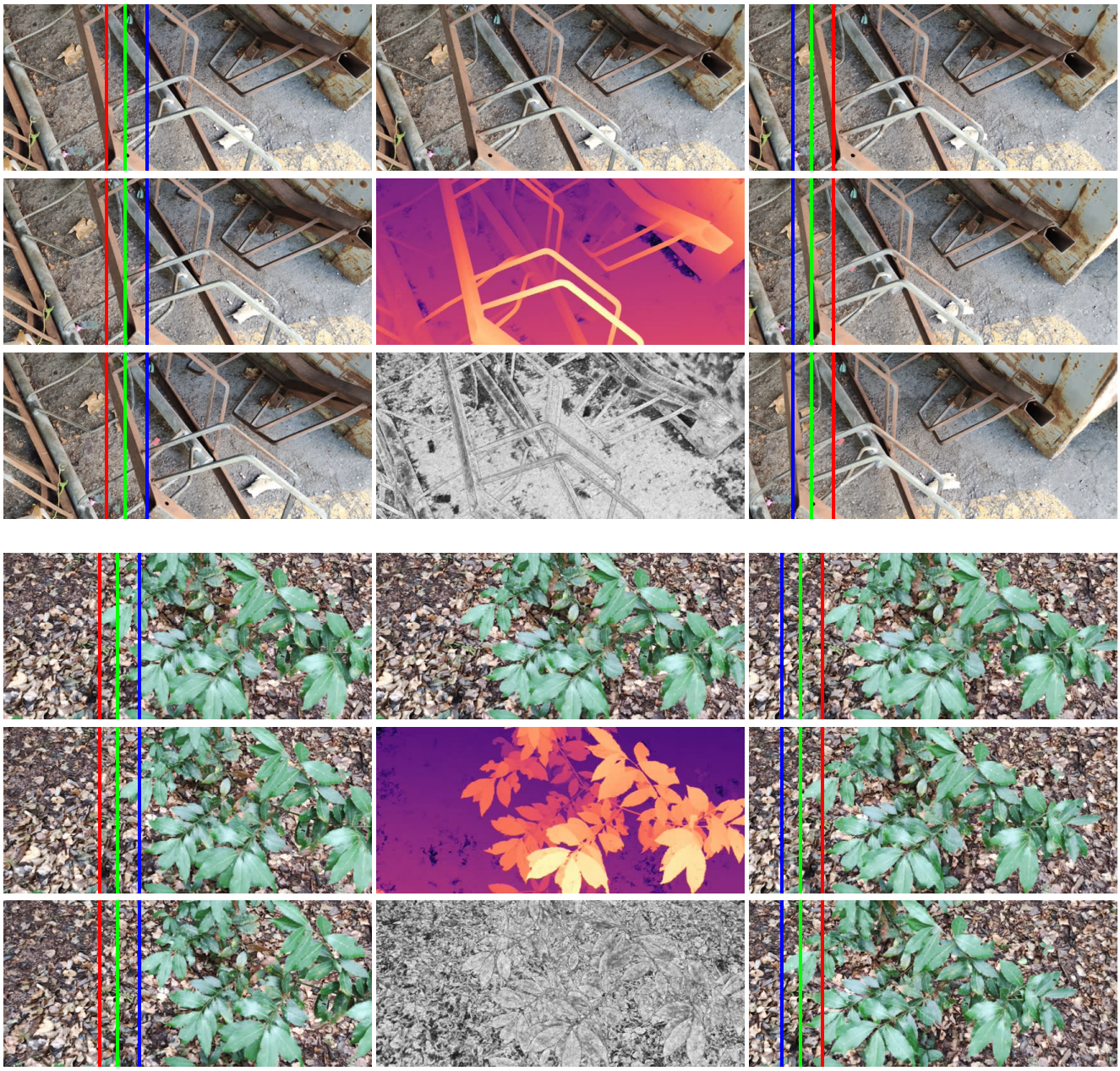

Figure 5. **Examples of rendered images and depth.** We show examples on 2 scenes of our dataset. In each case, the leftmost and rightmost columns show the rendered left and right images in a triplet, respectively. These images were obtained using small, medium, and large baselines, as indicated by the red, green, and blue lines. The center column, from top to bottom, shows the center image in the triplet, its corresponding rendered disparity map, and ambient occlusion map.

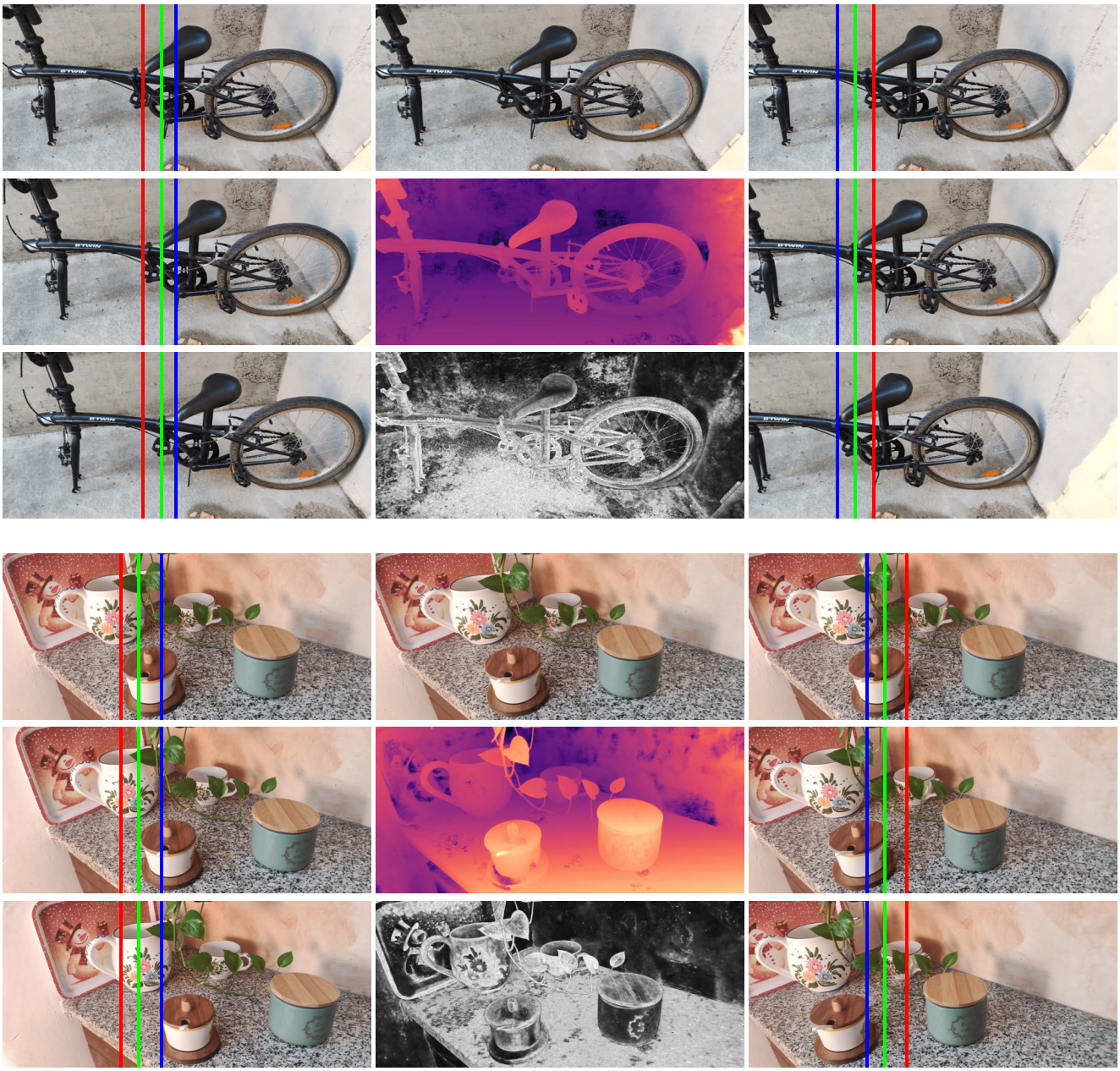

Figure 6. **Examples of rendered images and depth.** We show examples on 2 scenes of our dataset. In each case, the leftmost and rightmost columns show the rendered left and right images in a triplet, respectively. These images were obtained using small, medium, and large baselines, as indicated by the red, green, and blue lines. The center column, from top to bottom, shows the center image in the triplet, its corresponding rendered disparity map, and ambient occlusion map.

4. Ambient Occlusion as Rendering Confidence

In this section, we examine the use of Ambient Occlusion (AO) as a way to determine the accuracy of depth labels generated by our NeRF engine. Ambient Occlusion encodes the degree of opacity of a pixel in the rendered image and is represented in a range of $[0, 1]$. Our empirical findings suggest that the majority of depth map artifacts are caused by holes in the 3D scenes, i.e., regions labeled with large depth values, even though they are close in the scene. These holes can be identified by low AO values, which can be caused by factors such as varying illumination during image collection or low visibility. In Fig. 7, we demonstrate the use of Ambient Occlusion (AO) as a form of confidence for identifying inaccurate depth labels in the rendered images produced by our NeRF engine. The figure shows the results of filtering the rendered depth maps based on AO values, and the impact on the accuracy of the depth maps. First, the top row of the figure shows the rendered disparity and AO maps, as well as the prediction from an *oracle* network, RAFT-Stereo, which was trained on both synthetic and real data, and was labeled with ground-truth. This network was adopted for evaluation purposes, considering the absence of ground-truth depth labels in our dataset. Next, we present sparse depth maps obtained by removing pixels having AO values lower than 0.25, 0.50, 0.75, and 0.90, respectively. These maps are followed by their corresponding error maps with respect to the oracle disparities. The error maps are annotated with the density of pixels and the percentage of pixels with errors larger than 2. The results show that by visually inspecting the sparse maps, many artifacts can be removed from the depth maps by filtering pixels with AO values less than 0.25. By increasing this threshold, the artifacts can be completely removed. The error maps further confirm the effectiveness of this filtering procedure.

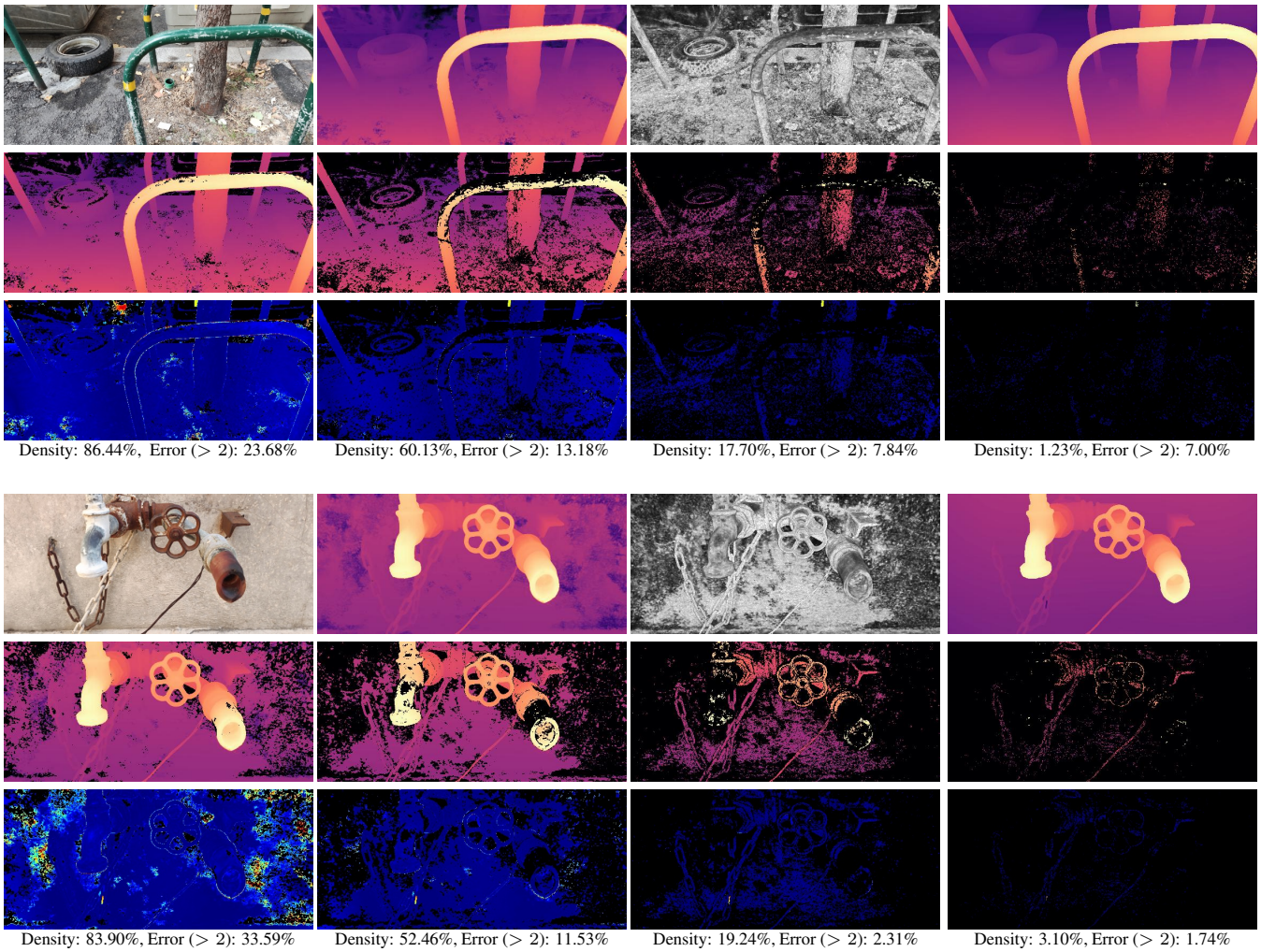

Figure 7. **Depth map filtering through Ambient Occlusion.** On top row: RGB, rendered disparity and AO maps by NeRF, *oracle* disparity map by RAFT-Stereo. On the second row, depth maps filtered by setting th to 0.25, 0.50, 0.75 and 0.90 respectively. At the bottom: error maps with respect to the *oracle* disparity by RAFT. In this examples, we consider rendered images and disparities at 2Mpx.

Here, we examine the effect of the choice of Ambient Occlusion (AO) threshold on the accuracy of our trained stereo model, RAFT-Stereo. The results on the validation set, trained using our $\mathcal{L}_{NS}$ loss with varying th values, are summarized in Table 1. The optimal results are obtained when th is set to 0.25 or 0.50, with the latter resulting in better performance on three out of five cases. However, a more stringent filtering reduces the label density, resulting in less effective training.

AO threshold	KITTI-12 (> 3px)	Midd-A F (> 2px)	Midd-A H (> 2px)	Midd-A Q (> 2px)	Midd- 21 (> 2px)
0.90	4.62	17.94	9.72	9.66	16.81
0.75	4.86	18.90	10.59	10.00	18.27
0.50	**4.31**	14.92	8.75	**8.28**	**14.87**
0.25	6.20	**14.59**	**8.45**	8.58	16.30

Table 1. **Ablation Study – Impact of AO threshold.** We set different thresholds th over AO values – 0.25, 0.50, 0.75 and 0.90.

5. Rendering Failure Cases

NeRF is capable of rendering images from any viewpoint in the scene, however, in some cases, it may fail when selecting camera positions and orientations that point to parts of the scene that were rarely or never observed during the image collection phase. As shown in Fig. 8, this can result in artifacts appearing in one of the three images in the triplets. This is due to the displacement caused by the virtual baseline (0.5 units in the examples), which moves the viewpoint towards parts of the scene that were rarely seen during acquisition. Our empirical observations show that AO is a good indicator of the presence of these artifacts, with images having a low average AO often exhibiting this issue. Therefore, when preparing our training dataset, we consider the normal distribution and standard deviation of the average AO for each rendered image of a scene. We then discard any triplet that contains at least one synthesized image with an average AO value below the standard deviation from the mean of the distribution. This results in 65 148 triplets out of the 81 000 that were obtainable by rendering frames from our 270 scenes (each consisting of 100 images) using the three virtual baselines.

Figure 8. **Rendering Failure Cases.** We show seven examples of image triplets characterized by large artefacts in one of the three frames.

6. Rendered Depth Labels

In this section, we examine in detail the proxy labels that can be obtained from our rendered triplets. As shown in Fig. 9, 7 examples from different scenes in our dataset are presented, exhibiting three different types of labels. From left to right, we show depth maps rendered by NeRF, followed by disparity maps calculated using SGM on the center-right stereo pair, as well as disparity maps computed by running SGM on the triplet itself. For the latter case, we compute two disparity maps between the center-right and center-left pairs, and filter them to remove outliers at occlusions by applying a left-right consistency check. For pixels that have a disparity value defined in both maps, we take the average of these values if their difference is less than 1, otherwise we mark them as invalid. For the remaining pixels, we choose the single disparity value that is available, allowing us to compensate for occlusions.

We can notice how any of the three types of labels are affected by artefacts of different nature. Finally, in the rightmost column, we show disparity maps rendered by our NS-RAFT-Stereo exploiting the proposed $\mathcal{L}_{NS}$ loss. The results demonstrate that the stereo network is able to estimate disparity maps with much greater accuracy compared to any of the proxy labels that can be obtained from the triplets, validating the effectiveness of our training pipeline.

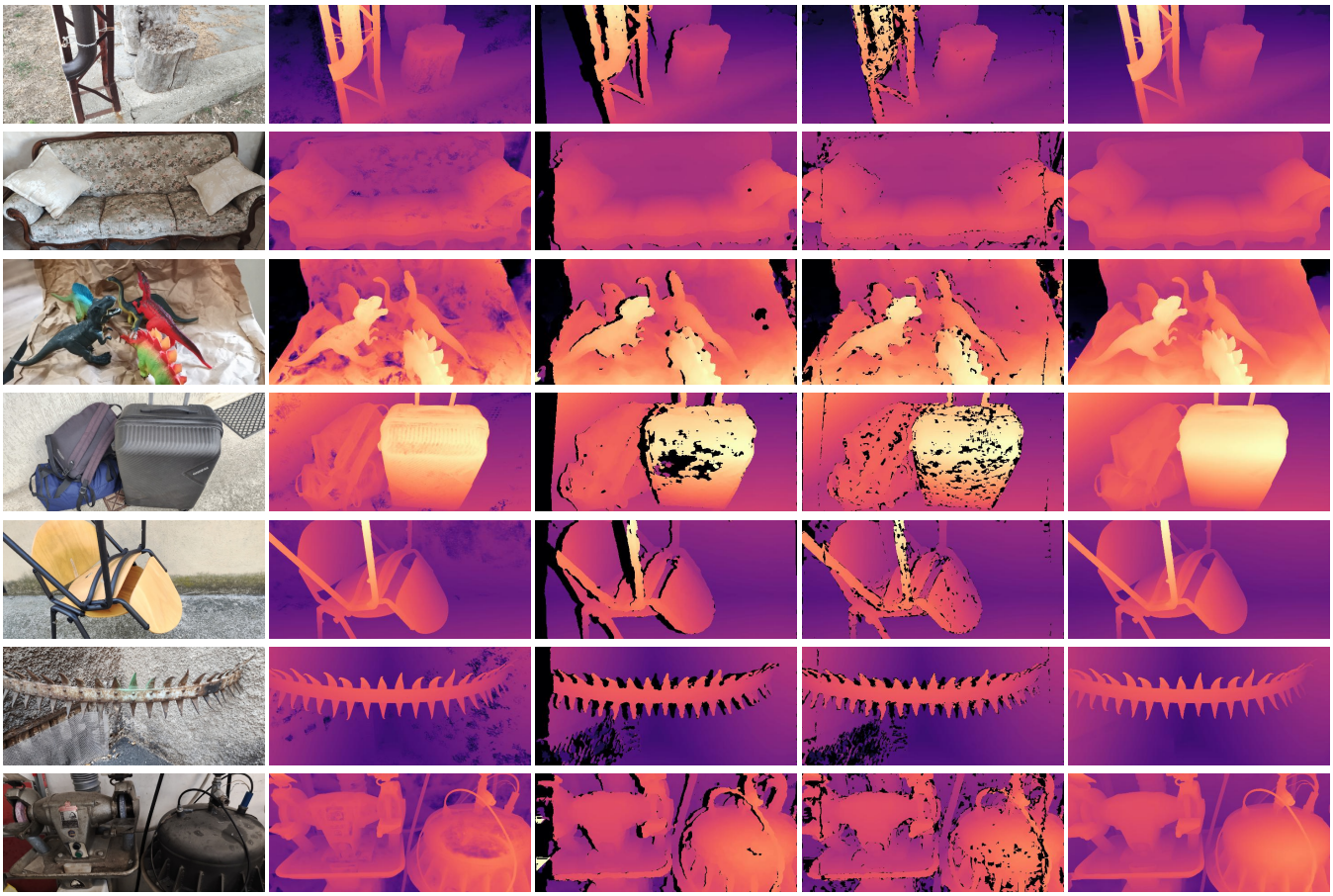

Figure 9. **Qualitative comparison – depth labels.** From left to right: RGB images from scenes in our dataset, corresponding disparity maps by Instant-NGP [6], SGM between center-right pair, SGM on image triplet and prediction by NS-RAFT-Stereo trained on our dataset.

7. Rendered Depth Labels – MfS vs NeRF

Now, we compare the two types of depth labels used in our experiments, which were reported in Table 5 of the main paper. These are the depth maps predicted by MidAs [7] and used by MfS [9], as well as the depth maps rendered by our trained NeRFs. As shown in Fig. 10, the depth maps generated by NeRFs are more detailed and contain more information, but may also contain some artifacts. In contrast, MidAs produces less detailed predictions that often fail to capture thin objects in the scene. This difference is reflected in the results obtained by models trained with these two types of depth labels, as can be seen from Table 5 in the main paper and in the qualitative comparison discussed in the following section.

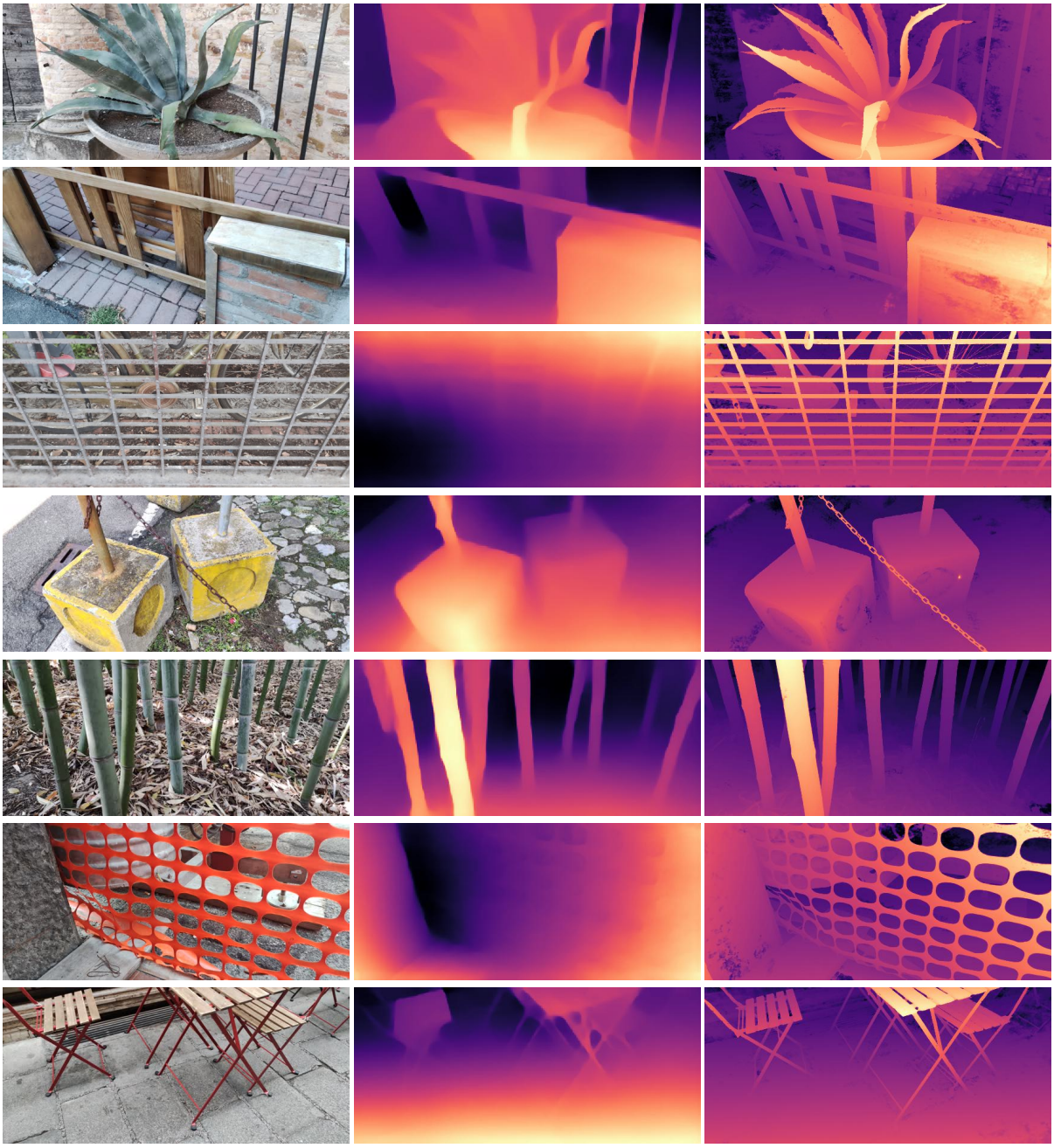

Figure 10. **Qualitative comparison – MfS vs NS labels.** From left to right: RGB images from scenes in our dataset and corresponding depth maps by MidAs [7] and Instant-NGP [6].

8. Qualitative Results – Ablation Study on Loss Function

In this section, we present qualitative results from training the RAFT-Stereo model with different configurations of our $\mathcal{L}NS$ loss, as analyzed in Table 1 of the main paper. Fig. 11 shows disparity maps estimated on Middlebury *Additional*, at half resolution. From left to right, disparity maps are obtained by RAFT-Stereo variants trained with the popular photometric loss $\mathcal{L}_\rho$ computed on stereo pairs, triplet loss $\mathcal{L}_{3\rho}$ computed by exploiting all of the three images, proxy-supervision coming from disparity labels produced by means of SGM coupled with Reversing [1], and finally by training with our full $\mathcal{L}_{NS}$ loss.

We can notice how exploiting stereo pairs alone as supervision yields many artefacts on the left of depth discontinuities – i.e., the regions resulting occluded in the right view. The triplet loss can compensate for most of these effects, although not allowing to properly learn how to deal with fine details and, often, textureless regions. Proxy labels from [1] can partially solve some of the problems, although not properly dealing with thin objects, such as the net in front of the backpack on the very last example. On the other hand, training RAFT-Stereo with $\mathcal{L}_{NS}$ constantly yields the best results, preserving fine details for thin structures and greatly improving results in low-texture regions.

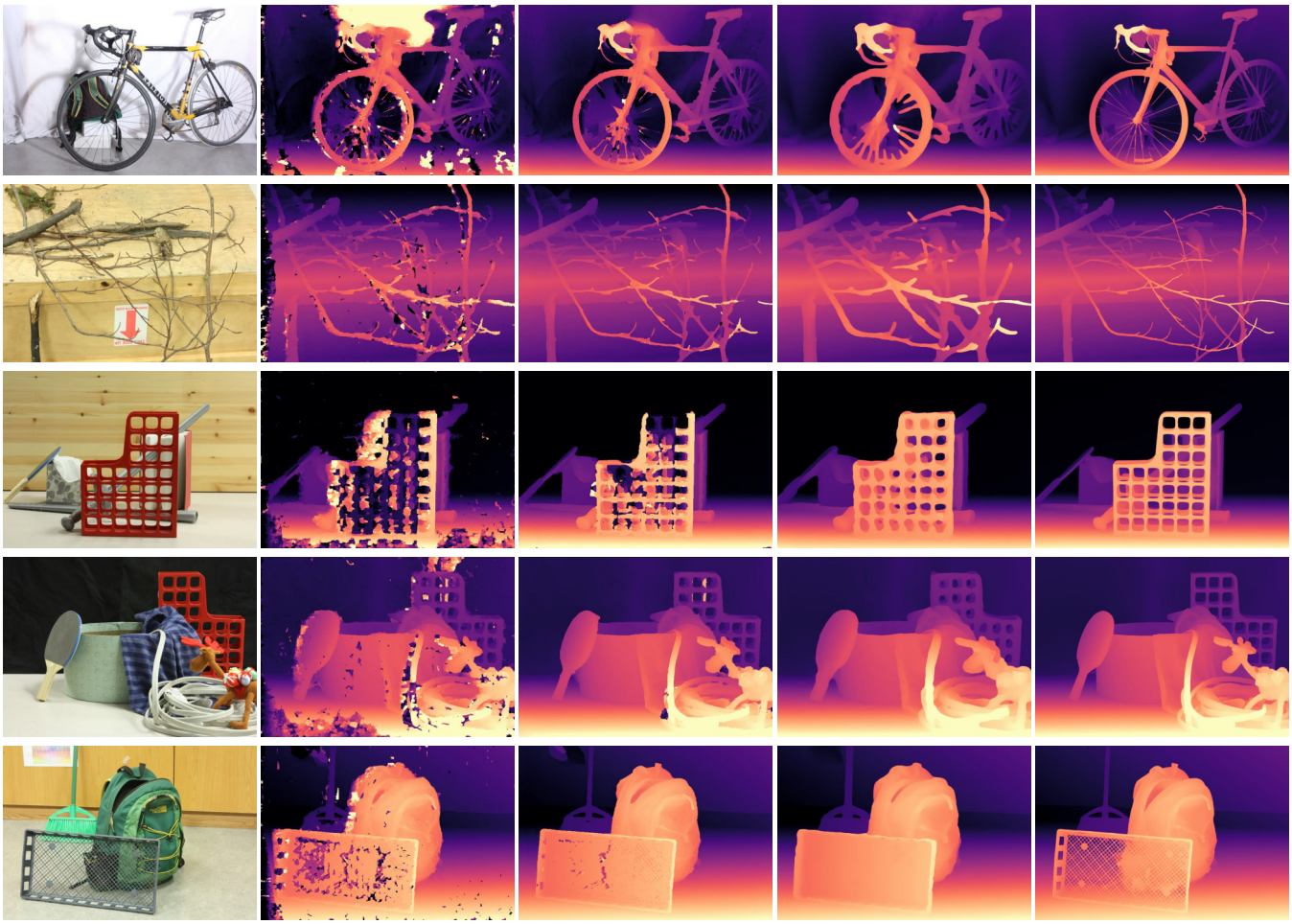

Figure 11. **Qualitative results – ablated versions of NS loss.** From left to right: reference image and disparity map obtained by training RAFT-Stereo with $\mathcal{L}_\rho$, $\mathcal{L}_{3\rho}$, SGM+Reversing labels and $\mathcal{L}_{NS}$ – respectively, (A), (C), (B'), (I) configuration from Tab. 1 of the main paper.

9. Qualitative results – MfS vs NS trained networks

In this section, a direct comparison is presented between the disparity maps predicted by RAFT-Stereo [4], CFNet [8] and PSMNet [2]. These networks were either trained with NS on the authors' dataset or with MfS on the dataset proposed in [9]. The results show that the networks trained with NS produce much more detailed and artifact-free results compared to their MfS counterparts.

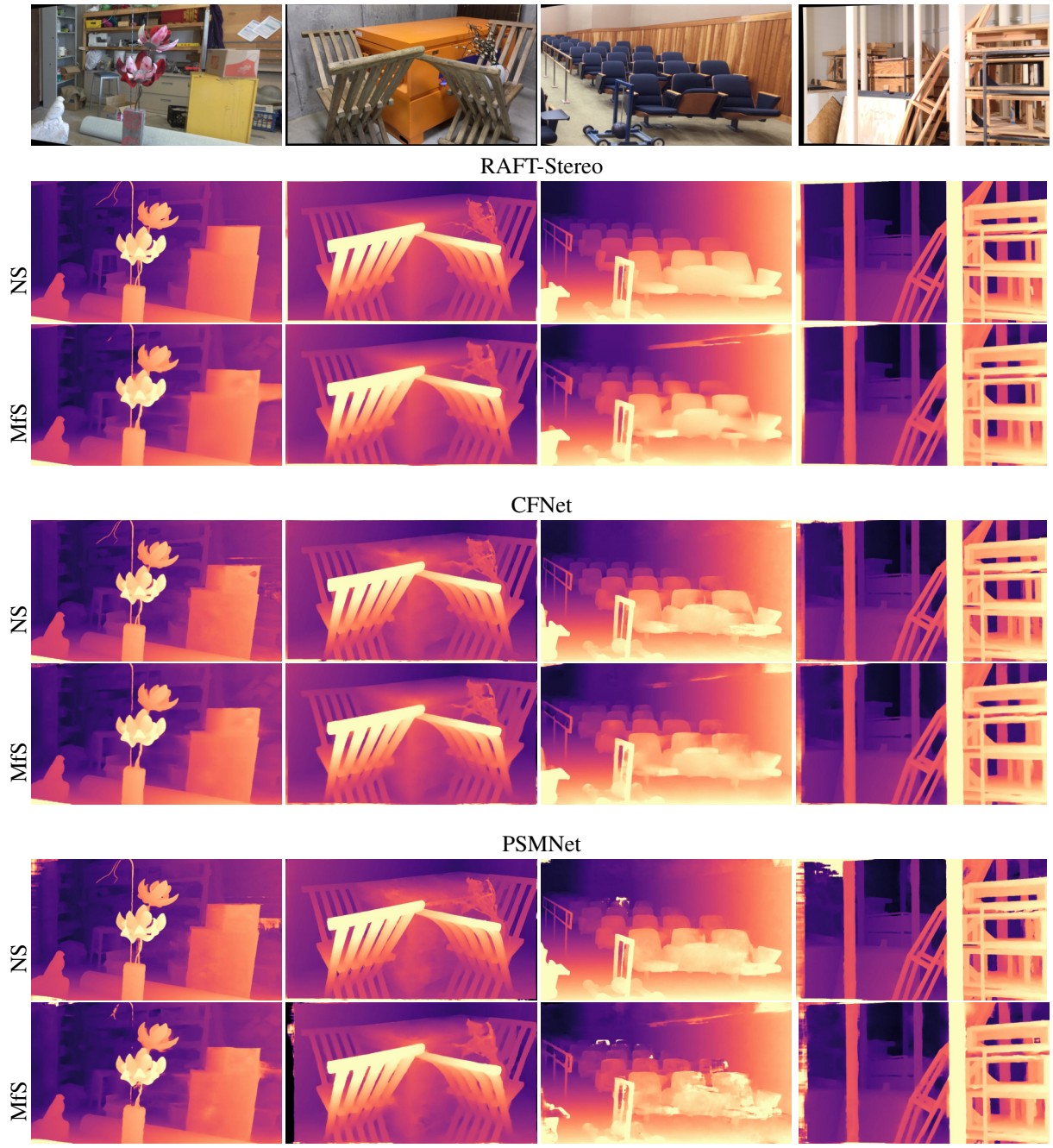

Figure 12. **Qualitative comparison – NS vs MfS networks.** From top to bottom, we show reference images from the Midd-21 dataset and two disparity maps predicted by each network, i.e. RAFT-Stereo, CFNet and PSMNet, when trained with NS or MfS respectively.

10. Qualitative Results – Comparison with PSMNet variants

Now, we report qualitative comparisons between methods built on the PSMNet backbone [2]. Fig. 13 reports results using the Midd-T dataset at half (H) resolution. We can appreciate the much higher quality of details in the disparity maps predicted by the network trained using our NeRF-supervised approach.

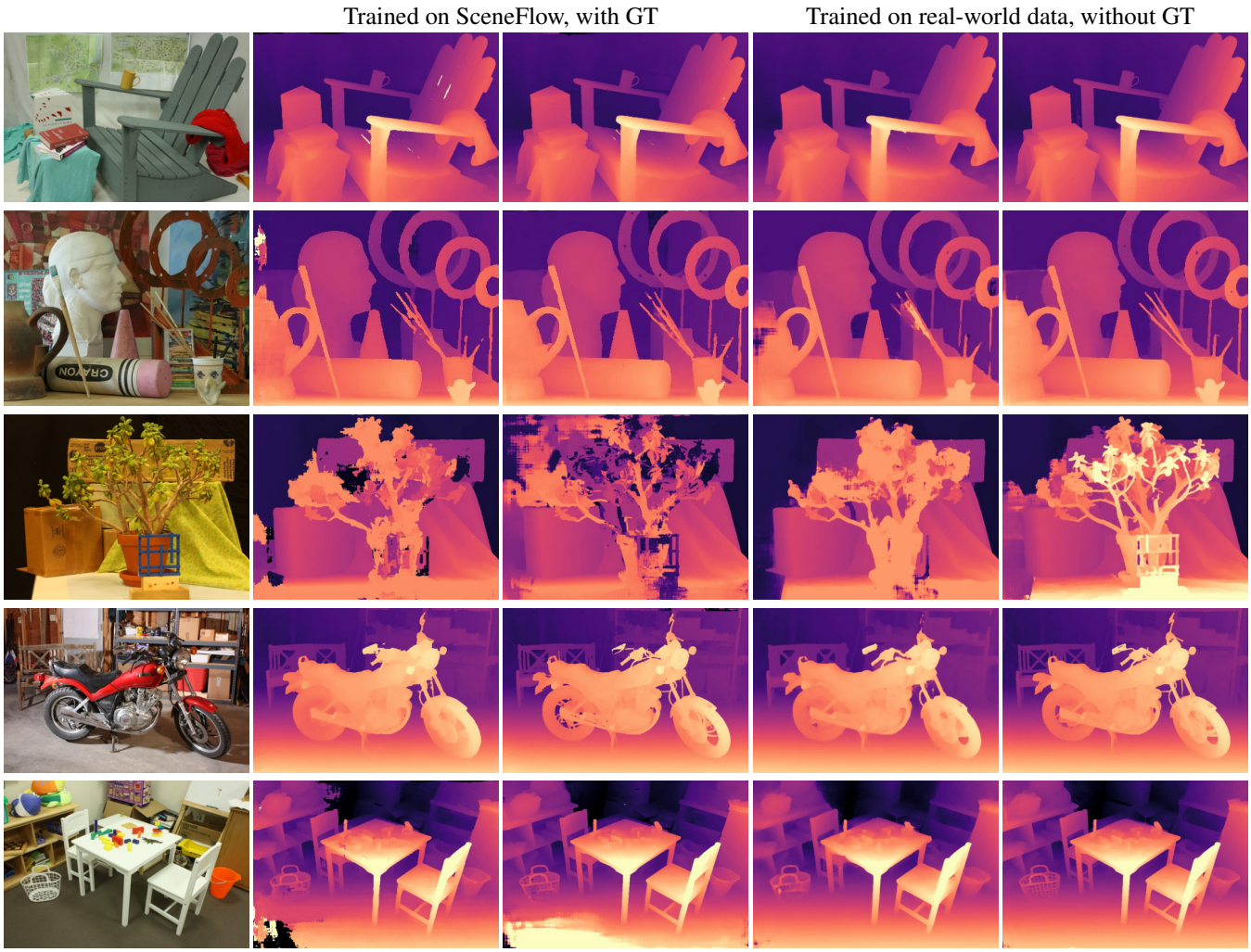

Figure 13. **Qualitative comparison – PSMNet variants.** From left to right: reference image, disparity maps predicted by networks trained on synthetic data with ground-truth (Graft-PSMNet, ITSA-PSMNet) or on real data without any ground-truth (MfS-PSMNet, NS-PSMNet).

11. Qualitative results – ETH3D and KITTI

To conclude, we report additional qualitative results achieved by NS-RAFT-Stereo. Figures 14 and 15 show disparity maps generated on the ETH3D and KITTI 2015 datasets, as part of the benchmark results listed in Table 6 of the main paper. Overall, we observe very clear and precise predictions. However, it is also notable that there are some clear failure cases, such as on car windows due to their transparency.

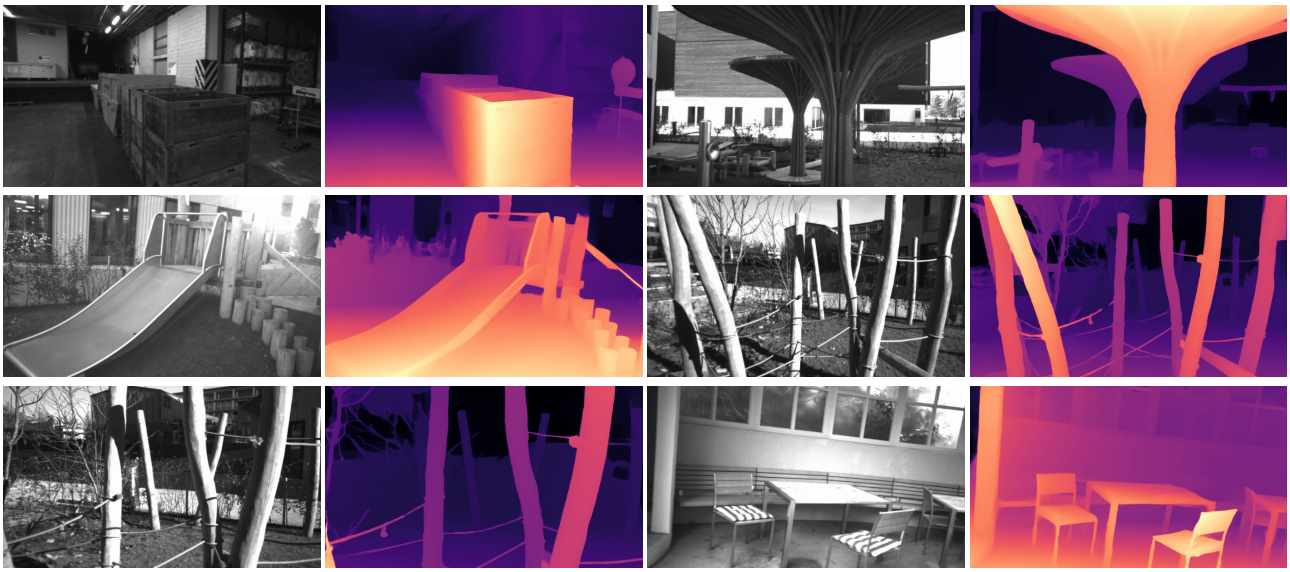

Figure 14. **Qualitative results on ETH3D.** We show reference images and disparity maps predicted by our NS-RAFT-Stereo.

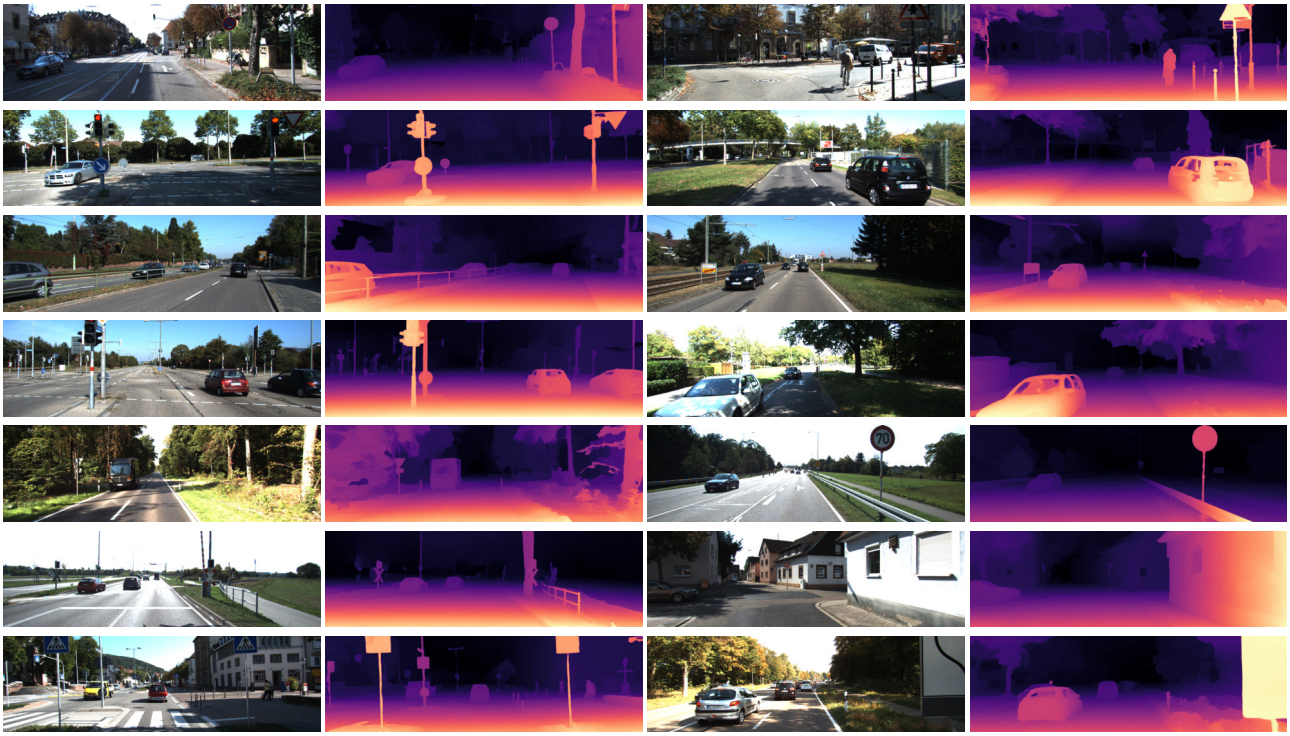

Figure 15. **Qualitative results on KITTI 2015.** We show reference images and disparity maps predicted by our NS-RAFT-Stereo.

References

[1] Filippo Aleotti, Fabio Tosi, Li Zhang, Matteo Poggi, and Stefano Mattoccia. Reversing the cycle: self-supervised deep stereo through enhanced monocular distillation. In *16th European Conference on Computer Vision (ECCV)*. Springer, 2020.

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018.

[3] WeiQin Chuah, Ruwan Tennakoon, Reza Hoseinnezhad, Alireza Bab-Hadiashar, and David Suter. Itsa: An information-theoretic approach to automatic shortcut avoidance and domain generalization in stereo matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13022–13032, 2022.

[4] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *International Conference on 3D Vision (3DV)*, 2021.

[5] Biyang Liu, Huimin Yu, and Guodong Qi. Graftnet: Towards domain generalized stereo matching with a broad-spectrum and task-oriented feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13012–13021, 2022.

[6] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv:2201.05989*, Jan. 2022.

[7] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[8] Zhelun Shen, Yuchao Dai, and Zhibo Rao. Cfnet: Cascade and fused cost volume for robust stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13906–13915, 2021.

[9] Jamie Watson, Oisin Mac Aodha, Daniyar Turmukhambetov, Gabriel J. Brostow, and Michael Firman. Learning stereo from single images. In *European Conference on Computer Vision (ECCV)*, 2020.