Deep Monocular Depth Estimation via Integration of Global and Local Predictions

Youngjung Kim, *Student Member, IEEE*, Hyungjoo Jung, *Student Member, IEEE*, Dongbo Min, *Senior Member, IEEE*, and Kwanghoon Sohn, *Senior Member, IEEE*

***Abstract*— Recent works on machine learning have greatly advanced the accuracy of single image depth estimation. However, the resulting depth images are still over-smoothed and perceptually unsatisfying. This paper casts depth prediction from single image as a parametric learning problem. Specifically, we propose a deep variational model that effectively integrates heterogeneous predictions from two convolutional neural networks (CNNs), named global and local networks. They have contrasting network architecture and are designed to capture the depth information with complementary attributes. These intermediate outputs are then combined in the integration network based on the variational framework. By unrolling the optimization steps of Split Bregman iterations in the integration network, our model can be trained in an end-to-end manner. This enables one to simultaneously learn an efficient parameterization of the CNNs and hyper-parameter in the variational method. Finally, we offer a new data set of 0.22 million RGB-D images captured by Microsoft Kinect v2. Our model generates realistic and discontinuity-preserving depth prediction without involving any low-level segmentation or superpixels. Intensive experiments demonstrate the superiority of the proposed method in a range of RGB-D benchmarks, including both indoor and outdoor scenarios.**

***Index Terms*— Depth estimation, 2D-to-3D conversion, non-parametric sampling, convolutional neural networks, RGB-D database.**

I. Introduction

PREDICTING 3D structure from a single monocular image has remained an active research topic in image processing and computer vision. This can be attributed to the fact that depth information often leads to significant improvements on a number of challenging problems, such as visual odometry [1], intrinsic image decomposition [2], pose recognition [3], and scene understanding [4]. Traditional methods to depth estimation from a single image exploited various monocular cues like parallax, motion [5], or shading [6]. However, these approaches are not applicable for general scenes, due to strict constraints imposed on prediction models, e.g., translational camera motion and static scenes. Alternatively, other methods require user-annotations such as sparse depth scribbles and segmentation mask [7], [8]. The sparse scribble is then propagated through the entire image in order to fill the remaining regions with no valid depth values. While the resulting depth image is convincing by means of user intervention, providing precise scribbles is very labor-intensive and time-consuming. Meanwhile, humans have no difficulty in perceiving depth from a monocular input, thanks to the knowledge and data accumulated over the years [9]. The capability of machines to replicate this effect would open new avenues, motivating for several state-of-the-art methods [15]–[22].

Recent works on single image depth estimation can be generally categorized into two groups[1]: non-parametric sampling and parametric learning methods. The first group addresses the question of whether it would be possible to correctly transfer depth from a large RGB-D database to a single query image [10], [12], [15]. For an input query image, a set of semantically similar images are first retrieved through k-nearest neighbors (kNNs) search from the RGB-D database. They then establish dense correspondences between the input and each of the retrieved RGB images. The corresponding depth images are warped and fused using local or global optimization procedure to recover a final depth image. However, the non-parametric sampling methods give good results only when the training dataset having sufficiently similar depth characteristics is provided. Moreover, they should retain a large RGB-D dataset for retrieving semantically similar images.

The second category casts the monocular depth estimation as a parametric learning process. For instance, a Markov random field (MRF) is learnt for mapping between RGB and depth space [17], [18]. The random field model encodes a priori assumption, so that the resulting depth image has statistical properties similar to those of the desired solution. Many other parametric models such as conditional random field (CRF) [19], logistic regression [20], and convolutional neural networks (CNNs) [21], [22], [24] have also been employed. Especially, recent approaches using the CNNs have shown a significant accuracy gain. The deeper CNN architecture is effective in increasing the capacity for exploiting input single image, but there is a tendency to produce coarse depth

Manuscript received June 20, 2017; revised February 7, 2018 and May 2, 2018; accepted May 9, 2018. Date of publication May 15, 2018; date of current version May 24, 2018. This work was supported in part by the Next Generation Information Computing Development Program through the National Research Foundation of Korea (NRF), Ministry of Science, ICT, under Grant NRF-2017M3C4A7069370, and in part by the Basic Science Research Program through the NRF under Grant NRF-2015R1D1A1A01061143. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Kalpana Seshadrinathan. *(Corresponding author: Kwanghoon Sohn.)*

Y. Kim, H. Jung, and K. Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, South Korea (e-mail: read12300@yonsei.ac.kr; coolguy@yonsei.ac.kr; khsohn@yonsei.ac.kr).

D. Min is with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03-760, South Korea (e-mail: dbmin@ewha.ac.kr).

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/TIP.2018.2836318

[1]Both groups leverage the discriminative power of a large-scale RGB-D database.

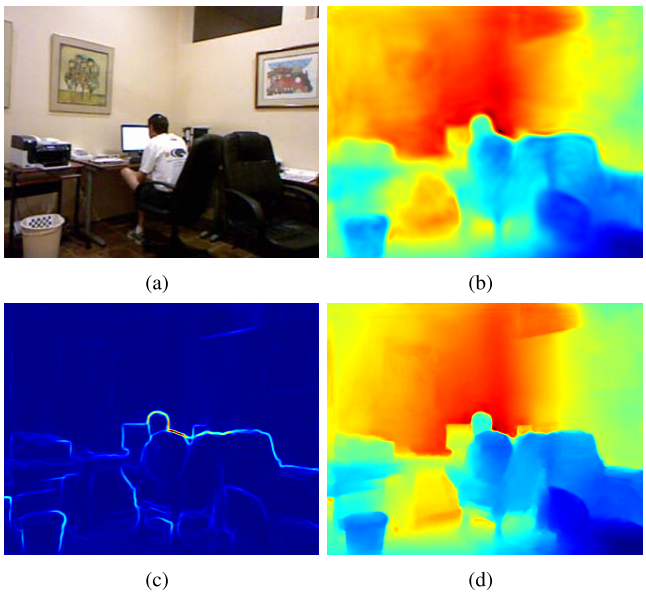


Fig. 1. (a) Input single RGB, (b) global prediction, (c) local prediction in gradient domain, and (d) result of integration ((b) and (c)). Our approach integrates different and complementary predictions of the CNNs, achieving good localization and the use of context at the same time. The integration step is also plugged in as a part of the CNNs, and thus the whole parameters can be learned in an end-to-end manner. The figure is best viewed in color.

outputs due to convolutional kernels with large receptive fields and max-pooling layers.

This paper presents a new framework for estimating depth from a single image. We propose a deep variational model which integrates complementary predictions of the CNNs (see Fig. 1(b) and (c)). Our model consists of two heterogeneous networks, named global and local networks, which are trained under different input and output configurations to capture both global metric (Fig. 1(b)) and local relative (Fig. 1(c)) information of the depth image. The integration step (Fig. 1(d)) using variational minimization is also plugged in as a part of the networks, making it possible to train the whole parameters end-to-end with the standard back-propagation algorithm. In addition, we further improve the results by employing adversarial loss [31]. Our model is able to generate realistic and structure-preserving depth prediction from a single image, without involving any low-level segmentation or superpixels.

Training a deep network requires a large RGB-D dataset for supervision. We additionally introduce a new RGB-D dataset for single image depth estimation, capturing 283 diverse indoor scenes (total 0.22 millon RGB-D pairs). It provides high-quality depth and corresponding RGB images captured by Microsoft Kinect v2. We will host our DIML RGB-D dataset at [52].

Overall, the main contributions of this work are highlighted as follows:

- We propose a deep variational model for single image depth estimation, which integrates the predictions from the complementary CNNs. The adversarial loss [31] is also used to make the depth prediction indistinguishable from natural depth images.
- We show that, by unrolling the optimization steps of Split Bregman (SB) iterations [28] in the integration network, our model can be trained in an end-to-end manner. This enables one to learn an efficient parameterization of the CNNs and hyper-parameter in the variational method simultaneously.
- We provide an intensive comparison study to demonstrate the effectiveness of the proposed method in several benchmarks including NYU v2 [39], Make3D [17], KITTI2012 [44], and DIML dataset [52].

The remainder of this paper is organized as follows. Section II describes related works for single image depth estimation. We present the deep variational model as well as training details in Section III. An extensive experimental comparison is then provided in Section IV. Finally, Section V concludes this paper.

II. Related Work

Understanding 3D structure of a scene has a rich history in image processing and computer vision. Early methods mainly focus on employing geometric constraints such as box and Manhattan models or utilizing video sequences that capture different viewpoints of a static scene over time. Recently, single image depth estimation has become increasingly popular thanks to the emergence of large-scale RGB-D dataset [39]. Among various methodologies, we review and discuss three lines of works that are most relevant to ours.

A. Nonparametric Sampling Methods

Nonparametric sampling approaches assume that a large-scale RGB-D dataset contain scenes that have appearance and geometric layout similar to an input query image. As a pioneering work, Karsch *et al.* [10] devised the depth transfer algorithm. For a given input, similar RGB images are retrieved by the GIST descriptor [11], corresponding depth images are warped via dense scene alignment [13], and the resulting depth prediction is spatially regularized using global optimization procedure. Konrad *et al.* [12] argued that dense scene alignment of [10] is computationally expensive and does not necessarily improve the quality of depth estimation. Instead, they directly fuse the retrieved depth images by computing a median value for each pixel. This initial estimate is then refined by using a joint bilateral filtering [14]. Note that the retrieved depth images do not always provide useful cues unless the database is carefully established. Such a dependency degenerates the quality of estimation when the depth distribution of the input image is quite different from that of the training RGB-D dataset. To address this problem, Choi *et al.* [15] devised the depth analogy method that transfers depth gradients as reconstruction cues, which are integrated by the Poisson equations. This method is less sensitive to the distribution of training dataset, but the resulting depth images suffer from a scale ambiguity since the reconstruction is performed with depth gradients only under Neumann boundary condition. It should be noted that most of nonparametric sampling methods require the entire RGB-D dataset to be available at test phase. They retrieve a subset of similar RGB images, which will be used for subsequent depth sampling tasks. This incurs prohibitively high computational

overhead and memory consumption, especially when the large scale RGB-D dataset is used.

B. Parametric Learning Methods

Another line of works attempt to model explicit dependency between RGB and depth images in a structured learning framework. Saxena *et al.* [17] modeled monocular cues based on the MRF whose edges encode a simple smoothness assumption between neighboring superpixels. This approach was further extended in [18] with a hierarchical MRF to model monocular cues at multiple spatial scales. Ladicky *et al.* [43] learned a pixel-wise depth classifier by assuming that the perceived size of the objects scales is inversely proportional to the distance. In [19], semantic object labels and better geometric priors were incorporated in the CRF framework. Recently, the CNNs have been successfully applied in single image depth estimation. Eigen *et al.* proposed a multi-scale architecture that first predicts the depth image from an input at coarser-scale network and refines it using finer-scale network [21]. In [22], relative depth annotations rather than metric depth were used to improve the performance in unconstrained settings. Laina *et al.* [23] devised a fast up-projection layer and combined it with the deep residual learning [35]. While the CNNs-based approaches have achieved state-of-the-art performance, they lack imposing the spatial smoothness constraint, often resulting in poor boundary localization and spurious regions. One notable exception is the work of Liu *et al.* [24], which learns the unary and pairwise potentials of continuous CRF with the CNNs. A superpixel pooling method is also proposed to speedup their patch-wise predictions in the CNNs. However, they considered the graphical model composed of nodes defined on superpixels and regressed the single depth value from a superpixel. This is problematic on the regions where the assumption of appearance-depth correlation is violated, e.g., highly textured surface. Chakrabarti *et al.* [25] estimated the distributions of depth gradients (including zero-order) using the CNNs. These predictions are then reconciled to form a final estimate of the scene depth through a separated globalization procedure. The network of [25] has a fairly high-dimensional output space (64×64 at each pixel in their implementation).

Beyond the monocular depth estimation, a large body of work has been devoted to predicting dense labels using CNNs. Ghiasi and Fowlkes [56] proposed a multi-scale CNN based on the Laplacian pyramid that uses activations of early layers for pixel-accurate semantic segmentation. Lin *et al.* [57] combined the inception module [22] and cascaded residual learning [35]. They used a sequence of four scales to refine activations and predictions on higher resolution. The context aggregation network introduced by Yu and Koltun [58] employed dilated convolutions for supporting large receptive fields without downsampling the spatial resolution. It shows state-of-the-art performance on semantic segmentation. However, dilated convolution kernels introduce a coarse sub-sampling of activations, which may lead to a loss of important details [57]. In a two-view correspondence problem, the coarse-to-fine formulation [59] was proposed to yield a pixel-accurate correspondence map by constraining the search space on finer levels through warping. It is conceptually similar to multi-scale network with residual learning used in the monocular depth estimation [21], [23].

Note that our approach belongs to the parametric learning method using the CNNs. But, it differs from previous works in that we predict the piecewise smooth depth image with global contexts and the discontinuous-preserving depth gradients using two contrasting CNNs, and then integrate them through variational approach in an end-to-end manner. To the best of our knowledge, metric- and relative-depth information have been predicted separately for monocular depth estimation, and were not integrated in a unified deep learning framework. For instance, [21] and [23] estimate metric depth only, while the works of [15] and [22] predict relative-depth information. We will show that our strategy is very effective in addressing the nonlinear regression problem.

C. Synergetic Methods

Baig *et al.* [26] retained the essence of non-parametric approach, but compressed the training dataset into a compact dictionary using clustering techniques. In addition, they learned a parametric transformation between the RGB and depth dictionaries to generate depth predictions. It was shown in [26] that the method establishes computational advantages without taking a massive hit on accuracy. Liu and Salzmann [27] formulated the discrete-continuous CRF, where the continuous variable encodes the depth and plane normal, and the discrete one represents relationships between neighboring superpixels. The corresponding optimization problem, initialized by the non-parametric sampling [12], is then solved using particle belief propagation.

III. Proposed Approach

The proposed deep variational model consists of global, local, and integration networks. The global network first estimate the piecewise smooth depth based on the entire RGB image [21], [23], [46], considering that the overall geometric layout is indistinguishable under local context. We also use depth gradients [15] as local cues, which are estimated through the local network. These two complementary predictions are then integrated in a unified deep CNN framework. We jointly train our model using the combination of two losses, L_1 loss and adversarial loss [31]. An overall framework is illustrated in Fig. 2.

A. Global Network

The global network takes a whole image as an input, and predicts an overall depth structure at a global level. Although the input and output differ in appearance, both are renderings of the same scene. Thus, structure of the input RGB image is roughly aligned with that of the output depth. We design the global network under these considerations.

Many previous approaches [21], [24], [46] based on the CNNs have used fully connected layers, which fix dimensions of the input and throw away spatial coordinates. Instead,

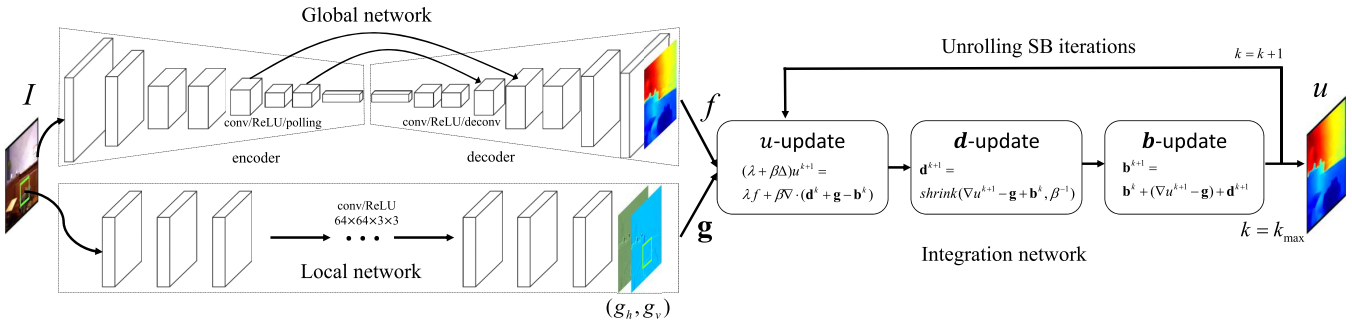


Fig. 2. The proposed deep variational model consists of three components. The global network uses a fully convolutional encoder-decoder architecture [29], and first computes an overall depth structure f at a global level. The local network consists of 10 convolution layers with 3×3 filters, and predicts fine details $\mathbf{g}$ in gradient domain. These complementary information are then merged in our integration network, where we unroll the optimization steps of the SB iteration [28] for minimizing the variational problem of (1). The proposed method is able to learn an efficient parameterization of the model including the filter coefficients and hyper-parameter λ in the variational problem of (1).

as shown in Fig. 2, we use a fully convolutional encoder-decoder architecture [29] that takes the input of arbitrary size and produces results proportional to the size of the input images. The encoder consists of a series of three 3×3 convolutions and rectified linear unit (ReLU), followed by 2×2 max-pooling with stride 2 for downsampling. After each downsampling step we double the number of feature channels. We use the first five convolution layers and the following pooling (called *conv5* and *pool5*) in the VGG [41] architecture. On the contrary, the decoder progressively enlarges the spatial resolution of convolutional activations through a sequence of deconvolution and convolution layers. The deconvolution layer is implemented using the transposed convolution and fixed (bilinear) filter kernel.

There is a great deal of low-level information shared between RGB and depth images, e.g., the location of prominent edges. Thus, it would be desirable to shuttle this information directly across the network. To this end, we add skip connections between convolution layers and their symmetric deconvolution layers, as shown in Fig. 2. There are basically two ways to realize the skip connection: summation and concatenation. We find that a simple element-wise sum of activations works well in our experiments. Using such connections boosts the performance and makes training the very deep network easier [35]. The global network captures the low-frequency structure accurately using a global view of the input, but produces coarse depth image. In the following sections, we will address this issue by developing local and integration networks.

B. Local Network

It is well-known in literatures [41], [53] that there is a trade-off between localization accuracy and the use of global context in deep network. The global network employs a series of convolution and max-pooling layers to robustly estimate the global 3D layout of the scene. The subtle details of the depth image, however, are lost during these processes although we add the skip connections. Inspired by [15], we additionally predict depth gradients by providing a local region (RGB-patch) as input. Note that using gradient information in the depth estimation is less sensitive to scene characteristics of training data [15].

The key idea is that the local network act as a feature extractor, which preserves the primary depth edges from the input RGB meanwhile eliminating the unwanted oscillation such as textures. The local network does not use pooling as it usually discards useful details essential for single image depth estimation. It consists of 10 convolution layers with 3×3 filters (a receptive field is of 21×21), followed by the ReLU (see Fig. 2). Since depth gradients contains both positive and negative values, the ReLU is not used for the last layer. We use the batch normalization [38] to alleviate the internal covariate shift by normalizing input distributions of every layer to the Gaussian distribution. The output channel of the local network is 2 for the horizontal and vertical depth gradients. Note that the output manifold of local network is topologically much simpler than that of global network, since depth gradients are very sparse. We will show that the local network with shallow and compact architecture can capture depth gradients, and can improve the performance of monocular depth estimation.

C. Integration Network

1) Formulation: Let f and $\mathbf{g} = [g_h, g_v]$ be the outputs of global and local networks, respectively. These complementary outputs are then seamlessly combined at the integration network. Formally, we solve the following variational problem to estimate the final depth image u:

$$\arg\min_{u} E(u) = \|\nabla u - \mathbf{g}\|_1 + \frac{\lambda}{2}\|u - f\|_2^2, \tag{1}$$

where the first term denotes a fidelity term penalizing the difference between the gradient of the estimated depth u and $\mathbf{g}$. The second one represents global prior knowledge about u, so that u becomes close to f. $\lambda > 0$ is a constant to balance the two terms, and is also learned in the deep network. ∇ denotes the gradient operator in the discrete setting. Minimizing the functional of (1) can be interpreted as the integration of

gradient field $\mathbf{g}$ using the quadratic prior from f. This formulation has a number of benefits over the simple Poisson reconstruction of [15] which exploits depth gradients only. First, the resulting depth image of [15] has scale ambiguity, and thus should be intentionally re-scaled to a certain range.[2] In contrast, the proposed method avoids such a problem by introducing f as quadratic prior to force a unique solution. Second, we use the L_1 norm so as to reduce the influence of outliers that may exist in the estimated gradient field $\mathbf{g}$.

The functional of (1) is convex, but cannot be minimized in a closed form. Thus, we choose Split Bregman (SB) iterations [28], as it guarantees fast convergence. We first replace $(\nabla u - \mathbf{g})$ by $\mathbf{d}$ and add a penalty constant β, yielding:

$$\min_{u,\mathbf{d}} \|\mathbf{d}\|_1 + \frac{\beta}{2}\|\mathbf{d} - (\nabla u - \mathbf{g}) - \mathbf{b}\|_2^2 + \frac{\lambda}{2}\|u - f\|_2^2, \quad (2)$$

where $\mathbf{b} = [b_h, b_v]^T$ is an auxiliary variable related to the Bregman distance [28]. The optimization procedure to obtain u is then given by:

$$\begin{aligned} u^{k+1} &= \arg\min_u \frac{\beta}{2}\left\|\mathbf{d}^k - (\nabla u - \mathbf{g}) - \mathbf{b}^k\right\|_2^2 + \frac{\lambda}{2}\|u - f\|_2^2, \\ \mathbf{d}^{k+1} &= \arg\min_{\mathbf{d}} \|\mathbf{d}\|_1 + \frac{\beta}{2}\left\|\mathbf{d} - (\nabla u^{k+1} - \mathbf{g}) - \mathbf{b}^k\right\|_2^2, \\ \mathbf{b}^{k+1} &= \mathbf{b}^k + (\nabla u^{k+1} - \mathbf{g}) - \mathbf{d}^{k+1}, \end{aligned} \quad (3)$$

Our key observations are that (i) each computation steps in the SB iterations [28] can be realized by layers of the CNNs and (ii) given a fixed number of iterations, the optimization procedure of (3) can be unrolled like the recurrent neural network [30]. These allow us to use the back-propagation algorithm to train the whole network end-to-end. A graphical depiction of the integration network is illustrated in Figs. 2 and 5. In the following, we detail how the individual steps in (3) are realized within the deep network.

2) Unrolling: The u-update in (3) is quadratic and the minimizer satisfies the following normal equations:

$$(\lambda + \beta\Delta)u^{k+1} = \lambda f + \beta\nabla\cdot(\mathbf{d}^k + \mathbf{g} - \mathbf{b}^k), \quad (4)$$

where $\nabla\cdot$ is the divergence operator, i.e., the adjoint of the gradient, and $\Delta = \nabla\cdot\nabla$ is the Laplacian operator. In matrix/vector form, the left-hand side of (4) becomes a block Toeplitz matrix, which can be diagonalized by the fast Fourier transform (FFT). Therefore, the u-update step can be implemented with three FFT calls and convolution layer that has fixed filter coefficients for the divergence operator.

Regarding the $\mathbf{d}$-update (basis pursuit problem), solutions are obtained by element-wise shrinkage [28]:

$$\begin{aligned} d_h^{k+1} &= shrink(\nabla_h u^{k+1} - g_h + b_h^k, 1/\beta), \\ d_v^{k+1} &= shrink(\nabla_v u^{k+1} - g_v + b_v^k, 1/\beta), \end{aligned} \quad (5)$$

where $shrink(z, \gamma) = \frac{z}{|z|}\max(|z| - \gamma, 0)$. It is a composition of element-wise division and max operator, and thus can be implemented with a standard Hinge function shifted by $1/\beta$.

[2]Since the objective function in [15] depends only on ∇u, their solution can be estimated up to an additive ambiguity, i.e., $u \mapsto u + c$, where c is a constant.

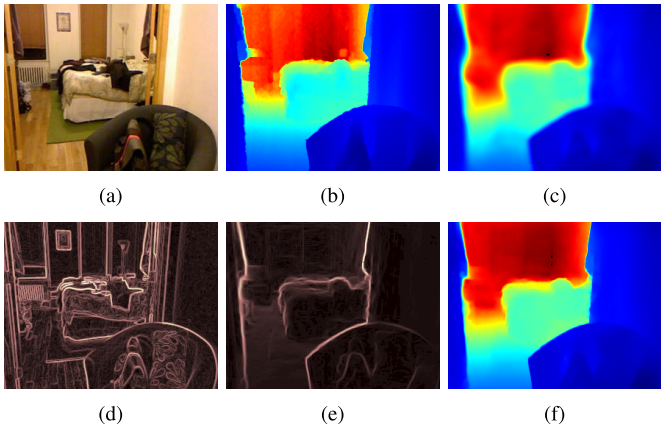


Fig. 3. The local network extracts depth gradients (d) from single RGB image (a), and meanwhile eliminates the unwanted oscillation such as texture. Note that $\mathbf{g}$ contains high-frequency gradients that do not coincide with depth images but the integration network robustly suppresses these defects, yielding the final depth prediction (f). (a) Input RGB. (b) Ground-truth. (c) f. (d) Gradient (RGB). (e) $\mathbf{g}$. (f) u.

Finally, the $\mathbf{b}$-update is an element-wise sum of four inputs of the same size.

In Fig. 3, we show the final and intermediate results from our deep variational model. The global network predicts the overall depth structure f (Fig. 3(c)) from a global perspective. The local network eliminates the unwanted oscillation caused by appearance (textures and colors), and extracts depth gradients $\mathbf{g}$ (Fig. 3(e)) from single image (Fig. 3(a)). It can be seen that while the local network which is shallow and compact can capture relative-depth information and fine detailes, $\mathbf{g}$ may often contain spurious depth gradients due to its local nature. However, our deep variational model robustly suppresses these defects in the final depth prediction u (Fig. 3(f)).

D. Training

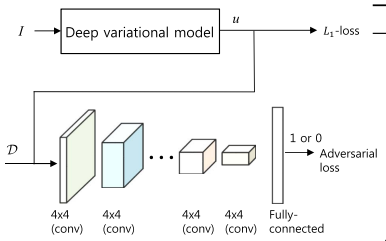
Given a large set of training samples, we now describe the training procedure in detail to find optimal parameters of our model. After pre-training, we fine-tune the whole parameters jointly in the combination of two losses, L_1 loss and adversarial loss.

1) Pre-Training: It is possible to randomly initialize the weights of our network and then train it end-to-end by unrolling the finite number of SB iterations. In practice, however, we observed that pre-training the global and local networks increases the prediction accuracy and accelerates convergence. We first apply the L_1 loss directly for the global network, minimizing:

$$\mathcal{L}_{global} = \frac{1}{M}\sum_p \|f^{(p)} - \mathcal{D}^{(p)}\|_1, \quad (6)$$

where $\mathcal{D}$ denotes ground-truth depth image and M is the total number of training samples. Similarly for the local network, we minimize:

$$\mathcal{L}_{local} = \frac{1}{M}\sum_p \|\mathbf{g}^{(p)} - \nabla\mathcal{D}^{(p)}\|_1. \quad (7)$$



Classifier architecture

Layer	Weight dimension (stride)
conv	$64 \times 1 \times 4 \times 4$ (2)
conv	$128 \times 64 \times 4 \times 4$ (2)
conv	$256 \times 128 \times 4 \times 4$ (2)
conv	$512 \times 256 \times 4 \times 4$ (2)
conv	$512 \times 512 \times 4 \times 4$ (2)
conv	$1024 \times 512 \times 4 \times 4$ (2)
FC	$1 \times 1024 \times 4 \times 5$ (1)
Sigmoid	-

Fig. 4. Deep variational model trained with L_1 reconstruction loss and adversarial loss for single image depth estimation. Every convolution layers in classifier are activated with LeakyReLU and batch-normalization layers. The adversarial loss drives depth predictions towards the ground-truth depth manifold, producing perceptually more convincing solutions.

A constant learning rate of 10^{-4} is used and momentum term set to 0.9. Note that in this stage, we use different kinds of input/output configurations for each network. The global network is pre-trained with the RGB-D pairs of full-resolution, exploiting global contextual information. On the other hand, we pre-train the local network by providing a local region (i.e., color patch) as an input to further improve the localization accuracy.

2) Loss Function: Recently, the adversarial loss [31], [32] based on a discriminative classifier have been used for generating sharp and realistic images. The classifier C takes the depth image u estimated from the integration network or the ground-truth depth image $\mathcal{D}$ as an input, and decides where it comes from. The adversarial loss is then defined as follows [31]:

$$\mathcal{L}_{adv} = \mathop{\mathbb{E}}_{\mathcal{D}\sim p_{\mathcal{D}}}[\log C(\mathcal{D})] + \mathop{\mathbb{E}}_{u\sim p_u}[\log(1 - C(u))], \quad (8)$$

where p_z represents a probability distribution over the kinds of data z. $\mathbb{E}$ is an expectation operator. The adversarial loss of (8) enables one to train the CNNs, deceiving the discriminative classifier C. That is, we can produce results that are highly similar to the ground-truth depth images or indistinguishable by C. Following the architectural guidelines introduced in [36], we build the classifier as in Fig. 4. It uses the LeakyReLU activation with 0.2 slope, and the strided convolution to reduce the spatial resolution instead of max-pooling. We also add the batch normalization [38] to the output of every convolution layer. The classifier C output a single scalar, representing the probability that the input comes from the ground-truth rather than p_u.

Finally, we jointly train the deep variational model and classifier C by combining the L_1 loss and adversarial loss. Thus, our final loss function is:

$$\mathcal{L}_{total} = \frac{1}{M}\sum_{p}\Big(\|u^{(p)} - \mathcal{D}^{(p)}\|_1 + \eta \log(1 - C(u^{(p)}))\Big), \quad (9)$$

where the constant η is set to 10^{-3}. Note that the classifier C is trained to maximize the adversarial loss of (8), and is used during training only. We use the stochastic gradient descent (SGD) and adaptively tune the learning rate beginning from 10^{-3}. After each 2 epoch, it decreases to 50 percent of the previous learning rate.

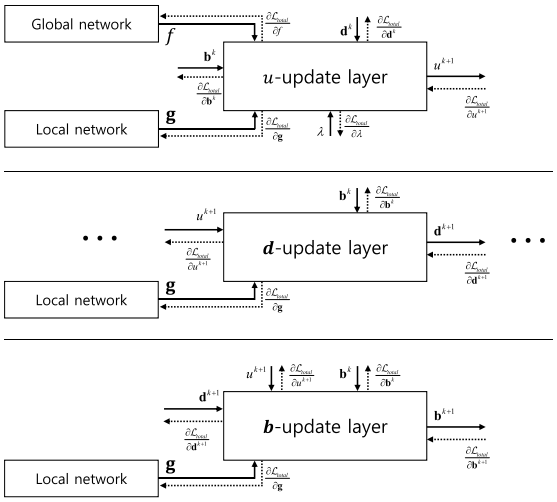


Fig. 5. Back-propagation paradigm through the integration network. Each layer in the integration network has closed form expressions for the gradients of its inputs (see the text for more details). These gradients are then further back-propagated onto the global and local networks that predict f and $\mathbf{g}$.

E. Back-Propagation

Our model parameters in the global and local networks are learned in an end-to-end fashion via the back-propagation algorithm. For each layer in the network, we need to receive the derivative of the final loss $\mathcal{L}_{total}$ with respect to its output. The layers then compute the gradients of its inputs and propagate them down through the network to the previous layer (see Fig. 5 for the graphical depiction). The derivative of $\mathcal{L}_{total}$ with respect to the output of integration network u can be obtained as follows:

$$\frac{\partial \mathcal{L}_{total}}{\partial u} \propto \sum_{p} \text{sgn}(u^{(p)} - \mathcal{D}^{(p)}) + \eta \frac{\partial \mathcal{L}_{adv}}{\partial u}, \quad (10)$$

where $\text{sgn}(z)$ denote the signum function that returns the pointwise sign of z.

To learn the parameters in the global and local networks, we require the expressions of gradients in the layers of integration network. We next derive these expressions.

1) u-Update Layer: Letting $L = (\lambda + \beta\Delta)$ and $s = (\lambda f + \beta\nabla\cdot(\mathbf{d}^k + \mathbf{g} - \mathbf{b}^\mathbf{k}))$, the $u-$update layer then outputs u^{k+1} by solving the linear system of (4), i.e., $Lu^{k+1} = s$. This implies the following relation:

$$\frac{\partial \mathcal{L}_{total}}{\partial s} = L^{-1}\frac{\partial \mathcal{L}_{total}}{\partial u^{k+1}}. \quad (11)$$

We denote the index of SB iterations [28] as $k = \{1, \ldots, K\}$. Using the multivariate chain rule, the derivative of $\mathcal{L}_{total}$ with respect to the inputs of $u-$update layer can be obtained as follows (except for λ):

$$\frac{\partial \mathcal{L}_{total}}{\partial in} = \frac{\partial s}{\partial in}\frac{\partial \mathcal{L}_{total}}{\partial s}, \quad (12)$$

where $in = f, \mathbf{g}, \mathbf{d}^k$, or $\mathbf{b}^k$. Substituting (11) into (12), we can compute $\partial\mathcal{L}_{total}/\partial in$. For instance, for the global and local

predictions (f and $\mathbf{g}$), we have:

$$\frac{\partial \mathcal{L}_{total}}{\partial f} \propto L^{-1}\left(\sum_{k=1}^{K}\frac{\partial \mathcal{L}_{total}}{\partial u^k}\right), \quad \frac{\partial \mathcal{L}_{total}}{\partial \mathbf{g}} \propto \nabla\frac{\partial \mathcal{L}_{total}}{\partial f} \tag{13}$$

We perform the summation over k since f and $\mathbf{g}$ contribute to each u-update layer. Note that s is the linear combination of in, and thus the derivatives of $\mathcal{L}_{total}$ with respect to $\mathbf{d^k}$ and $\mathbf{b^k}$ are now straightforward to obtain. These derivatives are further back-propagated onto the global and local networks or onto the previous $\mathbf{d}-$ and $\mathbf{b}-$update layers.

For λ, using the expression $\partial A^{-1}/\partial A = -A^{-T} \otimes A^{-T}$ where $\otimes$ denotes the Kronecker product, we arrive at:

$$\frac{\partial \mathcal{L}_{total}}{\partial \lambda} \propto \sum_{k=1}^{K}\left\{(f-u^k)^T\left(L^{-1}\frac{\partial \mathcal{L}_{total}}{\partial u^k}\right)\right\}. \tag{14}$$

The detailed derivations of (14) is available in Appendix. With the expression of (14), λ can be learned using gradient descent. Note that, in (13) and (14), we do not mean back-propagating through a multiplication of a matrix L^{-1}. We instead obtain $\partial \mathcal{L}_{total}/\partial f$ by solving the linear system:

$$\frac{\partial \mathcal{L}_{total}}{\partial f} \propto \mathcal{F}^{-1}\left(\frac{\sum_k \mathcal{F}(\partial \mathcal{L}_{total}/\partial u^k)}{\mathcal{F}(L)}\right). \tag{15}$$

Thus, both forward and backward steps in $u-$update layer can be performed efficiently using the FFT.

2) **d**- *and* **b**-*Update Layers:* The $\mathbf{d}-$update layer takes u^{k+1}, $\mathbf{g}$, and $\mathbf{b}^k$ and compute $\mathbf{d}^{k+1}$ using the shrinkage operator [28]. Its back-propagation is easily defined via the (absolute) indicator function $\mathbb{I}(|z| > 1/\beta)$ that returns 1 if the argument is true, and 0 otherwise. The $\mathbf{b}-$update layer consists of element-wise summation, and thus its back-propagation is trivial. We initialize $\mathbf{d^1}$ and $\mathbf{b^1}$ with zero vectors.

IV. Experimental Validation

In this section, we present an exhaustive experimental evaluation of the proposed deep variational model for single image depth estimation. We report the quantitative and qualitative comparison with the state-of-the-art methods in both indoor and outdoor scenes. For the quantitative comparison, we employ several metrics which have been used in prior works [21], [24]:

- Threshold: % s.t. $\max\left(\frac{\mathcal{D}_i}{u_i}, \frac{u_i}{\mathcal{D}_i}\right) = \delta < thr$
- abs rel: $\frac{1}{N}\sum_i |\mathcal{D}_i - u_i| / \mathcal{D}_i$
- sqr rel: $\frac{1}{N}\sum_i \|\mathcal{D}_i - u_i\|^2 / \mathcal{D}_i$
- RMS(lin): $\sqrt{\frac{1}{N}\sum_i \|\mathcal{D}_i - u_i\|^2}$
- RMS(log): $\sqrt{\frac{1}{N}\sum_i \|\log \mathcal{D}_i - \log u_i\|^2}$
- $\log_{10}$: $\frac{1}{N}\sum_i \left|\log_{10}\mathcal{D}_i - \log_{10} u_i\right|$

where u_i denotes the predicted depth at pixel indexed by i, and N is the total number of pixels.

For the implementation of the proposed method, we use the MatConvnet library [40], and train on 12GB NVIDIA GeForce GTX Titan. The encoder part of global network is initialized by the VGG [41] model pre-trained on the ILSVRC [42] dataset for image classification. We use the random initialization using Gaussian distributions for the decoder part and local network. Regarding the integration network, we initially set $\lambda = 0.01$, fix $\beta = 10$ and unroll the 10 SB iterations [28]. The u-update step is performed by GPU-enabled FFT function built in the MATLAB with the periodic boundary condition. In all cases, the momentum and weight decay parameters are set to 0.9 and 0.0005, respectively. The source codes for training and testing will be made publicly available.[3]

A. NYU v2 Depth Dataset

The proposed method is first applied to depth prediction on NYU v2 depth dataset [39] that consists of 0.5 million RGB-D images of indoor scenes. Among the entire NYU v2 dataset [39], we sample 0.12 million training RGB-D pairs using the official training/testing scene split. The RGB-D pairs are resized to 256×320 pixels for the efficient training. In this setting, we train our model with a batch size of 16 for 10 epochs jointly. We then use the common 654 test image including filled-in areas with colorization technique [54], but constrained to the axis-aligned rectangle as in [21]. We validate the performance of the proposed method against several state-of-the-art methods, including depth transfer (DT) [10], Im2Depth [26], Ladicky *et al.* (POP) [43], Liu *et al.* (DCNF) [24], Chen *et al.* (DPW) [22], Laina *et al.* (FCRN) [23], Chakrabarti *et al.* (HOP) [25], and Eigen and Fergus [21]. All methods including ours are data-driven approaches using a large scale RGB-D training database. Specifically, the first method [10] is based on the non-parametric sampling, the second one [26] corresponds to synergetic approach, and the others are the parametric learning methods. Especially, the last five methods use the deep neural network (CNN) for single image depth estimation. The results for the comparison with other methods are obtained from source codes provided by the authors, or are taken from their project websites.[4] We do not include any post-processing for fair comparison. Since the network output of the MSC [21] is 109 × 147 pixels, we bilinearly upsample the results and fill the missing border using joint bilateral filters [14]. Note that the MSC [21], DPW [22], and ours are trained on the full NYU v2 [39] training set (0.12 million images). However, the DCNF [24] is trained on RGB-D patches from the standard training set (795 images).

Table I compares the quantitative results of the proposed method and the state of the art. The best results for each metric are highlighted in bold. Since the DPW [22] predicts ordinal depth value only, quantitative results are obtained by normalizing the predictions such that the mean and standard deviation are the same as those of the mean depth image of the training set. This table shows that the CNNs-based methods [21]–[25] tend to give better quantitative results than non-parametric sampling [10] and boosted classifier [43], but that our deep variational model outperforms all methods. For the threshold metric, we outperform other methods until the

[3]http://diml.yonsei.ac.kr/DIML_singleDepth.

[4]https://www.cs.nyu.edu/~deigen/depth/

TABLE I

QUANTITATIVE COMPARISON OF THE PROPOSED METHOD AGAINST THE STATE OF THE ART ON 654 NYU V2 [39] TEST IMAGES. FOR THE REPORTED METRICS ABS REL, SQR REL AND RMS LOWER IS BETTER, IN CONTRAST FOR THE THRESHOLD (δ) HIGHER IS BETTER. THE RESULTS OF DPW [22] ARE OBTAINED BY ADJUSTING THE MEAN AND STANDARD DEVIATION TO BE SAME AS THOSE OF THE MEAN DEPTH IMAGE OF THE TRAINING SET

NYU v2 dataset [39]	DT [10]	Im2Depth [26]	POP [43]	DCNF [24]	DPW [22]	MSC [21]	HOP [25]	FCRN [23]	Ours	–
$\delta < 1.25$	–	0.597	0.542	0.614	–	0.769	0.896	0.811	**0.825**	Higher
$\delta < 1.25^2$	–	–	0.829	0.883	–	0.950	0.958	0.953	**0.976**	is
$\delta < 1.25^3$	–	–	0.940	0.971	–	0.988	0.987	0.988	**0.993**	better
abs rel	0.350	0.259	–	0.230	0.340	0.158	0.149	0.127	**0.117**	
sqr rel	–	–	–	–	0.421	0.121	0.118	–	**0.109**	Lower
RMS(lin)	1.2	0.839	–	0.824	1.104	0.641	0.620	0.573	**0.525**	is
RMS(log)	–	–	–	–	0.382	0.214	0.205	0.195	**0.172**	better

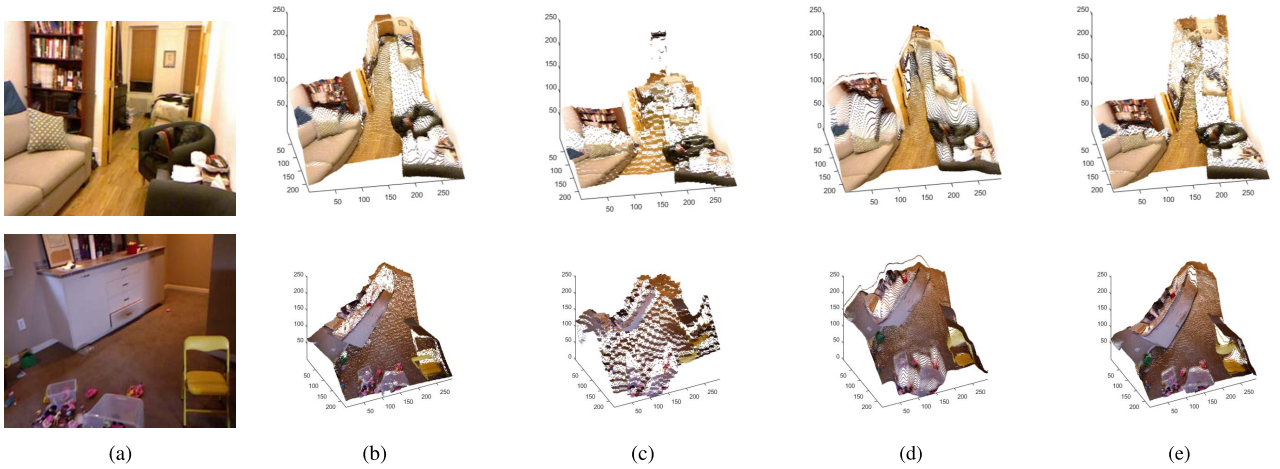
(a) (b) (c) (d) (e) (f)

Fig. 6. Visual comparison of the single image depth estimation on NYU v2 dataset [39]: (a) the input single image, (b) the ground-truth, (c) DCNF [24], (d) DPW [22], (e) MSC [21], and (f) the proposed method. The proposed method produces visually more plausible predictions with sharp depth transitions, aligning to RGB details. Since the network output of the MSC [21] is 109×147 pixels, we bilinearly upsample the results.

tolerance δ reaches 1.25^2, and after that the CNNs-based methods show the similar performance.

Figure 6 shows a visual comparison of the single image depth estimation on NYU v2 [39] test images. It clearly demonstrates that the proposed method yields visually compelling predictions with sharp depth transitions, aligning to RGB details. The CNNs-based methods [21], [22] usually lack imposing the regularity constraint that encourage spatial and appearance consistency of the output. In contrast, our method equipped with variational minimization of (1) avoids such problem (see Fig. 6, e.g., the human boundary and books on the desk). The MSC [21] progressively refines the prediction using 3-scale deep network but the output depth image is still visually unsatisfactory, often resulting in poor boundary localization and spurious regions. The DCNF [24] has relatively sharper depth transitions compared to [21] and [22] thanks to superpixel segmentation, but includes false texture edges. The superpixel-wise prediction of [24] is problematic in a textured surface where the assumption of appearance-depth correlation is violated. We also observe that the predictions of neighbouring superpixles are spatially inconsistent although they learn the pairwise potentials of the CRF. Similar to [15] the DPW [22] predicts ordinal relations of depth images only using a ranking loss that encourages a small difference between depths if the ground-truth relation is equality. We see that the method [22] has difficulties in estimating accurate metric depth from the input image.

Figure 7 compares the 3D reconstruction quality of the proposed method and the state-of-the-art methods. Depth values are normalized between $[0, 255]$ for the visualization purpose.



(a) (b) (c) (d) (e)

Fig. 7. Visual comparison of the 3D reconstruction on the NYU v2 dataset [39]: (a) the input single image, (b) the ground-truth, (c) DCNF [24], (d) MSC [21], and (e) the proposed method. Aside from sharp depth discontinuities, our method produces a much more accurate 3D reconstruction compared to the other deep learning-based approaches. Depth values are normalized between [0, 255] for visualization purposes.

TABLE II

THE RESULTS OF ABLATION STUDY ON THE NYU DATASET [39]. JT DENOTES THE JOINT TRAINING. FOR GLOBAL + LOCAL NETS W/O JT, WE MANUALLY CHOOSE THE BALANCING PARAMTER λ, SHOWING THE LOWEST RMS(LIN) VALUES

NYU dataset	abs rel	RMS(lin)	RMS(log)
Global net	0.179	0.627	0.226
Global net + $\mathcal{L}_{adv}$ ($\eta = 10^{-2}$)	0.184	0.639	0.232
Global net + $\mathcal{L}_{adv}$ ($\eta = 10^{-3}$)	0.168	0.624	0.221
Global + local nets w/o JT	0.135	0.551	0.193
Global + local nets w/ JT	0.121	**0.516**	0.180
Ours	**0.117**	0.525	**0.172**

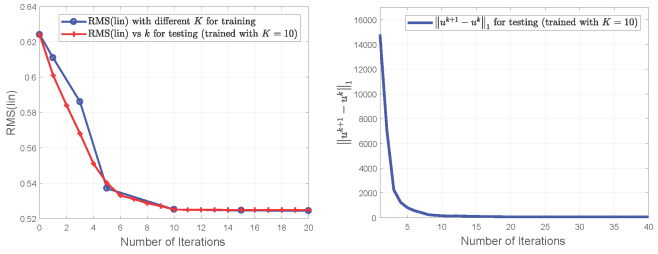


Fig. 8. Analysis on the number of SB iterations both at training and testing: (Left) RMS(lin) according to the number of SB iterations both at training and testing. (Right) the convergence behavior of integration network at testing. We use the model trained with $K = 10$ for testing.

It can be seen that, aside from sharp depth discontinuities, our method produces a much more accurate 3D reconstruction than other methods.

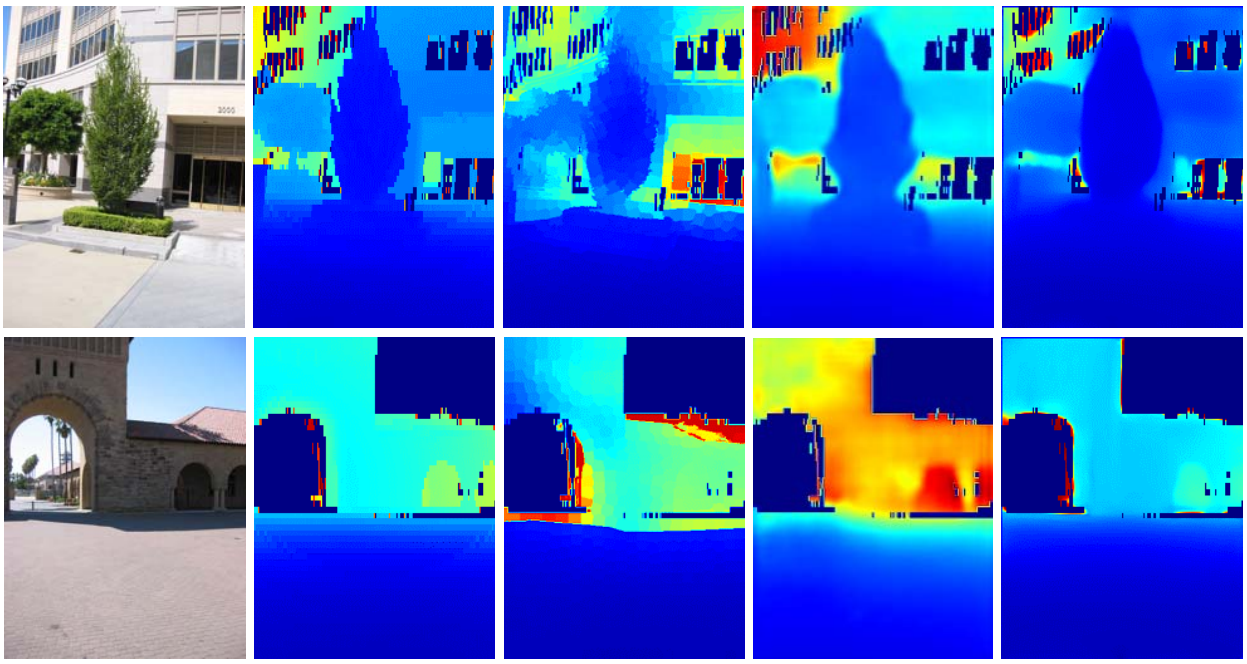
B. Ablation Study

We conduct ablation studies to analyze the contributions of our approach with different training schemes. Specifically, we compare the results on NYU v2 dataset [39] with the following parts stripped off: local network, adversarial loss, and joint training of the whole model. All variants were initialized with the VGG model and were trained identically with the same training schedule. We show the quantitative evaluation in Table II. "Global + local nets w/o JT" in Table II denotes that we separately train the global and local networks and minimize the variational energy of (1), i.e., there is no backpropagation through the integration network. In this case, the balancing paramter λ is chosen manually, showing the lowest RMS(lin) values. "Ours" in Table II denotes the full method. All components are useful for monocular depth estimation. However, we found that using larger values of η than 10^{-3} decreases the performance and makes the training unstable. We therefore set η to 10^{-3}. The integration of metric- and relative-depth information has the most impact on performance, and joint training of the whole model results in the further improvement. In the end-to-end joint training, the parameters of our model including λ get optimized to increase the overall accuracy.

Additionally, we analyze the influence of number of SB iterations at both training and testing. To this end, we trained our model with $K = [1, 3, 5, 10, 15, 20]$, and compared the performances. Figure 8(a) shows the results for different K at training (blue line) and testing (red line). It can be observed that the larger K at training improves the performance, but the gain is saturated after $K = 10$ iterations. We thus choose $K = 10$. This offers the best trade-off between computational complexity and accuracy in our experiments. Figure 8(b) shows how the differences $\|u^{k+1} - u^k\|_1$ evolve at each testing iteration. The integration network converges around $K = 10$ that was used for training. It is consistent with the results that more iterations at testing do not improve the performance further (see the red line of Fig. 8(a)).

C. Make3D Dataset

Here, we evaluate our method on the Make3D dataset [17] that contains 534 RGB-D pairs depicting outdoor scenes. We use the official training/test split provided with the dataset [17], i.e., 400 RGB-D pairs for training and 134 for testing.



(a) (b) (c) (d) (e)

Fig. 9. Visual comparison of the single image depth estimation on the MAKE3D dataset [17]: (a) the input single image, (b) the ground-truth, (c) DCNF [24], (d) DPW [22], and (e) the proposed method. We mask out the regions that correspond to distances larger than 70 meters in the ground-truth. Since the network output of the FCRN [21] is 115 × 86 pixels, we bilinearly upsample the results to 256×192. The figure is best viewed in color (all colormaps are not scaled for fair comparison).

Data augmentation [21] is performed to increase the number of training samples:

- Rotation: Training pairs are rotated by $[-5, 5]$ degrees.
- Scaling: Training pairs are scaled by $s \in [1, 1.5]$ times, and the depths are divided by s.
- Color shift: Input values are multiplied by a random value $\in [0.8, 1.2]$ for each channel separately.
- Flip: Training pairs are horizontally flipped with 0.5 probability,

resulting in around 20K training samples. The ground-truth depth image is of size 305×55, but the RGB image originally has 1704×2272 resolution. All training pairs are resized to 256×192 for an efficient implementation. Due to limitations of the depth sensing device used for capturing ground-truth depth data, objects that are more than 80 meters (e.g., trees behind the arch and windows in building in Fig. 9) are all equally mapped to a single distance of 80 meters. Thus, we masked out pixels of distances over 70 meters. With this processing step, we train our model for 40 epochs.

In Fig. 9, we qualitatively compare the results obtained by our method to the publicly available results of [23] and [24]. Due to the low visibility of Make3D dataset [17], the depth images are pseudo-colored using ColorJet. Our model produces predictions that well capture local details aligning with the ground-truth depth images. Furthermore, the predicted metric depth by our method is most similar to that of the ground-truth (all colormaps are not scaled for fair comparison). The DCNF [24] fails to obtain spatially consistent results for foreground objects, e.g., tree and building, and is perturbed by imprecise over-segmentation. The FCRN [23] occasionally mis-estimates the global depth scale (the last row in Fig. 9) and smooths depth discontinuities.

TABLE III

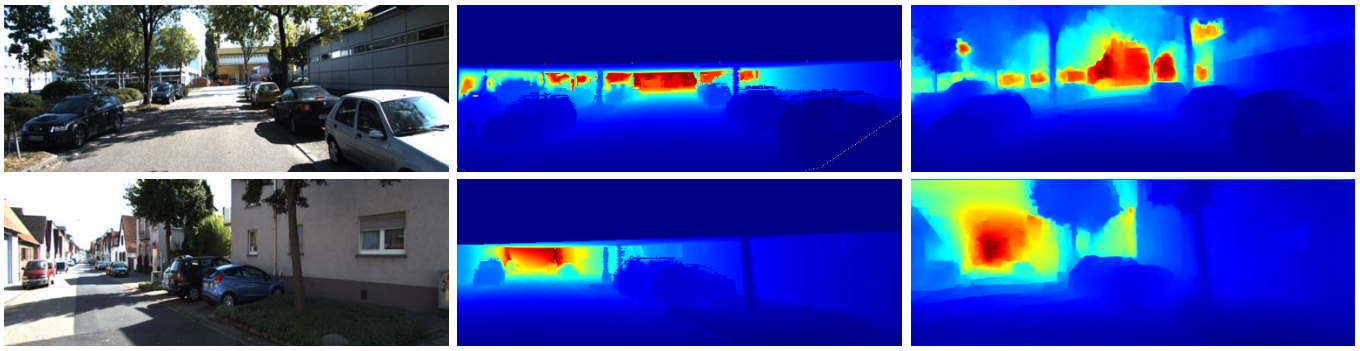
QUANTITATIVE COMPARISON OF SINGLE IMAGE DEPTH ESTIMATION ON 134 MAKE3D [17] TEST IMAGES. SINCE THE FAR-AWAY OBJECTS ARE ALL MAPPED TO THE ONE DISTANCE OF 80 METERS, ERRORS ARE CALCULATED ONLY IN THE REGIONS WHERE THE GROUND-TRUTH DEPTH LESS THAN 70 METERS

Make3D dataset [17]	abs rel	RMS(lin)	$\log_{10}$
DT [10]	0.355	9.20	0.127
DCNF [24]	0.287	7.36	0.109
FCRN [23]	0.176	**4.46**	0.072
Ours	**0.141 (0.141)**	4.85 (4.87)	**0.058** (0.059)

The quantitative results of the proposed method and the state of the art are reported in Table III. The numbers in brackets represent the results of ours obtained without adversarial loss $\mathcal{L}_{adv}$. Compared to the FCRN [23], we get better abs rel and RMS(log) errors but degraded RMS(lin). This can be explained by the limitations of Make3D dataset [17] that depth measurement is not accurate and depth borders in the ground-truth are not accurately registered to the color image. For similar reasons, the adversarial learning does not show a significant performance gain on this dataset [17].

D. KITTI Dataset

We further present results for the KITTI dataset [44], which consists of outdoor scenes with depths captured by the Velodyne LiDAR [45]. Since there is no official split for the raw data, we use the same split from Eigen [46] resulting in 697 and 33K images for testing and training, respectively. The RGB images are originally 375×1242, and are resized to 192×512 (about half-resolution) for the network inputs.



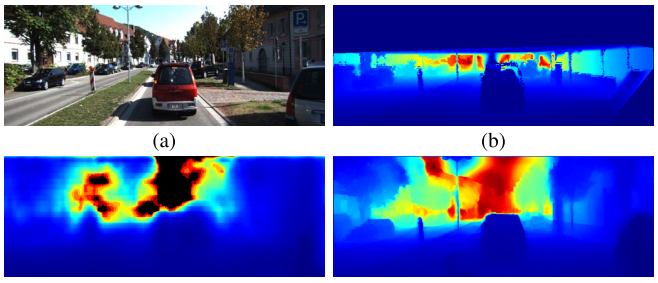
(a) (b) (c)

Fig. 10. Visual comparison of the single image depth estimation on the KITTI dataset [44]: (a) the input single image, (b) the ground-truth, and (c) the proposed method. We use the test/training split of 697 images as proposed by Eigen [46]. Since the ground-truth (velodyne depth) is very sparse, we interpolate it for visualization purposes. The figure is best viewed in color.

To generate the ground-truth depth images, we reproject the 3D points from the Velodyne LiDAR [45] to the left color camera. The velodyne LiDAR [45], however, produces sparse depth values for less than 20∼30% of the pixels in half-resolution. Aside from this, the depth values are provided only at the bottom part of the RGB image (see Fig. 10(b)). Thus, we adopt an alternative approach for generating training data.

Given the rectified stereo image pairs from the KITTI dataset [44], we first generate disparity map using the offline stereo matching method, called MC-CNN [48], which employs deep neural networks for measuring similarity between two patches. Specifically, the cost volume filtering of [49] and semi-global matching method [50] are applied to the raw matching cost computed from CNNs [48]. We then recover a depth image using calibration parameters (baseline and focal length) of the rectified stereo image pairs. It is important to note that our training data is converted from disparity maps, and thus may contain large quantization errors especially when the disparity value is small, i.e., an object is far away. However, we found that depth images obtained by the MC-CNN method [48] works well for training our model. More sophisticated global stereo methods considering slanted surface or providing sub-pixel precision could be employed to further improve the depth accuracy of the training data. We reserve this as future works. Our model was trained with a batch size of 8 for 10 epochs. The adversarial loss $\mathcal{L}_{adv}$ is not applied to the KITTI dataset [44] as we artificially generate the training data.

Figure 10 shows examples of predictions on the KITTI dataset [44]. We interpolate sparse Velodyne [45] depths for the visualization purpose, as shown in Fig. 10(b). It shows that the proposed method produces sharp transitions, particularly near the road and car edges. As we use RGB-D training data generated from stereo matching method [48], it is possible to predict depths at the upper part of the RGB image (Fig. 10(c)) although they do not exist in the KITTI dataset [44] (Fig. 10(b)). A qualitative comparison to Garg *et al.* [47] is shown in Fig. 11. Garg *et al.* [47] proposed the unsuperivsed reconstruction loss for monocular depth estimation, penalizing photometric errors between stereo pairs. This enables to learn the network without ground-truth depth images [47]. However, the reconstruction loss prefers smooth



(a) (b) (c) (d)

Fig. 11. Effectiveness of our approach for generating training label: (a) the input single image, (b) the ground-truth, (c), the result of Garg *et al.* [47], and (d) the proposed method. We use the MC-CNN [48] stereo matching algorithm to generate the training data. This enables us to predict depths within the upper part of the RGB image.

TABLE IV

QUANTITATIVE COMPARISON OF SINGLE IMAGE DEPTH ESTIMATION ON 697 KITTI DATASET [44] TEST IMAGES (EIGEN SPLIT [46]). [47] SET THE MAXIMUM OUTPUT DEPTH RANGE TO 50m

KITTI dataset	abs rel	RMS(lin)	RMS(log)
Make3D [17]	0.280	8.73	0.361
Eigen *et al.* [46]	0.190	7.15	0.270
DCNF [24]	0.217	7.04	–
Garg *et al.* [47]	**0.169**	**5.11**	0.273
Ours	0.177	6.57	**0.254**

depth transitions due to occlusions, as shown in Fig. 11(c). The proposed method achieves superior qualitative results (Fig. 11(d)) on the KITTI dataset [44].

The quantitative results using the ground-truth depths obtained from Velodyne [45] are reported in Table IV. Garg *et al.* [47] obtains lower RMS(lin) error compared to ours (they set the maximum output depth range to 50m). We conjecture that it is because our pseudo ground-truth contains large quantization erros when the disparity value is small. However, in RMS(log) and abs rel in which large errors are penalized the proposed method achieves comparable performance.

E. DIML Kinect v2 Dataset

For a more comprehensive comparison of single image depth estimation, we perform additional experiments on our own RGB-D dataset [52].

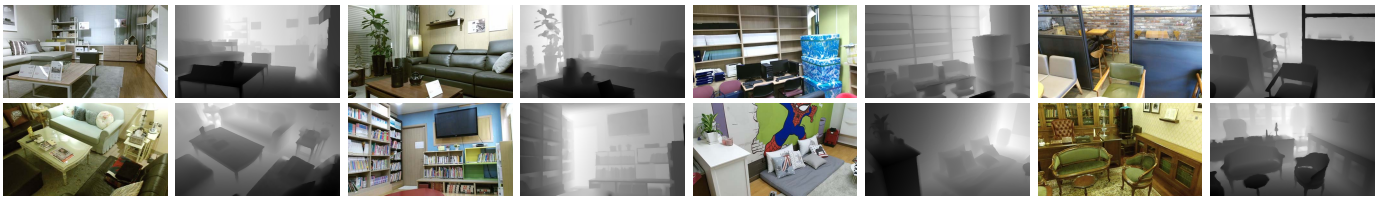
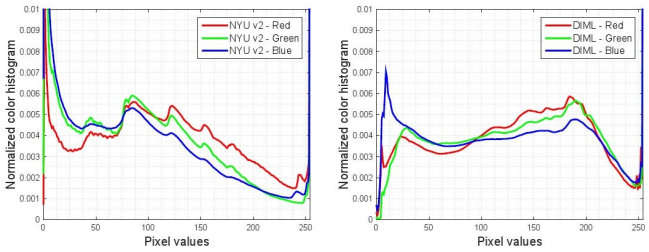


Fig. 12. Sample RGB-D pairs from our Kinect v2 dataset (DIML [52]). We provide synchronized RGB-D frames from Kinect v2, consisting of more than 200 indoor scenes. Our scenes are captured at various places, e.g., offices, rooms, dormitory, exhibition center etc.



(a) (b)

Fig. 13. Normalized color histograms for 15K training images. (a) NYU v2 dataset [39] and (b) Our DIML dataset [52]. The color distribution of our DIML dataset [52] is very different from that of the NYU v2 [39] dataset.

1) Data Capturing: We captured the RGB-D images of various indoor scenes using the Kinect v2 (time-of-flight sensor) camera. The scenes were captured steadily with a tripod. Our scenes are from a variety of categories, including living room, cafe, corridor, kitchen, store, and classroom. The total numbers of category and scene are 18 and 283, respectively and the entire raw data contains 0.22 million RGB-D images. We provide the official training/testing split (0.15 million for training and 70K for testing). The color image is originally captured with 1080×1920 resolution and the depth image is of size 424×512 resolution. The captured depth values range from 0.5m to 7m.

For the registration, we first calibrate RGB and IR cameras using [51] and then estimate shift coefficient between IR and depth images. After projecting depth values to a color image domain, we discard the region exceeding the field of view of IR camera and resize the depth image to match the aspect ratio of color camera. The RGB and depth images are thus of size 756×1344 and 288×512, respectively. Sample RGB-D images are shown in Fig. 12.

2) Further Experiments: As shown in Fig. 13, the color distribution of our DIML dataset [52] is very different from that of the NYU v2 [39] dataset. To analyze the sensitivity of single image depth estimation on scene characteristics of training data, we first test on the DIML dataset [52] using the model only trained on the NYU v2 [39]. Quantitative results on our 503 test images are reported in Table V. All methods show degraded performance compared to when they are tested on the NYU v2 [39] test data in Table I. However, we see that our deep variational model outperforms all other existing methods. Finally, we fine-tune the NYU v2 pre-trained model using the whole 0.15 million training pairs in the DIML dataset [52]. The RGB-D pairs are resized to 192×384 for the network inputs. We show qualitative results in Fig. 14. Our fine-tuned model produces visually plausible depth prediction for the test set that was captured at different locations and times.

TABLE V

QUANTITATIVE COMPARISON OF SINGLE IMAGE DEPTH ESTIMATION ON 503 KINECT V2 DATASET [52] TEST IMAGES. FOR ALL METHODS, WE USE THE MODEL TRAINED ON THE NYU V2 DATASET [39] ONLY. OUR TEST IMAGES ARE AVAILABLE AT [52]

DIML dataset [52]	abs rel	RMS(lin)	RMS(log)
DCNF [24]	0.524	1.138	0.392
FCRN [23]	0.252	0.656	0.257
Ours	**0.226** (0.235)	**0.551** (0.560)	**0.215** (0.228)

(a) (b)

Fig. 14. (a) Input single RGB and (b) depth prediction. Fine-tuning the NYU v2 pre-trained model on our dataset produces visually plausible depth prediction for the test set that was captured on different locations and times.

V. CONCLUSION

In this paper we have introduced a deep variational model for single image depth estimation. Inspired by the depth analogy [15], we predict depth gradients using deep networks and use it as the local cues. A global and coarse depth prediction is further estimated to resolve the scale ambiguity arising when recovering depth values from depth gradients.

These complementary predictions are integrated in a unified deep CNN framework for estimating the final depth image. We showed that the whole network parameters can be trained in an end-to-end manner by unrolling the optimization steps of the SB iteration. We also offered Kinect v2 RGB-D dataset, capturing 283 diverse indoor scenes. The raw dataset consists of 0.22 million RGB-D pairs. Experimental results demonstrate the flexibility and effectiveness of the proposed method in several benchmarks including both indoor and outdoor scenarios. In future work, we plan to apply the proposed method to other dense prediction tasks, such as semantic segmentation and motion estimation.

APPENDIX

Consider the single step of SB iterations [28]. With the multivariate chain rule, we have the following expression for the derivative of $\mathcal{L}_{total}$ with respect to λ:

$$\begin{aligned}\frac{\partial \mathcal{L}_{total}}{\partial \lambda} &= \frac{\partial L}{\partial \lambda}\frac{\partial u}{\partial L}\frac{\partial \mathcal{L}_{total}}{\partial u} + \frac{\partial h}{\partial \lambda}\frac{\partial u}{\partial h}\frac{\partial \mathcal{L}_{total}}{\partial u} \\ &= \frac{\partial L}{\partial \lambda}\frac{\partial \mathcal{L}_{total}}{\partial L} + \frac{\partial h}{\partial \lambda}\frac{\partial \mathcal{L}_{total}}{\partial h}, \end{aligned} \tag{16}$$

where $L = (\lambda + \beta\Delta)$ and $h = \lambda f + \beta\nabla \cdot (\mathbf{d}^k + g - \mathbf{b}^k)$. The second term in (16) is:

$$\frac{\partial h}{\partial \lambda}\frac{\partial \mathcal{L}_{total}}{\partial h} = f^T(L^{-1}\frac{\partial \mathcal{L}_{total}}{\partial u}). \tag{17}$$

Next, we derive $\frac{\partial \mathcal{L}_{total}}{\partial L}$ for the first term in (16). Using (4) and the identity $\partial A^{-1}/\partial A = -A^{-T} \otimes A^{-T}$ [55], we obtain:

$$\frac{\partial \mathcal{L}_{total}}{\partial L} = -\left(L^{-1}\frac{\partial \mathcal{L}_{total}}{\partial u}\right) \otimes u. \tag{18}$$

Note that since λ is spatially invariant and the off-diagonal elements of L do not depend on λ, the first term in (16) can be computed more efficiently as follows:

$$\frac{\partial L}{\partial \lambda}\frac{\partial \mathcal{L}_{total}}{\partial L} = -u^T\left(L^{-1}\frac{\partial \mathcal{L}_{total}}{\partial u}\right). \tag{19}$$

Substituting (17) and (19) into (16), we finally conclude:

$$\frac{\partial \mathcal{L}_{total}}{\partial \lambda} = (f - u)^T\left(L^{-1}\frac{\partial \mathcal{L}_{total}}{\partial u}\right). \tag{20}$$

REFERENCES

[1] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 3748–3754.

[2] Q. Chen and V. Koltun, "A simple model for intrinsic image decomposition with depth cues," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 241–248.

[3] J. Shotton *et al.*, "Efficient human pose estimation from single depth images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2821–2840, Dec. 2013.

[4] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 567–576.

[5] D. Kim, D. Min, and K. Sohn, "A stereoscopic video generation method using stereoscopic display characterization and motion analysis," *IEEE Trans. Broadcast.*, vol. 54, no. 2, pp. 188–197, Jun. 2008.

[6] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 690–706, Aug. 1999.

[7] O. Wang, M. Lang, M. Frei, A. Hornung, A. Smolic, and M. Gross, "StereoBrush: Interactive 2D to 3D conversion using discontinuous warps," in *Proc. EUROGRAPHICS*, 2011, pp. 47–54.

[8] M. Liao, J. Gao, R. Yang, and M. Gong, "Video stereolization: Combining motion analysis with user interaction," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 7, pp. 1079–1088, Jul. 2012.

[9] W. Zhou, M. Salzmann, X. He, and M. Liu, "Indoor scene structure analysis for single image depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 614–622.

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

[11] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IEEE Trans. Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, May/Jun. 2001.

[12] J. Konrad, M. Wang, C. Wu, D. Mukherjee, and P. Ishwar, "Learning-based, automatic 2D-to-3D image and video conversion," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3485–3496, Sep. 2013.

[13] C. Liu, J. Yuen, and A. Torralba, "SIFT FLOW: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[14] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 96–100, 2007.

[15] S. Choi, D. Min, B. Ham, Y. Kim, C. Oh, and K. Sohn, "Depth analogy: Data-driven approach for single image depth estimation using gradient samples," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5953–5966, Dec. 2015.

[16] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM. Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 24.

[17] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

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

[19] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1253–1260.

[20] H.-T. Chen and T.-L. Liu, "Finding familiar objects and their depth from a single image," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2007, pp. VI-389–VI-392.

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

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

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

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

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

[26] M. H. Baig, V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan, "Im2Depth: Scalable exemplar based depth transfer," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 145–152.

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

[28] T. Goldstein and S. Osher, "The split Bregman method for L_1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[29] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1520–1528.

[30] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1529–1537.

[31] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2672–2680.

[32] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1125–1134.

[33] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrel, and A. A. Efros, "Context Encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2536–2544.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[36] A. Radford, L. Metz, and S. Chintala. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434
[37] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vision*, Jun. 2015, pp. 2758–2766.
[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 448–456.
[39] N. Silberman, P. Kohli, D. Hoiem, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
[40] *MatConvnet*. Accessed: Feb. 6, 2017. [Online]. Available: http://www.vlfeat.org/matconvnet/
[41] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[42] O. Russakovsky *et al.*, "ImageNet Large scale visual recognition challenge," *Int. Journ. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[43] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 89–96.
[44] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[45] *Velodyne*. Accessed: Feb. 15, 2017. [Online]. Available: http://velodynelidar.com/
[46] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
[47] R. Garg, V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 740–756.
[48] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1592–1599.
[49] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073–1079, Jul. 2009.
[50] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.
[51] *iai Kinect2*. Accessed: Feb. 7, 2017. [Online]. Available: https://github.com/code-iai/_kinect2/
[52] *DIML RGBD*. Accessed: Oct. 7, 2017. [Online]. Available: http://diml.yonsei.ac.kr/DIML_rgbd_dataset/
[53] S. Kim, D. Min, S. Lin, and K. Sohn, "Deep self-convolutional activations descriptor for dense cross-modal correspondence," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 679–695.
[54] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.
[55] S. Chandra and I. Kokkinos, "Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 402–418.
[56] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 519–534.
[57] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1925–1934.
[58] F. Yu and V. Koltun. (2015). "Multi-scale context aggregation by dilated convolutions." [Online]. Available: https://arxiv.org/abs/1511.07122
[59] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 4161–4170.

Youngjung Kim (S'14) received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2013, where he is currently pursuing the joint M.S. and Ph.D. degrees in electrical and electronic engineering. His current research interests include variational method and continuous optimization, both in theory and applications in image processing and computer vision.

Hyungjoo Jung (S'16) received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2015, where he is currently pursuing the joint M.S. and Ph.D. degrees in electrical and electronic engineering. His current research interests include image processing and computer vision.

Dongbo Min (M'09–SM'15) received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2003, 2005, and 2009, respectively. From 2009 to 2010, he was a Post-Doctoral Researcher with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore. From 2015 to 2018, he was an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, South Korea. Since 2018, he has been an Assistant Professor with the Department of Computer Science and Engineering, Ewha Womans University, Seoul. His current research interests include computer vision, 2D/3D video processing, computational photography, and continuous/discrete optimization.

Kwanghoon Sohn (M'92–SM'12) received the B.E. degree in electronic engineering from Yonsei University, Seoul, South Korea, in 1983, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1985, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1992. From 1992 to 1993, he was a Senior Member of research engineer with the Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, South Korea. He was a Post-Doctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. From 2002 to 2003, he was a Visiting Professor with Nanyang Technological University, Singapore. He is currently a Professor with the School of Electrical and Electronic Engineering, Yonsei University. His research interests include 3D image processing and computer vision.