Behind the Scenes: Density Fields for Single View Reconstruction

Felix Wimbauer[1] Nan Yang[1] Christian Rupprecht[2] Daniel Cremers[1]
[1]Technical University of Munich [2]University of Oxford
{felix.wimbauer, nan.yang, cremers}@tum.de chrisr@robots.ox.ac.uk

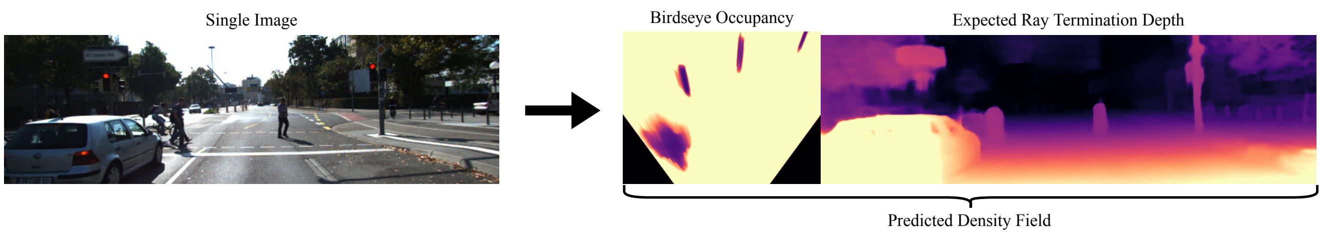

Figure 1. **Predicting a Density Field from a Single Image.** Through a novel "density field" formulation, which decouples geometry from color, and a novel self-supervised training scheme, our method learns to predict a volumetric scene representation from a single image in challenging conditions. In the birdseye occupancy view you can clearly make out the different objects in true 3D, which is not possible in traditional depth prediction. Please view our **video** provided in the **supplementary material** for more visualizations.

Abstract

Inferring a meaningful geometric scene representation from a single image is a fundamental problem in computer vision. Approaches based on traditional depth map prediction can only reason about areas that are visible in the image. Currently, neural radiance fields (NeRFs) can capture true 3D including color but are too complex to be generated from a single image. As an alternative, we introduce a neural network that predicts an implicit density field from a single image. It maps every location in the frustum of the image to volumetric density. Our network can be trained through self-supervision from only video data. By not storing color in the implicit volume, but directly sampling color from the available views during training, our scene representation becomes significantly less complex compared to NeRFs, and we can train neural networks to predict it. Thus, we can apply volume rendering to perform both depth prediction and novel view synthesis. In our experiments, we show that our method is able to predict meaningful geometry for regions that are occluded in the input image. Additionally, we demonstrate the potential of our approach on three datasets for depth prediction and novel-view synthesis.

1. Introduction

The ability to infer information about the geometric structure of a scene from a single image is of high importance for a wide range of applications from robotics to augmented reality. While traditional computer vision mainly focused on reconstruction from multiple images, in the deep learning age the challenge of inferring a 3D scene from merely a single image has received renewed attention.

Traditionally, this problem has been formulated as the task of predicting per-pixel depth values (*i.e.* depth maps). One of the most influential lines of work showed that it is possible to train neural networks for accurate single-image depth prediction in a self-supervised way only from video sequences. [13–15, 28, 41, 48, 52, 53, 55] Despite these advances, depth prediction methods are *not* modeling the true 3D of the scene: they model only a *single* depth value per pixel. As a result, it is not directly possible to obtain depth values from views other than the input view without considering interpolation and occlusion. Further, the predicted geometric representation of the scenes does not allow reasoning about areas that lie *behind* another object in the image (*e.g.* a house behind a tree), inhibiting the applicability of monocular depth estimation to 3D understanding.

Due to the recent advance of 3D neural fields, the related task of novel view synthesis has also seen a lot of progress. Instead of directly reasoning about the scene geometry, the goal here is to infer a representation that allows rendering views of the scene from novel viewpoints. While geometric properties can often be inferred from the representation, they are usually only a side product and lack in quality.

Even though neural radiance field [31] based methods achieve impressive results, they require many training images per scene and do not generalize to new scenes. Efforts have been made to condition the neural network on global or local scene features to enable generalization. However, this has only been shown to work well on simple scenes, for example scenes containing an object from a single category [40, 51]. Nevertheless, obtaining a neural radiance field from a single image has not been achieved before.

In this work, we tackle the problem of inferring a geometric representation from a single image by generalizing the depth prediction formulation to a continuous density field. Concretely, our architecture contains an encoder-decoder network that predicts a dense feature map from the input image. This feature map locally conditions a density field inside the camera frustum, which can be evaluated at any spatial point through a multi-layer perceptron (MLP). The MLP is fed with the coordinates of the point and the feature sampled from the predicted feature map by reprojecting points into the camera view. To train our method, we rely on simple image reconstruction losses.

Our formulation is different from existing methods that condition a radiance field on features in two key ways: First, we sample color values directly from the input frames through reprojection when performing volume rendering instead of using the MLP to predict color values. We find that only predicting density drastically reduces the complexity of the function the network has to learn, thereby enabling generalization across scenes. Second, we find a key component for generalization is network capacity. In many previous works an *encoder* extracts image features to condition local appearance, but the MLP is expected to generalize to multiple scenes. In contrast, we use a more powerful *encoder-decoder* that captures the entire scene in the features. the MLP then only evaluates those features locally.

While this formulation maintains simplicity, it enables a more powerful training scheme than traditional depth prediction. First, the continuous nature of the density field allows to directly render novel views from any viewpoint. Second, we can learn a true 3D representation of the scene by using frames other than the input to sample the color.

We demonstrate the potential of our new model and training formulation in a number of experiments on different datasets regarding the aspects of capturing true 3D, depth estimation and novel view synthesis. On KITTI [11] and KITTI-360 [25], we show both qualitatively and quantitatively that our model can indeed capture true 3D, and that our model achieves state-of-the-art depth estimation accuracy. On RealEstate10K [42] and KITTI, we achieve competitive novel view synthesis results, even though we rely on a purely geometry-based method.

2. Related Work

In the following, we review the most relevant works that are related to our proposed method.

2.1. Single-Image Depth Prediction

One of the predominant formulations to capture the geometric structure of a scene from a single image is predicting a per-pixel depth map. Learning-based methods have proven able to overcome the inherent ambiguities of this task by correlating contextual cues extracted from the image with certain depth values. One of the most common ways to train a method for single-image depth prediction is to immediately regress the per-pixel ground-truth depth values. [9] proposed a multi-scale convolutional neural network, which inspired many follow-up works [26]. Later approaches supplemented the fully-supervised training with reconstruction losses [20, 50], or specialise the architecture and loss formulation [10]. Recently, attention- and transformer-based methods have shown impressive accuracy [1, 21, 23, 24]. However, these approaches can only be trained on datasets that provide ground-truth depth annotations. To overcome this limitation, several papers focused on relying exclusively on reconstruction losses to train prediction networks. Both temporal video frames [55] and stereo frames [12], as well as combinations of both [13, 53] can be used as the reconstruction target. Different followup works refine the architecture and loss [14, 15, 28, 41, 48, 52]. [54] first predicts a discrete density volume as an intermediate step, from which depth maps can be rendered from different views. While they use this density volume for regularization, their focus is on improving depth prediction and their method does not demonstrate the ability to learn true 3D.

2.2. Neural Radiance Fields

Many works have investigated alternative approaches to representing scenes captured from a single or multiple images, oftentimes with the goal of novel view synthesis. Recently, [31] proposed to represent scenes as neural radiance fields (NeRFs). In NeRFs, a multi-layer perceptron (MLP) is optimized per scene to map spatial coordinates to color (appearance) and density (geometry) values. By evaluating the optimized MLP along rays and then integrating the color over the densities, novel views can be rendered under the volume rendering formulation [29]. Training data consists of a large number of images of the same scene from different viewpoints. To compute the camera poses, most datasets rely on traditional SFM and SLAM methods [4, 37, 38]. The training goal is to reconstruct these images as accurately as possible. NeRFs impressive performance inspired many follow-up works, that improve different parts of the architecture. [2, 3] refine the formulation for multi-scale and unbounded scenes. [17, 19, 33] introduce techniques to reduce the number of required training frames. [6, 36] speed up training by adding additional depth supervision.

In the traditional NeRF formulation, the entire scene is captured in a single, large MLP. This means that the trained network cannot be adapted to a different setting or used for other scenes. Further, the MLP has to have a high capacity, resulting in slow inference. Several methods propose to condition such MLPs on feature grids or voxels [5,27,30,32,35,40,44,51]. Through this, the MLP needs to store less information and can be simplified, resulting in significantly faster inference [5, 27, 32, 44]. Additionally, this allows for some generalization to different scenes [40, 51]. However, generalization is mostly limited to a single object category, or simple synthetic data, where the scenes differ mostly in local details. In contrast, our proposed method can generalize to highly complex outdoor scenes. This is a result of detaching color from the geometry representation.

2.3. Single Image Novel View Synthesis

While traditional NeRF-based methods achieve very impressive performance when provided with enough images per scene, they do not work when there is only a single image of a scene available. In recent years, a number of methods specialized for novel-view synthesis (NVS) from a single image emerged, often incorporating ideas from both depth prediction and neural radiance fields.

[46] predicts a layered depth image (LDI) [39] (two depth maps with associated color values), which allows to reason about occluded areas when viewed from a new angle. LDIs are extended in [7] to work per-object and in [8] to work with GANs to better hallucinate occluded areas. [45] proposes a method that can directly predict a multiplane image (MPI) [56] (multiple layers at certain depths with color and alpha values), which is further improved by [43]. [22] predicts a generalized multiplane image. Instead of directly outputting the discrete layers, the architecture's decoder receives a variable depth value, for which it outputs the layer. In [49], a network predicts both a per-pixel depth and feature map. The features are then reprojected into the novel view, where they are fed through an image-to-image network to obtain the rendering. While these methods achieve impressive NVS results, the quality of predicted geometry usually falls short. Most of the more advanced methods rely on a feature-based representation of the input image, from which the reconstructed color is retrieved. This relaxes the multi-view consistency assumptions and allows for leeway in the predicted geometry. Some methods even predict novel views directly without having any geometric representation [57].

3. Method

In the following, we describe a neural network architecture that predicts the geometric structure of a scene from a single image $\mathbf{I}_\text{I}$, as shown in Fig. 2. We first cover how we represent a scene as a continuous density field, and then propose a training scheme that allows our architecture to learn geometry even in occluded areas.

3.1. Notation

Let $\mathbf{I}_\text{I} \in [0,1]^{3\times H\times W} = (\mathbb{R}^3)^\Omega$ be the input image, defined on a lattice $\Omega = \{1,\dots,H\}\times\{1,\dots,W\}$. $T_\text{I} \in \mathbb{R}^{4\times 4}$ and $K_\text{I} \in \mathbb{R}^{3\times 4}$ are the corresponding world-to-camera pose matrix and projection matrix, respectively. During training, we have available an additional set of $N = \{1,2,\dots,n\}$ frames $\mathbf{I}_k, k \in N$ with corresponding world-to-camera pose and projection matrices $T_k, K_k, k \in N$. When assuming homogeneous coordinates, a point $\mathbf{x} \in \mathbb{R}^3$ in world coordinates can be projected onto the image plane of frame k with the following operation: $\pi_k(\mathbf{x}) = K_k T_k \mathbf{x}$

3.2. Predicting a Density Field

We represent the geometric structure of a scene as a function, which maps scene coordinates $\mathbf{x}$ to volume density σ. We term this function "density field". Inference happens in two steps. From the input image $\mathbf{I}_\text{I}$, an encoder-decoder network first predicts a pixel-aligned feature map $\mathbf{F} \in (\mathbb{R}^C)^\Omega$. The idea behind this is that every feature $f_\mathbf{u} = \mathbf{F}(\mathbf{u})$ at pixel location $\mathbf{u} \in \Omega$ captures the distribution of local geometry along the ray from the camera origin through the pixel at $\mathbf{u}$. It also means that the density field is designed to lie inside the camera frustum. For points outside of this frustum, we extrapolate features from within the frustum.

To obtain a density value at a 3D coordinate $\mathbf{x}$, we first project $\mathbf{x}$ onto the input image $\mathbf{u}'_\text{I} = \pi_\text{I}(\mathbf{x})$ and bilinearly sample the feature $f_{\mathbf{u}'} = \mathbf{F}(\mathbf{u}')$ at that position. This feature $f_{\mathbf{u}'}$, along with the positional encoding [31] $\gamma(d)$ of the distance d between $\mathbf{x}$ and the camera origin, and the positional encoding $\gamma(\mathbf{u}'_\text{I})$ of the pixel, is then passed to a multi-layer perceptron (MLP) ϕ. During training, ϕ and $\mathbf{F}$ learn to describe the density of the scene given the input view. We can interpret the feature representation $f_{\mathbf{u}'}$ as a descriptor of the density along a ray through the camera center and pixel $\mathbf{u}'$. In turn, ϕ acts as a decoder, that given $f_{\mathbf{u}'}$ and a distance to the camera, predicts the density at the 3D location $\mathbf{x}$.

$$\sigma_\mathbf{x} = \phi(f_{\mathbf{u}'_\text{I}}, \gamma(d), \gamma(\mathbf{u}'_\text{I})) \tag{1}$$

While most current work on neural fields also models the color using ϕ, here we do not. This drastically reduces the complexity of the distribution along a ray as density distributions tend to be simple, while color often contains complex high-frequency components. In our experiments, this makes capturing such a distribution in a single feature, so that it can be evaluated by an MLP at any given position, much more tractable.

3.3. Volume Rendering with Color Sampling

When rendering the scene from a novel viewpoint, unlike radiance field-based methods, we do not retrieve color from our scene representation directly. Instead, we sample the color for a point in 3D space from the available images.

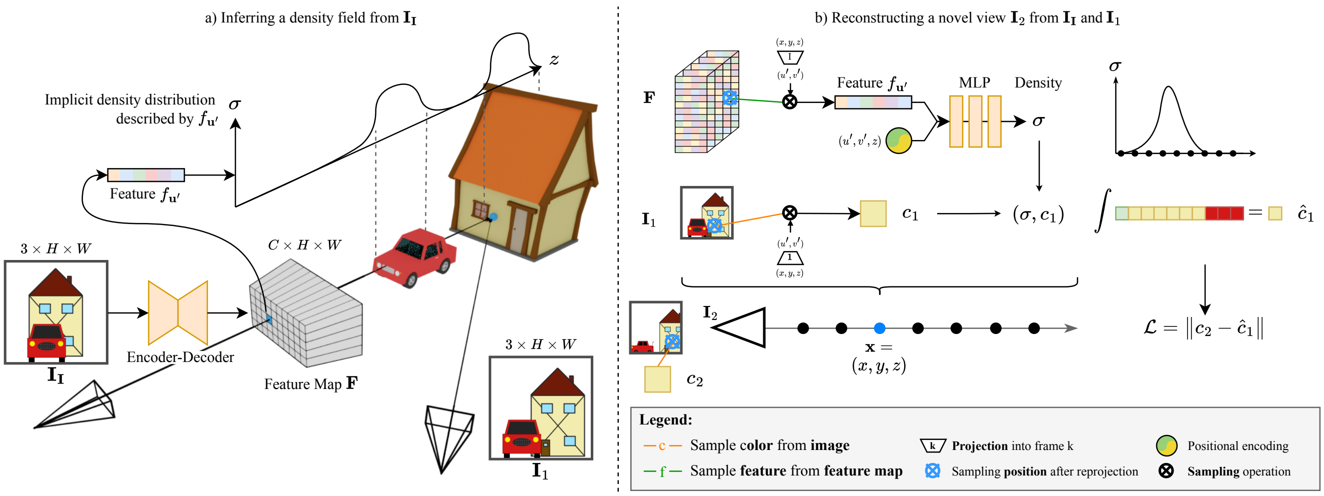

Figure 2. **Overview.** *a)* Our method first predicts a pixel-aligned feature map $\mathbf{F}$, which describes a density field, from the input image $\mathbf{I_I}$. For every pixel $\mathbf{u}'$, the feature $f_{\mathbf{u}'}$ implicitly describes the density distribution along the ray from the camera origin through $\mathbf{u}'$. Crucially, this distribution can model density even in occluded regions (*e.g.* the house). *b)* To render novel views, we perform volume rendering. For any point $\mathbf{x}$, we project $\mathbf{x}$ into $\mathbf{F}$ and sample $f_{\mathbf{u}'}$. This feature is combined with positional encoding and fed into an MLP to obtain density σ. We obtain the color c by projecting $\mathbf{x}$ into one of the views, in this case $\mathbf{I}_1$, and directly sampling the image.

Concretely, we first project a point $\mathbf{x}$ into a frame k and then bilinearly sample the color $c_{\mathbf{x},k} = \mathbf{I}_k(\pi_k(\mathbf{x}))$.

By combining $\sigma_{\mathbf{x}}$ and $c_{\mathbf{x},k}$, we can perform volume rendering [18, 29] to synthesise novel views. We follow the discretization strategy of other radiance field-based methods, *e.g.* [31]. To obtain the color $\hat{c}_k$ for a pixel in a novel view, we emit a ray from the camera and integrate the color along the ray over the probability of the ray ending at a certain distance. To approximate this integral, the density and color are evaluated at S discrete steps $\mathbf{x}_i$ along the ray. Let δ_i be the distance between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, and α_i be the probability of a ray ending between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. From the previous α_js, we can compute the probability T_i that the ray does not terminate before $\mathbf{x}_i$, *i.e.* the probability that $\mathbf{x}_i$ is not occluded.

$$\alpha_i = \exp(1 - \sigma_{\mathbf{x}_i}\delta_i) \qquad T_i = \prod_{j=1}^{i-1}(1 - \alpha_j) \tag{2}$$

$$\hat{c}_k = \sum_{i=1}^{S} T_i \alpha_i c_{\mathbf{x}_i,k} \tag{3}$$

Similarly, we can also retrieve the expected ray termination depth, which corresponds to the depth in a depth map. Let d_i be the distance between $\mathbf{x}_i$ and the ray origin.

$$\hat{d} = \sum_{i=1}^{S} T_i \alpha_i d_i \tag{4}$$

This rendering formulation is very flexible. We can sample the color values from any frame, most importantly it can be a different frame from the input frame. Crucially, it is even possible to obtain multiple colors from multiple different frames for a single ray, which makes it feasible to reason about occluded areas during training. Note that even though different frames can be used, the density is always based on features from the input image and does not change. During inference, color sampling from different frames is not necessary, everything can be done based on a single input image.

3.4. Loss Formulation

The training goal is to optimize both the encoder-decoder network and ϕ to predict a density field only from the input image that allows reconstructing other views.

Similarly to radiance fields and self-supervised depth prediction methods, we rely on an image reconstruction loss. For a single sample, we first compute the feature map $\mathbf{F}$ from $\mathbf{I_I}$ and randomly partition *all* frames $\hat{N} = \{\mathbf{I_I}\} \cup N$ into two sets $N_{\text{loss}}, N_{\text{render}}$. Note that the input image can end up in any of the two sets. We reconstruct the frames in N_{loss} by sampling from N_{render} using the camera poses and the predicted densities. The photometric consistency between the reconstructed frames and the frames in N_{loss} serves as the supervision signal of the density field. In practice, we randomly sample p *patches* P_i to use patch-wise photometric measurement. For every patch P_i in N_{loss}, we obtain a reconstructed patch $\hat{P}_{i,k}$ from *every* frame $k \in N_{\text{render}}$. We aggregate the costs between P_i and every $\hat{P}_{i,k}$ by taking the per-pixel *minimum* across the different frames k, similar to [13]. The intuition behind this is that for every patch, there is a frame in N_{render}, which "sees" the

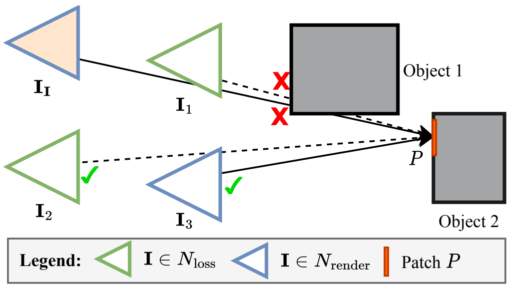

Figure 3. **Loss in Occluded Regions.** Patch P on Object 2 is occluded by Object 1 in the input frame $\mathbf{I}_\mathrm{I}$ and $\mathbf{I}_1$. In order to correctly reconstruct P in $\mathbf{I}_2$ from $\mathbf{I}_3$, the network needs to predict density for Object 2 *behind* Object 1.

same surface. Therefore, if the predicted density is correct, then it results in a very good reconstruction and a low error.

For the final loss formula, we use a combination of L1 and SSIM [47] to compute the photometric discrepancy, as well as an edge-aware smoothness term. Let $d_i^\star$ denote the inverse, mean-normalized expected ray termination depth of patch P_i.

$$\mathcal{L}_\text{ph} = \min_{k \in N_\text{render}} \left(\lambda_\text{L1} \text{L1}(P_i, \hat{P}_{i,k}) + \lambda_\text{SSIM} \text{SSIM}(P_i, \hat{P}_{i,k}) \right) \tag{5}$$

$$\mathcal{L}_\text{eas} = |\delta_x d_i^\star| \, e^{-|\delta_x P_i|} + |\delta_y d_i^\star| \, e^{-|\delta_y P_i|} \tag{6}$$

$$\mathcal{L} = \sum_{i=1}^{p} \sum_{x,y \in P} \left(\mathcal{L}_\text{ph} + \lambda_\text{eas} \mathcal{L}_\text{eas} \right)(x, y) \tag{7}$$

Learning true 3D. Our loss formula Eq. (7) is the same as self-supervised depth prediction methods, like [13]. The key difference, however, is that by design depth prediction methods can only densely reconstruct the input image, for which the per-pixel depth was predicted.

In contrast, our density field formulation allows us to reconstruct *any* frame from *any other* frame. Consider an area of the scene, which is occluded in the input $\mathbf{I}_\mathrm{I}$, but visible in two other frames $\mathbf{I}_\mathrm{k}, \mathbf{I}_\mathrm{k+1}$, as depicted in Fig. 3. During training, we aim to reconstruct this area in $\mathbf{I}_\mathrm{k}$. The reconstruction based on colors sampled from $\mathbf{I}_\mathrm{k+1}$ will give a clear training signal to correctly predict the geometric structure of this area, even though it is occluded in $\mathbf{I}_\mathrm{I}$. Note, that in order to learn geometry about occluded areas, we require at least **two additional** views besides the input during training, *i.e.* to look *behind the scenes*.

Handling invalid samples. While the frustums of the different views overlap for the most part, there is still a chance of a ray leaving the frustums, thus sampling invalid features, or sampling invalid colors. Such invalid rays lead to noise and instability in the training process. Therefore, we propose a policy to detect and remove invalid rays. Our intuition is that when the amount of contribution to the final aggregated color, that comes from invalidly sampled colors or features, exceeds a certain threshold τ, the ray should be discarded. Consider a ray that is evaluated at positions $\mathbf{x}_i, i \in [1, 2, \ldots, S]$ and reconstructed from frames K. $O_{i,k}, k \in \{\text{I}\} \cup K$ denotes the indicator function that $\mathbf{x}_i$ is outside the camera frustum of frame k. Note that we always sample features from the input frame. We define $\text{IV}(k)$ to be the function indicating that the rendered color based on frame k is invalid as:

$$\text{IV}(k) = \sum_{i=1}^{S} T_i \alpha_i \left(O_{i,\text{I}} \vee O_{i,k} \right) > \tau \tag{8}$$

Only if $\text{IV}(k)$ is true for *all* frames the ray was reconstructed from, we ignore the ray when computing the loss value. The reasoning behind this is that non-invalid rays will still lead to the lowest error. Therefore, the $\min$ operation in equation 5 will ignore the invalid rays.

3.5. Implementation Details

We implement our model in PyTorch [34] on a single Nvidia RTX A40 GPU with 48GB memory. The encoder-decoder network follows [13] using a ResNet encoder [16] and predicts feature maps with 64 channels. The MLP ϕ is made lightweight with only 2 fully connected layers and 64 hidden nodes each. We predict feature maps at four different scales and compute the loss for each scale. We use a batch size of 16 and sample 32 patches of size 8×8 from the images for which we want to compute the reconstruction loss. We sample every ray at 64 locations, based on a linear spacing in inverse depth. For more details, *e.g.* exact network architecture and further hyperparameters, please refer to the supplementary material.

4. Experiments

To demonstrate the abilities and advantages of our proposed method, we conduct a wide range of experiments. First, we demonstrate that our method is uniquely able to capture a holistic geometry representation of the scene, even in areas that are occluded in the input image. Additionally, we also show the effect of different data setups on the prediction quality. Second, we show that our method, even though depth maps are only a side product of our scene representation, achieves depth accuracy on par with other state-of-the-art self-supervised methods, that are specifically designed depth prediction. Finally, we demonstrate that, even though our representation is geometry-only, our method can be used to perform high-quality novel view synthesis from a single image.

4.1. Data

For our experiments, we use three different datasets: KITTI [11], KITTI-360 [25], and RealEstate10K [56]. These are very challenging datasets, consisting of in-the-wild scenes with a high diversity. KITTI and KITTI-360 both provide forward-facing frame sequences with synchronized stereo frames, captured from calibrated cameras mounted on a driving car. KITTI-360 additionally offers two fisheye cameras pointing to the left and right. RealEstate10K only provides monocular video sequences. For KITTI, we compute camera poses using ORB-SLAM 3 [4], a non learning-based SLAM system. Both for KITTI-360 and RealEstate10K, we rely on the groundtruth poses.

For KITTI, we train our models based on the temporal and static stereo frames of two or three timesteps. When training on KITTI-360, we additionally use frames from the two fisheye cameras with an offset of ten timesteps. As the fisheye cameras face sideways, the offset ensures that the camera frustums of all frames of a sample overlap. In a pre-processing step, we resample the fisheye cameras to match the camera parameters of the forward-facing, perspective cameras. When training on RealEstate10K, we use three frames, with an offset randomly picked between one and thirty frames.

Training is performed for 50 epochs on KITTI (approx. 125k steps), 20 epochs on KITTI-360 (approx. 143k steps), and 360k iterations on RealEstate10K. We use a resolution of 640×192 for KITTI and KITTI-360, and follow [22] in using a resolution of 384×256 for RealEstate10K. Running a full training takes around four days for each dataset.

4.2. Capturing true 3D

Evaluation of fully geometric 3D representations like a density field is difficult. Real-world datasets usually only provide ground truth data captured from a single viewpoint, *e.g.* RGB-D frames and Lidar measurements. Nevertheless, we aim to evaluate and compare this key advantages of our method both qualitatively and quantitatively. Through our proposed training scheme, our networks are able to learn to also predict meaningful geometry in occluded areas.

To overcome the lack of volumetric ground truth, we leverage the 3D bounding boxes and semantic segmentation masks provided by KITTI-360. For any point in space, we check whether this point is within any of the bounding boxes and whether it is occluded by another object in the input frame. We thereby construct a coarse occupancy and visibility field. As bounding boxes do not have a tight fit around an object, we further refine the occupancy by checking, whether a point falls into the segmentation mask, when projected into the input frame.

Using this coarse occupancy groundtruth, it is possible to quantitatively evaluate density fields. We first define a subset of the KITTI-360 Segmentation validation split,

Method	O_{acc}	IE_{acc}	IE_{rec}
Depth prediction	**0.93**	0.00	0.00
Ours, 3xT (L)	<u>0.92</u>	**0.95**	0.10
Ours, 3xT (L + R)	**0.93**	0.76	<u>0.12</u>
Ours, 2xT (L + R + F)	**0.93**	<u>0.79</u>	**0.38**

Table 1. **3D Scene Occupancy Accuracy.** We evaluate the capability of the model to predict occupancy *behind* objects in the image. Depth prediction naturally has no ability to predict behind occlusions, while our method improves when training with more views. Inference from a single image. Samples are evenly spaced in a cuboid $w = [-4m, 4m], h = [-1m, 0m], d = [3m, 20m]$ relative to the camera. **Legend**: see Fig. 4.

which contains 457 test frames and around 114000 training frames. For every frame, we sample points in a cuboid area in the camera frustum and compute the following metrics: 1. Occupancy accuracy (O_{Acc}), 2. Invisible and empty accuracy (IE_{Acc}), and 3. Invisible and empty recall (IE_{Rec}). O_{Acc} evaluates the occupancy predictions across the whole scene volume. IE_{Acc} and IE_{Rec} specifically evaluate invisible regions, evaluating performance beyond depth prediction.

We train a MonoDepth 2 model to serve as a baseline representing ordinary depth prediction methods. Here, we consider all points behind the predicted depth to be occupied. Then we train several versions of our model dataset with different data configurations. One with frames from three consecutive timesteps from a single camera, one with three consecutive timesteps from the left and right camera, and one with two consecutive timesteps from the left and right camera and the left and right fisheye cameras. Tab. 1 reports the obtained results.

The depth prediction baseline achieves a strong overall accuracy, but is, by design, not able to predict free space in occluded areas. Our model achieves the same overall accuracy as the baseline, while it can additionally recover the geometry of the occluded parts of the scene. More importantly, our model becomes better in predicting *free space in occluded areas* when training with more views, naturally providing a better training signal for occluded areas.

To qualitatively visualize this effect, we sample the camera frustum in horizontal slices from the center of the image downwards and aggregate the density in Fig. 4. This shows the layout of the scene, similar to the birds-eye perspective but for density. Since we do not require ground truth, we use examples from KITTI instead of KITTI-360. This allows us to compare against available pretrained models.

Even when only trained on two timesteps with stereo, our method is already able to reason about the geometry of smaller object like the light post in the left image or the cyclist in the middle image. In this setting, the baseline between frames is very small, meaning there are few occlusions and consequently only very weak training signal

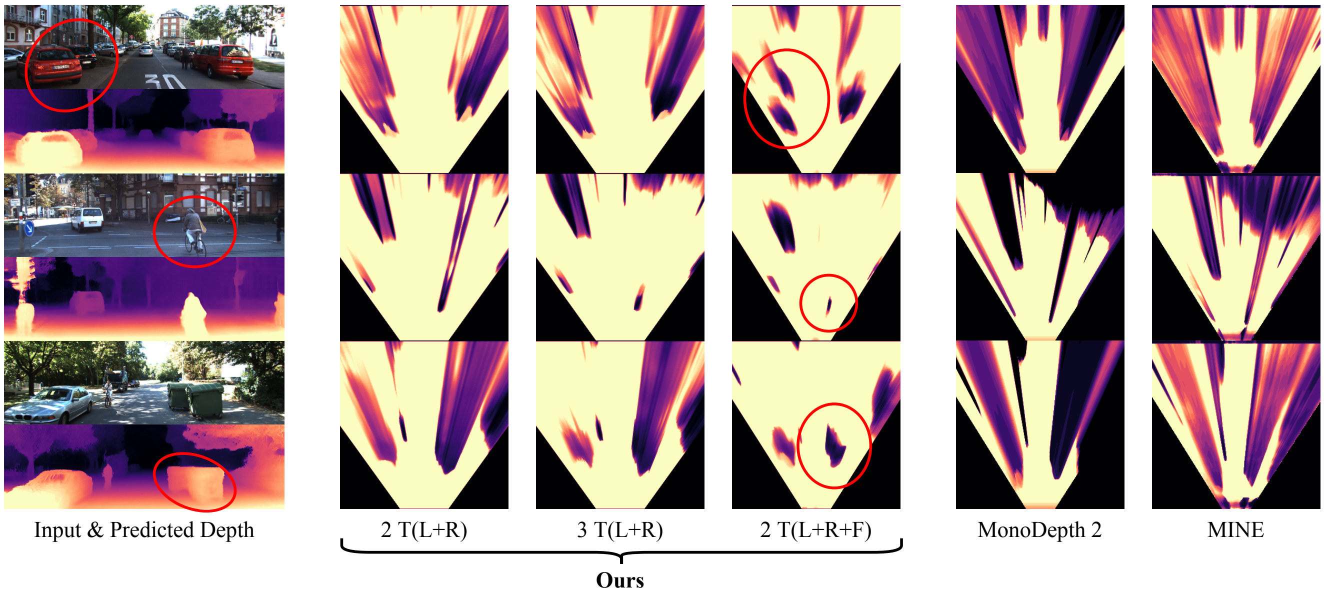

Figure 4. **Occupancy Estimation.** Top-down visualization of the occupancy map predicted by different methods. We show an area of $x = [-15m, 15m], z = [5m, 30m]$ and aggregate density from the y-coordinate of the camera $1m$ downward. Depth prediction methods such as MonoDepth2 do not predict a full 3D volume. Thus, objects cast "occupancy shadows" behind them. Our method predicts the full occupancy volume and can thus reason about the space behind objects. Training with more views improves occupancy estimation. Inference is from a single image. **Legend**: n T(): *n timesteps of*, L: *left camera*, R: *right camera*, F: *left and right fisheye camera*.

for areas behind other surfaces. The more information we give the network during training, the better the reasoning for occluded area becomes. With three timesteps, the network can reasonably reconstruct cars, and objects become less stretched. With the addition of fisheye cameras in the model trained on KITTI-360, the model has to reconstruct significant areas that are occluded in the input during training. This strong signal lets the model learn sharp object boundaries, as can be seen for several cars in the examples. Additionally to our models, we show profiles for MonoDepth 2 [13], a representative depth prediction approach. We build the profiles by considering everything behind the predicted depth value to be of full density. Thus, all objects cast occupancy shadows along the viewing direction. MINE [22] also predicts a volumetric representation from a single image. However, it does not produce meaningful density prediction behind objects. Instead, similar to depth prediction, all objects cast occupancy shadows along the viewing direction. We hypothesize that this problem is grounded in the fact that the scene representation in MINE also contains color information. Therefore, the problem of reconstructing multiple frames during traininig is much more underconstrained. Through color prediction, the model can take "shortcuts" and only learn to predict the right color instead of the right geometry.

4.3. Depth Prediction

While our method does not predict depth maps directly, they can be synthesized as a side product from our representation through the expected ray termination depth $\hat{d}$. To demonstrate that our predicted representation achieves high accuracy, we train our model on KITTI sequences with the established Eigen [9] split and compare to self-supervised depth prediction methods.

As can be seen in Tab. 2, our method performs on par with the current state-of-the-art methods for self-supervised depth prediction. Our synthesized depth maps capture fine details and contain less noise, as often seen with depth maps obtained from neural radiance fields.

MINE [22] is the only other volumetric method evaluating depth prediction. However, their scene representation also contains color information, which relaxes the requirement for photometric consistency between views during training and reduces geometric accuracy. As can be observed in the bottom part of Fig. 5, we also achieve significantly sharper and more accurate depth prediction. Overall, we achieve competitive performance, even though depth prediction is not the main objective of our approach.

4.4. Novel View Synthesis from a Single Image

As we obtain a volumetric representation of a scene from a single image, we are able to synthesize images from novel viewpoints by sampling color form the input image. Thus, we also evaluate novel view synthesis from a single image.

Model	Volumetric	Split	Abs Rel	Sq Rel	RMSE	$\text{RMSE}_{\log}$	$\alpha < 1.25$	$\alpha < 1.25^2$	$\alpha < 1.25^3$
EPC++ [28]	✗	Eigen [9]	0.128	1.132	5.585	0.209	0.831	0.945	0.979
MonoDepth 2 [13]	✗		0.106	0.818	4.750	0.196	0.874	0.957	0.975
PackNet [15]	✗		0.111	0.785	4.601	0.189	0.878	0.960	0.982
DepthHint [48]	✗		0.105	0.769	4.627	0.189	0.875	0.959	0.982
FeatDepth [41]	✗		0.099	0.697	4.427	0.184	0.889	0.963	0.982
DevNet [54]	(✓)		**0.095**	**0.671**	**4.365**	**0.174**	**0.895**	**0.970**	**0.988**
Ours	✓		0.102	0.751	4.407	0.188	0.882	0.961	0.982
MINE [22]	✓	Tulsiani [46]	0.137	1.993	6.592	0.250	0.839	0.940	0.971
Ours	✓		**0.132**	**1.936**	**6.104**	**0.235**	**0.873**	**0.951**	**0.974**

Table 2. **Depth Prediction.** While our goal is fully volumetric scene understanding, we compare to the state of the art in depth estimation trained only with reconstruction losses. Our approach achieves competitive performance with specialized methods while improving over the only other fully volumetric approach MINE [22]. DevNet [54] performs better, but does not show any results from the volume directly.

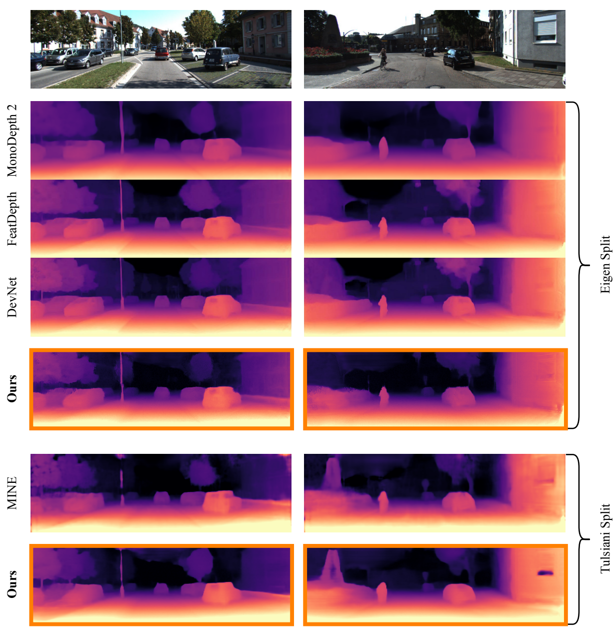

Figure 5. **Depth Prediction.** Expected ray termination depth compared with depth prediction results of other state-of-the-art methods [13, 22, 41, 54] on both the Eigen [9] and [46] split. Our predictions are very detailed and sharp, and capture the structure of the scene, even when trained on a smaller split like like Tulsiani. Visualizations for DevNet and FeatDepth are taken from [54].

To demonstrate the variability of our approach, we train two models, one on RealEstate10K [56] (indoor video sequences of house viewings), and one on the KITTI split proposed by Tulsiani et al. [46] (mostly sequences from dense, urban areas). As Tab. 3 shows, our method achieves strong performance on both datasets, despite the fact, that we only predict geometry and obtain color by sampling the input image. Our results are comparable with many recent methods, that were specifically designed for this task, and of which some even use sparse depth supervision during training for RealEstate10K (MPI, MINE). MINE [22] achieves slightly better accuracy. This can be attributed to them being able to predict color and thereby circumventing issues arising from imperfect geometry.

	KITTI			RealEstate10K		
Model	LPIPS	SSIM	PSNR	LPIPS	SSIM	PSNR
SynSin [49]	n/a	n/a	n/a	1.180	0.740	22.3
Tulsiani [46]	n/a	0.572	16.5	0.176	0.785	23.5
MPI [45]	n/a	0.733	19.5	n/a	n/a	n/a
MINE [22]	**0.112**	**0.828**	**21.9**	**0.156**	**0.822**	**24.5**
Ours	0.144	0.764	20.1	0.194	0.755	24.0

Table 3. **Novel View Synthesis.** We test the NVS ability on KITTI (Tulsiani split [46]) and RealEstate10K (MINE split [22], target frame randomly sampled within 30 frames). Even though our method does not predict color, we still achieve strong results.

5. Conclusion

In this paper, we introduced a new approach for learning to estimate the 3D geometric structure of a scene from a single image. Our method predicts a continuous density field, which can be evaluated at any point in the camera frustum. A key difference to neural radiance field based methods is that, by disentangling color and geometry, we strongly encourage multi-view consistency. Additionally, only predicting geometry makes the representation significantly more tractable. This enables us to train a network on large in-the-wild datasets with challenging scenes. Our training scheme is self-supervised and uses a new reconstruction loss formulation based on frames from mono or stereo video. We show that our method is able to capture geometry in occluded areas. We evaluate depth maps synthesized from the predicted representation achieving comparable results to state-of-the-art methods which are specifically designed for depth prediction. Despite only predicting geometry, our model can also achieve high accuracy for novel view synthesis from a single image on both KITTI and RealEstate10K.

Acknowledgements. C. R. is supported by VisualAI EP/T028572/1 and ERC-UNION-CoG-101001212.

References

[1] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11746–11752. IEEE, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[4] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 2, 6

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022. 3

[6] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 2

[7] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5369–5378, 2019. 3

[8] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 125:333–340, 2019. 3

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 2, 7, 8

[10] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 6

[12] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. 2

[13] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1, 2, 4, 5, 7, 8

[14] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33:12626–12637, 2020. 1, 2

[15] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 1, 2, 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[17] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 2

[18] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 4

[19] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022. 2

[20] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017. 2

[21] Sihaeng Lee, Janghyeon Lee, Byungju Kim, Eojindl Yi, and Junmo Kim. Patch-wise attention network for monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1873–1881, 2021. 2

[22] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12578–12588, 2021. 3, 6, 7, 8

[23] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *arXiv preprint arXiv:2203.14211*, 2022. 2

[24] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 2

[25] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 6

[26] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep

convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015. 2

[27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[28] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2624–2641, 2019. 1, 2, 8

[29] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 2, 4

[30] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 3

[31] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3, 4

[32] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 3

[33] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[35] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 3

[36] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 2

[37] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[38] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[39] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998. 3

[40] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Ambrus, Adrien Gaidon, William T Freeman, Fredo Durand, Joshua B Tenenbaum, and Vincent Sitzmann. Seeing 3d objects in a single image via self-supervised static-dynamic disentanglement. *arXiv preprint arXiv:2207.11232*, 2022. 2, 3

[41] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *European Conference on Computer Vision*, pages 572–588. Springer, 2020. 1, 2, 8

[42] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 2

[43] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 3

[44] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 3

[45] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 3, 8

[46] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 3, 8

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[48] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2162–2171, 2019. 1, 2, 8

[49] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7467–7477, 2020. 3, 8

[50] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 817–833, 2018. 2

[51] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2, 3

[52] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. New crfs: Neural window fully-connected crfs for monocular depth estimation. *arXiv preprint arXiv:2203.01502*, 2022. 1, 2

[53] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 340–349, 2018. 1, 2

[54] Kaichen Zhou, Lanqing Hong, Changhao Chen, Hang Xu, Chaoqiang Ye, Qingyong Hu, and Zhenguo Li. Devnet: Self-supervised monocular depth learning via density volume construction. *arXiv preprint arXiv:2209.06351*, 2022. 2, 8

[55] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 1, 2

[56] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 3, 6, 8

[57] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016. 3