Boosting Self-Supervision for Single-View Scene Completion via Knowledge Distillation

Keonhee Han[*,1] Dominik Muhle[*,1,2] Felix Wimbauer[1,2] Daniel Cremers[1,2]
[1]TU Munich [2]Munich Center for Machine Learning
{keonhee.han, dominik.muhle, felix.wimbauer, cremers}@tum.de

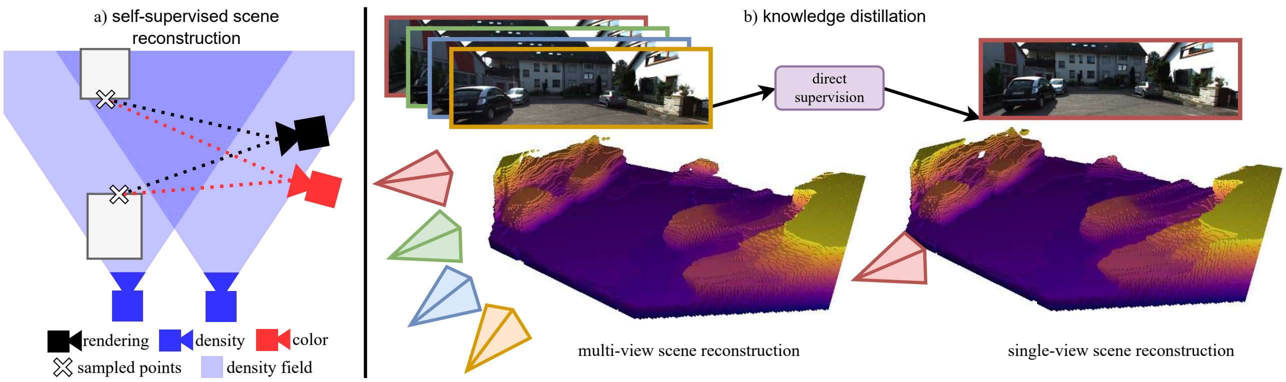

Figure 1. **Knowledge Distillation from Multi-View to Single-View.** We propose to boost single-view scene completion by exploiting additional information from multiple images. a) we first train a novel multi-view scene reconstruction algorithm that is able to fuse density fields from multiple images in a fully self-supervised manner. b) we then employ knowledge distillation to directly supervise a state-of-the-art single-view reconstruction model in 3D to boost its performance. https://keonhee-han.github.io/publications/kdbts/

Abstract

Inferring scene geometry from images via Structure from Motion is a long-standing and fundamental problem in computer vision. While classical approaches and, more recently, depth map predictions only focus on the visible parts of a scene, the task of scene completion aims to reason about geometry even in occluded regions. With the popularity of neural radiance fields (NeRFs), implicit representations also became popular for scene completion by predicting so-called density fields. Unlike explicit approaches e.g. voxel-based methods, density fields also allow for accurate depth prediction and novel-view synthesis via image-based rendering. In this work, we propose to fuse the scene reconstruction from multiple images and distill this knowledge into a more accurate single-view scene reconstruction. To this end, we propose Multi-View Behind the Scenes (MVBTS) to fuse density fields from multiple posed images, trained fully self-supervised only from image data. Using knowledge distillation, we use MVBTS to train a single-view scene completion network via direct supervision called KDBTS. It achieves state-of-the-art performance on occupancy prediction, especially in occluded regions.

1. Introduction

Obtaining accurate 3D geometry of a scene is crucial to achieving a holistic understanding of our surroundings. This knowledge enables a plethora of tasks both in robotics and autonomous driving. Reconstructing the scene's geometry from raw images, also called Structure-from-Motion [42], is a long-standing task in computer vision. Classical methods of scene reconstruction rely on multiple images and triangulation of matched keypoints, such as Harris Corner Detection, SIFT, and ORB [17, 34, 41], to infer the geometry of a scene. More recently, the breakthrough of deep learning gave rise to monocular depth estimation (MDE) methods [12, 13], that allowed to predict a dense per pixel depth from only a single image. These are either trained with ground truth supervision or in a self-supervised manner from video data [13, 14, 16, 35, 44, 49, 56, 57, 63]. Despite the impressive results of MDE for geometry estimation, the depth map representations are limited in their expressiveness. Projecting them into other frames will lead to holes or possibly wrong estimates due to occlusions in the scene. Furthermore, they cannot make any statements

[*] equal contribution.

about the scene beyond the visible surfaces.

Scene reconstruction broadens the task of scene representation from depth maps to reasoning about the whole geometry of a scene, including occluded regions [45] by doing shape completion. [45] introduced the task of semantic scene completion, which extends scene reconstruction to include semantic predictions. The 3D nature of both tasks results in many methods relying on voxel representations for their predictions that either use images [4, 30] or LiDAR/depth maps [40, 45] as input.

Neural implicit representations in the form of neural fields offer an alternative scene representation. Popularized by NeRFs, which models a scene's geometry and view-dependent colors in the weights of a neural network, they allow for photorealistic novel view synthesis. However, the original NeRF work [37] cannot generalize to different scenes and often suffers from inaccurate geometry [47]. These limitations were addressed by many follow-up works to NeRF [38, 47, 55]. Similarly, density fields [51] solely model the scene's geometry, addressing the geometric inconsistencies of NeRFs. The work Behind the Scenes (BTS) [51] uses density fields predicted from a single image for scene completion, while IBRnet and NeuRay leverage density predictions from multiple images together with images-based rendering for novel-view synthesis. MVBTS presents an extension of BTS from a single image to multiple images for more accurate scene reconstructions. In the absence of accurate 3D ground truth geometry, we leverage these scene reconstructions to directly supervise single-view density prediction models. We retrain the original BTS architecture in this pseudo-supervised setting as KDBTS to boost its performance. We release the code to further facilitate research.

Our **contributions** within this work are threefold:

- We extend the density field prediction of BTS to a multi-view setting trained in a fully self-supervised manner.
- We propose knowledge distillation training to directly supervise single-view density fields on fused multi-view predictions.
- Our method achieves state-of-the-art performance for both multi- and single-view occupancy prediction on the KITTI-360 benchmark.

2. Related Work

Neural Scene Representations Neural radiance field (NeRF) [37] has popularized the method of representing a scene with neural implicit functions. NeRF allows to query color and occupancy of a scene in continuous space by storing the scene within the weights of a neural network. This allows NeRF to do photorealistic novel view synthesis (NVS) of a scene. A downside of the original NeRF is the need to retrain the network weights for each scene. This was addressed by works such as PixelNeRF [55] and MINE [28] that construct neural radiance fields on the fly conditioned on sparse input views. [43] maps 2D image observations of a scene to a persistent 3D scene representation, disentangling movable and immovable components of the scene. Neo360 [21] uses a tri-planar representation together with foreground and background MLPs to allow for representing unbounded scenes.

However, most NeRF methods focus on tasks such as novel view synthesis, often leading to inaccurately reconstructed scene geometry. Works such as [47] address this by modeling effects such as reflections on surfaces.

Image Based Rendering Imaged-based rendering synthesizes images for novel views based on the color of surrounding reference images. Early methods focused on blending reference pixels based on geometry proxies [1, 8, 20]. Other works on light fields reconstruct 4D plenoptic functions from input views [7, 15]. More recent approaches are based on NeRF architectures. IBRnet [48] uses input images to construct a radiance field and density field on the fly. The density is predicted from pooled feature vectors in a ray transformer, sharing information between 3D points. The color is reconstructed as a weighted sum of the colors of surrounding views. GeoNeRF [22] uses cost volumes together with an attention-based aggregation for density prediction while replacing the ray transformer with an autoencoder. NeuRay [33] addresses the problem of inconsistent image features due to occlusions. Based on either cost volumes or depth maps, it predicts the visibility of 3D points in different views, leading to better renderings.

Scene Completion Scene completion, sometimes also called scene reconstruction, refers to the task of estimating the 3D geometry of a scene from images. Depending on the task, scene completion can range from depth prediction [12, 13, 36, 46, 50] to reconstructing even occluded parts of a scene [4, 51]. Depth prediction can either come from monocular images [13, 61] or stereo images [23, 58]. Especially MDE has been successfully used in monocular visual odometry applications [24, 54].

The information from depth prediction is restricted to the visible surfaces of a scene, leaving out important information about the shapes of objects. Given a depth map [10, 45, 52] complete shape geometry in an environment using voxel-grid representations. Voxlets [10] use random forests to predict signed distance values in the grid, whereas [52] models a probabilistic distribution on voxels. Generalizable NeRF methods such as PixelNeRF [55] and MINE [28] can also be used to complete the scene geometry. Recently, Behind the Scenes (BTS) [51] introduced a density field representation based on NeRF to model the scene geometry implicitly together with imaged-based rendering to explicitly learn to "look behind" objects.

Semantic scene completion (SSC), first introduced in [45], extends this task even further to the semantic setting.

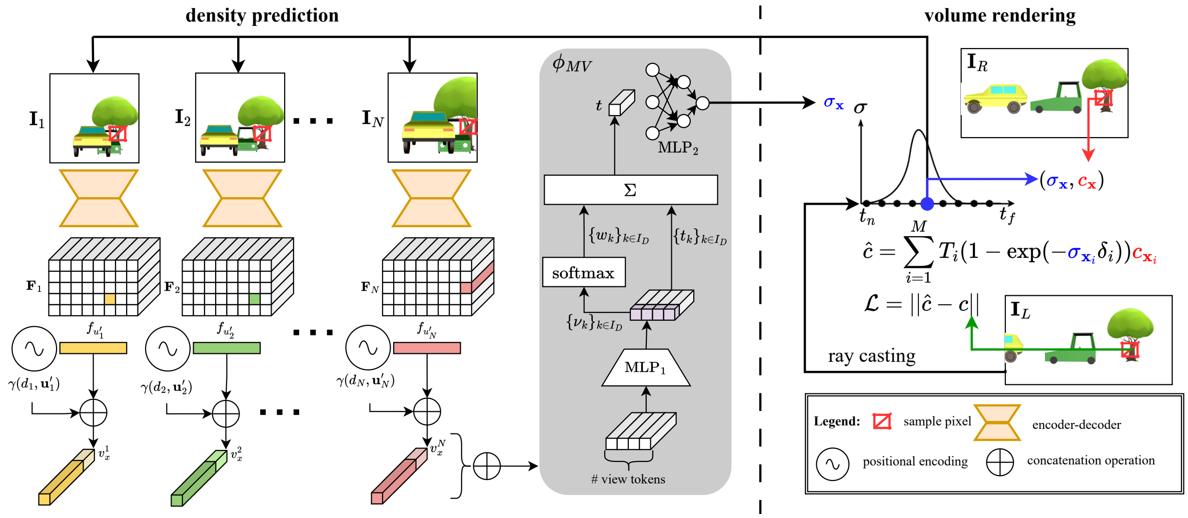

Figure 2. **Overview.** Given multiple input images $\mathbf{I}_k$ ($k \in I_D$) an encoder-decoder backbone predicts per image a pixel-aligned feature map $\mathbf{F}_k$ (top left). The feature $f_\mathbf{u}$ of pixel $\mathbf{u}$ encodes the occupancy and confidence distribution of a ray cast through pixel $\mathbf{u}$. Given a 3D point $\mathbf{x}$ and its projections $\mathbf{u}'_k$ into the different camera images, we extract the corresponding feature vectors and positional embeddings $\gamma(d, \mathbf{u})$. A multi-view network ϕ_{MV} decodes all feature vectors into a density prediction $\sigma_\mathbf{x}$ (middle). Together with color samples from another image ($\mathbf{I}_R$), this can be used to render novel views in an image-based rendering pipeline. $\mathbf{I}_R$ is not required to be close to the input images, as our method can predict density in occluded regions. See Fig. 3 for more details about the importance of covering the whole scene. We train our networks by using a photometric consistency loss of an image $\mathbf{I}_L$ close to $\mathbf{I}_R$ (right).

For occupied regions, semantic labels are predicted. In the beginning, SSC methods either focused on indoor scenarios with image data [2, 5, 25–27, 32, 59, 60] or outdoor scenarios with LiDAR data [6, 29, 39, 40, 53]. MonoScene [4] was the first method to tackle outdoor scenarios from image data by using line-of-sight projection of deep image features. VoxFormer [30] utilizes a transformer architecture with deformable attention layers. Both rely on voxels for their scene representation. S4C [18] uses an implicit scene representation and a self-supervised training approach to remove the need for annotated 3D data.

In this work, we build upon the ideas of scene completion methods from single images, BTS [51]. Specifically, to exploit all the available image data at inference time for more accurate occupancy predictions. We then leverage our accurate predictions to further improve single-image scene completion. This is in contrast to the works of [22, 33, 48] that focus on NVS and rely on close-by views to provide both color and density information.

3. Method

In the following, we briefly recap the Behind the Scenes (BTS) model as we extend its architecture and use it in our knowledge distillation scheme. We also cover training setup to predict the scene's geometry, even in occluded areas. We then introduce the extension of the BTS architecture to a multi-view setting and propose a knowledge distillation scheme for improved single-view predictions.

Notation We denote images by $\mathbf{I}_k \in [0,1]^{3\times H\times W} = (\mathbb{R}^3)^\Omega$ which are defined on a lattice $\Omega = \{1,\dots,H\} \times \{1,\dots,W\}$. The corresponding camera poses and projection matrices of the images are given as $T_i \in \mathbb{R}^{4\times4}$ and $K_i \in \mathbb{R}^{3\times4}$, respectively. The density of a point $\mathbf{x} \in \mathbb{R}^3$ in world coordinates is given by $\sigma_\mathbf{x}$. Its projection into an image k is given by $\pi_k(\mathbf{x}) = K_k T_k \mathbf{x}$ in homogeneous coordinates. From our set of all image indices $I = \{1,\dots,N\}$, we create the following subsets I_D, I_L, and I_R for density, loss, and render. For these subsets, the following holds:

$$I_D \subset I \tag{1}$$
$$I_L \cup I_R = I \tag{2}$$
$$I_L \cap I_R = \emptyset \tag{3}$$

We give examples together with a more detailed explanation for these splits in the supplementary material.

3.1. Prerequisites: Density Field from a Single View

BTS models a scene by predicting a volumetric density for a 3D point $\mathbf{x}$ conditioned on an input image $\mathbf{I}_\text{I}$ [51]. This is done by predicting a pixel-aligned feature map $\mathbf{F}_\text{I} \in (\mathbb{R}^C)^\Omega$ with an encoder-decoder architecture from the image. The point $\mathbf{x}$ is projected into the input image $\mathbf{u}'_\text{I} = \pi_\text{I}(\mathbf{x})$ to obtain the corresponding feature vector $f_{\mathbf{u}'} = \mathbf{F}(\mathbf{u}')$ from the feature map. A small multi-layer perceptron (MLP) ϕ_{SV} decodes the density from the feature vector as well a positional encoding $\gamma(d_\text{I}, \mathbf{u}'_\text{I})$ of the depth d_I and pixel position $\mathbf{u}'_\text{I}$ as

$$\sigma_\mathbf{x} = \phi_{SV}(f_{\mathbf{u}'_\text{I}}, \gamma(d_\text{I}, \mathbf{u}'_\text{I})) \tag{4}$$

The intuition behind this density prediction is that the feature vector of a pixel learns to describe the density distribution along the corresponding ray [51]. Together with the positional encodings, the MLP (ϕ_{SV}) decodes this distribution into a density prediction.

3.2. MVBTS: Density Field from Multiple Views

We extend the single-view density prediction by replacing the ϕ_{SV} with a confidence-based multi-view architecture

$$\sigma_{\mathbf{x}} = \phi_{MV}(\{f_{\mathbf{u}'_k}, \gamma(d_k, \mathbf{u}'_k)\}_{k \in I_D}), \tag{5}$$

that is able to aggregate the information from the $|I_D|$ images. In the following, we will break down how to predict a density value for a point $\mathbf{x}$ from multiple input images.

For each of the input images $\{\mathbf{I}_k\}_{k \in I_D}$, we predict a pixel-aligned feature map $\mathbf{F}_k$ with a shared encoder-decoder architecture. The point $\mathbf{x}$ is then projected into each of the input images $\mathbf{u}'_k = \pi_k(\mathbf{x})$, giving us $|I_D|$ feature vectors $f_{\mathbf{u}'_k} = \mathbf{F}_k(\mathbf{u}'_k)$. We keep track of valid projections into the frustum of a camera with a binary masking vector $\mathbf{m} \in \{0,1\}^{|I_D|}$. The feature vectors are concatenated with the respective positional encodings $\gamma(d_k, \mathbf{u}'_k)$ and fed through a small MLP

$$(\nu_k, t_k) = \mathrm{MLP}_1(f_{\mathbf{u}'_k}, \gamma(d_k, \mathbf{u}'_k)) \tag{6}$$

to produce both a confidence ν_k and a view-dependent features t_k. A softmax layer then converts the confidence values into weights

$$\{w_k\}_{k \in I_D} = \mathrm{softmax}(\{\nu_k\}_{k \in I_D}, \mathbf{m}), \tag{7}$$

that are then used in a weighted summation

$$t = \sum_{k \in I_D} w_k t_k \tag{8}$$

to produce a *view-aggregated* feature t. This feature is then decoded by a second MLP to produce the density estimate

$$\sigma_{\mathbf{x}} = \mathrm{MLP}_2(t). \tag{9}$$

For more details, see the left-hand side of Fig. 2.

3.3. Training via Volumetric Rendering

For convenience, this section recaps the training scheme from BTS [51], which we follow closely. As with BTS, this work aims to predict accurate 3D geometry with a continuous scene representation trained only on posed images in a self-supervised manner. We use our geometry prediction and image-based rendering techniques in a differentiable volumetric rendering pipeline to reconstruct images, allowing us to leverage a photometric reconstruction loss to train our networks. See the right-hand side of Fig. 2 for an overview of the volumetric rendering.

Color Sampling. Unlike most NeRF models, we do not predict color that is used to render novel views directly. Instead, we rely on other images to provide us with color samples. Given the point $\mathbf{x}$ and a frame $k \in I_R$, we obtain a color sample $c_{\mathbf{x},k}$ by projecting $\mathbf{x}$ into frame k and interpolating the color value of the image $c_{\mathbf{x},k} = \mathbf{I}_k(\mathbf{u}'_k)$.

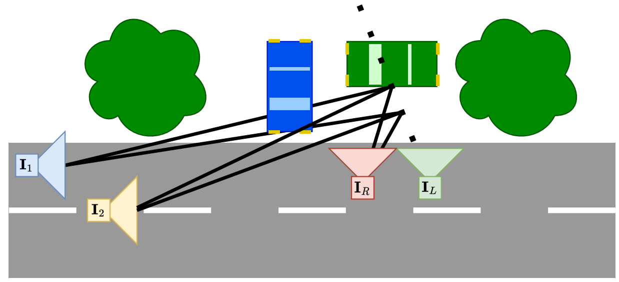

Figure 3. **Training Setup.** Given an input view ($\mathbf{I}_1$, blue) we want to reconstruct the scene, including partially occluded objects such as the green car. To learn to reconstruct both the free space behind the blue car and the surface of the green car, we try to render a pixel of the green view ($\mathbf{I}_L$) by casting a ray and sampling points on it (dotted line). We project the points into the blue view to estimate the density and into the red view ($\mathbf{I}_R$) to sample colors. The pixel of $\mathbf{I}_L$ is only reconstructed correctly if $\mathbf{I}_1$ estimates the correct density of the scene, although the green car and the free space next to it is occluded. In our multi-view method, we propose to use more views, *e.g.* the yellow view $\mathbf{I}_2$ to aggregate more information about the scene to better density predictions. In this example, the yellow view has a slightly better visibility on the green car.

Volumetric rendering. To render a pixel for NVS from an image $\mathbf{I}_L$, we cast a ray from the camera through the pixel. We sample M discrete points along the ray at positions $\mathbf{x}_i$. This discrete sampling lets us approximate the continuous volume rendering equation [37]. For each point, we extract its density $\sigma_{\mathbf{x}_i}$ from our multi-view density field and colors $c_{\mathbf{x}_i,k}$ from different images $k \in I_R$. We render the color of a pixel by aggregating the sampled colors over the density distribution on the ray

$$\hat{c}_k = \sum_{i=1}^{M} T_i \alpha_i c_{\mathbf{x}_i,k}, \tag{10}$$

with α_i, the probability of the ray ending between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, δ_i, the distance between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, and T_i, the accumulated transmittance, given by

$$\alpha_i = 1 - \exp(-\sigma_{\mathbf{x}_i}\delta_i) \qquad T_i = \prod_{j=1}^{i-1}(1-\alpha_j). \tag{11}$$

Based on the density and d_i, the distance between $\mathbf{x}_i$ and the ray origin, we can also calculate the expected ray termination depth

$$\hat{d} = \sum_{i=1}^{m} T_i \alpha_i d_i. \tag{12}$$

We want to note that this volumetric rendering reconstructs multiple color suggestions $\hat{c}_k$ for a single pixel based on the different frames k, where the color samples come from.

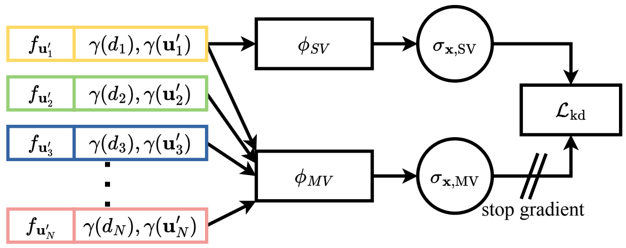

Figure 4. **Knowledge Distillation.** To improve the single-view (SV) density field reconstruction, we propose leveraging knowledge distillation from the multi-view (MV) predictions. Both ϕ_{SV} and ϕ_{MV} make use of the same encoder-decoder architecture and, therefore, the same feature vectors. The knowledge distillation loss $\mathcal{L}_{\text{kd}}$ pushes the ϕ_{SV} MLP to predict the same density as ϕ_{MV} while relying only upon a single feature vector. The stop gradient operator prevents $\mathcal{L}_{\text{kd}}$ influencing ϕ_{MV}.

A key insight of BTS was the setup of the cameras during training. Separating rendering, color sampling, and density prediction to different views allows one to learn the density even in occluded regions of the image. We follow this training setup, adapted to account for multiple input views; see Fig. 3 for more details on this.

Loss functions. We do not reconstruct individual pixels for training the networks, but rather patches of pixels P_i, randomly sampled from the loss images $\mathbf{I}_L$. Our reconstruction loss is a photometric-consistency loss, combining an L1 and an SSIM loss, following the strategy of [13, 51] of only selecting the per-pixel *minimum* over the reconstructions $\hat{P}_{i,k}$ when aggregating the costs.

$$\mathcal{L}_{\text{ph}} = \min_{k \in I_R} \left(\lambda_{\text{L1}} \text{L1}(P_i, \hat{P}_{i,k}) + \lambda_{\text{SSIM}} \text{SSIM}(P_i, \hat{P}_{i,k}) \right) \tag{13}$$

Additionally, we also employ an edge-aware smoothness loss term on $d^\star$ an inverse parameterized, mean-normalized version of $\hat{d}$,

$$\mathcal{L}_{\text{eas}} = |\delta_x d_i^\star| \, e^{-|\delta_x P_i|} + |\delta_y d_i^\star| \, e^{-|\delta_y P_i|} \tag{14}$$

to regularize the expected ray termination depth, giving us the final loss

$$\mathcal{L} = \mathcal{L}_{\text{ph}} + \lambda_{\text{eas}} \mathcal{L}_{\text{eas}} \, . \tag{15}$$

We give additional details for the training on the different datasets in Sec. 4.

3.4. KDBTS: Knowledge Distillation for Single-View Reconstruction

Given the scene reconstructions from our multi-view model, we want to distill this knowledge into a single-view prediction model that can reconstruct the scene from fewer input data, *e.g*. a single image. We use the original BTS model architecture for our KDBTS. The motivation behind our knowledge distillation approach is threefold. First, it reduces the network size, as the decoder is slightly smaller,

Dataset	Split.	# Train	# Val	# Test
KITTI	Eigen [9]	39810	4424	697
KITTI-360	BTS [51]	98008	11451	446

Table 1. **Dataset Split Overview.** Number of images in each split of the datasets. Evaluation is performed exclusively on the test splits.

speeding up the inference time. Second, removing the necessity of pose information at inference time. The advantage of single-view reconstruction is that it neither requires a calibrated stereo camera setup nor relative pose information from visual odometry systems to reconstruct the scene. Third, it leverages the advantages of self-supervised and supervised training. MVBTS can be trained on image data alone, allowing it to scale to vast amounts of data. This allows our method to be employed in the absence of ground truth data and still provide direct supervision by generating a pseudo ground truth. This knowledge distillation scheme is illustrated in Fig. 4.

Given the input images $\mathbf{I}_\text{I}$ and $\{\mathbf{I}_k\}_{k \in D}$ for single- and multi-view, respectively, $\sigma_{\mathbf{x},\text{SV}} = \phi_{SV}(f_{\mathbf{u}'_\text{I}}, \gamma(d_\text{I}, \mathbf{u}'_\text{I}))$ and $\sigma_{\mathbf{x},\text{MV}} = \phi_{MV}(\{f_{\mathbf{u}'_k}, \gamma(d_k, \mathbf{u}'_k)\}_{k \in I_D})$ should give the same density prediction for a 3D point $\mathbf{x}$. In practice, we choose the input view $\mathbf{I}_\text{I}$ for the single-view head to be from $\{\mathbf{I}_k\}_{k \in D}$, but this is not strictly necessary. Knowledge distillation is then performed via a simple L1 loss

$$\mathcal{L}_{\text{kd}} = \|\sigma_{\mathbf{x},\text{MV}} - \sigma_{\mathbf{x},\text{SV}}\| \, . \tag{16}$$

To avoid the multi-view head being influenced by the single-view prediction, we apply a stop gradient operator on $\sigma_{\mathbf{x},\text{MV}}$ to treat it as a pseudo ground truth.

4. Experiments

We demonstrate the advantages of having multiple views to predict density fields in the tasks of *depth prediction* and *occupancy estimation*. We evaluate both our multi-view model (MVBTS) as well as our single-view model (KDBTS) boosted by knowledge distillation. We follow the evaluation of [51] and evaluate the depth prediction on the KITTI [11] dataset against the ground truth depth and occupancy estimation on KITTI-360 [31] against occupancy maps from aggregated LiDAR scans over multiple time steps.

4.1. Datasets and Training Details

KITTI [11] and KITTI-360 [31] are both autonomous driving datasets that provide time-stamped stereo images with ground truth poses - although for KITTI we use poses generated by ORB-SLAM 3 [3]. KITTI-360 also provides fisheye cameras looking to the sides. For our training, they are rectified to the same camera intrinsics as the stereo cameras. A randomly applied dropout of 0.5, helps our model

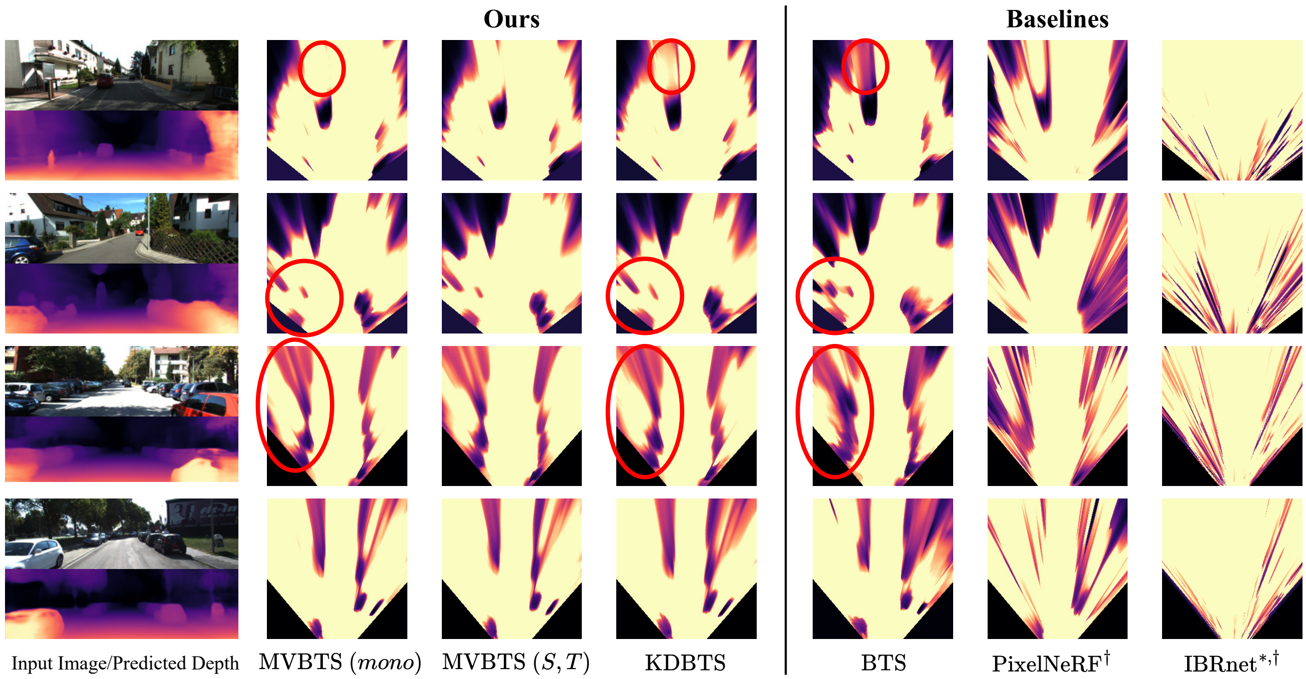

Figure 5. **Density Fields.** Top-down rendering of the density fields for an area of $x = [-9m, 9m]$, $y = [0m, 1m]$, $z = [3m, 23m]$. Images are taken from KITTI-360 (top half) and KITTI (bottom half) with profiles coming from models trained on KITTI-360. Every model except for MVBTS (S, T) and IBRnet [48] get the same input data. Our MVBTS can predict accurate geometry even in distant regions for both a single image and multiple images. KDBTS learns to recreate the accurate density structure from MVBTS. Both models reduce the amount of shadows produced by BTS [51], especially in distant regions. They also produce cleaner boundaries for close-by objects. Note that KDBTS has a smaller model capacity than MVBTS $(mono)$. $*$: changed sensitivity for visualization purposes, †: retrained on KITTI-360.

to not overfit on a specific camera setup. During training, we randomly select three stereo input views to be in I_D.

We train our model on a single 48GB Nvidia RTX A40 GPU with half the batch size of BTS, 8, due to increased memory demand from the multiple views. We sample 32 patches of size 8×8 with 64 points per ray for possible reconstruction. The encoder-decoder follows [51] to be a ResNet50 encoder [19] with 64 output channels. MLP_1 and MLP_2 are chosen to be two-layer networks with residual connections. MLP_1 therefore corresponds to a larger version of the decoder network of BTS, as it needs to decode additional information. For detailed hyperparameters, please refer to the supplementary material.

We train for 150,000 steps on KITTI and for 200,000 steps on KITTI-360, compensating for our smaller batch size. We train on the splits detailed in Tab. 1.

In our experimental setup, both KDBTS and MVBTS share a backbone architecture. To train the KDBTS model, we use a trained MVBTS model and freeze the encoder-decoder backbone as well as ϕ_{MV}. We found training both in parallel to give a slightly decreased performance. To additionally boost the performance in distant parts of the scene, we randomly sample the timestamp offset of the additional stereo frames between $0.1s$ and $0.8s$. As direct supervision and a shared backbone lead to a strong training signal, we train the knowledge distillation for only 20,000 steps.

4.2. Depth Prediction

Although a byproduct of the density field, depth estimates provide useful information leveraged by downstream tasks, such as segmentation. In Tab. 2, we compare our two methods against self-supervised depth prediction methods and volume reconstruction methods. Fig. 6 shows qualitative results on KITTI. For depth prediction, we keep the same training setup for KITTI so that we report the values for PixelNeRF [55], MonoDepth2 [13], DevNet [62], and BTS [51] from [51]. For more single-view methods, please refer to [51].

4.3. Occupancy Prediction

The main evaluation of our method is with regard to the occupancy prediction. Next to our method, we evaluate with PixelNeRF [55] and the original BTS [51] two additional single-view reconstruction methods. For multi-view methods, we compare against another image-based rendering method, IBRnet [48] retrained on KITTI-360. Fig. 5 shows top-down renderings of occupancy predictions both on KITTI and KITTI-360 made with models trained on KITTI-360. They show qualitative results for difficult sce-

Model	Volum.	Abs Rel ↓	RMSE ↓	$\alpha < 1.25$ ↑
PixelNeRF [55]	✓	0.130	5.134	0.845
MonoDepth2 [13]	✗	0.106	4.750	0.874
DevNet [62]	(✓)	**0.095**	**4.365**	**0.895**
BTS [51]	✓	<u>0.102</u>	<u>4.407</u>	<u>0.882</u>
KDBTS (**Ours**)	✓	0.105	4.497	0.873
MVBTS (**Ours**)	✓	0.105	4.501	0.873

Table 2. **Monocular Depth Evaluation.** Our method performs very similar to BTS due to the similarity of the architecture. We suspect that the slight differences in the presented metrics arise from some of the model capacity being shifted to better predict the occluded parts of the scene.

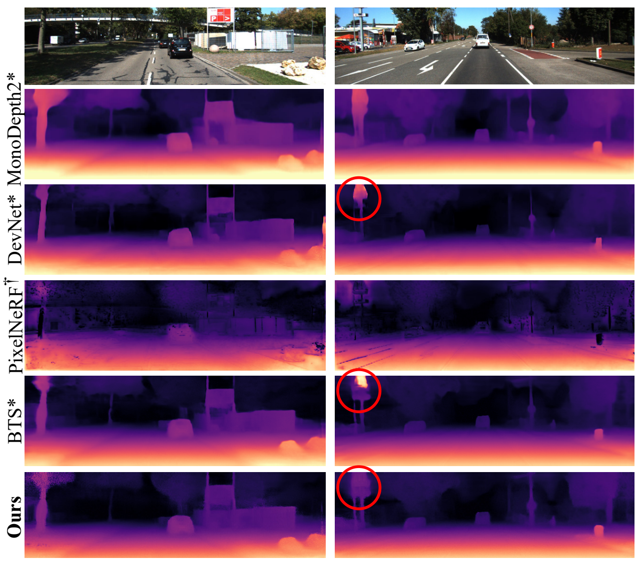

Figure 6. **Monocular Depth Prediction.** Qualitative comparison with state-of-the-art monocular depth prediction and other volumetric methods. The expected ray termination depth $\hat{d}$ gives a detailed scene reconstruction.

narios. For IBRnet, we increased the sensitivity of the visualization as the density values were, on average, significantly lower than for the other methods. Our models reason better in far-away parts of the scene (see first example). Additionally, they produce cleaner occupancy edges in occluded areas of the scene, leading to fewer artifacts. For quantitative evaluation, we follow [51] with carving out free space based on LiDAR scans. We evaluate accuracy, precision, and recall for two settings: 1. Occupancy predictions (O_{acc}, O_{prec}, O_{rec}) and 2. Invisible and empty (IE_{acc}, IE_{prec}, IE_{rec}), giving us six metrics in total. Tab. 3 details the results on KITTI-360. Unlike BTS [51], we also choose to include the precision and recall for the occupancy due to the large imbalance between occupied and empty space. As the distribution is heavily skewed to the empty space, achieving a high accuracy score is easy, as can be seen for IBRnet. On KITTI-360, IBRnet seems to give fog-like predictions with few clear boundaries, as can be seen in Fig. 5, resulting in low precision and recall. We also evaluate IBRnet in the depth + $4m$ setting to address this. This reduces the IE_{rec} but gives competitive other metrics. Our models, both MVBTS and KDBTS, are evaluated in the single-view setting in the top half of Tab. 3. Both methods consistently improve the original BTS across most metrics. Despite the lower model capacity, KDBTS performs on par with MVBTS, validating our knowledge distillation approach. In the multi-view setting, where the models have access to four frames - stereo and temporal -, MVBTS performs the most consistently across all metrics. IBRnet has a better O_{acc} and, depending on the prediction mode, either a slightly better O_{rec} or significantly better IE_{rec}. We suspect this results from the large imbalance of empty to occupied space. MVBTS either performs on par or significantly better than IBRnet. We present additional evaluation results in the supplementary material.

4.4. Ablation Studies

Tab. 4 gives an overview of the impact different design choices in our architecture and training scheme have on the performance of our model. Our ablation study is split into two parts. In the first part, we are concerned with the architecture of ϕ_{MV}, whereas ablations on the backbone have already been done in [51]. The second part investigates the influence of the number of available views on the performance of the model. For our ablation study, we use our final model as a baseline and report the differences to this model in the table. We investigate the following: adding attention layers similar to GeoNeRF [22] for enhanced information sharing between views; also including fisheye views in I_D; the effects of different dropout rates; the size of the MLPs. In general, our method shows robust performance in the ablation settings. Attention layers allow for information sharing between tokens and can implicitly learn to shift focus to certain views. However, within this architecture, they seem to hinder the model's performance. We detail a possible explanation together with examples in the supplementary material. Including fisheye cameras leads to a mostly on-par performance. This is likely due to most sampled points lying outside of the fisheye cameras' frustums. Less dropout leads to worse generalization capabilities in the network, which can especially be seen in the monocular setting reported in the supplementary material. Directly summing densities from different images (MLPs: small) also leads to a decreased performance. An evaluation of the number of images at inference time concludes the ablation study. Including both stereo and temporal frames seems to give a similar boost in performance, whereas having both available results in the most accurate predictions.

5. Discussion

Our method for accurately reconstructing density fields relies on several assumptions. In the following, we will

Method	Multi View	O_{acc} ↑	O_{prec} ↑	O_{rec} ↑	IE_{acc} ↑	IE_{prec}	IE_{rec} ↑
PixelNeRF [55]	✗	93.82%	51.94%	69.43%	61.33%	37.86%	42.21%
BTS [51]	✗	94.47%	58.73%	84.24%	77.04%	**54.21%**	43.99%
MVBTS (**Ours**)	✗	**94.76%**	**60.83%**	84.51%	78.00%	53.69%	**44.04%**
KDBTS (**Ours**)	✗	**94.76%**	60.68%	**84.78%**	**78.30%**	53.62%	44.00%
IBRnet [48]	✓	96.03%	4.14%	4.78%	34.36%	32.97%	**96.02%**
IBRnet (depth + 4m) [48]	✓	**98.12%**	43.35%	**86.01%**	59.67%	25.18%	9.85%
MVBTS (**Ours**)	✓	94.91%	**61.73%**	85.78%	**79.47%**	**55.08%**	45.23%

Table 3. **Occupancy Prediction on KITTI-360.** We follow the evaluation setup of BTS [51] and evaluate both the occupancy of the whole scene (O) and only the invisible and empty parts (IE). In both cases, we report accuracy, precision, and recall, as there is a large imbalance between occupied and empty parts of the scene. We evaluate MVBTS and KDBTS in a single-view and MVBTS additionally in a multi-view setting. KDBTS can improve upon the original BTS consistently. Due to IBRnet [48] predicting little density (see Fig. 5), we also evaluate IBRnet in the setting of depth + 4m, giving better occupancy precision and recall.

Inference	dropout	MLPs	attn. layers	encode fisheye	O_{acc} ↑	O_{prec} ↑	O_{rec} ↑	IE_{acc} ↑	IE_{prec}	IE_{rec} ↑
(*S*, *T*)	0.5	middle	✓	✗	-1.67%	-0.96%	-15.09%	-9.64%	-7.67%	13.70%
(*S*, *T*)	0.0	middle	✗	✓	-1.47%	-4.88%	-6.04%	-6.45%	-9.54%	-1.33%
(*S*, *T*)	0.2	middle	✗	✓	-0.34%	-1.88%	-2.58%	-2.34%	-0.67%	0.41%
(*S*, *T*)	0.5	middle	✗	✓	0.02%	-1.01%	-0.31%	-0.42%	0.65%	1.16%
(*S*, *T*)	0.8	middle	✗	✓	0.04%	-1.30%	0.62%	-0.06%	0.27%	-0.43%
(*S*, *T*)	0.5	small	✗	✗	-0.26%	-2.09%	0.86%	-0.93%	-0.04%	-1.53%
(*S*, *T*)	0.5	large	✗	✗	-0.53%	-1.84%	-3.27%	-2.23%	-3.41%	1.66%
(*Mono*)	0.5	middle	✗	✗	-0.15%	-0.90%	-1.27%	-1.47%	-1.38%	-1.19%
(*S*)	0.5	middle	✗	✗	-0.07%	-0.61%	-0.50%	-0.85%	-1.06%	-1.19%
(*T*)	0.5	middle	✗	✗	-0.09%	-0.77%	-0.88%	-0.63%	-1.01%	-0.04%
(*S*, *T*)	0.5	middle	✗	✗	94.91%	61.73%	85.78%	79.47%	55.08%	45.23%

Table 4. **Ablation Studies.** To motivate the choice of our architecture for MVBTS, we detail the results for different settings. We also report the influence of additional images at inference time. Overall, the architecture is robust against the most hyperparameter settings. The ablation study shows that keeping the multi-view aggregation simple. We specifically address the use of attention layers together with a detailed breakdown of the different settings in the supplementary material.

discuss these assumptions and their possible effects on the reconstruction quality. As with many NeRF or scene reconstruction methods - with the exception of dynamic NeRFs, *etc.*-, our method uses the static scene assumption. However, datasets such as KITTI contain moving objects in the form of cars, pedestrians, and cyclists. Dynamic objects influence both the training of MVBTS and KDBTS during knowledge distillation, possibly resulting in drawn-out shadows of objects. We share this limitation with other methods that require images from multiple time steps during training and inference. Similar to the static scene assumption, sampling color assumes photo consistency for reconstruction that does not model view-dependent effects of different materials. Methods such as IBRnet and Neuray[33] are able to model these effects, resulting in better novel view synthesis. However, we argue that this leads to a worse geometric reconstruction as shown in Tab. 3 with PixelNeRF and IBRnet. More images during inference time give MVBTS more computational overhead compared to BTS. Additionally, our multi-view model needs - at least - relative camera poses at inference time as well. Our single-view model can address these last limitations while simultaneously giving better occupancy predictions than comparable models such as BTS.

6. Conclusion

We introduce a novel approach to enhancing single-view geometric scene reconstruction by leveraging multi-view information. This involves extending a state-of-the-art density prediction model to improve scene geometry, followed by direct supervision in 3D via knowledge distillation to boost a single-view model. Training is done fully self-supervised on video data. We evaluate both the proposed multi-view and boosted single-view models on depth estimation and occupancy prediction tasks. While our method performs close to state-of-the-art for depth estimation, being outperformed by methods explicitly trained for this task, our boosted single-view reconstruction model consistently achieves state-of-the-art performance for occupancy prediction. Future work on modeling moving objects can address conflicting information in dynamic scenes, improving overall accuracy and reliability in 3D reconstructions.

Acknowledgements. This work was supported by the ERC Advanced Grant SIMULACRON, by the Munich Center for Machine Learning, and by the German Federal Ministry of Transport and Digital Infrastructure (BMDV) under grant 19F2251F for the ADAM project.

References

[1] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. *SIGGRAPH*, 2001. 2

[2] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 324–333, 2021. 3

[3] 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. 5

[4] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 2, 3

[5] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4193–4202, 2020. 3

[6] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Liu Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *Conference on Robot Learning*, pages 2148–2161. PMLR, 2021. 3

[7] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, pages 305–314. Wiley Online Library, 2012. 2

[8] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. *Computer graphics and interactive techniques*, 1996. 2

[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. 5

[10] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. 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. 5

[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. 1, 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, 5, 6, 7

[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

[15] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 1996. 2

[16] 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

[17] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 1988. 1

[18] Adrian Hayler, Felix Wimbauer, Dominik Muhle, Christian Rupprecht, and Daniel Cremers. S4c: Self-supervised semantic scene completion with neural fields. *arXiv preprint arXiv:2310.07522*, 2023. 3

[19] 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. 6

[20] Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and Luc J. Van Gool. Plenoptic modeling and rendering from image sequences taken by hand-held camera. In *Mustererkennung 1999, 21. DAGM-Symposium*, 1999. 2

[21] Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Vitor Guizilini, Thomas Kollar, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Neo 360: Neural fields for sparse view synthesis of outdoor scenes. In *ICCV*, 2023. 2

[22] M. Johari, Y. Lepoittevin, and F. Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 7

[23] Sameh Khamis, Sean Fanello, Christoph Rhemann, Adarsh Kowdle, Julien Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[24] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Conference on Robot Learning*, pages 34–45. PMLR, 2022. 2

[25] Jie Li, Yu Liu, Dong Gong, Qinfeng Shi, Xia Yuan, Chunxia Zhao, and Ian Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2019. 3

[26] Jie Li, Yu Liu, Xia Yuan, Chunxia Zhao, Roland Siegwart, Ian Reid, and Cesar Cadena. Depth based semantic scene completion with position importance aware loss. *IEEE Robotics and Automation Letters*, 5(1):219–226, 2019.

[27] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3351–3359, 2020. 3

[28] 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. 2

[29] Pengfei Li, Yongliang Shi, Tianyu Liu, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Semi-supervised implicit scene completion from sparse lidar, 2021. 3

[30] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *CVPR*, 2023. 2, 3

[31] 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. 5

[32] Shice Liu, Yu Hu, Yiming Zeng, Qiankun Tang, Beibei Jin, Yinhe Han, and Xiaowei Li. See and think: Disentangling semantic scene completion. In *Advances in Neural Information Processing Systems*, 2018. 3

[33] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *CVPR*, 2022. 2, 3, 8

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 1

[35] 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

[36] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *AAAI*, pages 2294–2301, 2021. 2

[37] 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. 2, 4

[38] 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

[39] Christoph B Rist, David Emmerichs, Markus Enzweiler, and Dariu M Gavrila. Semantic scene completion using local deep implicit functions on lidar data. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7205–7218, 2021. 3

[40] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, 2020. 2, 3

[41] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 1

[42] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[43] Prafull Sharma, Ayush Tewari, Yilun Du, Sergey Zakharov, Rares Andrei Ambrus, Adrien Gaidon, William T. Freeman, Fredo Durand, Joshua B. Tenenbaum, and Vincent Sitzmann. Neural groundplans: Persistent neural scene representations from a single image. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[44] 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

[45] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017. 2

[46] Jaime Spencer, Richard Bowden, and Simon Hadfield. Defeat-net: General monocular depth via simultaneous unsupervised representation learning. In *CVPR*, pages 14402–14413, 2020. 2

[47] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2

[48] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2, 3, 6, 8

[49] 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

[50] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6112–6122, 2021. 2

[51] Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. Behind the scenes: Density fields for single view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9076–9086, 2023. 2, 3, 4, 5, 6, 7, 8

[52] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2

[53] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point

cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3101–3109, 2021. 3

[54] 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

[55] 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, 6, 7, 8

[56] 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

[57] 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

[58] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019. 2

[59] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 733–749, 2018. 3

[60] Pingping Zhang, Wei Liu, Yinjie Lei, Huchuan Lu, and Xiaoyun Yang. Cascaded context pyramid for full-resolution 3d semantic scene completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7801–7810, 2019. 3

[61] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. In *BMVC*, 2021. 2

[62] 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. 6, 7

[63] 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