Camera Pose Matters: Improving Depth Prediction by Mitigating Pose Distribution Bias

Yuhan Zhao¹ Shu Kong² Charless Fowlkes¹
¹UC Irvine ²Carnegie Mellon University
{yunhaz5, fowlkes}@ics.uci.edu shuk@andrew.cmu.edu

Abstract

Monocular depth predictors are typically trained on large-scale training sets which are naturally biased w.r.t the distribution of camera poses. As a result, trained predictors fail to make reliable depth predictions for testing examples captured under uncommon camera poses. To address this issue, we propose two novel techniques that exploit the camera pose during training and prediction. First, we introduce a simple perspective-aware data augmentation that synthesizes new training examples with more diverse views by perturbing the existing ones in a geometrically consistent manner. Second, we propose a conditional model that exploits the per-image camera pose as prior knowledge by encoding it as a part of the input. We show that jointly applying the two methods improves depth prediction on images captured under uncommon and even never-before-seen camera poses. We show that our methods improve performance when applied to a range of different predictor architectures. Lastly, we show that explicitly encoding the camera pose distribution improves the generalization performance of a synthetically trained depth predictor when evaluated on real images.

1. Introduction

Monocular depth prediction aims to estimate 3D scene geometry from a 2D image. Despite being a largely underdetermined problem, convolutional neural network (CNN) based depth predictors trained on a sufficiently large-scale dataset are able to learn the joint statistics of scene geometry and appearance, and achieve impressive performance [10, 9, 12, 15, 26, 48].

However, an important overlooked fact is that the distribution of camera poses in training sets are naturally biased. As a result, a learned depth predictor is unable to make reliable predictions on images captured from uncommon camera poses, as shown in Fig. 1. Importantly, camera poses of testing examples may follow a different distribution from that in the training set. This will exacerbate prediction errors on images that are captured by cameras with uncommon poses relative to the training set.

Contributions. To this end, we propose two novel approaches that significantly improve depth prediction under diverse test-time camera poses. First, we introduce a simple perspective-aware data augmentation (PDA) that synthesizes new geometrically consistent training examples with more diverse viewpoints by perturbing the camera pose of existing samples. In contrast, common data augmentation (CDA) methods such as random-crop, though widely adopted in prior work [12, 24, 51, 21, 48, 5], produce training examples where the resulting image and target depth are...
inconsistent with the perspective geometry (Fig. 2). Second, we propose training a conditional depth predictor which utilizes the camera pose (e.g., acquired from IMU or other pose predictors) as a prior (CPP) when estimating depth. We propose an effective approach to encode CPP as an additional channel alongside the RGB input. We find incorporating pose using CPP yields more accurate depth predictors that generalize much better under diverse test-time camera poses.

Through extensive experiments, we show that these techniques significantly improve depth prediction on images captured from uncommon and even never-before-seen camera poses. Both techniques are general and broadly applicable to any network architecture. We show that incorporating them in recent state-of-the-art architectures improves their performance further. Lastly, we show that explicitly handling the biased camera pose distribution can improve the performance of a synthetically trained depth predictor when tested on real images. This highlights the importance that camera pose distribution plays in domain adaptation for 3D geometric prediction tasks.

2. Related Work

Monocular Depth Prediction and scene layout estimation have been greatly advanced since the seminal works [20, 39]. State-of-the-art approaches train increasingly sophisticated CNN-based predictors [33, 10, 26], utilize better training losses [12, 40, 48] and train on larger-scale training datasets [31, 27, 50, 53].

Surprisingly little attention has been paid to camera pose bias and out-of-distribution generalization. The recent study of [8] concluded that the learned depth predictors have a strong implicit bias causing them to perform poorly on test images when captured from differing camera poses. Our work systematically analyzes this pose bias/robustness in detail and offers technical contributions that improve generalization on test images captured under diverse camera poses.

Camera Pose Estimation plays an essential role in many traditional 3D geometry vision problems such as SLAM [2, 43, 22] and 3D reconstruction [41, 1]. Predicting the relative camera pose between a pair of frames has been widely exploited to perform self-supervised learning of monocular depth prediction [14, 55, 44]. Absolute camera pose is often represented implicitly in predictions of room layout [28, 57, 46, 45]. Closer to our work is [3] which estimates the absolute camera pose (height, pitch, roll) in order to regularize depth predictions in world coordinates.

We explore the benefit of providing the camera pose as an additional input to depth prediction. In practical applications, such pose information may come from other sensors (e.g., pitch from IMU) or prior knowledge (e.g., camera mounted at a known height and pitch on an autonomous vehicle). The work of [18, 11] encode camera intrinsics (e.g., focal length) as a part of the input for depth prediction, with a goal to learn a universal depth predictor that generalizes to images captured by different cameras. Similarly, we propose to encode camera extrinsic parameters (e.g., camera height) which we exploit for training better depth predictors that perform well on testing images captured with diverse camera extrinsic parameters.

Distribution Bias. Challenges of class imbalance and bias have been a widely discussed topic in the literature on classification and recognition [16, 56, 34]. Distribution bias naturally exists in a training set, implying that some examples are underrepresented or few in number w.r.t some attributes (e.g., class labels). As a result, a trained model is unlikely to perform well on the underrepresented testing inputs. Even worse, testing examples may come from out-of-distribution data [19, 32, 29], meaning that the training set does not have similar examples. For example, in monocular depth prediction, training examples might be collected with cameras held vertical with minimal pitch and roll variations, but the testing scenario (e.g., AR/VR headset) might have potentially large variations in pitch and roll.

3. Perspective-aware Data Augmentation

Due to the biased distribution of camera poses in the training set, some examples are underrepresented w.r.t camera poses. To resolve this issue, we would like to augment the training data to cover these underrepresented poses.

Resampling (RS) the training data with replacement to enlarge the prevalence of uncommon poses in the train-set is perhaps the simplest approach. However, this does not increase the diversity of the train-set because it cannot generate new training examples. Perhaps even worse, in practice, training on repeated examples from uncommon camera
poses forces the model to weight them more (cf. overfitting), while sacrificing performance on other examples captured under common camera poses.

Conventional data augmentation (CDA), specifically random-cropping of the training examples, is widely used to enrich the training data in prior work \[10, 25, 12, 24, 48\].

While CDA seems to increase the diversity of the training set, its generated data could adversely affect depth prediction. Cropping a subregion from an original example naively copies depth values without considering the viewpoint-dependent nature of depth maps that depend on both the camera pose and scene geometry. Such a cropped image is equivalent to capturing another image of the same scene using a camera with an off-center principle point (Fig. 2) and can make it difficult to train depth predictors \[11\]. In our work, we find CDA helps only if we crop sufficiently large regions, which presumably reduces this effect.

Perspective-aware data augmentation (PDA) is our solution that augments training examples consisting of RGB images and depth maps. Given a training example, PDA first perturbs the camera pose, re-projects all pixels to a new image, and recomputes the depth values for the new image using the new camera pose and the original depth map. Despite its simplicity, to the best of our knowledge, no prior work exploits this idea for data augmentation during the training of a monocular depth predictor.

Given the training image $I$, depth $D$ and camera intrinsic matrix $K$, we would like to synthesize a new image $I_s$ and depth $D_s$ corresponding to a perturbed viewing direction. Let $T_{rel}$ be the relative rotation between the two poses. Then for any point $q = [u \ v]$ on $I$, we compute the corresponding point $q'$ on the new image $I_s$ via the homography:

$$q_h \sim KT_{rel}zK^{-1}q_h,$$

where $z$ is the corresponding depth value of the point $q$ from the original depth map $D$; $q_h$ and $q_h'$ are the homogeneous coordinates of points $q$ and $q'$, respectively.

Similarly, we compute the depth value for each pixel in the new depth map $D_s(q')$:

$$z' = v_{proj}^T T_{rel}zK^{-1}q_h,$$

where $v_{proj} = [0, 0, 1]^T$ is a unit vector pointed along the $z$-axis of the camera.

While the above demonstrates the computation of per-pixel depth values, in practice, we can compute the whole depth map $D_s$ efficiently by using backward-warping and standard bilinear interpolation. For larger rotations, we note that the synthesized views will have void regions on the boundary, as shown Fig. 2. This does not pose a problem for depth since we simply exclude those regions from the loss during training. However, we find it helpful to pad the RGB void regions using values from the RGB images (using the "reflection" mode during warping).

We also considered applying PDA with camera translation. However, this requires forward-warping and introduces "holes" in the synthesized result \[36\] at disocclusion boundaries. Handling disocclusions is still an open problem and out of our scope \[4, 37, 35\]. therefore, we choose to only augment camera rotations to avoid disocclusion artifacts and allow for efficient computation.

4. Depth Prediction with Camera Pose Prior

Depth maps are view-dependent representations that depend on both the scene geometry and camera pose. The camera pose inherently provides prior knowledge about the expected scene depth map. For example, knowing a camera is pointing down to the ground at one-meter height, we should expect a depth map of one-meter height roughly everywhere. Therefore, we are motivated to train a conditional depth predictor on camera pose as prior (CPP).

Camera pose is a six-degree-of-freedom (6DoF) vector that describes translation and rotation in 3D world coordinates. In typical terrestrial man-made scenes, we consider a global coordinate with one axis pointing upwards (as specified by gravitational acceleration) and fix the origin along that axis to be 0 at the ground plane. Since there is no unique origin along the two remaining axes, we assume that our
camera pose prior should be uniform over translations parallel to the ground plane. Similarly, there is no unique way to specify the orientation of the axes parallel to the ground plane so our prior should necessarily be uniform over rotations of the camera about the up-axis. This leaves three degrees-of-freedom (3DoF): the height of the camera above the ground plane \(h\), the pitch (angle relative to the up-axis) of the camera \(\theta\), and any roll \(\omega\) of the camera around its optical axis (Fig. 3).

A naive encoding approach. We now consider using this 3DoF camera pose as a part of the input (along with RGB) to depth predictors. To incorporate this as input into a CNN, inspired by the literature \([11, 49]\), we convert 3DoF camera poses into 2D maps, which are concatenated with the RGB image as a whole input to learn the depth predictor. Naively, we can create three more channels of resolution \(H \times W\), which copy values of roll \(\omega\), pitch \(\theta\) and height \(h\), i.e., \(M_h[; ; ] = \omega\), \(M_\theta[; ; ] = \theta\) and \(M_h[; ; ] = h\), respectively. However, the effect of pose on the depth distribution depends strongly on the position in the image relative to the camera center so translation-equivariant convolutions cannot fully exploit this encoding (except by relying on boundary artifacts). This is supported by an experimental comparison showing this naive encoding is inferior to our proposed CPP encoding method, elaborated in the following.

CPP encoding encodes the pose locally by assuming that the camera is placed in an empty indoor scene with an infinite floor and ceiling. Intuitively, it encodes the camera pose by intersecting rays from the camera center with pre-defined ground/ceiling planes and recording the depth, see Fig. 3.

Let \(p = [x \ y \ z]\) be a 3D point in the global coordinate, \(q = [u \ v]\) be a 2D point on the image plane whose homogeneous form is \(q_h\), \(n \in \mathbb{R}^3\) denotes the normal vector of ground planes, \(C\) be the distance between two planes in the up direction.

The projection from 3D coordinates to 2D image plane is:

\[
\lambda q_h = KR^{-1}(p - t), \tag{3}
\]

where \(\lambda\) is a scale factor; \(K\) is the intrinsic matrix; \(R \in SO(3)\) and \(t \in \mathbb{R}^3\) are rotation and translation matrices known from the camera pose, respectively. We compute the 3D point \(p\) where the ray shooting from the camera center through point \(q\) eventually intersects with planes or the horizon. However, \(\lambda\) is an unknown scalar and we need extra constraints to compute it. Taking ground plane as an example, we know the collections of 3D points intersecting with ground plane is \(\{p : n^T p = 0\}\). With this new constraint, we rearrange Eqn. 3 and multiply \(n^T\) on both sides of the equation:

\[
n^T p = \lambda n^T R K^{-1} q_h + n^T t \Rightarrow \lambda = \frac{-n^T t}{n^T R K^{-1} q_h} \tag{4}
\]

Now, we plug the computed \(\lambda\) back to Eqn. 3, then the 3D point \(p_g\) that intersects the ground plane for \(q\) is:

\[
p_g = \frac{-n^T t}{n^T R K^{-1} q_h} R K^{-1} q_h + t \tag{5}
\]

Similarly, with the constraint \(\{p : n^T p = C\}\), the 3D point \(p_c\) that intersects with the ceiling plane for \(q\) is:

\[
p_c = \frac{C - n^T t}{n^T R K^{-1} q_h} R K^{-1} q_h + t \tag{6}
\]

Once we have the 3D intersection point \(p\) for each point \(q\) on the image plane, we compute the projection on the camera \(z\) direction (i.e., the depth direction):

\[
z(p) = v_{proj}^T R^{-1} (p - t), \tag{7}
\]

where \(v_{proj}\) is the projection vector that computes the projection of a 3D point along the camera depth direction. Finally, for each point \([u, v]\) in the encoding map \(M \in \mathbb{R}^{H \times W}\), we compute both \(p_g\) and \(p_c\) and take the maximum (positive) value as the pseudo depth value (Fig. 3):

\[
M[u, v] = \max\{z(p_g), z(p_c)\} \tag{8}
\]

The encoding map \(M\) can have infinite values, e.g., when the ray shooting from the camera is parallel to the ground plane. To map values into a finite range, we apply the inverse tangent operator \(\tan^{-1}(\cdot)\) to obtain our final encoding \(M_{CPP} = \tan^{-1}(M)\) which takes on values in the range \([\tan^{-1}(\min\{h, C - h\}), \frac{\pi}{2}]\). We visualize some CPP maps in Fig. 3-right.

To train a conditional depth prediction, we simply concatenate the CPP encoded map with the corresponding RGB as a four-channel input. This implies that our CPP encoding approach applies to any network architectures with a simple modification on the first convolution layer.

5. Experiments

We validate our methods through extensive experiments. Specifically, we aim to answer the following questions:

- Do our methods improve depth prediction on a test-set that has a different camera pose distribution from the train-set?
- Does resampling improve depth prediction on a test-set that has a different camera pose distribution?
- Do our methods improve depth estimation on testing images captured with out-of-distribution camera poses?
- Does applying our methods to other state-of-the-art depth predictors improve their performance?
- Do our methods improve the performance of a depth predictor when training and testing on different datasets?

Answers: yes, no, yes, yes, yes.
Datasets. In our work, we use two publicly available large-scale indoor-scene datasets, InteriorNet [30] and ScanNet [7] which come with ground-truth depth and camera pose. Compared to other datasets such as [13, 6, 42], these were selected because they illustrate a much wider variety of camera poses. InteriorNet [30] consists of photorealistic synthetic video sequences with randomly generated camera trajectories with a wide variety of pitch and height but minimal roll. ScanNet [7] contains real-world RGBD videos of millions of views from 1,513 indoor scenes collected with substantial variation in pitch, roll, and height. For each dataset, we create train/test sets by randomly splitting scenes into two disjoint sets with 85%/15% examples, respectively. We use a stratified sampling approach to avoid picking adjacent video sequences in the same set.

Sampling. To explore how the biased camera pose distribution affects depth learning and prediction, we sample three subsets from the test/train set (Fig. 4) which each have 10k/1k images but with different distributions of camera poses.

- **Natural** selects samples at random to reflect the natural distribution of the dataset.
- **Uniform** selects samples that simulate a uniform distribution over poses with priority: pitch→roll→height. Concretely, we quantize the range of camera pitch/roll/height into equal-size bins, and sample an approximately equal number of examples in each bin.
- **Restricted** samples images within a narrow range: camera pitch $\theta \in [85^\circ, 95^\circ]$ and height $h \in [1.45, 1.55]$ (meters). While InteriorNet does not have roll variations, ScanNet does: roll $\omega \in [-5^\circ, 5^\circ]$. We create Restricted to particularly study how depth predictor performs on testing images captured with out-of-distribution camera poses.

Evaluation Metrics. There are several evaluation metrics widely used in the literature [10, 12, 48], including absolute relative difference ($\text{Abs}^*$), squared relative difference ($\text{Sq}^*$), root mean squared log error (RMS-log), and accuracy with a relative error threshold of $\delta^k < 1.25^k$, $i = 1, 2$.

Implementation. We use a standard U-Net structured model to perform most of our experimental analysis [53, 52, 54]. We also demonstrate that our techniques apply to other depth predictor architectures (Section 5.3). Unless specified, all models are trained with the L1 loss and applied random left-right flip augmentation during training.

While we initially learn the conditional depth predictor using the true camera pose, we also tested encoding a predicted camera pose in Section 5.4. For predicting camera pose, we train a pose predictor with the ResNet18 architecture [17] to directly regress to camera pitch, roll, and height. We resize all images and depth maps to $240 \times 320$, and adjust camera intrinsic parameters (for CPP encoding) accordingly. We use PyTorch [38] to train all the models for 200 epochs on a single Titan Xp GPU. We use the Adam optimizer [23] with a learning rate 1e-3, and coefficients 0.5

<table>
<thead>
<tr>
<th>Models</th>
<th>Natural/Test-Set</th>
<th>Uniform/Test-Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; better</td>
<td>&lt; better</td>
<td>δ1 / δ2</td>
</tr>
<tr>
<td>Vanilla</td>
<td>154 / 148 / 229</td>
<td>803 / 945</td>
</tr>
<tr>
<td>+ RS</td>
<td>192 / 203 / 267</td>
<td>726 / 918</td>
</tr>
<tr>
<td>+ CDA</td>
<td>142 / 137 / 222</td>
<td>825 / 950</td>
</tr>
<tr>
<td>+ PDA</td>
<td>138 / 123 / 207</td>
<td>834 / 957</td>
</tr>
<tr>
<td>+ Naive + CPP</td>
<td>132 / 135 / 219</td>
<td>835 / 944</td>
</tr>
<tr>
<td>+ Both</td>
<td>895 / 101 / 180</td>
<td>898 / 966</td>
</tr>
<tr>
<td>ScanNet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vanilla</td>
<td>125 / 068 / 186</td>
<td>837 / 962</td>
</tr>
<tr>
<td>+ RS</td>
<td>216 / 158 / 279</td>
<td>619 / 809</td>
</tr>
<tr>
<td>+ CDA</td>
<td>116 / 062 / 179</td>
<td>853 / 964</td>
</tr>
<tr>
<td>+ PDA</td>
<td>115 / 059 / 171</td>
<td>860 / 970</td>
</tr>
<tr>
<td>+ Naive + CPP</td>
<td>120 / 069 / 184</td>
<td>846 / 959</td>
</tr>
<tr>
<td>+ Both</td>
<td>102 / 052 / 169</td>
<td>882 / 973</td>
</tr>
</tbody>
</table>

Figure 5: Breakdown analysis of depth prediction w.r.t. pitch. The background shade denotes the camera pose distribution w.r.t. pitch. Clearly, both PDA and CPP improve depth prediction in under-represented camera poses. Surprisingly, CPP remarkably boosts depth prediction, while applying both CPP and PDA achieves the best performance “everywhere”.

Table 1: Within & cross-distribution evaluation. In each dataset, we train depth predictors on their Natural train-sets and evaluate on both Natural and Uniform test-sets. We apply different methods to a Vanilla model. All models use the same network architecture. Vanilla performs poorly in cross-distribution evaluation (cf. Natural-test vs. Uniform-test), demonstrating that the biased camera pose distribution affects the training of depth predictors. As expected, RS hurts depth prediction compared to Vanilla. In contrast, CPP and PDA show better performance; jointly applying them performs the best (i.e., “Both”). Finally, comparing alternative methods to our CPP (vs. Native) and PDA (vs. RS and CDA) shows the merits of our methods.
and 0.999 for computing the running averages of gradient and its square.

For CPP encoding, we set the ceiling height $C = 3$ meters. We have studied different settings of $C$, but find little difference (details in the supplement). For PDA, we randomly perturb camera pose within $[-0.1, 0.1]$ radius angle jointly w.r.t pitch, roll, and yaw; we also ablate the perturbation scale in Section 5.5.

### 5.1. Within & Cross-Distribution Evaluation

We start with initial experiments designed to reveal how bias in camera pose statistics affects depth predictor performance (Fig. 4). The experiments also explore the design choices of our methods and validate their effectiveness. Specifically, we train depth predictors on the Natural train-sets, and test them on both Natural and Uniform test-sets. We apply a sequence of different modifications to a “Vanilla” baseline model, i.e., a UNet-based predictor. Table 1 lists detailed comparisons, and Fig. 6 shows qualitative comparisons.

The Vanilla model degrades notably when evaluated on a test-set that has a different pose distribution, i.e., from Natural to Uniform, showing the clear influence from the biased distribution of camera poses. In terms of data augmentation, simply resampling the training data (RS) hurts performance (cf. Vanilla and “+RS”). Moreover, our PDA outperforms CDA, demonstrating the importance of synthesizing geometric-aware training examples using the corresponding camera and the scene geometry in depth prediction. As for camera pose encoding, our CPP encoding method clearly outperforms the Naive method. Importantly, jointly applying CPP and PDA performs the best on both within-distribution (Natural) and cross-distribution (Uniform) test-sets. Finally, we breakdown the performance in Fig. 5 to analyze when (i.e., w.r.t pitch angle) PDA and CPP improve depth prediction. Generally, both of them help reduce prediction errors on testing images captured with underrepresented camera poses, while CPP yields the largest benefits. Applying both achieves the best performance “everywhere”.

### Table 2: Out-of-distribution evaluation

We train depth predictors on the Restricted train-sets and test on both Restricted and Natural test-sets of each datasets. Vanilla model performs poorly on Natural test-sets, clearly showing the challenge of depth prediction on images captured under novel/never-before-seen camera poses. CPP slightly improves performance, but PDA helps more. CDA also improves performance presumably because it synthesizes more training examples, but underperforms PDA. As expected, jointly applying both CPP and PDA achieves the best performance on both Restricted and Natural test-sets.

### 5.2. Out-of-Distribution Evaluation

We now study how our methods help when training depth predictors on a train-set which have a rather restricted range of camera poses. Specifically, we train models on the Restricted train-sets of InteriorNet and ScanNet, respectively, and test the models on their Natural test-sets. This setup is synthetic and unlikely to be a real-world scenario, but it allows for exclusive analysis of depth predictors when tested on images captured with out-of-distribution or never-before-seen camera poses.

Table 2 lists detailed comparisons. Vanilla model performs poorly when tested on Natural test-set. CPP slightly improves prediction, but PDA helps a lot. CDA (i.e., random-crop as augmentation) also helps, presumably owing to more diverse training examples, but underperforms our PDA. This further confirms the importance of generating geometric-aware new training examples for the given scene and camera in depth prediction. Under expectation, applying both PDA and CPP performs the best.
Table 3: Applicability to other predictor architectures. In each dataset, we train state-of-the-art depth predictors (DORN [12] and VNL [48]) by optionally applying our CPP and PDA approaches. All models are trained/tested on the Natural train/test-sets per dataset. Clearly, both CPP and PDA boost their performance.

<table>
<thead>
<tr>
<th>Models</th>
<th>InteriorNet</th>
<th>ScanNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPP+DORN</td>
<td>↓ better</td>
<td>↑ better</td>
</tr>
<tr>
<td>Abs/Sq/RMS-log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Natural-T est-Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DORN</td>
<td>129 / 126 / 218</td>
<td>.844 / .957</td>
</tr>
<tr>
<td>+ CPP</td>
<td>.099 / .105 / .197</td>
<td>.899 / .962</td>
</tr>
<tr>
<td>+ PDA</td>
<td>.115 / .112 / .207</td>
<td>.867 / .958</td>
</tr>
<tr>
<td>+ Both</td>
<td>.095 / .085 / .124</td>
<td>.912 / .971</td>
</tr>
<tr>
<td>VNL</td>
<td>113 / 140 / 235</td>
<td>.853 / .954</td>
</tr>
<tr>
<td>+ CPP</td>
<td>.101 / .120 / .207</td>
<td>.893 / .960</td>
</tr>
<tr>
<td>+ Both</td>
<td>.086 / .085 / .131</td>
<td>.909 / .970</td>
</tr>
</tbody>
</table>

5.3. Applicability to Other Predictor Networks

CPP and PDA are general and applicable to different model architectures. We study applying them to training two well-established depth predictors: DORN [12] and VNL [48]. Compared to the Vanilla model, both predictors adopt different network architectures [47, 17] and different loss functions. They convert continuous depth values to discrete bins and model depth prediction as a classification problem. Moreover, VNL incorporates a virtual surface normal loss which provides a geometric-attribute regularization during training. We implement DORN and VNL using publicly-available third-party code. We train and test all models on the Natural train/test-sets, in InteriorNet and ScanNet, respectively.

Table 3 lists detailed results, which are comparable to the Natural-Test-Set column in Table 1. Consistent with previous experiments, both CPP and PDA improve the performance further based on DORN and VNL. As our CPP and PDA improve other depth predictors as a general approach, we suggest using them in future research of monocular depth estimation.

5.4. Synthetic-to-Real Generalization

Previous experiments have validated that addressing the biased camera pose distribution helps train a depth predictor that works better on another test-time distribution, or more generally another domain. Here we evaluate performance in the presence of substantial synthetic-to-real domain shift which includes both low-level shifts (e.g., local material appearance) and high-level shifts (novel objects, layouts and camera poses) [53]. Specifically, we synthetically train a depth predictor (on InteriorNet Natural train-set) and test it on real images (ScanNet Natural and Uniform test-sets). We also consider a more practical scenario that one does not have access to the true camera pose but instead must rely on the predicted poses. To this end, we train a camera pose predictor on ScanNet Natural test-set to predict camera pitch, height and roll for a given RGB image. Then, we perform CPP encoding with the predictive pose, i.e., CPP

Table 4: Mitigating distribution bias of camera poses improves synthetic-to-real domain adaptation. We train depth predictors synthetically on InteriorNet (Natural train-set) and test them on real-world images from ScanNet Natural and Uniform test-sets. This is a typical setup for synthetic-to-real domain adaptation in the context of depth prediction. Interestingly, we find that CDA hurts the performance, presumably because the generated training examples by CDA do not obey the relations among camera model, scene geometry and camera pose, and hence do not necessarily help training a generalizable depth predictor. In contrast, our PDA helps synthetic-to-real generalization and applying CPP improves further. Importantly, applying CPP with predictive poses (CPP

<table>
<thead>
<tr>
<th>Methods</th>
<th>Natural-Test-Set</th>
<th>Uniform-Test-Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs/Sq/RMS-log</td>
<td>better</td>
<td>better</td>
</tr>
<tr>
<td>Natural-T est-Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vanilla</td>
<td>242 / .315</td>
<td>530 / 812</td>
</tr>
<tr>
<td>CDA</td>
<td>246 / .321</td>
<td>568 / 843</td>
</tr>
<tr>
<td>PDA</td>
<td>239 / .311</td>
<td>575 / .857</td>
</tr>
<tr>
<td>PDA+CPP+pred</td>
<td>219 / .305</td>
<td>556 / .866</td>
</tr>
<tr>
<td>PDA+CPP</td>
<td>208 / .282</td>
<td>677 / .893</td>
</tr>
</tbody>
</table>

Table 4 lists detailed setups and results; we summarize the salient conclusions. Vanilla achieves worse performance compared to Table 1, showing a clear domain gap between the two datasets. Applying CDA hurts the performance, presumably because the generated training data by CDA disobey the projective geometry relations between scene geometry and camera model and pose (cf. Section 3). In contrast, our PDA helps, but CPP improves even more notably. Applying CPP with the predictive pose (CPP

5.5. Further Discussion and Ablation Study

“Blind” depth prediction without RGB. To characterize the prior knowledge carried by camera poses in terms of depth prediction, we train a “blind predictor” on the InteriorNet Natural train-set, taking as input only the CPP encoded maps of camera poses without RGB images. For comparison, we compute an average depth map over the whole Natural train-set. We qualitatively compare results on two testing examples in Fig. 8 (quantitative results in the supplement). Visually, encoding camera pose alone using CPP reliably provides depth estimate on floor regions. This intuitively explains that using camera pose does serve as strong prior knowledge of scene depth.

Resilience to noisy camera pose. Camera pose estimates (e.g., from IMU sensors) are potentially noisy, and the predicted camera poses are undoubtedly erroneous (cf. the previous experiment using predicted poses in CPP). We study how resilient CPP is to test-time errors in the camera...
We plot depth prediction error (Abs$^2$) w.r.t different levels of noise in camera height and pitch. We apply CPP to train/test a depth predictor (based on Vanilla) on InteriorNet Natural train/test-set. For a given noise level $\delta$, the trained model makes depth predictions using a CPP map computed with a perturbed camera pose, e.g., the pitch is sampled from $\theta_{gt} + U[-\delta, \delta]$. The black dot at the origin stands for the (best) performance using the true camera pose (i.e., no noises are presented in pitch and height). The dashed lines represent the average performance levels for the Vanilla and CPP with predictive poses. Right: We visualize depth prediction by CPP model when encoding perturbed camera pitch angles $\theta_{gt} \pm 18^\circ$ and heights $h_{gt} \pm 0.1$(m). CPP model predicts shallower depth when both pitch and camera height decrease (i.e., camera is tilted down or translated closer to the floor). This qualitatively confirms that the camera pose prior induces a meaningful shift in the estimator. The corresponding RGB image and ground-truth depth appear in Fig. 8.

Camera pose alone provides a strong depth prior even for “blind” depth prediction. Specifically, over the InteriorNet Natural train-set, we train a depth predictor solely on the CPP encoded maps $M$ without RGB as input. For visual comparison, we compute the average depth map (shown left). We visualize depth predictions on two random examples. All the depth maps are visualized with the same colormap range. Perhaps not surprisingly, $M$ presents nearly the true depth in floor areas, suggesting that camera pose alone does provide strong prior depth information for these scenes.

6. Conclusion

While large-scale datasets allow for end-to-end training of monocular depth predictors, we find the training sets naturally biased w.r.t distribution of camera poses. As a result, trained predictors fail to make reliable depth predictions for testing examples captured under uncommon camera poses. We mitigate this bias with two novel methods, perspective-aware data augmentation (PDA) and camera pose prior encoding (CPP). We show that applying both our methods improves depth prediction on images captured under uncommon or never-before-seen camera poses. Moreover, our methods are general and readily applicable to other depth predictors, which can perform better when trained with PDA and CPP, suggesting using them in the future research of monocular depth estimation.

Acknowledgements We thank anonymous reviewers for their valuable suggestions. This research was supported by NSF grants IIS-1813785, IIS-1618806, a research gift from Qualcomm, and a hardware donation from NVIDIA.
References


[18] Lei He, Guanghui Wang, and Zhanqi Hu. Learning depth from single images with deep neural network embedding focal length. IEEE Transactions on Image Processing, 2018. 2


[31] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep


