

Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation

Dan Xu¹, Wei Wang¹, Hao Tang¹, Hong Liu^{2*}, Nicu Sebe¹, Elisa Ricci^{1,3*}

¹Multimedia and Human Understanding Group, University of Trento,

²Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University,

³ Technologies of Vision Group, Fondazione Bruno Kessler

{dan.xu, wei.wang, hao.tang, niculae.sebe, e.ricci}@unitn.it hongliu@pku.edu.cn

Abstract

Recent works have shown the benefit of integrating Conditional Random Fields (CRFs) models into deep architectures for improving pixel-level prediction tasks. Following this line of research, in this paper we introduce a novel approach for monocular depth estimation. Similarly to previous works, our method employs a continuous CRF to fuse multi-scale information derived from different layers of a front-end Convolutional Neural Network (CNN). Differently from past works, our approach benefits from a structured attention model which automatically regulates the amount of information transferred between corresponding features at different scales. Importantly, the proposed attention model is seamlessly integrated into the CRF, allowing end-to-end training of the entire architecture. Our extensive experimental evaluation demonstrates the effectiveness of the proposed method which is competitive with previous methods on the KITTI benchmark and outperforms the state of the art on the NYU Depth V2 dataset.

1. Introduction

The problem of recovering depth information from images has been widely studied in computer vision. Traditional approaches operate by considering multiple observations of the scene of interest, *e.g.* derived from two or more cameras or corresponding to different lighting conditions. More recently, the research community has attempted to relax the multi-view assumption by addressing the task of monocular depth estimation as a supervised learning problem. Specifically, given a large training set of pairs of images and associated depth maps, depth prediction is casted as a pixel-level regression problem, *i.e.* a model is learned to directly predict the depth value corresponding to each pixel of an RGB image.

In the last few years several approaches have been proposed for addressing this task and remarkable perfor-

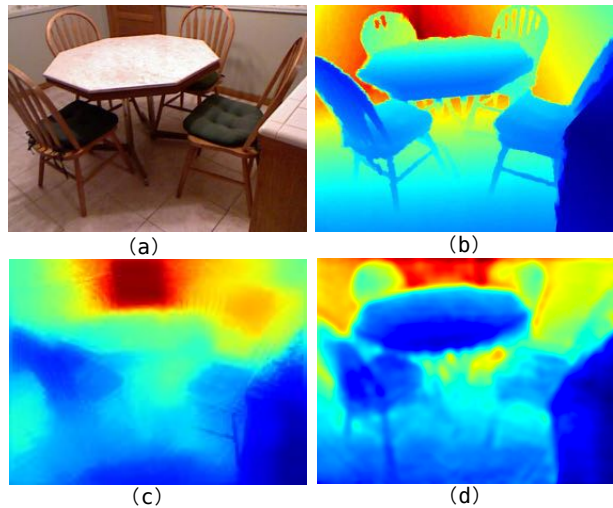


Figure 1. Monocular depth prediction from different CRF-based models: Xu *et al.* [36] (c) and ours (d). (a) and (b) are the input RGB image and the corresponding ground truth depth map.

mance has been achieved thanks to deep learning models [5, 6, 22, 36, 18]. Recently, various Convolutional Neural Network (CNN) architectures have been proposed, tackling different sub-problems such as how to jointly estimate depth maps and semantic labels [35], how to build models robust to noise or how to combine multi-scale features [10]. Focusing on the latter issue, recent works have shown that CRFs can be integrated into deep architectures [22, 31] and can be exploited to optimally fuse the multi-scale information derived from inner layers of a CNN [36].

Inspired by these works, in this paper we also propose to exploit the flexibility of graphical models for multi-scale monocular depth estimation. However, we significantly depart from previous methods and we argue that more accurate estimates can be obtained operating not only at the prediction level but exploiting directly the internal CNN feature representations. To this aim, we design a novel CRF model which automatically learns robust multi-scale features by integrating an attention mechanism. Our attention model allows to automatically regulate how much information should flow between related features at different scales.

*corresponding author.

Attention models have been successfully adopted in computer vision and they have shown to be especially useful for improving the performance of CNNs in pixel-level prediction tasks, such as semantic segmentation [4, 13]. In this work we demonstrate that attention models are also extremely beneficial in the context of monocular depth prediction. We also show that the attention variables can be jointly estimated with multi-scale feature representations during CRF inference and that, by employing a structured attention model [17] (*i.e.* by imposing similarity constraints between attention variables for related pixels and scales), we can further boost performance. Through extensive experimental evaluation we demonstrate that our method produces more accurate depth maps than traditional approaches based on CRFs [22, 31] and multi-scale CRFs [36] (Fig.1). Moreover, by performing experiments on the publicly available NYU Depth V2 [30] and on the KITTI [8] datasets, we show that our approach outperforms most state of the art methods.

Contributions. In summary, we make the following contributions: (i) We propose a novel deep learning model for calculating depth maps from still images which seamlessly integrates a front-end CNN and a multi-scale CRF. Importantly, our model can be trained end-to-end. Differently from previous works [36, 22, 31] our framework does not consider as input only prediction maps but operates directly at feature-level. Furthermore, by adopting appropriate unary and pairwise potentials, our framework allows a much faster inference. (ii) Our approach benefits from a novel attention mechanism which allows to robustly fuse features derived from multiple scales as well as to integrate structured information. (iii) Our method demonstrates state-of-the-art performance on the NYU Depth V2 [30] dataset and is among the top performers on the more challenging outdoor scenes of the KITTI benchmark [8]. The code is made publicly available¹.

2. Related work

Monocular Depth Estimation. The problem of monocular depth estimation has attracted considerable attention in last decade. While earlier approaches are mostly based on hand-crafted features [12, 16, 19, 28], more recent works adopt deep architectures [5, 22, 31, 26, 20, 36, 9]. In [6] a model based on two CNNs is proposed: a first network is used for estimating depth at a coarse scale, while the second one is adopted to refine predictions. In [20] a residual network integrating a novel reverse Huber loss is presented. In [2] a deep residual network is also employed but the problem of depth estimation from still images is translated from a regression to a classification task. Recent works have also shown the benefit of adopting multi-task

learning strategies, *e.g.* for jointly predicting depth and performing semantic segmentation, ego-motion estimation or surface normal computation [5, 38, 31]. Some recent papers have proposed unsupervised or weakly supervised methods for reconstructing depth maps [9, 18]. Other works have exploited the flexibility of graphical models within deep learning architectures for estimating depth maps. For instance, in [31] a Hierarchical CRF is adopted to refine depth predictions obtained by a CNN. In [22] a continuous CRF is proposed for generating depth maps from CNN features computed on superpixels. The most similar work to ours is [36], where a CRF is adopted to combine multi-scale information derived from multiple inner layer of a CNN. Our approach develops from a similar intuition but further integrates an attention model which significantly improves the accuracy of the estimates. To our knowledge this is the first paper exploiting attention mechanisms in the context of monocular depth estimation.

Fusing Multi-scale Information in CNNs. Many recent works have shown the benefit of combining multi-scale information for pixel-level prediction tasks such as semantic segmentation [3], depth estimation [36] or contour detection [32]. For instance, dilated convolutions are employed in [3]. Multi-stream architectures with inputs at different resolutions are considered in [1], while [25] proposed skip-connections to fuse feature maps derived from different layers. In [32] deep supervision is exploited for fusing information from multiple inner layers. CRFs have been considered for integrating multi-scale information in [36]. In [4, 33] an attention model is employed for combining multi-scale features in the context of semantic segmentation and object contour detection. The approach we present in this paper is radically different, as we employ a structured attention model which is jointly learned within a CRF-CNN framework.

3. Estimating Depth Maps with Structured Attention Guided Conditional Neural Fields

In this section we describe our approach for estimating depth maps from still images. We first provide an overview of our method and then introduce the proposed CRF model with structured attention. We conclude this section providing some details about our implementation.

3.1. Problem Formulation and Overview

As stated in the introduction, the problem of predicting a depth map from a single RGB image can be treated as a supervised learning problem. Denoting as \mathcal{I} the space of RGB images and as \mathcal{D} the domain of real-valued depth maps, given a training set $\mathcal{T} = \{(\mathbf{I}_i, \mathbf{D}_i)\}_{i=1}^M$, $\mathbf{I}_i \in \mathcal{I}$ and $\mathbf{D}_i \in \mathcal{D}$, we are interested in learning a non-linear mapping $\Phi : \mathcal{I} \rightarrow \mathcal{D}$.

¹<https://github.com/danxuhk/StructuredAttentionDepthEstimation>

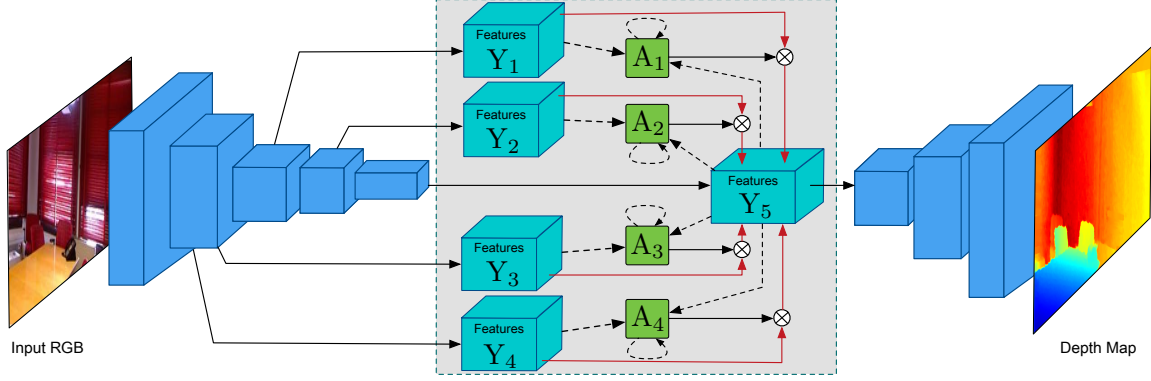


Figure 2. Illustration of the proposed network for monocular depth estimation. The blue blocks indicate the front-end CNN, which in our implementation is made by an encoder and associated decoder (Section 3.3). The gray box contains a schematic representation of the proposed Structured Attention guided CRF model. Inside, the green boxes indicate the estimated attention maps, while the light blue ones represent the features jointly inferred with our CRF. The arrows indicates the dependencies among the estimated variables used in our message passing algorithm (Section 3.2.3). The dashed arrows indicate the updates involving the attention model.

In analogy with previous works [22, 36], we propose to learn the mapping Φ by building a deep architecture which is composed by two main building blocks: a front-end CNN and a CRF model. The main purpose of the proposed CRF model is to combine multi-scale information derived from the inner layers of the front-end CNN. Differently from previous research [22, 36], our CRF model does not simply act in order to refine the final prediction map of the CNN neither requires as input multiple score maps of the same size. In this paper we argue that better estimates can be obtained with a more flexible model which accepts as inputs a set of S multi-scale feature maps $\mathbf{X} = \{\mathbf{X}_s\}_{s=1}^S$ derived directly from the front-end intermediate layers. To facilitate the modeling, all the multi-scale feature maps are resized to the same resolution via upsampling or downsampling operations. Here $\mathbf{X}_s = \{\mathbf{x}_s^i\}_{i=1}^N$, $\mathbf{x}_s^i \in \mathbb{R}^N$, indicates a set of feature vectors.

The main idea behind the design of the proposed multi-scale CRF model is to estimate the depth map \mathbf{D} associated to an RGB image \mathbf{I} by exploiting the features at the last layer \mathbf{X}_S and a set of auxiliary feature representations derived from the intermediate scales $s = 1, \dots, S-1$. To do that, we propose to learn a set of latent feature maps $\mathbf{Y}_s = \{\mathbf{y}_s^i\}_{i=1}^N$, $s = 1, \dots, S$ and to model the dependencies between the representations learned at the last layer and those corresponding to each intermediate scales by introducing an appropriate attention model $\mathbf{A} = \{\mathbf{A}_s\}_{s=1}^{S-1}$, parameterized by binary variables $\mathbf{A}_s = \{a_s^i\}_{i=1}^N$, $a_s^i \in \{0, 1\}$. Intuitively, the attention variable a_s^i regulates the information which is allowed to flow between each intermediate scale s and the final scale S for pixel i . In other words, by learning the attention maps we automatically discover which information derived from inner CNN representations is relevant for final depth estimation. Furthermore, in order to obtain accurate attention maps \mathbf{A}_s we propose to learn a structured attention model, *i.e.* we impose structural constraints on the estimated

variables a_s^i enforcing those corresponding to neighboring pixels to be related. Importantly, the proposed CRF jointly infers the hidden features and the attention maps.

Figure 2 schematically depicts the proposed framework and our CRF model. The idea of modeling the relationships between the learned representations at the finer scale and the features corresponding to each intermediate layer is inspired by the recent DenseNet architecture [14]. As demonstrated in our experiments (Section 4), this strategy leads to improved performance with respect to a cascade model as proposed in [36].

3.2. Structured Attention Guided Multi-Scale CRF

3.2.1 Proposed Model

Given the observed multi-scale feature maps \mathbf{X} , we jointly estimate the latent multi-scale representations \mathbf{Y} and the attention variables \mathbf{A} by designing a Conditional Random Field model with the following associated energy function:

$$E(\mathbf{Y}, \mathbf{A}) = \Phi(\mathbf{Y}, \mathbf{X}) + \Xi(\mathbf{Y}, \mathbf{A}) + \Psi(\mathbf{A}) \quad (1)$$

The first term in (1) is the sum of unary potentials relating the latent features representations \mathbf{y}_s^i with the associated observations \mathbf{x}_s^i , *i.e.* :

$$\Phi(\mathbf{Y}, \mathbf{X}) = \sum_{s=1}^S \sum_i \phi(\mathbf{y}_s^i, \mathbf{x}_s^i) = - \sum_{s=1}^S \sum_i \frac{1}{2} \|\mathbf{y}_s^i - \mathbf{x}_s^i\|^2 \quad (2)$$

As in previous works [22, 36] we consider Gaussian functions, such as to enforce the estimated latent features to be close to their corresponding observations. The second term is defined as:

$$\Xi(\mathbf{Y}, \mathbf{A}) = \sum_{s \neq S} \sum_{i,j} \xi(a_s^i, \mathbf{y}_s^i, \mathbf{y}_S^j) \quad (3)$$

It models the relationship between the latent features at the last scale with those of each intermediate scale. This term also involves the attention variables a_s^i which regulate the

flow of information between related scales. We define:

$$\xi(a_s^i, \mathbf{y}_s^i, \mathbf{y}_S^j) = a_s^i \xi_y(\mathbf{y}_s^i, \mathbf{y}_S^j) = a_s^i \mathbf{y}_s^i \mathbf{K}_{i,j}^s \mathbf{y}_S^j \quad (4)$$

where $\mathbf{K}_{i,j}^s \in \mathbb{R}^{C_s \times C_S}$ and C_s, C_S refer to the number of channels of features scale s and S , respectively. Finally, the third term in (1) aims to enforce some structural constraints among attention variables. For instance, it is reasonable to assume that the estimated attention maps for related pixels and scales should be similar. To keep the computational cost limited, we only consider dependencies among attention variables at the same scale and we define:

$$\Psi(\mathbf{A}) = \sum_{s \neq S} \sum_{i,j} \psi(a_s^i, a_s^j) = \sum_{s \neq S} \sum_{i,j} \beta_{i,j}^s a_s^i a_s^j \quad (5)$$

where $\beta_{i,j}^s$ are coefficients to be learned. To model dependencies between pairs of attention variables we consider a bilinear function, in analogy with previous works [17].

3.2.2 Deriving Mean-Field Updates

Following previous works [37, 36] we resort on mean-field approximation. We derive mean-field inference equations for both latent features and attention variables. By denoting as \mathbb{E}_q the expectation with respect to the distribution q , we get:

$$q(\mathbf{y}_s^i) \propto \exp\left(\phi(\mathbf{y}_s^i, \mathbf{x}_s^i) + \right. \quad (6)$$

$$\left. \mathbb{E}_{q(a_s^i)}\{a_s^i\} \sum_j \mathbb{E}_{q(\mathbf{y}_S^j)}\{\xi_y(\mathbf{y}_s^i, \mathbf{y}_S^j)\}\right),$$

$$q(\mathbf{y}_S^j) \propto \exp\left(\phi(\mathbf{y}_S^j, \mathbf{x}_S^j) + \right. \quad (7)$$

$$\left. \sum_{s \neq S} \sum_j \mathbb{E}_{q(a_s^j)}\{a_s^j\} \mathbb{E}_{q(\mathbf{y}_s^i)}\{\xi_y(\mathbf{y}_S^j, \mathbf{y}_s^i)\}\right),$$

$$q(a_s^i) \propto \exp\left(a_s^i \mathbb{E}_{q(\mathbf{y}_s^i)}\left\{\sum_j \mathbb{E}_{q(\mathbf{y}_S^j)}\{\xi_y(\mathbf{y}_s^i, \mathbf{y}_S^j)\}\right\} \right. \quad (8)$$

$$\left. + \sum_s \sum_j \mathbb{E}_{q(a_s^j)}\{\psi(a_s^i, a_s^j)\}\right),$$

By considering the potentials defined in (2), (3) and (5) and denoting as $\bar{a}_s^i = \mathbb{E}_{q(a_s^i)}\{a_s^i\}$ and $\bar{\mathbf{y}}_s^i = \mathbb{E}_{q(\mathbf{y}_s^i)}\{\mathbf{y}_s^i\}$, the following mean-fields updates can be derived for the latent feature representations:

$$\bar{\mathbf{y}}_s^i = \mathbf{x}_s^i + \bar{a}_s^i \sum_j \mathbf{K}_{i,j}^s \bar{\mathbf{y}}_S^j \quad (9)$$

$$\bar{\mathbf{y}}_S^j = \mathbf{x}_S^j + \sum_{s \neq S} \sum_j \bar{a}_s^j \mathbf{K}_{i,j}^s \bar{\mathbf{y}}_s^i \quad (10)$$

Since a_s^i are binary variables, $\bar{a}_s^i = \frac{q(a_s^i=1)}{q(a_s^i=0)+q(a_s^i=1)}$. Therefore, the updates for \bar{a}_s^i can be derived considering (8) and the definitions of potential functions (3) and (5):

$$\bar{a}_s^i = \sigma\left(-\sum_j \bar{\mathbf{y}}_s^i \mathbf{K}_{i,j}^s \bar{\mathbf{y}}_S^j - \sum_s \sum_j \beta_{i,j}^s \bar{a}_s^j\right) \quad (11)$$

where $\sigma()$ denotes the sigmoid function. Eqn. (11) shows

that, in analogy with previous methods employing an attention model [4, 13], in our framework we also compute the attention variables by applying a sigmoid function to the features derived by our CNN model. In addition, as we also consider dependencies among different a_s^i as in structured models [17], our updates also involve related attention variables.

To infer the latent multi-scale representations \mathbf{Y} and the attention variables \mathbf{A} , we implement the mean-field updates as a neural network (see Section 3.2.3). In this way we are able to simultaneously learn the parameters of the CRFs and those of the front-end CNN. When the inference is complete, the final depth map is obtained considering the final estimate associated to the last scale $\bar{\mathbf{y}}_S$ (see Section 3.3).

3.2.3 Implementation with Neural Networks

To enable end-to-end optimization of the whole network, we implement the proposed multi-scale model in neural networks. The target is to perform mean-field updates for both the attention variables and the multi-scale feature maps according to the derivation described in Section 3.2.2.

To perform mean-field updates of the attention model \mathbf{A} we follow (11). In practice, the update of each attention map \mathbf{a}_s can be implemented in several steps as follows: (i) perform the message passing from the two associated feature maps $\bar{\mathbf{y}}_s$ and $\bar{\mathbf{y}}_S$ ($\bar{\mathbf{y}}_s$ and $\bar{\mathbf{y}}_S$ are initialized with corresponding feature observations \mathbf{x}_s and \mathbf{x}_S , respectively). The message passing is performed via convolutional operations as $\hat{\mathbf{a}}_s \leftarrow \mathbf{y}_s \odot (\mathbf{K}_s \otimes \bar{\mathbf{y}}_S)$, where \mathbf{K}_s is a convolutional kernel corresponding to the s -th scale and the symbols \otimes and \odot denote the convolutional and the element-wise product operation, respectively; (ii) perform the message passing on the attention map with $\tilde{\mathbf{a}}_s \leftarrow \beta_s \otimes \hat{\mathbf{a}}_s$, where β_s is a convolutional kernel; (iii) perform the normalization with sigmoid function $\bar{\mathbf{a}}_s \leftarrow \sigma(-(\hat{\mathbf{a}}_s \oplus \tilde{\mathbf{a}}_s))$, where \oplus denotes element-wise addition operation.

When the attention maps are updated, we use them as guidance to update the last scale feature map \mathbf{y}_S . The mean-field updates of \mathbf{y}_S can be carried out according to (10) as follows: (i) perform the message passing from the s -th scale to the S -th scale by $\hat{\mathbf{y}}_S \leftarrow \mathbf{K}_s \otimes \bar{\mathbf{y}}_s$; (ii) multiply for the attention model and add the unary term \mathbf{x}_S by $\bar{\mathbf{y}}_S \leftarrow \mathbf{x}_S \oplus \sum_s (\bar{\mathbf{a}}_s \odot \hat{\mathbf{y}}_S)$. The computation of mean-field updates for the latent features corresponding to intermediate scales can be performed similarly, according to (9). In our implementation to reduce the computational overhead, we do not perform the mean-field updates for the intermediate scales. The attention maps and the last scale feature map are iteratively updated.

We would like to remark that, as a consequence of the definition of the potential functions in (2), (3) and (5), the computations of the mean-field updates in our approach are much more efficient than in [36] where Gaussian functions

Table 1. NYU Depth V2 dataset: comparison with state of the art. In bold we indicate the best method adopting on the original set, while in italics we indicate the best method using the extended set.

Method	Extra Training Data ?	Error (lower is better)			Accuracy (higher is better)		
		rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena <i>et al.</i> [29]	No (795)	0.349	-	1.214	0.447	0.745	0.897
Karsch <i>et al.</i> [16]	No (795)	0.35	0.131	1.20	-	-	-
Liu <i>et al.</i> [24]	No (795)	0.335	0.127	1.06	-	-	-
Ladicky <i>et al.</i> [19]	No (795)	-	-	-	0.542	0.829	0.941
Zhuo <i>et al.</i> [39]	No (795)	0.305	0.122	1.04	0.525	0.838	0.962
Wang <i>et al.</i> [31]	No (795)	0.220	0.094	0.745	0.605	0.890	0.970
Liu <i>et al.</i> [23]	No (795)	0.213	0.087	0.759	0.650	0.906	0.976
Roi and Todorovic [26]	No (795)	0.187	0.078	0.744	-	-	-
Xu <i>et al.</i> [36]	No (795)	0.139	0.063	0.609	0.793	0.948	0.984
Ours	No (795)	0.125	0.057	0.593	0.806	0.952	0.986
Eigen <i>et al.</i> [6]	Yes (120K)	0.215	-	0.907	0.611	0.887	0.971
Eigen and Fergus [5]	Yes (120K)	0.158	-	0.641	0.769	0.950	0.988
Laina <i>et al.</i> [20]	Yes (12K)	0.129	0.056	0.583	0.801	0.950	0.986
Li <i>et al.</i> [21]	Yes (24K)	0.139	0.058	0.505	0.820	0.960	0.989
Xu <i>et al.</i> [36]	Yes (12K)	<i>0.121</i>	<i>0.052</i>	<i>0.586</i>	<i>0.811</i>	<i>0.954</i>	<i>0.987</i>

are considered for pairwise potentials. Indeed, Gaussian convolutions involve a much higher computational overhead both in the forward and in the backward pass. We further discuss this aspect in Section 4.

3.3. Network Structure and Optimization

Network Structure and Implementation. The overall framework for monocular depth estimation is made by a CNN architecture and the proposed CRF model (Fig. 2). The CNN architecture is made of two main components, *i.e.* a fully convolutional encoder and a fully convolutional decoder. The encoder naturally supports any network structure. In this work we specifically employ ResNet-50 [11]. In our implementation the proposed CRF is adopted to refine the last scale feature map derived from the semantic layer *res5c*, which receives message from the other scale feature maps derived from *res3c* and *res4f*. *res3c*, *res4f* and *res5c* are the last layers of different convolutional blocks. In each convolutional block, every layer outputs a feature map with the same number of channels. Before message passing, all the feature maps are first upsampled using a deconvolutional operation to the same size, *i.e.* 1/4 resolution of the original input image, and the number of channels is set to 256 for all of them. The kernel size for both \mathbf{K}_s and β_s is set to 3 with stride 1 and padding 1 to have a local receptive field and to speed up the calculation.

The proposed multi-scale CRF module outputs a refined feature map. To obtain the final prediction we upsample the feature map to the original resolution as the input image using deconvolutional operations. Each time we upsample the feature map by a factor of 2, at the same time reducing by half the number of feature channels.

End-to-end optimization. As stated above, the proposed model can be trained end-to-end, *i.e.* the parameters of the front-end encoder Θ_e , those associated to the structured at-

tention guided CRF Θ_c , and those of the decoder Θ_d can be jointly optimized. Given the training data set \mathcal{T} , following previous works [36], we use a square loss function for the optimization, *i.e.* :

$$\mathcal{L}_F(\mathcal{I}, \mathcal{D}; \Theta_e, \Theta_c, \Theta_d) = \sum_{i=1}^M \|F(\mathbf{I}_i^l; \Theta_e, \Theta_c, \Theta_d) - \mathbf{D}_i^l\|_2^2$$

The whole network is jointly optimized via back-propagation with standard stochastic gradient descent.

4. Experiments

We demonstrate the effectiveness of the proposed approach performing experiments on two publicly available datasets: the NYU Depth V2 [30] and the KITTI [8] datasets. The following subsections describe our experimental setup and the results of our evaluation.

4.1. Experimental Setup

Datasets. The NYU Depth V2 dataset [30] has 120K pairs of RGB and depth maps gathered with a Microsoft Kinect. The image resolution is 640×480 pixels. The dataset is split into a training (249 scenes) and a test set (215 scenes). Following previous works [22, 39, 36] in our experiments we consider a subset of 1449 RGB-D pairs, of which 795 are used for training and the rest for testing. The data augmentation is performed on the fly by cropping the images to 320×240 pixels, randomly flipping and scaling them with a ratio $\rho \in \{1, 1.2, 1.5\}$.

The KITTI dataset [8], originally built for testing computer vision algorithms in several tasks in the context of autonomous driving, contains depth images captured with a LiDAR sensor mounted on a driving vehicle. In our experiments we follow the experimental protocol proposed by Eigen *et al.* [6] and consider 22,600 images corresponding to 32 scenes as training data and 697 images associated

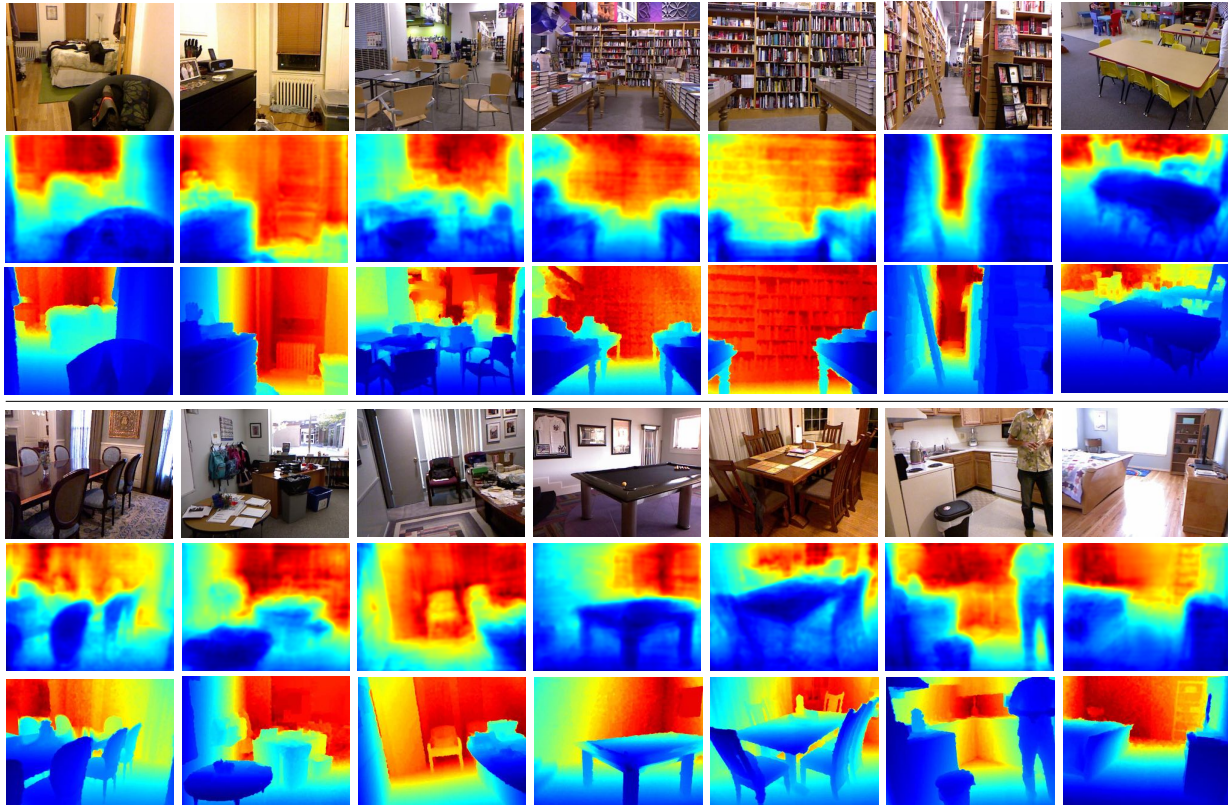


Figure 3. Examples of predicted depth maps on the NYU V2 test dataset: original RGB images (top row), predicted depth maps (center) and ground truth (bottom row).

Table 2. KITTI dataset: comparison with state of the art. In bold we indicate the best performances in the monocular setting, while in italics those corresponding to the stereo setting.

Method	Setting		Error (lower is better)			Accuracy (higher is better)		
	range	stereo?	rel	sq rel	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Saxena <i>et al.</i> [29]	0-80m	No	0.280	-	8.734	0.601	0.820	0.926
Eigen <i>et al.</i> [6]	0-80m	No	0.190	-	7.156	0.692	0.899	0.967
Liu <i>et al.</i> [23]	0-80m	No	0.217	0.092	7.046	0.656	0.881	0.958
Zhou <i>et al.</i> [38]	0-80m	No	0.208	1.768	6.858	0.678	0.885	0.957
Kuznetsov <i>et al.</i> [18] (only supervised)	0-80m	No	-	-	4.815	0.845	0.957	0.987
Ours	0-80m	No	0.122	0.897	4.677	0.818	0.954	0.985
Garg <i>et al.</i> [7]	0-80m	Yes	0.177	1.169	5.285	0.727	0.896	0.962
Garg <i>et al.</i> [7] L12 + Aug 8x	1-50m	Yes	0.169	1.080	5.104	0.740	0.904	0.958
Godard <i>et al.</i> [9]	0-80m	Yes	0.148	1.344	5.927	0.803	0.922	0.963
Kuznetsov <i>et al.</i> [18]	0-80m	Yes	-	-	4.621	<i>0.852</i>	<i>0.960</i>	<i>0.986</i>

to other 29 scenes as test data. The RGB image resolution is reduced by half with respect to the original 1224×368 pixels. The ground-truth depth maps are generated by re-projecting the 3D points collected from velodyne laser into the left monocular camera as detailed in [7].

Implementation Details. The proposed approach is implemented using the Caffe framework [15] and runs on a single Nvidia Titan X GPU with 12 GB memory. While the proposed framework is general, following recent works [36, 20], we adopt the ResNet50 [11] as the front-end network architecture. As stated above in the implementation, we consider three-level feature maps derived from different semantic convolutional layers (*i.e.* *res3c*, *res4f* and

res5c). These feature maps are fused with the proposed CRF model for the final prediction of the depth map. During training, the front-end network is initialized with ImageNet pretrained parameters. Differently from [36] which requires a pretraining phase of the front-end CNN, we jointly optimize the whole network. The initial learning rate is set to $10e-9$, and is decreased 10 times every 40 epochs. In total 60 epochs are used for training. The mini-batch size is set to 16. The weight decay and the momentum are 0.0005 and 0.99, respectively.

Evaluation Metrics. In analogy with previous works [5, 6, 31, 36], to quantitatively assess the performance of our method we consider several evaluation metrics. Specif-

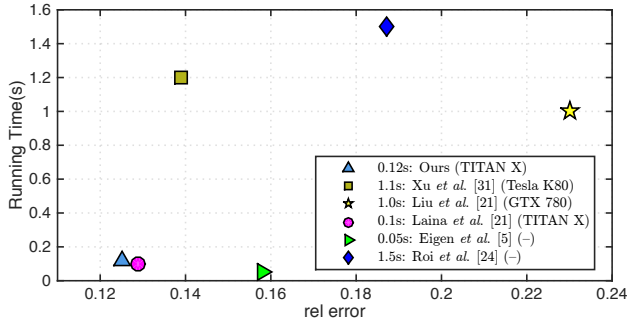


Figure 4. NYU V2 dataset. Comparison with previous methods: running time vs. rel error.

ically if Q is the total number of pixels of the test set and \bar{d}_i and d_i denote the estimated and the ground-truth depth for pixel i , we compute: (i) the mean relative error (rel): $\frac{1}{Q} \sum_{i=1}^Q \frac{|\bar{d}_i - d_i|}{d_i}$; (ii) the root mean squared error (rms): $\sqrt{\frac{1}{Q} \sum_{i=1}^Q (\bar{d}_i - d_i)^2}$; (iii) the mean log10 error (log10): $\frac{1}{Q} \sum_{i=1}^Q \|\log_{10}(\bar{d}_i) - \log_{10}(d_i)\|$ and (iv) the accuracy with threshold t , *i.e.* the percentage of \bar{d}_i such that $\delta = \max(\frac{d_i}{\bar{d}_i}, \frac{\bar{d}_i}{d_i}) < t$, where $t \in [1.25, 1.25^2, 1.25^3]$.

In order to compare our results with previous methods on the KITTI dataset we crop our images using the evaluation crop applied by Eigen *et al.* [6].

4.2. Experimental Results

To demonstrate the effectiveness of the proposed framework we first conduct a comparison with state of the art methods both on the NYU Depth V2 dataset and on the KITTI benchmark. We also conduct an in-depth analysis of our method, evaluating both accuracy and computational efficiency.

NYU Depth V2 Dataset. Table 1 shows the results of the comparison with state of the art methods on the NYU Depth V2 dataset. As baselines we consider both approaches based on hand-crafted features (Saxena *et al.* [29], Karsch *et al.* [16], Ladicky *et al.* [19]) and deep learning architectures. Concerning the latter category, we compare with methods which exploit multiscale information (Eigen *et al.* [6], Eigen and Fergus [5], Li *et al.* [21]), with approaches which consider graphical models (Liu *et al.* [24], Liu *et al.* [22], Zhuo *et al.* [39], Wang *et al.* [31], Xu *et al.* [36]) and neural regression forests ([26]), and with methods which explore the utilization of the reverse Huber loss function (Laina *et al.* [20]). The numerical results associated to previous methods are taken directly from the original papers. For a fair comparison in the table we also report information about the adopted training set, as it represents an important factor for CNN performance. In particular, we separate methods which adopt the original training set in [6] and those which consider an extended dataset for learning their deep models.

Our results clearly show that the proposed approach outperforms all supervised learning methods adopting the original dataset in [6]. Importantly, the performance improvements over previous works based on CRFs models [31, 22, 36] are significant. In particular, we believe that the increase in accuracy with respect to [36] confirms our initial intuition that operating directly at feature-level and integrating an attention model into a CRF leads to more accurate depth estimates. Finally, we would like to point out that our approach also outperforms most methods considering an extended training set. Furthermore, the performance gap between our framework and the deep model in [36] trained on 95K samples is very narrow. We also provide some examples of depth maps estimated with the proposed method in Fig. 3. Comparing our prediction with ground truth it is clear that our approach is quite accurate even at objects boundaries (notice, for instance, the accuracy in recovering fine-grained details in case of objects like chairs and tables).

Finally, we compare the proposed approach with previous methods considering the computational cost in the test phase. Figure 4 depicts the mean relative error vs the running time (*i.e.* time to classify one image) for some of the baseline methods (numbers are taken from the original papers). Our approach guarantees the best trade-off between accuracy and time (notice that the deep model in Laina *et al.* [20] is trained on an extended dataset). It is interesting to compare our method with [36]: the proposed framework not only outperforms [36] in terms of accuracy when both models are trained on the original set [6] but, by adopting different potential functions in the CRF, results into a much faster inference. Another interesting comparison is with [31] and [22], as these works are also based on CRFs. Our model significantly outperforms [22] and [31] both in terms of accuracy and of running time (see Fig.4 and Table 1): due to visualization issues we do not show [31] in Fig.4 as the original paper report a time of 40 seconds to recover the depth map for a single image.

KITTI Dataset. A comparison with state of the art methods is also conducted on the KITTI dataset and the associated results are shown in Table 2. As baselines we consider the work by Saxena *et al.* [27], Eigen *et al.* [6], Liu *et al.* [22], Zhou *et al.* [38], Garg *et al.* [7], Godard *et al.* [9] and Kuznetsov *et al.* [18]. Importantly, the first four methods only employ monocular images to predict depth information, while in [7], [9] and [18] a stereo setting is considered in training and therefore these methods are not directly comparable with our approach. As shown in the table, our approach outperforms all previous methods considering a supervised setting with the exception of the recent method in [18]. With respect to [18] we obtain a lower error, while the accuracy is slightly inferior. For sake of completeness we also report the performance of previous methods considering a stereo setting. Among these methods, Kuznetsov

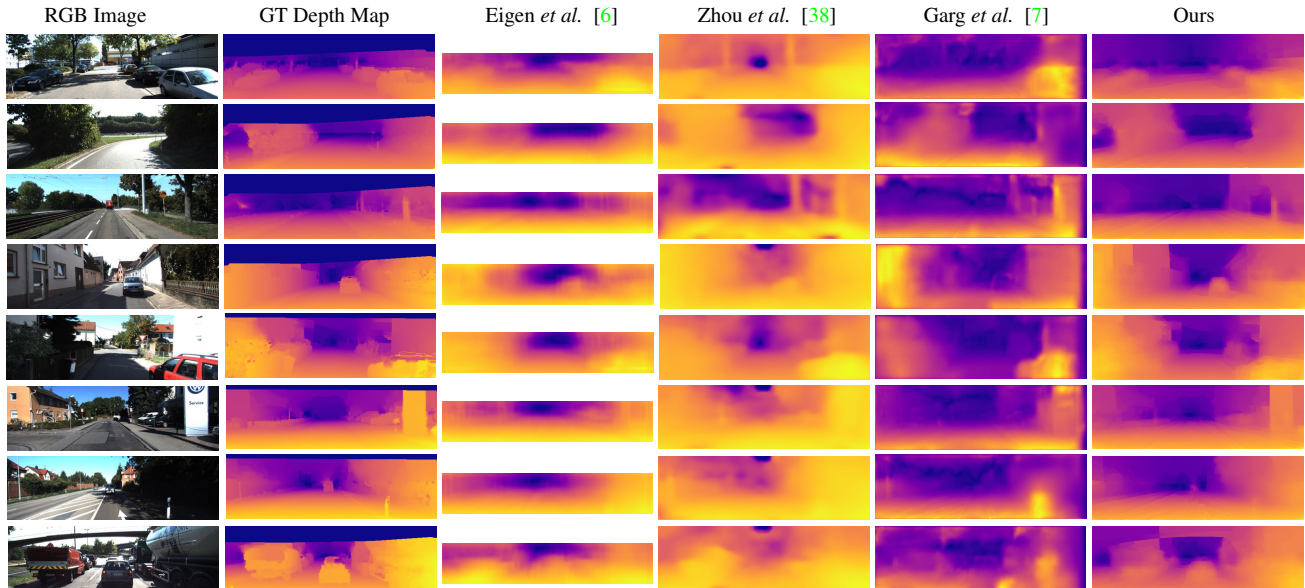


Figure 5. Examples of depth prediction results on the KITTI dataset: qualitative comparison with previous methods. The sparse ground-truth depth maps are interpolated for better visualization.

Table 3. Quantitative analysis of the main components of our method on the KITTI dataset.

Method	Error (lower is better)			Accuracy (higher is better)		
	rel	log10	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Front-end CNN (w/o multiple deep supervision)	0.168	1.072	5.101	0.741	0.932	0.981
Front-end CNN (w/ multiple deep supervision)	0.152	0.973	4.902	0.782	0.931	0.974
Multi-scale feature fusion with naive concatenation	0.143	0.949	4.825	0.795	0.939	0.978
Multi-scale feature fusion with CRFs (w/o attention model)	0.134	0.895	4.733	0.803	0.942	0.980
Multi-scale feature fusion with CRFs (w/ attention model)	0.127	0.869	4.636	0.811	0.950	0.982
Multi-scale feature fusion with CRFs (w/ structured attention model)	0.122	0.897	4.677	0.818	0.954	0.985

et al. [18] achieve the best performance by exploiting both ground truth supervision and stereo information. Following the same idea, we believe that an interesting future research direction will be to integrate stereo cues into our framework. A qualitative comparison with some state of the art methods is also shown in Fig. 5.

Ablation Study. To further demonstrate the effectiveness of the proposed method we conduct an ablation study on the KITTI dataset. Table 3 shows the results of our analysis. In the table, “multiple deep supervision” refers to training the front-end CNN with the approach in [32]; “w/ attention model” refers to considering attention variables a_s^i in the optimization but discarding the structured potential; “w/ structured attention model” indicates the using of the structured attention model. In line with findings from previous works [31, 36, 22], embedding a CRFs model into a deep architecture provides a significant improvement in terms of performance. Furthermore, adopting a CRFs is an extremely effective strategy for combining multi-scale features, as it is evident when comparing our results with CRF and those corresponding to naive feature concatenation. Finally and more importantly, by introducing the proposed CRF model with an attention mechanism and, in particular,

with a structured attention one, we can significantly boost performance.

5. Conclusions

We presented a novel approach for monocular depth estimation. The main contribution of this work is a CRF model which optimally combines multi-scale information derived from the inner layers of a CNN by learning a set of latent features representations and the associated attention model. We demonstrated that by combining multi-scale information at feature-level and by adopting a structured attention mechanism, our approach significantly outperforms previous depth estimation methods based on CRF-CNN models [37, 22, 36]. Importantly, our framework can be used in combination with several CNN architectures and can be trained end-to-end. Extensive evaluation shows that our method outperforms most baselines. Future research could perform cross-domain detection tasks [34] based on the prediction of the scene depth.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (NSFC, No.U1613209). The authors would like to thank NVIDIA for GPU donation.

References

- [1] P. Buysens, A. Elmoataz, and O. Lézoray. Multiscale convolutional neural networks for vision-based classification of cells. In *ACCV*, 2012. 2
- [2] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *TCSVT*, PP(99):1–1, 2017. 2
- [3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015. 2
- [4] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *CVPR*, 2016. 2, 4
- [5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 1, 2, 5, 6, 7
- [6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 1, 2, 5, 6, 7, 8
- [7] R. Garg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 6, 7, 8
- [8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 2, 5
- [9] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, 2017. 2, 6, 7
- [10] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 1
- [11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6
- [12] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM transactions on graphics (TOG)*, 24(3):577–584, 2005. 2
- [13] S. Hong, J. Oh, H. Lee, and B. Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *CVPR*, 2016. 2, 4
- [14] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *CVPR*, 2017. 3
- [15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 6
- [16] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *TPAMI*, 36(11):2144–2158, 2014. 2, 5, 7
- [17] Y. Kim, C. Denton, L. Hoang, and A. M. Rush. Structured attention networks. *ICLR*, 2017. 2, 4
- [18] Y. Kuznetsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. *CVPR*, 2017. 1, 2, 6, 7, 8
- [19] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *CVPR*, 2014. 2, 5, 7
- [20] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *arXiv preprint arXiv:1606.00373*, 2016. 2, 5, 6, 7
- [21] B. Li, Y. Dai, and M. He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *arXiv preprint arXiv:1708.02287*, 2017. 5, 7
- [22] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015. 1, 2, 3, 5, 7, 8
- [23] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 38(10):2024–2039, 2016. 5, 6
- [24] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014. 5, 7
- [25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2
- [26] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016. 2, 5, 7
- [27] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, 2005. 7
- [28] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 76(1):53–69, 2008. 2
- [29] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 31(5):824–840, 2009. 5, 6, 7
- [30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 2, 5
- [31] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015. 1, 2, 5, 6, 7, 8
- [32] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 2, 8
- [33] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *NIPS*, 2017. 2
- [34] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe. Learning cross-modal deep representations for robust pedestrian detection. In *CVPR*, 2017. 8
- [35] D. Xu, W. Ouyang, X. Wang, and N. Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, 2018. 1
- [36] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7, 8
- [37] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 4, 8
- [38] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *CVPR*, 2017. 2, 6, 7, 8
- [39] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015. 5, 7