1. Supplementary Material

1.1. Ablation

To understand the effects of the augmentations chosen, we carry out another ablation study, first breaking down the augmentations into three categories; weather, image degradation and positional augmentation, where weather also contains time-related augmentations. We use Robust-Depth to train individual models, selecting just the augmentations from each category. The results of each of these models, when tested on both the *sunny* and *bad weather* data, are shown in Table 1. Robust-Depth uses a CNN-based architecture throughout the experiments in Tables 1 and 2. An interesting finding of this study is that positional augmentations seem to significantly improve the capability of the depth network on unaugmented images. This signifies that the positional augmentations are helping the network develop a wider variety of cues to estimate depth.

Method	Tests	Abs Rel	Sq Rel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Robust-Depth	*Sunny*	0.115	0.937	4.840	0.873	0.959	0.981
	Bad w.	**0.133**	1.115	**5.259**	**0.842**	**0.948**	**0.977**
Weather	*Sunny*	0.120	**0.889**	4.845	0.864	0.958	0.981
	Bad w.	0.145	**1.089**	5.512	0.808	0.935	0.974
Img. degradation	*Sunny*	0.123	1.000	5.049	0.860	0.954	0.979
	Bad w.	0.181	1.654	6.512	0.741	0.900	0.953
Positional aug.	*Sunny*	**0.111**	0.897	**4.740**	**0.884**	**0.961**	**0.982**
	Bad w.	0.301	3.002	9.268	0.510	0.760	0.878

Table 1. **Ablation 2:** We split the augmentations into three categories; weather, corruption and positional augmentations. Each uses pretrained ImageNet [3] weights and a data resolution of 640×192.

As would be expected, positional augmentations does not help the network with other domains or lead to greater overall robustness. Furthermore, the *bad weather* test, which contains weather and image degradation augmentations, sees the greatest benefit when training with all augmentations. Indicating that no single augmentation is most beneficial for multiple domains.

Method	Tests	Abs Rel	Sq Rel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Vertical	*Sunny*	0.120	0.995	4.949	0.868	0.957	0.980
	Bad w.	0.288	3.194	8.597	0.555	0.790	0.901
Tile	*Sunny*	**0.117**	**0.901**	**4.819**	**0.871**	**0.958**	**0.981**
	Bad w.	0.313	3.184	9.475	0.497	0.748	0.875
Rand. Erase	*Sunny*	0.119	0.985	4.953	**0.871**	0.957	0.980
	Bad w.	**0.256**	**2.368**	**7.967**	**0.589**	**0.819**	**0.921**
Scale	*Sunny*	0.119	1.040	4.937	0.869	**0.958**	**0.981**
	Bad w.	0.300	3.022	8.927	0.525	0.769	0.884

Table 2. **Ablation 3:** We further break down positional augmentations into vertical cropping, tiling cropping, random erase and scaling. Each uses pretrained ImageNet [3] weights and a data resolution of 640×192.

We now further explore each subcategory. We break positional augmentations into its components; vertical cropping, tiling, random erase and scaling. Table 2 demonstrates that each individual positional augmentation does not lead to an improved performance for depth estimation when compared to the baseline of Monodepth2 [5]. We believe these positional augmentations largely benefit from each other, and an over-reliance on each individual augmentation leads to the worsening of the depth network's standard cues.

Another interesting finding shown in Table 2, is that tile cropping augmentation gives rise to the lowest *sunny* error, suggesting that a greater local region understanding is the most beneficial feature of positional augmentation, at least for a CNN-based backbone. On top of that, random erase leads to the best robust performance for *bad weather* testing. This is because random erase aims to improve the model's capabilities with occlusion, and many weather and corruption-related augmentations would benefit from this.

1.2. Eigen Benchmark

We provide the test results from the KITTI dataset with the improved ground truth data. The improved ground truth results, shown in Table 3, look very similar to Table 2 from the main paper. Robust-Depth can maintain *sunny* depth quality while improving the quality on the *bad weather* test set. In other words, it is more robust to weather changes and image degradation while maintaining capabilities in sunny scenes. Furthermore, Robust-Depth*, uses MonoViT [16] as a backbone and shows greater overall performance for *bad weather* testing yet competitive performance for the *sunny* test.

Method	Tests	Abs Rel	Sq Rel	RMSE	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [5]	*Sunny*	0.090	0.545	3.942	0.914	0.983	0.995
	Bad w.	0.223	2.136	7.464	0.654	0.850	0.931
HR-Depth [7]	*Sunny*	0.085	0.471	3.769	0.919	0.985	0.996
	Bad w.	0.251	2.331	8.093	0.590	0.814	0.912
CADepth [14]	*Sunny*	0.080	0.450	3.649	0.927	0.986	0.996
	Bad w.	0.243	2.252	7.761	0.611	0.824	0.919
DIFFNet† [17]	*Sunny*	0.076	0.412	3.494	0.935	0.988	0.996
	Bad w.	0.183	1.542	6.842	0.717	0.888	0.949
MonoViT [16]	*Sunny*	**0.075**	**0.389**	**3.419**	**0.938**	**0.989**	**0.997**
	Bad w.	0.148	1.133	5.931	0.785	0.930	0.972
Robust-Depth	*Sunny*	0.091	0.579	3.975	0.912	0.981	0.994
	Bad w.	0.110	0.777	4.511	0.879	0.969	0.990
Robust-Depth*	*Sunny*	0.077	0.417	3.548	0.932	0.988	**0.997**
	Bad w.	**0.093**	**0.583**	**4.130**	**0.904**	**0.979**	**0.994**

Table 3. **Eigen improved ground truth test:** All tests are performed at a resolution of 640×192 and pretrained with ImageNet [3] weights.

1.3. Qualitative results

To show how our model can handle changes in domains, we also present qualitative results from some out-of-distribution data. Specifically, we will be looking at DrivingStereo [15], Foggy CityScape [11, 2] and Nuscenes-Night [1].

Figure 1 clearly demonstrates the visual improvements in our method compared with Monodepth2 and MonoViT. Our method learns to ignore fog in scenes and predict realistic depth. Methods like Monodepth2 display poor depth estimations in foggy scenes, and even current SotA models are unable to reconstruct sharp edges around objects when in the foggy domain. Robust-Depth generalises to this dataset and solves both issues without seeing this dataset.

In Figure 2, we evaluate our method in the nighttime domain. We see this is a much more challenging domain due

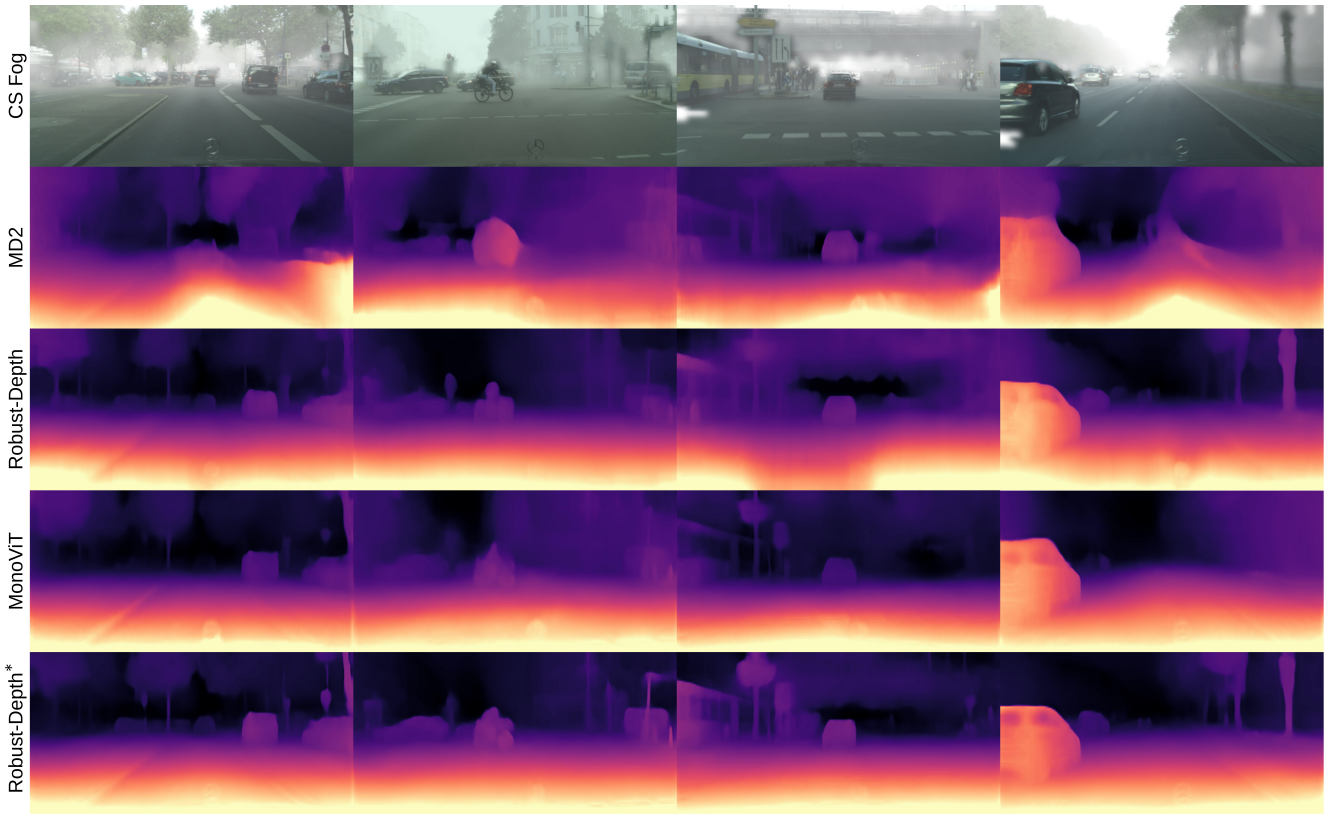

Figure 1. Demonstrating the qualitative results on the Foggy CityScape test dataset.

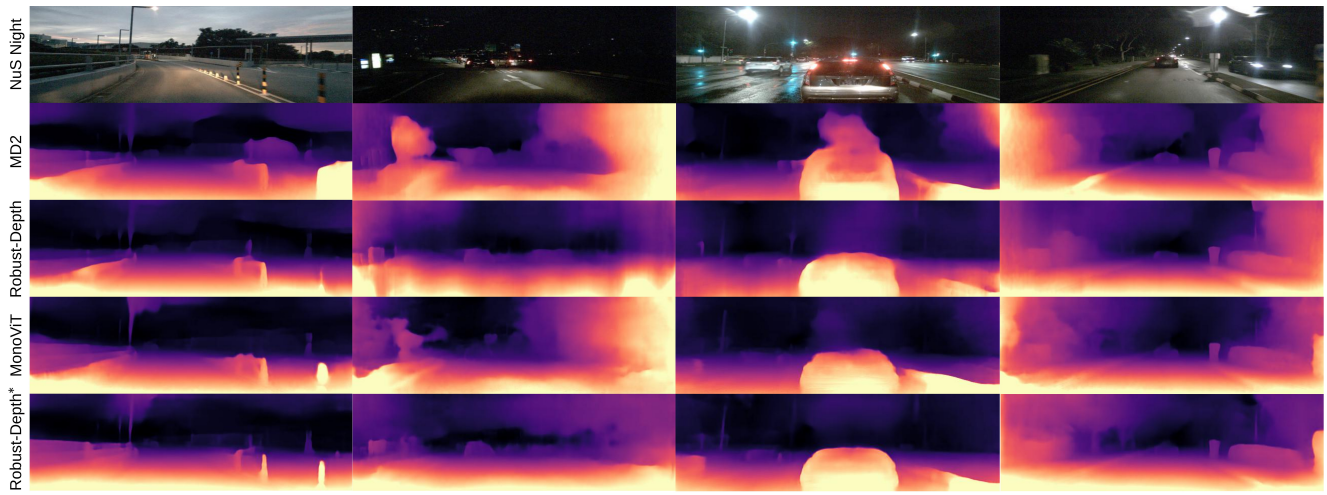

Figure 2. Demonstrating the qualitative results on the NuScenes Night test dataset.

to illuminations and lack of texture changes. Nevertheless, results indicate that our model can more clearly see objects and infer smoother surfaces.

Furthermore, we look at the DrivingStereo dataset with all four domains; sunny, rainy, foggy and cloudy in Figure 3. Clear and significant improvements can be seen when comparing Robust-Depth to Monodepth2. Also, the difference between MonoViT and our Robust-Depth* shows finer advancements in all presented qualitative results. Most improvements with this backbone involve finer details in edge definition and smoother depth maps.

Further qualitative results on the KITTI Eigen test are

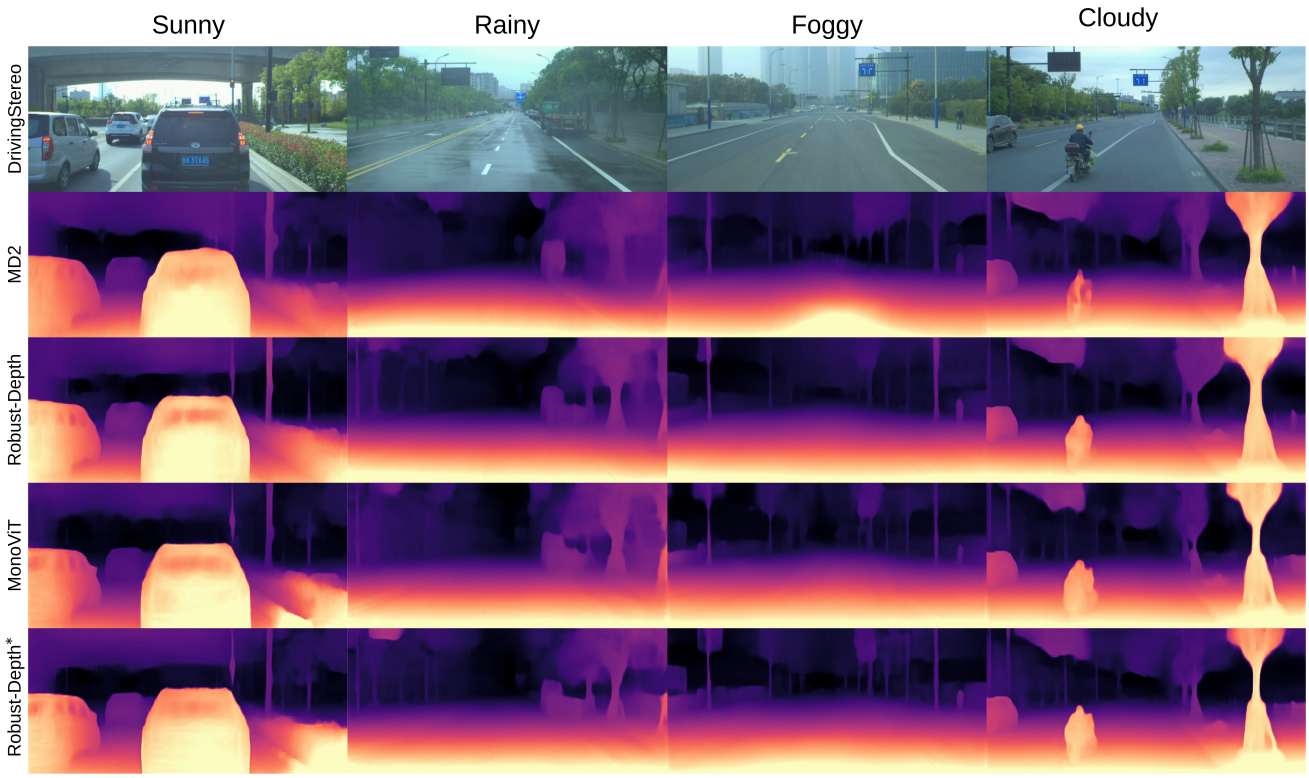

Figure 3. A demonstration of the qualitative results from the DrivingStereo test dataset.

shown in Figures 8 and 9. These figures show the collection of augmentations used during the training of our method. Here we notice significant improvements from each augmentation, especially when looking at rain-related images, compared to Monodepth2. Interestingly, Robust-Depth removes dependence on colour channels and colour overall. Also, when looking at Gaussian/impulse/shot noise, we witness remarkable improvements when using both backbones from our method.

We can also explore the understanding of these depth networks in how they hallucinate depth from a single image based on our results. We observe that:

- Colour channels are not vital in determining depth (Figure 9 row four)
- Robust-Depth does not need to rely on vertical cues (Figure 8 row one, column four)
- Robust-Depth can handle occlusions (Figure 8 rows one, column three)
- Robust-Depth can understand texture changes (Figure 8 rows two-five)

We show that we can infer depth from a very wide variety of images and not significantly negatively affect depth performance. Giving evidence that our model uses a much wider assortment of cues for monocular depth estimation.

1.4. Bi-directional pseudo-supervised depth loss:

The pseudo-supervised depth loss, as discussed in the main text, allows our augmented and unaugmented depths to pseudo-supervise each other. In this section, we will discuss the effects of each depth estimation on the final loss. In the beginning stages of learning the model capitalises on the use of two depth maps to learn depth faster, as augmented images are a view of the same image with variations in texture, shading and illumination patterns. The depth maps teach each other and result in faster learning. In Figure 4, we visualise the masks described in equations 8 and 9 from the main text, multiplied by the depth.

For the first column, $D_t \odot M_v$ is the unaugmented depth pixels that result in the lowest reprojection error. On the other hand, $\tilde{D}_t \odot M_a$ is the augmented depth pixels that result in the lowest reprojection error. We see throughout training that the unaugmented depth is moderately more accurate than the augmented depth estimation. This suggests that unaugmented depth will have a larger weight to the bi-directional depth loss throughout training. However, augmented depth will still hold a significant influence as many pixels of the augmented depth estimation lead to

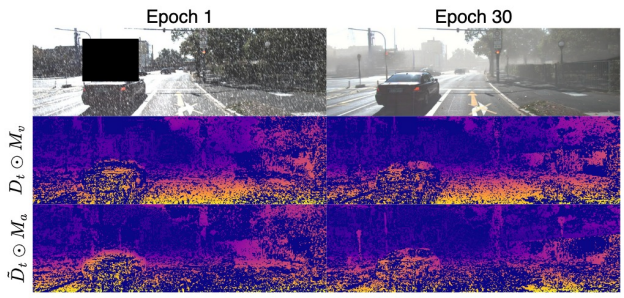

Figure 4. Depth masks $D_t \odot M_v$ and $\tilde{D}_t \odot M_a$ are both shown for epoch one and epoch thirty out of thirty epochs.

greater reprojections (row three), encouraging the use of a bi-directional depth loss.

1.5. Practicalities of vertical cropping and tiling

As discussed in the introduction, vertical and tile cropping help the depth network to remove pixel positional dependencies and learn that the lower sections of an image are not always close to the camera. In Figure 5, we see two examples of cliff-side edges, in both scenarios, it would be dangerous to assume that the pixels closest to the ground represent the road. When using our vertical/tile cropping we

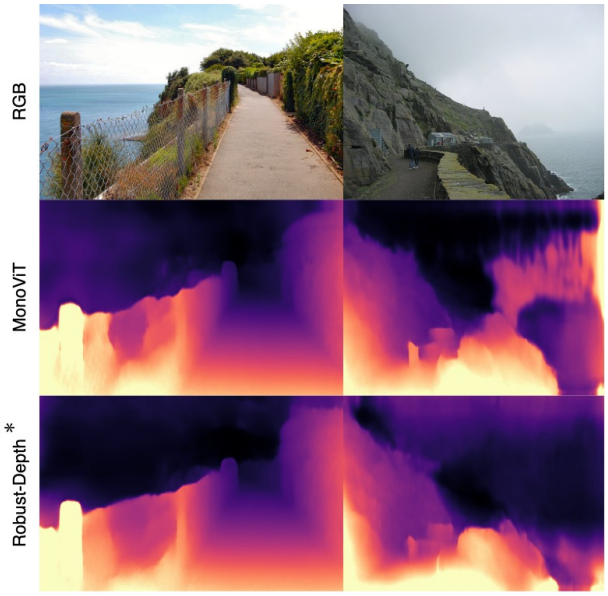

Figure 5. Image from column one [10] and image from column two [9] demonstrates that Robust-Depth* is more likely to assume large gradient changes than previous methods

can observe improvements in the understanding of depth. As the network is not over-reliant on vertical cues, it can assume that there are large gradient changes over walls.

1.6. Limitations

There are still many limitations to self-supervised monocular depth estimation. As we know from [4] self-supervised monocular depth relies on many naive cues, specifically when looking at Figure 6, we see that in nighttime scenes, our method, as well as the current state-of-the-art methods, cannot accurately detect vehicles. We believe this is because these methods use the shadows under the vehicles to determine the object's depth [4], and with nighttime scenes, this cue cannot be relied upon. When we use CoMoGAN [8] to generate nighttime augmentations, we see that, although realistic, the night scene maintains some shadow structures underneath the cars. To improve upon this flaw in the future, the method of augmentation chosen can focus on recreating even harsher nighttime scenes, removing any indication of shadows underneath the vehicle.

Moreover, even with the use of vertical cropping and tiling (section 1.5), there is still a lack of understanding of large distance change (see Figure 5). This is an area for future work.

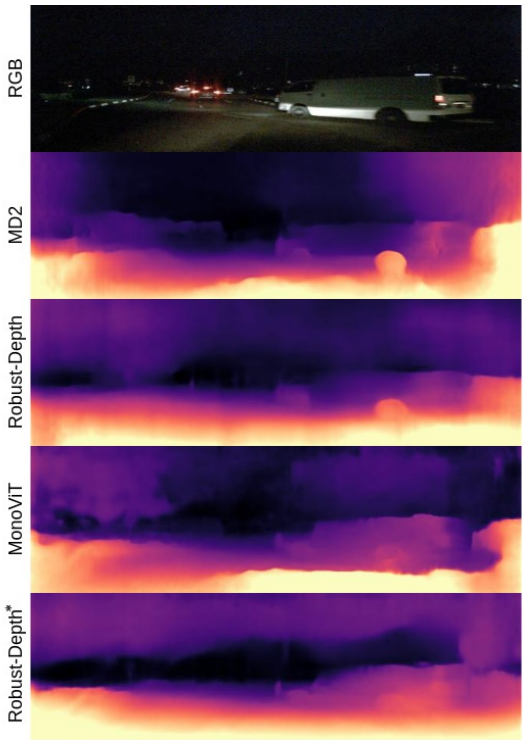

Figure 6. Vehicles disappearing in dark scenes.

Furthermore, due to the over-reliance on the KITTI dataset, we overfit on tree structures. Figure 7 shows that the lampposts are being reconstructed as trees because the KITTI dataset has many examples of trees and fewer examples of tall lampposts. A simple solution to this problem is to train on a greater variety of data.

From Figure 7, we can also see that there are reflections

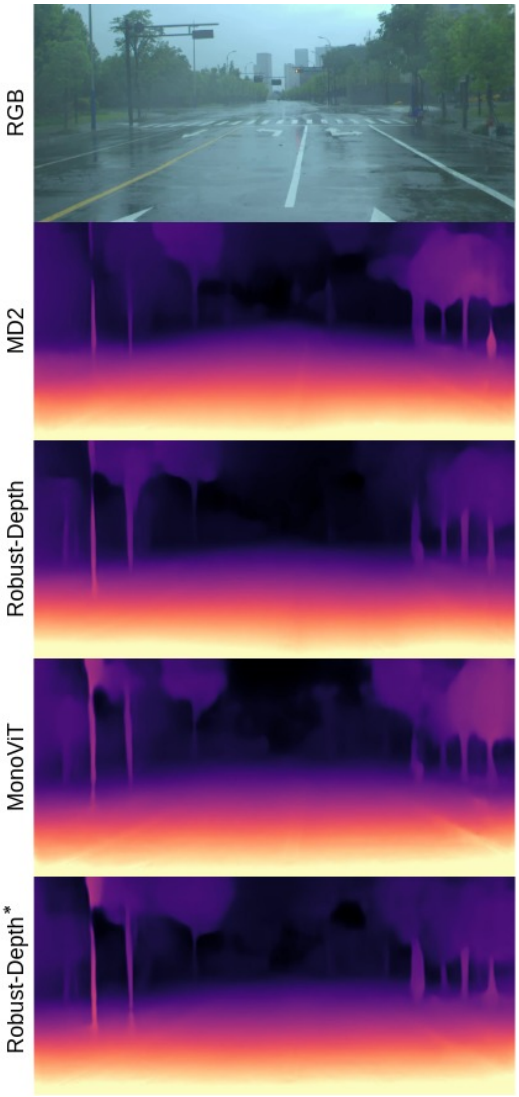

Figure 7. Trees forming out of lampposts and reflections causing errors. The depth map is generated using Robust-Depth*.

caused by the rain, leading to the depth network inferring depth on the reflected lamppost. This, although a simple-looking problem, requires a sophisticated understanding of the scene and reflections themselves. A potential solution from an augmentation perspective is to use more realistic GANs that facilitate the creation of wet scenes that contain many reflections.

1.7. Data creation

As discussed in the main text, we generate any computationally expensive augmentations before training, which speeds up the training process. However, depending on the augmentations chosen, our method could be trained end-to-end. For example, vertical cropping, tiling, random erase, colour jitter, horizontal flips and scaling are all randomised and applied during training. On the other hand, we create dusk, dawn and night version of the KITTI dataset using CoMoGANs pretrained model. Furthermore, we create a realistic rainy version of the KITTI dataset using a physics-based render [13] and a GAN trained on the NuScenes rainy data. The GAN [18] converts the KITTI images from clear to rainy, creating reflections, rain on the camera, and creating desaturated scenes. Then, we apply the physics-based render, where we specify a volume of rainfall per KITTI scene which will be described on the project's GitHub page. Also, using the physics base renderer, we create foggy scenes, which have parameters of beta set to random for training and set to 1 for all test images.

At this point, we create all combinations of the augmented data, as follows: Dusk, Dawn, Night, Rain, Fog, Rain+Fog, Rain+Dawn, Rain+Night, Rain+Dusk, Fog+Night, Fog+Dusk, Fog+Dawn, Rain+Fog+Night, Rain+Fog+Dawn and Rain+Fog+Dusk. We also add motion blur as it can negatively affect self-supervised depth estimation, as well as ground snow to represent more extreme weather. Both of these augmentations were created using the Automold GitHub page [12], and the severity of the augmentations were set to random for training but max severity for the test data. Note that the *Bad weather* test contains augmentations from weather, time of day and image degradation, but no positional augmentations. To create the corrupted data, we directly use the code provided by [6]. Here we set the severity of each corrupted image to the maximum for testing data, but random for the training data. Finally, we create simple greyscale, red, green and blue components of the images. All of these augmentations are set to have a uniform distribution of being selected, without replacement, so each augmentation is sampled equally during training. The augmented data represents half of the data seen during training and each augmentation has a $1/n$ chance of being selected, where n is the number of augmentations chosen. All the information provided should aid with the reproducibility of this work and potential further development. We highlighted in Figure 8 and Figure 9 a multitude of cases, that clearly demonstrate the improvements of our model (Robust-Depth) over Monodepth2 (MD2).

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceed-*

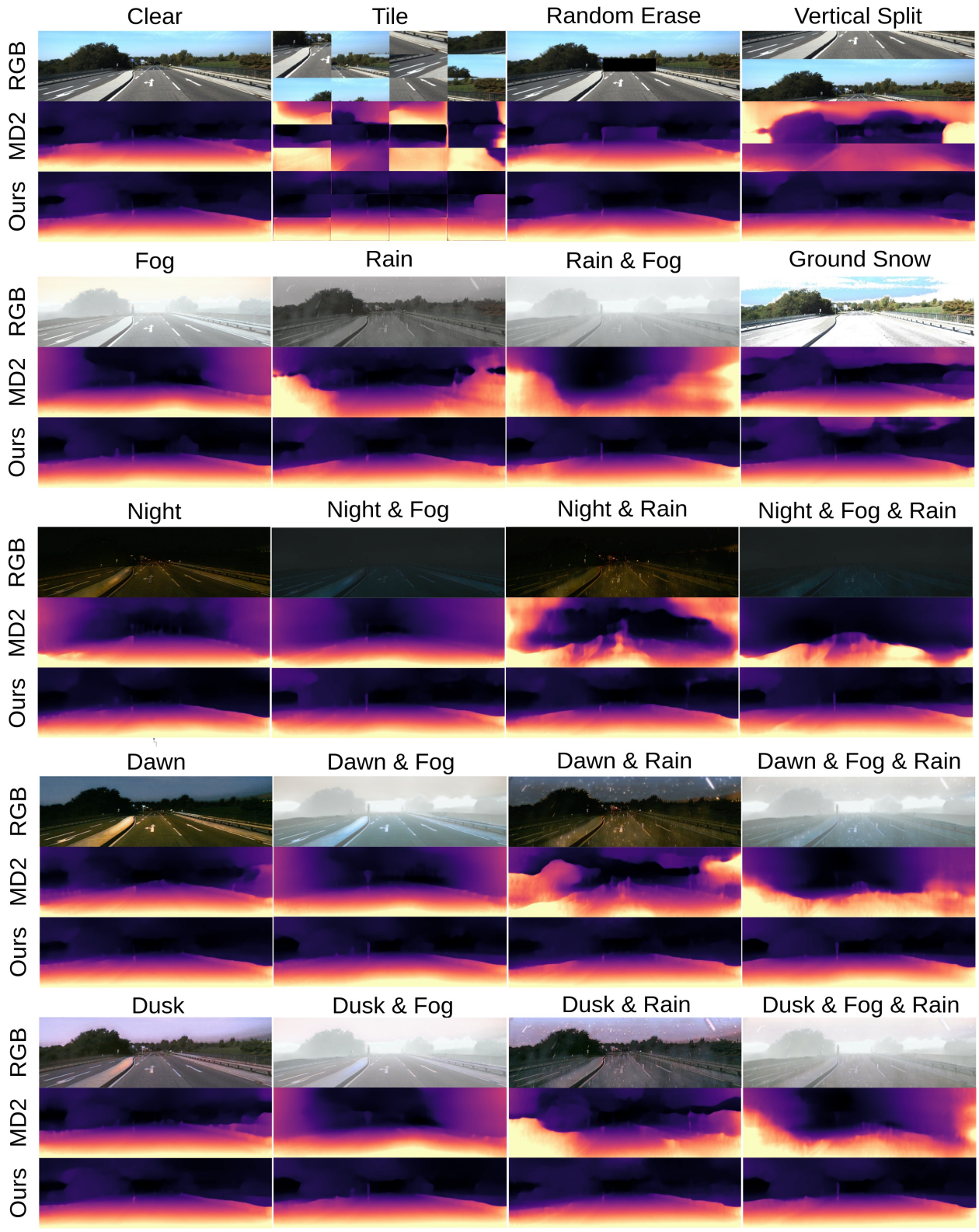

Figure 8. We demonstrate a majority of the weather-related and positional augmentations. MD2 represents depth estimations using Monodepth2, and "Ours" is Robust-Depth. All images are from the KITTI Eigen test data.

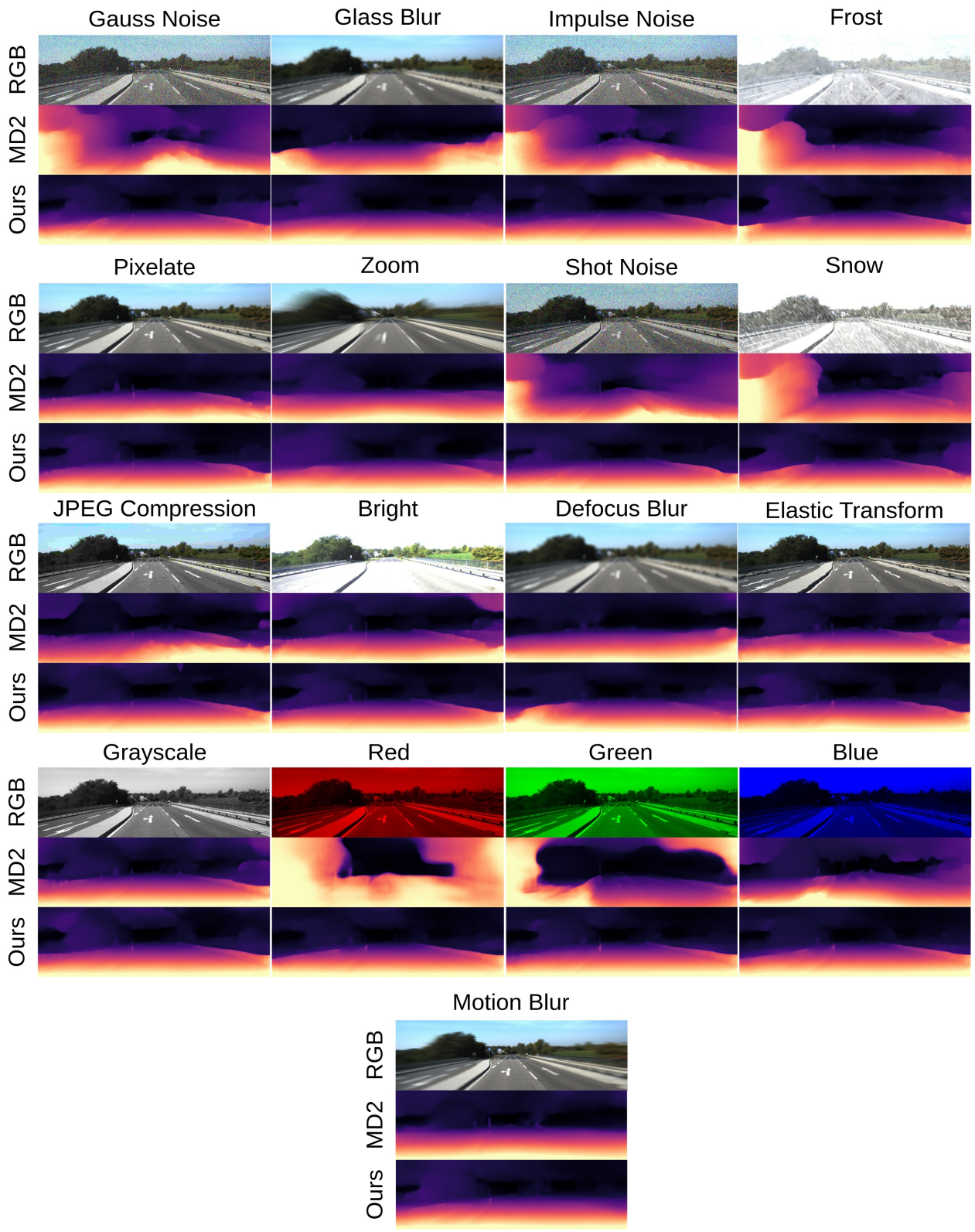

Figure 9. Corruptions, colour channels, greyscale and motion blur augmentations are shown. MD2 represents depth estimations using Monodepth2, and "Ours" is Robust-Depth. All images are from the KITTI Eigen test data.

ings of the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016. 1

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[4] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2183–2191, 2019. 4

[5] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1

[6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 5

[7] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2294–2301, 2021. 1

[8] Fabio Pizzati, Pietro Cerri, and Raoul de Charette. Comogan: continuous model-guided image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14288–14298, 2021. 4

[9] Rob Burke. Lighthouse road, skellig michael, 2004. [Online; accessed March 13, 2023]. 4

[10] Ronald Saunders. Cliff edge walk. sandown. isle of wight uk, 2010. [Online; accessed March 13, 2023]. 4

[11] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Sep 2018. 1

[12] Ujjwal Saxena and Rohan Giriraj. Automold. https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library. Accessed: 2022-12-20. 5

[13] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul De Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 129(2):341–360, 2021. 5

[14] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 464–473. IEEE, 2021. 1

[15] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 899–908, 2019. 1

[16] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. Monovit: Self-supervised monocular depth estimation with a vision transformer. *arXiv preprint arXiv:2208.03543*, 2022. 1

[17] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. *arXiv preprint arXiv:2110.09482*, 2021. 1

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 5