NVS-MonoDepth: Improving Monocular Depth Prediction with Novel View Synthesis

Zuria Bauer[1], Zuoyue Li[2], Sergio Orts-Escolano[1], Miguel Cazorla[1], Marc Pollefeys[2,4], Martin R. Oswald[2,3]
[1]University of Alicante [2]ETH Zurich [3]University of Amsterdam [4]Microsoft
{zuria.bauer, sorts, miguel.cazorla}@ua.es; {li.zuoyue, marc.pollefeys, moswald}@inf.ethz.ch

Abstract

Building upon the recent progress in novel view synthesis, we propose its application to improve monocular depth estimation. In particular, we propose a novel training method split in three main steps. First, the prediction results of a monocular depth network are warped to an additional view point. Second, we apply an additional image synthesis network, which corrects and improves the quality of the warped RGB image. The output of this network is required to look as similar as possible to the ground-truth view by minimizing the pixel-wise RGB reconstruction error. Third, we reapply the same monocular depth estimation onto the synthesized second view point and ensure that the depth predictions are consistent with the associated ground truth depth. Experimental results prove that our method achieves state-of-the-art or comparable performance on the KITTI and NYU-Depth-v2 datasets with a lightweight and simple vanilla U-Net architecture.

1. Introduction

Monocular depth estimation has been an important topic in the computer vision community for years, and has been widely applied in many tasks such as 3D reconstruction, semantic segmentation, object detection, autonomous driving, or improving super-resolution on low-quality images. Recently, several papers [46, 7, 68] which used novel view synthesis (NVS) have demonstrated that accurate geometry priors can well benefit the synthesis quality. SynSin [68] showed that a rough but plausible depth estimation can be obtained even without depth supervision and that a depth estimation network can obtain supervision signals from an end-to-end view synthesis pipeline. Furthermore, the recent NVS methods using implicit representations [44, 43] do not use the depth information to optimize a neural radiance field, but the geometry can still be well learned showing that view constraints can guide a network to learn the depth of the image.

In this paper, we focus on monocular depth prediction and propose a new training pipeline for monocular depth prediction architectures which uses novel view consistency constraints as a supervisory signal in addition to the common depth supervision. To this end, our pipeline is trained with two neighboring viewpoints, which are simultaneously predicted during training to provide further network supervision via differentiable rendering of both the RGB image and the depth data. We evaluate our method on various indoor and outdoor datasets and investigate the key elements of our pipeline, carrying out an ablation study that shows the benefits of the proposed training strategy. Our approach yields lower RMSE scores than other state-of-the-art methods and outperforms them on all metrics on the NYU-Depth-v2 [45] dataset, while yielding on par performance with the best method on the KITTI [18] dataset.

Overall, our **contributions** are two-fold:

1. We propose to use novel-view synthesis as an additional supervisory signal to improve the training of a monocular depth estimation network. To this end, we propose two loss functions that augment the traditional depth supervision.
2. We present comprehensive experiments on both indoor and outdoor datasets that demonstrate the benefits of our approach, as well as an ablation study which empirically justifies our design choices.

We will make all source code and trained models publicly available upon publication to ensure reproducibility.

2. Related Work

This literature review focuses on the main related fields, namely depth estimation and novel view synthesis.

Depth estimation. Originally, the task of depth estimation from 2D image data relied on classical stereo vision [59] approaches. Below, we present a review of the monocular centered methods. Existing approaches can be roughly divided into supervised, self-supervised and unsupervised models. Here, we focus on supervised methods.

A pioneering work which took advantage of supervised learning for depth estimation from a single monocular image appeared in 2005 by Saxena *et al*. [53]. Their model used a discriminatively-trained Markov random field that incorporated local and global features from the image. Since then, various approaches to provide additional consistency to the prediction have been tested. The works of [40, 67, 66] opted to use semantic segmentation as an additional key feature to guide the training. [28, 31, 29] built upon the hypothesis of similarity between image pairs, assuming that two similar images were more likely to also have a similar 3D structure.

Numerous works [41, 37, 52, 33, 38, 36, 17, 9] based their proposed architectures on fully convolutional neural networks (CNN). [41] paired the network with a learning scheme which learns the unary and pairwise potentials of continuous Conditional Random Fields (CRFs). [37] used a DCNN model to map patches from multi-scale images. [33] proposed a modified ResNet-50 architecture with novel upsampling blocks allowing higher resolution. [38] captured scene details by considering information contained in depth gradients – with this purpose, they proposed a fast-to-train two-streamed CNN to regress the depth and depth gradients. [52] combined CNNs with Regression Forest to predict depth. [36] aimed to predict depth pixel-wise for a single color image. [17] integrated an affinity layer into a CNN, being able to combine learned absolute and relative features in a fully end-to-end model. [9] proposed a new lightweight and fast supervised CNN architecture combined with novel feature extraction models which are designed for real-world autonomous navigation. [58] proposed a feature-metric loss defined on the feature-level representation in their entangled dual task of depth and pose estimation, leading to an improved accuracy and demonstrated its effectiveness.

[4, 71, 67, 41] made use of the potentials of continuous CRFs to fuse multi-scale information obtained from different CNN layers that are then used to obtain the final depth estimation. [27, 25] took advantage of recent advances in Generative Adversarial Networks (GANs) [21] to sequentially estimate global and local structures of the depth images. [13, 69] made use of the internal camera parameters to improve the generalization capabilities of the architectures.

[12] introduced a coarse-scale network refined by a fine-scale network. Both networks were applied to the original input, but, in addition, the coarse network's output is passed to the fine network as additional first-layer image features. [16] introduced a network architecture to obtain high-resolution depth maps, using a spacing-increasing discretization (SID) strategy for depth and recasting depth network learning as an ordinal regression problem. [34] introduced a network composed of a dense feature extractor (the base network), a contextual information extractor (ASPP), local planar guidance layers and their dense connection for final depth estimation.

AdaBins [3] proposed a transformer-based architecture block that divides the depth range into bins whose center value is estimated adaptively per image. The final depth values are estimated as linear combinations of the bin centers. Until now, this work defines the state of the art on the monocular depth prediction benchmarks. Finally, [60] proposed a simple but effective scheme by incorporating the Laplacian pyramid into the decoder architecture.

While most previous works proposed various network architectures for performance improvements, we propose a novel training scheme. Our method is complementary to most approaches and can potentially be used with many architectures to augment their training. Moreover, a common practice in monocular depth estimation is the use of a deep, often pretrained encoder network like VGG-16 [11], ResNet-50 [33, 19, 32, 17, 49, 60], ResNet-101 [16], ResNext-101 [72, 34], or SeNet-154 [24, 6] with large parameter counts. In our experiments, we show that a simple U-Net [50] without any architectural modifications can achieve state-of-the-art or comparable performance using our proposed training scheme.

Novel view synthesis. Traditional novel view synthesis approaches are typically based on multi-view reconstruction using a geometric formulation [10, 14, 57], while most recent approaches rely more on deep neural networks which can even make predictions from a single input image. Some works do not directly build geometry for the source image, but resort to the scene flows. [63] directly output flows followed by a weighted aggregation based on the self-learned confidence. [47] performs transformation on the 3D latent space, using the target view's coarse RGB and visibility map as intermediate representations. [7] follows such a transformation, but supervises the network to predict the target depth and further calculate the flows. More works have better performance using to build geometry. [46] generates Ken Burns effects based on the supervised depth prediction on the source image. [68] shows that a plausible dense depth map can even be obtained through its end-to-end pipeline without depth supervision demonstrating the benefits of novel view synthesis for geometric reconstruction.

In contrast to our hybrid approach, [19, 75, 65] use novel view synthesis as main supervision and are fully unsupervised. The main difference to our approach can be condensed to their explicit usage of novel view synthesis to create stereo/multi-view depth predictions, while our approach only uses them for consistency at training time.

A recent popular NVS approach is via a neural radiance field, NeRF [44] and many derived works such as [43] which used multi-view data to train a latent appearance code to enable learning a neural scene representation. [48] used deformable videos using a second MLP applying a deformation for each frame of the video. [73] proposed a multi-image approach at test time. [42] organized the scene into a sparse voxel octree to speed up rendering by a factor of 10. [74]

handled unbounded scenes. NeRF-based models typically require several input images and expensive optimization, which makes them less suitable for our targeted monocular depth prediction problem. Rather than striving for perfect novel view synthesis, we merely use the technology as a tool for better network training.

3. Methodology

The main motivation for our approach is to provide additional supervisory signals from novel view synthesis constraints. The prediction of monocular depth networks may sometimes lack geometrical consistency with other view points. We address this issue by using novel view synthesis to provide additional geometrical consistency.The proposed pipeline follows a simple design, using light architectures to avoid memory shortages. Also, the structure of our pipeline is flexible with respect to user needs, since additional losses can be simply added or removed. A detailed analysis of all these elements can be found below.

3.1. Pipeline

We provide a detailed overview of our proposed pipeline in Figure 1. The pipeline consists of three major building blocks (a monocular depth prediction network – *DepNet*, a View Transformation procedure, and an image synthesis network – *SynNet*) aligned to provide additional consistency to the basic monocular network. The data flow through the pipeline is as follows. The RGB image from the source viewpoint (RGB 1) is used as the input of the depth network (*DepNet*). The output depth prediction together with the known camera intrinsics and the pose (extrinsics) are utilized to create a 3D point cloud. This 3D representation is then reprojected directly to the target view point (RGB 2) to obtain the initial 2D projection, which is used as the input for the synthesis network (*SynNet*). *SynNet* mainly fills occlusions and holes that were created due to the reprojection such that a complete RGB image of the target viewpoint (RGB 2) is estimated. Finally, the synthesized RGB of the target viewpoint is again used as an input for the same monocular depth network (*DepNet*). With the depth prediction for the second RGB viewpoint, we have an additional loss function to supervise the depth prediction network. In the following we detail all building blocks and provide additional information about the loss functions used for training the pipeline.

Monocular Depth Network – *DepNet*. Since the proposed pipeline uses two networks which can create memory shortages we used light architectures, with a small number of layers. To this end, we utilize a standard U-Net [50] architecture. It consists of a downsampling path and an upsampling path which give the U-shaped form to the network. The encoder and decoder part of the U-Net are connected with skip connections such that the upsampling path has additional concatenations with the high-resolution features obtained during the downsampling. Some benefits of this architecture are that it is lightweight, it can be effectively trained with a small amount of images and has proven to be able to outperform state-of-the art networks in its field. Another advantage of this architecture is its fast training and inference capability, e.g. the depth prediction inference for a 256×768 image from the test set, takes an average time of 36ms on an Nvidia 1080Ti.

Thus, the monocular depth architecture is a standard U-Net [50], with input size $256 \times 256 \times 3$ for the Replica [62] and NYU-Depth-v2 [45] datasets and a size of $256 \times 768 \times 3$ for the KITTI [18] dataset. The output sizes are $256 \times 256 \times 1$ and $256 \times 768 \times 1$, respectively. The network has a depth of 7 layers, downsampling the image to $1 \times 1 \times 1024$ and $1 \times 3 \times 512$. Downsampling the image to this small size for any other task than classification is not needed, since the final feature does not provide enough information for the network. Nevertheless, we experimented with fewer layers, downsampling the image to a size of 8×8, and the random artifacts the network created on the output were much higher than with downsampling the image to 1×1. In between the layers are the skip connections that provide high-resolution features to the decoder part. A more detailed description of the architectures can be found in the supplementary material.

View Transformation. From the predicted depth, 3D points in world coordinates are obtained through a unprojection procedure using the intrinsics of the camera and its pose (extrinsics). In the re-projection procedure, the points' world coordinates are first transformed to the coordinates of the novel view, followed by a conventional z-buffer to obtain the 2D image. The pixels without any projected points are initialized with zero for each channel.

Synthesis Network – *SynNet*. The third key component of the proposed pipeline is an image synthesis network. The purpose of this network is to improve the quality of the input image after warping it to the second view point. Due to disocclusions and missing data, the output of the view transformation might be incomplete and noisy. Example input, outputs pairs of this network are later shown in Fig. 5. Since we had the same prerequisites as for the monocular depth architecture, we decided on a similar architecture for this network. The synthesis network is a standard U-Net [50] architecture with also 7 layers. The input size of the network is $256 \times 256 \times 3$ or $256 \times 768 \times 3$ respectively for the Replica [62], NYU-Depth-v2 [45] and KITTI [18] datasets. The output of the network are RGB images with size $256 \times 256 \times 3$ or, $256 \times 768 \times 3$ respectively.

3.2. Loss functions

In the following we detail all utilized loss functions.

Pixel-wise losses. For the monocular depth network loss, $\mathcal{L}_1$ we use the traditional L1 loss for ground truth supervi-

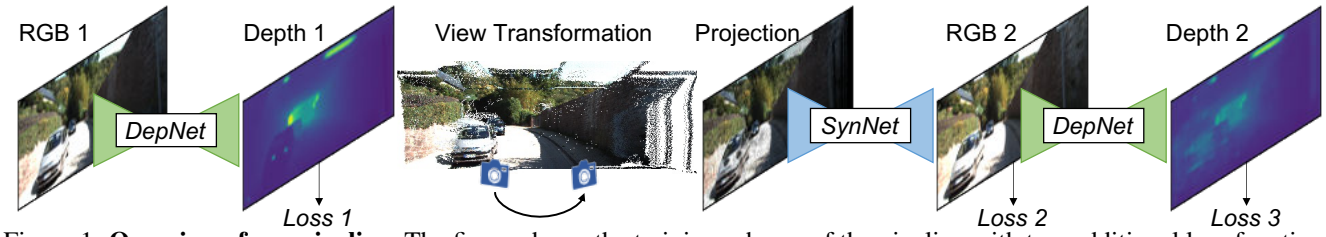

Figure 1: **Overview of our pipeline.** The figure shows the training scheme of the pipeline with two additional loss function that augment the traditional supervised depth loss (Loss 1). The depth prediction of *DepNet* is used to warp the reference view (RGB 1) to a target view (RGB 2). Due to occlusions and missing data this image gets completed with *SynNet* to provide a high-quality image prediction for the target view point and is supervised with RGB 2 (Loss 2). By applying the same *DepNet* again on the prediction of the target view we can further supervise the network with its corresponding GT depth. During test time, only *DepNet* is used for the monocular depth estimation. RGB 1 and RGB 2 are typically consecutive view points from a video or corresponding stereo images.

sion:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y_i}| \ . \tag{1}$$

The loss summarizes the absolute pixel-wise differences between the target depth prediction y_i and the predicted depth map $\hat{y_i}$. This value is the divided by the number of valid pixels N – those valid pixels are extracted from the ground truth depth map, excluding the locations where no information was provided.

The loss to train the synthesis network ($\mathcal{L}_2$) is an L1 loss between the target RGB image and the RGB image predicted by the network. In this case, we used the entirety of the image pixels without any masking.

The final loss applied to the output of the pipeline after the second depth prediction ($\mathcal{L}_3$) aims to preserve the accuracy of the second viewpoint and also provide an additional discrimination factor on the synthesis network. This is also an L1 loss function, but it is applied to the second view point instead of the first one. This third loss supervises the same *DepNet* twice, before and after the view transformation, but due to the view transformation in the loop, small prediction errors may lead to larger prediction errors in the second view after the view transformation. In sum, this defines a loss for the depth prediction network which is much more sensitive to small view point variations than the standard L1 loss on the input view alone.

Overall loss. The overall loss function is the weighted sum of all the previously listed individual loss functions. The weights can be used to control the training by enhancing or diminishing the importance of different networks, *e.g.*, one can choose to enhance the depth network and provide less importance to the synthesis. This loss function guarantees backpropagation over the entire pipeline and can be formalized as follows:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_1 + \beta \cdot \mathcal{L}_2 + \alpha \cdot \mathcal{L}_3 \ , \tag{2}$$

where α and β are the weighting hyper-parameters.

4. Experiments

We carried out a variety of monocular depth estimation experiments on indoor as well as outdoor scenes. We first briefly describe the data and evaluation metrics, then we present quantitative comparisons to state-of-the-art monocular depth estimation methods.

All our experiments have been carried out on two GPUs: an Nvidia 1080Ti and an Nvidia Titan X. Our method was implemented in TensorFlow [1]. For training, we used Adam as the optimizer, with a learning rate of 10^{-4} and a batch size of 8 to comply with the GPU memory limitations. The typical training span for one epoch is between 20 and 30 minutes.

Note that all the qualitative results shown in this paper are images from the Eigen test splits of the KITTI [18] or NYU-Depth-v2 [45] datasets. In case of the Replica [62] dataset, we selected our own unseen test split (details are provided in Section 4.1).

4.1. Datasets and evaluation metrics

Datasets. To train and assess our method, we considered a variety of datasets used for monocular depth estimation. Since we propose to train with multiple input images with overlapping view points we focus on multi-view datasets with ground truth depth.

We first discuss a number of datasets suitable for our multi-view setting which are mostly not used for monocular depth estimation.

The main focus when choosing a dataset relied on the size of the dataset, if it provides stereo pairs or sequences, if camera poses are provided for each frame, and if the dataset is outdoor or indoor. From the publicly available datasets the most suitable for indoor scenarios were: SceneNET-RGBD [22], Matterport 3D [5], SUNCG [61], ScanNet [8], Sun3D [70], InteriorNet [39], Middlebury [55] and Replica [62]. The standard benchmark dataset NYU-Depth-v2 [45] does not originally provide camera pose information, but since it is the benchmark dataset for monocular depth prediction, we wanted to include this dataset and used

Model	Backbone	#params (M)↓	REL↓	RMSE↓	$\text{RMSE}_{\log}$↓	Sq.Rel.↓	δ_1↑	δ_2↑	δ_3↑
Saxena *et al*. [54]	-	-	0.280	8.734	0.361	3.012	0.601	0.820	0.926
Eigen *et al*. [12]	-	-	0.190	7.156	0.270	1.515	0.692	0.899	0.967
Liu *et al*. [41]	-	40	0.217	6.986	0.287	1.841	0.647	0.882	0.961
Godard *et al*. [19]	ResNet-50	31	0.085	3.938	0.135	0.427	0.916	0.980	0.994
Kuznietsov *et al*. [32]	ResNet-50	-	0.138	3.610	0.138	**0.121**	0.906	0.989	0.995
Gan *et al*. [17]	ResNet-50	-	0.098	3.933	0.173	0.666	0.890	0.964	0.985
Fu *et al*. [16]	ResNet-101	110	0.072	2.727	0.120	0.307	0.932	0.984	0.994
Yin *et al*. [72]	ResNeXt-101	114	0.072	3.258	0.117	-	0.938	0.990	0.998
BTS [34]	ResNeXt-101	113	0.064	2.540	0.100	0.254	0.950	0.993	**0.999**
Song *et al*. [60]	ResNet-50	-	0.059	2.446	0.091	0.212	0.962	0.994	**0.999**
AdaBins [3]	EfficientNet-B5	78	0.058	**2.360**	**0.088**	0.190	**0.964**	**0.995**	**0.999**
U-Net baseline (*DepNet*)	U-Net	54	0.057	3.023	0.104	0.441	0.936	0.975	0.991
NVS-MonoDepth (Ours)	U-Net	54	**0.031**	2.702	0.089	0.292	0.963	0.989	0.997

Table 1: **State-of-the-art comparison on the KITTI [18] dataset.** For reference, we additionally show the results of our U-Net baseline in the second-to-last row, which is the same network, but trained without the proposed NVS losses. The reported numbers are from the corresponding original papers. Best results are shown in **bold** and second best results in blue.

the camera pose information provided by [64] for 13776 samples pairs. We trained our method on those samples and afterwards evaluated on the Eigen [12] split provided for the NYU-Depth-v2 [45] dataset. Additionally, we used the Replica dataset [62], to train our pipeline on a synthetic dataset with perfect camera poses. We split the provided 18 scenes in train (15), validation (2) and test sets (1). We extracted from the scenes and camera motion paths pairs of images with a maximum camera movement of 5 degrees in all the possible directions, including zoom. The network was trained with 10k images for training, and was validated and tested on 1k images that were never seen during training.

For the outdoor datasets, suitable ones were: Synthia [51], KITTI [18], UASOL [2], ETH3D [56] and Tanks and Temples [30]. Synthia [51] is also a synthetic dataset for indoor environments. ETH3D [56] was not considered as it provides too few images to train the network properly. Tanks and Temples [30] mostly provides scans of objects with only few complete scenes, making it less suitable for our setting. UASOL [2] would be a suitable dataset, but the camera poses provided by the dataset are position tracking which uses the first frame of each sequence as the fixed world coordinates and a vision-based algorithm is used to compute the rotation and translation between two consecutive frames. Since these results are less accurate than the camera poses provided by the KITTI [18] dataset, we decided to perform our experimentation on that dataset. Moreover, it is also one of the monocular depth benchmark datasets. To train the network on the KITTI [18] dataset, we used the specific Eigen [12] split which consist on 22k images for training, 888 for validation and 689 for test.

The main qualitative experimentation has been carried out on the KITTI [18] dataset.

Evaluation metrics. To evaluate the network against the state-of-the-art, we used the metrics from [12]. Given the ground truth depth image y with N valid pixels and the predicted depth image $\hat{y}$, the metrics are defined as follows: relative error (REL): $\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{y}$; root mean squared error (RMSE): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}$; log root mean squared error ($\text{RMSE}_{\log}$): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}|\log y_i-\log \hat{y}_i|^2}$; squared relative difference (Sq. Rel.): $\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|^2}{y_i}$; and threshold accuracy (δ_j): fraction of y_i such that $max(\frac{y_i}{\hat{y}_i},\frac{\hat{y}_i}{y_i})=\delta<1.25^j$ for $j\in\{1,2,3\}$.

4.2. Comparison to the state of the art

Quantitative results. We provide quantitative depth prediction results for the KITTI [18] and NYU-Depth-v2 [45] datasets compared to state-of-the-art methods. Since the Replica [62] dataset is not part of the monocular depth benchmark, we did not assess other state-of-the-art methods on this dataset but performed ablations studies.

For the KITTI [18] and NYU-Depth-v2 [45] datasets, we calculated the evaluation metrics by clipping the prediction values to the maximum depth of the sensors that filmed those datasets. In case of the KITTI [18] dataset, the minimum value is 0.01 meters and the maximum 80 meters. For the NYU-Depth-v2 [45], the minimum value is 0.01 meters and the maximum is 10 meters. This clipping is also common in previous works such as [15], [33], [3] or [34].

KITTI dataset. Table 1 lists the performance metrics on the KITTI [18] dataset. All the networks were trained on the KITTI-Eigen training set and evaluated on the KITTI-Eigen test set. On this dataset, our method yields comparable performance to the currently best state-of-the-art method. Our method yields a significantly lower relative error (REL) on all datasets. Based on the RMSE results, it can be noticed that our network performs worse with the larger residuals but performs better for small ones as indicated by the $\text{RMSE}_{\log}$ score and by the delta values - for δ_1 and δ_3 the performance is 0.1% worse than the state-of-the-art method, and for δ_2, 0.6% worse. Note that the number of network parameters is substantially smaller than the ones of the best performing

Model	Backbone	#params↓	REL↓	RMSE↓	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$
Eigen *et al.* [12]	-	141 M	0.158	0.641	0.769	0.950	0.988
Laina *et al.* [33]	ResNet-50	64 M	0.127	0.573	0.811	0.953	0.989
Hao *et al.* [23]	ResNet-101	60 M	0.127	0.555	0.841	0.966	0.991
Lee *et al.* [35]	-	119 M	0.131	0.538	0.837	0.971	0.994
Fu *et al.* [16]	ResNet-101	110 M	0.115	0.509	0.828	0.965	0.992
SharpNet [49]	ResNet-50	80 M	0.139	0.502	0.836	0.966	0.993
Hu *et al.* [24]	SENet-154	157 M	0.115	0.530	0.866	0.975	0.993
Chen *et al.* [6]	SENet-154	210 M	0.111	0.514	0.878	0.977	0.994
Yin *et al.* [72]	ResNeXt-101	110 M	0.108	0.416	0.875	0.976	0.994
BTS [34]	DenseNet-161	47 M	0.110	0.392	0.885	0.978	0.994
DAV [26]	DRN-D-22	25 M	0.108	0.412	0.882	0.980	0.996
AdaBins [3]	EfficientNet-B5	78 M	0.103	0.364	0.903	0.984	**0.997**
U-Net (*DepNet*)	U-Net	54 M	0.132	0.571	0.815	0.839	0.854
Ours	U-Net	54 M	**0.058**	**0.331**	**0.989**	**0.995**	**0.997**

Table 2: **State-of-the-art comparison on the NYU-Depth-v2 [45] dataset.** Please note the substantial reduction of the relative error by our approach. The reported numbers are from the corresponding original papers. Best results are shown in **bold** and second best results in blue.

approaches.

NYU-Depth-v2 dataset. Table 2 lists the performance metrics on the NYU-Depth-v2 [45] dataset. Despite the simple network architecture, our method outperforms all state-of-the-art networks trained on this dataset which indicates the effectivity of the proposed training scheme for monocular depth estimation. The proposed network is able to predict 4.8% more accurate than the state-of-the-art network, having also the smallest RMSE results. On the other side, the proposed method also achieves the highest results on all the δ thresholds, proving its effectiveness also on indoor datasets.

Replica dataset. Table 3 lists the performance metrics on the Replica [62] dataset. All networks were trained on 15 of the rooms (Apartments 0,1 and 2; frl apartments 1 to 5, office 0 to 3, rooms 1 and 2 and hotel 0), validated on 2 rooms (Office 3 and 4) and tested on 1 room (Room), the amount of data used for each split is provided in Section 4.1. The scores show reasonable performance on this synthetic indoor dataset as well. As can be seen in this case, the network proves to be able to improve on large (according to the RMSE) and on small residuals (according to delta values).

Qualitative results. Qualitative results of the proposed method are provided in Figure 2 and Figure 3. Figure 2 provides a comparison between the GT and the predictions provided by the proposed pipeline on the Replica [62] and NYU-Depth-v2 [45] dataset. In Figure 3, we compare the GT images provided by the KITTI [18] dataset with the predictions from our network. It can be noticed that the network is able to recover even small details found in the scene. Furthermore, it is also able to perform equally good on different depth ranges and on different image types (synthetic, real scenario). On the negative side, it can be seen that the network creates artifacts based on reflections or other complex lighting conditions.

Additionally, Figure 3 provides a qualitative comparison

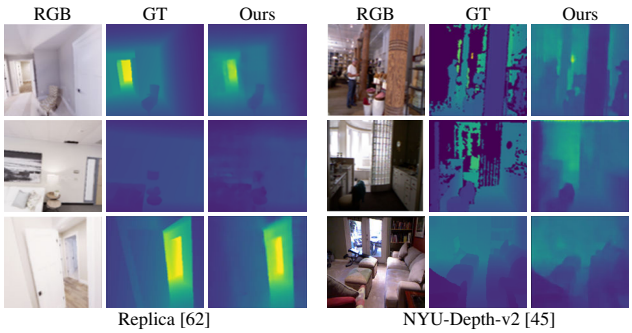

Figure 2: **Qualitative results on the Replica [62] and NYU-Depth-v2 [45] datasets.** We show the predictions of our proposed method compared to the GT images, proving its effectiveness on real and synthetic indoor datasets. Color scale: 0 (purple) to 80 meters (yellow).

between the state-of-the-art network AdaBins [3] and our predictions. Our method tends to perform better on thin structures while AdaBins [3] has a tendency to oversmooth the surrounding depth as visible at the pole on the left side in the third example. Otherwise, AdaBins [3] performs much better on the borders of closer objects such as the car in the second example, it also handles better shiny surfaces and specular reflections. Overall, the results are comparable, even though we are using a much simpler architecture. Additional qualitative comparison can be found in the supplementary material.

All the qualitative results provided from the KITTI [18] dataset show cropped results. This is due to the nature of the dataset, as the sensors used to film the depth sequences provides a depth range from 0.01 to 80 meters – these values were used to clip the depth predictions by the sensor by the authors. Figure 4 details this in a qualitative manner. In the first line we provide the color image and the GT provided by the dataset; as can be seen, the upper part of this prediction is set to zero since those values are further away than 80 meters. In the second line, we provide the actual prediction from the *DepNet* after training – the prediction is a dense depth map with hallucinations on the upper part of the image due to the absence of depth values for this part of the entire dataset. In the same line, masked prediction is the prediction of *DepNet* after applying a function to ignore the values that the original dataset does not provide (*i.e.*, masking them out). The last line of the figure provides the error image between the GT and the prediction or the masked prediction, respectively. As can be seen, for Error 1 which is the GT vs the prediction without alteration, the error values are accurate for the depth information provided by the network, but the error introduced by the depth pixels of the background provides a final error value of 17 meters. Instead, in Error 2 where we masked out the values that the actual dataset does not provide, it can be seen that the prediction is quite accurate with an error of 2.91 meters in the entire scene.

Figure 3: **Qualitative predictions on the KITTI [18] dataset.** We show the performance of **NVS-MonoDepth** (Ours) compared to AdaBins [3] leading to similar predictions, but with a much simpler architecture. Color scale: 0 (purple) to 80 meters (yellow).

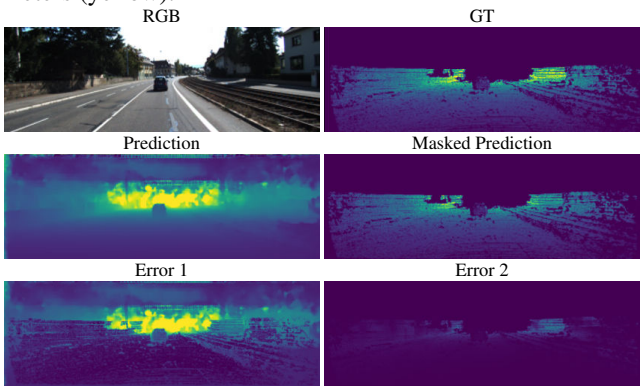

Figure 4: **Qualitative results of the prediction, masked prediction and error images for the KITTI [18] dataset.** The figure provides a qualitative explanation for the cropped predictions provided in the paper and further shows the masking method applied through the training and the evaluation process. Color scale: 0 (purple) to 80 meters (yellow).

Masking invalid pixels in the predictions is a common practice in monocular depth prediction, e.g. [12, 33, 20].

4.3. Ablation study

We further asses the individual building blocks of our pipeline in an ablation study that was performed on all three datasets: KITTI [18], NYU-Depth-v2 [45] and Replica [62].

Table 3 shows the performance scores for three variants of our pipeline. For the first variant ("DepNet"), we trained the monocular depth network (*DepNet* only with a supervised loss as the simplest baseline. In a second variant ("DepNet+SynNet"), we added the view transformation and synthesis network *SynNet* with the image loss in the second view. The third variant ("Ours") is our full proposed pipeline, which additionally adds the depth estimation based on the generated novel view output of *SynNet*.

All the architectures were trained in a supervised manner – we used the additional information for the additional loss functions used throughout the pipeline. Since the synthesis network has a big impact on the architecture loss, we set its loss weight to 0.5 such that the main focus stayed on the monocular network. By adding the synthesis step to the monocular architecture, we can already see a significant improvement on the error metrics, proving that the additional geometrical consistency is valuable for the training. In this case, after obtaining the new RGB view, it is passed through the monocular architecture to predict the depth of the second view which is compared with the GT from the second view. This leaves us in the end with a complex training pipeline, that, after training, can be used without need of providing any other information than the ground truth depth for evaluation purposes.

The values in Table 3 demonstrate that for the KITTI [18] dataset, the network shows a considerable improvement on all the error values compared to the baseline architecture trained without the proposed losses. The worse performance may be related to dynamic objects and the clipping of large depth values. Moreover, the KITTI exhibits a larger depth range and error residuals that can affect the method performance.The same conclusion can be drawn on the other datasets. Another conclusion is that using the overall loss function makes the network more sensitive to the smaller residuals achieving better delta results, even if it is not capable to outperform the state of the arts on all metrics.

Generally, one can observe that our method has more impact on the $\text{RMSE}_{\log}$ score which emphasizes small residual errors, in contrast to RMSE and Sq.Rel. which emphasize large residuals. This behavior can be expected since small depth residuals lead to smaller rendering errors in a second view and can be better corrected with a differentiable rendering loss, because large depth errors lead to highly non-local changes in a second view which are hard to be picked up by a network with a local receptive field.

Qualitative results of these experiments are provided in the supplementary material.

In Figure 6 we provide qualitative results between *DepNet*

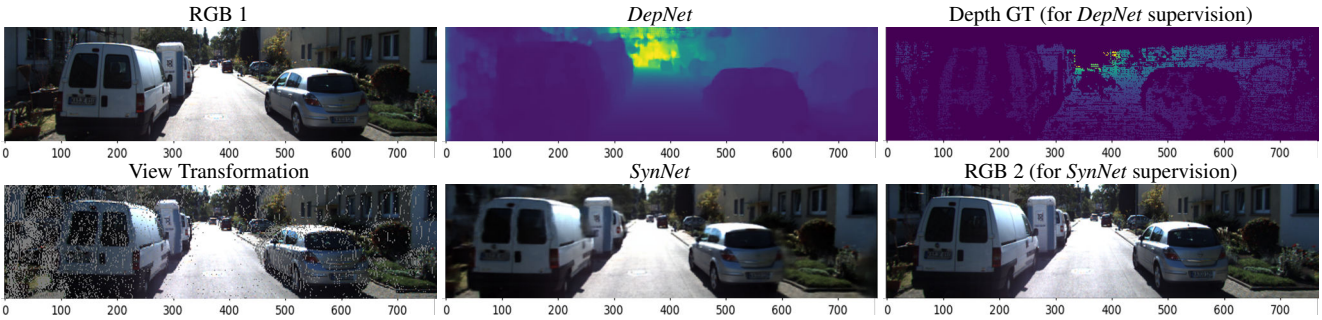

Figure 5: **Qualitative predictions from each of the steps of the proposed pipeline on the KITTI [18] dataset**. RGB 1 and RGB 2 are consecutive views or the corresponding stereo image. Color scale: 0 (purple) to 80 meters (yellow).

	Model	REL↓	RMSE↓	$\text{RMSE}_{\log}$↓	Sq.Rel.↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Replica	DepNet	0.905	2.071	-	1.765	0.421	0.708	0.802
	DepNet + SynNet	0.403	**0.775**	0.423	**0.276**	0.658	0.796	0.832
	Ours	**0.398**	0.940	**0.418**	0.406	**0.795**	**0.836**	**0.901**
NYU-V2	DepNet	0.132	0.571	0.147	0.915	0.815	0.839	0.854
	DepNet + SynNet	0.112	0.411	0.113	0.731	0.912	0.971	0.985
	Ours	**0.058**	**0.331**	**0.055**	**0.511**	**0.989**	**0.995**	**0.997**
KITTI	DepNet	0.057	3.023	0.104	0.441	0.936	0.975	0.991
	DepNet + SynNet	0.047	3.518	0.127	0.521	0.953	0.984	0.994
	Ours	**0.031**	**2.702**	**0.089**	**0.292**	**0.963**	**0.989**	**0.997**

Table 3: **Ablation study.** Comparison between the different variants of our approach on the Replica [62] dataset, NYU-Depth-v2 [45] dataset and on the KITTI [18] dataset. Each of our proposed building blocks leads to significant improvements of the baseline across all datasets.

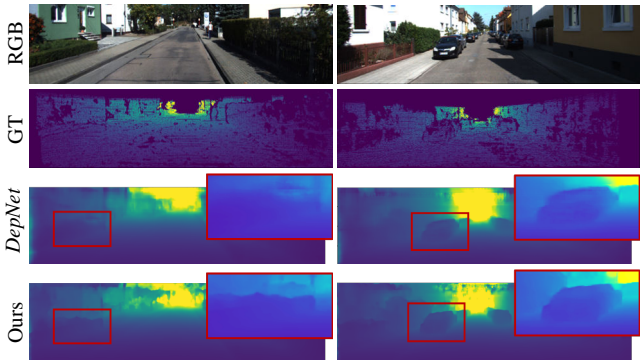

Figure 6: **Qualitative comparison between the vanilla U-Net [50] (*DepNet*) and our NVS-MonoDepth variants on the KITTI [18] dataset.** As shown in the close-ups, our method improves the prediction of *DepNet* with better sharpness and finer details. The contrast in the close-ups was adjusted for better visualization.

(standard U-Net [50] using only the first loss $\mathcal{L}_1$ during training) and the proposed method, **NVS-MonoDepth**, showing clear improvements on the contours of the predictions by adding the additional steps to the training.

Figure 5 provides qualitative results for each step of the proposed pipeline. As can be seen, the network is able to provide in each of these steps an accurate prediction. Since it is difficult to see the warping differences, we provided a measuring line below each image to make it easier to see the differences between the two viewpoints.

Related to the training loss function outputs of the different experiments we show a figure with those outputs in the supplementary material. Based on the loss functions, we can conclude that the monocular baseline is mostly the one that stays the most stable through the training process. By adding the synthesis step and the additional monocular architecture, we generate additional noise on the loss curve but, at the same time, the networks is able to outperform the simple monocular architecture. Also, this generated noise basically shows that by adding additional steps to the pipeline the network gets more difficult to train on one side, but on the other, the predictions improve achieving a much smaller error. More results, additional figures and explanations is provided in the supplementary material.

5. Conclusions

We presented a novel training method that makes use of additional loss functions to improve monocular depth estimation. The key idea is to use consistency constraints from other views via novel view synthesis as an additional supervisory signal for training in a multi-view setting, while testing only requires monocular input. This novel training procedure leads to substantial performance improvements compared to traditional supervised training. Our method achieves comparable results with the state of art for the KITTI [18] dataset. On the NYU-Depth-v2 [45] dataset our method outperforms all state-of-the-art methods across all metrics. The ablation of the individual pipeline parts of our building blocks demonstrates significant improvements supporting our design choices empirically across three datasets. Overall, we demonstrated the effectiveness of the proposed training method to improve the training of a monocular depth estimation network. As future work and to foster further research in this field, we believe that the proposed training principles can be combined with other monocular depth prediction approaches as also losses to further push their limits.

Acknowledgments. This work has been supported by the Spanish Government PID2019-104818RB-I00 Grant, co-funded by EU Structural Funds. Further funding was provided by a Fellowship from the Swiss Data Science Center, Innosuisse funding (Grant No. 34475.1 IP-ICT), and a research grant by FIFA.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Z. Bauer, Francisco Gomez-Donoso, Edmanuel Cruz, S. Orts-Escolano, and M. Cazorla. Uasol, a large-scale high-resolution outdoor stereo dataset. *Scientific Data*, 6, 2019.

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins, 2020.

[4] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2018.

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[6] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 694–700. ijcai.org, 2019.

[7] Xu Chen, Jie Song, and Otmar Hilliges. Monocular neural image based rendering with continuous view control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

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

[9] Raul de Queiroz Mendes, Eduardo Godinho Ribeiro, Nicolas dos Santos Rosa, and Valdir Grassi. On deep learning techniques to boost monocular depth estimation for autonomous navigation. *Robotics and Autonomous Systems*, 136:103701, 2021.

[10] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, page 11–20, New York, NY, USA, 1996. Association for Computing Machinery.

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2650–2658. IEEE Computer Society, 2015.

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2366–2374, Cambridge, MA, USA, 2014. MIT Press.

[13] José M. Fácil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. *CoRR*, abs/1904.02028, 2019.

[14] Fitzgibbon, Wexler, and Zisserman. Image-based rendering using image-based priors. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1176–1183 vol.2, 2003.

[15] Michaël Fonder, Damien Ernst, and Marc Van Droogenbroeck. M4depth: A motion-based approach for monocular depth estimation on video sequences. *CoRR*, abs/2105.09847, 2021.

[16] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *CoRR*, abs/1806.02446, 2018.

[17] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 232–247. Springer, 2018.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[19] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CoRR*, abs/1609.03677, 2016.

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. *CoRR*, abs/1806.01260, 2018.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[22] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. *CoRR*, abs/1511.07041, 2015.

[23] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision, 3DV 2018, Verona, Italy, September 5-8, 2018*, pages 304–313. IEEE Computer Society, 2018.

[24] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher

resolution maps with accurate object boundaries. *CoRR*, abs/1803.08673, 2018.

[25] Xun Huang, Yixuan Li, Omid Poursaeed, John E. Hopcroft, and Serge J. Belongie. Stacked generative adversarial networks. *CoRR*, abs/1612.04357, 2016.

[26] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVI*, volume 12371 of *Lecture Notes in Computer Science*, pages 581–597. Springer, 2020.

[27] H. Jung, Y. Kim, D. Min, C. Oh, and K. Sohn. Depth prediction from a single image with conditional adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1717–1721, 2017.

[28] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth extraction from video using non-parametric sampling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, pages 775–788, Berlin, Heidelberg, 2012. Springer-Verlag.

[29] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *In ICRA*, 2013.

[30] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[31] Janusz Konrad, Meng Wang, and Prakash Ishwar. 2d-to-3d image conversion by learning depth from examples. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–22, 2012.

[32] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. *CoRR*, abs/1702.02706, 2017.

[33] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016.

[34] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *CoRR*, abs/1907.10326, 2019.

[35] Wonwoo Lee, Nohyoung Park, and Woontack Woo. Depth-assisted real-time 3d object detection for augmented reality. In *ICAT*, volume 11, pages 126–132, 2011.

[36] Bo Li, Yuchao Dai, Huahui Chen, and Mingyi He. Single image depth estimation by dilated deep residual convolutional neural network and soft-weight-sum inference. *CoRR*, abs/1705.00534, 2017.

[37] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, 2015.

[38] Jun Li, Reinhard Klein, and Angela Yao. Learning fine-scaled depth maps from single RGB images. *CoRR*, abs/1607.00730, 2016.

[39] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018.

[40] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[41] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015.

[42] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *CoRR*, abs/2007.11571, 2020.

[43] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *CoRR*, abs/2008.02268, 2020.

[44] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934, 2020.

[45] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[46] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics*, 38(6):184:1–184:15, 2019.

[47] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[48] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *arXiv preprint arXiv:2011.12948*, 2020.

[49] Michaël Ramamonjisoa and Vincent Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*, pages 2109–2118. IEEE, 2019.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[51] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[52] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5506–5514, 2016.

[53] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In Y. Weiss, B.

Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1161–1168. MIT Press, 2006.

[54] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, May 2009.

[55] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, volume 8753, pages 31–42. Springer, 09 2014.

[56] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[57] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528, 2006.

[58] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *ECCV*, 2020.

[59] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, pages 746–760, Berlin, Heidelberg, 2012. Springer-Verlag.

[60] M. Song, S. Lim, and W. Kim. Monocular depth estimation using laplacian pyramid-based depth residuals. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2021.

[61] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[62] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[63] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *European Conference on Computer Vision*, 2018.

[64] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *CoRR*, abs/1812.04605, 2018.

[65] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. *CoRR*, abs/1712.00175, 2017.

[66] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[67] Peng Wang, Xiaohui Shen, Zhe Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 2800–2809, June 2015.

[68] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. SynSin: End-to-end view synthesis from a single image. In *CVPR*, 2020.

[69] Yicheng Wu, Vivek Boominathan, Huaijin Chen, Aswin Sankaranarayanan, and Ashok Veeraraghavan. Phasecam3d—learning phase masks for passive single view depth estimation. In *2019 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2019.

[70] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *International Conference on Computer Vision (ICCV)*, Dec. 2013.

[71] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. *CoRR*, abs/1803.11029, 2018.

[72] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[73] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *CoRR*, abs/2012.02190, 2020.

[74] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *CoRR*, abs/2010.07492, 2020.

[75] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *CoRR*, abs/1704.07813, 2017.