# Height and Uprightness Invariance for 3D Prediction from a Single View

Manel Baradad Antonio Torralba
Massachusetts Institute of Technology
mbaradad@mit.edu torralba@csail.mit.edu

## Abstract

*Current state-of-the-art methods that predict 3D from single images ignore the fact that the height of objects and their upright orientation is invariant to the camera pose and intrinsic parameters. To account for this, we propose a system that directly regresses 3D world coordinates for each pixel. First, our system predicts the camera position with respect to the ground plane and its intrinsic parameters. Followed by that, it predicts the 3D position for each pixel along the rays spanned by the camera. The predicted 3D coordinates and normals are invariant to a change in the camera position or its model, and we can directly impose a regression loss on these world coordinates.*

*Our approach yields competitive results for depth and camera pose estimation (while not being explicitly trained to predict any of these) and improves across-dataset generalization performance over existing state-of-the-art methods.*

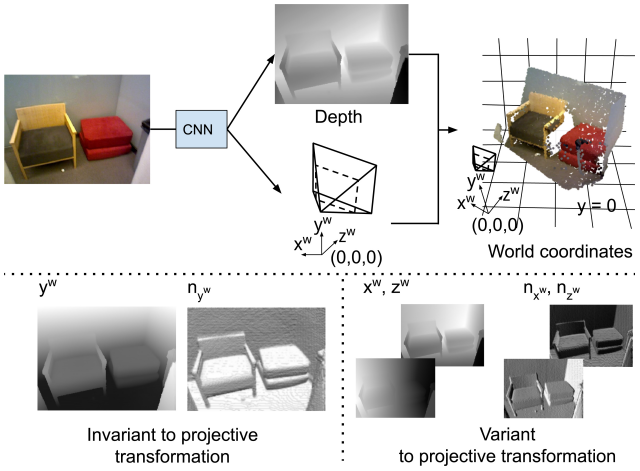


Figure 1. Given a single image, we predict 3D world coordinates (where $y = 0$ corresponds to the floor) for each pixel. To this end, we predict the camera intrinsics, the extrinsics and the depth per pixel, and use these to recover world coordinates. We note that, with this representation, $n_y$ and $y$ per pixel are invariant to a change in the position of the camera or its model.

## 1. Introduction

Scene understanding from single images has greatly improved in the last decade, with major successes in a wide variety of dense prediction tasks such as depth regression [11, 18, 26], intrinsic image decomposition [24, 29] and semantic segmentation [39, 43, 47], between others [41]. Though state-of-the-art methods for all these tasks use similar architectures and training techniques, there is an inherent difference between these tasks that is usually disregarded: whether the prediction for each pixel is invariant or not to projective transformations.

If we had access to an algorithm that could derender an image and produce novel views of the scene from different positions, some of these tasks would behave differently. For example, if a point on some element of the scene appeared in both views, it would have the same values for the semantics and the albedo, but it would have different values for the depth. That is, semantics and albedo are invariant to projective transformations, but depth is not.

Convolutional neural networks are somewhat invariant to this type of transformations and there are known techniques that can make them generalize better [2, 31, 42]. Additionally, augmenting the training data with random transformations of carefully chosen classes (such as affine transformations) has shown to improve generalization of CNN's at test time [8].

To exploit this robustness when predicting 3D structure from a single image, we propose using a neural network that predicts 3D coordinates in an appropriate reference frame (which we call *world* frame). Using this reference frame, one spatial dimension ($y^w$, the height relative to a reference plane) and one normal dimension ($n_y^w$, the upright normal) become invariant to projective transformations.

As seen in Figure 1, our model does not produce three independent coordinate maps. Instead, we propose predicting the camera model and the depth per pixel, which define the 3D coordinates for each pixel. This technique allows reducing the possible configurations for the point cloud of an image to those that are possible under the predicted camera

model.

Furthermore, a simple regression loss over both coordinates and normals has non-local effects that regressing other representations (such as depth) does not capture, as we illustrate in Section 3.2. Non-local losses and models, which exploit relations between distant pixels, have been shown to increase robustness and generalization for a wide set of inference problems [3, 35] and, particularly, for 3D regression problems [40].

When the testing distribution matches that of training, our method performs robustly on depth and extrinsics regression, leveraging both depth and extrinsics information already available in datasets such as Scannet [9]. Furthermore, when testing on different datasets than those used for training, our method outperforms state of the art methods that are trained with the same sources of data.

Finally, we show that the output of our system is particularly amenable to be used for downstream task. In particular, we show how to use its output to place 3D objects in arbitrary images in the wild with minimal user intervention.

## 2. Related work

**Depth and extrinsic parameter prediction.** State-of-the-art methods for regressing metric depth based on CNNs have made major progress by combining better losses [11, 40], better architectures [6, 21] or by factorizing the problem [22, 26]. However, these design choices are usually informed by train/test performance and ignore two challenges that appear when using these systems in images in the wild: (a) the distribution of the elements in the scene may change, and (b) the distribution of the camera intrinsics and extrinsics may change.

The first has attracted a lot of attention, particularly for the case where the objects are semantically the same but have different visual appearance (e.g. simulation vs real objects [1, 44]). In this work, we particularly address the latter, which has received less attention, although recent studies [14, 34] point out that this is an unresolved issue limiting generalization of these methods in practice.

Other methods avoid some of these two challenges by regressing non-metric depth, for which supervision is easier to acquire. These produce metric depth maps up to an unknown scale factor [23, 25], or up to an unknown non-linear transformation [5, 20]. By design, these methods are not able to regress metric depth, and their loss and their performance evaluation (without using groundtruth) is non-metric.

Learning to estimate camera intrinsics and extrinsics has also been studied [13, 36]. Though end-to-end approaches have shown promising results outperforming geometric based methods, [37] shows that leveraging dense predictions per pixel in both camera and world frames improves the performance compared to approaches that directly regress these parameters. They show that this is specially useful when testing on a different data distribution than the one used for training.

Additionally, unsupervised methods based on photometric consistency have been proposed to estimate both depth and camera poses when the focal length is either known [12] or not [46]. Despite this, methods based on photometric consistency are inherently limited to predict depth up to an unknown scale factor.

**Semantic priors for 3D structure prediction** Some of the most successful methods to obtain metric 3D from a single image use simple hand-crafted priors for the size and spatial extent of typical elements in natural scenes. These, methods, usually referred to as Single View Metrology [7], exploit such priors to derive the camera model and the structure for other elements.

One of the priors that these methods often use is constructed over the height and the upright normal. Certain semantic classes (such as people) can be modeled as standing planes whose height is distributed according to a Gaussian distribution. Works such as [19] and [32] use this fact to reconstruct the 3D structure of a scene from its semantic map. To do this, they start by manually constructing a prior for the heights of restricted set of classes (usually only people), and then refine this prior using a database of semantically annotated images. Once they have refined this prior, they are able to infer the 3D structure for a new image using its semantic annotations.

While some CNN learning based approaches have incorporated semantic knowledge into their pipeline [15], it has been pointed out [34] that CNNs can fail to learn these simple priors. With this, though [19] and [32] are prone to fail when their assumptions do not hold, they generalize better to scenarios that were not seen during training, such as novel viewpoints or different layouts of the scene.

**Non-local losses** Natural scenes often contain simple 3D structures which extend through large regions of an image. Consequently, the depth values for pixels across large regions of the image are closely related. Therefore, a loss that treats all pixels as independent entities does not properly capture the structure of the problem. The Virtual Normal method [40] uses this idea to derive a non-local loss that penalizes inconsistencies between distant pixels, by comparing estimated and ground truth planes formed by randomly sampled triplets of pixels. While these planes do not have to correspond to physical planes in the world, they capture the non-local structure of the depth map and achieved better results than methods using only local losses.

Similarly, PlaneRCNN [26] forces the prediction to be consistent in planar regions in the image. To this end, it first detects and estimates the parameters of planar regions,

and then composes them into a unified depth map for the whole image. Though PlaneRCNN relies on two independent stages, they show that reasoning explicitly about the planar structure of the scene improves test time performance.

## 3. Method

To obtain a 3D point cloud in world frame, our system first predicts the intrinsics and extrinsics of a pinhole camera and the depth per pixel, and then computes the 3D point-cloud from these. The losses we use at training time then only operate on the 3D point cloud, which allows us to impose priors on the structure of the world which are independent of the camera pose and parameters.

### 3.1. 3D Structure regressor

As depicted schematically in Figure 1, our network predicts the intrinsic and extrinsic camera parameters, where the camera pose is parameterized via the pitch, roll, and height of the camera to the ground plane, and the depth per pixel $z_i^c$.

To predict dense depth values per pixel, we use an encoder-decoder CNN with the same Hourglass architecture as [25]. We augment this CNN by substituting all its convolutional layers by CoordConv layers [28]. These layers add coordinate maps as an additional input to the convolutional layers in order to easily compute translationally variant functions. The reasoning behind this is that the function that our system should predict is not translation invariant in the image plane. For example, the extrinsics are tightly coupled with the position of the horizon line in the image plane.

To predict the camera parameters, we use the features from the second-to-last layer, average pooled and followed by three linear layers with ReLU activations after the first two layers. We then use a sigmoid to normalize the output values, and scale and bias them to known ranges. Assuming a pinhole camera and using the depth $z_i^c$ and the intrinsic matrix $\mathbf{K}$ estimated by the network, we obtain the 3D coordinates for a pixel with coordinates $(x_i^{im}, y_i^{im})$ in the camera reference frame as

$$\mathbf{c}_i^c = z_i^c \mathbf{K}^{-1} \begin{pmatrix} x_i^{im} \\ y_i^{im} \\ 1 \end{pmatrix} \tag{1}$$

Once we have obtained the 3D coordinates in the camera reference frame, we transform them into a reference frame where $n_y$ (upright normal) and $y$ (height) are invariant to the camera parameters. We refer to 3D coordinates in this reference frame as *world* coordinates $\mathbf{c}^w$. This reference frame compensates for camera height ($h^c$), roll rotation ($\mathbf{R}_{roll}$) and pitch rotation ($\mathbf{R}_{pitch}$) with respect to the floor plane that lies below the camera, and can be obtained from the camera coordinates simply as:

$$\mathbf{c}_i^w = \mathbf{R}_{roll}^{-1} \mathbf{R}_{pitch}^{-1} \mathbf{c}_i^c - \begin{pmatrix} 0 \\ h^c \\ 0 \end{pmatrix} \tag{2}$$

### 3.2. Loss

To account for the ambiguity between the camera parameters and depth, we use the following regression loss directly on the predicted world coordinates

$$\mathbf{L_{3D}} = \sqrt{\frac{1}{N} \sum_{i=0}^{N} \|\mathbf{c}_i^w - \mathbf{c}_i^{w*}\|^2} \tag{3}$$

where $\mathbf{c}_i^{w*}$ are the ground truth 3D coordinates of the point corresponding to pixel $(x_i^{im}, y_i^{im})$. This is in contrast to independently regressing the depth and the camera parameters, as this enforces both to be consistent, and at the same time it does not penalize ambiguous cases, since the 3D coordinates are still computed using the camera parameters. We note that other losses that are usually used to regress depth such as ordinal regression loss [11] could be adapted to regress our output representation. However, this simple RMSE loss we use tends to perform good for indoor scenes, where the depth range is limited.

From the depth values, we predict the world normals using the cross-product between the 3D coodinates of adjacent pixels in the image plane,

$$\mathbf{n}_{k,l}^w = (\mathbf{c}_{k,l}^w - \mathbf{c}_{k+1,l}^w) \times (\mathbf{c}_{k,l+1}^w - \mathbf{c}_{k,l}^w) \tag{4}$$

where $k, l$ are the row and column indices corresponding to index $i$. We then use a cosine similarity loss between the ground truth and predicted world normals.

This loss serves two purposes: first, it acts as a regularizer, following similar strategies as previous work [10], where the difference between the predicted and ground-truth depth gradients in the image plane are penalized. The second purpose is forcing distant pixels to be consistent for large planar regions through the predicted camera parameters: the predicted normal for a pixel corresponding to the ground plane has to be consistent with the predicted camera, as all the normals for these regions are connected through the predicted camera parameters. This loss is simply computed as

$$\mathbf{L_{Normal}} = \frac{1}{2N} \sum_{i=0}^{N} (1 - \mathbf{n}_i^w \cdot \mathbf{n}_i^{w*})^2 \tag{5}$$

Finally, the loss we use for training the system is simply the weighted sum of both terms,

$$\mathbf{L} = \mathbf{L_{3D}} + \lambda \mathbf{L_{Normal}} \tag{6}$$

In all our experiments, we use a weight of $\lambda = 1$.

**Loss Non-locality.** To illustrate that this loss has non-local effects we show that, for some particular cases, the 3D coordinates are constrained by the combination of the model and the priors that the network is biased to learn on $y^w$ and $n_y^w$.

For all pixels in the horizon line, their $y^w$ coordinate is defined by the camera height and, consequently, by three single predicted parameters in our model (the camera height, the roll and the pitch). For some predicted $\mathbf{c}^w$ for an image, perturbing any $y^w$ in the horizon line can only be accomplished by perturbing the camera height, which would cause a perturbation in all the $y^w$ on this line of the image plane.

Similarly, for planar regions such as the floor, the model may be able to learn a strong prior for $y^w \approx 0$ and $n_y^w \approx 1$, enforced by our loss. If this is the case, the depth values that the network should predict to be consistent with this prior are controlled by the camera intrinsics and extrinsics. Consequently, a perturbation that causes these parameters to change can potentially affect all the points corresponding to floor pixels.

# 4. Experiments

We train the model on center crops of $256 \times 192$, after resizing the smallest dimension of the original image appropriately. We train the model for 200k iterations with a batch size of 48, using Adam [16] with a starting learning rate of $10^{-3}$ and decreasing it to $10^{-4}$ after 100k iterations. Our trained models are available at: https://github.com/mbaradad/im2pcl.

## 4.1. Data

Our method relies on having ground-truth world coordinates $\mathbf{c}^{w*}$ for images taken with a diverse set of poses and intrinsics. These can be easily acquired from datasets containing video sequences with depth (such as Scannet [9]) or from datasets containing panoramic depth images (such as Matterport3D [4]). In the first case, the ground plane can be estimated by automatically fusing the videos into a single mesh and annotating it once per scene. In the second case, it can be robustly estimated if the images are captured with a tripod, by considering the region that lies below the camera.

In this paper, we only use Scannet, as it contains a more diverse set of extrinsics than Matterport3D (particularly, a more diverse range of camera heights). To generate ground truth poses, we use annotated floor pixels when there are more than 100 visible, fitting a plane to them and using this to estimate the extrinsics. When there are no floor points, we rely on the precomputed poses available on the dataset. We have empirically found our ground-plane based method to yield more accurate extrinsics than those already available in the dataset, which are computed by fusing all the images in a video sequence, and are prone to drift errors that make floor points not lie on a plane.

| Parameter | Range |
|---|---|
| Pitch | $(-58.73°, 11.46°)$ |
| Roll | $(-9.56°, 9.15°)$ |
| Camera height | $(1.10\ m, 2.77\ m)$ |
| Scannet FoV | $(57.54°, 59.02°)$ |
| Full FoV | $(30.00°, 59.02°)$ |

Table 1. Valid ranges for the camera parameters used to train the system, which are derived from those found in Scannet (for pitch, roll, camera height and Scannet FoV). The Full FoV range corresponds to the maximum FoV found in Scannet and a smaller minimum Field of View, to simulate images taken with cameras with a diverse set of focal lengths.

Finally, to compute the ground truth normals ($\mathbf{n}_i^{w*}$) for each image we first back-project each point into 3D using the ground truth depth. We then fit a plane to each point using k-nearest neighbors [33], where we use a maximum number of 300 neighbors and maximum search radius of 1m. This differs from the way in which we compute estimated normals ($\mathbf{n}_i^w$), as we require this computation to be fast and differentiable during training. We have empirically found this to yield better estimates of the normals than just considering the two closest neighbors in the image plane.

## 4.2. Camera model and priors

We assume a pinhole camera model with center projection and known ranges for the extrinsic and intrinsic parameters, shown in Table 1. These correspond to the 1st and 99th percentile of the empirical distribution of the Scannet dataset. As our method naturally handles images taken with different fields of view (FoV), we propose augmenting the dataset with random crops to simulate images taken with longer focal lengths. This allows the system to work on images in the wild and under a less restrictive setting. Since the camera FoV of the images in the Scannet dataset is wide (roughly $60°$), we can generate images of arbitrary smaller FoV simply by cropping.

For a fair comparison against previous methods, we train two systems: one with a reduced FoV matching the range found in Scannet (which we call Scannet FoV) and one with an extended range of FoV's (which we call Full FoV). The first one allows a fair comparison against previous systems (that are implicitly trained with this field of view) while the second allows testing the system on images in the wild, which may have a smaller field of view.

## 4.3. In-dataset performance

Results for both depth and camera parameter estimation are reported in Tables 2 and 3. Though our system is unable to outperform previous methods on all the metrics when testing in the validation set of Scannet, we note that our

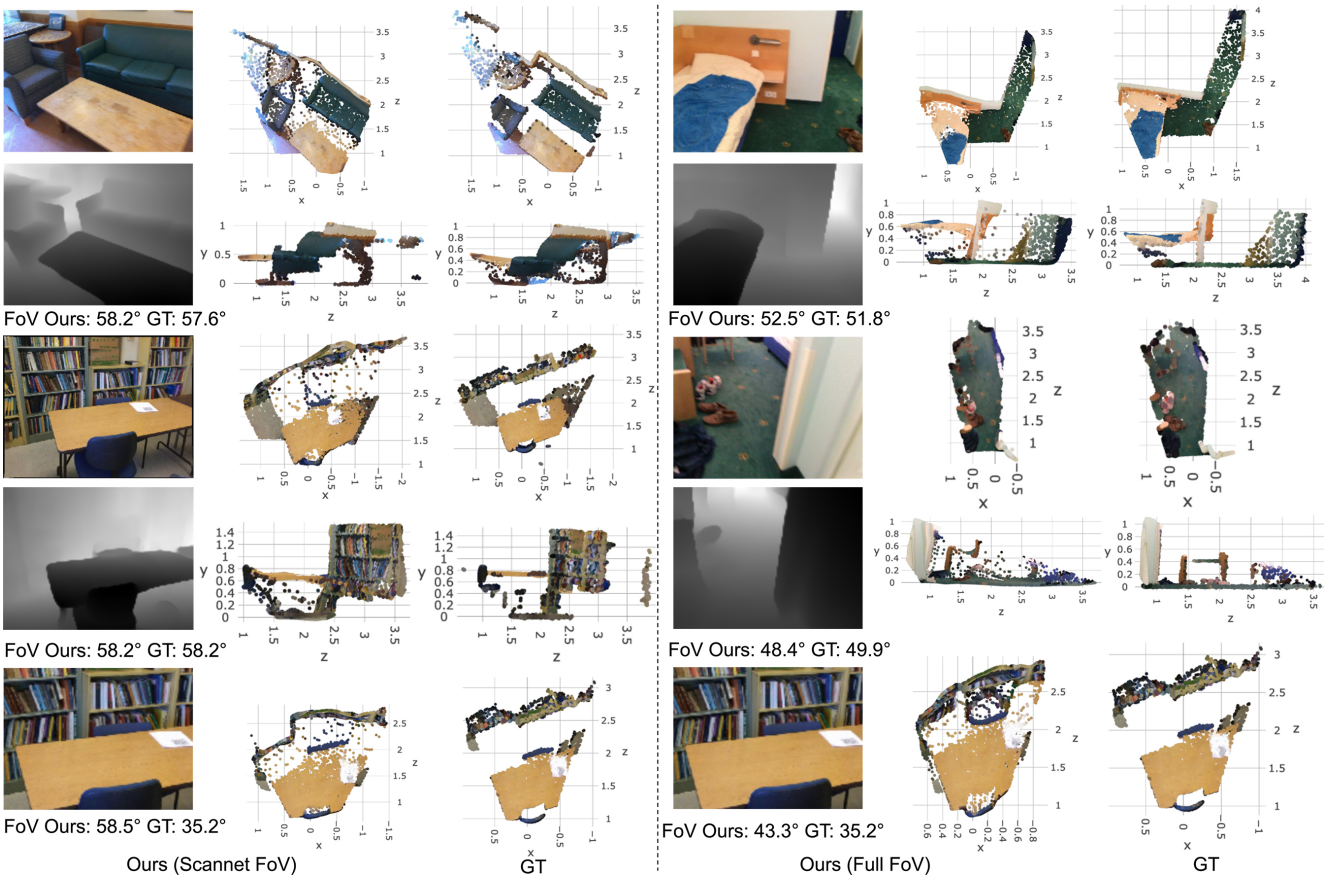


Figure 2. Results on random samples of the Scannet validation dataset for our method. Each pair of rows shows a top-down and right ortographic view of the reconstructed pointcloud. We note that our resconstructions properly capture planar regions such as floors and walls. As expected, the Full FoV model performs better than the Scannet FoV model on cropped images, as the last row depicts.

| Method | RMS | sq. rel. | log RMS |
|---|---|---|---|
| APMoE [17] | 0.38 | 0.10 | 0.22 |
| DORN[11] | **0.29** | **0.06** | **0.17** |
| Ours (Full FoV) | 0.38 | 0.11 | 0.21 |
| Ours (Scannet FoV) | 0.33 | 0.08 | 0.18 |

Table 2. Depth performance of our method on the validation set of Scannet, when fixing the field of view to that of the dataset (Scannet FoV) and for variable field of view (Full FoV) as in Table 1.

| Method | Pitch error (°) | | Roll error (°) | |
|---|---|---|---|---|
| | avg. | med. | avg. | med. |
| DeepHorizon [36] | 3.81 | 2.56 | 2.51 | 1.82 |
| Hold-Geoffroy et al. [13] | 3.53 | 2.33 | 2.15 | 1.50 |
| UprightNet [37] | **2.88** | **2.04** | 2.12 | 1.48 |
| Ours (Full FoV) | 4.29 | 2.94 | 2.61 | 2.00 |
| Ours (Scannet FoV) | 3.25 | 2.09 | **1.98** | **1.46** |

Table 3. Camera parameter performance on Scannet test set. Performance metrics for [36] and [13] correspond to those reported in [37].

method does not explicitly regress to either of these, and is solving a problem that handles a different type of uncertainty. For example, the $y$ for each point on a wall may be hard to estimate if the floor is not visible, but the depth may be easier to estimate for fixed focal length if there is some object in contact with the wall with a size that is easy to estimate.

Figures 2 and 3 show qualitative results for samples in the validation set of Scannet for both systems. As expected, the qualitative performance of the Scannet FoV system is better when tested on an appropriate field of view, but the Full FoV is able to produce reasonably good results for a wider set of FoV's. This outperforms the Scannet FoV when tested on the Full FoV data distribution, as depicted in the last rows of Figures 2 and 3.

### 4.4. Cross-dataset performance

To assess how our method performs on new datasets, we test it on the NYU [30] and SUN360 [38] datasets, which the system is not trained on. This evaluation method fol-

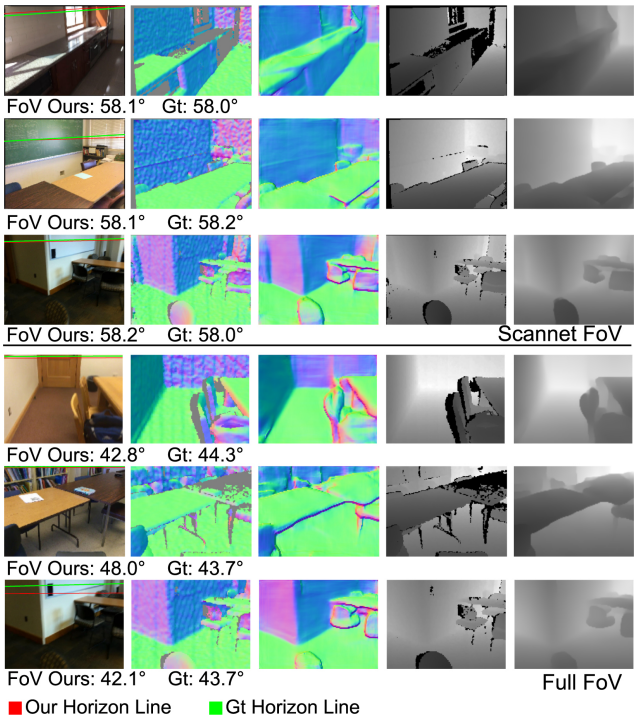


Figure 3. Predicted extrinsics, world normals ($n_y^w$) and depth for the Scannet FoV model (three top rows) and Full FoV model (three bottom rows). From left to right: the input image with the ground truth (green) and predicted (red) horizon lines; ground truth normals; estimated normals; ground truth depth; estimated depth. We note that the Full FoV system solves a problem with more uncertainty, and may have access to less information about the scene (because of the reduced FoV), as illustrated in the last row for each system.

| Method | RMS | rel | log10 |
|---|---|---|---|
| PlaneNet [27] | 0.858 | 0.220 | 0.114 |
| PlaneRCNN [26] | 0.644 | 0.164 | **0.077** |
| Ours (Full FoV) | 0.646 | 0.187 | 0.107 |
| Ours (Scannet FoV) | **0.566** | **0.158** | 0.082 |

Table 4. NYU cross-dataset depth performance, using Scannet data for training.

lows the same cross-dataset evaluation methods from [26] and [37], where authors report results for systems trained and validated using Scannet data, but tested on NYU and SUN360 respectively. Performance metrics for depth and extrinsics prediction can be found in Tables 4 and 5. In terms of performance, our method is better than both methods in both depth and camera parameter regression.

For visual comparison, we show qualitative results for state of the art methods compared to ours when predicting the structure for indoor images of the ADE20k dataset [45]. This dataset consists of images in the wild, annotated with semantics), but without ground truth depth. In Figure 4 we

| Method | Pitch error (°) | | Roll error (°) | |
|---|---|---|---|---|
| | avg. | med. | avg. | med. |
| DeepHorizon [36] | 8.68 | 5.50 | 2.98 | 1.89 |
| Hold-Geoffroy et al. [13] | 9.57 | 6.09 | 3.11 | 2.20 |
| UprightNet [37] | 7.59 | **4.94** | 2.30 | **1.53** |
| Ours (Full FoV) | 8.18 | 6.07 | 2.73 | 2.11 |
| Ours (Scannet FoV) | **7.45** | 5.39 | **2.18** | 1.65 |

Table 5. Generalization performance of camera parameter estimation on SUN360 dataset. We follow the evaluation method propsed in [37], which consists of sampling 6 different examples for each panoramic image using Scannet statistics.

show several examples, illustrating the advantages of our method and its shortcomings compared to other state of the art methods.

To visually compare state of the art methods against ours, we postprocess their output using the available semantic groundtruth, fitting a plane to the floor pixels, and compute the point cloud using the focal length corresponding to the dataset the methods are trained on. Our results correspond directly to the output of our system, without any further postprocessing.

The qualitative results show that, whereas state of the art methods produce plausible depth results, our method correctly estimates the point cloud in world reference frame. Qualitatively, our results match those computed by combining estimated depth and ground truth semantics, while giving a more consistent structure of the whole scene. For example, our method is able to recover better orthogonality between the floor, walls and other elements (such as the pool table in the first row) while, other methods are not able to recover orthogonality. This can be seen in either right-side views, where walls can be seen as not being orthogonal to the floor, or on top views, where walls do not project into a line.

**Failure cases.** Though we have shown our system to be more robust than state-of-the-art methods, it still fails if the testing time distribution is too diferent than that of training. In particular, it does not properly generalize to objects not present in the training dataset such as people. In Figure 5 we illustrate the behavior of our method when there is a person occupying a large portion of the image, together with the behavior of another state of the art method for the same image.

In the bottom row of the same figure, we display another failure case that is particular of our method. When it produces a bad estimate for some regions that account for large planar regions (such as floors having table like heights or the converse), this causes a shift on all the predicted coordinates. For this cases, we have qualitatively found depth prediction methods to recover a more plausible solution.


Figure 4. Pointcloud results on indoor images of ADE20k for 3 state of the art methods and ours. The methods are trained using NYU data (DORN [11] and Virtual Normal [40]) or Scannet (PlaneRCNN [26] and our two proposed systems). For each image, we display a top-down ortographic view and a right-side ortographic view. The depth region used to estimate the ground plane for the methods other than ours is highlighted in red in the images.

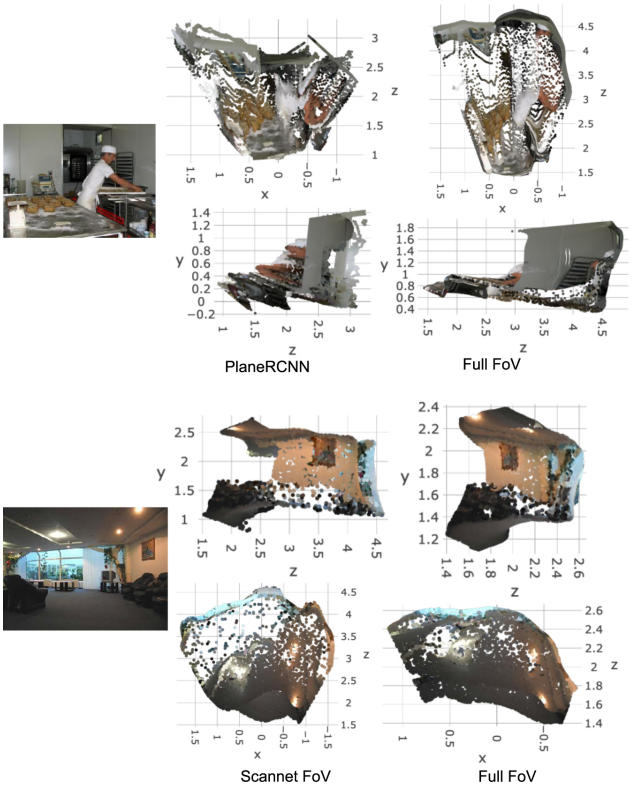


Figure 5. Failure cases on ADE20k. The result on top shows a failure case that is also present for other models predicting depth: these models are usually unable to generalize to objects not present in the training dataset, such as humans. The bottom row shows a failure case characteristic of our model.

### 4.5. Use in downstream tasks

Since the representation we obtain provides a point cloud in a canonical world frame, it can be more amenable to use for downstream tasks that require estimating the 3D structure of a scene.

To illustrate this with a simple example, in Figure 6 we show how the intrinsics and extrinsics predicted by our method can be of use for placing 3D objects in images in the wild. To obtain the results shown there, we only require a single user click on a valid ground point of the image. When placed there, the object has to be unoccluded for our method to produce plausible results, as our point-cloud representation does not allow to trivially test whether occlusions occur. Given this point, we place the object by rendering it according to the predicted camera intrinsics and extrinsics, and then we blend it with the original image.

## 5. Conclusion

In this paper, we have proposed a method to regress 3D coordinates from a single image in an upright and ground plane centered reference frame. To this end, we use a CNN

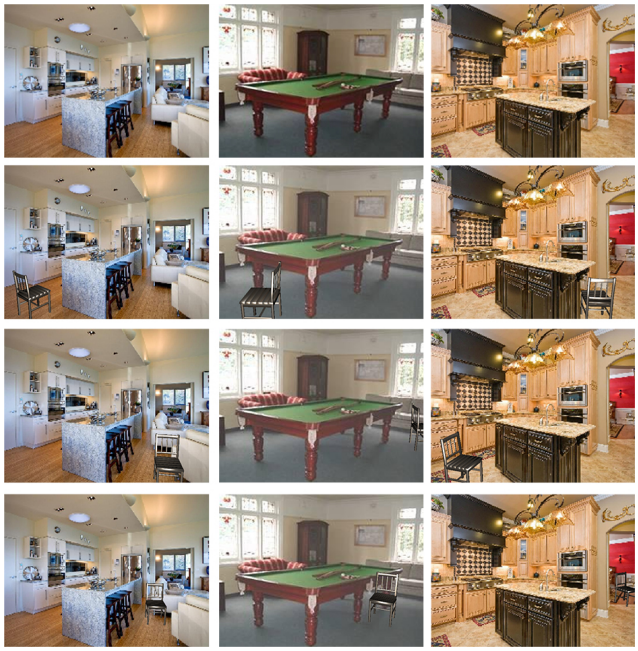

Figure 6. Metric 3D object placement. Given an image (top row, from the ADE20k dataset) and three image positions where a chair can be placed, we automatically place it on the ground plane position that projects to that point, using the predicted camera parameters of our method.

that predicts the 3D coordinates for all pixels, but does so in the subspace of scenes defined by a set of plausible camera models. We show that this empirically leads to better generalization to unseen datasets.

We argue that the reasons for these are that our representation of 3D structure is invariant to the camera pose, and can thus pick up more general structure during training, and non-local effects of our loss, that penalizes inconsistencies across large and distant regions of the image.

Furthermore, we have shown that the representation obtained by our model is amenable to downstream tasks such as 3D object placement, which can benefit from predicted metric 3D structure in a canonical reference frame.

**Acknowledgements** Funding for this research was partially supported by the Obra Social la Caixa Fellowship for Post-Graduate Studies to AR.

## References

[1] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2800–2810, June 2018. 2

[2] A. Azulay and Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations?, 2019. 1

[3] A. Buades, B. Coll, and J. . Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, June 2005. 2

[4] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 4

[5] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 730–738, USA, 2016. Curran Associates Inc. 2

[6] X. Chen, X. Chen, and Z.-J. Zha. Structure-aware residual pyramid network for monocular depth estimation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Aug 2019. 2

[7] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, Nov 2000. 2

[8] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. 2019. 1

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 4

[10] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014. 3

[11] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 1, 2, 3, 5, 7

[12] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. 2019. 2

[13] Y. Hold-Geoffroy, K. Sunkavalli, J. Eisenmann, M. Fisher, E. Gambaretto, S. Hadap, and J.-F. Lalonde. A perceptual measure for deep single image camera calibration. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 2, 5, 6

[14] J. Hu, Y. Zhang, and T. Okatani. Visualization of convolutional neural networks for monocular depth estimation, 2019. 2

[15] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother. Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017. 2

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[17] S. Kong and C. Fowlkes. Pixel-wise attentional gating for parsimonious pixel labeling, 2018. 5

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *2016 Fourth International Conference on 3D Vision (3DV)*, Oct 2016. 1

[19] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. *ACM Transactions on Graphics / SIGGRAPH*, 26(3), Aug. 2007. 2

[20] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2019. 2

[21] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation, 2019. 2

[22] J.-H. Lee and C.-S. Kim. Monocular depth estimation using relative depth maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[23] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. T. Freeman. Learning the depths of moving people by watching frozen people. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[24] Z. Li and N. Snavely. Learning intrinsic image decomposition from watching the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[25] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 2, 3

[26] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 6, 7

[27] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 6

[28] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *NeurIPS*, 2018. 3

[29] W.-C. Ma, H. Chu, B. Zhou, R. Urtasun, and A. Torralba. Single image intrinsic decomposition without a single intrinsic image. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[30] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5

[31] A. Ruderman, N. C. Rabinowitz, A. S. Morcos, and D. Zoran. Pooling is neither necessary nor sufficient for appropriate deformation stability in cnns, 2018. 1

[32] B. C. Russell and A. Torralba. Building a database of 3d scenes from user annotations. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2711–2718, June 2009. 2

[33] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009. 4

[34] T. van Dijk and G. C. H. E. de Croon. How do neural networks see depth in single images?, 2019. 2

[35] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. 2

[36] S. Workman, M. Zhai, and N. Jacobs. Horizon lines in the wild. *Procedings of the British Machine Vision Conference 2016*, 2016. 2, 5, 6

[37] W. Xian, Z. Li, M. Fisher, J. Eisenmann, E. Shechtman, and N. Snavely. Uprightnet: Geometry-aware camera orientation estimation from single images, 2019. 2, 5, 6

[38] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702, June 2012. 5

[39] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. *Lecture Notes in Computer Science*, page 432–448, 2018. 1

[40] W. Yin, Y. Liu, B. D. Schaefer, and Y. Yan. Enforcing geometric constraints of virtual normal for depth prediction. *ArXiv*, abs/1907.12209, 2019. 2, 7

[41] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese. Taskonomy: Disentangling task transfer learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 1

[42] R. Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 1

[43] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 1

[44] S. Zhao, H. Fu, M. Gong, and D. Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9788–9798, 2019. 2

[45] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

[46] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2

[47] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1