SDC-Depth: Semantic Divide-and-Conquer Network for Monocular Depth Estimation

Lijun Wang¹, Jianming Zhang², Oliver Wang², Zhe Lin², and Huchuan Lu¹,³ *
¹Dalian University of Technology, ²Adobe Research, ³Peng Cheng Lab
{ljwt1chuan}@dlut.edu.cn, {jianmzha,owang,zlin}@adobe.com

Abstract

Monocular depth estimation is an ill-posed problem, and as such critically relies on scene priors and semantics. Due to its complexity, we propose a deep neural network model based on a semantic divide-and-conquer approach. Our model decomposes a scene into semantic segments, such as object instances and background stuff classes, and then predicts a scale and shift invariant depth map for each semantic segment in a canonical space. Semantic segments of the same category share the same depth decoder, so the global depth prediction task is decomposed into a series of category-specific ones, which are simpler to learn and easier to generalize to new scene types. Finally, our model stitches each local depth segment by predicting its scale and shift based on the global context of the image. The model is trained end-to-end using a multi-task loss for panoptic segmentation and depth prediction, and is therefore able to leverage large-scale panoptic segmentation datasets to boost its semantic understanding. We validate the effectiveness of our approach and show state-of-the-art performance on three benchmark datasets.

1. Introduction

Depth estimation is an important component of 3D perception. Compared to reconstruction techniques based on active sensors or multi-view geometry, monocular depth estimation is significantly more ill-posed, and is therefore critically reliant on learning strong scene priors and semantics. Recent works studying this problem [4, 14, 39] have achieved significant progresses using deep convolutional neural networks (CNNs) supervised by depth data, showing that they are able to capture complex high-level scene semantics. In addition, some works [39, 28] further feed semantic segmentation labels to their models to boost depth estimation accuracy in some specific domains. However, monocular depth estimation in the wild remains challenging due to the diversity of real world scenes.

We propose a Semantic Divide-and-Conquer Network (SDC-Depth Net) for monocular depth estimation. We decompose a natural image into a number of semantic segments, and then predict, for each segment, a normalized depth map in the range [0, 1]. We refer to this normalized depth map as the canonical depth for a given segment. This decomposition simplifies the depth prediction problem, as semantic categories have much consistent depth structures when viewed in isolation, and training category-specific depth decoders makes it easier to learn these priors. For example, the sky region is always infinitely far away and depth in the ground region often varies smoothly along the vertical direction. For object categories like people, instance-level depth maps also have a high degree of similarity to each

*Corresponding author.
other. Once we predict the canonical depth for each semantic segment, we assemble the results together based on a global context derived from the input image. Specifically, our model predicts the scale and shift for each segment’s depth using the global context. The model is trained fully end-to-end using a multi-task loss for segmentation and depth estimation, for which we can use separate datasets to increase the diversity of our supervision. Figure 1 demonstrates sampled results of our approach.

Our approach is inspired by the classical divide-and-conquer algorithm [17], but relies on semantic and instance segmentation to divide the problem. Luckily, diverse panoptic segmentation annotations are relatively easy to collect compared to depth supervision, and we can leverage existing large-scale panoptic segmentation datasets such as COCO Panoptic Segmentation dataset [23] to complement the limited depth supervision. As an auxiliary task, semantic and instance segmentation not only helps split objects and categories for local depth prediction, but also necessitates the model’s understanding of shape and contour regions. Thus, it can improve the model’s generalization ability, and is also useful in cases where only sparse or low res depth annotations (e.g., depth order of point pairs as in [2, 37], or Kinect data as in [32]) are available.

We show experiments on three benchmark datasets, demonstrating that our method can significantly improve the performance of depth estimation. Particularly, on the challenging “Depth in the Wild” (DIW) dataset [2], we achieve a new state-of-the-art error rate of 11.21% improving upon the previous best result of 13.02% [39].

In summary, we present a novel framework for monocular depth estimation based on a semantic divide-and-conquer strategy. We present an implementation of this high-level framework through SDC-Depth Net, a carefully designed end-to-end trainable architecture. Experimental validation of our approach shows consistent improvements over the state-of-the-art on three benchmark datasets.

2. Related Work

Single Image Depth Prediction There has been a long history of methods that have attempted to predict depth from a single image [11, 31, 24, 26]. Recently monocular depth estimation has gained popularity due to the ability of CNNs to learn strong priors from images corresponding to geometric layout. Among others, Laina et al. [14] propose a fully convolutional architecture with up-projection blocks to handle high-dimensional depth regression. In [19], a two-stream convolutional network is proposed, which simultaneously predicts depth and depth gradients to preserve more depth details. Besides using deep networks alone, recent works have shown that the combination of deep networks and shallow models [18, 25, 36, 40, 30] can also deliver superior depth estimation performance. Meanwhile, different forms of supervision and learning techniques have also been explored in recent works to improve the generalization ability of depth estimation models, including self-supervised learning with photometric losses from stereo images [6, 8] or multiple views [43, 34, 7], transfer learning using synthetic images [42, 42, 1], and those using sparse [2, 37] or dense [21, 35, 33, 20] relative depth as supervisions.

Augmenting Depth with Semantic Segmentation Some recent works [41, 39, 28] propose to improve monocular depth estimation with semantic segmentation annotations. For instance, Liu et al. [24] propose to guide single image depth estimation with semantic labels using Markov random fields. Xu et al. [39] develop a multi-modal distillation module, which can leverage intermediate depth and segmentation predictions to refine the final output. In [12] a synergy network together with an attention-driven loss is proposed to better propagate semantic information to depth prediction. In comparison, [41] presents a task-recursive learning strategy, which can refine both depth and segmentation predictions through task-level interactions. Another related work to ours is [28], where depth estimation is learned in an unsupervised manner by integrating both semantic segmentation and instance edges as input.

Although improvement has been achieved, these approaches have their own drawbacks. For one, existing works estimate depth for different categories with a single model. We argue that the depth values of different categories may exhibit different properties and subject to a variety of data distributions. Covering all these variations with one model may be sub-optimal. In addition, besides semantic categories, object instance information may also play a crucial role in depth estimation. However, compared to semantic segmentation, instance detection/segmentation is less explored in monocular depth estimation. Compared to these existing works, our proposed method performs depth estimation for each segment independently by investigating their semantic and instance information. We disentangle canonical depth estimation and depth scale inference leading to more accurate depth prediction results.

3. Semantic Divide-and-Conquer Network for Monocular Depth Estimation

We present SDC-Depth Net, an end-to-end trainable depth prediction network based on the aforementioned Semantic Divide-and-Conquer strategy. Our SDC-Depth Net consists of four parts: a backbone network, a segmentation module, a depth prediction module, and a depth aggregation module. Figure 2 overviews the proposed method.

The backbone network, shared by the segmentation and the depth prediction modules, extracts features of an input image. The segmentation module performs semantic and instance segmentation to divide the image into semantic
segments. For each semantic segment, the depth estimation module infers a category-specific depth map in a canonical space, as well as scale and shift parameters based on the global context. The aggregation module then stitches and aggregates the per-segment depth maps to generate a globally consistent depth map.

In our experiments, we adopt a feature pyramid network (FPN) \cite{lin2017feature} with ResNet-50 \cite{he2016deep} as the backbone network. We use a fully convolutional network (FCN) \cite{long2015fully,ronneberger2015u} and a Mask R-CNN model \cite{he2017mask} for semantic and instance segmentation respectively. The FCN network performs semantic segmentation for \( C \) categories, where the first \( K \) categories are object classes (e.g., person, car) and the rest belong to stuff classes (e.g., road, grass). The Mask R-CNN network detects object instance masks for the \( K \) object classes. We now discuss each part in detail.

### 3.1. Per-Segment Depth Estimation

Given a semantic segment, such as a category mask or an instance mask, the depth prediction module predicts a segment-centered canonical depth map, as well as a transformation to convert the canonical depth to the global depth space. In this way, we decompose depth prediction into local segment depth prediction and global transformation estimation, which we will show to be beneficial compared to the direct prediction baseline.

We use two depth prediction streams to handle semantic category segments and instance segments respectively. The category segment stream operates in the category-level by predicting depth for each entire category jointly, whereas for countable object classes, the absolute depth of an instance can vary a lot depending on its position in the scene. Therefore, the instance-wise depth stream is further designed to improve the depth map on a *per instance* basis.

#### Category-wise Depth Estimation

Given a semantic category, we use a two-branch architecture to predict its canonical depth and global transformation. As shown in Figure 3, the local branch consists of a stack of convolutional layers, which takes as input the backbone image feature pyramid and predicts the canonical depth for each semantic category. We use the sigmoid function to normalize the output depth into the canonical space. The global depth decoding branch contains a Global Average Pooling (GAP) layer and a stack of fully connected layers. It maps the input feature pyramid to a vector characterizing the global context of the input image, which is used to infer the global transformation \( \mathcal{T}_c(\cdot) \) for the \( c \)-th semantic category. Then, the global depth for the \( c \)-th category is computed as \( D_c = \mathcal{T}_c(\tilde{D}_c) \). In our experiments, we adopt an affine transformation \( \mathcal{T}_c(\tilde{D}_c) = w_c \cdot \tilde{D}_c + b_c \) for simplicity.

#### Instance-wise Depth Estimation

For object classes, such as human, car, etc., we can borrow the ROIAlign technique from Mask R-CNN \cite{he2017mask} to extract features per object instance, and map these to a depth map. However, the resolution of the default ROIAlign features is too low (28 \( \times \) 28) for accurate depth prediction, especially for larger objects.

To address this issue, we propose a new network architecture for high-resolution instance depth estimation (c.f. Figure 3). The instance stream consists of two branches, a...
fully convolutional local branch and an instance depth decoding branch. The local branch operates in a fully convolutional manner, which takes the backbone image feature pyramid as input and predicts a category-agnostic depth representation map $\tilde{F}$ of size $H \times W \times Z$ for the entire input image ($Z$ is set to 32 in our experiments). Given the bounding box location of the $i$-th object instance, its instance level depth map representation $\tilde{F}^i \in \mathbb{R}^{H_i \times W_i \times Z}$ can be computed by cropping from $\tilde{F}$, where $H_i \times W_i$ is the spatial size of its bounding box.

To predict the depth for the $i$-th instance, the depth decoding branch extracts a fixed-length feature vector from the instance region using ROIAlign\(^1\) on the backbone feature. Given the category $c_i \in \{1, 2, \ldots, K\}$ of the instance inferred by the segmentation module, the depth decoding branch then takes as input the ROIAlign feature vector as well as the normalized bounding box coordinates of the instance, and predicts a linear depth decoding function $H^i(\cdot) = G^i \circ C^i(\cdot)$ corresponding to the $c_i$-th object category. The function $C^i$ is a $1 \times 1$ convolutional layer, which linearly combines the $Z$ channels of the instance depth representation map $\tilde{F}^i$ into a instance-centric canonical depth map. The function $G^i$ is an affine transformation, which further transforms the canonical depth into the global depth $F^i \in \mathbb{R}^{H_i \times W_i}$ by adjusting its scale and shift. The parameters of the two functions for different object categories are produced by the depth decoding branch in a category-specific manner. Assume the total number of transformation parameters for each category is $n$. The depth decoding branch predicts $K$ sets of parameters through an output vector of length $n \times K$. We select the $c_i$-th set of parameters for the $i$-th instance.

### 3.2. Segmentation Guided Depth Aggregation

Now we have produced the category depth maps $\{D_c \in \mathbb{R}^{H \times W} | c = 1, \ldots, C\}$ and a set of object instance depth maps $\{F^i \in \mathbb{R}^{H_i \times W_i} | i = 1, 2, \ldots, N\}$ for a total number of $N$ object instances within the input image. To make the final depth prediction, a depth aggregation module combines per-segment depth maps based on the semantic segmentation and the instance segmentation results.

Our depth aggregation module proceeds in two steps. Given instance depth maps $F^i$ and their category labels, the first step performs local updates to the region of each instance in its corresponding category depth map $\{D_c \in \mathbb{R}^{H \times W} | c = 1, 2, \ldots, K\}$. To this end, we associate each object category depth map $D_c$ with an normalization mask $M_c$ with the same spatial size $H \times W$, whose elements are all initialized to constant value 1. The normalization masks are used to record the update from each instance depth map, and normalize the final depth map accordingly. Given the category $c_i$ and bounding box location of the $i$-th instance, we denote the instance region on the corresponding depth map $D_{c_i}$ and the normalization mask $M_{c_i}$, as $D_{c_i} \in \mathbb{R}^{H \times W}$ and $M_{c_i} \in \mathbb{R}^{H \times W}$, respectively. The depth map and the normalization mask can then be locally updated as follows:

$$D'_{c_i} \leftarrow D_{c_i} + v \times p^i \circ S^i \odot F^i, \quad M'_{c_i} \leftarrow M_{c_i} + v \times p^i \circ S^i, \quad (1)$$

where $v$ is a hyper-parameter to balance the weight of instance depth maps ($v$ is set to 10 in our experiments); $\circ$ indicates the element-wise multiplication; $p^i$ denotes the probability of the $i$-th instance belonging to category $c_i$, and $S^i$ of spatial size $H_i \times W_i$ represents the upsampled segmentation mask of the $i$-th instance. Both $p^i$ and $S^i$ are generated by the instance segmentation model of our segmentation module, and are used to measure the reliability of the $i$-th instance prediction.

After all instance regions have been updated, each category depth map $D_c$ is computed:

$$D_c \leftarrow D_c / M_c. \quad (2)$$

where the division is performed element-wisely. More details of the first step is summarized in Algorithm 1.

The second step aggregates all of the updated category depth maps $D_c$ according to the semantic segmentation results. This can be performed through the following weighted combination:

$$D = \sum_{c=1}^{C} P_c \odot D_c. \quad (3)$$

---

\(^1\)We use the same implementation of ROIAlign in [9].
Algorithm 1 Update category depth maps with instance depth maps.

Input: Category-specific depth maps \( \{D_c|c = 1, 2, \ldots, K\} \), instance-specific depth maps \( F^i \), instance segmentation masks \( S^i \), instance category \( c \), instance classification probability \( p_i \), \( i = 1, 2, \ldots, N \).

Output: Updated category-specific depth maps \( \{D_c|c = 1, 2, \ldots, K\} \)

1. Initialize normalization mask \( M_c \) for each category \( c \)
2. for \( i = 1, 2, \ldots, N \) do
3. Locate instance regions \( D^i_{c_i} \) and \( M^i_{c_i} \) on depth map and normalization masks.
4. Locally update depth map by \( D^i_{c_i} \leftarrow D^i_{c_i} + v \times p^i \circ S^i \circ F^i \)
5. Locally update energy mask by \( M^i_{c_i} \leftarrow M^i_{c_i} + v \times p^i \circ S^i \)
6. end for
7. Normalize each category depth map \( D_c \leftarrow D_c / M_c \)

where \( D \) represents the final depth map, \( P_c \) is the per-class, per-pixel segmentation result predicted by the semantic segmentation module, where for a class \( c \), an element located at \((x, y)\) represents the probability of the corresponding pixel belonging to that class.

3.3. Network Training

The whole system can be trained in an end-to-end manner using the following loss function:

\[
L = \alpha_1 L_1 + \alpha_S L_S + \alpha_D L_D,
\]

where we use standard implementations of the instance segmentation loss \( L_1 \) [9], and semantic segmentation losses \( L_S \) [27]. The depth prediction loss \( L_D \) varies depending on the depth supervision. On training datasets with dense depth annotations (e.g., NYU-Depth V2 [32] and Cityscapes [3]) we use a standard \( L_1 \) loss, and on datasets with relative depth annotations between pairs of random points (e.g., DIW dataset [2]), we use the ranking loss proposed in [2]. More details of the loss functions and their loss weights \( \alpha_1, \alpha_S, \alpha_D \) are present in the supplementary materials. All four modules of our method can then be jointly trained by minimizing the overall loss function in (4).

4. Experiments

4.1. Implementation

We adopt ResNet50 pre-trained on the ImageNet classification task to initialize our backbone network. Detailed architecture design for our depth prediction module can be found in the supplementary materials. We resize each input image to have a minimum side of 256 pixels while maintaining its aspect ratio. Data augmentation techniques including random flipping, scaling and color jitter have also been employed to avoid over-fitting. Our network is trained using Adam optimizer [13] with a mini-batch of 4 input images. Our whole network has 50.4 M parameters and runs at 10.23 FPS for 19 semantic categories on one NVIDIA GTX 1080 Ti GPU. Source code is available at https://bit.ly/39oty26.

We evaluate our method on three depth datasets, including Cityscapes [3], DIW [2] and NYU-Depth V2 [32], which involve either dense or sparse depth annotations, and contain diverse scenes. The performance of compared methods are measured by: RMSE in both linear and log space, absolute and squared relative error (Abs Rel and Sq Rel), depth accuracy (with thresholds 1.25, 1.25^2 and 1.25^3), and weighted human disagreement rate (WHDR) [4, 2]. We adopt the evaluation code of [5] to calculate the above metrics.

4.2. Cityscapes Results

Cityscapes [3] is a large dataset for urban scene understanding, containing both depth and panoptic segmentation annotations of 20 semantic categories. We train our model for 25 epochs on the training set of 2975 images with the initial learning rate of 5e-3. We evaluate the trained model on the validation (500 images) and test (1525 images) sets, and compare to 3 state-of-the-art methods including Laina et al. [14], Xu et al. [39], and Zhang et al. [41]. Among them, Xu et al. [39], and Zhang et al. [41] train their model on both depth estimation and semantic segmentation in a multi-task manner. Table 1 reports the results. Our method achieves higher performance than compared methods, particularly in terms of RMSE and depth accuracy. This should be mainly attributed to the fact that our method predicts each category and instance depth independently with specific depth decoders. Since Xu et al. and Zhang et al. also leverage semantic segmentation annotation, their performance is superior than Laina et al. Qualitative results are shown in Figure 4.

4.3. DIW Results

DIW [2] is a large-scale dataset containing images of diverse scenes in the wild, where each image is manually annotated with the relative depth order (either closer or further away from the camera) between one randomly sampled point pair. The whole dataset is split into 421K training images and 74K test images. Since DIW dataset does not contain segmentation annotations and the COCO panoptic segmentation dataset [23] also contains images of unconstrained scenes, we simultaneously train our model on DIW and COCO for relative depth estimation and segmentation, respectively. In order to reduce computational complexity, we adopt the super-class annotation of COCO dataset to train our segmentation module, containing 15 stuff and 12 object classes. During training, we sequentially feed training images from both datasets to the network in each itera-
### Table 1: Comparison with state-of-the-art methods on Cityscapes test set [3]. Best results are in **bold** font, second best are *underlined*.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>RMSE (log)</td>
</tr>
<tr>
<td>Laina et al. [14]</td>
<td>7.273</td>
<td>0.448</td>
</tr>
<tr>
<td>Xu et al. [39]</td>
<td>7.117</td>
<td>0.428</td>
</tr>
<tr>
<td>Zhang et al. [41]</td>
<td>7.104</td>
<td>0.416</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>6.917</strong></td>
<td><strong>0.414</strong></td>
</tr>
</tbody>
</table>

Figure 4: Qualitative results on Cityscapes dataset. The first three rows are input images, ground truth, and our predicted depth maps, respectively. The last two rows are error map comparisons of our method against Xu et al. [39] and Zhang et al. [41], respectively, where dark red indicates our method achieves lower error and dark blue is the opposite.

### Table 2: Comparison with state-of-the-art methods on DIW dataset [2]. The best result is in **bold** font.

<table>
<thead>
<tr>
<th>Method</th>
<th>WHDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. [2]</td>
<td>22.14%</td>
</tr>
<tr>
<td>Xian et al. [37]</td>
<td>14.98%</td>
</tr>
<tr>
<td>Xu et al. [39]</td>
<td>13.02%</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>11.21%</strong></td>
</tr>
</tbody>
</table>

We compare our method against three state-of-the-art approaches, including Chen et al. [2], Xian et al. [37], and Xu et al. [39], where Xu et al. [39] is trained on both DIW and COCO dataset using the same training strategy as ours. Table 2 shows the comparison results in terms of WHDR. Xian et al. achieves lower WHDR than Chen et al. Meanwhile, Xu et al. outperforms Xian et al. by exploring additional segmentation data. In comparison, our proposed method adopt a divide-and-conquer strategy to estimate depth for each segments independently, thus achieves the best performance. Figure 5 compares the predicted depth maps of Xu et al. [39] and our proposed method.
### 4.4. NYU-Depth V2 Results

The NYU-Depth V2 dataset contains 464 indoor scenes, where 249 of them are for training and the rest for testing. We train our networks by randomly sampling 40K images from the 249 training scenes and following the multi-task training strategy introduced in Section 4.3. We adopt an initial learning rate of 1e-3 and train the network for 15 epochs.

We compare our method to seven state-of-the-art methods. Among them, Lee et al. [15] and Fu et al. [5] use all the 120K training images, and Xu et al. [39] and Zhang et al. [41] also use the available segmentation supervision. As shown in Table 3, the proposed method performs favorably against state-of-the-art approaches, particularly in terms of depth accuracy by using a limited amount of segmentation annotations. We believe our performance can be further improved by using more segmentation data from indoor scenes.

### 4.5. Ablation Study

To achieve more comprehensive understanding of our method, we perform ablative study on Cityscapes [3] and DIW [2] datasets by adjusting different modules of our method. Unless otherwise stated, we follow the same experimental setup as described in Section 4.1.

#### Effects of semantic divide-and-conquer.

The proposed SDC-Depth Net learns category and instance aware depth estimation with disentangled canonical depth and scale inference (DEnt). To investigate the impact of the above design choices, we compare the performance of baselines (SDC-A to SDC-D) containing different subsets of these choices on the Cityscapes dataset as shown in Table 4. It can be observed that category aware depth estimation plays a very important role in improving the depth accuracy. Instance aware depth estimation and disentangled depth prediction also yield a considerable performance gain. To further verify their effectiveness, Figure 6 compares the depth accuracy of baseline methods with respect to different categories on Cityscapes validation set. The performance gain

---

**Table 3:** Comparison with state-of-the-art methods on NYU-Depth V2 dataset [32].

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
<th>RMSE (log)</th>
<th>Abs Rel</th>
<th>Accuracy δ &lt; 1.25</th>
<th>Accuracy δ &lt; 1.25^2</th>
<th>Accuracy δ &lt; 1.25^3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laina et al. [14]</td>
<td>0.584</td>
<td>0.198</td>
<td>0.136</td>
<td>0.822</td>
<td>0.956</td>
<td>0.989</td>
</tr>
<tr>
<td>Xu et al. [40]</td>
<td>0.593</td>
<td>-</td>
<td>0.125</td>
<td>0.806</td>
<td>0.952</td>
<td>0.986</td>
</tr>
<tr>
<td>Qi et al. [29]</td>
<td>0.569</td>
<td>-</td>
<td>0.128</td>
<td>0.834</td>
<td>0.960</td>
<td>0.990</td>
</tr>
<tr>
<td>Lee et al. [15]</td>
<td>0.572</td>
<td>0.193</td>
<td>0.139</td>
<td>0.815</td>
<td>0.963</td>
<td>0.991</td>
</tr>
<tr>
<td>Fu et al. [5]</td>
<td>0.509</td>
<td>0.188</td>
<td><strong>0.116</strong></td>
<td>0.828</td>
<td>0.965</td>
<td><strong>0.992</strong></td>
</tr>
<tr>
<td>Zhang et al. [41]</td>
<td>0.501</td>
<td>0.181</td>
<td>0.144</td>
<td>0.815</td>
<td>0.962</td>
<td><strong>0.992</strong></td>
</tr>
<tr>
<td>Xu et al. [39]</td>
<td>0.582</td>
<td>-</td>
<td>0.120</td>
<td>0.817</td>
<td>0.954</td>
<td>0.987</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.497</strong></td>
<td><strong>0.174</strong></td>
<td>0.128</td>
<td><strong>0.845</strong></td>
<td><strong>0.966</strong></td>
<td>0.990</td>
</tr>
</tbody>
</table>

**Table 4:** Ablation study on Cityscapes dataset [3]. Components tested are category (Cat.) and instance (Ins.) depth estimation, and disentangling canonical depth and scale inference (DEnt). The best results are in **bold** font.

<table>
<thead>
<tr>
<th>Design Choice</th>
<th>SDC-A</th>
<th>SDC-B</th>
<th>SDC-C</th>
<th>SDC-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat.</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Ins.</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DEnt</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Err. RMSE</td>
<td>7.203</td>
<td>6.962</td>
<td>6.962</td>
<td><strong>6.917</strong></td>
</tr>
<tr>
<td>Abs Rel</td>
<td>0.276</td>
<td>0.236</td>
<td>0.234</td>
<td><strong>0.227</strong></td>
</tr>
<tr>
<td>Acc. δ &lt; 1.25</td>
<td>0.767</td>
<td>0.794</td>
<td>0.797</td>
<td><strong>0.801</strong></td>
</tr>
<tr>
<td>δ &lt; 1.25^2</td>
<td>0.895</td>
<td>0.911</td>
<td>0.911</td>
<td><strong>0.913</strong></td>
</tr>
<tr>
<td>δ &lt; 1.25^3</td>
<td>0.941</td>
<td>0.949</td>
<td>0.951</td>
<td><strong>0.950</strong></td>
</tr>
</tbody>
</table>

Figure 5: Qualitative results on DIW test set [2]. All methods are trained on sparse point-level supervision.
of the category-aware depth estimation is consistent across all categories while instance-aware depth estimation is more effective for object categories.

We also visualize the average canonical depth maps, for a number of different segments (Figure 7). We can see how by splitting the depth prediction at a segment level, the network can learn simpler category-specific depth priors.

**Benefits of segmentation annotation.** To evaluate how much our method benefits from additional segmentation annotations, we train 4 instances of our SDC-Depth Net on DIW and COCO datasets using different portions (25%, 50%, 75%, and 100%, respectively) of the COCO dataset for training. In addition to this, we also train two baseline networks that are encoder-decoder architectures that directly predict depth maps with a similar parameter count to SDC-Depth Net. One of the baseline is trained only on DIW, named as BNet-DIW. The other one is trained on both DIW and COCO datasets in a multi-task learning manner, named as BNet-DIW-COCO, as in [35]. The comparison results in terms of depth accuracy on DIW test set are shown in Figure 8. It can be observed that the performance can be consistently improved by using more segmentation training data, and when using all the COCO training data, the proposed method outperforms BNet-DIW-COCO with a significant margin.

**5. Conclusion**

We present a semantic divide-and-conquer strategy to reduce monocular depth estimation into that of individual semantic segments. Based on this idea, the SDC-Depth Net is designed, which decomposes an input image into segments of different categories and instances, and infers the canonical depth as well as the scale-and-shift transformation for each segment using specifically trained parameters. An aggregation method is also developed to stitch the per-segment depth into the final depth map. The whole network can be trained fully end-to-end by leveraging additional segmentation annotations. Experiments on three popular benchmarks demonstrates the effectiveness of our method.

**Acknowledgments.** This work is supported by National Key R&D Program of China (2018AAA0102001), National Natural Science Foundation of China (61725202, U1903215, 61829102, 91538201, 61751212, 61906031), Fundamental Research Funds for the Central Universities (DUT19GJ201), China Postdoctoral Science Foundation (2019M661095), National Postdoctoral Program for Innovative Talent (BX20190055), and Adobe Research.
References


[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016. 5, 6, 7, 8


549


[33] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes, 2019. 2


