StruMonoNet: Structure-Aware Monocular 3D Prediction

Zhenpei Yang[1], Li Erran Li[2,3], Qixing Huang[1]
[1]The University of Texas at Austin, [2]Columbia University,[3]Amazon

Abstract

Monocular 3D prediction is one of the fundamental problems in 3D vision. Recent deep learning-based approaches have brought us exciting progress on this problem. However, existing approaches have predominantly focused on end-to-end depth and normal predictions, which do not fully utilize the underlying 3D environment's geometric structures. This paper introduces StruMonoNet, which detects and enforces a planar structure to enhance pixel-wise predictions. StruMonoNet innovates in leveraging a hybrid representation that combines visual feature and a surfel representation for plane prediction. This formulation allows us to combine the power of visual feature learning and the flexibility of geometric representations in incorporating geometric relations. As a result, StruMonoNet can detect relations between planes such as adjacent planes, perpendicular planes, and parallel planes, all of which are beneficial for dense 3D prediction. Experimental results show that StruMonoNet considerably outperforms state-of-the-art approaches on NYUv2 and ScanNet.

1. Introduction

Monocular 3D prediction is a long-standing problem in 3D vision. Recent approaches [9, 8, 21, 35, 20, 12, 22], which apply end-to-end feature learning, have shown great promise of applying deep learning to this problem. 3D prediction involves many correlated tasks. An interesting problem is how to explore the interconnections among these tasks that can benefit each other. This paper studies the interconnections between predictions of local geometric elements such as depth and normal and predictions of middle-level planar structures rich in 3D scenes. Our goal is to answer critical questions in developing suitable geometric representations for plane detection and extracting rich relations among planes to enhance the predictions of depth, normal, and plane equations.

Specifically, we introduce StruMonoNet, which takes a single RGB image as input and outputs joint predictions of depth, normal, and a planar structure (See Figure 1). Instead of training a network to regress a fixed number of plane equations (c.f. [25]), StruMonoNet utilizes an intermediate representation that combines surfels (positions + normals) and dense visual features. This formulation enables a simple clustering module for plane detection, where visual features guide the clustering procedure through a trainable sub-module. It also fully incorporates depth/normal labels for plane detection through predicted surfels, which are unavailable in black box plane detection.

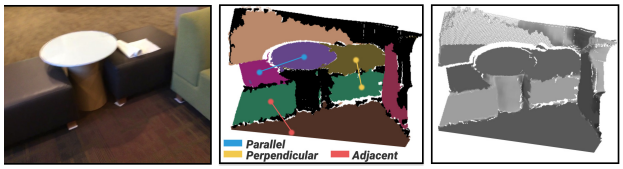

Figure 1. StruMonoNet takes a single RGB image of a 3D scene as input (Left) and outputs a joint prediction of the underlying planar structure and relations (Middle) and surfels (Right).

Unlike merely detecting individual planes, StruMonoNet detects and enforces geometric relations between planes, e.g., adjacent planes, perpendicular planes, and parallel planes. Enforcing such structures enhances the prediction accuracy of individual planes significantly. StruMonoNet introduces a novel plane synchronization module that automatically detects such relations and enforces them to enhance the predicted planes' accuracy.

StruMonoNet takes inspiration from the observation that depth and normal prediction errors of a deep-learning approach typically have large variance and small bias. Therefore, one can rectify the prediction error by applying suitable averaging operations. Although it is impossible to rectify the predictions across different images, StruMonoNet achieves the partial goal of averaging them among detected planar regions of each image. The improved predictions then propagate to non-planar regions. Note that the adjacency, perpendicular, and parallel planes are critical from this aspect. They allow us to incorporate more pixels for rectification.

Our approach outperforms the state-of-the-art approaches on two benchmark datasets ScanNet [6] and NYUv2 [26] for monocular depth prediction. We also achieve considerable improvements on normal prediction

(Table 3) on NYUv2 and plane detection (Table 5) on ScanNet over state-of-the-art methods.

In summary, our contributions include

- A hybrid representation that combines positions, normals, and visual features for joint predictions of planar structures and pixel-wise depth and normal.
- A synchronization module that detects planes and their geometric relations such as adjacent planes, perpendicular planes, and parallel planes.
- State-of-the-art results on depth prediction on ScanNet and NYUv2, state-of-the-art results on normal prediction on NYUv2, and state-of-the-art results on planar detection on ScanNet.

2. Related Work

Early monocular 3D perception approaches were based on geometric cues, such as shape-from-shading [43] and parallel lines [5]. These approaches place critical restrictions on the input images and do not generalize well to natural scenes. Recent monocular 3D perception approaches [32, 14, 13, 21, 35, 20, 12] leveraged cutting-edge machine learning techniques to learn mappings from visual features of the input to 3D geometric structures. In particular, very recent deep neural techniques [9, 8, 21, 35, 20, 12] have shown remarkable performance gains due to their ability to learn sophisticated visual features unavailable in hand-crafted visual features. Despite the significant progress in predicting depth, these approaches typically do not consider rich geometric structures in natural environments (e.g., primitive shapes and symmetric relations) beneficial for 3D depth perception.

Some recent works [23, 41] proposed to enforce different kinds of geometric constraints in the depth prediction network, for example, local planar structure [23]. Such approaches are shown to considerably boost the state-of-the-art performance on depth prediction, revealing the potential of geometric constraints. Compared with [23], our method takes a step further in that we are not constrained to local planar patch. Our design enables us to aggregate across a much larger geometric neighborhood, greatly enhancing prediction precision. Furthermore, we also leverage the relational cues from planar patches to further improve the performance.

StruMonoNet is also motivated by recent advances in monocular 3D structure prediction. [25] pioneered to predict planes from a single image using learning. [24, 42] further enhanced the performance with new prediction modules. Besides planes, [45] proposed to predict semantic-line structures from a single image. [46] generalized the results to achieve 3D wire-frame reconstruction. However, all these methods mainly focus on structure prediction. They do not focus on depth and normal prediction. In contrast, StruMonoNet integrates the predictions of depth, normal, and planar structure. It first combines visual features and predictions of depth and normal to predict the planar structure through a plane synchronization module. The resulting planar structure is then used to rectify predictions of depth and normal.

StruMonoNet is relevant to the methodology of establishing a neural network from a source domain to a target domain by composing two neural networks through an intermediate domain. This methodology has been adopted across many AI tasks. Examples include learning a machine translator between two minor languages by composing machine translators via a mother language [19], solving 6D object pose prediction via intermediate keypoint detections [1, 31, 28, 36, 27, 29, 34], and predicting 3D human poses through 2D keypoint predictions [44]. This paper innovates in aggregating predicted point positions, point normals, and point descriptors as an intermediate feature representation to predict planar structures. We also introduce a novel rectification module that leverages predicted planar structures to refine depth and normal.

3. Approach

In this section, we present the technical details of StruMonoNet. We begin with the problem statement and an overview of StruMonoNet in Section 3.1. Section 3.2 to Section 3.4 elaborate on the design of each module. Section 3.5 discusses network training.

3.1. Problem Statement and Approach Overview

Problem statement. Consider a single RGB image $I \in \mathcal{R}^{m \times n \times 3}$ with known intrinsic matrix $K \in \mathcal{R}^{3 \times 3}$ ($m = 480$ and $n = 640$ for this paper). The goal of StruMonoNet is to predict a set of surfels $\mathcal{S} = \{s\}$ that encodes the 3D position and normal associated with each pixel in the camera coordinate system and a collection of planar patches $\mathcal{P} = \{p\}$. Here each plane p collects indices $\mathcal{I}$ of the points that belong to this plane and the associated plane equation $(d_p, \boldsymbol{n}_p)$, where d_p and $\boldsymbol{n}_p$ are distance to the origin and plane normal, respectively. In particular, predictions of depth, normal, and plane equations are consistent with each other.

Overview of StruMonoNet. As illustrated in 2, StruMonoNet has three components. The design emphasizes the combination of geometric representations and feature descriptors. Specifically, the first component outputs an initial prediction of the surfels $\mathcal{S} = \{s\}$ and high dimensional descriptors. The descriptor is used later for extracting semantic features such as plane embedding, i.e., an embedding that distinguishes pixels belongs to different planes. This module also predicts pixels that lie at the

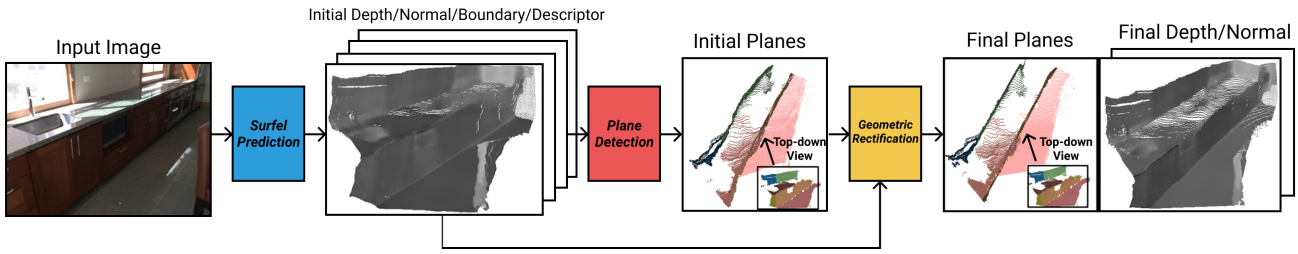

Figure 2. This figure illustrates the pipeline of StruMonoNet, which consists of three components. The first component provides initial predictions of depth/normal/boundary/descriptor. The second and component perform plane detection. The third component synchronize the detected planes and refines the surfels among non-planar regions. We illustrate the top-down view of the predicted surfels in the plane prediction figures to highlight the effect of geometric rectification.

intersection of 3D planes. They will be used to link adjacent planes when performing plane synchronization.

The second component performs plane detection through a generalized mean-shift procedure on the predicted surfels $\mathcal{S} = \{s\}$ to initialize plane detection. The clustering procedure is driven by relative weights that aggregate surfel feature and surfel geometry.

The third component performs geometric rectification using the detected planar structure. This is done using a synchronization module to detect pairwise relations between the detected planes and enforce them to enhance the plane predictions. This component also refines surfel geometry, taking the detected planes and the first component's output as input. StruMonoNet is trained by combining supervisions of pixel depth, pixel normal, planar patches, and relations between planes.

3.2. Surfel Prediction Module

The surfel prediction module includes a backbone encoder and four separate decoders for predicting depth, normal, descriptor (dimension = 32), and a heat-map that encodes boundary pixels. Following [23], we use DenseNet-161[15] as the backbone encoder. We add skip-connections between the corresponding encoder and decoder layers.

We determine the ground-truth boundary pixels using plane annotations. Specifically, we compute the intersection of 3D lines between all pairs of ground truth planes and then project the 3D lines into the image plane. Please refer to the supp. material for details.

3.3. Plane Detection Module

The plane detection module generalizes mean-shift clustering [4] to compute initial plane predictions. Denote $\mathcal{S}^{\text{init}} = \{s\}$ as the dense output of the first module. Let $\boldsymbol{s} = (\boldsymbol{p}_s; \boldsymbol{n}_s; \boldsymbol{f}_s)$ collect the position $\boldsymbol{p}_s$, normal $\boldsymbol{n}_s$, and descriptor $\boldsymbol{f}_s$ of s. Our mean-shift procedure computes a series of updated surfels $\mathcal{S}^{(t)} = \{s^{(t)}\}_{t=1}^{T}$ through the following recursion:

$$\boldsymbol{s}^{(t+1)} = \phi\Big(\sum_{s' \in \mathcal{S}^{(t)}} w(s^{(t)}, s', \theta_{\text{ms}}) \boldsymbol{s}' \Big/ \sum_{s' \in \mathcal{S}^{(t)}} w(s^{(t)}, s', \theta_{\text{ms}}) \Big) \tag{1}$$

where $\phi(\boldsymbol{s})$ is an operator that normalizes the normal component of $\boldsymbol{s}$ while keeps other elements of $\boldsymbol{s}$ unchanged.

Weighting module. Instead of performing range query (c.f. [4]), StruMonoNet employs a weighting sub-module $w(\boldsymbol{s}, \boldsymbol{s}', \theta_{\text{ms}})$ to predict the closeness between $\boldsymbol{s}$ and $\boldsymbol{s}'$. We define $w(\boldsymbol{s}, \boldsymbol{s}', \theta_{\text{ms}})$ by combing a geometric distance d_g and a feature distance d_f.

Specifically, we define

$$w(s, s', \theta_{\text{ms}}) = \exp\Big(- \frac{d_g^2(s, s', \theta_g)}{2\sigma_g^2} - \frac{d_f^2(s, s')}{2\sigma_f^2} \Big) \tag{2}$$

where σ_g and σ_f are trainable parameters.

For plane detection, we define the geometric distance and feature distance as

$$d_g^2(s, s', \theta_g) = \big((\boldsymbol{p}_s - \boldsymbol{p}_{s'})^T \boldsymbol{n}_s \big)^2 + \theta_g \|\boldsymbol{n}_s - \boldsymbol{n}_{s'}\|^2, \tag{3}$$

$$d_f^2(s, s') = \|\boldsymbol{f}_s - \boldsymbol{f}_{s'}\|^2 \tag{4}$$

where θ_g is another trainable parameter.

Plane extraction. Let $\mathcal{S}^T$ denote the updated surfels after mean-shift clustering, StruMonoNet employs the standard approach of binning $(\boldsymbol{p}_s^T \boldsymbol{n}_s, \boldsymbol{n}_s), s \in \mathcal{S}^T$ to determine the resulting clusters (c.f. [18]). The geometry of each detected plane is determined by averaging the normals and positions of the surfels inside the bin.

3.4. Geometric Rectification Module

This module detects and enforces relations between planes to rectify the geometry of the detected planes. As illustrated in Figure 3, StruMonoNet considers three types of relations, namely, adjacent planes, perpendicular planes, and parallel planes. Note that one pair of planes may possess multiple relations (e.g., perpendicular and adjacent). We enforce such relations through a synchronization

Figure 3. Illustrations of different types of planar relations. (a) Adjacent planes. (b) Perpendicular planes. (c) Parallel planes.

s	surfel	p	plane	b	boundary
$\mathcal{S}$	all surfels	$\mathcal{P}$	all planes	$\mathcal{B}$	all boundaries
p_s	position of s	n_s	normal of s	f_s	descriptor of s
d_p	distance of p	n_p	normal of p	$\bar{b}$	$K^{-1}(b_0, b_1, 1)^T$
t	clustering iteration	T	max clustering step	w_x	weight for term x
$\theta_{\text{ms}} = \{\theta_g, \sigma_g, \sigma_f\}$		trainable parameters for plane detection module			
$\theta_{\text{reg}}, \theta_{\text{perp}}, \theta_{\text{par}}, \theta_{\text{adj}}$		trainable parameters for plane synchronization module			

Table 1. We summarize here the notations and corresponding definitions used in the paper.

network derived from solving a non-linear robust least square formulation. The key idea is to minimize a robust norm to automatically detect and prune incorrect constraints (c.f. [2, 17]). In the following, we first introduce the objective function of the optimization problem. We then describe the induced synchronization module.

Objective function. Let $\mathcal{P} = \{p\}$ denote the detected planes, where $(d_p^{\text{init}}, \boldsymbol{n}_p^{\text{init}})$ encodes the initial plane equation. With $\mathcal{B} = \{b\}$ we denote candidate boundary pixels (i.e., the output of the first module) that are used to bridge adjacent planes. Each boundary pixel b encodes its homogeneous coordinate $\bar{\boldsymbol{b}} = K^{-1}(b_0, b_1, 1)^T$ and its latent feature $\boldsymbol{f}_b$ (from the output of the first module).

Motivated from the iteratively reweighted non-linear squares for robust regression [7], we set up the following objective function to jointly optimize all plane questions $\{d_p, \boldsymbol{n}_p\}$.

$$\begin{aligned}\min_{\{d_p, \boldsymbol{n}_p\}} & \sum_{p\in\mathcal{P}} w_{\text{reg}}(p, p^{\text{init}}, \theta_{\text{reg}}) d^2_{\text{reg}}(p, p^{\text{init}}) \\ + & \sum_{p,p'\in\mathcal{P}} w_{t(p,p')}(p, p', \theta_{t(p,p')}) d^2_{t(p,p')}(p, p') \\ + & \sum_{p,p'\in\mathcal{P}^{\text{init}}} \sum_{b\in\mathcal{B}} w_{\text{adj}}(p, p', b, \theta_{\text{adj}}) d^2_{\text{adj}}(p, p', b) \end{aligned} \tag{5}$$

where $t(p, p') \in \{\text{perp}, \text{par}\}$ is the relation type between p and p'; the geometric distance measures d_{reg},d_{perp},d_{par},d_{adj} are defined as follows:

$$\begin{aligned} d^2_{\text{reg}}(p, p') &= (d_p - d_{p'})^2 + \alpha \|\boldsymbol{n}_p - \boldsymbol{n}_{p'}\|^2 \\ d_{\text{perp}}(p, p') &= \boldsymbol{n}_p^T \boldsymbol{n}_{p'} \\ d_{\text{par}}(p, p') &= \|\boldsymbol{n}_p - \text{sign}(\boldsymbol{n}_p^T \boldsymbol{n}_{p'}) \boldsymbol{n}_{p'}\| \\ d_{\text{adj}}(p, p', b) &= d_p / (\boldsymbol{n}_p^T \bar{\boldsymbol{b}}) - d_{p'} / (\boldsymbol{n}_{p'}^T \bar{\boldsymbol{b}}) \end{aligned} \tag{6}$$

where α is a tradeoff parameter between distance and normal. Specifically, w_{reg}, w_{perp},w_{par} and w_{adj} apply a similar formulation of (2), while replacing the geometric distances by d_{reg}, d_{perp}, d_{par} and d_{adj}, respectively.

Synchronization module. The synchronization module applies an iteratively reweighted scheme (c.f. [7]) to solve (5). This module starts with the output of the clustering module. At each step, it first applies the weighting module to determine the weight of each term. It then fixes the terms weight and applies one step of Gauss-Newton optimization to solve (5). The entire synchronization module is a feed-forward sub-network.

Let $\boldsymbol{v}^{(t)}$ collect a parameterization of plane parameters at iteration t. The Gauss-Newton step admits the following form:

$$\boldsymbol{v}^{(t+1)} = \boldsymbol{v}^{(t)} - H(\boldsymbol{v}^{(t)}, \Theta)^{-1} \boldsymbol{g}(\boldsymbol{v}^{(t)}, \Theta) \tag{7}$$

where H and g are the Gauss-Newton Hessian and the gradient $\boldsymbol{g}$ evaluated at $\boldsymbol{v}^{(t)}$, respectively; Θ collects all the parameters. Applying chain-rule, we use the following recursion to derive the derivatives between $\boldsymbol{v}$ and Θ for network training:

$$\begin{aligned} \frac{\partial \boldsymbol{v}^{(t+1)}}{\partial \Theta} = & (I - H^{-1} \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{v}}) \frac{\partial \boldsymbol{v}^{(t)}}{\partial \Theta} - H^{-1} \cdot \frac{\partial \boldsymbol{g}}{\partial \Theta} \\ & + H^{-1} \big(\frac{\partial H}{\partial \boldsymbol{v}} \frac{\partial \boldsymbol{v}^{(t)}}{\partial \Theta} + \frac{\partial H}{\partial \Theta} \big) H^{-1} \boldsymbol{g}. \end{aligned}$$

Refinement module We also refine the depth/normal among the predicted non-planar region by treating this goal as a variance of the image in-painting problem. This is done by a small encoder-decoder network. The input consists of predicted offsets of depth/normal among the predicted planar regions obtained from the geometric rectification module. This module outputs the final offsets of depth/normal.

3.5. Network Training

We employ a combination of three groups of loss terms to train StruMonoNet. The first term applies loss on the first module's output to train the initial prediction. We use the SmoothL_1 [11] loss for depth, the L_2 loss for normal, and the binary cross-entropy loss for boundary pixel prediction. We train the descriptor using the contrastive loss [42] to differentiate pixels that belong to different planes. The second group again applies the contrastive loss [42] on the mean-shift clustering module's output (Eq 1). The third group combines a regression loss on the final depth/normal prediction, and a cross-entropy loss for plane relation detection. As formulations of these loss terms are relatively standard, we defer the details to the supp. material. We firstly train the first component for 15 epochs. Then we add the second component and train for 5 epochs. We train the last component for another 5 epochs.

Method	AbsRel	SqRel	Log10	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena et al. [33]	0.349	-	-	1.214	-	0.447	0.745	0.897
Eigen et al. [8]	0.158	0.121	0.067	0.639	0.215	0.771	0.950	0.988
Laina et al. [21]	0.127	-	0.055	0.573	-	0.811	0.953	0.988
Xu et al. [40]	0.121	-	0.052	0.586	-	0.811	0.954	0.987
Qi et al. [30]	0.128	-	0.057	0.569	-	0.834	0.960	0.990
Wang et al. [37]	0.156	0.118	0.067	0.643	0.214	0.768	0.951	0.989
Liu et al. [25]	0.142	0.107	0.060	0.514	0.179	0.812	0.957	0.989
Yu et al. [42]	0.134	0.099	0.057	0.503	0.172	0.827	0.963	0.990
Liu et al. [24]	0.124	-	0.073	-	0.395	-	-	-
Fu et al. [10]	0.115	-	0.051	0.509	-	0.828	0.965	0.992
Yin et al. [41]	0.108	-	0.048	0.416	-	0.875	0.976	0.994
Lee et al. [23]	0.110	-	0.047	0.392	-	0.885	0.978	0.994
Ours-baseline	0.113	0.070	0.049	0.407	0.148	0.868	0.978	**0.995**
Ours	**0.107**	**0.065**	**0.046**	**0.392**	**0.139**	**0.887**	**0.980**	**0.995**

Table 2. Depth evaluation results on NYUv2. Our method achieve state-of-the-art performance on all metrics. We take the reported numbers directly from the respective paper. And we left out "-" if the author does not report the corresponding metric.

Figure 4. Qualitative comparisons between StruMonoNet and state-of-the-art monocular depth prediction approach(Lee et al.[23]) on NYUv2. Our approach exhibits salient gains over planar regions, including floor, wall, and the planar surface of furniture.

4. Experimental Results

This section presents an experimental evaluation of StruMonoNet. Section 4.2 and Section 4.1 analyze the results of StruMonoNet and compare them against baseline approaches on two popular benchmark datasets NYUv2 and ScanNet, respectively. Section 4.3 presents an analysis of StruMonoNet.

4.1. Analysis of Results on NYUv2

Experimental setup. NYUv2 [26] is a popular benchmark dataset for single image depth estimation. We employ

Method	Mean	Median	$< 11.25°$	$< 22.5°$	$< 30°$
Eigen et al. [8]	23.7	15.5	0.392	0.620	0.711
GeoNet [30]	19.0	11.8	0.484	0.715	0.795
FrameNet [16]	21.6	13.5	0.437	0.657	0.742
VPLNet [38]	18.0	9.83	0.543	0.738	0.807
Ours	**16.8**	**9.68**	**0.557**	**0.750**	**0.819**

Table 3. Normal evaluation results on NYUv2. We report the mean and median of angular normal errors as well as the percentage of pixels whose errors fall within a varying threshold.

the same experimental setup of the state-of-the-art approach [23] on this dataset. There are 24231 training images and 654 testing images. Unlike ScanNet, NYUv2 does not provide annotated plane structures. We apply a RANSAC-based plane detection method to detect plane regions from the raw point cloud associated with each image. Then we employ an agglomerative clustering technique to merge similar planes. Please refer to the supp. materials on details of such automatic plane annotations. Overall, the resulting plane annotation is of adequate quality, although less accurate than ScanNet's annotation [25] due to the fact that Liu et al. [25] leverage instance segmentation to derive accurate planes. In this context, our goal is to assess the robustness of StruMonoNet from inexact plane annotations. Since we do not have accurate ground-truth labels for planes, we evaluate the depth and normal predictions. We generate the ground-truth normal for all training data following the same approach as described in [39, 30].

Analysis of depth prediction. Table 2 compares StruMonoNet with state-of-the-art depth prediction approaches on NYUv2. We employ widely used metrics such as absolute relative depth error (AbsRel), square root of mean square error (RMSE), and the percentage of relative depth errors that fall into a varying threshold $1.25^i, i = 1, 2, 3$. Please refer to [23] for a detailed explanation of each metric.

StruMonoNet reduces the depth error over the state-of-the-art method across all eight metrics. Note that although our approach jointly predicts depth, normal, and planar structures, we do not leverage any additional human annotation for NYUv2. Such improvements suggest that although we do not use ground-truth planes as supervision, the output of an off-the-shelf plane detection approach still offers considerable performance gains.

Figure 4 visualizes the difference between StruMonoNet and baseline approaches on predicted depth images. Note that StruMonoNet not only exhibits salient performance gains among structural regions like floor and walls but also in furniture surfaces where a salient planar structure can be found. Depth of non-planar regions is also improved, thanks to the refinement module that propagates the improvements among planar regions to non-planar regions.

Analysis of normal prediction. StruMonoNet outperforms state-of-the-art approaches for normal prediction (See Table 3). The StruMonoNet improves the mean/median angular error from the top-performing baseline from $18.0°/9.83°$ to $16.8°/9.68°$ respectively. Moreover, StruMonoNet improves the percentages of pixels whose normal error falls within $11.25°$ from 54.3% to 55.7%. Accurate normal enables accurate detection of pair-wise relations between planar structure, which will, in turn, boost the accuracy of normal prediction.

4.2. Analysis of Results on ScanNet

Experimental setup. We also benchmark our methods on ScanNet [6]. ScanNet contains 807 unique scenes, where each scene may contain multiple data sequences (captured from different trajectories). We random sample 28k images for training and 666 images for testing. Differently from [25, 42], our setup ensures the test images do not contain the same scene in the training set. Such a setup is necessary for an accurate evaluation of the depth estimation model. We train [25, 42] by removing the images in their training set that appears in the scene of the test set, resulting 31099 training instances. We also train the top-performing approach [22] on our training data. Furthermore, we compare against our baseline that only consists of a surfel prediction module to show the relative gains. We use the plane annotations provided in PlaneNet[25] for training StruMonoNet.

Analysis of depth prediction. Our baseline implementation already reduces the AbsRel error of the top-performing baseline Lee et al. [23] by 9.2%, i.e., from 0.119 to 0.108, thanks to the joint learning of depth and normal. StruMonoNet further improves on the baseline from 0.108 to 0.103. The improvements are consistent across all the error metrics, indicating the robustness of StruMonoNet. Figure 5 provides visual comparisons between StruMonoNet and the state-of-the-art approach [23]. We can see that StruMonoNet results in considerable performance gains among structural regions.

Analysis of plane detection. Table 5 compares the output of StruMonoNet and state-of-the-art monocular planar detection approach[42]. We employ the same metric as described in [25, 42]. Our approach achieves significant improvements over both pixel recall metrics and plane recall metrics, indicating the superior quality of our pixel-plane association as well as accurate 3-D plane prediction. Qualitative comparisons can be found in Figure 6. It can be seen that our approach yields a much accurate boundary than the baseline. Such improvements indicate the clear advantage of integrating visual features and surfel geometries, which enable us to incorporate interpretable geometric distances for clustering. From another perspective, this design maximizes the influence of depth and normal supervision on detecting planar structures. Please refer to supp. for more experimental results on plane detection.

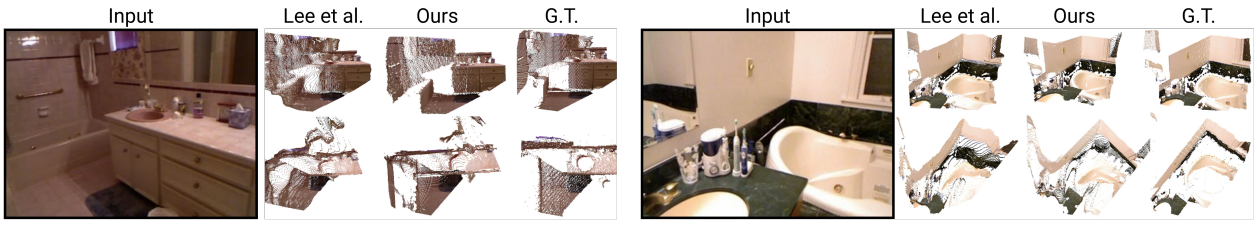

Figure 5. Qualitative comparisons between state-of-the-art monocular depth prediction approaches (Lee et al.[23]) and StruMonoNet StruMonoNet(Ours) on ScanNet.

Method	AbsRel	SqRel	Log10	RMSE	RMSELog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Liu et al. [25]	0.202	0.120	0.094	0.443	0.253	0.617	0.891	0.969
Yu et al. [42]	0.154	0.083	0.072	0.360	0.206	0.754	0.938	0.983
Lee et al. [23]	0.119	0.054	0.052	0.272	0.155	0.860	0.965	0.992
Ours-baseline	0.108	0.038	0.046	0.226	0.129	0.890	0.979	**0.997**
Ours.	**0.103**	**0.035**	**0.043**	**0.218**	**0.125**	**0.894**	**0.981**	**0.997**

Table 4. Depth evaluation results on ScanNet [6]. We compare against both state-of-the-art depth estimation method [23], and method that jointly predict depth and planar structure [25, 42]. Our approach achieve 9.2% improvement over top-performing method [23] on AbsRel. The improvements are consistent across all metrics.

Method	Pixel Recall			Plane Recall		
	0.1	0.3	0.5	0.1	0.3	0.5
Liu et al. [42]	0.096	0.298	0.402	0.051	0.153	0.205
Yu et al. [42]	0.192	0.442	0.550	0.115	0.275	0.344
Ours	**0.258**	**0.543**	**0.595**	**0.174**	**0.370**	**0.410**

Table 5. Plane detection results on ScanNet. We follows the same evaluation metric as Liu et.al [25]. We report the pixel and plane recall at depth difference thresholds 0.1m, 0.3m, 0.5m. Our approach yields considerable gains.

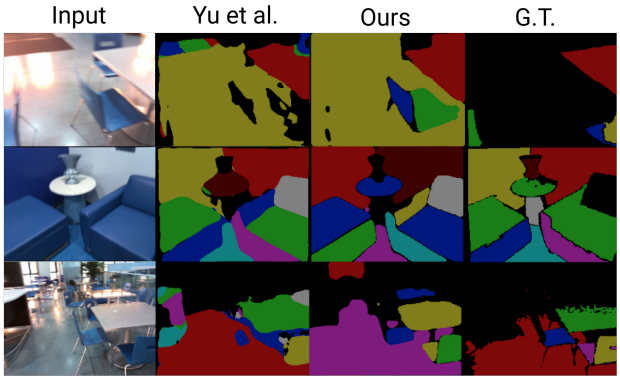

Figure 6. Qualitative comparisons between Ours and the state-of-the-art plane detection algorithm (Yu et.al [42]) on ScanNet.

4.3. Analysis of StruMonoNet

This section presents an analysis of different components of StruMonoNet. As shown in Table 6, we treat the full model of StruMonoNet as the baseline and report the percentages of increments in the AbsRel when removing a component. The AbsRel is reported based on the prediction (-D) and the rectified prediction (-D-G) that factors out the global scale of the prediction and the ground-truth. We also show the improvements on pixels that belong to large planes (occupies more than 8% of the total image pixels) and small planes (occupies less than 8% of the total pixels) to further understand the behavior of StruMonoNet.

Structure prediction module. Without the structure prediction module, the depth errors increase by 4.85% and 5.62% on ScanNet and NYUv2, respectively. The performance gaps are consistent with factoring out the global scale of each image, i.e., 12.3% (ScanNet) and 13.1%(NYUv2). These numbers show the importance of enforcing and detecting plane structures. Note that depth improvements on small planes are bigger than those on large planes as the predictions among large planes are already good, e.g., floors and walls, while the inter-plane constraints help rectify the depth of small planes.

Pairwise relations. Enforcing pairwise relations among predicted planes is a crucial component for StruMonoNet. Without this component, the depth error increases by 3.32%/7.4% (ScanNet) and 3.82%/7.8% (NYUv2) with respect to the metrics of (-D)/(-D-G), respectively. In particular, such performance gaps dominate the performance gaps when dropping the entire structure prediction module. In contrast to the marginal performance gains derived from enforcing pairwise constraints for structure prediction on images [3], these numbers show the importance of detecting and enforcing pairwise relations among 3D planar structures.

Method	No-Structure			No-Relative			No-Adjacency			No-Normal-Relation			No-Refine			
	All	Large	Small	All	Large	Small	All	Large	Small	All	Large	Small	All	No-P.	Large	Small
ScanNet-D	4.85	6.5	8.5	3.32	4.91	8.41	2.02	4.55	7.21	1.32	1.88	2.13	1.12	1.45	0.32	0.44
ScanNet-D-G	12.3	14.3	19.3	7.4	12.6	19.1	5.33	13.2	17.1	3.42	6.12	7.34	3.36	5.13	1.13	1.72
NYUv2-D	5.62	7.4	9.6	3.82	4.77	8.4	2.12	4.11	7.25	1.27	2.14	2.96	1.56	1.82	0.46	0.56
NYUv2-D-G	13.1	15.6	18.4	7.8	11.7	19.3	5.83	11.1	15.6	3.64	6.8	7.9	4.14	5.17	1.31	0.93

Table 6. We show error increments (percentages) in relative absolute depth (-D) when removing a component from the full model of StruMonoNet. (-D-G): Depth after removing the global-scale. (No-Struture): Remove the structure prediction module completely. (No-Relative): Remove the component of detecting end enforcing pairwise relations between planes. (No-Adjacency): Do not enforce depth continuity between adjacent planes. (No-Normal-Relation): Do not enforce relations between planar normals. (No-Refine): no refinement among non-planar regions. (All): all pixels. (Large): pixels of large planes (>8% total pixels). (Small): pixels of small planes (<8% total pixels). (No-P.): pixels of non-planar regions.

Adjacent planes. Table 6 shows that removing the relations between adjacent planes has significant impacts on the performance of StruMonoNet. The depth error increases by 2.02%/5.33%(ScanNet) and 2.12%/5.83%(NYUv2), respectively. Moreover, the improvements in depth prediction are salient on small planes, i.e., 7.2%/17.1%(ScanNet) and 7.25%/15.6%(NYuv2). These results suggest that the geometric approach of integrating normal prediction and depth continuity between adjacent planes is critical. In particular, the improvements on small planes largely come from this constraint, e.g., depth continuity between adjacent planes and relations between plane normals.

Relations between plane normals. The accuracy of plane normals also impacts the rectified planes, which influence the depth accuracy. Geometric relations between plane normals, e.g., perpendicular planes and parallel planes, are important for normal prediction. As shown in Table 6, dropping these relations leads to 1.32%/3.42%(ScanNet) and 1.27%/3.64%(NYUv2) increments in depth error. Moreover, the improvements on small planes are bigger than those on large planes; we can again understand this from the fact that the normal accuracy on small planes is relatively low, and enforcing the inter-plane constraints can rectify the normal of these planes.

Refinement module. The surfel refinement module refines the depth and normal prediction after the planar rectification step, which enhances pixel depth among non-planar regions of a 3D scene, is also essential for StruMonoNet. Without this module, the mean depth error increases by 1.12%/3.36% (ScanNet) and 1.56%/4.14% (NYUv2). We can also see that this module significantly impacts the non-planar regions when compared to the planar regions, i.e., 1.45%/5.13% (ScanNet) and 1.82%/5.17%(NYUv2). Such differences indicate strong correlations between planar regions and non-planar regions, and improved depth among planar regions helps that among non-planar regions.

Alternative structure prediction modules. We have tested alternative structure detection approaches that first detect planes and edges and then solve a global optimization to jointly refine pixel depth, pixel normal, and planes (the same as SURGE [37]). However, we found that our approach outperforms these network designs considerably, which only achieved marginally performance gains from the baseline. In contrast, the advantages of StruMonoNet come from combing geometric and visual features for plane detection, inter-plane relations, and the refinement network for rectifying non-planar regions.

5. Conclusions

This paper introduces StruMonoNet, a monocular depth prediction network that leverages planar structures of the underlying 3D environment to enhance depth prediction. StruMonoNet innovates in combing visual features and a surfel representation to predict and enforce planar structures. This approach offers a unique way to detect not only individual planes but also rich geometric relations among them (e.g., adjacent, perpendicular, and parallel planes). Experimental results show that the latter is critical for boosting the performance of depth prediction. Overall, StructMonoNet outperforms state-of-the-art monocular depth prediction networks by considerable margins on both NYUv2 and ScanNet.

There are ample opportunities for future work. The plane structures can be considered an abstraction of the underlying 3D environment. An interesting question is how to extend the idea to develop a hierarchical structural representation of 3D scenes for depth prediction. Another direction is to explore how to leverage such rich intermediate representations and object size priors to rectify the the absolute scale of depth prediction, which remains a challenge in monocular depth prediction.

Acknowledgement This work is supported in part by NSF IIS-2047677 and NSF IIS-1934932. The views presented in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: speeded up robust features. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part I*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006.

[2] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 521–528. IEEE Computer Society, 2013.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[4] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[5] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *Int. J. Comput. Vis.*, 40(2):123–148, 2000.

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017.

[7] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C. Sinan Gunturk. Iteratively re-weighted least squares minimization for sparse recovery, 2008.

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

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2366–2374, 2014.

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

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3827–3837. IEEE, 2019.

[13] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *Int. J. Comput. Vision*, 75(1):151–172, Oct. 2007.

[14] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *Int. J. Comput. Vis.*, 80(1):3–15, 2008.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[16] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas J Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8638–8647, 2019.

[17] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J. Guibas, and Qixing Huang. Learning transformation synchronization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8082–8091. Computer Vision Foundation / IEEE, 2019.

[18] John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.

[19] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351, 2017.

[20] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2215–2223. IEEE Computer Society, 2017.

[21] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, October 25-28, 2016*, pages 239–248. IEEE Computer Society, 2016.

[22] 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 preprint arXiv:1907.10326*, 2019.

[23] 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 preprint arXiv:1907.10326*, 2019.

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

[25] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single RGB image. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2579–2588. IEEE Computer Society, 2018.

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

[27] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, volume 11219 of *Lecture Notes in Computer Science*, pages 125–141. Springer, 2018.

[28] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G. Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 2011–2018. IEEE, 2017.

[29] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4561–4570. Computer Vision Foundation / IEEE, 2019.

[30] 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*, pages 283–291, 2018.

[31] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3848–3856. IEEE Computer Society, 2017.

[32] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, page 11611168, Cambridge, MA, USAh, 2005. MIT Press.

[33] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. NIPS'05, pages 1161–1168, Cambridge, MA, USA, 2005. MIT Press.

[34] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 428–437. IEEE, 2020.

[35] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6565–6574. IEEE Computer Society, 2017.

[36] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 292–301. IEEE Computer Society, 2018.

[37] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In *Advances in Neural Information Processing Systems*, pages 172–180, 2016.

[38] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 689–698, 2020.

[39] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015.

[40] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

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

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

[43] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, 1999.

[44] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 398–407. IEEE Computer Society, 2017.

[45] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 962–971, 2019.

[46] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7698–7707, 2019.