This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Single-Image Depth Estimation Based on Fourier Domain Analysis

Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim
Korea University
{jaehanlee, mhheo, krkim}@mcl.korea.ac.kr, changsukim@korea.ac.kr

Abstract

We propose a deep learning algorithm for single-image depth estimation based on the Fourier frequency domain analysis. First, we develop a convolutional neural network structure and propose a new loss function, called depth-balanced Euclidean loss, to train the network reliably for a wide range of depths. Then, we generate multiple depth map candidates by cropping input images with various cropping ratios. In general, a cropped image with a small ratio yields depth details more faithfully, while that with a large ratio provides the overall depth distribution more reliably. To take advantage of these complementary properties, we combine the multiple candidates in the frequency domain. Experimental results demonstrate that proposed algorithm provides the state-of-art performance. Furthermore, through the frequency domain analysis, we validate the efficacy of the proposed algorithm in most frequency bands.

1. Introduction

Depth estimation is the process of predicting the depth map of a scene using one or more images. The depth information serves as an important cue for understanding geometric relationship in the scene. For instance, an RGBD image, which has color and depth channels, can be applied in a variety of tasks, such as 3D model reconstruction [13, 30, 33], scene recognition [27, 32, 33], human pose estimation [37]. Depths can be estimated from stereo images [31] or motion sequences [2, 7, 17, 31, 40], which provide relatively rich information for understanding 3D structures. In contrast, it is more challenging and ambiguous to estimate depths from a single image [1, 3, 5, 6, 15, 19–22, 24, 28, 38, 42, 43], which does not allow the correspondence matching between stereo images or temporal frames.

Various geometric or image composition assumptions have been made to overcome the ambiguities in single-image depth estimation [9, 10, 13, 23, 36, 41]. For example, 3D reconstruction can be performed, assuming that a scene is composed of flat planes [13], that an image is composed in certain perspective [9], or that a scene has the floor-walls geometry [4]. Also, focal blurs were exploited in [41], and haze strengths inferred from the dark channel prior were used in [9]. These techniques, however, can reconstruct depths in specific cases only when the corresponding assumptions are valid.

Recently, several methods have been proposed to employ graph-based models [21, 23, 29, 30], such as Markov random field (MRF) and conditional random field (CRF). Moreover, with the fast development of deep learning technology [11, 12, 16], various attempts have been made to use convolutional neural networks (CNNs) for single-image depth estimation [3, 5, 6, 19, 38].

In this work, we propose a CNN-based algorithm for single-image depth estimation, which makes multiple predictions and combines the results in the Fourier frequency domain. First, we develop a CNN based on the ResNet [11] architecture. It includes additional paths for extracting intermediate features. Also, in order to train the network reliably for a wide range of depths, we propose the depth-balanced Euclidean (DBE) loss function. Then, we generate multiple depth map candidates by cropping an input image with various cropping ratios. In general, a cropped image with a small cropping ratio reconstructs local depth details more faithfully, while that with a large ratio recovers the overall depth distribution more reliably. To take advantages of these complementary properties, we combine the multiple candidates in the Fourier frequency domain.

Extensive experimental results on the NYUv2 depth dataset [33] demonstrate that the proposed algorithm yields the state-of-the-art performance, outperforming the conventional algorithms significantly. Moreover, by analyzing estimated depth maps in the frequency domain, we validate the efficacy of the components of the proposed algorithm in a wide range of frequencies.

This paper has three main contributions:

- We design a ResNet-based depth estimation network.
- We propose the DBE loss to enable more reliable training of the network.
- To the best of our knowledge, this is the first work to

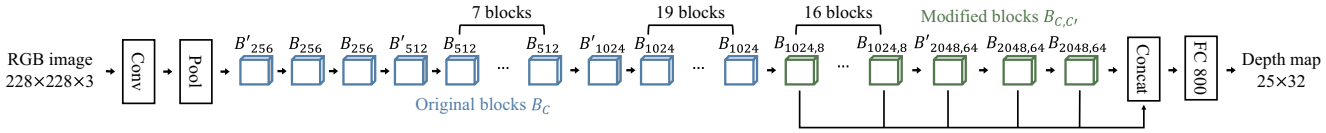


Figure 1. The CNN structure of the proposed single-image depth estimator. The details of blue blocks B_C or green blocks $B_{C,C'}$ are shown in Figure 2. Also, B'_C and $B'_{C,C'}$ are identical with B_C and $B_{C,C'}$, respectively, except that their short connections include convolution and batch normalization layers [11].

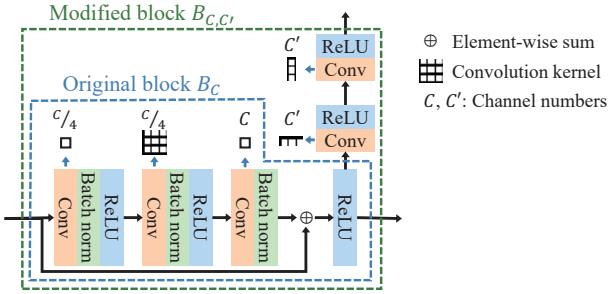


Figure 2. The detailed structures of original and modified blocks.

perform the Fourier analysis of the single-image depth estimation problem. We also propose an accurate and reliable scheme to combine multiple depths in the frequency domain.

2. Related Work

Various attempts have been made for single-image depth estimation. For instance, Saxena *et al*. [29] adopted a multi-scale MRF to take into account the global context of an image, as well as its local features, for depth estimation. In [30], they divided an image into homogeneous patches, used an MRF to obtain 3D parameters at each patch, and reconstructed a 3D structure. Kersh *et al*. [14] proposed transferring the depth map of a reference RGBD image in a dataset to an input color image. Similarly, Liu *et al*. [24] exploited the depth information in a reference RGBD image. They used it to formulate a unary potential in a CRF model for depth estimation. Also, Ladicky *et al*. [18] formulated depth estimation as the problem of predicting the likelihood that each pixel is at a canonical depth, and attempted to solve depth estimation and semantic segmentation jointly.

Recently, deep-learning-based techniques have been developed [5, 6, 19, 28]. Eigen *et al*. [6] proposed combining two deep networks: a coarse network to predict a global depth distribution and a fine network to refine the depth map locally. Eigen and Fergus [5] extended this work to a three-level network structure, and performed surface normal estimation and semantic label estimation, as well as depth estimation. Roy and Todorovic [28] integrated relatively shallow CNNs into a regression forest. Laina *et al*. [19] designed a depth estimation network based the ResNet architecture [11]. They proposed an up-projection structure to improve the resolution of a depth map. Also, for network training, they proposed the Huber loss, which is a combination of the Euclidean function and the L1 function.

Depth estimation techniques, combining CNNs with CRF models, also have been proposed [20, 23, 38]. Wang *et al*. [38] trained a CNN for joint depth estimation and semantic segmentation and adopted a CRF model to improve CNN prediction results. Liu *et al*. [23] proposed a scheme to learn the unary and pairwise potentials of a continuous CRF in a unified CNN framework. Li *et al*. [20] performed the pixel-wise refinement of superpixel-based CNN prediction results through CRF optimization. Chakrabarti *et al*. [3] predicted depth derivatives of different orders probabilistically and then estimated a depth map through a globalization process. Xu *et al*. [39] used multiple continuous CRFs to integrate side output maps of a CNN. This method is similar to the proposed algorithm in that it extracts features from various layers of a network and combines them to estimate a depth map. However, we exploit the intermediate information within the network only without using CRFs.

3. Proposed Algorithm

3.1. Depth Estimation Network

The CNN structure of the proposed algorithm is shown in Figures 1 and 2, which is based on ResNet-152 [11]. Note that ResNet-152 is a very deep network, including 151 convolution layers and 1 fully-connected layer. It is divided into smaller blocks, each of which has three convolution layers, followed by batch normalization and ReLU layers, with a shortcut connection. In Figure 1, B_C denotes this block, where C is the number of channels in the output feature map. The structure of B_C is enclosed by the blue dotted line in Figure 2. The original ResNet-152 contains 50 such blocks. Among these, we modify the last 19 blocks, depicted by green blocks in Figure 1, while maintaining the original structures of the first 31 blocks.

Figure 2 shows the structure of a modified block, which has an additional path for intermediate feature extraction. Let $B_{C,C'}$ denote a modified block, where C' is the number of channels in the feature map, extracted through the additional path. Note that we extract the intermediate feature map from the last ReLU layer of B_C. To this end, we use two convolution layers with 1×3 and 3×1 kernels

sequentially, followed by ReLU layers. The intermediate feature maps from the 19 modified blocks and the feature map from the end of the last $B_{2048,64}$ are all concatenated, as shown in Figure 1. Then, through a fully connected layer, we obtain 800 output responses, each of which corresponds to an estimated depth in a 25×32 depth map.

For training, we adopt a two-phase method. In the first phase, we train the network after removing the additional feature extraction parts and maintaining the original structure of ResNet-152 only. We begin with the ResNet-152 parameters, pre-trained for the image classification task, and fine-tune them using training images and their ground-truth depth maps. In the second phase, we begin with the parameters from the first phase but initialize the parameters of the additional feature extraction parts with Gaussian random values. This two-phase method facilitates faster training and also improves the depth estimation performance, in comparison with learning the entire network from scratch.

3.2. Depth-Balanced Euclidean Loss

In regression problems, the Euclidean loss is often used, which is given by

$$L_{\mathrm{E}} = \frac{1}{2N} \sum_{\mathbf{x}} (\hat{d}_{\mathbf{x}} - d_{\mathbf{x}})^2 \tag{1}$$

where $\mathbf{x}$ denotes the coordinate of a pixel in a depth map, $\hat{d}_{\mathbf{x}}$ is the estimated depth, $d_{\mathbf{x}}$ is the ground-truth depth, and N is the size of the depth map. Let ω denote a network parameter. In case of the Euclidean loss, the update of ω is proportional to the partial derivative

$$\frac{\partial L_{\mathrm{E}}}{\partial \omega} = \sum_{\mathbf{x}} \frac{\partial L_{\mathrm{E}}}{\partial \hat{d}_{\mathbf{x}}} \cdot \frac{\partial \hat{d}_{\mathbf{x}}}{\partial \omega} = \frac{1}{N} \sum_{\mathbf{x}} (\hat{d}_{\mathbf{x}} - d_{\mathbf{x}}) \frac{\partial \hat{d}_{\mathbf{x}}}{\partial \omega}. \tag{2}$$

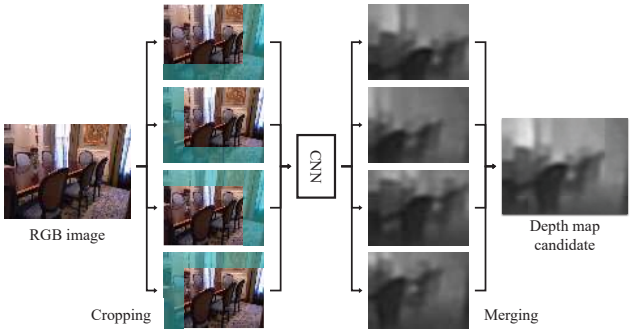
In practice, the absolute estimation error $|\hat{d}_{\mathbf{x}} - d_{\mathbf{x}}|$ tends to be larger, as the ground-truth depth $d_{\mathbf{x}}$ is bigger. For example, 3% depth error of a far object is bigger than the same 3% error of a near object. Thus, the partial derivative in (2) is affected more strongly by the depth estimation errors of distantly located objects than by those of near objects. Therefore, when the Euclidean loss is employed, the network is trained to estimate the depths of distant objects more reliably than those of near ones in general.

To overcome this problem, we propose a new loss, called depth-balanced Euclidean (DBE) loss, given by

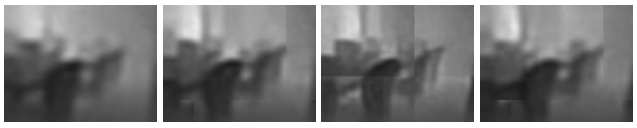
$$L_{\mathrm{DBE}} = \frac{1}{2N} \sum_{\mathbf{x}} \Big(g(\hat{d}_{\mathbf{x}}) - g(d_{\mathbf{x}}) \Big)^2 \tag{3}$$

where g is a quadratic function for the balancing

$$g(d) = a_1 d + \frac{a_2}{2} d^2. \tag{4}$$



(a) Generating a depth map candidate



(b) $\hat{\mathbf{D}}^1$ (c) $\hat{\mathbf{D}}^{0.8}$ (d) $\hat{\mathbf{D}}^{0.6}$ (e) $\hat{\mathbf{D}}^{0.8}_{\mathrm{flip}}$

Figure 3. Depth map candidate generation in (a) and candidate examples in (b)∼(e).

Then, we have

$$\begin{aligned} \frac{\partial L_{\mathrm{DBE}}}{\partial \omega} &= \sum_{\mathbf{x}} \frac{\partial L_{\mathrm{DBE}}}{\partial g(\hat{d}_{\mathbf{x}})} \cdot \frac{\partial g(\hat{d}_{\mathbf{x}})}{\partial \hat{d}_{\mathbf{x}}} \cdot \frac{\partial \hat{d}_{\mathbf{x}}}{\partial \omega} \\ &= \frac{1}{N} \sum_{\mathbf{x}} \Big(g(\hat{d}_{\mathbf{x}}) - g(d_{\mathbf{x}}) \Big) (a_1 + a_2 \hat{d}_{\mathbf{x}}) \frac{\partial \hat{d}_{\mathbf{x}}}{\partial \omega}. \end{aligned} \tag{5}$$

We set a_1 to be a relatively large number and a_2 to be a negative number. Then, in general, for a deeper depth $d_{\mathbf{x}}$, the effect of a bigger error $(g(\hat{d}_{\mathbf{x}}) - g(d_{\mathbf{x}}))$ can be reduced by a smaller factor $(a_1 + a_2 \hat{d}_{\mathbf{x}})$. Thus, the network can be trained to reliably estimate shallow depths, as well as deep depths. Experimental results in Section 4 will also confirm that the proposed DBE loss is more effective for depth estimation than the original Euclidean loss.

3.3. Depth Map Candidate Generation

Using the proposed CNN trained with the DBE loss, we generate multiple depth map candidates for an input image. Figure 3(a) illustrates how to generate a depth map candidate. First, we crop the input image at the four corners, respectively, with a cropping ratio r. The cropping ratio is defined as the size ratio of the cropped image to the entire image. Second, we process each cropped image through the CNN to yield the corresponding depth map. Finally, we merge these four partially estimated depth maps into a single depth map candidate. In the merging process, note that all depth values should be scaled by a factor of $1/r$ to compensate for the zooming effect that objects in a cropped image look closer. After the scaling, the partial depth maps are translated to their positions and then superposed. For the superposition in overlapping regions, the averaging is

performed. Let $\hat{\mathbf{D}}^r$ denote the resultant depth map candidate. When $r = 1$, $\hat{\mathbf{D}}^1$ is simply obtained by processing the entire input image through the CNN.

Since the CNN parameters are not symmetric, a flipped image does not yield the flipped depth map. Therefore, we horizontally flip an input image, obtain a depth map candidate with a cropping ratio r, and flip back the depth map candidate. This is denoted as $\hat{\mathbf{D}}^r_{\text{flip}}$. Figure 3(b)∼(e) shows examples of depth map candidates.

3.4. Candidate Combination in Fourier Domain

As illustrated in Figure 3, in general, a depth map candidate $\hat{\mathbf{D}}^r$ with a larger cropping ratio r reconstructs the overall depth distribution more reliably, whereas that with a smaller r estimates local details more accurately. To exploit these complementary properties, we combine depth map candidates in the Fourier frequency domain. Notice that the overall distribution and the local details correspond to low and high frequency coefficients, respectively.

The discrete Fourier transform (DFT) [26] of an input signal $I(x, y)$ of size $M \times N$ is given by

$$F(u,v) = \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} I(x,y) \exp\left(-i2\pi\left(\frac{xu}{N} + \frac{yv}{M}\right)\right) \quad (6)$$

where u and v are horizontal and vertical frequencies.

We transform each depth map candidate and rearrange the 2D-DFT coefficients into a column vector. During the rearrangement, we remove two kinds of redundancy. First, DFT is periodic, *i.e.* $F(u, v) = F(u + Nk, v + Ml)$ for all $k, l \in \mathbb{Z}$. Second, since a depth map is real, its DFT is conjugate symmetric, *i.e.* $F(u, v) = F^*(-u, -v)$. Let $\hat{\mathbf{f}}^m$ denote the rearranged DFT vector of the m-th depth map candidate. Also, let $\hat{\mathbf{f}}$ be the DFT vector of the combined depth map of all candidates, and $\mathbf{f}$ be that of the ground-truth depth map. Also, $\hat{f}^m_k$, $\hat{f}_k$, and f_k denote the k-th coefficients in $\hat{\mathbf{f}}^m$, $\hat{\mathbf{f}}$, and $\mathbf{f}$, respectively. We obtain $\hat{f}_k$ as follows.

$$\hat{f}_k = \sum_{m=1}^{M} w^m_k (\hat{f}^m_k - b^m_k) \quad (7)$$

where w^m_k is a weighting parameter, b^m_k is a bias, and M is the number of depth map candidates.

First, the bias b^m_k should compensate for the average deviation of $\hat{f}^m_k$ from the ground-truth f_k. Hence, we determine this bias, using the training dataset, by

$$b^m_k = \frac{1}{T} \sum_{t=1}^{T} (\hat{f}^m_{kt} - f_{kt}) \quad (8)$$

where t is the index of a training image, and T is the total number of images in the training dataset. Also, $\hat{f}^m_{kt}$ and f_{kt} denote $\hat{f}^m_k$ and f_k for the tth image, respectively.

Second, we determine the weighting parameters w^m_k in (7) to minimize the mean squared error (MSE) between $\hat{f}_k$ and f_k. To this end, we define a matrix $\mathbf{T}_k$, in which the (t, m)th element equals to $\hat{f}^m_{k,t} - b^m_k$. We also define the ground-truth vector $\mathbf{t}_k = [f_{k1}, \cdots, f_{kT}]'$. Then, the MSE minimization problem is to find the optimal weight vector $\mathbf{w}_k = [w^1_k, \cdots, w^M_k]'$, given by

$$\mathbf{w}_k = \arg\min_{\mathbf{w}} \|\mathbf{T}_k \mathbf{w} - \mathbf{t}_k\|^2. \quad (9)$$

This can be solved using the pseudo-inverse of $\mathbf{T}_k$,

$$\mathbf{w}_k = \mathbf{T}^+_k \mathbf{t}_k. \quad (10)$$

We repeat this process for all k to determine all weight and bias parameters.

In testing, we combine the DFT vectors of multiple depth map candidates $\hat{\mathbf{f}}^m$ into that of the final estimate $\hat{\mathbf{f}}$ via (7). Then, we perform the inverse Fourier transform to generate the final estimated depth map $\hat{\mathbf{D}}$. It is worth pointing out that, because of the Parseval's relation [26], minimizing the MSE in the frequency domain is equivalent to the minimizing the MSE in the spatial domain. In other words, no other combination of $\hat{\mathbf{f}}^m$ can lead to a smaller MSE, $\|\hat{\mathbf{D}} - \mathbf{D}\|$, between the estimated and ground-truth depth maps.

4. Experimental Results

4.1. Implementation Details

We use the NYUv2 depth dataset [33], which contains about 280,000 training images. To train the proposed CNN, we perform the data augmentation with the scale, rotation, color, and horizontal flip transformations [6]. The NYUv2 dataset also provides 654 separate test images. For the testing, we extract only the central area of size 427×561 from each image, as done in [3, 28].

In the training, we use the NAG solver [25, 35]. For the DBE loss in (3) and (4), we set $a_1 = 1.5$ and $a_2 = -0.1$. As mentioned in 3.1, we adopt the two-phase training method. In the first phase, we initialize the parameters with those of the pre-trained ResNet-152 [11]. Then, we perform the iterative training 500,000 times with a batch size of 4 and a learning rate of 0.00016. In the second phase, we reduce the learning rate by a factor of 10^{-3} for the existing parts. For the additional feature extraction parts, we adopt the Xavier initialization [8] and set the learning rate to 0.00016. The batch size of 4 is also maintained in the second phase.

For each image, we generate 9 depth map candidates with cropping ratios $r \in \{0.60, 0.65, ..., 1.00\}$. We also use a flipped candidate for each r. Therefore, we obtain 18 candidates in total.

4.2. Performance Comparison

We use the five performance metrics in [3, 6], given by

Table 1. Performance comparison of the proposed algorithm and the conventional algorithms. The best results are boldfaced, and the second best ones are underlined.

	The lower, the better				The higher, the better		
	RMSE (lin)	RMSE (log)	Abs Rel	Sqr Rel	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zoran *et al*. [43]	1.220	0.430	0.410	0.570	-	-	-
Li *et al*. [20]	0.821	-	0.232	-	62.1%	88.6%	96.8%
Liu *et al*. [21]	0.824	-	0.230	-	61.4%	88.3%	97.1%
Baig *et al*. [1]	0.802	-	0.241	-	61.0%	-	-
Eigen *et al*. [6]	0.877	0.283	0.214	0.204	61.4%	88.8%	97.2%
Wang *et al*. [38]	0.745	0.262	0.220	0.210	60.5%	89.0%	97.0%
Roy *et al*. [28]	0.744	-	0.187	-	-	-	-
Eigen and Fergus [5]	0.641	0.214	0.158	0.121	76.9%	95.0%	98.8%
Chakrabarti *et al*. [3]	0.620	0.205	0.149	0.118	80.6%	95.8%	98.7%
Laina *et al*. [19]	0.597	0.204	0.140	0.106	81.1%	95.3%	98.8%
Proposed	**0.572**	**0.193**	**0.139**	**0.096**	**81.5%**	**96.3%**	**99.1%**

$$\text{RMSE (lin)}: \quad \left(\frac{1}{N}\Sigma_{\mathbf{x}}(\hat{d}_{\mathbf{x}} - d_{\mathbf{x}})^2\right)^{\frac{1}{2}}$$

$$\text{RMSE (log)}: \quad \left(\frac{1}{N}\Sigma_{\mathbf{x}}(\log \hat{d}_{\mathbf{x}} - \log d_{\mathbf{x}})^2\right)^{\frac{1}{2}}$$

$$\text{Abs Rel}: \quad \frac{1}{N}\Sigma_{\mathbf{x}}\frac{|\hat{d}_{\mathbf{x}} - d_{\mathbf{x}}|}{d_{\mathbf{x}}}$$

$$\text{Sqr Rel}: \quad \frac{1}{N}\Sigma_{\mathbf{x}}\frac{|\hat{d}_{\mathbf{x}} - d_{\mathbf{x}}|^2}{d_{\mathbf{x}}}$$

$$\delta < t: \quad \text{Percentage of } d_{\mathbf{x}} \text{ such that } \max\{\frac{\hat{d}_{\mathbf{x}}}{d_{\mathbf{x}}}, \frac{d_{\mathbf{x}}}{\hat{d}_{\mathbf{x}}}\} < t \text{ where } t = 1.25, 1.25^2 \text{ or } 1.25^3$$

Here, $d_{\mathbf{x}}$ and $\hat{d}_{\mathbf{x}}$ denote ground-truth and estimated depths, respectively. Also N is the total number of pixels in all images in the test dataset.

In Table 1, we compare the proposed algorithm with the recent conventional algorithms [1, 3, 5, 6, 19–21, 28, 38, 43]. In [19], their performance was computed with the single precision. The single precision is not sufficient in case of the RMSE and Rel metrics, since a huge number of depth differences should be summed up. Therefore, we measure their performance again in the double precision using their source codes. We see that, in terms of all metrics, the proposed algorithm provides the best results, outperforming the second best results considerably.

Figure 4 compares depth maps qualitatively. For easier comparison, we also visualize the errors in the depth maps. Bigger errors are represented by bright yellow colors, while smaller errors are by dark red colors. It is observed that the proposed algorithm estimates the depth information accurately and reliably and also reduces blur artifacts in comparison with the conventional algorithms.

4.3. Ablation Study

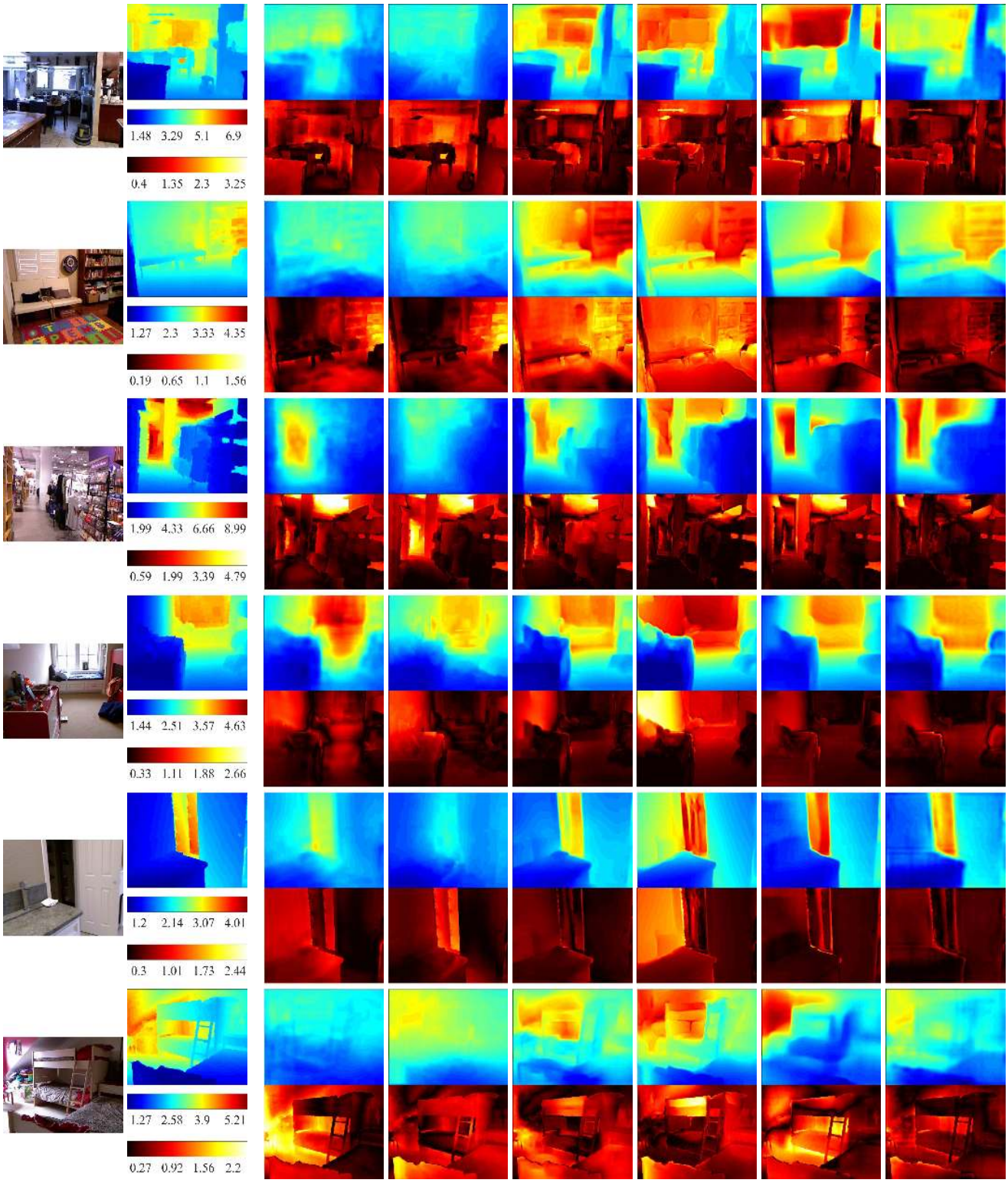
We develop the depth estimation network (DEN) in Figures 1 and 2, based on the ResNet-152 [11] structure. Table 2 compares the depth estimation performances of DEN with those of ResNet-152, which is fine-tuned using the NYUv2 dataset. We see that the modification of the structure leads to the performance improvement, *e.g*. in case of 'RMSE (lin)' from 0.597 to 0.586.

Table 2. 'RMSE (lin),' 'Abs Rel,' and '$\delta < 1.25$' performances of the proposed algorithm in various settings. The baseline ResNet-152 is also included for comparison. DEN: depth estimation network, DBE: DBE loss, and FDC(k): Fourier domain combination of k depth map candidates. Thus, 'DEN+DBE+FDC(18)' is the complete proposed algorithm.

	RMSE	Rel	$\delta < 1.25$
ResNet-152	0.597	0.148	79.6%
DEN	0.586	0.145	80.3%
DEN+DBE	0.585	0.142	81.3%
DEN+DBE+FDC(2)	0.594	**0.139**	81.2%
DEN+DBE+FDC(6)	0.581	**0.139**	81.4%
DEN+DBE+FDC(10)	0.576	**0.139**	**81.5%**
DEN+DBE+FDC(18)	**0.572**	**0.139**	**81.5%**

In addition to the DEN structure, this work has two contributions. First, we use the DBE loss to estimate shallow depth more reliably. Second, we perform the Fourier domain combination (FDC) of multiple depth map candidates. Note that each of these two components improves the performances meaningfully. As a result, the complete proposed algorithm 'DEN+DBE+FDC(18)' provides the state-of-the-art performances. Figure 5 also shows that each component of the proposed algorithm contributes to the reconstruction of a high quality depth map.

In the complete proposed algorithm, for each image, 18 depth map candidates are generated by varying the cropping ratio from 0.60 to 1.00 in 0.05 increments. We can reduce the number k of candidates to 10, 6, or 2, respectively, by raising the lower bound of r to 0.8, 0.9 or 1.0. In Table 2, FDC(k) means the combination of k candidates. In general,



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

Figure 4. Comparison of estimated depth maps. Upper images show depth maps, and lower images are the corresponding error maps. (a) Input images, (b) ground-truth depth maps and color coding schemes, (c) Eigen *et al*. [6], (d) Wang *et al*. [38], (e) Eigen and Fergus [5], (f) Chakrabarti *et al*. [3], (g) Laina *et al*. [19], and (h) the proposed algorithm.

Table 3. The two components DBE and FDC of the proposed algorithm improve the depth estimation performance of three popular networks: AlexNet, VGG19, and ResNet-50.

	RMSE	Rel	$\delta < 1.25$
AlexNet	0.836	0.244	60.4%
AlexNet+DBE	0.870	**0.243**	60.3%
AlexNet+DBE+FDC(18)	**0.826**	0.247	**61.5%**
VGG19	**0.616**	0.163	76.6%
VGG19+DBE	0.619	0.158	76.9%
VGG19+DBE+FDC(18)	0.617	**0.157**	**77.0%**
ResNet-50	**0.591**	0.151	79.4%
ResNet-50+DBE	0.603	0.147	**79.6%**
ResNet-50+DBE+FDC(18)	0.597	**0.145**	79.3%

Table 4. Detail reconstruction performance of FDC using SSIM and NCC scores on the NYUv2 dataset.

	SSIM	NCC
DEN+DBE	0.960	0.890
DEN+DBE+FDC(18)	**0.961**	**0.897**
AlexNet+DBE	0.938	0.758
AlexNet+DBE+FDC(18)	**0.939**	**0.781**
VGG19+DBE	0.957	0.875
VGG19+DBE+FDC(18)	**0.958**	**0.882**
ResNet-50+DBE	0.958	0.881
ResNet-50+DBE+FDC(18)	**0.959**	**0.888**

the performance improves as k gets larger. However, the performance saturates when k is bigger than 18. Note that FDC(k) with a relative small $k = 6$ still yields competitive results.

Next, we replace the DEN with three popular networks with fewer layers: AlexNet [16], VGG19 [34] and ResNet-52 [11]. All these networks are pre-trained for the image classification task. In the same way as we develop DEN, we modify each of these three networks. Specifically, we modify the last fully-connected layer to yield 800 output responses, which correspond to a 25×32 depth map. Then, we fine-tune the modified network using training images and their ground-truth depth maps. Then, as in the proposed algorithm, we incorporate the two components DBE and FDC sequentially into the network. Table 3 shows that each component also contributes to performance improvement in the networks with fewer parameters. Also in Table 4, we measure SSIM and NCC scores to see how well the FDC reconstructs the depth map details. We confirm that FDC faithfully restores the structure of depth maps of all the networks.

4.4. Performance Analysis in Fourier Domain

Let us analyze estimated depth maps in the frequency domain. To this end, we classify 2D frequencies into groups according to their magnitudes: a frequency group contains 2D frequencies, the magnitudes of which range from 2^{n-1} to 2^n Hz. Figure 6 shows the 'RMSE (lin)' and 'Abs Rel' errors of reconstructed depths, when all frequency coefficients in each group are set to zero. An exception is the DC coefficient case, whose impacts are much bigger than the other AC coefficients. To plot their impacts in the same graph, we replace the DC coefficient with the overall mean depth of the test images, instead of zero. It is observable from Figure 6 that a lower frequency group has greater impacts on the reconstruction performance of depth maps in general. However, relatively high frequency groups ($< 2^6$) also contribute to depth maps, since they represent local details and depth discontinuities. If the frequency magnitudes are higher than 2^6, their impacts are negligible.

In Figure 7, we analyze the MSE between the frequency coefficients of an estimated depth map and its ground-truth. Since there are large variations in error scales across different frequency groups, we plot relative errors by setting the errors of ResNet-152 as the reference points.

Figure 7(a) shows the contribution of each component of the proposed algorithm. The DEN structure reduces errors in low frequency groups. The DBE loss makes the estimation of DC components more accurate, while increasing errors in higher frequency groups. However, DBE improves the overall performance by estimating shallow depths more reliably. The FDC of multiple depths improves the estimation performance in most frequency groups. In particular, the improvements around $2^1 \sim 2^4$ Hz are remarkable.

Figure 7(b) compares the proposed algorithm with the state-of-the-art conventional algorithms [3, 5, 19]. The proposed algorithm yields the best performances in most frequency groups. Especially, from 2^1 to 2^4 Hz, corresponding to fine-grain details of depth maps, the proposed algorithm reduces errors about 5%, in comparison with the conventional algorithms.

Figure 7(c) shows the results when the proposed FDC technique is combined with the Laina *et al.*'s algorithm [19]. FDC also improves the Laina *et al.*'s algorithm [19] in range of $2^1 \sim 2^5$ Hz. This indicates that FDC is independent of the network structure and thus can be combined with various CNN-based depth estimators to improve the fine-grain details of depth maps.

Finally, Figure 7(d) shows the results when changing the number k of candidates to 2, 6, 10 or 18 of the FDC. The increase of k leads to a performance improvement in all frequency groups. In particular, improvement in the high frequency group is remarkable.

5. Conclusions

We proposed a CNN-based single-image depth estimator, which generates multiple depth map candidates and combines these candidates in the Fourier frequency domain. Specifically, we developed the CNN structure based

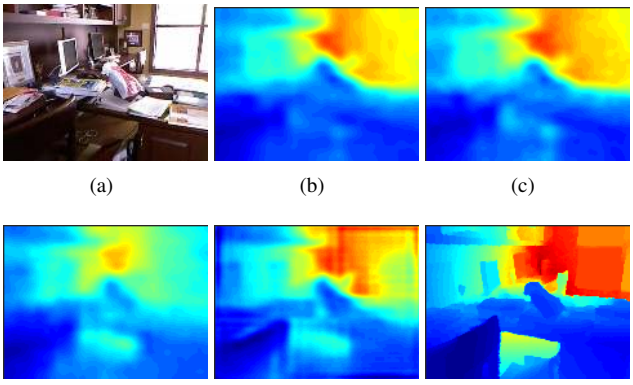


Figure 5. Examples of depth maps, obtained by the proposed algorithm in various settings: (a) input image, (b) ResNet-152, (c) DEN, (d) DEN+DBE, (e) DEN+DBE+FDC(18), and (f) ground-truth.

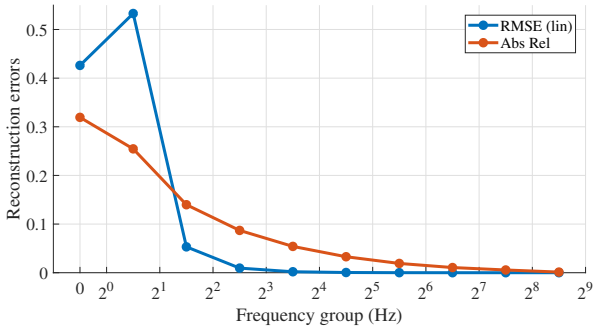


Figure 6. Reconstruction errors of depth maps when all frequencies in each frequency group are set to zero.

on ResNet-152, and introduced the DBE loss to train the network reliably for a wide range of depths. Also, we generated multiple depth map candidates by cropping an input image with various cropping ratios. Then, to exploit the complementary properties of different depth map candidates, we combined them in the Fourier domain. Experimental results demonstrated that proposed algorithm outperforms the conventional algorithms significantly. Moreover, the Fourier domain analysis validated that each component of the proposed algorithm contributes to the reduction of estimation errors in most frequency bands.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2015R1A2A1A10055037, NRF-2018R1A2B3003896), and in part by the Agency for Defense Development (ADD) and Defense Acquisition Program Administration (DAPA) of Korea (UC160016FD).

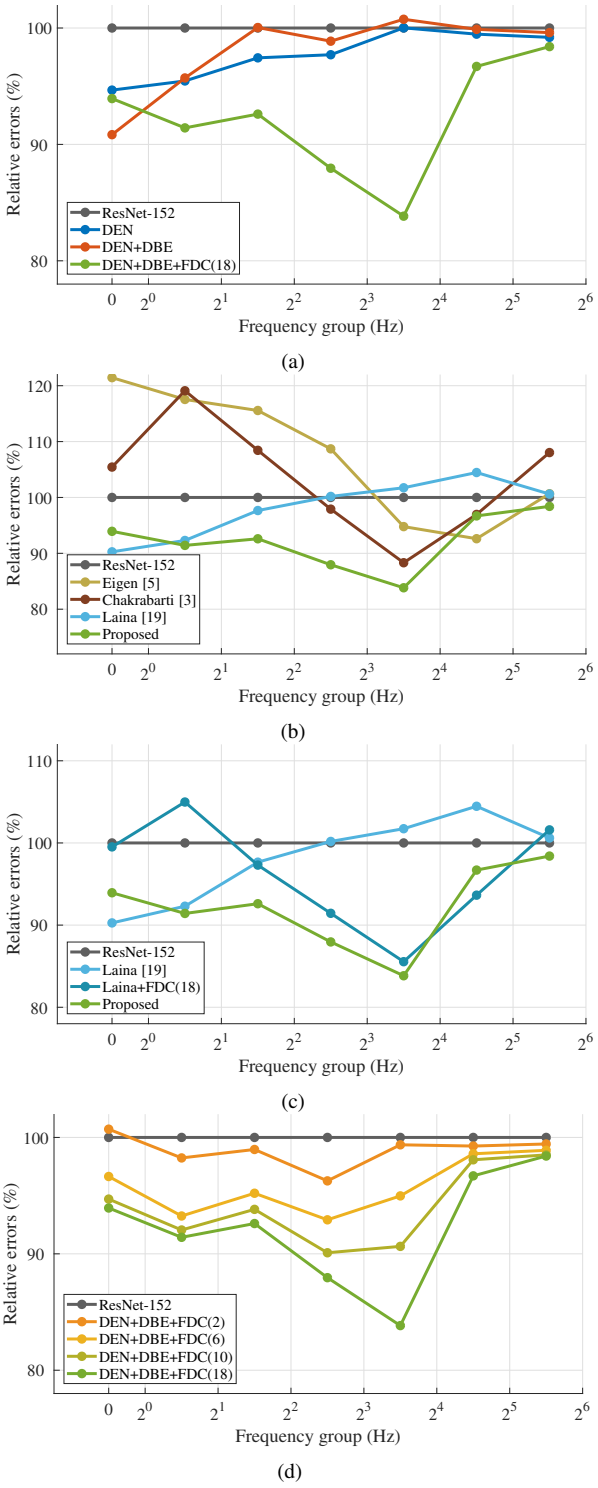


Figure 7. Frequency domain analysis of relative depth errors: (a) component analysis of the proposed algorithm, (b) comparison with the conventional algorithms, (c) combination of the proposed FDC technique with the Laina *et al.*'s algorithm [19], and (d) component analysis according to the candidate number in FDC.

References

[1] M. Baig and L. Torresani. Coupled depth learning. In *Proc. IEEE WACV*, pages 1–10, Mar. 2016.

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, pages 44–57, Oct. 2008.

[3] A. Chakrabarti, J. Shao, and G. Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *NIPS*, pages 2658–2666, Dec. 2016.

[4] E. Delage, H. Lee, and A. Y. Ng. A dynamic Bayesian network model for autonomous 3D reconstruction from a single indoor image. In *Proc. IEEE CVPR*, pages 2418–2428, Jun. 2006.

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. IEEE ICCV*, pages 2650–2658, Dec. 2015.

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, Dec. 2014.

[7] A. Flint, D. W. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3D features. In *Proc. IEEE ICCV*, pages 2228–22235, Nov. 2011.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. AISTATS*, pages 249–256, Mar. 2010.

[9] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, pages 482–496, Sep. 2010.

[10] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2341–2353, Dec. 2011.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, Jun. 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, Oct. 2016.

[13] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584, Jul. 2005.

[14] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *ECCV*, pages 775–788, Oct. 2012.

[15] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2144–2158, Oct. 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, Dec. 2012.

[17] A. Kundu, Y. Li, F. Daellert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *ECCV*, pages 703–718, Sep. 2014.

[18] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proc. IEEE CVPR*, pages 89–96, Jun. 2014.

[19] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. IEEE 3DV*, pages 239–248, Oct. 2016.

[20] 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. IEEE CVPR*, pages 1119–1127, Jun. 2015.

[21] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proc. IEEE CVPR*, pages 5162–5170, Jun. 2015.

[22] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, Oct. 2016.

[23] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, Oct. 2016.

[24] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Proc. IEEE CVPR*, pages 716–723, Jun. 2014.

[25] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k2)$. *Soviet Mathematics Doklady*, 27(2):372–376, Feb. 1983.

[26] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 1989.

[27] X. Ren, L. Bo, and D. Fox. RGB-D scene labeling: Features and algorithms. In *Proc. IEEE CVPR*, pages 2759–2766, Jun. 2012.

[28] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proc. IEEE CVPR*, pages 5506–5514, Jun. 2016.

[29] A. Saxena, M. Sun, and A. Y. Ng. 3-D depth reconstruction from a single still image. *Int. J. Comput. Vis.*, 76(1):53–69, Oct. 2008.

[30] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3-D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):824–840, Oct. 2009.

[31] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.*, 47:7–42, Apr. 2002.

[32] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124, Jan. 2013.

[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, pages 746–760, Oct. 2012.

[34] K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. [Online]. Available: https://arxiv.org/abs/1409.1556.

[35] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147, Feb. 2013.

[36] S. Suwajanakorn and C. Hernandez. Depth from focus with your mobile phone. In *Proc. IEEE CVPR*, pages 3497–3506, Jun. 2015.

[37] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for oneshot human pose estimation. In *Proc. IEEE CVPR*, pages 103–110, Jun. 2012.

[38] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proc. IEEE CVPR*, pages 2800–2809, Jun. 2015.

[39] 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. IEEE CVPR*, pages 5354–5362, Jun. 2017.

[40] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, pages 756–771, Sep. 2014.

[41] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, Aug. 1999.

[42] W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *Proc. IEEE CVPR*, pages 614–622, Jun. 2015.

[43] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *Proc. IEEE ICCV*, pages 388–396, Dec. 2015.