Monocular Depth Estimation with Virtual-view Supervision

Nikolas Hars[1], Casimir Feldmann[1], Rong Zou[1], Kirat Virmani[1], Zuria Bauer[1], Martin R. Oswald[1, 2]
[1]ETH Zurich, [2]University of Amsterdam
{nihars, cfeldmann, ronzou, kvirmani, zbauer, moswald}@ethz.ch

Abstract

Data-driven methods have gained popularity for addressing the monocular depth estimation (MDE) task. Among them, supervised methods, which yield state-of-the-art (SOTA) results, require large amounts of labeled training data, posing challenges in terms of costly ground-truth label collection. This work aims to enhance the performance of existing supervised learning-based MDE methods by generating a substantial number of virtual views as additional supervision signals, circumventing the laborious and time-consuming process of collecting extra data. We propose leveraging the capabilities of Neural Implicit Surface Reconstruction (NISR) techniques to augment an existing limited-scale dataset by generating novel scene perspectives and corresponding high-quality depth maps. Experimental results demonstrate that the augmented dataset significantly boosts the performance of supervised-learning MDE networks. This highlights the potential of the NISR approach for scaling small-scale datasets and provides a valuable solution to further improve the efficacy of existing supervised MDE models without the need for an expensive label collection process.

1. Introduction

Depth estimation is a critical problem in computer vision, playing a pivotal role in various applications such as autonomous driving, robotics, and augmented reality. Particularly in scenarios where computational resources, memory, or installation space are limited, the task of Monocular Depth Estimation (MDE) becomes essential. MDE involves predicting the depth of a 3D point using only a single viewpoint, presenting a challenge due to scale ambiguity. The ill-posed nature of the task of MDE leads to an infinite number of potential solutions.

In recent years, data-driven methods have emerged as a prominent approach to solving MDE. However, obtaining high-quality ground truth depth data, such as through LiDAR, is prohibitively expensive. While self-supervised MDE methods capable of utilizing large-scale unlabeled datasets have gained attention, they still fall short of supervised methods in terms of performance. The latter demands abundant labeled data with reliable supervisory depth maps, creating a pressing need to generate a substantial amount of reliable image-depth pairs from limited data, thereby improving the efficacy of supervised learning methods and circumventing the costly process of collecting ground-truth depth labels.

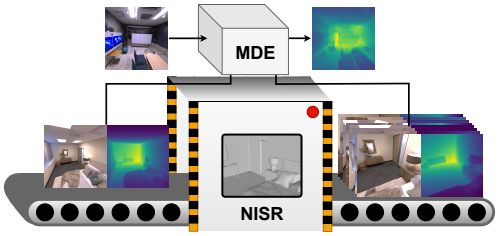

Figure 1: **Next-level data augmentation.** We reconstruct scenes from existing data and render novel views to augment the original small-scale dataset for MDE training.

To address this problem, we propose to exploit the capabilities of Neural Implicit Surface Reconstruction (NISR) techniques. NISR methods, such as Neural Radiance Field (NeRF) [18] and implicit Signed Distance Functions (SDFs) [11, 31], have recently gained popularity for multi-view 3D reconstruction. By leveraging a small number of views, these methods enable the generation of novel scene perspectives and corresponding high-quality depth maps. Therefore, NISR methods offer a viable solution for augmenting a limited-scale dataset by producing large-scale image-depth pairs from the original data.

Our experimental results demonstrate that this approach substantially enhances the performance of supervised-learning MDE networks. After augmenting the original dataset with the virtual views generated by MonoSDF [31], the extended dataset enables enhanced supervision in MDE. Consequently, the MDE model DeepLabv3+ [6] trained on the augmented dataset demonstrates notable improvements in depth estimation accuracy compared to the performance exhibited by an identical model trained solely on the original small-scale dataset.

Overall, our **contributions** can be summarized as follows:

- We propose leveraging NISR methods to augment a

limited-scale dataset via scene reconstruction and virtual view-depth pair generation for supervised training of data-driven MDE networks.

- We conduct extensive experiments and present experimental results that validate the effectiveness of our approach, showcasing significant improvements in the performance of supervised-learning MDE networks.

2. Related Work

Monocular Depth Estimation (MDE). Data-driven approaches for MDE can be broadly classified into three categories: supervised, self-supervised, and unsupervised methods. In this paper, we focus on end-to-end supervised methods, which require high-quality pixel-wise depth labels for training and mapping a single image to its depth map at inference time. One of the pioneering endeavors in this domain is represented by [10], where Eigen et al. employed Convolutional Neural Networks (CNNs) to regress depth values directly. Their approach involved making an initial global depth prediction by a CNN and then applying another to refine the coarse prediction with local information. Additionally, they employed a scale-invariant error (SI log loss) for training, ensuring the focus on spatial relations rather than general scale in the depth prediction. Following this seminar work, subsequent studies have explored novel architectural designs, ranging from deep residual networks [15] to transformers [32] to improve the modeling capabilities and performance of MDE models. Researchers have also introduced alternative loss functions, such as the reverse Huber (berHu) loss [15], or adaptive combinations of multiple losses [16]. Some approaches have tackled the direct regression problem by translating it into ordinal regression [3] or classification [4] and utilizing ordinal regression loss or classification loss to facilitate training. Additionally, the concept of multi-task learning has been leveraged to jointly learn the MDE task alongside other mutually beneficial tasks, such as normal prediction and semantic segmentation [22, 34]. Among this type of work, DeepLabv3+ [6] is a prominent and successful model that can produce dense prediction by utilizing an encoder-decoder architecture with an atrous spatial pyramid pooling (ASPP) module. While originally designed for semantic segmentation, we use it for MDE to exploit its ability to capture multi-scale contextual information provided by the ASPP module.

Dataset Augmentation for MDE. To expand the available training data for supervised learning, data augmentation is a widely-used and effective approach. Generic data augmentation techniques, including parameter-free methods such as [33, 35], and learning-based ones such as [7], have been widely adopted across various computer vision tasks to augment the dataset size. Recently, there have emerged several methods specifically tailored for MDE tasks, including data grafting [20], (Vertical) CutDepth [14], and CutFlip [24]. However, our proposed method fundamentally distinguishes itself from these approaches in several critical aspects. Firstly, while the aforementioned methods do not introduce any new information in the augmented data, as it is directly derived from and hence overlaps with the original data, our method generates novel data encompassing previously unobserved information. This fundamental difference leads to a distinctive characteristic of our method, enabling the incorporation of fresh perspectives into the MDE models. Secondly, while the primary aim of the existing data augmentation methods for MDE is to enhance the networks' ability to learn diverse cues and mitigate overfitting, our method tackles the issue of limited availability of high-quality image-depth pairs. To the best of our knowledge, we are the first to exploit NISR techniques to augment small-scale datasets for MDE, leading to improved MDE performance by integrating novel virtual view-depth pairs.

Neural Implicit Surface Reconstruction (NISR). NISR has recently emerged as a potent paradigm for the reconstruction of 3D surfaces from multi-view images. Pertaining to this field, previous studies can be broadly classified into surface rendering-based and volume rendering-based methods. Surface rendering-based methods such as [13] assume that the color of a ray depends solely on its intersection with the scene geometry, thereby confining the backpropagated gradient to a localized region around the intersection. Typically, these methods necessitate object masks for supervision. NeRF [18] is an influential technique that synthesizes novel views by employing volume rendering to accumulate the colors as well as densities of sampled points along each ray into an image. Although NeRF has yielded exceptional outcomes in novel view synthesis, extracting high-quality surfaces and recovering nice 3D geometry from its output remains a problem due to the absence of surface constraints within its geometry representation. Inspired by NeRF, UNISURF [19] leverages volume rendering to learn an implicit surface. By incorporating an implicit surface representation into the volume rendering framework, UNISURF achieves 3D geometry reconstruction without relying on object masks, presenting a significant advancement in the field. In contrast to UNISURF, which adopts occupancy values for representing the 3D surface, VolSDF [29] and NeuS [27] utilize an alternative approach by representing the surface as a signed distance function (SDF). This alternative representation leads to enhanced reconstruction accuracy. Further improvements were introduced by incorporating geometric cues in the process of reconstruction, NeuRIS [26] propose to use normal cues for indoor scene reconstruction; MonoSDF [31] incorporates depth and normal priors to refine implicit surface reconstruction method, enabling them to attain significantly

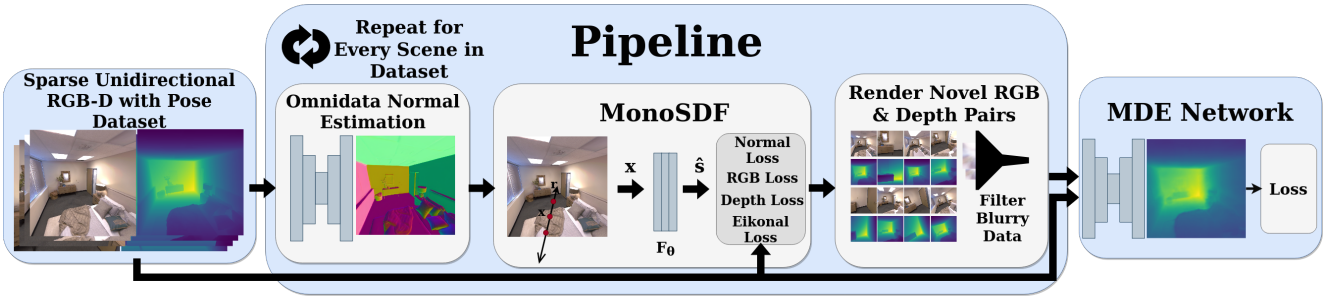

Figure 2: **Overview of the MDE with Virtual-view Supervision pipeline.** We use Replica dataset [25] as mentioned in Section 2, but any appropriate dataset can be used. Additionally, this pipeline can be applied to improve any MDE network. We use DeepLabv3+ [6] for our experiments as mentioned in Section 2.

more detailed reconstructions while reducing optimization time. In this paper, we leverage the MonoSDF framework to generate novel data for MDE, capitalizing on its ability to produce highly detailed and accurate surface reconstructions.

3. Methods

We propose a novel dataset augmentation technique for the task of MDE that leverages the 3D reconstruction capability of MonoSDF [31], which enables it to generate novel RGB-D images given multiple input poses and corresponding RGB-D images. We evaluate our proposed approach on the digitized indoor dataset Replica [25], to benefit from the perfect supervision for MonoSDF and our chosen off-the-shelf MDE networks DeepLabv3+ [6] and U-Net [23].

3.1. Proposed Pipeline

Our proposed pipeline works as follows: First, we must define trajectories for each scene in the Replica [25] dataset. These trajectories are then rendered using a customized version of the *ReplicaRenderer* provided by Replica. We use the views from the rendered trajectories to form a small-scale dataset. Following this step, we train MonoSDF [31] on each scene separately and, after convergence, render virtual RGB-D images using MonoSDF and StudioSDF [30]. We filter out inferior virtual RGB-D images using hand-crafted filtering algorithms. Together with the original data, the filtered novel views form the augmented large-scale dataset. Finally, we train a MDE network on both the original dataset and the augmented dataset for comparison. Figure 2 shows this pipeline visually.

3.2. Dataset Generation

In contrast to other 3D datasets [8, 5], the Replica dataset [25] has to be rendered to obtain RGB-D images. It is integrated into Habitat-Sim [17], which enables the simulation and rendering of Replica scenes, but unfortunately, it is impossible to install Habitat-Sim on the ETH cluster. Due to this issue, we utilize the provided ReplicaSDK [25], which contains the tools *ReplicaRenderer* and *ReplicaViewer* to render RGB-D images. Since the *ReplicaRenderer* as a standalone program does not provide an interface to traverse Replica scenes on customized trajectories, we modify its C++ source code to enable this functionality. We enable the traversal along linearly interpolated trajectories between sequences of two poses. We query the poses for the traversal through a slightly modified version of the *ReplicaViewer*, which features a new button on its interface that saves the poses of the current viewpoint in a CSV file. This file is automatically parsed by the modified *ReplicaRenderer*.
For the generation of our small-scale dataset, used for the training of MonoSDF and our MDE network, we render 50 RGB-D images along a total of 40 unique, linear trajectories from 14 scenes, which makes 2000 training images. We restrict ourselves to straight paths to evaluate MonoSDF in a problematic, more challenging setting. Within the limitations of available computational resources, we vary the ratio between the volume of augmented data and the volume of original data to examine the influence of different augmentation levels on the performance of the MDE network. We withhold 3 scenes from the Replica dataset and use 2 of them to generate a validation set with 1234 RGB-D images and 1 scene for our test set with 273 RGB-D images.

3.3. Novel View Synthesis with MonoSDF

To render novel views of a scene, we first have to reconstruct it. As mentioned in Section 2, out of the many different techniques that can do so, MonoSDF was chosen for its SOTA reconstruction quality. The depth and normal cues necessary for the MonoSDF pipeline are predicted using the pre-trained V2 DPT-based Omnidata [9] normal and depth estimation networks. For our purposes, only the normals are estimated as we use the ground truth depth data instead of the depth prediction. This has two benefits: depth reconstruction at the correct scaling and overall improved depth detail reconstruction quality. Since MonoSDF has been integrated into the SDF Studio [30] framework, it is necessary to convert the dataset into the correct format to be parsed. Modifying the provided example script, image

poses are converted from OpenGL to OpenCV format, centered, and scaled.

$$\mathcal{L} = \mathcal{L}_{RGB} + \lambda_1 \mathcal{L}_{Eikonal} + \lambda_2 \mathcal{L}_{\substack{Sensor \\ Depth}} + \lambda_3 \mathcal{L}_{Normal} \quad (1)$$

For each scene, a MonoSDF model is trained until convergence with the loss function given above, with $\lambda_1, \lambda_2, \lambda_3$ set to $0.1, 0.1, 0.05$ respectively. To generate novel RGB-D images, we must determine the novel poses from which these are rendered. Empirically we determined that image quality quickly deteriorates as we move further away from the given training poses associated with the images used to train the network. Hence we seek to disturb the training poses only slightly, finding a balance between the novelty of the viewpoint and satisfactory image quality. Based on the data size multiplication factor M, we generate M randomly rotated, and M randomly translated poses for each of the N training poses. This means we generate a total of $2MN$ novel poses. To get the novel rotated poses, we repeat the following procedure M times for every training pose: Pick one axis (x,y,z) at random and then draw an angle from the uniform distribution between -20 and 20 degrees. Similarly, to get the novel translated pose, we randomly pick an axis, draw from the uniform distribution between -0.05 and 0.05, and then translate the training pose in the chosen axis by this amount.

We can generate RGB-D images instead of a video by modifying the existing SDF Studio render script. Each RGB-D image takes around 20s to render, hence with 2,000 training images and $M = 10$, the rendering process alone takes around 9.3 days on a single RTX 3090. Some of these novel views will still have unsatisfactory reconstruction results, hence the need for a filtering algorithm. Following the findings in [1], we can detect blurred images by first convolving the image with the Laplacian kernel and then taking the variance of the response. The images with variance below an experimentally determined threshold of 20 are removed. Hereafter, we also filter out images based on the depth maps. If the average depth is below as preset threshold of 0.5 meters, the RGB-D image is also removed.

3.4. Monocular Depth Estimation Architecture

We use DeepLabv3+ [6] and U-Net [23], two off-the-shelf Semantic Segmentation networks. Since U-Net has been shown to perform well in MDE when paired with a specialized training scheme [2], we hypothesize that DeepLabv3+ can be a good choice as well.

We choose both networks due to their low computational complexity compared to current SOTA MDE networks such as [21, 28], while still performing well enough to draw possible conclusions about the effectiveness of our novel augmentation method.

We train the MDE network with the Scale Invariant Log Loss ($\mathcal{L}_{SI_{Log}}$) [10], which is widely accepted as a common loss function for MDE as well as the reverse Huber loss ($\mathcal{L}_{berHu}$) [15], which behaves like an $\mathcal{L}_1$ loss for smaller errors and like an $\mathcal{L}_2$ loss for higher errors.

3.5. Evaluation Metrics for MDE

We utilize the MDE evaluation metrics used in [2]. The metrics are defined as follows given prediction $\hat{y}_i$ and ground truth value y_i:

Relative Error (REL): $\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{y_i}$

Squared Relative Error (SQ. REL): $\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|^2}{y_i}$

Root Mean Squared Error (RMSE): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}$

Log RMSE: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log y_i - \log \hat{y}_i)^2}$

Threshold Accuracy (δ_j): $\max(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i}) = \delta < 1.25^j$

4. Experiments

4.1. Training and Evaluation of MonoSDF

Scene training image allocation varied between 50 images for small scenes such as "Room 1" and up to 650 for large scenes like "Apartment 0". All scenes were trained for 200,000 steps except "Apartment 0" which was trained for 400,000 steps, taking 11 and 22 hours per scene respectively on an RTX 3090 GPU. The network was not able to converge on "Office 1" and "Office 4". After passing these images through the filter, around 12% of the results are filtered out.

Novel view samples can be seen in Figure 4 compared with their Replica ground truth rendering. Produced RGB-D images are visually appealing. However, small details like the farthest chair in "Room 2" or the whiteboard in "Office 3" are lost in the depth map. We hypothesize that this is a result of the network unifying surfaces with the same normal values. Furthermore, scenes with undefined regions, such as the ceiling in "FRL apartment" lead to bad depth map reconstructions. We address this issue in a later section with a masking technique.

4.2. Training of the MDE Networks

In this section, we evaluate our novel dataset augmentation method in multiple ways. We first evaluate the performance of the two chosen off-the-shelf MDE models described in Section 3.4 against each other and test the influence of different loss functions. We then evaluate the effect of different ratios of virtual- to real-view RGB-D images, and introduce different ways the augmented data can be used in the training procedure.

Comparison of MDE Networks and Loss Functions

Here we train both networks using only the original RGB-D images to isolate the influence that the model and loss

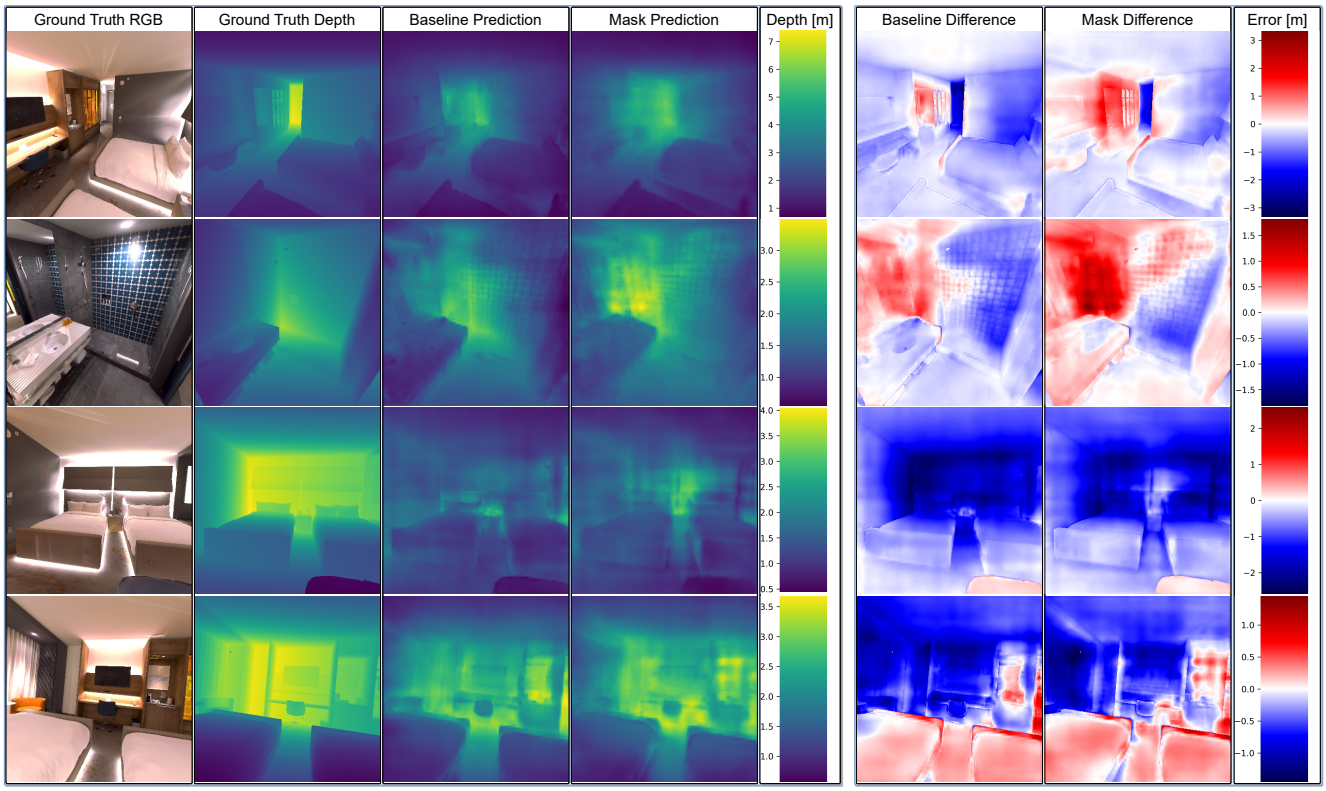

Figure 3: **Qualitative depth prediction results.** Ground truth, baseline, and 400% mask depth prediction performance are compared on the test set. Despite the baseline depth prediction results exhibit a slightly sharper appearance, there is a noticeable improvement in the depth values for walls.

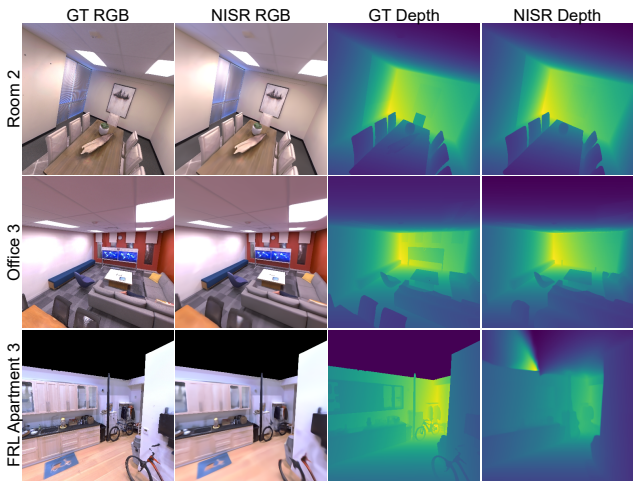

Figure 4: **Novel viewpoint quality comparison.** Although missing fine-grained details in the depth maps, the overall quality of both the RGB and depth maps is high and reliable.

function choice make. For a fair comparison, we train both networks for 50,000 steps with batch size 8. We choose this number of steps as the performance of both networks saturates already before step 50,000, and the networks start to overfit slightly, which is expected with only 2,000 training images.

The results of our analysis show that DeepLabv3+ [6] reaches significantly better values for all measured performance metrics as shown in Table 2. This is an expected result as DeepLabv3+ contains a pre-trained ResNet50 [12] backbone and as such has more than 2.6 times the parameters as U-Net. The results also show that $\mathcal{L}_{berHu}$ [15] is the better choice as it reaches better values independent of the model architecture.
Based on the results of these experiments, we decide only to consider DeepLabv3+ with $\mathcal{L}_{berHu}$ in the following sections.

Comparing Techniques for Utilizing Virtual-View Data

This section compares different methods to incorporate the virtual-view RGB-D images into the training procedure. Quantitative results can be seen in Table. 1. We first test the influence that the ratio of original data to augmented data has. Here the given percentage refers to how much additional synthetic data composed of novel viewpoints have been added compared to the original dataset size. We note that even a 40% addition of novel views leads to a significant jump in threshold accuracy 1 (δ_1). A general trend is that using virtual views is beneficial in almost every metric.

Table 1: **Comparison of Different Augmentation Techniques (Validation).** The majority of our implementations improve upon the baseline across all metrics. The best results are in **bold font**, second best are underlined. Results worse than baseline on a particular metric are highlighted in red.

Method	RMSE ↓	RMSE LOG ↓	REL ↓	SQ. REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
0%	0.390	0.169	0.1269	0.063	0.801	0.976	0.9975
40%	0.370	0.160	0.1244	0.061	0.845	0.982	0.9967
80%	0.361	0.159	0.1250	0.057	0.836	0.985	0.9984
200%	0.379	0.160	0.1280	0.062	0.849	0.981	0.9974
400%	0.373	0.158	0.1272	0.059	0.846	0.986	0.9985
400% Pretrain	0.380	0.167	0.1227	0.059	**0.856**	0.981	0.9962
400% Mask	**0.356**	**0.154**	**0.1226**	**0.054**	0.850	**0.990**	**0.9986**

Table 2: **Results of the Different MDE Networks and Loss Functions.** DeepLabv3+[6] outperforms U-Net[23] in every metric with both loss functions. $\mathcal{L}_{berHu}$ [15] outperforms $\mathcal{L}_{SI_{Log}}$ [10] in every metric. The best results are in **bold font**, second best are underlined.

Network	Loss	RMSE ↓	LOG ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
DeepLab v3+	$\mathcal{L}_{SI_{Log}}$	0.464	0.193	0.142	0.753	0.959	0.995
	$\mathcal{L}_{berHu}$	**0.390**	**0.169**	**0.127**	**0.801**	**0.976**	**0.998**
U-Net	$\mathcal{L}_{SI_{Log}}$	0.678	0.275	0.198	0.615	0.876	0.964
	$\mathcal{L}_{berHu}$	0.610	0.263	0.203	0.620	0.914	0.983

However, we do not observe a clear relationship between how much additional data is added and the performance increase.

In addition to dataset size scaling, we also evaluate two techniques we label as pretrain and mask. For the pretraining, we begin by training the MDE network with pure novel data, then reduce the learning rate and fine-tune it on the original data. For the masking technique, we assign zeros to pixels in the image that contain highly inconsistent depth values with respect to its RGB image and ones to all other pixels. We multiply the pixel-wise loss with the mask. This, in theory, prevents the MDE network from being punished in areas that the MonoSDF failed to reconstruct.

Results on Test Data

We take the augmentation technique that was shown to be the best-performing, which is DeepLabv3+ [6] with $\mathcal{L}_{berHu}$ [15], 400% more virtual data and depth-penalty mask and evaluate it on the test dataset. We compare it to the same MDE model trained without any virtual-view supervision. The results in Table 3 show that our novel dataset augmentation technique outperforms the mentioned baseline model by a large margin. The most impressive results are the gains in threshold accuracy 2 (δ_2) and threshold accuracy 3 (δ_3), where our technique surpassed the baseline technique by more than 7 and 5 percentage points in absolute values which correspond to relative gains of more than 9% for the former, and exactly 12% for the latter.

Table 3: **Results (Test).** The best augmentation technique from the experiment on the validation dataset (400% masked) beats the non-augmented dataset (baseline) by a large margin in every evaluation metric.

Data	RMSE ↓	LOG ↓	REL ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Base	0.677	0.352	0.196	0.450	0.756	0.933
400% Masked	**0.562**	**0.318**	**0.179**	**0.504**	**0.825**	**0.963**

5. Conclusion

We demonstrate the effectiveness of our proposed novel dataset augmentation technique for Monocular Depth Estimation in a very constrained but controlled indoor environment. Although the resulting depth maps do not appear qualitatively better to the subjective human eye, they outperform the non-augmented version in every evaluated depth estimation metric across the validation and test datasets. These results make us optimistic about possible future endeavors to explore the potential of this method further. Due to the flexibility of our pipeline, as its components can be exchanged with any other component with similar usage, we hope to explore the utility of our pipeline in an outdoor setting, which could be very attractive for Autonomous Driving scenarios. Furthermore, we would like to experiment if a SOTA MDE network could also benefit from our augmentation technique.

References

[1] R. Bansal, G. Raj, and T. Choudhury. Blur image detection using laplacian operator and open-cv. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 63–67, 2016. 4

[2] Z. Bauer, Z. Li, S. Orts-Escolano, M. Cazorla, M. Pollefeys, and M. R. Oswald. NVS-MonoDepth: Improving monocular depth prediction with novel view synthesis. In *2021 International Conference on 3D Vision (3DV)*. IEEE, dec 2021. 4

[3] S. F. Bhat, I. Alhashim, and P. Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4008–4017, 2020. 2

[4] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28:3174–3182, 5 2016. 2

[5] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 3

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 1, 2, 3, 4, 5, 6

[7] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019. 2

[8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3

[9] A. Eftekhar, A. Sax, R. Bachmann, J. Malik, and A. Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans, 2021. 3

[10] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2, 4, 6

[11] H. Guo, S. Peng, H. Lin, Q. Wang, G. Zhang, H. Bao, and X. Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. 5

[13] P. Kellnhofer, L. C. Jebe, A. Jones, R. Spicer, K. Pulli, and G. Wetzstein. Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4287–4297, June 2021. 2

[14] D. Kim, W. Ga, P. Ahn, D. Joo, S. Chun, and J. Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *arXiv preprint arXiv:2201.07436*, 2022. 2

[15] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 2, 4, 5, 6

[16] J.-H. Lee and C.-S. Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 785–801. Springer, 2020. 2

[17] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[19] M. Oechsle, S. Peng, and A. Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[20] R. Peng, R. Wang, Y. Lai, L. Tang, and Y. Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[21] L. Piccinelli, C. Sakaridis, and F. Yu. idisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[22] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 3, 4, 6

[24] S. Shao, Z. Pei, W. Chen, R. Li, Z. Liu, and Z. Li. Urcdc-depth: Uncertainty rectified cross-distillation with cutflip for monocular depth estimatione. *https://arxiv.org/abs/2302.08149*, 2023. 2

[25] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3

[26] J. Wang, P. Wang, X. Long, C. Theobalt, T. Komura, L. Liu, and W. Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[27] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 2

[28] R. Wang, Z. Yu, and S. Gao. Planedepth: Self-supervised depth estimation via orthogonal planes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21425–21434, June 2023. 4

[29] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NIPS)*, 2021. 2

[30] Z. Yu, A. Chen, B. Antic, S. P. Peng, A. Bhattacharyya, M. Niemeyer, S. Tang, T. Sattler, and A. Geiger. Sdfstudio: A unified framework for surface reconstruction, 2022. 3

[31] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction, 2022. 1, 2, 3

[32] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan. New crfs: Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[33] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[34] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018. 2

[35] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2