Contextualized CNN for Scene-Aware Depth Estimation From Single RGB Image

Wenfeng Song, Shuai Li, Ji Liu, Aimin Hao, Qinping Zhao, and Hong Qin

***Abstract*—Directly benefited from deep learning techniques, depth estimation from single image has gained great momentum in recent years. However, most of the existing approaches treat depth prediction as an isolated problem without taking into consideration high-level semantic context information, which results in inefficient utilization of training dataset and unavoidably requires a large number of captured depth data during the training phase. To ameliorate, this paper develops a novel scene-aware contextualized convolution neural network (CCNN), which characterizes the semantic context relationship at the class-level and refines depth at the pixel-level. Our newly-proposed CCNN is built upon the intrinsic exploitation of context-dependent depth association, including inner-object continuous depth and inter-object depth change priors nearby. Specifically, rather than conducting regression on depth in single CNN, we make the first attempt to integrate both class-level and pixel-level conditional random fields (CRFs) based probabilistic graphical model into the powerful CNN framework to simultaneously learn different-level features within the same CNN layer. With our CCNN, the former model will guide the latter one to learn the contextualized RGB-Depth mapping. Hence, CCNN has desirable properties in both class-level integrity and pixel-level discrimination, which makes it ideal to share such two-level convolutional features in parallel during the end-to-end training with the commonly-used back-propagation algorithm. We conduct extensive experiments and comprehensive evaluations on public benchmarks involving various indoor and outdoor scenes, and all the experiments confirm that, our method outperforms the state-of-the-art depth estimation methods, especially for the cases where only small-scale training data are readily available.**

***Index Terms*—Depth Estimation, CNN, Single RGB Image, Contextualization, Scene-Aware Algorithm.**

I. Introduction

DEPTH estimation from single image/multiple images is a necessary process for image-based 3D reconstruction [1]. Previous works have been focusing on exploiting geometric priors or additional information sources based on sufficient observations. So far, there still exist two main challenges in depth estimation. First, in the case of multiple images, the observations are usually required to be captured from multiple views under various lighting conditions. Second, in the case of monocular single image, depth estimation is hard because its depth distribution closely relates to the scene content.

To grapple with the challenges, many works [2], [3] attempt to estimate depth from single image via carefully designing hand-crafted features, including texton, GIST, SIFT, HOG, object bank, etc. Although these methods have achieved great success in certain scene types, their ability of generalization on arbitrary scene is still limited. Meanwhile, during the last few years, convolution neural networks (CNNs) have achieved great success in both semantic-level recognition and pixel-level processing tasks, such as classification [4], [5], and semantic segmentation [6]–[8]. Some works take depth estimation as a pixel-level processing task [1], [9], [10], wherein the monocular depth estimation is usually obtained via regressing the RGB-Depth mapping based on deep learning methods. With sufficient training datasets and multiple-scale networks [9], such regression can learn more information than that from hand-crafted features. However, these methods inevitably tend to ignore the fine-gained details, such as the corners and the edges. Some recent works attempt to further deal with the challenges in two aspects: preserving the local features and extracting the contextualized features.

For local feature preservation, some works [11], [12] cast depth estimation to pixel-wise classifier learning, so as to respect the geometry details of the object but ignoring the consistence in the same object. In addition, some other works have demonstrated that, combining the context information can improve the performance. This can be achieved in different ways, for example, leveraging the continuous characteristics of the scene depth to learn a depth-fitting model by combining CRF with CNN [12], and combining the semantic segmentation with the depth estimation tasks [13]. However, such efforts on jointly

Manuscript received January 11, 2019; revised May 19, 2019; accepted September 8, 2019. Date of publication September 18, 2019; date of current version April 23, 2020. This work was supported in part by National Key R&D Program of China (2017YFB1002602), in part by National Key R&D Program of China (2017YFF0106407), in part by National Natural Science Foundation of China (61672077 and 61532002), in part by Applied Basic Research Program of Qingdao (161013xx), in part by National Science Foundation of USA (IIS-0949467, IIS-1047715, IIS-1715985, and IIS-1049448), in part by Capital Health Research and Development of Special 2016-1-4011, in part by Fundamental Research Funds for the Central Universities, and in part by Beijing Natural Science Foundation-Haidian Primitive Innovation Joint Fund (L182016). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mohammed Daoudi. *(Corresponding authors: Shuai Li; Hong Qin.)*

W. Song, J. Liu, A. Hao, and Q. Zhao are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China (e-mail: songwenfenga@gmail.com; liujiu@buaa.edu.cn; ham@buaa.edu.cn; zhaoqp@buaa.edu.cn).

S. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with the Beihang University Qingdao Research Institute, Qingdao 266000, China (e-mail: lishuai@buaa.edu.cn).

H. Qin is with the Stony Brook University, Stony Brook, NY 11794 USA (e-mail: qin@cs.stonybrook.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material includes more demonstrations about the equations and visualization results. This material is 53 M in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2941776

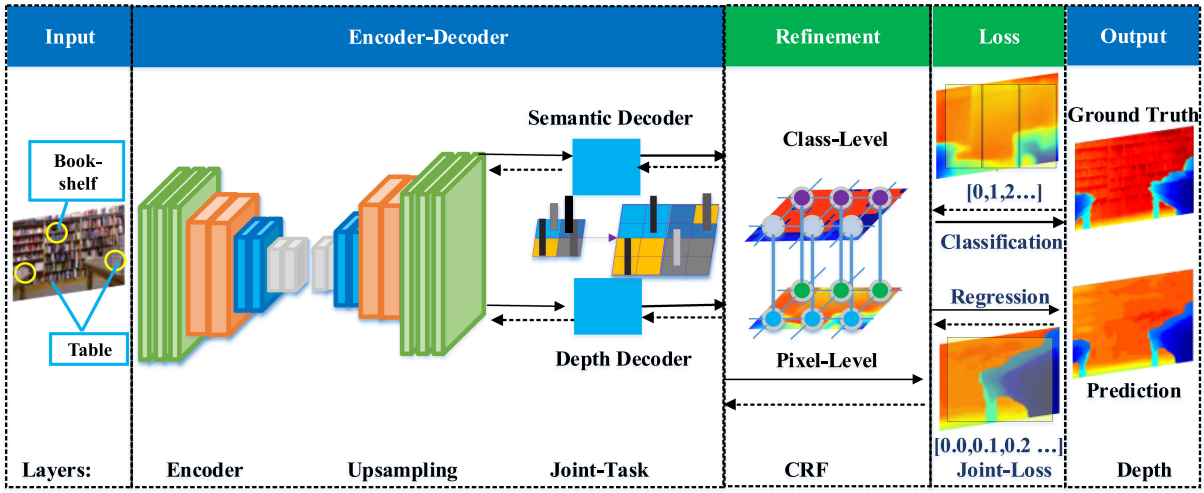


Fig. 1. The architecture of our CCNN. The solid arrow means the forward propagation and the dotted line means the back propagation. The input image is first encoded into a feature map, and then it is fed into two paths: 'semantic decoder' is used to up-sample the semantic class labels, and 'depth decoder' predicts the depth map. Then, the two paths are incorporated into the same CRF layer for class-level and pixel-level refinements. The CRF layer outputs a probability map for the joint-loss layer.

tackling the semantics and depth estimation are still preliminary, wherein they only coarsely encode the region-level and pixel-level RGBD information, and thus cannot preserve geometry details well. Taking into account the flexibility and representation power of pixel-level classification and the smoothness property of class-level regression, we cast it as a regression and classification task to preserve details of the shallow features, while estimating the depth with float precision.

As contextualized class-level feature extraction, most existing works ignore the semantic relations among different objects. However, in fact, human vision system can successfully handle the depth estimation based on the context of objects. Accordingly, we observe that, the range and change of certain object depth are associated with the semantic instances. Hence, the instances of the certain classes should be encoded into the CRF layer to integrate the semantic context information at the class-level.

Motivated by the aforementioned observations, this paper will focus on instinct contextualized relations of instances for scene-aware depth estimation by integrating the deep neural network architecture with our designed contextualized CRF. As shown in Fig. 1, our method can incorporate semantic segmentation, class-level depth estimation, and pixel-level smoothing tasks into depth estimation. Specifically, the salient contributions of this paper can be summarized as follows.

- We pioneer a generic contextualized convolution neural network (CCNN) for scene-aware depth estimation by integrating class-level and pixel-level CRF layer into the powerful CNN framework, which can well exploit the contextualized features of the scenes and further transfer the depth distributions among the similar scenes.
- We design a novel class-level refinement layer to handle depth and depth changing variance, which can better accommodate both the continuous depth changes within single object and the abrupt depth changes across neighboring objects.
- We propose an efficient joint-loss handling framework to simultaneously learn the class-level and pixel-level features within the same layer, which can jointly integrate regression and classification tasks yet involving no additional data augmentation and parameters.

II. Related Works

Depth information is a common intermediate component in understanding 3D scene structure [14]–[17]. Some of the previous works use multi-view images to extract the low-level features and conduct shape reconstruction (e.g., structure-from-motion (SFM) [18]), which usually requires capturing overlapping images from different viewpoints [19] or from temporal image sequences [20]. Some recent works start to focus on the reconstruction from single image, and many works [21] have achieved great improvement based on 3D CNNs. Specifically, inspired by the analogy of human depth perception from monocular cues, some works concentrate on monocular depth estimation [10].

So far, the methods for depth estimation from single RGB image can be roughly classified into two categories, which are briefly reviewed as follows.

The first category tends to improve the hand-crafted features [3], [22], including texton, GIST, SIFT, HOG, object bank, geometry properties and the distributions of the neighbor super-pixels. The disadvantage is that, the involved features are usually dependent on specific scene. Thus, most recent works tend to formulate depth estimation as a Markov random field (MRF) [23] or conditional random field (CRF) [22] based learning problem. These methods manage to learn the parameters of MRF/CRF from a training set of monocular images and the corresponding ground-truth depth images. Then the depth estimation problem is formulated as a maximum a posteriori (MAP) inference problem based on the CRF model. Ladicky *et al.* [24] showed how to integrate semantic labels with monocular depth features to improve performance, however, their method heavily relied on handcrafted features and super-pixels based segmentation. Karsch *et al.* [3] attempted to produce more consistent image-level predictions. They learned depths from the most suitable candidates of the training images at the whole image level.

Nevertheless, the drawback of this approach is that, it relies on the quality of available training dataset and requires to look up the entire dataset at testing time.

The second category relates to data-driven methods, most of which are based on complex CNN architecture and large scale well-labelled ground truth of depth images. For example, Eigen *et al.* [9], [25] shows that it is possible to estimate dense pixel depth using a two-scale deep network trained on RGB-Depth images. Meanwhile, they also show the limitations of single scale CNN based depth regression. Liu *et al.* [22] proposed a discrete-continuous CRF model to take into account the relations between adjacent super-pixels. They need to use approximation methods for inference. Besides, their method relies on image retrieval to obtain a reasonable initialization at first. Several works further improved such methods by replacing regression based loss with classification based on [11], introducing more robust loss functions [10], [26], [27], incorporating strong scene priors [28]–[33], and using local geometric structure [34]–[36]. However, these approaches still require high-quality pixel-aligned ground truth of depth at training phase.

On the other hand, the works combining CRF and CNN within efficient structure show promising results. Liu *et al.* [22] integrated CRF with CNN based on the super-pixel features, wherein the close-form solution of log-likelihood optimization can be directly solved using back propagation. Li *et al.* [14] designed a DCNN model to learn the mapping from multi-scale patches to depth/surface normal values at super-pixel level, and the estimated super-pixel depth/surface normal can be further refined to the pixel level by exploiting various potentials on the depth/surface normal map, including a data term, a smoothness term among super-pixels and an auto-regression term characterizing the local structure of the estimation map. Wang *et al.* [13] proposed an unified approach to jointly estimate the depth and semantic labels from single image with a hierarchical CRF, which embedded the potentials derived from a global CNN and a local regional CNN. Laina *et al.* [10] also employed CNN for depth estimation. Different from previous works, they improved the typical fully-connected layers by using a fully convolutional model to efficiently incorporate residual up-sampling blocks to solve the high-dimensional regression problems. Recently some works [37]–[46] combined CNN with PGM by taking advantage of the relationship between depth and objects. Despite the limited success of CNN/PGM based depth estimation methods, it is still under-developed in aspects of how to combine them to intrinsically model the continuous depth and semantic labels. Specially, to train such a large network, an extremely large scale (i.e. hundreds of thousands of images) training dataset with well-labelled RGB-Depth image pairs are required. In contrast, this paper will exploit the sharing features from the semantic labels, contextual relations and depth information based on a contextualized Bayesian segmentation network, and leverage them to conduct depth estimation.

III. Overview of CCNN Framework

For pixel-level depth estimation, conventional works [25] commonly rely on up-sampling operation, wherein they usually up-sample the feature maps via bilinear interpolations. However, this operation may lead to losing local details. Since the depth values within the same object are in fact continuous, we only need to know the discrete depth and its changing variance. Meanwhile, based on the fact that depth feature map only contains sparse information, we propose a two-step up-sampling scheme, which first coarsely predicts the semantic labels and depth then finely refines the depth result by a two-level CRF with the contextualized semantic relationships.

Architecture of CCNN: The pipeline of our CCNN is shown in Fig. 1, of which, we use boxes with different colors to distinguish the CNN layers from our newly-designed layers, including the CRF layer and the joint-loss layers.

A FCN-based backbone network is first trained to encode the RGB image. The input data is RGB image with semantic class labels and depth labels, and the RGB images are encoded into high-level semantic feature maps with a backbone convolution network. In this network, we decode the feature map (whose size is smaller compared with the original RGB image) to make it have the same size as the original-resolution depth feature map. Afterwards, the feature map will be fed into two paths: 'semantic decoder' is used to up-sample the semantic class labels; 'depth decoder' predicts the depth map. Then, the two paths are incorporated into the CRF layer for class-level and pixel-level refinement. The CRF layer establishes the contextualized relationship based on semantic objects, which encodes the depth distribution within single object and across different objects. Finally, a joint-loss layer is employed to model the depth estimation process by simultaneously integrating the classification and regression tasks.

Class-Level and Pixel-Level CRF: In order to decode the feature map with more detailed local instance features and global context clues, we further employ the aforementioned two-level CRF to refine the depth estimation. In this way, the depth within single object region can well preserve the local details, and in global scope the depth can be estimated with the contextual relations. Furthermore, to keep the depth of certain point being consistent with its neighboring points, it still needs to refine the depth changing among the point and its neighboring points. Therefore, we constrain the variance of the depth changing to be close to the variance of the ground truth depth changing as much as possible.

Depth-Changing Refinement: Motivated by the Taylor's formula approximation method, in order to make the error being stable and smaller, we employ the depth-changing variance (gradient) to serve as the depth change principle. It is defined as the output of g, which is the gradient operator on the feature maps output by the class-level and pixel-level CRF. We encourage it to be close to the ground truth in the gradient domain to preserve the local depth changes, which should make the depth estimation result more accurate at each point.

IV. Notations and Definitions

In a rigid math manner, we formulate the depth estimation as a CRF model. We model the both of the input images and the labels domain as two random fields to explicitly encode the unary and pairwise relations between the pixels. Consider a random field $\mathbf{x}$ defined over a set of variables $x_1, \ldots, x_N$, the domain of each variable is a set of labels. Given a random field I defined over variables $I_1, \ldots, I_N$, in our settings, I ranges over possible input images with size n, and $\mathbf{x}$ ranges over pixel-level labels of image I. In our CRF based CNN model, C_g is the set of all

unary and pairwise cliques. Let d be a vector of continuous depth values corresponding to all of the n pixels in image I. Similar to the conventional CRF, we model the conditional probability distribution of the data with the following density function:

$$Q(d|I) = \frac{1}{Z(I)} \exp\left(-\sum_{c \in C_g} E(d_c|I)\right). \quad (1)$$

Here E is the energy function, $Z(\cdot)$ is the partition function defined as: $Z(I) = \sum_c \exp(-E(d_c|I))$. Since d is continuous, the integral in Eq. (1) can be analytically calculated under certain circumstances, which will be shown in Section V-D. This is different from the discrete case, wherein certain approximation method needs to be applied. To predict the depth of a new image, we only need to solve the maximum a posteriori (MAP) inference problem by mean field approximation, let d^* denote predicted depth:

$$d^* = \arg\max_d Q(d|I). \quad (2)$$

Thus, based on Eq. (1), we can encode the class-level and the pixel-level context relationship as the condition probability of the CRF. Furthermore, we propose a joint-task loss handling method to efficiently approximate Eq. (2), which can organically combine regression, classification, and context-aware depth learning.

We first define the 'class-level CRF' and the 'pixel-level CRF' in a rigorous way. In order to make the definition and explanation technically continuous, we define the 'class-level CRF' in two stages. At first, we define 'class-level label', and then define 'class-level CRF'.

Class-level label: the label corresponds to the semantic category of certain object in RGB image, of which, it does not require the fine-grained labels for the objects in the same category. Here we use the semantic labels in NYUDV2 dataset and Make3D datasets. It is obvious that, few objects of the same class are occluded in each image, thus, the semantic labels are enough to distinguish different objects.

Class-level CRF: CRF could be defined on the mean field of the class-level labels, which facilitates to smooth the depth values within the same-class instances, while maintaining the depth changes among different instances. To better estimate the depth of different instances, the depth maps are simultaneously refined by the spatial distances and depth-changing variances, which is detailed in Section V-A. In fact, the class-level CRF controls the overall depth distributions. We further define and analyze the class labels in Section V-A.

Pixel-level CRF: it is used to represent the details of the object's appearance information, so that we can embed the details into our CCNN.

Based on pixel-level and class-level CRF based on the two-level CRF, the local depth in the single object is accurate in average. However, it is hard to guarantee depth information accurate among the object-object boundaries without the cross-object context clues. And the depth in the single object tends to have little variance and be overly smoothed due to lacking detailed depth gradient information. Therefore, with the refinement in CRF layer, it is not enough to keep the depth gradient accurate at the single point due to lacking the depth difference refinement among neighboring points. To this end, we leverage the depth changing variance to conduct further refinement.

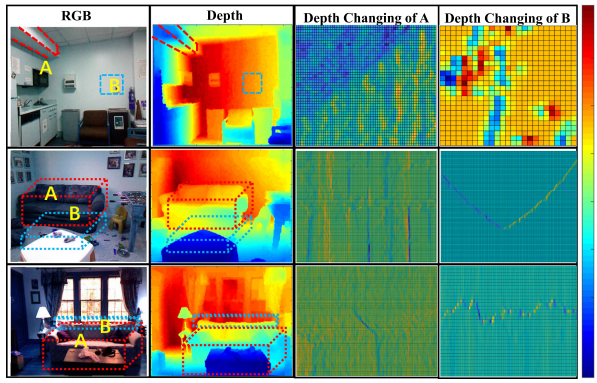


Fig. 2. Illustrations on the depth changing variance. Each image has two regions to be analyzed. Region A has a single instance, while region B has at least two neighboring objects. Region A has gradually changing depth, region B has a large gradient between different instances.

Depth-changing variance: it is defined as the statistic distribution of depth gradient (changing) at class-level. For objects of the same class in different scenes, the depth changing principles are similar, as shown in Fig. 2. Motivated by Talyer's approximation method, in order to keep the estimation error within a smaller scale, we introduce depth-changing variance to serve as the depth changing principle. 'Depth-changing variance' is detailed described in Section V-B.

V. Contextualized Scene-Aware CNN Model

In this section, to simultaneously learn the global instance relations and the local geometry features in the corresponding scene, we derive a two-level new CRF function, one is for class-level context information aggression, and the other is for pixel-level depth changing variance refinement. On that basis, we further introduce a novel CNN implementation for mean field iterations in the two-level CRF. All the critical symbols are listed in Table I.

A. Class-Level and Pixel-Level CRF

For each pixel $i \in I$, we instantiate $x_i \in \mathbf{x}$ in Section IV as $x_i = (d_i, l_i)$ to denote the inference depth and semantic class at pixel i. Given features extracted by CNN, CRF layer can produce a distribution over the depth label assignment x_i. The CRF layer has two energy functions: unary and pairwise potentials. The unary potential enables to incorporate both low-level features (such as RGB pixel values, shapes, and edges) and high-level features (including semantic features and the distribution relationship). The pairwise potentials are used to describe the depth relationship among different objects and positions. And the potentials are provided by feature maps extracted by CNN, wherein the CNN learns the depth changing principles in local-global way via a probabilistic graph defined by semantic distance. Considering the semantic objects, class-level depth refinement and the depth-changing variance, we then define energy function to make the depth map being consistent with the CNN predictions.

TABLE I
KEY NOTATIONS LIST

Symbol	Representation
x_N	A set of labels for Image I, N is the number of the classes
d, d', I	The predicted depth, the ground truth depth, and its corresponding RGB image
d_i, I_i	The predicted depth and its corresponding RGB image at pixel i
$E, E_c, E_p(I;\theta)$	Overall, pixel-level, and class-level energy functions
g	Element-wise gradient operator
θ	Parameters of the CCNN model, including the CNN layers and CRF layer
r	Semantic regions with the same class labels
Q, C	CNN function (without CRF) and CCNN function (with CRF)
U, P	Unary/Pairwise potential function of CRF

Let $E(I;\theta)$ denote the energy function of the two-level CRF,

$$E(I;\theta) = E_p(I;\theta) + E_c(I;\theta). \tag{3}$$

Here, $E_p(I;\theta)$ denotes the pixel-level feature map from the intermediate CNN layer (which mainly extracts the object appearance features in the RGB image). Meanwhile, $E_c(I;\theta)$ denotes the class-level energy function, which refines the depth distance based on the semantic instance regions. It is designed to force the predicted depth to follow the laws of the depth changes in the image. θ is the parameter of the CNN and CRF model. The first term of Eq. (3) is calculated via softmax loss. The second term of Eq. (3) is calculated via the joint-loss and will be further described in Section V-B.

The depth feature map is refined finely smooth in the same object, while refined the distance based on the semantic labels in pixel-level. Thus, the depth variance should be small for the instances that are close in semantic and spatial relations, while it should be large across the instances that are irrelevant in semantic and spatial relations.

B. Depth-Changing Constraint for CRF

To implement the two-level CRF with the two potential functions' constraint, we further formulate the E_c as two concrete variables: the distance between objects in depth and the gradient constraint for depth.

$$E_c(I;\theta) = E_d(I;\theta) + E_g(I;\theta). \tag{4}$$

Here, $E_d(I;\theta)$ encodes the depth distance based on the semantic instance regions and $E_g(I;\theta)$ is designed to force the predicted depth to follow the depth changes in the image.

We then convert the image into a graph based on the pixels' spatial distances and their semantic relations by merging the similar pixels inside the same region. The region r serves as a node, the edge of the graph encodes the region-to-region relationship. In the predicted feature map obtained from CNN, the potential energy measures the depth distance and the depth similarity of the neighboring pixels based on the semantic instances. It predicts the depth range within certain object and the distances of objects, for example, the bed in the indoor scene will have a relative large depth range (about 2.1 meters). And the lamp of the bed will have a high probability at the end of the bed, with 0.1 meter as depth range. The depth is inferred from the relation graph involving semantic labels, spatial location, and the depth. This term tends to make pixels to be assigned with different labels if they belong to different objects.

$$E_d(I;\theta) = \sum_{r=1}^{R} \parallel Rad(I_r) - Rad(C_d(I_r)) \parallel + \lambda_{de} \sum_{r\neq t} \parallel D(d'_r, d'_t) - D(C_d(I_r), C_d(I_t)) \parallel . \tag{5}$$

Here, $Rad(I_r)$ represents the depth range of the region r. I_r denotes the ground truth value of the pixel in region r. Especially, r, t denote the object region set obtained from the CNN-predicted semantic labels, $C_d(\cdot)$ means the convolution function defined by CCNN model (CRF layer included), and $D(\cdot)$ represents the Euclidean distance. The first term in Eq. (5) makes the depth of the same object relatively stable and fixed, the second one enforces constraints on the distance between objects r and t, and the parameter λ_{de} is used to balance the two terms. $\parallel \cdot \parallel$ is $L2$ norm. At the start of iterations, the first term is more important than the second one, then the second one starts to dominate when the class-level depth is of high importance. The balance process is adaptive without any manual intervention.

The feature map is refined by semantic instance regions. Intuitively speaking, when the starting depth and the ending depth are determined, the law of depth change can be refined as

$$E_g(I;\theta) = \sum_{i=1}^{n} \parallel g(C_d(I_i;\theta)) - g(d'_i) \parallel + \lambda_{ge} \sum_{i\neq j} \parallel g(C_d(I_i, I_j;\theta)) - g(d'_i, d'_j) \parallel . \tag{6}$$

Here, g denotes the gradient operation on the input data. $g(d'_i)$ denotes the gradient operation on the ground truth data. $g(C_d(I_i, I_j;\theta)$ describes the depth gradient in different instances, wherein $C_d(,)$ denotes the predicted depth map from the last convolution layer. The '$-$' is implemented by the element-wise minus operator in both two terms. The first term refines the depth-changing variance within object. The second one refines the depth change across different instances' pixels I_i and I_j. This term encourages the depth of object regions to change smoothly in the pixel level.

C. Optimization of Class-Level and Pixel-Level CRF

The perception for depth feature map built by CNN has fixed-size neighboring nodes, which contains limited context information. Intuitively speaking, the fixed-size window also includes irrelevant information from different objects that are

neighbors in spatial space. Thus, the context information extracted by CNN is not enough for depth estimation. In fact, the depth image usually involves more regular depth changes for certain-class objects w.r.t RGB image. The class-level refinement is inspired by the observation that, the depth in the objects of the same classes may change more similarly than that among irrelevant objects. We integrate depth distance and gradient changes both at class-level and pixel-level.

To take full advantage of the contextualized information in certain scenes, we should estimate the semantic objects and the relations between depth and objects. Therefore, we first run an encoder-decoder network on RGB images to obtain the CNN prediction map. Then the class-level predictions are fed into the depth decoder. With the depth map from CNN, we compute the probability map for each pixel by finding the maximal state from candidate depth labels, given the semantic labels. Thus, the first energy term in Eq. (3) can be further described as:

$$E_p(\mathbf{x};\theta) = \sum_{i \in C_g} \psi_u(x_i;\theta) + \sum_{j \in C_g} \phi_u(x_j;\theta), \tag{7}$$

where $\psi_u(x_i)$ describes the unary potential of semantics and depth in class level, $\phi_u(x;\theta)$ represents the Gibbs energy of the label assignment $x \in L$ based on the gradient and original feature maps activated by the depth estimation task. Moreover, if CNN produces different predictions for neighboring pixels, we encourage them to be consistent with the spatial and semantic relations.

Based on Eq. (5) and Eq. (6), we can further formulate class-level relations via the pair-wise potential function as

$$P(\mathbf{x};\theta) = \sum_{i \neq j} \mu(l_i, l_j) k(C_i, C_j) k(g_i, g_j), \tag{8}$$

where $k(\cdot)$ denotes the filter kernel defined to smooth the pair-wise nodes, C_i denotes the CNN output at pixel i, g_i is the gradient related to the CNN feature at pixel i, μ is a label compatibility function. Given an image I, the aforementioned semantic class-level prediction function can provide a class-level depth feature map.

As for the prediction of the joint depth-semantic label, we leverage two kernel potentials defined in terms of color vectors I_i, I_j, spatial positions p_i, p_j, and labels l_i, l_j.

$$k(i,j) = \omega^{(1)} \exp\left(-\frac{|p_i - p_j|}{2\theta_\alpha^2} - \frac{|I_i - I_j|}{2\theta_\beta^2}\right) + \omega^{(2)} \exp\left(-\frac{|l_i - l_j|^2}{2\theta_\gamma^2} - \frac{|p_i - p_j|}{2\theta_\alpha^2}\right), \tag{9}$$

where the kernel of position (i, j) is used to measure the pixel-level feature, which will change the predicted value, $\omega^{(\cdot)}$ denotes the weight of the two kernels, θ_α, θ_β, and θ_γ represent the variances of the distances, including RGB, position, and the labels of semantics and depth. Eq. (9) smoothes the value in neighboring regions, and sharpens the depth values with long semantic distance.

In order to model the contextualized instance clique, instead of calculating the marginalization with regard to $p(x_i)$, we propose to construct the convolutional neural networks and directly learn the messages. As shown in Fig. 1, we design a CRF layer, which is combined with RGB, semantic and depth channels both in the estimated probability map and in the context feature map, to capture the context pattern and geometry details at two levels.

The unary of the Gibbs energy of Eq. (7) can be defined as:

$$U(I;\theta) = U[Q(I_i), g(Q(I_i)), x_i, g(x_i);\theta)], \tag{10}$$

where the unary function $U(\cdot)$ is implemented with a CNN model, which is trained to learn the depth biasing towards the ground truth and to refine the changing laws of class-level depth to be the same as that of ground truth, $Q(I_i)$ and $g(Q(I_i))$ are the output of the backbone CNN layer: depth feature map and its corresponding gradient in depth feature map, $g(x_i)$ represents the class-level object's depth gradient operation.

D. Joint-Task Loss Handling

In this section, we describe how to train CNN to predict the depth with semantic labels in a coarse-to-fine manner. Specifically, we convert the depth estimations task as a classification task and refinement of the depth estimation as a regression task.

As shown in Fig. 1, we find that, the depth within certain objects changes continuously, and the potential depth range in an object is relatively stable. Thus, Eq. (1) is formulated as follows in order to maximize the loss of Eq. (5), and Eq. (6).

$$E_{crf} = \| \mathbf{Wd} - \mathbf{d}' \|_2^2 + \| \mathbf{RWd} \|_2^2 + \| \mathbf{Gd} - \mathbf{Gd}' \|_2^2, \tag{11}$$

where $\mathbf{d}$ is the output of the CNN network with batch size B, $\mathbf{W}$ is the weight of CNN that corresponds to semantic objects from the entire set, $\mathbf{R}$ expresses the neighboring relationship in the context instance level, while $\mathbf{G}$ is the weight of gradient regression model. When combined with the Softmax loss from backbone CNN, the joint-task loss is defined as,

$$\theta^* = \arg\min_\theta \sum_I \sum_i \log Q(d_i|I;\theta) + E_{crf}(I;\theta) + \frac{\lambda}{2}\|\theta\|_2^2. \tag{12}$$

Here $\log Q(x_i, d_c|I;\theta)$ is the Softmax loss to maximize the posterior probability of the depth, given the features extracted by CNN. d_i is the predicted label. It measures the depth estimation error. The second term of Eq. (12) forces the predicted depth feature maps to follow the changing laws between neighboring pixels and class objects, which is modeled as $L2$ norm based regression to minimize the energies. The third term is the regulation term to avoid over-fitting problem. In our experiment, we empirically set $\lambda = 0.5$, which can make good balance and has been proven effective in Section VI-B2. The softmax and $L2$ norm are embedded in CRF layer in Eq. (5) and Eq. (6).

To optimize the energy function E_c (Eq. (3)), we employ the mean-field approximation [6] to solve this problem. In order to be consistent with the CNN, we implement the CRF message passing based on Gaussian convolution (defined in Eq. (9)) in the feature space:

$$Q_i(x_i) = \frac{1}{Z_i} \exp\left\{-\phi_u(x_i) - \sum_{i \neq j} \mu(l_i, l_j) \sum_{m=1}^{2} \omega^{(m)} \sum_{i \neq j} k^m(i,j) Q_j(x_i, x_j)\right\}. \tag{13}$$

TABLE II
NETWORK STRUCTURE DESCRIPTION. U: UP-SAMPLING, C: CONVOLUTIONAL LAYER, P: POOLING

Net Structure	Layer	1	2	3	4	5	6	7
Bayesian Network	Feature Map Size	(180,240)	(90,120)	(45,60)	(23,30)	(12,15)	(23,30)	(23,30)
Encoder	Type	CCP	CCP	CCCP	CCCP	CCCP	UCC	UCC
	Channel	64	128	256	512	512	512	512
	Kernel, Stride, Padding	Conv(3,1,1) Pooling(2,2,0)					Conv(1,1,0)	
Net Structure	Layer	8	9	10	11	12	13,14,15	
Bayesian Network	Feature Map Size	(45,60)	(90,120)	(180,240)	(360,480)	(360,480)	CRF Layer	
Decoder	Type	CC	UCCC	UCCC	UCC	CC	Softmax	
	Channel	512	512	256	128	64	L2 Norm	
	Kernel, Stride, Padding	Conv(3,1,1) Pooling(2,2,0)					(360,480)	

Algorithm 1: The tth Iteration of CCNN's Training Process

Require: The set of training images for current batch, $\mathbf{I}_n$;
Ensure: Train CCNN on the current batch, I_n;
1: Extract feature map $U(I_n;\theta)$ from the last 'conv' layer of the SegNet;
2: Refine the $U(I_n;\theta)$ via the two-level CRF $E_c(I;\theta)$, output a fine model $C_d(\cdot)$;
3: Refine the $C_d(\cdot)$ via depth changing variance $D(\cdot)$ at two levels of CRF;
4: Optimize joint-task loss via θ^* in an alternative way;
5: **return** Final predicted d^*;

Here Q_i represents the distribution of x_i. m is the order of the filters. We subtract $Q_i(x_i)$ from the convolved function, because the operation of convolution in fact sums over all variables while message passing only sums over the neighbors. Our contribution is based on the observation that, for dense CRF, filter-based approximate mean-field inference relies on Gaussian/bilateral filters, as well as the approximated semantic distances in each iteration. Unlike previous CRF based CNNs, we propose Algorithm 1 to preserve detailed geometry features.

Each iteration of step 2 and 3 in Algorithm 1 performs a series of tasks, including message passing, compatibility transform and local updating. Both the compatibility transform and the local update run in linear time, which is very efficient. The computational bottleneck is message passing. For each variable, this step requires evaluating a sum over all other variables. When passing messages from two spaces, we use two Gaussian filters, which is similar to bilateral filter. As for the three spaces, we use the message passing in a trilateral way with spatial, RGB, and semantic labels. Thus, given the optimized depth feature map, we can further optimize the depth values.

To back propagate the depth error differentials (w.r.t. its input) and the network parameters in each layer, it is straightforward to perform back-propagation algorithm through the local update, the compatibility transform and the joint-loss layer. The novelty is the message passing, for this operation, the gradient w.r.t. its input is defined as:

$$\frac{\partial E_c}{\partial \mu(x_i)} = \sum_m \omega^m(I) \frac{\partial E_c}{\partial Q}(x_i). \tag{14}$$

Here it can also be calculated by performing bilateral filter over the error differential map $\frac{\partial E_c}{\partial Q}(x_i)$.

VI. EXPERIMENT RESULTS AND EVALUATIONS

In CCNN model, we compute one of the three terms of the energy function Eq. (3) by fixing the other two. To predict the depths, our implementation is built upon the publicly-available implementation of Bayesian SegNet and CRF-as RNN, whose performance is close to the state-of-the-art ones in semantic segmentation.

We first up-sample the feature maps by simply reversing the forward and backward propagated messages of the convolution as [47]. Then we employ the dropout strategy [48] to avoid over-fitting on the single depth map. Our network structure is demonstrated in Table II. Specially, our network can well combine the pixel-level continuous depth regression and classification tasks in a coarse-to-fine manner.

Specifically, we first conduct training over the RGB images with ground-truth semantic labels using the encoder and decoder operations in Bayesian SegNet [7]. With the pre-trained semantic network labels, we build depth estimation model by considering the depth distribution of inner object and that across different objects. We further fine-tune the semantic models to train the mapping between RGB and depth, with the proposed CRF layer, which integrates the class-level refinement with the pixel-level refinement to learn context relationship of objects.

It should be noted that, we compute the last layer of the feature map with a tensor of $256 \times 360 \times 480$, and each voxel has a probability of the depth value. We use the mean field approximation to inference the depth, and normalize all the depth to a range of $[0.0, 255.0]$. In addition, to evaluate our models, we train several kinds of networks, including the network respectively with E_0 (L2 norm), E_s (softmax and L2 norm), E_1 (E_p), E_c ($E_d + E_g$), and CCNN ($E_p + E_c$). The last two terms cannot be divided into two independent ones, due to the distance and the gradient of the single objects are embedded into one CRF layer. The basic network settings are shown in Table II.

In our experiments, we firstly train semantic labels with NYU depth v2 (NYUD) dataset [49], which exploits the distribution of the entire NYUDV2 dataset. Based on Bayesian SegNet, we proposed a deeper Bayesian network, and Fig. II shows our network architecture with CRF layer and loss function, wherein the input images are resized to 360×480. Then, with the proposed network, we train our model based on two other classic datasets: NYUD and Make3D. We transfer the parameters learned from NYUDV2 dataset. Here we train from scratch for the NYUDV2 dataset, and we further fine-tune the make3D dataset with a relative small dataset, without any change about the network.

TABLE III
COMPARISON OF OUR METHOD WITH THE STATE-OF-THE-ART METHODS ON NYUDV2 DATASET. THE COMPARED VALUES ARE THOSE ORIGINALLY REPORTED BY THE AUTHORS IN THEIR PAPERS

NYUDV2	ABS rel	RMSE(lin)	RMSE(log)	log10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Karsch et al.[3]	0.374	1.12	-	0.134	-	-	-
Ladicky et al.[24]	-	-	-	-	0.542	0.829	0.941
Liu et al.[22]	0.335	1.06	-	0.127	-	-	-
Li et al.[14]	0.232	0.821	-	0.094	0.621	0.886	0.968
Liu et al.[12]	0.230	0.824	-	0.095	0.614	0.883	0.971
Wang et al.[13]	0.220	0.745	0.262	0.094	0.605	0.890	0.970
Eigen et al.[25]	0.215	0.907	0.285	-	0.611	0.887	0.971
Roy et al. [50]	0.187	0.744	-	0.078	-	-	-
Eigen and Fergus [9]	0.158	0.641	0.214	-	0.769	0.950	0.988
Laina et al.[10]	**0.127**	0.573	**0.195**	**0.055**	0.811	0.953	0.988
Xu et al.[31]	**0.125**	0.593	-	0.057	0.806	0.952	0.986
Zheng et al.[36]	0.257	0.915	0.305	-	0.540	0.832	0.948
$E_0(Ours)$	0.224	0.768	0.327	0.112	0.426	0.786	0.912
$E_s(Ours)$	0.176	0.524	0.292	0.085	0.734	0.984	0.995
$E_1(Ours)$	0.177	0.528	0.284	0.086	0.728	0.983	0.994
$E_2(Ours)$	0.175	0.522	0.263	0.085	0.733	0.984	0.995
CCNN (Ours)	0.159	**0.487**	0.242	0.076	**0.835**	**0.991**	**0.997**

We compare CCNN with several state-of-the-art works [3], [9], [10], [14], [22], [24], [25], [50] based on their published configurations/results. Meanwhile, we also compare our result with the mean depth image computed from the training set. We test Make3D [23] (detailed in section. VI-C) over the official split dataset with 400 images, because it requires the least training images compared with other methods [10]. Object depths are filled by using the colorization routine of the NYUD development kit. We evaluate each method using several error indicators according to prior works, including Threshold: % of d_i s.t. $\max(\frac{d_i}{d_i'}, \frac{d_i'}{d_i}) = \delta < thr$; Abs Relative difference: $\frac{1}{|T|}\sigma_{d\in T}|d - d'|/d$; RMSE(linear): $\sqrt{\frac{1}{|T|}\sigma_{d\in T}||d_i - d_i'||^2}$; RMSE(log): $\sqrt{\frac{1}{|T|}\sigma_{d\in T}||\log d_i - \log d_i'||^2}$ and $|T|$ is the number of valid pixels. $\log 10(d, d') = \frac{1}{|T|}\sum_{d\in T}|\log_{10} d - \log_{10} d'|$.

A. Evaluations on NYU Depth Dataset

The NYUDV2 dataset consists of 1449 RGBD images of indoor scenes, among which 795 are used for training and 654 for testing (we use the standard training/test split provided by the dataset). In the training process, though depth estimation needs relatively high resolutions, our method only requires a small number of training images. To the best of our knowledge, we are the first to use the smallest number of images to achieve high-resolution accuracy. It should be noted that, the higher the resolution, the harder the estimation. The original values of the depth elements are measured in meters, which are normalized for training [51]. To make comparison, we list the numbers of training images in Table IV. (The 'CROP' is to crop a subimage in the whole image to get more samples in the limited training images, but will increase the complexity for training. The "Y/N" means whether or not to use the crop. The resolution represents the output of estimation image size.)

The learning rate is initialized to 0.001, and then it will be reduced to 0.1 times every 1000 iterations, the momentum is set to be 0.9. The λs in Eq. (3) are all set to 1.0. In Eq. (9), θ_α is imperially set to 3.0 to measures the window of RGB relations of the image, θ_β, and θ_γ are set to 25, 65 based on the best result comparison. And the detailed quantitative results are shown in Table III. The results show that our method achieves the state-of-the-art in most of the metrics. Our method is even better than methods with 1000 times larger of training size [10]. Our single network architecture is more efficient than the multiple scales network [25] in all the metrics. The detailed comparison results are shown in Fig. 3.

TABLE IV
COMPARISON OF TRAINING IMAGE SIZE

Methods	Size	CROP	Resolution
Eigen et al.[25]	120k	N	147,109
Eigen et al.[9]	120k	N	147,109
Laina et al.[10]	95k	Y	304, 228
Li et al.[14]	800k	Y	304, 228
Xu et al.[31]	12K	Y	320, 240
Ours	**0.795k**	N	**360, 480**

In contrast to the prior works that only consider the mapping of RGB image and depth, we add the two-level CRF layer to refine the instance's geometry details, and thus our CCNN model can greatly improve the depth estimation results. Besides, our CCNN takes the context information and instance relationship into consideration, which can better estimate the detailed features of the single instance and the depth-changing relations between neighboring instances. For example, previous works can only estimate the rough outline of small objects, such as lamp, things on the desk, and curtain, in sharp contrast, the geometry details of such small objects can be well preserved in our CCNN framework. Furthermore, benefiting from our class-level constraint formulation, the depth values within these instances are more continuous than those estimated by the earlier work [10]. The objects are close in semantic if they have the prior knowledge to affect each other. The ID of labels are the same as those over NYUDV2 dataset [49]. It is still a challenge to reconstruct from single image, for the low accuracy of points. In order to test if our CCNN

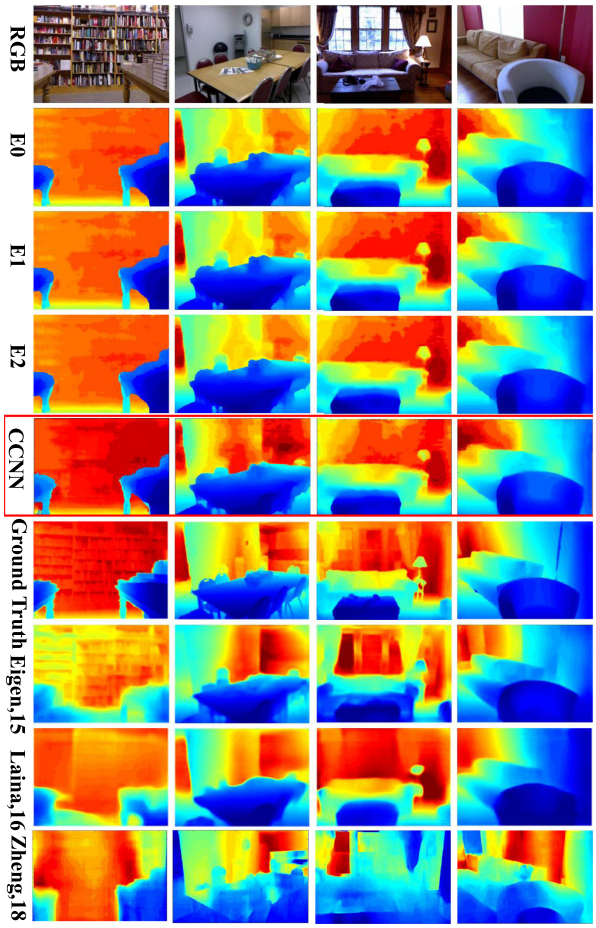


Fig. 3. Comparisons on NYUDV2 dataset. The 2-5 rows show our results, wherein our predicted depth results can well preserve the details of the lamp, the objects on the desk and the curtain.

is good enough, after we get the estimated depth, we further reconstruct the corresponding 3D objects. We segment the object with the semantic labels, and employ the alpha shape with parameter 0.5 to perform re-meshing. The reconstruction results are shown in Fig. 4. Compared with the previous work [25], the reconstructed mesh is more continuous. The results from other networks output have many distortions on geometry details, and the mesh is blurred and distorted across boundaries, yielding non-satisfactory quality. Our results can greatly alleviate such problems, giving rise to much more satisfied results.

Furthermore, we design two experiments over NYUDV2 dataset to prove our CCNN structure can benefit the complex scene and the single object with complex feature details. For complex scenes, the experiment is to reconstruct the scene with more than three objects, the scenes with people activities, and the scenes with high-contrast RGB features. The results in Fig. 3 (more results are shown in our supplementary materials) illuminate our three aspects of contributions: 1) the scene with multiple objects shows that, the depth changes continuously within the single object while abruptly across neighboring objects; 2) the scene with people activities shows our CCNN is robust to dynamic scenes; 3) the results for high-contrast RGB images show that, the gradient term has positive effects on the depth estimation across RGB channels.

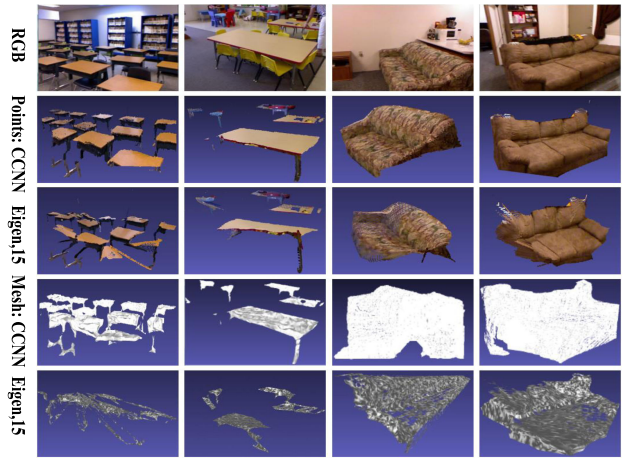


Fig. 4. 3D object reconstruction results based on the estimated depth. The 1st row shows RGB images, the 2nd and 3rd rows are point clouds, while the 4th and 5th rows show the meshes reconstructed from CCNN and Eigen *et al.* [25], respectively.

As for the single object with complex feature details, some typical examples are used to demonstrate the geometric feature-preserving ability. Our class-level and pixel-level architecture can well encourage the preservation of the geometry structures. For example, in Fig. 4 the results for single bed, sofa, bookshelf and chair prove our CCNN can benefit the 3D reconstruction from single images (More demonstrations are respectively shown in Fig. 2 of the supplement material). The beds are well structured, including the edges and corners, whose depth ranges are accurate. The sofa meshes are smooth in geometry, whose components are complete. The chairs have more components than other objects, which is hard for single image based reconstruction. However, our CCNN can still reconstruct the chairs reasonably well.

Our results are refined in class level and pixel level from local to global, which could avoid potential over-fitting in global range and preserve the detailed local geometry features. However, this process will increase the error in log10 metric, due to the trend to preserve the detailed local geometry features. Hence, our CCNN will perform slightly worse than others in log10 metric, when being tested on the NYUDV2 [49] dataset with the common settings (training on the datasets from the same sources). On the other hand, when being tested on the target dataset without training, our CCNN will perform better than other methods. We conduct experiments on VKitti [52] dataset with the model that is only trained on NYUDV2 dataset. The results in Tab. VI show the superiority of our CCNN in unknown (un-trained) scenarios. This process verifies the generalization ability of our CCNN, when being tested in unknown datasets without pre-training. (Detailed in Section VI-C)

B. Ablation Studies

To further evaluate the effects of CRF refinement and the joint-task loss, we conduct experiments by changing the controlling conditions of the main parameters in the CRF energy and the loss functions. The experiments include the following three aspects.

1) Evaluations on CRF Refinement: The novelty of our method is to take the semantic labels into consideration. So we

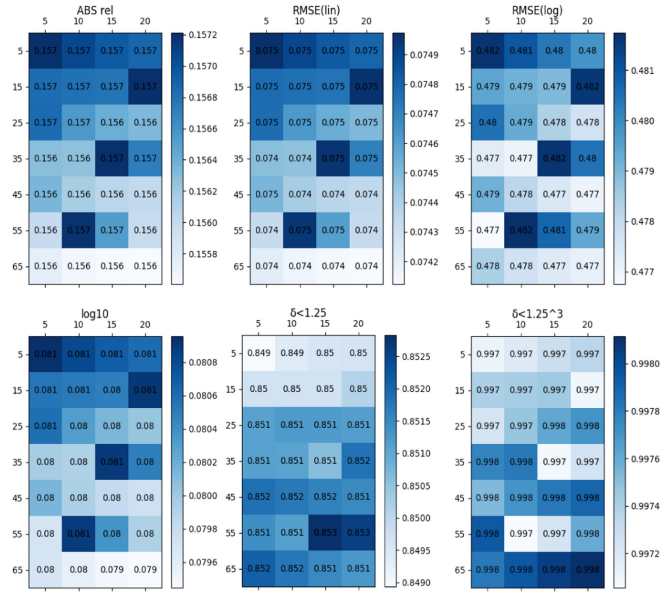


Fig. 5. The experimental results of different parameters of θ_β and θ_γ, θ_β ranges from 5 to 25, while θ_γ ranges from 5 to 65. The result becomes more and more accurate while the two parameters becomes larger and larger.

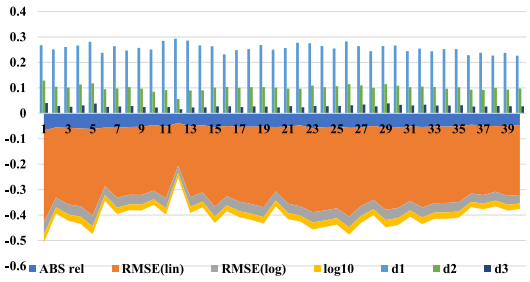


Fig. 6. Comparisons of our method with and without the semantic labels in the first 40 classes of NYUDV2 dataset. The values represent 7 metrics corresponding to the 7 column names of Table III, d1, d2, d3 respectively represent the short for the last three names.

design two experiments to demonstrate the contribution of the semantic labels. One experiment is used to evaluate different values of θ_β and θ_γ in Eq. (9). The values of θ_β range from 5 to 65, while the values of θ_γ range from 5 to 25. For each pair of parameters, we evaluate the seven metrics for each test value. In order to detect the best combinations, we use the 7*4 models. The result is shown in Fig. 5. The matrices show that, both the error metrics and the precise metrics have significant improvement as the range of the CRF message passing becomes larger. The other experiment is designed to control the proportion of semantic labels. We use 10%,. . ., 90% semantic labels to test the contribution of our CRF model, the result is shown in Fig. 7. To evaluate the improvement of each class, we design a comparative experiment to observe 40 classes' improvement benefiting from the semantic labels. The result is shown in the lower right of Fig. 6. The base line model is trained with E_0, while CCNN model is trained with all the CRF energy terms serving as the refinement of the CRF layer. The result shows that, with the newly-added CRF layer, the error of the first 40 classes is reduced from 20.7% to 49.3% , while the accuracy is

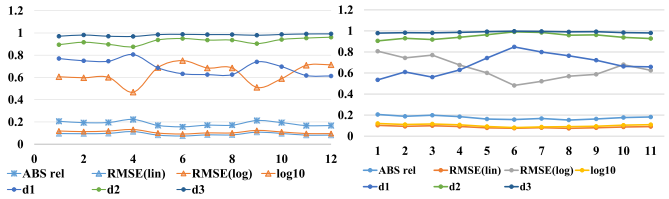


Fig. 7. Left: The experimental results produced with different proportions of semantic labels: 5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90 (%). Right: Evaluations of our method for loss-joint tasks on NYUDV2 dataset. The regression task ranges from 0 to 1 with step 0.1, while the classification task ranges from 1 to 0 with step 0.1. The values represent the 7 metrics corresponding to the 7 column names of Table III.

improved 3.2% to 30.6%. It proves that, our method is capable of robustly improving the single classes' performance. Furthermore, it is also obvious that, the proposed model gives rise to more accurate and stable depth estimation than those without considering the CRF refinement, since the objects are varied.

2) Evaluations on Joint-Task Loss: We then use two kinds of loss in a coarse-to-fine manner, on the one hand, we use the softmax loss to discretely refine the depth at 255 levels. On the other hand, we use the L2-norm to precisely train the depth at each level with float precision. To find the best combination of the two losses, we use different weights for the two kinds of loss. The curves show that weight of (0.5 classifier, 0.5 regression) works better than others. The results are shown in the lower right of Fig. 7. This indicates that the detailed features of the depth image and the coarse features have the equal contribution to the tasks.

3) Pixel Level and Class-Level CRF Evaluation: We use two experiments to give more evaluations on the two-level CRFs' contributions to the final result. Besides, in general, the class-level refines single object, while the pixel-level refines the depth distance among different objects. The contributions of the pixel-level and class-level are different for different scenes. For indoor scene, the class-level CRF is more important; as for the outdoor scene, such as Make3D and VKitti dataset, pixel-level CRF contributes more than class-level CRF. Besides, when the scene contains heavy fog, the class-level CRF will contribute more. We perform the pixel-level CRF and the class-level CRF separately on the three datasets to evaluate their contributions on indoor and outdoor scenes.

a) Separately using pixel-level CRF on the three datasets: The global depth distribution is improved on both indoor and outdoor dataset. Specially, for the outdoor dataset, we consider the appearance information when predicting the depth between different instances, which is an auxiliary clue to guide the depth estimation. Such refinement makes the depth gradient more reasonable between different objects. For example, for the objects far from camera and near camera, the range of their depth distance is more close to the ground truth than that produced by class-level CRF (more demonstrations are shown in Fig. 1 of the supplement material). The baseline network (SegNet) fails to accurately refine the depth range due to the lack of original RGB information. In addition, for the inner region of single instance, the depth predicted by the pixel-level CRF has more clear boundary and more depth changing details than the baseline network. The improvement is especially significant for the outdoor

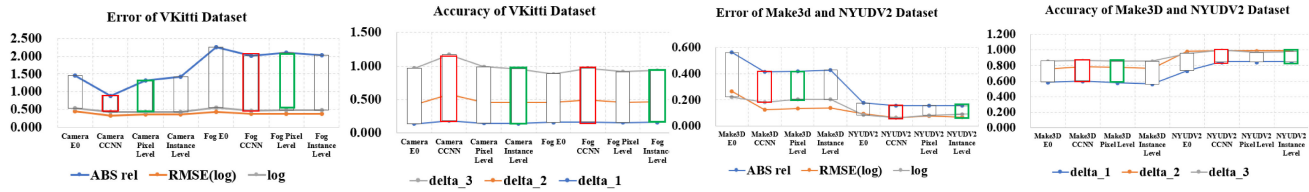


Fig. 8. The performance statistics when using pixel-level, class-level, and both levels of CRFs in CCNN on three datasets, including VKitti, Make3D dataset, and NYUDV2 datasets. Red boxes are performs best. Green boxes performs second best. It is obvious that, CCNN with two-level CRF achieves the best results.

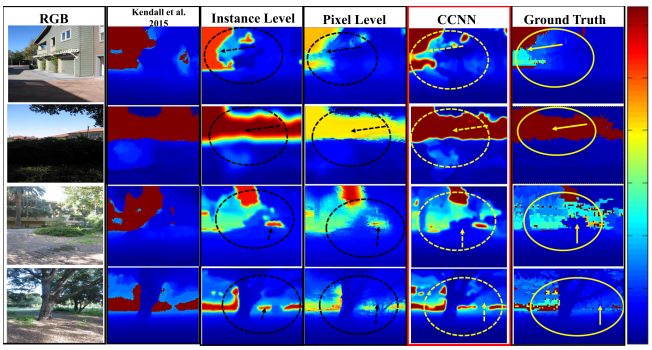


Fig. 9. Results and comparisons when using pixel-level CRF, instance-level CRF, and two levels of CRFs in CCNN on the Make3D dataset.

TABLE V
QUANTITATIVE COMPARISON ON MAKE3D DATASET

Method	ABS rel	RMSE(lin)	RMSE(log)
Karsch et al.[3]	0.417	8.172	0.144
Liu et al.[14]	0.462	9.972	0.161
Laina et al. [10]	**0.198**	**5.461**	**0.082**
Godard et al. [1]	1.000	19.11	2.527
E_g	0.53	16.17	0.241
CCNN (No Augmentation)	0.49	15.58	0.24
CCNNK	***0.44***	***14.76***	***0.228***
CCNN CCNN(VGG16)	***0.41***	***13.93***	***0.128***
CCNN(ResNet50)	***0.22***	***7.23***	***0.086***
CCNN(ResNet101)	***0.14***	***4.89***	***0.078***

scenes with a relatively-larger depth range, such as scene with trees and far-away houses. In addition, pixel-level CRF cannot perform well on heavy fog dataset, the comparison is shown in Fig. 8, due to the distraction of color distributions.

b) Separately using class-level CRF on the three datasets: The class-level refinement gives rise to accurate depth estimation in single instance, as shown in Fig. 9, more local features are preserved. Therefore, for the inner instance region, the depth is estimated continuously, which has stable gradient. Meanwhile, the depth difference at the boundary of two neighboring instances is more clear, for example, the depth between the desk and the cupboard has a clear boundary. Furthermore, the difference of the depth between the wall and the desks is larger and consistent with that of the ground truth. Class-level CRF constrains the depth continuity and integrity, which ensures the depth estimation at the across-instance boundary having a relative accurate range. Thus, class-level CRF performs better than pixel-level CRF on heavy fog dataset, as shown in Fig. 8.

c) Quantitative evaluations on the contributions of the two-level CRFs: For the NYUDV2 dataset, the results produced by class-level CRF embedded CCNN achieve a lower error and higher accuracy than those of the pixel-level CRF. The decreased error ranges from 0.004 to 0.027, while the accuracy improvement ranges from 0.009 to 0.114. For the Make3D dataset, the results produced by pixel-level CRF embedded CCNN achieves a lower error and higher accuracy than those of the class-level CRF. The decreased error ranges from 0.019 to 0.147, while the accuracy improvement ranges from 0.001 to 0.029. For the VKitti Fog dataset, the results produced by pixel-level embedded CCNN achieves a lower error and higher accuracy than those of the class-level CRF. The decreased error ranges from 0.059 to 0.231, and the accuracy improvement ranges from 0.001 to 0.029. For the VKitti Camera view dataset, the results produced by pixel-level embedded CCNN achieves a lower error and higher accuracy than those of the class-level CRF. The decreased error ranges from 0.042 to 0.571, and the accuracy improvement ranges from 0.007 to 0.107.

In summary, for indoor dataset, the class-level CRF is more important, as for the outdoor scene such as Make3D and VKitti dataset, the pixel-level CRF contributes more.

4) Efficiency Evaluations: Our model further improves the speed of depth estimation benefiting from the GPU implementation of the CRF layer. Our models are tested on 4 Tesla K80 GPUs, which have 8 GPU cores. The speed of the prediction is compared with different methods and different terms. The result is shown in Table VII. It is obvious that our model does not increase much time cost compared to the other works done, benefiting from the fixed size of the message passing windows and the GPU acceleration power.

C. Evaluations on Model Generalization

In order to evaluate the generalization ability over scenes without being trained, we also evaluate our model generality on Make3D dataset [23], which includes outdoor scenes. It consists of 400 training images and 134 testing images. We resize all images to 360×480, which is a relatively higher resolution than existing networks to feed our network. The dataset is small to be trained on raw dataset. However, in order to test the generalization ability of our method, we train the images without using any data-augmented technology. To evaluate the generalization ability of our CCNN model, we fine-tune on NYUDV2 dataset. Despite the dissimilarities in content and camera parameters, we still achieve reasonable results. We also fine-tune on Kitti dataset [53], which is also an outdoor dataset. The result

TABLE VI
PERFORMANCE COMPARISON ON THE INCLINED-VIEW IMAGES THAT NEVER SIMILARLY APPEAR IN THE TRAINING DATASET. OUR CCNN ACHIEVES BETTER PERFORMANCE IN BOTH LOCAL AND GLOBAL DEPTH ESTIMATION

Scenarios	Models	ABS rel	RMSE(lin)	RMSE(log)	log10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Inclined-View	Laina. et al.[10]	0.584	0.335	16199	0.467	0.188	0.39	0.587
Inclined-View	CCNN	0.583	0.33	16198.4	0.45	0.18	0.395	0.593
Inclined-View	CCNNK	**0.582**	**0.32**	**16202**	**0.45**	**0.1846**	**0.389**	**0.593**
Heavy Fog	Laina. et al.[10]	0.691	0.328	18228	0.432	0.21	0.42	0.512
Heavy Fog	CCNN	0.532	0.301	18219	0.485	0.23	0.45	0.558
Heavy Fog	CCNNK	**0.489**	**0.26**	**18528**	**0.445**	**0.24**	**0.46**	**0.592**

is better than that over the indoor dataset, as shown in Table V. The predicted depth is reasonable based on the 400 training images. With the CRF layer, the result is even better than the whole network, and it achieves the best result when transferring from the Kitti dataset instead of the NYUDV2 dataset. Here CCNNK represents the CCNN model that is finely tuned on Kitti dataset. Since Kitti dataset mainly relates to outdoor scenes, it has closer relations with make3d dataset. The two datasets share more common classes with respect to NYUDV2 dataset, including cars, grass, trees, roads, etc. The result shows that, our CCNN model has better transfer ability for an unknown dataset that is small scaled, especially when the pre-trained dataset is correlated with the testing dataset. Compared with the method [1], CCNN improves in all the metrics, especially the RMSE(log) error is only the 0.09% of model [1]. Compared with the completely trained model, CCNN is very close to the state-of-the-art [10] with around 15k samples, which is 400 times larger than ours. While result gets better when fine tuned on Kiiti dataset, which have more objects as outdoor dataset.

In order to demonstrate the performance, we also compare our CCNN with Laina's method by using augmented (15k images) Make3D dataset to respectively train the models. we similarly pre-process Make3D to obtain an augmented dataset (its scale is also as large as 15k). Given the augmented Make3D dataset, the results are shown in Table V, and it shows that, the performance of our CCNN has a significant improvement (the error is decreased from 0.49 to 0.41).

Besides, the backbone (VGG16) of our CCNN contains 16 layers, and its layer depth is much smaller than that of Laina's network (ResNet50), which contains 50 layers. To demonstrate that our CCNN could also reach high performance, we further use the ResNet50 and ResNet101 as our backbones. As a result, our performance is largely improved, which is better than Laina's method, shown in Fig. 12.

Moreover, our CCNN can easily accommodate hard cases with view and weather changes. To prove this, we have designed an experiment on VKitti dataset [52]. Given the same scenes, for the images captured under different weather or at different time, our CCNN can well handle such complex cases. Based on the CCNN trained on the Make3D dataset, it can also be used to estimate the scene depth on the outdoor VKitti dataset, including two representative kinds of scenes: (1) inclined-view images; (2) scene images with heavy fog (there are no similar images in the testing dataset). The results show our method is better than most of the existing methods. The detailed quantitative comparisons are documented in Table VI, Fig. 10 and Fig. 11.

TABLE VII
EFFICIENCY COMPARISONS

Model	CCNN	Eigen et al.[9]	Laina et al.[10]	Zheng et al.[36]
Speed(ms)	**624.24**	57679.92	11623.11	4116.23

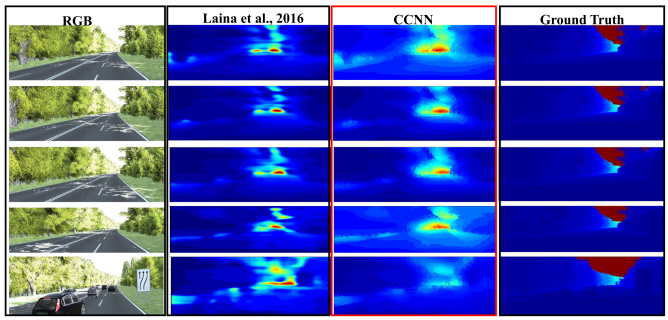


Fig. 10. Depth estimation results over the inclined-view images from VKitti dataset. Our CCNN gives rise to proper depth range estimation within single object, and the depth distribution is reasonable on the across-object boundary.

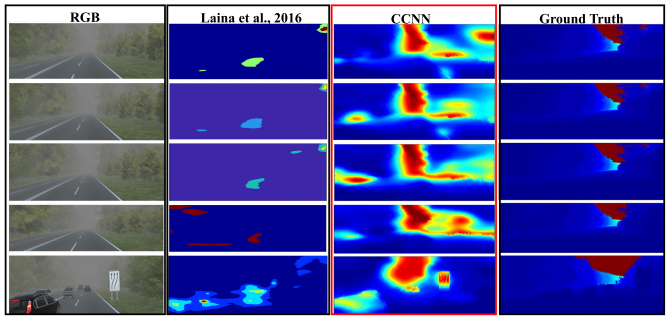


Fig. 11. Depth estimation results over the images of fog scene from VKitti dataset. Laina's method tends to predict all the depth values within a small range. In contrast, our CCNN gives rise to more proper depth range estimation within single object and more reasonable depth changes between different objects than Laina's method.

In summary, our CCNN can transfer the learned knowledge to the unknown dataset, specially, when the scenes are similar, it can achieve better performances.

D. Limitations

Our approach mainly focuses on the depth estimation from single RGB image, thus, when the quality of the input image is low, the predicted depth result might be poor. Meanwhile, since our method emphasizes to leverage the semantic relations of

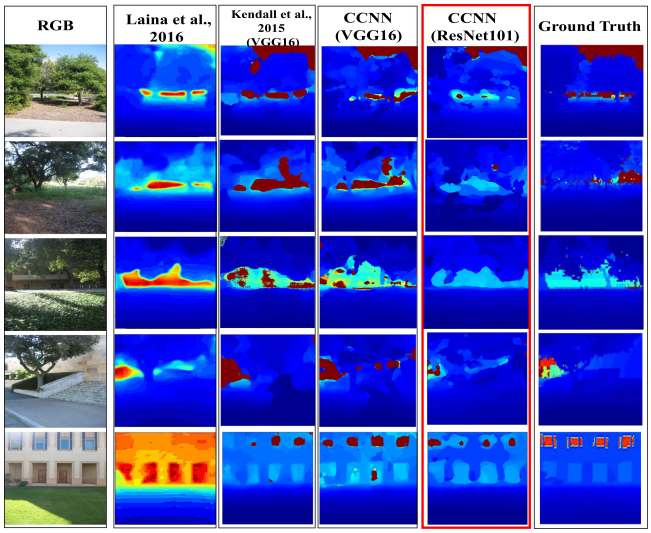


Fig. 12. Comparison between Laina's method and our CCNN over the Make3D dataset, wherein VGG16 and ResNet101 are employed as the backbones. Our CCNN with ResNet101 outperforms the state-of-the-art methods.

the scene context, when little context information could be obtained, the performance of our CCNN framework may degenerate to those of the commonly-used CNN based depth estimation methods. Besides, currently we don't consider the relations of multiple-view images, even though it might supplement more information for depth estimation.

VII. Conclusion and Future Work

In this paper we have detailed a novel approach to tackle the problem of depth estimation from single image. Unlike typical CNN-based approaches which commonly require large-scale training images, our newly-proposed CCNN offers a new probabilistic graphical model through the integration of class-level and pixel-level CRF, which not only affords training much deeper network configurations but also greatly reduces the number of required training samples. Meanwhile, comprehensive evaluations on numerous architectural components have been carried out. It manifests that, CCNN is simpler than the existing methods, can be trained with far less data, and can achieve higher-quality results at the same time. Benefited from all of the above, our method obtains the state-of-the-art results in fully-trained NYUDV2 dataset, and can transfer the pre-trained model to an untrained dataset.

As for our on-going research efforts, we would like to validate our method on more datasets, migrate our model to the mobile platform, and further improve the computational efficiency of our method. Besides, we plan to exploit more specific transfer learning ideas into our CCNN framework, so that we could more effectively leverage the existing training datasets to exploit the intrinsic information underlying the similar scenes for depth estimation.

References

[1] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. Comput. Vision Pattern Recognit.*, 2017, pp. 270–279.

[2] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. Int. Conf. Comput. Vision*, 2010, pp. 1253–1260.

[3] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 775–788.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 770–778.

[5] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 1–9.

[6] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 1529–1537.

[7] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," 2015, *arXiv:1511.02680*.

[8] B. Kang, Y. Lee, and T. Q. Nguyen, "Depth-adaptive deep neural network for semantic segmentation," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2478–2490, Sep. 2018.

[9] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 2650–2658.

[10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. Int. Conf. 3D Vision*, 2016, pp. 239–248.

[11] Y. Cao, Z. Wu, and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3174–3182, Nov. 2018.

[12] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 5162–5170.

[13] P. Wang *et al.*, "Towards unified depth and semantic prediction from a single image," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 2800–2809.

[14] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He, "Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 1119–1127.

[15] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *Int. J. Comput. Vision*, vol. 76, no. 1, pp. 53–69, 2008.

[16] C. Li, P. An, L. Shen, and K. Li, "A modified just noticeable depth difference model built in perceived depth space," *IEEE Trans. Multimedia*, vol. 21, no. 6, pp. 1464–1475, Jun. 2019.

[17] Y. Pan, R. Liu, B. Guan, Q. Du, and Z. Xiong, "Accurate depth extraction method for multiple light-coding-based depth cameras," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 685–701, Apr. 2017.

[18] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. Comput. Vision Pattern Recognit. Workshop*, 2016, pp. 16–22.

[19] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," CVPR, pp. 4040–4048. 2016.

[20] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *Proc. Comput. Vision Pattern Recognit.*, 2016, pp. 4058–4066.

[21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[22] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *Proc. Comput. Vision Pattern Recognit.*, 2014, pp. 716–723.

[23] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-D scene structure from a single still image," in *Proc. Comput. Vision Pattern Recognit.*, 2007, pp. 1–8.

[24] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. Comput. Vision Pattern Recognit.*, 2014, pp. 89–96.

[25] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.

[26] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," CVPR, pp. 2002–2011, 2018.

[27] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, "Single-image depth estimation based on Fourier domain analysis," in *Proc. Comput. Vision Pattern Recognit.*, 2018, pp. 330–339.

[28] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proc. Comput. Vision Pattern Recognit.*, 2018, pp. 2800–2810.

[29] 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 *Proc. Comput. Vision Pattern Recognit.*, 2018, pp. 675–684.

[30] X. Wang, D. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 539–547.

[31] D. Xu *et al.*, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proc. Comput. Vision Pattern Recognit.*, 2018, pp. 3917–3925.

[32] B. Chen, C. Jung, and Z. Zhang, "Variational fusion of time-of-flight and stereo data for depth estimation using edge selective joint filtering," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2882–2890, Nov. 2018.

[33] W. Dong *et al.*, "Color-guided depth recovery via joint local structural and nonlocal low-rank regularization," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 293–301, Feb. 2017.

[34] A. Chakrabarti, J. Shao, and G. Shakhnarovich, "Depth from a single image by harmonizing overcomplete local network predictions," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 2658–2666.

[35] A. Bansal, B. C. Russell, and A. K. Gupta, "Marr revisited: 2D-3D alignment via surface normal prediction," in *Proc. Comput. Vision Pattern Recognit.*, 2016, pp. 5965–5974.

[36] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 767–783.

[37] L. Wang and C. Jung, "Example-based video stereolization with foreground segmentation and depth propagation," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1905–1914, Nov. 2014.

[38] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun, "Monocular object instance segmentation and depth ordering with CNNs," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 2614–2622.

[39] E. Shelhamer, J. T. Barron, and T. Darrell, "Scene intrinsics and depth from a single image," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 235–242.

[40] T. Zhou, P. Krahenbuhl, and A. A. Efros, "Learning data-driven reflectance priors for intrinsic image decomposition," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 3469–3477.

[41] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 730–738.

[42] P. Wang *et al.*, "Surge: Surface regularized geometry estimation from a single image," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 172–180.

[43] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[44] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 109–117.

[45] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Multi-scale continuous CRFs as sequential deep networks for monocular depth estimation," in *Proc. Comput. Vision Pattern Recognit.*, 2017, pp. 5354–5362.

[46] F. Shen, R. Gan, S. Yan, and G. Zeng, "Semantic segmentation via structured patch prediction, context CRF and guidance CRF," in *Proc. Comput. Vision Pattern Recognit.*, 2017, pp. 1953–1961.

[47] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.

[48] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[49] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 746–760.

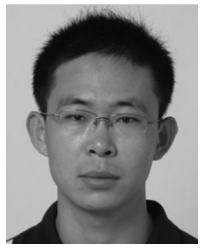
[50] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Proc. Comput. Vision Pattern Recognit.*, 2016, pp. 5506–5514.

[51] F. Guney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 4165–4175.

[52] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. Comput. Vision Pattern Recognit.*, 2016, pp. 4340–4349.

[53] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 3061–3070.

Wenfeng Song received the M.S. degree in software engineering from Beihang University, Beijing, China, in 2015. She is currently working toward the Ph.D. degree in technology of computer application from Beihang University, Beijing, China. Her research interests include pattern recognition, computer vision, and machine learning.



Shuai Li received the Ph.D. degree in computer science from Beihang University, Beijing, China. He is currently an Associate Professor with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests include computer graphics, pattern recognition, computer vision, and medical image processing.

Ji Liu received the B.S. degree in computer science from Yantai University of Technology, Yantai, China, in 2016. She is currently working toward the M.S. degree in technology of computer application from Beihang University, Beijing, China. Her research interests include pattern recognition, computer vision, and machine learning.

Aimin Hao received the B.S., M.S., and Ph.D. degrees in computer science from Beihang University, Beijing, China. He is a Professor with Computer Science School and the Associate Director of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His research interests are on virtual reality, computer simulation, computer graphics, geometric modeling, image processing, and computer vision.

Qinping Zhao received the Ph.D. degree in computer science from Nanjing University, Nanjing, China, in 1989. He is a Professor with Beihang University, a member with the China Academy of Engineering, a Chief Scientist with State Key Laboratory of Virtual Reality Technology and System, and the President with China Simulation Federation. His research interests are on virtual reality, computer simulation, and computer vision.

Hong Qin received the B.S. and M.S. degrees in computer science from Peking University, Beijing, China, and the Ph.D. degree in computer science from the University of Toronto, Toronto, ON, Canada. He is a Professor of computer science with the Department of Computer Science, Stony Brook University, Stony Brook, NY, USA. His research interests include geometric and solid modeling, graphics, physics-based modeling and simulation, computer-aided geometric design, visualization, and scientific computing.