P3Depth: Monocular Depth Estimation with a Piecewise Planarity Prior

Vaishakh Patil[1]   Christos Sakaridis[1]   Alexander Liniger[1]   Luc Van Gool[1,2]

[1]Computer Vision Lab, ETH Zürich   [2]PSI, KU Leuven

Abstract

Monocular depth estimation is vital for scene understanding and downstream tasks. We focus on the supervised setup, in which ground-truth depth is available only at training time. Based on knowledge about the high regularity of real 3D scenes, we propose a method that learns to selectively leverage information from coplanar pixels to improve the predicted depth. In particular, we introduce a piecewise planarity prior which states that for each pixel, there is a seed pixel which shares the same planar 3D surface with the former. Motivated by this prior, we design a network with two heads. The first head outputs pixel-level plane coefficients, while the second one outputs a dense offset vector field that identifies the positions of seed pixels. The plane coefficients of seed pixels are then used to predict depth at each position. The resulting prediction is adaptively fused with the initial prediction from the first head via a learned confidence to account for potential deviations from precise local planarity. The entire architecture is trained end-to-end thanks to the differentiability of the proposed modules and it learns to predict regular depth maps, with sharp edges at occlusion boundaries. An extensive evaluation of our method shows that we set the new state of the art in supervised monocular depth estimation, surpassing prior methods on NYU Depth-v2 and on the Garg split of KITTI. Our method delivers depth maps that yield plausible 3D reconstructions of the input scenes. Code is available at: https://github.com/SysCV/P3Depth

1. Introduction

Depth estimation is a fundamental problem in computer vision. It consists in predicting the perpendicular coordinate of the 3D point depicted at each pixel. Applications range from robotics to autonomous cars. There is experimental evidence [84] that depth is the most vital vision-level cue for executing actions, together with semantic segmentation. In this work, we focus on monocular depth estimation, which involves the challenge of scale ambiguity, as the same input image can be generated by infinitely many 3D scenes.

The current trend in solving this task involves fully convolutional neural networks that output a dense depth predic-
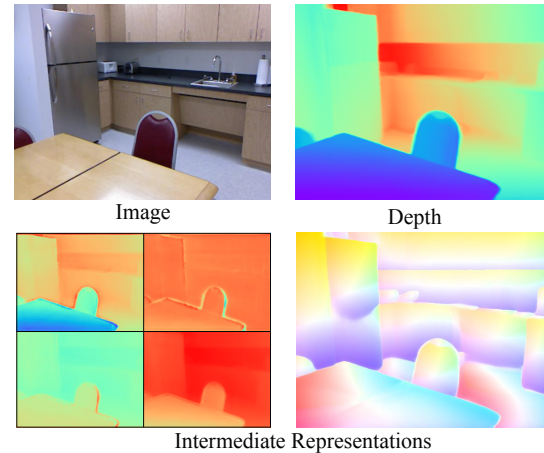

Figure 1. Real-world 3D scenes have a high degree of regularity. We propose a method which can exploit this regularity, by implicitly learning intermediate representations that contain useful information about local planes in the scene. The proposed end-to-end model predicts high-quality depth maps with sharp edges at occlusion boundaries, which yield consistent 3D reconstructions.

tion either with standard supervision on depth [9, 12, 32, 71] or with self-supervision by using the predicted depth to reconstruct neighboring views of the scene [16, 18, 19, 85]. Most supervised approaches use a pixel-level loss which treats predictions at different pixels separately. This regime ignores the high degree of *regularity* of real-world 3D scenes, which generally yield *piecewise smooth* depth maps.

A common choice for modeling this prior knowledge of the geometry of real 3D scenes are planes [2, 6, 41, 42]. Planes are the local first-order Taylor approximation for locally differentiable depth maps and they are easy to parameterize using three independent coefficients. Once a pixel is associated with a plane, its depth can be recovered from the position of the pixel and the coefficients of the associated plane. In [83], such a plane coefficient representation is used to learn to predict planes explicitly.

We adopt the plane representation from [83], but we depart from the explicit prediction of planes and rather use this representation as an appropriate output space for defining interactions between pixels based on planarity priors. In particular, the first head of our network outputs dense

plane coefficient maps which are afterwards converted to depth maps, as shown in Fig. 2. Predicting plane coefficients is motivated by the fact that two pixels $\mathbf{p}$ and $\mathbf{q}$ that belong to the same plane ideally have equal plane coefficient representations, whereas they generally have different depth. Thus, using the plane coefficient representation of $\mathbf{q}$ for predicting depth at the position of $\mathbf{p}$ results in a correct prediction if the pixels belong to the same plane.

We leverage this property by learning to identify *seed pixels* which share the same plane as the examined pixel, whenever such pixels exist, in order to selectively use the plane coefficients of these pixels for improving the predicted depth. This approach is motivated by a piecewise planarity prior which states that for each pixel $\mathbf{p}$ with an associated 3D plane, there is a seed pixel $\mathbf{q}$ in the neighborhood of $\mathbf{p}$ which is associated with the same 3D plane as $\mathbf{p}$. To predict depth with this scheme, we need to identify (i) the regions where the prior is valid and (ii) the seed pixels in these regions, by predicting the offsets $\mathbf{q} - \mathbf{p}$. We thus design a second head in the network, which outputs a dense offset vector field and a confidence map, as shown in Fig. 2. The predicted offsets are used to resample the plane coefficients from the first head and generate a second depth prediction. The depth predictions from the two heads are then fused adaptively using the confidence map as fusion weights, in order to down-weigh the offset-based prediction and rely primarily on the basic depth prediction in regions where the piecewise planarity prior is not valid, e.g. on parts of the scene with high-frequency structures. Supervision on the offsets and confidence map is applied implicitly, by supervising the fused depth prediction. Thanks to using seed pixels for prediction, our model implicitly learns to group pixels based on their membership in smooth regions of the depth map. This helps preserve sharp depth discontinuities, as shown in Fig. 1. Last but not least, we propose a mean plane loss which enforces first-order consistency of our predicted 3D surfaces with the ground truth and further improves performance.

We evaluate our method extensively on 6 datasets for supervised monocular depth estimation: NYU Depth-v2, KITTI, ScanNet, SUN-RGBD, DIODE Indoor, and ETH-3D. Comparisons to competing approaches demonstrate that we set a new state of the art on NYU Depth-v2 and KITTI, surpassing the former best-performing method in all commonly used evaluation metrics on NYU and on the Garg split [16] of KITTI. Moreover, in a challenging zero-shot transfer setup, we outperform the prior state of the art on ScanNet, SUN-RGBD, DIODE Indoor, and ETH-3D. We conduct a thorough ablation study and show quantitatively the merit of our novel formulation for depth prediction. We also provide qualitative comparisons with the prior state of the art, which evidence the high quality of our predictions, in particular when the latter are used for 3D reconstruction.

2. Related Work

Supervised monocular depth estimation assumes that ground-truth depth maps are available for training images and requires inference on single images. A notable early approach is Make3D [59], which explicitly handcrafts a piecewise planar structure on the scene and learns the associated parameters locally using a Markov random field. The multiscale network of [9] pioneered the usage of deep CNNs in depth estimation by learning an end-to-end mapping from images to depth maps. Several works have afterwards focused on this setting, proposing i.a. (i) more advanced architectures such as residual networks [32], convolutional neural fields [43, 73], fusion of multiple scales in the frequency domain [34], transformer-based blocks that attend to global depth statistics [1] and depth merging networks for handling multiple resolutions [50], (ii) losses that are better suited for depth prediction such as reverse Huber loss [32], classification loss [3], ordinal regression loss [12], pairwise ranking loss [71] and adaptive combinations of several depth-related losses [35], and (iii) joint learning of depth with normals or semantic labels [8, 53, 72]. The ambiguity in depth shift and focal length scale in mixed-data setups is addressed in [78] by applying 3D point cloud encoders to the lifted depth map. Our method belongs to this category and casts the depth prediction to a more appropriate space for exploiting regularities of input scenes.

Other depth estimation setups include unsupervised and semi-supervised monocular depth estimation as well as stereo-based depth estimation. Unsupervised learning of depth with stereo pairs based on novel view synthesis [10] that uses an image reconstruction loss in which the predicted depth is used for warping one image of the pair to the frame of the other was introduced in [16] and was cast in a fully differentiable formulation in [18]. Further works in this direction leverage temporal information [47, 52, 82]. The need for stereo pairs in this framework was lifted in [85], which operates on monocular videos. Consistency of the estimated 3D structures and of ego-motion across video frames is enforced in [27, 40, 48]. Depth and ego-motion are combined with optical flow and semantics in [26, 79] and with edges in [76]. Robustness to occlusions across video frames is achieved in [19] with a minimum reprojection loss. The optimization is facilitated with specialized losses in [20, 61]. Recent methods exploit video input at test time [69], consistency to segmentation outputs [86] and scale consistency across adjacent frames [67]. Unsupervised approaches generally assume more complex training data than supervised ones and suffer from scale ambiguity and violations of the Lambertian assumption. Semisupervised depth estimation is introduced in [29], which combines sparse depth measurements with an image reconstruction loss. Dataset-specific assumptions on the pres-

ence and the format of depth supervision are also relaxed in [39], which utilizes multi-view image collections for generating reliable large-scale depth supervision, and in [56], where diverse datasets providing different forms of supervision for monocular depth estimation are leveraged to generalize better on unseen data. Early stereo methods rely on hand-crafted matching costs [22] for estimating disparity. Initial approaches that learned the matching function include [30, 65], while subsequent works rely on fully convolutional architectures [5, 49]. Stereo methods also assume more complex data in the form of stereo pairs both at training and testing, which prevents their application to more general and uncontrolled monocular settings.

Geometric priors for depth have been extensively studied in the literature. In particular, the piecewise planarity prior has been traditionally used in multi-view stereo [14] and 3D reconstruction [2, 6] in order to make these problems amenable to faster optimization. These approaches involve explicit depth planes and fit these planes on image superpixels or point sets from input point clouds. Superpixel-level depth planes are also leveraged in depth denoising and completion [58, 66]. In more recent, deep learning-based approaches, the incorporation of geometric priors is performed either explicitly by segmenting planes [36, 41, 42, 81] or implicitly by properly designing the loss [80]. Non-local 3D context is leveraged in the virtual normal framework of [77] by using supervision from virtual planes which correspond to triplets of non-collinear points of the depth map. A non-local coplanarity constraint is embedded to the network in [23] via a depth-attention volume. Surface normals are used in [46] to increase geometric consistency on regular structures. A representation directly associated with coefficients of 3D planes in the image space without dependence on intrinsic camera parameters is used in [74, 83] for estimating the dominant depth planes in the scenes. The same representation with plane coefficients is employed in [33] to guide the upsampling modules of the decoder part of depth networks, achieving state-of-the-art performance. We also use this representation with plane coefficients, but contrary to [74, 83], we learn it without requiring annotations for planes. Instead, we optimize plane coefficients together with spatial offset vectors to learn to identify coplanar pixels and use this coplanarity for predicting depth. While offset vectors are also used in [54] for post-processing depth by merely resampling the prediction, we incorporate offset vectors in a single end-to-end architecture and generate the prediction via interpolation with the plane associated with the seed pixel pointed by the offset. Our approach is loosely inspired by [51], which trains offset vectors to identify instance segmentation centers from annotated images, while we focus on depth prediction and operate without supervision for plane instances.

3. Method

As pointed out in Sec. 1, our network estimates depth by selectively combining depth from each pixel and its corresponding seed pixel. For this formulation to work, it is vital to use a common representation which can capture pixelwise depth as well as planarity information. We achieve this by using a plane coefficient representation similar to [83]. We explain this representation and derive an analytical relation between plane coefficients and depth in Sec. 3.2, which allows us to supervise the network only with depth. The main advantage of the plane coefficient representation is that the depth of a pixel in the image can be directly computed by the plane coefficients of a different pixel, under the assumption that the two pixels are on the same plane. This advantage forms the basis of our planarity prior and the selective planar depth bootstrapping using seed pixels, which we explain in Sec. 3.3. Finally, in Sec. 3.4 we present an additional patch-level mean plane loss, which is complementary to the previous components and contributes independently to learning regular depth maps.

3.1. Preliminaries

Monocular depth estimation requires learning a dense mapping $f_\theta : I(u, v) \to D(u, v)$, where I is the input image with spatial dimensions $H \times W$, D is the corresponding depth map of the same resolution, (u, v) are pixel coordinates in the image space and θ are the parameters of the mapping f. In the supervised setup, a ground-truth depth map D^* is available for each image I at training time. During training, the parameters θ are optimized such that the function f_θ minimizes the difference between the predicted depth and the ground-truth depth over the training set $\mathcal{T}$. This can be formalized as

$$\min_\theta \sum_{(I, D^*) \in \mathcal{T}} \mathcal{L}(f_\theta(I), D^*), \qquad (1)$$

where $\mathcal{L}$ is a loss function that penalizes deviations between the prediction and the ground truth. Furthermore, given a depth map D along with the camera intrinsics, we can backproject each pixel to the 3D space. Using the pinhole camera model and given the focal lengths (f_x, f_y) and the principal point (u_0, v_0), every pixel $\mathbf{p} = (u, v)^T$ is mapped to a 3D point $\mathbf{P} = (X, Y, Z)^T$ according to

$$Z = D(u, v), \ X = \frac{Z(u - u_0)}{f_x}, \ Y = \frac{Z(v - v_0)}{f_y}. \quad (2)$$

3.2. Plane Coefficient Representation for Depth

Suppose that the backprojected 3D point $\mathbf{P}$ corresponds to a planar part of the 3D scene. The equation of the associated plane in the point–normal form can be written as $\mathbf{n} \cdot \mathbf{P} + d = 0$, where $\mathbf{n} = (a, b, c)^T$ is the normal vector to
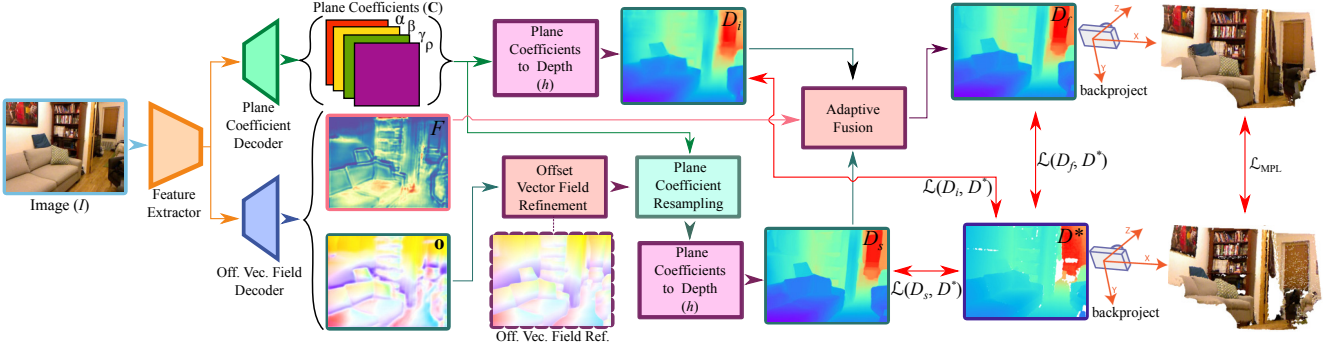
Figure 2. **Overview of our end-to-end P3Depth method.** P3Depth includes two output heads. The first head outputs pixel-level plane coefficients (**C**), while the second head outputs a dense offset vector field (**o**) identifying positions of seed pixels along with a confidence map (F). Then, the plane coefficients of seed pixels are used to predict depth at each position. The resulting prediction (D_s) is adaptively fused with the initial prediction (D_i) using the confidence map to account for potential deviations from precise local planarity.

the plane and $-d$ is the distance of the plane from the origin. Substituting **P** from (2) into the point-normal equation yields

$$\frac{1}{Z} = \underbrace{\frac{-a}{f_x d}}_{\hat{\alpha}} u + \underbrace{\frac{-b}{f_y d}}_{\hat{\beta}} v + \underbrace{\frac{1}{d}\left(\frac{a}{f_x}u_0 + \frac{b}{f_y}v_0 - c\right)}_{\hat{\gamma}}. \quad (3)$$

Thus, for image regions that depict planar 3D surfaces, the inverse depth is an affine function of pixel position, where the coefficients encode both the camera intrinsics and the 3D plane. We reformulate (3) by introducing $\rho = \sqrt{\hat{\alpha}^2 + \hat{\beta}^2 + \hat{\gamma}^2}$ and normalizing $\alpha = \frac{\hat{\alpha}}{\rho}$, $\beta = \frac{\hat{\beta}}{\rho}$ and $\gamma = \frac{\hat{\gamma}}{\rho}$ into

$$Z = [(\alpha u + \beta v + \gamma)\rho]^{-1}. \quad (4)$$

We term $\mathbf{C} = (\alpha, \beta, \gamma, \rho)^T$ as the *plane coefficients*. Using this notation, (4) can be written as $Z = h(\mathbf{C}, u, v)$. Instead of directly predicting depth, we design our model to have a plane coefficient head, which first predicts a dense plane coefficient representation $\mathbf{C}(u, v)$ and then applies (4) to compute an initial depth prediction which we denote with D_i. More formally, the mapping f_θ from Sec. 3.1 is now a composition $f_\theta = h \circ (\mathbf{g}_\theta, \mathbf{p})$, where $\mathbf{g}_\theta : I(u, v) \rightarrow \mathbf{C}(u, v)$ maps the input image to the plane coefficient representation and $h : (\mathbf{C}(u, v), u, v) \rightarrow D_i(u, v)$ applies (4) at each pixel. Supervision is applied to the output depth D_i via (1).

Predicting the plane coefficients as an intermediate output does not give a immediate advantage compared to directly predicting the depth. However, two pixels that depict the same 3D plane have the same parameters $\mathbf{C}$, but generally a different depth. This fact is the core to the next part of the network, which allows to predict depth by selectively bootstrapping the plane coefficients from a seed pixel.

3.3. Learning to Identify Seed Pixels

Let us assume we have one pixel **p** which belongs to a planar surface in 3D. By definition, every other pixel on this planar surface has the same **C** values. Thus, in an ideal

world the network only has to predict **C** at one of these pixels, **q**, to get all their depth values correct. This pixel can be interpreted as the seed pixel that describes the plane. However, defining this seed pixel and the region in which the depth should be bootstrapped from it is non-trivial. Thus, in this work we let the network discover this seed pixel and the respective region.

Formally, let us start by defining our piecewise planarity prior which is a relaxed version of the previous idea.

Definition 1. *(Piecewise planarity prior) For every pixel* **p** *with an associated 3D plane, there exists a seed pixel* **q** *in the neighborhood of* **p** *which is also associated with the same plane as* **p**.

Note that in general, there may exist multiple seed pixels or no seed pixel for **p**.

Given that the prior holds, the task of depth prediction for **p** can also be solved by identifying **q**, i.e., by predicting the offset $\mathbf{o}(\mathbf{p}) = \mathbf{q} - \mathbf{p}$. Thus, we design our model so that it features a second, offset head and let this offset head predict a dense offset vector field $\mathbf{o}(u, v)$. The two heads of the network share a common encoder and have separate decoders, as shown in Fig. 2. We use the predicted offset vector field to resample the plane coefficients via

$$\mathbf{C}_s(\mathbf{p}) = \mathbf{C}(\mathbf{p} + \mathbf{o}(\mathbf{p})), \quad (5)$$

using bilinear interpolation to handle fractional offsets. The resampled plane coefficients are then used to compute a second depth prediction

$$D_s(u, v) = h(\mathbf{C}_s(u, v), u, v), \quad (6)$$

based on the seed locations. This allows the network to bootstrap the depth from the seed pixel.

However, the prior is not always valid, so the initial depth prediction D_i may actually be preferable compared to the seed-based prediction D_s. To account for such cases, the second head additionally predicts a confidence
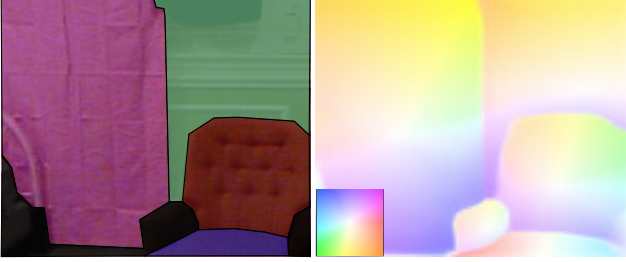
Figure 3. **Ground truth planes vs. predicted offset vector field.** The predicted offset vector at a pixel tends to point towards a seed pixel with which it shares the same plane coefficients. The left image shows the overlayed labels of the segmented planes on an example from NYU Depth-v2 and the right image shows the respective predicted offset vector field. The bottom left legend in the right image shows the color coding for the vector field.

map $F(u, v) \in [0, 1]$, which indicates the confidence of the model in using the predicted seed pixels for estimating depth via D_s. The confidence map is leveraged to compute the final depth prediction by adaptively fusing D_i and D_s:

$$D_f(u, v) = F(u, v)D_s(u, v) + (1 - F(u, v))D_i(u, v). \quad (7)$$

We apply supervision on each of D_f, D_i and D_s in our model, by optimizing the following loss:

$$\mathcal{L}_{\text{depth}} = \mathcal{L}(D_f, D^*) + \lambda\mathcal{L}(D_s, D^*) + \mu\mathcal{L}(D_i, D^*), \quad (8)$$

with λ and μ being hyperparameters. In this way, we encourage (i) the plane coefficient head to output a representation that is accurate across all pixels even when they have a high confidence value and (ii) the offset head to learn high confidence values for pixels for which the planarity prior holds and low confidence values for the converse.

However, there is a caveat in this formulation. In particular, the model is not supervised directly on the offsets. In fact, it could simply predict zero offsets everywhere and still produce valid predictions D_s and D_f, which would be identical to D_i. This unwanted behavior is avoided in practice thanks to the fact that the initial predictions D_i are erroneously smoothed near depth boundaries, due to the regularity of the mapping f_θ for the case of neural networks. As a result, for pixels on either side of a boundary, predicting a non-zero offset that points away from the boundary yields a lower value for $\mathcal{L}_{\text{depth}}$, because such an offset uses a seed pixel for D_s which is further from the boundary and suffers from smaller error owing to smoothing. Also due to regularity of the mapping that generates the offset vector field, these non-zero offsets are propagated from the boundaries to the inner parts of regions with smooth depth, helping the network to predict non-trivial offsets.

In the fully-fledged version of our method, we cascade the offset vectors multiple times before resampling the plane coefficient maps. For example, a single cascading step samples the position $\mathbf{p} + \mathbf{o}(\mathbf{p}) + \mathbf{o}(\mathbf{p} + \mathbf{o}(\mathbf{p}))$. Our

motivation for this cascaded refinement is that seed pixels within the same planar region should converge to the center of the region, which helps accumulate information from more pixels in predicting the plane coefficients of the region. At the same time, pixels without a reliable seed pixel are anyway assigned a low confidence value, so cascading the offsets does not hurt the respective depth prediction.

3.4. Mean Plane Loss

The assumption we use for formulating our mean plane loss is that given a pixel coordinate, its neighboring pixels should lie on the same plane in the 3D space. The normal $\mathbf{n}$ of this plane should satisfy an overdetermined system of linear equations. However, ground-truth depth maps are usually captured by consumer-level sensors with noisy measurements and limited precision, which renders the above regime for local fitting of normals inapplicable, as finding the true optimal solution is not guaranteed.

Even though this is a valid observation, depth measurements still contain comprehensive details about the scene structure. This information can be aggregated locally to enforce first-order consistency between the predicted and the ground-truth 3D surface. Normals are one way how this aggregation across patches can be performed. For an input patch, the corresponding normal $\mathbf{n}$ needs to satisfy $\mathbf{A}\mathbf{n} = \mathbf{b}$, s.t. $\|\mathbf{n}\|_2 = 1$, where $\mathbf{A}$ is a data matrix build by stacking the 3D points in the patch and $\mathbf{b}$ is a vector of ones. Following [11, 53], the closed-form solution of this least-squares problem is:

$$\mathbf{n} = \frac{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}}{\left\|(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}\right\|_2}. \quad (9)$$

To compute the mean plane loss, we first estimate surface normals for all K non-overlapping patches in D and D^* and then penalize their difference via

$$\mathcal{L}_{\text{MPL}} = \sum_{k=1}^{K} \|\mathbf{n}_k - \mathbf{n}_k^*\|_1. \quad (10)$$

For patches with depth discontinuities, even when the $\mathbf{n}_k^*$ for patch k does not correspond to a ground-truth 3D plane, the mean plane loss still provides a useful supervision signal, as it penalizes local depth profiles that are inconsistent with $\mathbf{n}_k^*$. Also, we do not require ground-truth normals, as opposed to [53]. Given (9), we can see that the loss (10) directly affects the depth of all points inside the patch via $\mathbf{A}$. Finally, the complete loss is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{MPL}}$.

4. Experiments

We structure this section as follows. We first discuss our experimental setup, i.e., datasets, evaluation metrics and implementation details for our method. We then compare our method to state-of-the-art approaches, followed by a thorough ablation study of our method.
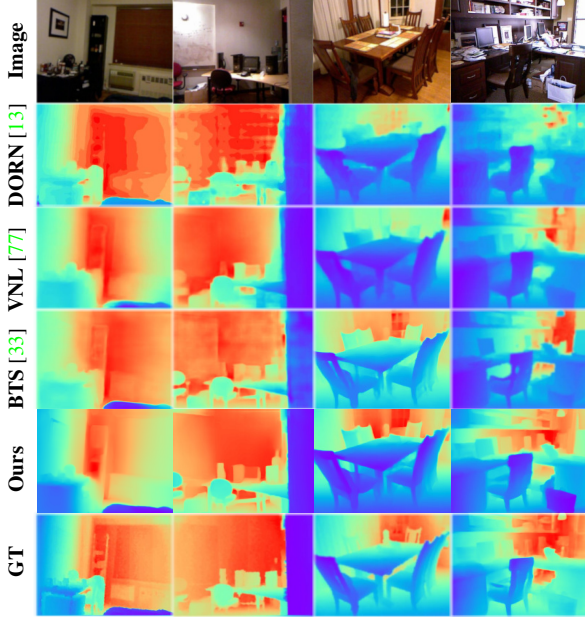
Figure 4. **Qualitative results on NYU Depth-v2.** We compare our method against SOTA methods using the same examples as [77].

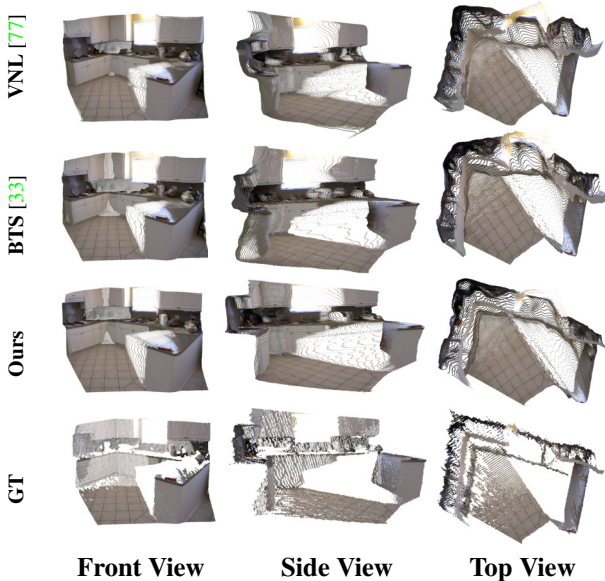**Front View**   **Side View**   **Top View**

Figure 5. **Reconstruction example from NYU Depth-v2.** We compare the 3D reconstruction induced by our predicted depth to results using two SOTA depth estimation methods [33, 77].

4.1. Experimental Setup

In this section, we present the datasets used to evaluate our approach. The NYU Depth-v2 and KITTI datasets are used as the main sets for training and testing. Additionally, we use four more datasets for testing our method in a zero-shot transfer setup in order to evaluate its generalization potential. Evaluation on all six datasets is performed using standard depth evaluation metrics explained below.

NYU Depth-v2 [62]. The NYU Depth-v2 dataset consists of 464 indoor scenes of size 640×480. These scenes are

Table 1. **Datasets used in our experiments.** (*) uses mix of RGBD sensors. Sup.: supervision.

Dataset	# Training	# Testing	Sup. Type	Scene Type
NYU Depth-v2 [62]	24,231	654	Kinect	Indoor
KITTI [17]	23,488	697	LiDAR	Outdoor
ScanNet [7]	-	2167	Kinect	Indoor
SUN-RGBD [63]	-	5050	Mixed*	Indoor
DIODE Indoor [64]	-	325	LiDAR	Indoor
ETH-3D [60]	-	454	LiDAR	Mixed

split into 249 scenes for training and 215 for testing. We use the official split provided by previous methods [33] for training and the test set is based on [9].

KITTI [17]. KITTI is an autonomous driving dataset consisting of 61 outdoor scenes of different types. We employ the standard depth estimation split proposed by Eigen *et al*. [9] and Garg *et al*. [16], for training and testing. We use 32 scenes for training and 29 scenes for testing.

Datasets used for zero-shot testing. To test the generalization of our P3Depth, we evaluate it on four datasets which are not seen during training: ScanNet, SUN-RGBD, DIODE Indoor and ETH-3D. The resolution of all images is reduced to 640×480. Details are provided in Table 1.

Evaluation metrics. We use the standard depth estimation metrics for evaluation. In particular, we use root mean square error (RMSE) and its log variant (RMSE*log*), Log10 error, absolute (A.Rel) and squared (S.rel) mean relative error and the percentage of inlier pixels with δ. The maximum depth for KITTI is set to 50m and 80m for the Garg and Eigen splits respectively. For NYU Depth-v2, the maximum depth is set to 10m as per the Eigen split. Zero-shot transfer is performed by using a model trained on NYU Depth-v2 without additional fine-tuning. The maximum depth for this task is set to 10m across all four test datasets. In all evaluations, the predicted depth is rescaled so that its median matches that of the ground truth, as per standard practice.

Implementation details. Our network includes two heads. The first head outputs four channels, one for each plane coefficient. The second head outputs three channels: one for each coordinate of the offsets and one for confidence. These heads are fed by a ResNet101 encoder [21] initialized with pre-trained ImageNet [28] weights. This initialization is important to achieve competitive results as in [13, 23, 33, 77]. The decoders, inspired from [70], are initialized with weights drawn from a normal distribution with σ= 0.01. The plane coefficient decoder is additionally equipped with a guidance module. See supplemental for details. The offset vectors are restricted via a tanh layer to have a maximum length of τ in normalized image coordinates. We set τ to 0.1 by default and apply two steps of cascaded refinement to the offsets. The confidence map is predicted through a sigmoid layer. For all experiments, we use a batch size of 8 and employ the Adam optimizer [25] with a learning rate of 10^{-4} and a weight decay of 10^{-4}. We

Table 2. **Comparison of depth estimation methods on NYU Depth-v2 [62] test set.** Comparison is performed on the Eigen split [9]. (nF) is number of frames, (*) indicates self-supervised methods and (†) denotes retrained results with train set from [33].

Method	A.Rel	Log10	RMSE	δ_1	δ_2	δ_3
	Lower is better			Higher is better		
Plane detection based methods						
PlaneNet [42]	0.142	0.060	0.514	0.812	0.957	0.989
PlaneRCNN [41]	0.124	0.077	0.644	–	–	–
Yu *et al.* [81]	0.134	0.057	0.503	0.827	0.963	0.990
P^2 Net (5F)* [80]	0.147	0.062	0.553	0.801	0.951	0.987
StruMonoNet [75]	0.107	0.046	0.392	0.887	0.980	0.995
Other monocular depth estimation methods						
Saxena *et al.* [59]	0.349	-	1.214	0.447	0.745	0.897
Karsch *et al.* [24]	0.349	0.131	1.21	-	-	-
Liu *et al.* [45]	0.335	0.127	1.06	-	-	-
Ladicky *et al.* [31]	-	-	-	0.542	0.829	0.941
Li *et al.* [37]	0.232	0.094	0.821	0.621	0.886	0.968
Wang *et al.* [68]	0.220	0.094	0.745	0.605	0.890	0.970
Liu *et al.* [44]	0.213	0.087	0.759	0.650	0.906	0.974
Roy *et al.* [57]	0.187	0.078	0.744	-	-	-
AdaBins† [1]	0.178	0.078	0.595	0.698	0.937	0.988
Eigen *et al.* [9]	0.158	-	0.641	0.769	0.950	0.988
Chakrabarti [4]	0.149	-	0.620	0.806	0.958	0.987
Li *et al.* [38]	0.143	0.063	0.635	0.788	0.958	0.991
Laina *et al.* [32]	0.127	0.055	0.573	0.811	0.953	0.988
Fu *et al.* [13]	0.115	0.051	0.509	0.828	0.965	0.992
Yin *et al.* [77]	0.108	0.048	0.416	0.875	0.976	0.994
Huynh *et al.* [23]	0.108	-	0.412	0.882	0.980	<u>0.996</u>
Lee *et al.* [33]	0.110	0.047	0.392	0.885	0.978	0.994
Long *et al.* [46]	**0.101**	<u>0.044</u>	0.377	0.890	<u>0.982</u>	**0.996**
Ranftl *et al.* [55]	0.110	0.045	<u>0.357</u>	**0.904**	**0.988**	**0.998**
Ours	<u>0.104</u>	**0.043**	**0.356**	<u>0.898</u>	0.981	<u>0.996</u>

Table 3. **Comparison of depth estimation methods on KITTI [17].** Comparison is performed on the Eigen test split.

Method	A.Rel	S.Rel	RMSE	RMSE*log*	δ_1	δ_2	δ_3
	Lower is better				Higher is better		
Garg split [16] cap: 50m							
Garg *et al.* [16]	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard *et al.* [18]	0.108	0.657	3.729	0.194	0.873	0.954	0.979
Kuznietsov [29]	0.108	0.595	3.518	0.179	0.875	0.964	0.988
Gan *et al.* [15]	0.094	0.552	3.133	0.165	0.898	0.967	0.986
Fu *et al.* [13]	0.071	0.268	2.271	0.116	0.936	0.985	0.995
AdaBins [1]	0.058	0.19	2.36	0.088	<u>0.964</u>	<u>0.995</u>	**0.999**
Lee *et al.* [33]	<u>0.056</u>	<u>0.169</u>	<u>1.925</u>	<u>0.087</u>	<u>0.964</u>	0.994	**0.999**
Ours	**0.055**	**0.130**	**1.651**	**0.081**	**0.974**	**0.997**	**0.999**
Eigen split [9] cap: 80m							
Saxena *et al.* [59]	0.280	3.012	8.734	0.361	0.601	0.820	0.926
Eigen *et al.* [9]	0.203	1.548	6.307	0.282	0.702	0.898	0.967
Liu *et al.* [43]	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Godard *et al.* [18]	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Kuznietsov [29]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Gan *et al.* [15]	0.098	0.666	3.933	0.173	0.890	0.964	0.985
Fu *et al.* [13]	0.072	0.307	<u>2.727</u>	0.120	0.932	0.984	0.994
Yin *et al.* [77]	0.072	-	3.258	0.117	0.938	0.990	<u>0.998</u>
Lee *et al.* [33]	**0.059**	**0.245**	2.756	<u>0.096</u>	<u>0.956</u>	0.993	<u>0.998</u>
AdaBins [1]	0.067	0.278	2.96	0.103	0.949	0.992	<u>0.998</u>
Ranftl *et al.* [55]	<u>0.062</u>	-	**2.573**	**0.092**	**0.959**	**0.995**	**0.999**
Ours	0.071	<u>0.270</u>	2.842	0.103	0.953	<u>0.993</u>	<u>0.998</u>

train our network for 25 epochs, although the model starts producing decent predictions from epoch 5. The learning rate is reduced every 5 epochs by a factor of 10 using a step scheduler. The training images are resized similarly to [33]. For all direct depth losses, we use the loss formulation from [9]. The loss weights λ and μ are set to 0.5. In addition, the mean plane loss is applied using the final depth prediction D_f. The offset head performs better with dense supervision. Hence, D_s is supervised using completed D^*. To complete D^*, the depth inpainting method from [62] is used. The inpainted ground truth is also used for computing the mean plane loss to provide stability to the SVD algorithm for least squares. We set the patch size to 32 and K in (10) is set indirectly by the patch size and the image size. The data augmentation techniques from [33] are used.

4.2. Comparison with State of the Art

NYU Depth-v2: The results on NYU Depth-v2, which is the major indoor depth benchmark, are presented in Table 2. We set the new state of the art on NYU Depth-v2, outperforming prior state-of-the-art (SOTA) methods across all six standard metrics. We achieve a superior relative performance gain of 9.18% in RMSE and 3.7% in A.Rel, while also improving δ_1 by 1.1%. This improvement in performance indicates that without using ground-truth planes as supervision, P3Depth learns an implicit representation of

the planes which can benefit the overall depth estimation capability of the network.

Qualitative results on NYU Depth-v2 support the above findings. In Fig. 4, we show the high-quality predictions generated by our method in comparison with SOTA methods using the same examples as in [77]. It can be clearly observed that the surfaces which fit our piece-wise planar assumption, such as table, cupboard, and even smaller objects, e.g. computer screens, photo frames etc. have consistent predictions with sharp details in comparison with the SOTA methods. Overall, our method generates superior visual results. In some cases, especially w.r.t. the metric scale of the scene, results from [33] are comparable to ours. Our method excels especially on man-made regular structures of the indoor scenes. What is more, the predicted depth maps produce 3D reconstructions which are consistent with ground-truth point clouds and preserve the structure of the scene better than competing methods, as shown in Fig. 5.

KITTI: The results on KITTI in Table 3 suggest that our method is fully applicable to outdoor datasets. In particular, we surpass prior state of the art on the Garg split (with maximum range of 50m) in all metrics by a significant margin. More specifically, we improve RMSE by 14.2% and δ_1 by 1.0%. This proves that our method takes advantage of regular structures in outdoor scenes to improve depth predictions. Moreover, our method is comparable to state of the art on the Eigen split [9], where the maximum range is 80m. The reason why our ranking is slightly lower on the Eigen split is that distant parts of the scene get projected to smaller regions and thus the extent of the respective smooth pieces of the depth map is also smaller, making it more difficult to predict a correct offset. Additionally, for the results on the KITTI benchmark suite, please refer to the supplement. Overall, the method is able to handle planar objects quite
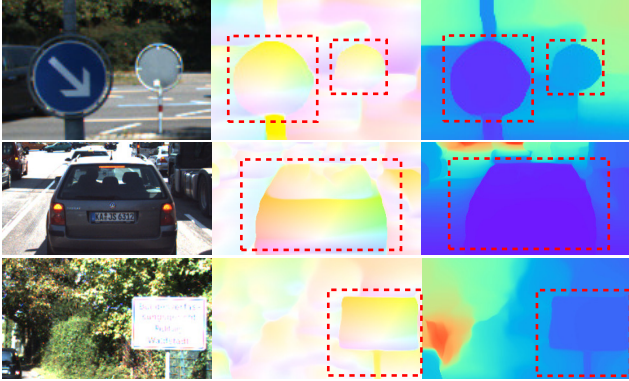
Figure 6. Qualitative results on KITTI. We present the predicted offset vector fields (middle) and the depth estimates (right).

well even in variable lighting conditions as shown in Fig. 6. Although the sign board in the bottom image is brightly lit and the car in the middle image is badly lit, the method is able to detect the regularity of the object surface and predict consistent depth. The top image of Fig. 6 demonstrates a limitation of our method. In particular, the road segments on either side of the pole of the traffic sign on the right get mapped to significantly different depth values because they are disconnected and thus do not interact in terms of plane coefficients, even though they belong to the same 3D plane. Additionally, the planar specular glass surface of the car in middle image is predicted incorrectly. This is due to the shortcomings of the sensor used to measure depth. The erroneous ground truth does not allow the network to learn the depth or the offset vector field in these regions.

Zero-shot experiments: In Table 4, we prove the generalization ability of our method in a zero-shot setting where the test domains have not been seen during training. We achieve the best performance on the ScanNet [7] and SUN-RGBD [63] indoor datasets in all metrics. On DIODE Indoor and ETH-3D, [77] performs best in terms of A.Rel, but we are by far the best in terms of both RMSE and δ_1. This comparison shows that even when our method is trained only on an indoor dataset as NYU Depth-v2, it works well on a variety of datasets with different types of scenes.

4.3. Ablation Studies

We study the importance of the components of our method by ablating them in Table 5. We observe that in a standalone setting, directly predicting depth is better than predicting planar coefficients. However, once we insert the second head which predicts offset vectors, a substantial benefit is obtained by using the plane coefficient representation compared to directly predicting depth. This demonstrates that the network learns to make effective use of local planar information at seed pixels to improve depth, thanks to the plane coefficient representation. Moreover, adding our guidance module provides a slight improvement. The ablation also verifies the utility of cascaded refinement of off-

Table 4. Comparison of SOTA methods on generalized learning of metric depth. All methods are trained on NYU Depth-v2 and tested on four other datasets without fine-tuning.

Dataset	Metric	VNL [77]	BTS [33]	Ours
ScanNet [7]	A.Rel ↓	0.227	0.255	**0.223**
	RMSE ↓	0.563	0.615	**0.538**
	δ_1 ↑	0.544	0.472	**0.551**
SUN-RGBD [63]	A.Rel ↓	0.317	0.317	**0.307**
	RMSE ↓	0.449	0.461	**0.431**
	δ_1 ↑	0.793	0.794	**0.797**
Diode Indoor [64]	A.Rel ↓	**0.291**	0.310	0.373
	RMSE ↓	0.890	0.981	**0.784**
	δ_1 ↑	0.635	0.559	**0.639**
ETH-3D [60]	A.Rel ↓	**0.331**	0.366	0.343
	RMSE ↓	1.649	1.840	**1.637**
	δ_1 ↑	0.462	0.398	**0.468**

Table 5. Ablation study of components of our method. D: directly predicting depth, C: predicting plane coefficients, "Guid.": guidance module for plane coefficient decoder, "OV": offset vectors, "Ref.": cascaded refinement of offsets, "MPL": mean plane loss, "+": offset length is restricted to $\tau=0.3$ instead of $\tau=0.1$.

Pred.	Guid.	OV	Ref.	MPL	A.Rel ↓	RMSE ↓	δ_1 ↑
D					0.142	0.458	0.821
C					0.144	0.487	0.811
C	✓				0.142	0.458	0.824
D		✓			0.140	0.453	0.824
C		✓			0.116	0.390	0.877
C				✓	0.118	0.395	0.872
C	✓	✓			0.115	0.384	0.879
C	✓	✓+			0.116	0.390	0.879
D		✓	✓		0.134	0.440	0.839
C		✓	✓		0.113	0.378	0.884
C		✓	✓	✓	0.109	0.370	0.890
C	✓	✓	✓		0.109	0.373	0.889
C	✓	✓	✓	✓	**0.104**	**0.356**	**0.898**

sets, which yields better results than simply using a higher maximum offset length. Finally, adding our mean plane loss on top of plane coefficients, offset vectors and cascaded offset refinement yields SOTA results on NYU Depth-v2.

5. Conclusion

We have presented a supervised method for monocular depth estimation which leverages local planar information in the 3D scene in order to predict consistent depth values across smooth parts of the scene. The method uses a plane coefficient representation for depth, which enables to share information from seed locations and improve the predicted depth. We implicitly learn to predict the offsets to these seed locations and to weigh the depth obtained from them adaptively according to accuracy. We empirically validated our method on the major indoor and outdoor benchmarks for monocular depth estimation and set the new state of the art among supervised approaches, which shows the potential of well-selected geometric priors for depth estimation.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 7

[2] Andras Bodis-Szomoru, Hayko Riemenschneider, and Luc Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 3

[3] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2018. 2

[4] Ayan Chakrabarti, Jingyu Shao, and Gregory Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, 2016. 7

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[6] Anne-Laure Chauve, Patrick Labatut, and Jean-Philippe Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 3

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6, 8

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014. 1, 2, 6, 7

[10] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[11] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013. 5

[12] 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 (CVPR)*, 2018. 1, 2

[13] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6, 7

[14] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 3

[15] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 7

[16] Ravi Garg, Vijay Kumar B.G., Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *The European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 6, 7

[17] 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. 6, 7

[18] Clement 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 (CVPR)*, 2017. 1, 2, 7

[19] Clement 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 (ICCV)*, 2019. 1, 2

[20] Juan Luis Gonzalez and Munchurl Kim. PLADE-Net: Towards pixel-level accuracy for self-supervised single-view depth estimation with neural positional encoding and distilled matting loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[21] 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*, 2016. 6

[22] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 3

[23] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *The European Conference on Computer Vision (ECCV)*, 2020. 3, 6, 7

[24] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014. 7

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations, ICLR*, 2015. 6

[26] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *The European Conference on Computer Vision (ECCV)*, 2020. 2

[27] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012. 6

[29] 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 (CVPR)*, 2017. 2, 7

[30] Lubor Ladický, Christian Häne, and Marc Pollefeys. Learning the matching function. *arXiv e-prints*, abs/1502.00652, February 2015. 3

[31] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 7

[32] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, 2016. 1, 2, 7

[33] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv e-prints*, abs/1907.10326, July 2019. 3, 6, 7, 8

[34] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. Single-image depth estimation based on Fourier domain analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[35] Jae-Han Lee and Chang-Su Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *The European Conference on Computer Vision (ECCV)*, 2020. 2

[36] Boying Li, Yuan Huang, Zeyu Liu, Danping Zou, and Wenxian Yu. StructDepth: Leveraging the structural regularities for self-supervised indoor depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[37] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[38] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 7

[39] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[40] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G. Narasimhan, and Jan Kautz. Neural RGB(r)D sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[41] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 7

[42] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. PlaneNet: Piece-wise planar reconstruction from a single RGB image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3, 7

[43] 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, 2016. 2, 7

[44] 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, 2016. 7

[45] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 7

[46] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Wei Li, Christian Theobalt, Ruigang Yang, and Wenping Wang. Adaptive surface normal constraint for depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3, 7

[47] V Madhu Babu, Kaushik Das, Anima Majumdar, and Swagat Kumar. UnDEMoN: Unsupervised deep network for depth and ego-motion estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018. 2

[48] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[49] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[50] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[51] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[52] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation

from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020. 2

[53] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. GeoNet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5

[54] Michael Ramamonjisoa, Yuming Du, and Vincent Lepetit. Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[55] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 7

[56] Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[57] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[58] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 3

[59] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 2, 7

[60] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 8

[61] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *The European Conference on Computer Vision (ECCV)*, 2020. 2

[62] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, 2012. 6, 7

[63] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6, 8

[64] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *arXiv e-prints*, abs/1908.00463, 2019. 6, 8

[65] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287—-2318, January 2016. 3

[66] Liang Wang, Hailin Jin, Ruigang Yang, and Minglun Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 3

[67] Lijun Wang, Yifan Wang, Linzhao Wang, Yunlong Zhan, Ying Wang, and Huchuan Lu. Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[68] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[69] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[70] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6

[71] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[72] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[73] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[74] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[75] Zhenpei Yang, Li Erran Li, and Qixing Huang. Strumononet: Structure-aware monocular 3d prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[76] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. LEGO: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[77] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth pre-

diction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 6, 7, 8

[78] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3D scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[79] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[80] Zehao Yu, Lei Jin, and Shenghua Gao. P2Net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *The European Conference on Computer Vision (ECCV)*, 2020. 3, 7

[81] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 7

[82] 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 (CVPR)*, 2018. 2

[83] Weidong Zhang, Wei Zhang, and Yinda Zhang. GeoLayout: Geometry driven room layout estimation based on depth maps of planes. In *The European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[84] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4(30), 2019. 1

[85] 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 (CVPR)*, 2017. 1, 2

[86] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2