This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

StructDepth: Leveraging the structural regularities for self-supervised indoor depth estimation

Boying Li*, Yuan Huang*, Zeyu Liu, Danping Zou†, and Wenxian Yu
Shanghai Key Laboratory of Navigation and Location-Based Services
Shanghai Key Laboratory of Intelligent Sensing and Recognition
Shanghai Jiao Tong University

Abstract

Self-supervised monocular depth estimation has achieved impressive performance on outdoor datasets. Its performance however degrades notably in indoor environments because of the lack of textures. Without rich textures, the photometric consistency is too weak to train a good depth network. Inspired by the early works on indoor modeling, we leverage the structural regularities exhibited in indoor scenes, to train a better depth network. Specifically, we adopt two extra supervisory signals for self-supervised training: 1) the Manhattan normal constraint and 2) the co-planar constraint. The Manhattan normal constraint enforces the major surfaces (the floor, ceiling, and walls) to be aligned with dominant directions. The co-planar constraint states that the 3D points be well fitted by a plane if they are located within the same planar region. To generate the supervisory signals, we adopt two components to classify the major surface normal into dominant directions and detect the planar regions on the fly during training. As the predicted depth becomes more accurate after more training epochs, the supervisory signals also improve and in turn feedback to obtain a better depth model. Through extensive experiments on indoor benchmark datasets, the results show that our network outperforms the state-of-the-art methods. The source code is available at `https://github.com/SJTU-ViSYS/StructDepth`.

1. Introduction

Inferring the dense 3D map from a single image is a challenging problem without satisfactory solutions until the booming of deep neural networks. With the deep convolutional neural networks (CNNs), we can predict the accurate depth from a single image, via training the network with a lot of ground-truth depth labels. The recent self-supervised learning paradigm does not require the ground-truth depth, while still obtaining high-quality results on benchmark datasets, using the photometric consistency as the major supervisory signal. Nevertheless, when existing self-supervised methods are trained on indoor images, the quality of depth estimation degrades notably[51][3]. The main reason is the lack of textures in indoor images. Unlike outdoor scenes, the indoor scenes are full of texture-less regions, such as white walls, ceilings, and floors. Without rich textures, the photometric loss becomes too weak to train a good depth model. Seeking stronger or extra supervisory signals is therefore necessary for training a better depth network.

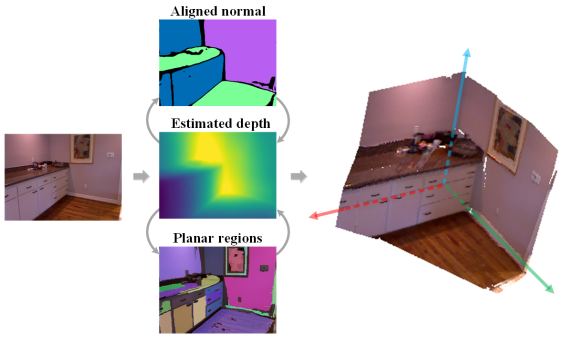

Figure 1. Our self-supervised monocular depth learning leverages the structural regularities of indoor environments for training. The aligned normal (with Manhattan directions) and the planar regions provide extra losses in training and lead to better 3D structures at inference.

There have been a few attempts. An optical-flow field propagated from the sparse SURF[1] flow by a self-supervised network, is used to guide training on texture-less regions [51]. Another attempt [48] is to use an image patch instead of individual pixels to compute the photometric loss and apply extra constraints to the depth within the planar regions extracted from image segmentation. Though those attempts improve the results, they did not fully ex-

*Both are the first authors with equal contributions. †Corresponding author: Danping Zou (`dpzou@sjtu.edu.cn`). This work was supported by NSFC (62073214).

ploit the structural regularities presented in indoor environments, a valuable source of information for 3D learning. The structural regularities, known as the Manhattan-world model[6], describe that the scene consists of major planes aligned with dominant directions. This simple yet effective high-level prior leads to a much better performance in many vision tasks, such as indoor modeling[16][17][5], visual SLAM[50][12][43], and visual-inertial odometry[54], but has not been applied to monocular depth learning.

In this work, we propose to apply the high-level prior of indoor structural regularities to self-supervised depth estimation as shown in Fig. 1. Specifically, we adopt two extra supervisory signals for training: 1) the Manhattan normal constraint and 2) the co-planar constraint. The Manhattan normal constraint enforces the major surfaces (the floor, ceiling, and walls) to be aligned with dominant directions. The co-planar constraint states that the 3D points be well fitted by a plane if they are located within the same planar region. We add two extra components into the training process. The first one is Manhattan normal detection. It classifies the major surface normal, computed from the depth predicted by the network, into the directions associated with the vanishing points by an adaptive thresholding scheme. The second one is planar region detection. We fuse the color and the geometric information derived from the depth and apply a classic segmentation algorithm to extract planar regions. During training, the two components incorporate the estimated depth to produce supervisory signals on the fly. Though those signals may be noisy in early epochs because of inaccurate depth, they will gradually improve as the depth quality improves, and in turn benefit the depth estimation.

We conduct experiments on the indoor benchmark datasets: NYU-v2 [39], ScanNet[7], and InteriorNet[28]. The results show that our method outperforms the existing state-of-the-art methods. Our main contributions are as follows:

1) A novel learning pipeline for self-supervised depth estimation leveraging structural regularities of indoor environments. To our best knowledge, this has not been presented in previous work.

2) Two novel components providing extra supervisory signals on the fly during the training process. Our components can be used to train a multi-task network including depth estimation, normal estimation, and planar region detection in a self-supervised manner, although the latter two tasks serve to train a better depth model in our current implementation.

3) We set a new state-of-the-art in self-supervised indoor depth estimation.

2. Related Work

Monocular depth estimation. Depth estimation from a single image is an ill-posed problem that is known as extremely hard to be solved. Since the pioneer works[10, 9] employed the convolution neural networks (CNNs) to regress the depth directly, a lot of CNN-based monocular depth estimation methods have been proposed [31, 25, 24, 42, 15], producing impressively accurate results in benchmark datasets. Most of them are supervised methods that require the ground-truth depth data for training.

Self-supervised depth learning without the ground-truth depth has emerged as a promising alternative as acquiring the ground-truth depth at a large scale is challenging. The image appearance was firstly introduced in [19] to replace the ground-truth depth as the supervisory signal to train a depth network. One image in a stereo pair was warped to the other view by the predicted depth. The difference between the synthesized image and the real image, or the photometric error, is then used for supervision. The idea was further extended to monocular settings [52][19]. By the careful design of network architectures[20], loss functions [38], and online refinement [4], self-supervised approaches obtain impressive results on benchmark datasets.

Despite achieving impressive performance on outdoor datasets, such as KITTI[18] and Make3D[36], existing self-supervised methods perform poorly in indoor datasets. The reason is that the indoor scenes are full of texture-less regions, such as white walls and ceilings, making the photometric loss become too weak to supervise the depth learning. Zhou et al.[51] adopted an optical-flow-based training paradigm supervised by the flow field from an optical flow network, initialized from sparse SURF [1] correspondences. The recent work [48] employed the more discriminative patches instead of individual pixels to compute the photometric loss, and also applied the piece-wise planar prior to depth learning by assuming that the homogeneous-color regions are planar regions. Though their approaches improve the performance. They did not fully exploit the structural prior of the environments. In addition, the planar-region assumption in [48] does not hold for planes with the same color, e.g. mutually perpendicular white walls. It therefore leads to false planar regions deteriorating the depth model.

Planar region detection. Though powerful planar-region detectors [29][44][49] have been proposed recently and have shown high-quality results in complex indoor images. Those CNN-based detectors require a huge number of plane labels for training and are not suited for the self-supervised learning scheme. Though detecting planes in the image is challenging, if the depth is available, this task becomes much easier[35][23]. Here, we detect the planar regions using a classic graph-based segmentation approach [11] simi-

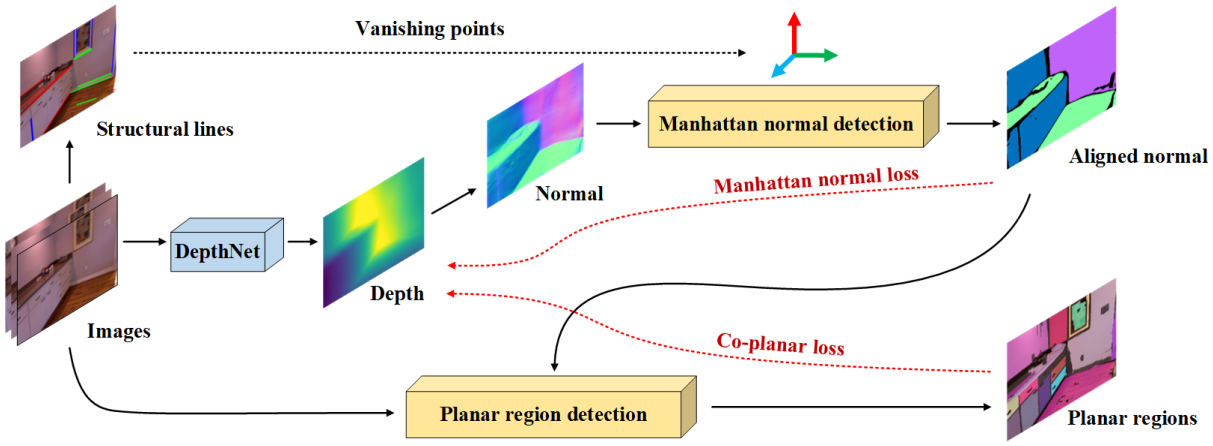

Figure 2. Our self-supervised monocular depth learning pipeline, which consists of three major components: a) **DepthNet**: The neural network to be trained to predict the depth from a single image. b) **Manhattan normal detection**: It classifies the surface normal estimated from depth prediction into dominant directions. c) **Planar region detection**: Both the color and geometric information are used to extract planar regions by a graph-based segmentation. The planar region detection is kept updated with the improved depth during training iterations. Two extra losses, *Manhattan normal loss* and *co-planar loss*, are used to train the network, as indicated by the red dot arrows.

lar to [48], while employing the additional geometric information extracted from the depth estimated on the fly when training. Though the depth may not be precise initially, it will gradually improve as the training progresses such that the segmentation will improve as well. With the additional geometric information, our approach avoids false planar regions that are indistinguishable by colors and produces less over-segmentation on texture-rich planar regions.

Structural regularities in indoor environments. Indoor scenes exhibit strong structural regularities, which can be described as the "Manhattan world". Namely, the scene can be decomposed into major planes, where their normal vectors are mutually orthogonal. These structural regularities are valuable priors that have been applied to a wide range of indoor 3D vision tasks, such as vSLAM[50][12][43], VIO[54], and mapping[16][17][5]. In fact, exploiting the structural prior of indoor scenes was probably the only geometric way to infer the 3D information from a single image in early days [8][26]. It is natural to think that structural regularities should also benefit the learning-based vision tasks in indoor environments.

Wang et al. [40] propose to use the vanishing points and lines to train a surface normal estimator which achieves the state-of-the-art performance. Our work adopts a similar spirit but differs from theirs in that our major task is depth estimation, where the surface normal is just an intermediate result that serves for better training. In addition, our depth network is trained in a fully self-supervised manner and does not require the line map as the extra input. To our best knowledge, our work is the first one incorporating the structural regularities of indoor environments into self-supervised monocular depth estimation.

3. Method

Our self-supervised depth learning pipeline is illustrated in Fig. 2. It consists of three major components. The first one is the depth network, which takes a single image as the input and predicts a depth map. We use the same architecture as in [48] for the depth network. Based on the predicted depth, the other two components, Manhattan normal detection and planar region detection, are used to produce the supervisory signals leveraging the structural prior of indoor environments. Manhattan normal detection aligns the normal computed from the depth map with the dominant orientations, estimated from the vanishing points in the image. Planar region detection applies a graph-based segmentation to detect the planar regions with the combination of color, normal, and plane-to-origin distance information. Both Manhattan normal detection and planar region detection may be inaccurate in the initial training epochs, but they will improve in later epochs as the depth prediction becomes better. The improved supervisory signals lead to a better depth prediction as well.

In the following sections, we'll describe how we apply the Manhattan normal constraint and the co-planar constraint in our training process.

3.1. Manhattan normal constraint

Dominant direction extraction. The structural regularities of indoor environments imply that most indoor scenes contain planar surfaces aligned with dominant directions. The dominant directions can be estimated from the structural lines in the image. The intersection of a set of parallel structural lines in the image is the vanishing point. Let $\boldsymbol{v}$ be a vanishing point extracted from the 2D image. One of

the dominant directions in the camera coordinate system is computed as

$$\boldsymbol{\eta} \propto \mathbf{K}^{-1}\boldsymbol{v}, \tag{1}$$

where $\boldsymbol{\eta} \in \mathbb{R}^3$ is a unit vector representing this dominant direction and $\mathbf{K}$ is the camera intrinsic matrix. Note that we need only two vanishing points to get all the dominant directions, since the third dominant direction can be obtained by the cross product. We apply the 2-Line searching method [32] to extract the dominant directions from the image. The dominant direction extraction is done only once before training.

Both the extracted directions and their reverse directions are considered to be the possible normal directions of the major planes in the scene, such as the ceiling, the floor, and the walls.

Surface normal estimation. To estimate the surface normal, we first get the 3D coordinates $\boldsymbol{X}_p \in \mathbb{R}^3$ of each pixel $\boldsymbol{p}$ from the predicted depth by

$$\boldsymbol{X}_p = D(\boldsymbol{p})\mathbf{K}^{-1}\boldsymbol{p}. \tag{2}$$

Here, $D(\boldsymbol{p})$ denotes the depth predicted by the depth network. Next, we adopt a differentiable point-to-normal layer[45, 46, 22] to estimate the surface normal from the 3D points. Specifically, the normal $\boldsymbol{n}_p$ of a given pixel $\boldsymbol{p}$ is calculated from a set of 3D points within a small neighborhood centering on point $\boldsymbol{X}_p$. The neighborhood is set as 7×7 in our implementation as the previous work[45].

Manhattan normal detection. Given the surface normal prediction $\boldsymbol{n}$, we propose the Manhattan normal detection to classify the surface normal that belongs to the dominant planes. Our strategy is to compare the difference between the estimated normal vector $\boldsymbol{n}_p$ and each dominant direction $\boldsymbol{\eta}^k$ by using a cosine similarity $s(\cdot,\cdot)$ and choose the one with the best similarity, namely

$$\boldsymbol{n}_p^{align} \leftarrow \arg\max_{\boldsymbol{\eta}^k} s(\boldsymbol{n}_p, \boldsymbol{\eta}^k) \tag{3}$$

where $\boldsymbol{n}_p^{align}$ is the aligned normal and the cosine similarity is defined as $s(\boldsymbol{n}_p, \boldsymbol{\eta}_k) = (\boldsymbol{n}_p \cdot \boldsymbol{\eta}_k)/(\|\boldsymbol{n}_p\| \cdot \|\boldsymbol{\eta}_k\|)$. Let the maximum similarity of each pixel be s_p^{max}. We define the Manhattan mask as:

$$\mathcal{M}_p^M = \begin{cases} 1 & s_p^{max} \geq \gamma \\ 0 & s_p^{max} < \gamma \end{cases} \tag{4}$$

where 1 and 0 represent Manhattan and non-Manhattan regions respectively

During the training, we use an adaptive thresholding scheme for detecting the Manhattan regions. We initially set a relatively small threshold to allow more pixels being classified into the Manhattan region because of inaccurate normal estimates, and gradually increase the threshold since the normal estimates become accurate in later epochs. In our implmentation, the threshold γ grows with the iteration number N^{train} linearly: $\gamma = \alpha \cdot N^{train} + \beta$, where α and β are set to $1.633e^{-3}$ and 0.9 respectively.

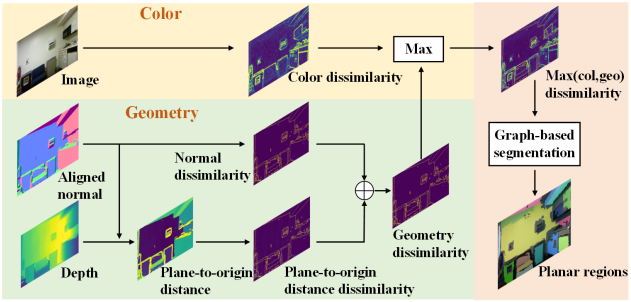

Figure 3. The pipeline of planar region detection. Both the color and geometric information are used to compute the dissimilarity for planar region segmentation. The color dissimilarity is calculated by comparing the RGB colors. The geometry dissimilarity is the sum of the normal and the plane-to-origin distance dissimilarities. Based on the proposed dissimilarity, a graph-based segmentation [11] is applied to extract the planar regions.

Manhattan normal loss. We apply the Manhattan normal constraint within the Manhattan region by using the aligned normal obtained in (3) as the supervisory signal. The constraint enforces the estimated normal to be as close to the aligned normal as possible, which is described by a loss function L_{norm} :

$$L_{norm} = \frac{1}{N_{norm}} \sum_p \mathcal{M}_p^M \mathcal{M}_p^P (1 - s(\boldsymbol{n}_p, \boldsymbol{n}_p^{align})) \tag{5}$$

where N_{norm} is the number of pixels located in Manhattan regions, and $\mathcal{M}_p^P$ indicates whether the pixel p locates in the planar regions, which we'll introduce how to detect them in the following section.

3.2. Co-planar constraint

Planar region detection. To enforce the co-planar constraint, we need to detect the piece-wise planar region correctly. Previous work [48] detects the planar regions by assuming the regions with homogeneous colors are planar. This simple strategy, however, usually leads to false detection or over-segmentation producing false supervisory signals. We propose a novel planar region detection method, as shown in Fig. 3, which integrates both the color and the online updated geometry information to extract the planar areas more reliably.

The key idea is that we adopt a novel dissimilarity map in the following graph-based segmentation. This dissimilarity takes the color, normal, and the plane-to-origin distance into consideration. We use the aligned normal to derive the dissimilarity instead of the estimated normal since we found the latter is too noisy. Let the 3D coordinates of a pixel $\boldsymbol{p}$ to be $\boldsymbol{X}_p$. Suppose this 3D point lies in the plane where the

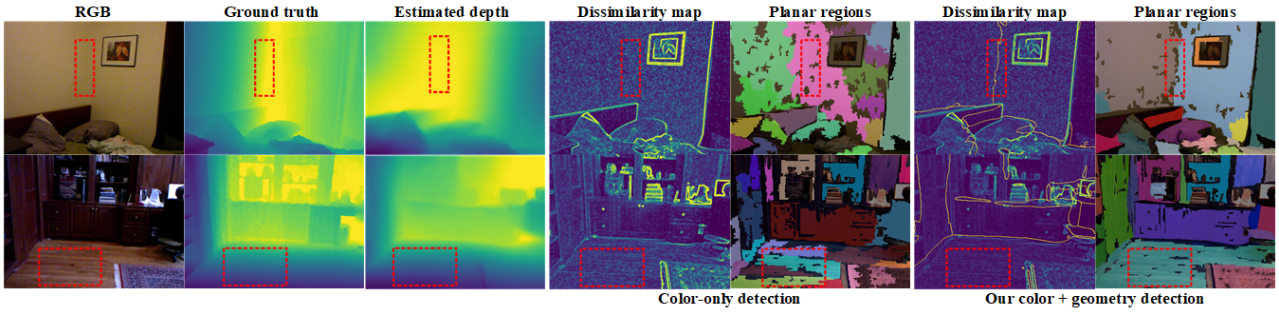

Figure 4. The proposed planar region detection during training. From the left to the right columns: the input images, the groud-truth depth, the estimated depth, the dissimilarity map, and the planar regions detected by only colors [48] and our method based on the color and geometric information. **First row**: Two walls cannot be distinguished by colors, but can be separated by our method. **Second row:** The floor is over-segmented by using only colors but can be correctly detected by our method.

normal is the aligned normal $\boldsymbol{n}_p^{align}$. The plane-to-origin distance is computed as

$$d_p = -\boldsymbol{X}_p^{\mathrm{T}} \boldsymbol{n}_p^{align}. \tag{6}$$

Let q be the adjacent pixel of p. The normal dissimilarity between them is defined as the Euclidean distance between the two vectors:

$$\mathcal{D}_n(p,q) = \|\boldsymbol{n}_p^{align} - \boldsymbol{n}_q^{align}\|. \tag{7}$$

Denoting the minimum and maximum dissimilarities among all the adjacent pixels by D_n^{max}, D_n^{min} respectively, we define a $[\cdot]$ operator to normalize the dissimilarity via

$$[\mathcal{D}_n(p,q)] = (\mathcal{D}_n(p,q) - D_n^{min})/(\mathcal{D}_n^{max} - \mathcal{D}_n^{min}). \tag{8}$$

The plane-to-origin distance dissimilarity is defined as

$$\mathcal{D}_d(p,q) = |d_p - d_q|. \tag{9}$$

The geometric dissimilarity combines the normalized version of the two dissimilarities as

$$\mathcal{D}_g(p,q) = [\mathcal{D}_n(p,q)] + [\mathcal{D}_d(p,q)]. \tag{10}$$

The color dissimilarity is computed as

$$\mathcal{D}_c(p,q) = \|\boldsymbol{I}_p - \boldsymbol{I}_q\|, \tag{11}$$

where $\boldsymbol{I}_p, \boldsymbol{I}_q$ are the RGB colors. Finally, we get the dissimilarity combining both the color and geometric information by

$$\mathcal{D}(p,q) = \max([\mathcal{D}_c(p,q)], [\mathcal{D}_g(p,q)]). \tag{12}$$

Based on the dissimilarity, we apply the graph-based segmentation [11] and filter out small areas to obtain the planar regions following [48]. The advantage of using such a dissimilarity definition can be seen in Fig. 4. Comparing with using only the color information, our method avoids false planar regions that cannot be distinguished by colors and also over-segmentation caused by different colors.

Note that our planar region segmentation is be updated during training. As the training progresses, the gradually improved depth leads to better segmentation and vise versa.

Generate the co-planar depth. After detection of planar regions, we invoke the co-planar constraint to flatten the 3D points located within those plane regions. The first step is plane fitting for 3D points within the planar region. We obtain the plane parameters $\boldsymbol{\theta} = -\boldsymbol{n}/d \in \mathbb{R}^3$ as previous work[27, 48] by solving the least squares problem

$$\boldsymbol{X}^{\mathrm{T}} \boldsymbol{\theta} = \mathbf{1}, \tag{13}$$

where each column of $\boldsymbol{X} \in \mathbb{R}^{3\times N}$ represents a 3D point within the planar region. After that, the inverse depth ρ_p of the pixel $\boldsymbol{p}$ by plane fitting is computed as

$$\rho_p^{plane} = \boldsymbol{\theta}^{\mathrm{T}} \mathbf{K}^{-1} \boldsymbol{p} = 1/D_p^{plane}, \tag{14}$$

where $\mathbf{K}$ represents the camera intrinsic matrix. We then transform the inverse depth to the depth D_p^{plane} with the maximum and minimum protection following [19, 20, 48].

Co-planar loss. The depth D_p^{plane} obtained from plane fitting is then used as an extra signal to constrain the estimated depth. The loss function is defined as

$$L_{plane} = \frac{1}{N_{plane}} \sum_p \mathcal{M}_p^P \left| D_p - D_p^{plane} \right|, \tag{15}$$

where N_{plane} is the number of pixels within the planar regions $\mathcal{M}^P$.

3.3. Total loss

We use the image patches instead of individual pixels to compute the photometric loss as suggested in [48], which is defined as the combination of L1 loss and a structure similarity loss SSIM[53]:

$$\begin{aligned} L_{photo} &= \omega L_{SSIM} + (1-\omega)\|I_t[\mathcal{N}_p^t] - I_s[\mathcal{N}_p^{t\to s}]\|_1 \\ L_{SSIM} &= SSIM(I_t[\mathcal{N}_p^t], I_s[\mathcal{N}_p^{t\to s}]) \end{aligned} \tag{16}$$

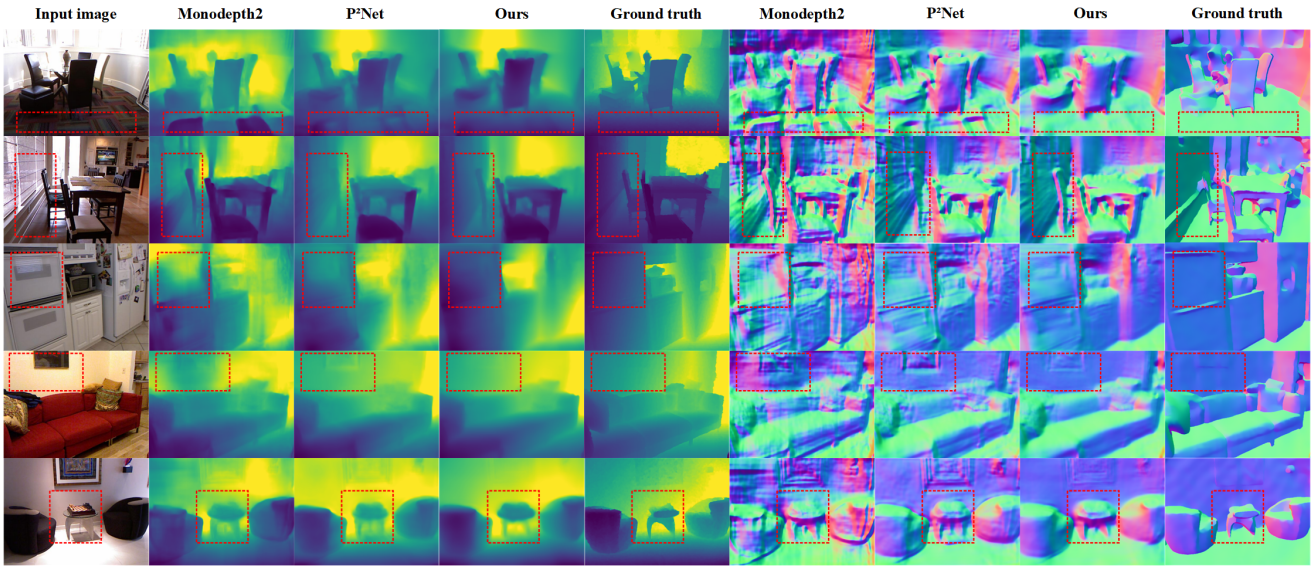

Figure 5. Visualization of the NYUv2 results, better viewed by zooming on screen. The depth results are on the left columns, and the surface normal results are on the right columns. The results of Monodepth2[20], P^{2}Net[48], and the ground-truth depth / normal are presented for comparison. Compared with P^{2}Net[48] and Monodepth2[20], our method obtains better surface normal and depth estimation as indicated by the red rectangles. Please refer to the Tab. 1 and Tab. 2 for the quantitative results.

Method	Sup.	RMS↓	AbsRel↓	Log10↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Hu et al.(2019)[21]	√	0.530	0.115	0.050	86.6	97.5	99.3
Yin et al.(2019)[47]	√	0.416	0.108	0.048	87.5	97.6	99.4
AdaBins(2021)[2]	√	0.364	0.103	0.044	90.3	98.4	99.7
Niklaus et al.(2019)[33]	√	0.300	0.080	0.030	94.0	99.0	100.0
PlaneNet(2018)[30]	√	0.514	0.142	0.060	81.2	95.7	98.9
PlaneReg(2019)[49]	√	0.503	0.134	0.057	82.7	96.3	99.0
MovingIndoor(2019)[51]	×	0.712	0.208	0.086	67.4	90.0	96.8
Monodepth2(2019)[20]	×	0.600	0.161	0.068	77.1	94.8	98.7
P^{2}Net(2020)[48]	×	0.561	0.150	0.064	79.6	94.8	98.6
Ours	×	**0.540**	**0.142**	**0.060**	**81.3**	**95.4**	**98.8**
Ours + pp	×	**0.534**	**0.140**	**0.060**	**81.7**	**95.5**	**98.8**

The first two blocks list the results of supervised methods. The second block contains the supervised methods with plane detection. The third and fourth blocks list the results of self-supervised methods. ↓ indicates the lower the better, ↑ indicates the higher the better. Our approach performs best among the self-supervised ones.

√ - supervised learning
× - self-supervised learning
pp - with post processing as in [19]

Table 1. Depth estimation results on NYUv2 dataset.

where $\mathcal{N}_p$ denotes the local window surrounding $\boldsymbol{p}$. ω is the relative weight of two parts and set as 0.85 the same as previous work[20]. We also adopt the edge-aware smoothness loss

$$L_{smooth} = |\partial_x \rho_t| \, e^{-|\partial_x I_t|} + |\partial_y \rho_t| \, e^{-|\partial_y I_t|}, \quad (17)$$

where $\rho_t \leftarrow \rho_t / \overline{\rho_t}$ is the mean-normalized inverse depth, and ∂_x, ∂_y are the gradients along the x and y directions. The overall loss is defined as

$$L = L_{photo} + \lambda_1 L_{smooth} + \lambda_2 L_{norm} + \lambda_3 L_{plane}, \quad (18)$$

where λ_1, λ_2 and λ_3 are set to 0.001, 0.05, 0.1, respectively.

4. Experimental results

We train our model on the NYUv2 dataset [39] using the data split the same as the previous work [51][48], and evaluate our methods on NYUv2[39], ScanNet[7], and InteriorNet[28] datasets. We detect the vanishing points on the training images and skip 18 image sequences that fail to detect valid vanishing points. This results in 21465 monocular training sequences and 654 images for validation. Each monocular training sequence consists of five frames. Our network model adopts the same architecture as [48].

We compare our method with the state-of-the-art methods of monocular depth estimation. Apart from depth estimation, we also evaluate the performance of surface normal estimation, and present ablation studies about the effectiveness of the proposed supervisory signals, and using different network architectures. More results can be found in the supplementary material.

Method	Train	Mean↓	11.2° ↑	22.5° ↑	30° ↑
Surface normal estimation networks					
3DP(2013)[13]	√	33.0	18.8	40.7	52.4
Fouhey et al.(2014)[14]	√	35.2	40.5	54.1	58.9
Wang et al.(2015)[41]	√	28.8	35.2	57.1	65.5
Eigen et al.(2015)[9]	√	23.7	39.2	62.0	71.1
Surface normal computed from the depth					
GeoNet(2018)[34]	√	36.8	15.0	34.5	46.7
DORN(2018)[15]	√	36.6	15.7	36.5	49.4
MovingIndoor(2019)[51]	×	43.5	10.2	26.8	37.9
Monodepth2(2019)[20]	×	45.1	10.4	27.3	37.6
P^2Net(2020)[48]	×	36.6	15.0	36.7	49.0
Ours	×	**34.5**	**21.9**	**44.4**	**55.2**
Ours + pp	×	**34.2**	**22.6**	**44.7**	**55.4**

Table 2. Surface normal estimation results on NYUv2. We report the results of surface normal estimation networks in the first block. The normal results computed from the depth networks are in the second and the third block, where '√' denotes supervised methods, and '×' denotes self-supervised ones. The normal computation is the same for all methods. Our method outperforms existing monocular depth estimation methods in surface normal estimation.

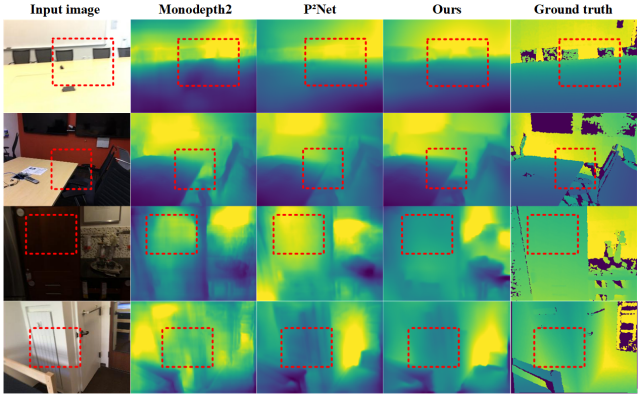

Figure 6. **ScanNet results** with the trained model on NYUv2. The holes in the ground truth are excluded from evaluation.

Method	RMS↓	AbsRel↓	Log10↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Monov2[20]	0.451	0.191	0.080	69.3	92.6	98.3
P^2Net [48]	0.420	0.175	0.074	74.0	93.2	98.2
P^2Net-finetune	0.412	0.172	0.073	74.3	93.5	98.4
Our	**0.400**	**0.165**	**0.070**	**75.4**	**93.9**	**98.5**

Table 3. **ScanNet results** with the trained model on NYUv2.

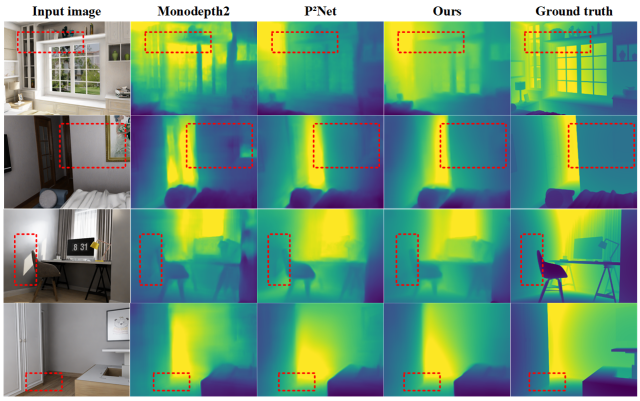

Figure 7. **InteriorNet results** with the trained model on NYU V2.

Method	RMS↓	AbsRel↓	Log10↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Monov2[20]	0.817	0.368	0.124	58.6	81.5	89.8
P^2Net [48]	0.737	0.346	0.115	64.2	83.3	90.2
P^2Net-finetune	0.736	0.340	0.114	64.4	83.3	90.3
Our	**0.715**	**0.330**	**0.111**	**66.0**	**84.0**	**90.5**

Table 4. **InteriorNet results** with the trained model on NYUv2.

4.1. Implementation details

The network is trained for a total of 50 epochs with a batch size of 32 based on the pre-trained model [48]. We use Adam optimizer and a multi-step learning rate reduction strategy. We set the initial learning rate as 10^{-4}, then decay it by 0.1 at the 26th epoch and 36th epoch. We perform random flipping and color augmentation during training. All images are firstly undistorted and cropped by 16 pixels from the border, and then scaled to 288×384 for training. The camera intrinsic parameters come from the official specification [39], and are adjusted to be consistent with the image cropping and scaling. We follow the same criteria used in [20, 48] for evaluation. Namely, we cap the depth to $10m$ and use the median scaling strategy to avoid the scale ambiguity of monocular depth estimation. The evaluation metrics include root mean squared error (RMS), absolute relative error (AbsRel), mean log10 error (Log10), and the accuracy under threshold $(\delta_i < 1.25^i, i = 1, 2, 3)$.

4.2. Results on NYUv2 Dataset

Depth estimation. The quantitative results of depth estimation are listed in Tab. 1. The results show that our method outperforms MovingIndoor[51] and P^2Net[48], the state-of-the-art self-supervised methods on indoor monocular depth estimation, by a large margin. The results also show that our method surpasses some supervised approaches. The depth estimation results are visualized in Fig. 5. We can see that our method obtains more accurate indoor structures and smoother planes than existing methods.

Surface normal estimation. We also evaluate the surface normal estimation as shown in Tab. 2. Our method outperforms existing methods, and also some supervised methods[13, 34, 15]. Results are also shown in Fig. 5.

4.3. Results on ScanNet and InteriorNet

We use the model trained only on NYUv2 to evaluate our methods generalized to other indoor datasets. **ScanNet**[7] is captured with a depth camera attached to a iPad, containing around 2.5M RGBD video captured in 1513 scenes. We use the test split proposed by [48] which includes 533 images. The evaluation results are shown in Tab. 3 and Fig. 6. **InteriorNet**[28] is a synthetic dataset of indoor video sequences containing millions of well-designed interior design layouts, furniture and object models. Because there is no current official train/test split on InteriorNet for depth estimation, here we selected 540 images randomly from the HD7 data of the full dataset as test images. The evaluation

results are shown in Tab. 4 and Fig. 7.

Although ScanNet and InteriorNet have not been used for training, the results show that our method still generalizes well and outperforms existing methods.

Methods	RMS↓	AbsRel↓	Log10↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
P^2Net[48]	0.561	0.150	0.064	79.6	94.8	98.6
P^2Net-finetune	0.555	0.147	0.062	80.4	95.2	98.7
Coplanar-only	0.548	0.144	0.061	80.8	95.3	98.8
Normal-only	0.543	0.143	0.061	81.0	**95.5**	**98.9**
Our(full)	**0.540**	**0.142**	**0.060**	**81.3**	95.4	98.8

Table 5. Ablation study about using different supervisory signals. We evaluate the performances using only the Manhattan normal constraint (Normal-only), using only the co-planar constraint (Coplanar-only), and the proposed method (Our(full)). We also present the result of fine-tuned P^2Net model (P^2Net-finetune). Note all the models were trained with the same number of epochs for fair comparison.

4.4. Ablation study

To better understand the effectiveness of each part of our method, we perform an ablation study by changing various components of our model on NYU V2 dataset. We initialize the network with the pre-trained model [48] and train it with the proposed supervisory signals. The results are shown in Tab. 5. Either the Manhattan normal loss or the co-planar loss leads to depth estimations better than that of the original and the original-finetune methods. Incorporating them together leads to the maximum gain in performance.

We also test our method using different network architectures. As shown in Tab. 6, using the proposed supervisory signals, both models are improved, indicating our method is universal to different network architectures. But the results based on Monodepth2 are worse than those based on P^2Net. This is largely due to the patch-based photometric loss that is better for texture-less regions as suggested in [48].

4.5. Planar-region detection in training

We show the intermediate planar region detection results during training in Fig. 8. The results show that the planar region segmentation gradually improves with the updated depth and normal estimates. By contrast, the color-only method produces false planar regions as indicated by the red rectangles.

Train	RMS↓	AbsRel↓	Log10↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Using the Monodepth2 [20] architecture						
Original	0.600	0.161	0.068	77.1	94.8	98.7
Original-finetune	0.598	0.159	0.067	77.5	94.9	98.7
Ours	**0.564**	**0.151**	**0.065**	**79.1**	**95.0**	**98.8**
Using the P^2Net [48] architecture						
Original	0.561	0.150	0.064	79.6	94.8	98.6
Original-finetune	0.555	0.147	0.062	80.4	95.2	98.7
Ours	**0.540**	**0.142**	**0.060**	**81.3**	**95.4**	**98.8**

Table 6. Ablation study about using different network architectures. Our extra training losses improves both models, indicating our method is universal to different architectures.

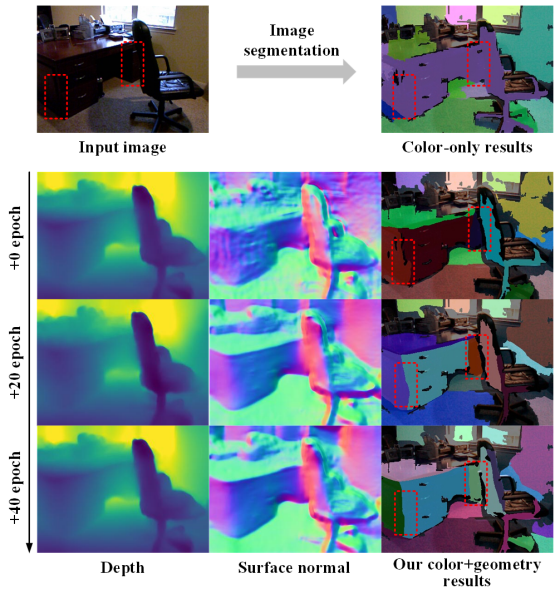

Figure 8. **First row**: The planar regions detected by the color-only method [48]. **Bottom rows**: The estimated depth, surface normal and segmentation results at different epochs on NYUv2. Our segmentation results gradually improve as the training progresses.

5. Limitation

We discuss the limitations of our method. The first limitation is that extracting dominant directions highly relies on the Manhattan world assumption. It may not work well in indoor scenes with irregular layouts containing slant planes. Possible solutions include using a relaxed version of Manhattan world assumption as in [37][54], or directly using the estimated direction from each detected vanishing point to derive the normal constraint. In other words, those dominant directions are not restricted to be mutually perpendicular. The second limitation is that the low quality of initial depth should be avoided. As our planar region detection relies on depth information, the low depth quality will deteriorate the segmentation results and generate false supervisory signals, which in turn prevent the network from converging to a good model. Our solution is to use a pre-trained depth model or train the model only with photometric and smoothness losses in early epochs. It leaves open to design a better planar region detector given low-quality initial depth estimates.

6. Conclusion

In this paper, we propose to leverage the structural regularities of indoor environments for monocular depth estimation. Two extra losses, Manhattan normal loss and co-planar loss, are used to supervise the depth learning. Those supervisory signals are generated on the fly during training by Manhattan normal detection and planar region detection. Our method achieves the state-of-the-art result on indoor benchmark datasets.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the European conference on computer vision*, pages 404–417. Springer, 2006.

[2] 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*, pages 4009–4018, 2021.

[3] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Tat-Jun Chin, Chunhua Shen, and Ian Reid. Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. *arXiv preprint arXiv:2006.02708*, 2020.

[4] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019.

[5] Alejo Concha, Muhammad Wajahat Hussain, Luis Montano, and Javier Civera. Manhattan and piecewise-planar constraints for dense monocular mapping. In *Robotics: Science and systems*, 2014.

[6] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, volume 2, pages 941–947. IEEE, 1999.

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[8] Erick Delage, Honglak Lee, and Andrew Y Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2418–2428. IEEE, 2006.

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2650–2658, 2015.

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. 2014.

[11] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[12] Alex Flint, Christopher Mei, Ian Reid, and David Murray. Growing semantically meaningful models for visual slam. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 467–474. IEEE, 2010.

[13] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3392–3399, 2013.

[14] David Ford Fouhey, Abhinav Gupta, and Martial Hebert. Unfolding an indoor origami world. In *Proceedings of the European conference on computer vision*, pages 687–702. Springer, 2014.

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[16] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Manhattan-world stereo. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1422–1429. IEEE, 2009.

[17] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 80–87. IEEE, 2009.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[19] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[20] 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.

[21] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019.

[22] Masaya Kaneko, Ken Sakurada, and Kiyoharu Aizawa. Tridepth: Triangular patch-based deep depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[23] Pyojin Kim, Brian Coltin, and H Jin Kim. Linear rgb-d slam for planar environments. In *Proceedings of the European conference on computer vision*, pages 333–348, 2018.

[24] Seungryong Kim, Kihong Park, Kwanghoon Sohn, and Stephen Lin. Unified depth prediction and intrinsic image decomposition from a single image via joint convolutional neural fields. In *Proceedings of the European conference on computer vision*, pages 143–159. Springer, 2016.

[25] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *International conference on 3D vision*, pages 239–248. IEEE, 2016.

[26] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *2009*

IEEE Conference on Computer Vision and Pattern Recognition, pages 2136–2143. IEEE, 2009.

[27] Boying Li, Danping Zou, Daniele Sartori, Ling Pei, and Wenxian Yu. Textslam: Visual slam with planar text features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2102–2108. IEEE, 2020.

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

[29] 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 International Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019.

[30] 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 International Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018.

[31] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2015.

[32] Xiaohu Lu, Jian Yaoy, Haoang Li, Yahui Liu, and Xiaofeng Zhang. 2-line exhaustive searching for real-time vanishing point estimation in manhattan world. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 345–353. IEEE, 2017.

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

[34] 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 International Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.

[35] Renato F Salas-Moreno, Ben Glocken, Paul HJ Kelly, and Andrew J Davison. Dense planar slam. In *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 157–164. IEEE, 2014.

[36] 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, 2008.

[37] Grant Schindler and Frank Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2004.

[38] Chang Shu, Kun Yu, Zhixiang Duan, and Kuiyuan Yang. Feature-metric loss for self-supervised learning of depth and egomotion. In *Proceedings of the European conference on computer vision*, pages 572–588. Springer, 2020.

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

[40] Rui Wang, David Geraghty, Kevin Matzen, Richard Szeliski, and Jan-Michael Frahm. Vplnet: Deep single view normal estimation with vanishing points and lines. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 689–698, 2020.

[41] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[42] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6101–6108. IEEE, 2019.

[43] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer. Pop-up slam: Semantic monocular plane slam for low-texture environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1222–1229. IEEE, 2016.

[44] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 0–0, 2018.

[45] 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 International Conference on Computer Vision and Pattern Recognition*, pages 225–234, 2018.

[46] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[47] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019.

[48] Zehao Yu, Lei Jin, and Shenghua Gao. P^2net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *Proceedings of the European conference on computer vision*, 2020.

[49] 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 International Conference on Computer Vision and Pattern Recognition*, pages 1029–1037, 2019.

[50] Huizhong Zhou, Danping Zou, Ling Pei, Rendong Ying, Peilin Liu, and Wenxian Yu. Structslam: Visual slam with building structure lines. *IEEE Transactions on Vehicular Technology*, 64(4):1364–1375, 2015.

[51] Junsheng Zhou, Yuwang Wang, Kaihuai Qin, and Wenjun Zeng. Moving indoor: Unsupervised video depth learning in

challenging environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8618–8627, 2019.

[52] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[53] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[54] Danping Zou, Yuanxin Wu, Ling Pei, Haibin Ling, and Wenxian Yu. Structvio: visual-inertial odometry with structural regularity of man-made environments. *IEEE Transactions on Robotics*, 35(4):999–1013, 2019.