This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

Monocular Depth Estimation Using Whole Strip Masking and Reliability-Based Refinement

Minhyeok Heo[1], Jaehan Lee[2], Kyung-Rae Kim[2],
Han-Ul Kim[2], and Chang-Su Kim[2]

[1] NAVER LABS
heo.minhyeok@naverlabs.com
[2] School of Electrical Engineering, Korea University, Korea
{jaehanlee,krkim,hanulkim}@mcl.korea.ac.kr, changsukim@korea.ac.kr

Abstract. We propose a monocular depth estimation algorithm based on whole strip masking (WSM) and reliability-based refinement. First, we develop a convolutional neural network (CNN) tailored for the depth estimation. Specifically, we design a novel filter, called WSM, to exploit the tendency that a scene has similar depths in horizonal or vertical directions. The proposed CNN combines WSM upsampling blocks with a ResNet encoder. Second, we measure the reliability of an estimated depth, by appending additional layers to the main CNN. Using the reliability information, we perform conditional random field (CRF) optimization to refine the estimated depth map. Experimental results demonstrate that the proposed algorithm provides the state-of-the-art depth estimation performance.

Keywords: Monocular depth estimation, whole strip masking, reliability, depth map refinement

1 Introduction

Estimating depth information from images is a fundamental problem in computer vision [1–3]. Humans can infer depths with ease, since we intuitively use various cues and have an innate sense. However, it is very challenging to imitate this ability computationally. Especially, in comparison with stereo matching [4] and video-based approaches, monocular (or single-image) depth estimation is even more difficult due to the lack of reliable visual cues, such as the disparity between matching points.

Early studies for monocular depth estimation attempted to compensate for this lack of information. Some techniques depend on scene assumptions, *e.g.* box models [5] and typical indoor rooms [6], which make the techniques useful for limited situations only. Some use additional data, *e.g.* user annotations [7] and semantic labels [8], which are not always available. Also, hand-crafted features based on geometric and semantic cues were designed [9–11]. For example, since a depth map often has similar values in horizontal or vertical directions, an

elongated rectangular patch was used in [9]. However, these hand-crafted features have become obsolete and replaced by machine learning approaches recently.

As labeled data increase, many data-based techniques have been proposed. In [12], a depth map was transferred from aligned candidates in an image pool. More recently, many convolutional neural networks (CNNs) have been proposed for monocular depth estimation [13–19]. They learn features to represent depths automatically and implicitly, without requiring the traditional feature engineering. Also, several techniques combine CNNs with conditional random field (CRF) optimization to improve the accuracy of a depth map [15–18].

In this work, we propose a novel CNN-based algorithm, which achieves accurate depth estimation by exploiting the characteristics of depth information to a greater extent. First, we develop a novel upsampling block, referred to as the whole strip masking (WSM), to exploit the tendency that depths are flat horizontally or vertically in scenes. We estimate a depth map by cascading these upsampling blocks together with the deep network ResNet [20]. Second, we use the notion of reliability of an estimated depth. Specifically, we measure the reliability (or confidence) of the estimated depth of each pixel and use the information to define unary and pairwise potentials of a CRF. Through the reliability-based CRF optimization, we refine the estimated depth map and improve its accuracy. We highlight our main contributions as follows:

- We propose a deep CNN with the novel WSM upsampling blocks for monocular depth estimation.
- We measure the reliability of an estimated depth and use the information for the depth refinement.
- The proposed algorithm yields the state-of-the-art depth estimation performance, outperforming conventional algorithms [8, 12–19, 21] significantly.

2 Related Work

Before the widespread adoption of CNNs, hand-crafted features had been used to estimate the depth information from a single image. An early method, proposed by Saxena *et al.* [9], adopted a Markov random field (MRF) model to predict the depth from multi-scale patches and a column patch of a vertically long shape. Saxena *et al.* [10] also predicted the depth, by assuming that a scene consists of small planes and inferring the set of plane parameters. Liu *et al.* [11] estimated the depth based on class-related depth and geometry priors, obtained through semantic segmentation. Assuming that semantically similar images have similar depth distributions, Karsch *et al.* [12] extracted a depth map by finding similar images from a database and warping them.

Recently, with the remarkable success of deep learning in many applications [22–24], various CNN-based methods for monocular depth estimation have been proposed. Eigen *et al.* [13] first applied a CNN to monocular depth estimation. They predicted a coarse depth map based on AlexNet [25] and refined it with another network in a fine scale. Eigen and Fergus [14] replaced AlexNet with the deeper VGGNet [26] and used the common network to predict depths,

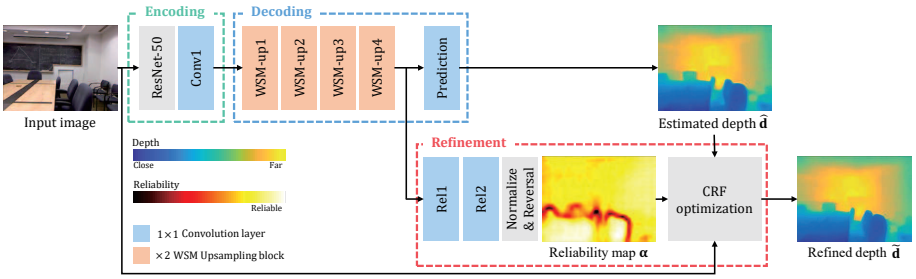


Fig. 1: Overview of the proposed depth estimation algorithm.

semantic labels, and surface normals jointly. Laina *et al.* [19] improved the depth estimation performance by combining upsampling blocks with ResNet [20], which is about three times deeper than VGGNet. Also, Lee *et al.* [27] introduced the notion of Fourier domain analysis into monocular depth estimation. These methods have gradually improved the estimation performance by adopting deeper networks in general. However, they often yield blurry depth maps.

Sharper depth maps can be obtained by combining CNNs with CRF optimization. Liu *et al.* [15] proposed a superpixel-based algorithm, which divides an image into superpixels and learns unary and pairwise potentials of a CRF during the network training. Li *et al.* [17] adopted hierarchical CRFs. They estimated depths at a superpixel level and then refined them at a pixel level. Also, Wang *et al.* [16] proposed a CNN for joint depth estimation and semantic segmentation, and refined a depth map using a two-layer CRF. These CNN-based methods [13–17, 19] provide decent depth maps. In this work, by exploiting the characteristics of depth information to a greater extent, as well as by adopting the merits of the conventional methods, we attempt to further improve the depth estimation performance.

3 Proposed Algorithm

Fig. 1 is an overview of the proposed monocular depth estimation algorithm. We first encode an input image into a feature vector based on the ResNet-50 architecture [20]. We then decode the feature vector using four WSM upsampling blocks. Then, we use the decoded result for two purposes: 1) to estimate the depth map $\widehat{\mathbf{d}}$ and 2) to obtain the reliability map $\boldsymbol{\alpha}$. Finally, we perform the CRF optimization using $\boldsymbol{\alpha}$ to process $\widehat{\mathbf{d}}$ into the refined depth map $\widetilde{\mathbf{d}}$.

3.1 Depth Map Estimation

Most CNNs for generating a high-resolution image (or map) as the output are composed of encoding and decoding parts. The encoding part decreases the spatial resolution of an input image through pooling or convolution layers with

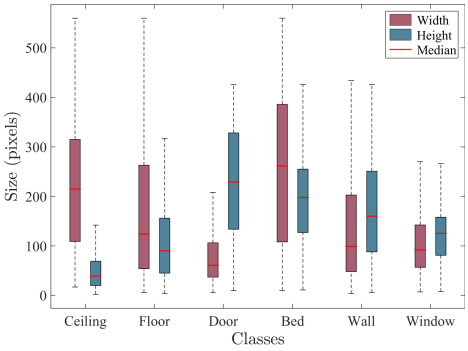
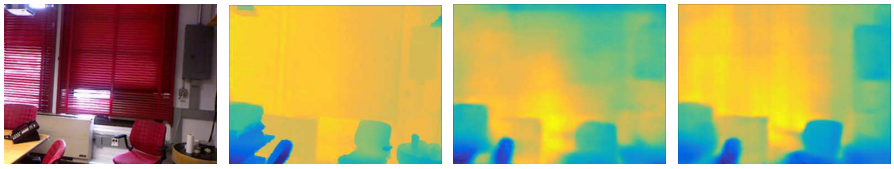


Fig. 2: The width and height distributions of six object classes, which are often observed in indoor scenes. A central red line indicates the median, and the bottom and top edges of a box indicate the 1^{st} and 3^{rd} quartiles.

strides. For the encoding part, in general, pre-trained networks on a very large dataset, *e.g.* ImageNet [28], are used without modification or fine-tuned with a smaller dataset to speed up the learning and alleviate the need for a large training dataset for each specific task. On the other hand, the decoding part processes input activations to yield a higher-resolution output map using unpooling layers or deconvolution layers. In other words, the encoder contracts a signal, whereas the decoder expands a signal. It is known that the contraction enables a network to have a theoretically large receptive field without demanding unnecessarily many parameters [29]. Also, as a network depth increases, the receptive field gets larger. Therefore, recent deep networks, such as VGGNet and ResNet-50, have theoretical receptive fields larger than input image sizes [29, 30].

However, even in the case of a deep CNN, the effective range is smaller than the theoretical receptive field. Luo *et al.* [30] observed that not all pixels in the receptive field affect an output response meaningfully. Thus, the information in a local image region only is used to yield a response. This is undesirable especially in the depth estimation task, which requires global information to estimate the depth of each pixel. Note that depths in a typical image exhibit very strong horizontal or vertical correlations. In Fig. 2, we analyze the width and height distributions of six object classes, which are observed in indoor scenes in the NYU Depth Dataset V2 [31], in which the semantic labels are available. For instance, a ceiling is horizontally wide, while a door is vertically long. Also, the average depth variation within such an object is very small, less than 0.3. Hence, to estimate the depth of a pixel reliably, all information in the entire rows or columns within an image is required. The limited effective receptive fields of conventional CNNs may degrade the depth estimation performance.

To overcome this problem, we propose a novel filter, called WSM, for up-sampling blocks. Note that a typical convolution layer performs zero-padding to maintain the same output resolution as the input resolution and uses a square kernel of a small size, *e.g.* 1×1, 3×3, or 5×5. Thus, an output value of the typical



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

Fig. 3: The efficacy of WSM layers: (a) an image, (b) its ground-truth depths, (c) estimated depths using convolution layers only, and (d) estimated depths using both convolution and WSM layers.

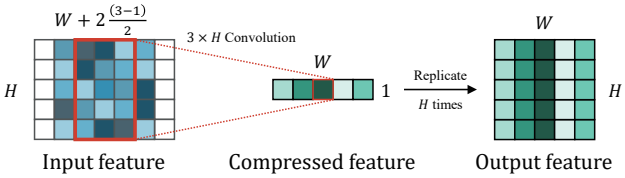


Fig. 4: Illustration of the proposed $3 \times H$ WSM layer.

convolution layer merges only the local information of the input feature. Hence, in Fig. 3(c), although the wall has similar features and depths, the estimation result of a network using convolution layers only does not yield flat depths on the wall. In contrast, to consider horizontally or vertically flat characteristics of depth maps, the proposed WSM adopts long rectangular kernels and replicates the kernel responses in the horizontal or vertical direction. Consequently, as shown in Fig. 3(d), the proposed WSM facilitates more faithful reconstruction of vertically flat depths on the wall.

Suppose an input feature of spatial resolution $W \times H$. Fig. 4 shows the $3 \times H$ WSM layer. We first apply zero-padding in the horizontal direction only. Then, we perform the horizontal convolution using the $3 \times H$ mask, which yields a compressed feature map of size $W \times 1$. This compressed feature map summarizes the information in the vertical strips of the input feature map and is forced to have the largest receptive field in the vertical direction. Next, we replicate the compressed feature to yield the output feature map that has the same size as the input. As a result, each response in the output feature map combines all information in the corresponding vertical strip, and all responses in the same column have an identical value. The $W \times 3$ WSM is also performed similarly.

We use both $3 \times H$ and $W \times 3$ WSM layers in each upsampling block in Fig. 1. Note that the proposed upsampling is also referred to as the WSM upsampling. However, it has some limitations to use only the WSM layers in the upsampling. First, it is important to exploit local information, as well as global information, when estimating depths. Second, a great number of parameters are required for the large $3 \times H$ and $W \times 3$ masks. To alleviate these limitations, we adopt the inception structure in [32]. The inception structure merges the results of

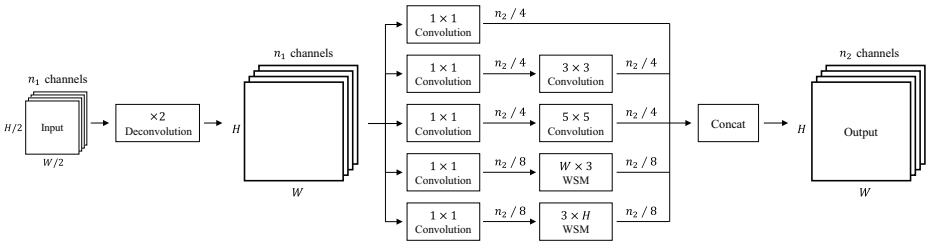


Fig. 5: The structure of the proposed WSM upsampling block.

various convolutions of different kernel sizes, but applies 1×1 convolution layers first to lower the dimension of the input feature and thus reduce the number of parameters. By incorporating the WSM layers into the inception structure, the proposed WSM upsampling attempts to maximize the network capacity and integrate both global and local information, while requiring a moderate number of parameters. Fig. 5 shows the WSM upsampling block. First, we double the spatial resolution of a feature map using a deconvolution layer. Then, we adopt 1×1 convolution layers to lower the feature dimension, before applying the conventional 3×3 and 5×5 convolutional layers and the proposed $W \times 3$ and $3 \times H$ WSM layers. We concatenate all results to yield the output feature map.

The WSM upsampling is employed by the entire network in Fig. 1. We use the ResNet-50 architecture in the encoding step, but remove the last two fully-connected layers and instead add a 1×1 convolution layer to lower the feature dimension since the last convolution layer of ResNet-50 yields a relative high feature dimension. For the decoding step, we cascade four WSM upsampling blocks to increase the output spatial resolution to 160×128. Finally, through a 1×1 convolution layer, we obtain an estimated depth map $\widehat{\mathbf{d}}$. To train the network in an end-to-end manner, we adopt the Euclidean loss to minimize the sum of squared differences between the ith estimated depth $\hat{d}_i$ and the corresponding ground truth d_i^{gt}. Table 1 presents detailed network configurations.

3.2 Depth Map Refinement

As shown in Fig. 6, even though the proposed depth estimation provides a promising result, the estimated depth map $\hat{\mathbf{d}}$ still contains residual errors especially around object boundaries. In a wide variety of estimation problems, attempts have been made not only to make an estimate, but also to measure the reliability or confidence (or inversely uncertainty) of the estimate. For example, in the classical depth-from-motion technique in [33], Matthies *et al.* predicted depth and depth uncertainty at each pixel and incrementally refined the estimates to reduce the uncertainty. In this work, we observe that the reliability of an estimated depth can be quantified, surprisingly, using the same features from the decoder for the depth estimation itself, as shown in Fig. 1.

We augment the network to learn the reliability. In Fig. 1, the reliability map is obtained by adding only two 1×1 convolution layers 'Rel1' and 'Rel2' after

Table 1: Configurations of the proposed network. Input and output sizes are given by $W \times H \times C$, where W, H, and C are the width, height, and number of channels, respectively.

	Layer Name	Input	Input Size	Output Size
Encoding	ResNet-50	Image	$304 \times 228 \times 3$	$10 \times 8 \times 2048$
	Conv1	ResNet-50	$10 \times 8 \times 2048$	$10 \times 8 \times 1024$
Decoding	WSM-up1	Conv1	$10 \times 8 \times 1024$	$20 \times 16 \times 1024$
	WSM-up2	WSM-up1	$20 \times 16 \times 1024$	$40 \times 32 \times 512$
	WSM-up3	WSM-up2	$40 \times 32 \times 512$	$80 \times 64 \times 256$
	WSM-up4	WSM-up3	$80 \times 64 \times 256$	$160 \times 128 \times 128$
	Prediction	WSM-up4	$160 \times 128 \times 128$	$160 \times 128 \times 1$
Refinement	Rel1	WSM-up4	$160 \times 128 \times 128$	$160 \times 128 \times 128$
	Rel2	Rel1	$160 \times 128 \times 128$	$160 \times 128 \times 1$

the final upsampling layer 'WSM-up4.' To train the two convolutional layers, the absolute prediction error, $|\hat{d}_i - d_i^{\mathrm{gt}}|$, is defined as the ground-truth and the Euclidean loss is employed. Thus, the output of the added convolution layers is not a reliability value but an error estimate (or uncertainty). We hence normalize the error estimate to $[0, 1]$, and subtract the normalized result from 1 to yield the reliability value. Fig. 6(d) shows a reliability map $\boldsymbol{\alpha}$. We see that the reliability map yields low values in erroneous areas in the actual error map in Fig. 6(c).

Next, based on the reliability map $\boldsymbol{\alpha}$, we model the conditional probability distribution of the depth field $\mathbf{d}$ for the CRF optimization as $p(\mathbf{d}|\widehat{\mathbf{d}}, \boldsymbol{\alpha}) = \frac{1}{Z} \cdot \exp\Big(-E(\mathbf{d}, \widehat{\mathbf{d}}, \boldsymbol{\alpha})\Big)$ where E is an energy function and Z is the normalization term. The energy function is given by

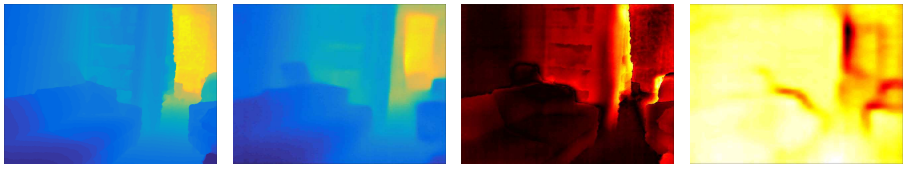
$$E(\mathbf{d}, \widehat{\mathbf{d}}, \boldsymbol{\alpha}) = U(\mathbf{d}, \widehat{\mathbf{d}}, \boldsymbol{\alpha}) + \lambda \cdot V(\mathbf{d}, \boldsymbol{\alpha}) \tag{1}$$

where U is a unary term to make the refined depth $\mathbf{d}$ similar to the estimated depth $\widehat{\mathbf{d}}$ and V is a pairwise term to make each refined depth similar to the weighted sum of adjacent depths. Also, λ controls a tradeoff between the two terms. The unary term is defined as

$$U(\mathbf{d}, \widehat{\mathbf{d}}, \boldsymbol{\alpha}) = \sum_i \alpha_i \big(d_i - \hat{d}_i\big)^2 \tag{2}$$

where d_i, $\widehat{d}_i$, and α_i denote the refined depth, estimated depth, and reliability of pixel i, respectively. By employing α_i, we strongly encourage a refined depth to be similar to an estimated depth only if the estimated depth is reliable. In other words, when an estimated depth is unreliable, it can be modified significantly to yield a refined depth during the CRF optimization.

To model the relation between neighboring pixels, we use the auto-regression model, which are employed in various applications, such as image matting [34], depth recovery [35], and monocular depth estimation [17]. In addition, to take



(a) Ground-truth $\mathbf{d}^{\text{gt}}$ (b) Estimated depth $\widehat{\mathbf{d}}$ (c) Error map $|\widehat{\mathbf{d}}-\mathbf{d}^{\text{gt}}|$ (d) Reliability map $\boldsymbol{\alpha}$

Fig. 6: An example of the reliability map. In (c) and (d), a bright color indicates a higher value than a dark one.

advantage of the different characteristics of image and depth map, we use the color similarity introduced in [36,37]. In this work, we generalize the color-guided auto-regression model in [35], based on the reliability map, to define the pairwise term

$$V(\mathbf{d},\boldsymbol{\alpha}) = \sum_i \left(d_i - \sum_{j\in\mathcal{N}_i} \omega_{ij} d_j \right)^2 \tag{3}$$

where $\mathcal{N}_i$ is the 11×11 neighborhood of pixel i. Also, ω_{ij} is the similarity between pixel i and its neighbor j, given by

$$\omega_{ij} = \frac{\alpha_j}{T} \cdot \exp\left(-\frac{\sum_{c\in\mathcal{C}} \|\mathbf{B}_i \circ (\mathcal{S}_i^c - \mathcal{S}_j^c)\|_2^2}{2\cdot 3\cdot \sigma_1^2} \right) \tag{4}$$

where $\mathcal{S}_i^c$ denotes the 5×5 patch centered at pixel i, extracted from color channel c of the image, and $\mathcal{C}$ is the set of three YUV color channels. Also, $\circ$ represents the element-wise multiplication, σ_1 is a weighting parameter, and T is the normalization factor. The color-guided kernel $\mathbf{B}_i$ is defined on the 5×5 patch centered at pixel i, and its element corresponding to neighbor pixel k is given by

$$B_{i,k} = \exp\left(-\frac{\sum_{c\in\mathcal{C}} (I_i^c - I_k^c)^2}{2\cdot 3\cdot \sigma_2^2} \right) \tag{5}$$

where I_i^c is the image value of pixel i in channel c, and σ_2 is a parameter. The exponential term in (4), through the pairwise term V in (3), encourages neighboring pixels with similar colors to have similar depths. Moreover, because of α_j in (4), we constrain the depth of pixel i to be more similar to that of neighbor pixel j, when neighbor pixel j is more reliable. This causes the depths of reliable pixels to propagate to those of unreliable ones, improving the accuracy of the overall depth map.

We can rewrite the energy function in (1) in vector notations.

$$E(\mathbf{d},\widehat{\mathbf{d}},\boldsymbol{\alpha}) = (\mathbf{d}-\widehat{\mathbf{d}})^T\mathbf{A}(\mathbf{d}-\widehat{\mathbf{d}}) + \lambda\,(\mathbf{d}-\mathbf{W}\mathbf{d})^T(\mathbf{d}-\mathbf{W}\mathbf{d}) \tag{6}$$

where $\mathbf{A}$ is the diagonal matrix whose ith diagonal element is α_i, and $\mathbf{W} \triangleq [\omega_{ij}]$ is the weight matrix. Finally, the refined depth $\widetilde{\mathbf{d}}$ can be obtained by solving the

maximum *a posteriori* (MAP) inference problem:

$$\widetilde{\mathbf{d}} = \arg\max_{\mathbf{d}} p(\mathbf{d}|\widehat{\mathbf{d}}, \boldsymbol{\alpha}) = \arg\min_{\mathbf{d}} E(\mathbf{d}, \widehat{\mathbf{d}}, \boldsymbol{\alpha}). \tag{7}$$

Since the energy function is quadratic, the closed-form solution is given by

$$\widetilde{\mathbf{d}} = (\mathbf{A} + \lambda\ (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W}))^{-1}\mathbf{A}\widehat{\mathbf{d}}. \tag{8}$$

4 Experiments

4.1 Experimental Setup

Implementation details: We implement the proposed network using the Caffe library [38] on an NVIDIA GPU with 12GB memory. We initialize the backbone network in the encoder with the pre-trained weights, and initialize the other parameters randomly. We train the network in two phases. First, we train the depth estimation network, composed of the encoding and decoding parts. The learning rate is initialized at 10^{-7} and decreased by 10 times when training errors converge. The batch size is set to 4. The momentum and the weight decay are set to typical values of 0.9 and 0.0005. Second, we fix the parameters of the encoding and decoding parts and then train the refinement part. The learning rate starts at 10^{-8}, while the batch size, the momentum, and the weight decay are the same as the first phase. The parameters λ in (1), σ_1 in (4), and σ_2 in (5) is set to 1.5, 6.5, and 0.1. It takes about two days to train the whole network.

Evaluation metrics: For quantitative evaluation, we assess the proposed monocular depth estimation algorithm based on the four evaluation metrics [8, 13, 14].

- Average absolute relative error (rel): $\frac{1}{N}\sum_i \frac{|\hat{d}_i - d_i^{\text{gt}}|}{d_i^{\text{gt}}}$
- Average $\log_{10}$ error ($\log_{10}$): $\frac{1}{N}\sum_i |\log_{10}(\hat{d}_i) - \log_{10}(d_i^{\text{gt}})|$
- Root mean squared error (rms): $\sqrt{\frac{1}{N}\sum_i (\hat{d}_i - d_i^{\text{gt}})^2}$
- Accuracy with threshold t: percentage of $\hat{d}_i$ such that $\max\{\frac{d_i^{\text{gt}}}{\hat{d}_i}, \frac{\hat{d}_i}{d_i^{\text{gt}}}\} = \delta < t$

4.2 NYU Depth Dataset V2

We evaluate the proposed algorithm on the large RGB-D dataset, NYU Depth Dataset V2 [31]. It contains 120K pairs of RGB and depth images, captured with Microsoft Kinect devices, with 249 scenes for training and 215 scenes for testing. Each image or depth has a spatial resolution of 640×480. We uniformly sample frames from the entire training scenes and extract approximately 24K unique pairs. Using the colorization tool [34] provided with the dataset, we fill in missing values of depth maps automatically. Since an image and its depth map are not perfectly synchronized, we eliminate top 2K erroneous samples, after

Table 2: Comparison of various network models on the NYU dataset. A number in the third column is the number of parameters in both encoder and decoder.

Encoding	Decoding	Parameters	rel	rms
AlexNet	FC	106M	0.215	0.833
	Deconv	6.7M	0.204	0.842
VGGNet-16	FC	60M	0.183	0.776
	Deconv	18.5M	0.194	0.746
ResNet-50	FC	74M	0.160	0.626
	Deconv	53.5M	0.152	0.602
	Deconv-Conv	66.0M	0.149	0.604
	UpProj [19]	63.6M	0.145	0.596
	Inception	62.1M	0.148	0.607
	Equivalent	61.0M	0.150	0.595
	WSM	61.1M	**0.141**	**0.582**

training the depth estimation network for one epoch. We perform the online data augmentation schemes *Scale*, *Flip*, and *Translataion*, introduced in [13]. Also, as in [15, 21], we center-crop images to 561×427 pixels containing valid depths, and then downsample them to 304×228 pixels, which are used as the input to the network. For the evaluation, we upsample the estimated depth map to the original size 561×427 through the bilinear interpolation and compare the result against the ground-truth depth map.

Comparison of network models: Table 2 compares the proposed algorithm with other network models. First, we test how the depth estimation performance is affected when a different backbone network (AlexNet [25], VGGNet16 [26], or ResNet-50 [20]) is adopted as the encoder. In this test, we use the fully-connected layer 'FC' or the deconvolution block 'Deconv' as the decoder. Specifically, FC is a fully-connected layer of 1280 ($= 40 \times 32$) dimensions directly connected to an output feature map of the encoder. Deconv is the upsampling block, composed of four 3×3 deconvolution layers only. As the backbone network gets deeper from AlexNet to ResNet-50, the depth estimation performance is improved.

Next, we compare the performances of various decoders, after fixing ResNet-50 as the encoder. 'Deconv-Conv' is the decoder, composed of four pairs of 3×3 deconvolution layer and 5×5 convolution layer. 'UpProj' is the Laina *et al.*'s decoder [19]. 'Inception' [32] uses a 7×7 convolution layer instead of the $W \times 3$ and $3 \times H$ WSM layers in Fig. 5. Similarly, 'Equivalent' replaces the two WSM layers with a square convolution layer, but set the square size to be about the same as the sum of $3 \times H$ and $W \times 3$. Consequently, Equivalent and the proposed WSM decoder require similar numbers of parameters. The output resolution is 160×128 except for FC, which yields 40×32 output because of GPU memory constraints. The WSM decoder provides outstanding performances. Especially, note that WSM significantly outperforms Equivalent, which is another method

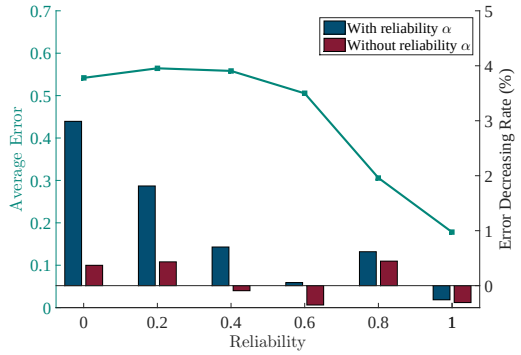


Fig. 7: Verifying reliability values and the reliability-based refinement. The line plot with the left axis shows the average absolute error for each quantized reliability value. The bar plot with the right axis shows the decreasing rate of the average error due to the refinement with or without reliability α.

using large kernels. This indicates that the improved performance of WSM is made possible not only by the use of large kernels, but also because horizontally or vertically flat characteristics of depth maps are exploited. Moreover, despite the large kernels, the proposed WSM algorithm requires a moderate number of parameters, and in fact demands less than Deconv-Conv, UpProj, and Inception.

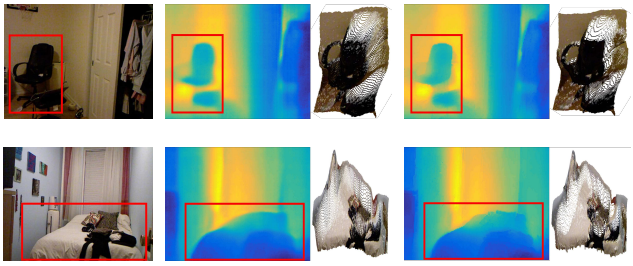
Efficacy of refinement step: The line graph in Fig. 7 shows the absolute average error for each quantized reliability value. As the reliability value increases, the average error decreases. This indicates that the proposed algorithm correctly predicts the confidence of an estimated depth using the reliability map.

The bar graph in Fig. 7 plots how the proposed reliability-based refinement decreases the average error. To confirm its impacts comparatively, we also provide the refinement result without the reliability, *i.e.* when α is fixed to 1 in (2) and (4). With the adaptive reliability, we see that the error decreases by up to 2.9%. In particular, estimation errors are significantly decreased by the refinement step, especially for the pixels with low reliability values. On the other hand, without the reliability, there are only little changes in the errors.

Fig. 8 shows point cloud rendering results of depth maps with and without the refinement step. We see that the refinement separates the objects from the background more clearly and more accurately.

Comparison with the state-of-the-arts: Table 3 compares the proposed algorithm with eleven conventional algorithms [8,12–19,21,39]. We report the performances of two versions of the proposed algorithm: 'WSM' uses only the depth estimation network and 'WSM-Ref' performs the reliability-based refinement additionally. Note that both WSM and WSM-Ref outperform all conventional algorithms.

Fig. 9 compares the depth maps of the proposed algorithm with those of the state-of-the-art monocular depth estimation algorithms [14, 18, 19] qualita-



(a) Input (b) with refinement (c) w/o refinement

Fig. 8: Point cloud rendering of depth maps with or without the refinement step.

Table 3: Quantitative comparison on the NYU Depth Dataset V2 [31]. The best performance is boldfaced, and the second best is underlined.

Methods	Error (↓)			Accuracy (↑)		
	rel	$\log_{10}$	rms	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Karsch *et al.* [12]	0.374	0.134	1.12	-	-	-
Ladicky *et al.* [8]	-	-	-	0.542	0.829	0.941
Liu *et al.* [21]	0.335	0.127	1.06	-	-	-
Li *et al.* [17]	0.232	0.094	0.821	0.621	0.886	0.968
Liu *et al.* [15]	0.230	0.095	0.824	0.614	0.883	0.971
Wang *et al.* [16]	0.220	0.094	0.745	0.605	0.890	0.970
Eigen *et al.* [13]	0.215	0.095	0.907	0.611	0.887	0.971
Eigen *et al.* [14]	0.158	0.067	0.641	0.769	0.950	0.988
Chakrabarti *et al.* [18]	0.149	0.062	0.620	0.806	0.958	0.988
Li *et al.* [39]	0.143	0.063	0.635	0.788	0.958	<u>0.991</u>
Laina *et al.* [19]	<u>0.140</u>	<u>0.060</u>	0.597	<u>0.811</u>	0.953	0.988
WSM	0.141	<u>0.060</u>	<u>0.582</u>	<u>0.811</u>	<u>0.962</u>	<u>0.991</u>
WSM-Ref	**0.135**	**0.058**	**0.571**	**0.816**	**0.964**	**0.992**

tively. The proposed WSM and WSM-Ref generate more faithful depth maps than the conventional algorithms. Through WSM, both WSM and WSM-Ref reconstruct flat depths on the walls more accurately. Moreover, WSM-Ref improves the depth maps through the reliability-based refinement. For instance, WSM-Ref reconstructs the detailed depths of the objects on the desk in the first row and the chairs in the second and third rows more precisely.

4.3 Make3D

We also test the proposed algorithm on the outdoor dataset Make3D [10], which contains 534 pairs of RGB and depth images: 400 pairs for training and 134 for testing. There is a difference of resolutions between RGB images (1704×2272) and depth images (305×55). Since the dataset is not large enough for training a deep network, training on Make3D needs a careful strategy. We follow the strategy of [15, 19]. Specifically, we resize RGB images to 345×460 pixels and

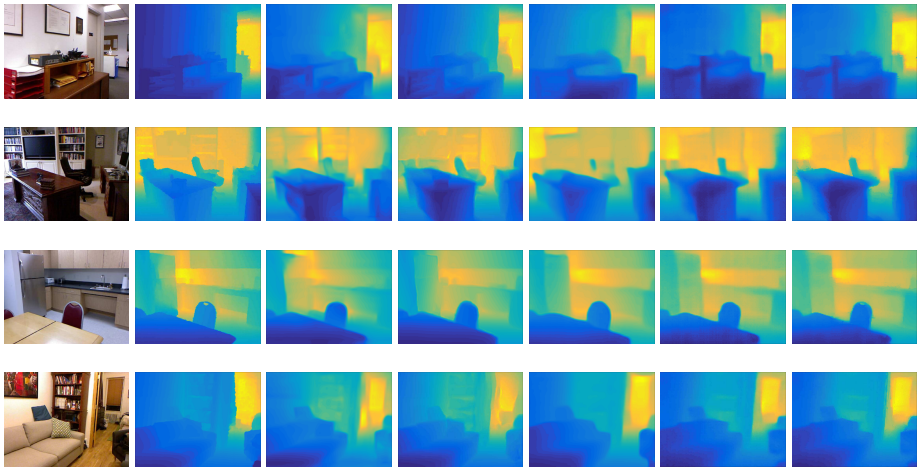


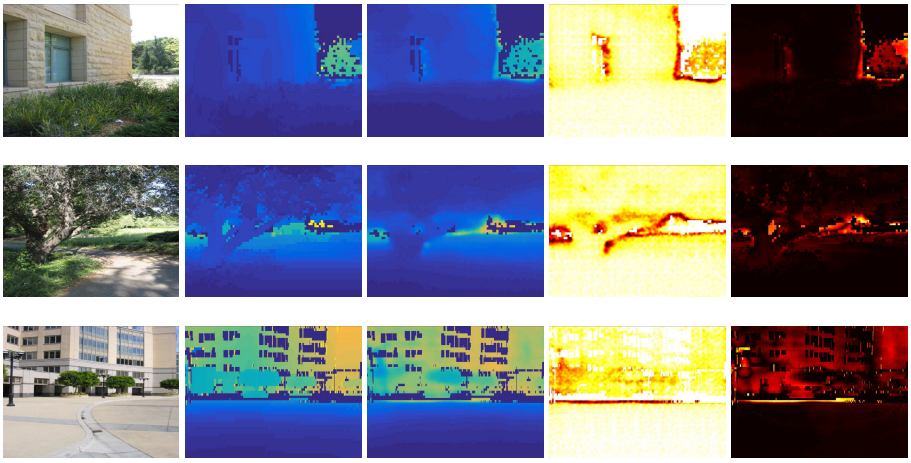
Fig. 9: Qualitative comparison: (a) input image, (b) ground-truth, (c) Eigen *et al.* [14], (d) Chakrabarti *et al.* [18], (e) Laina *et al.* [19], and (f) the proposed WSM, and (g) the proposed WSM-Ref.

downsample them to 173×230 pixels. Since Make3D expresses depths up to 80m only, the depths of far objects, *e.g.* sky, are often inaccurate. Thus, we train the network after masking out pixels with depths over 70m. This criterion, called C1, was first suggested by [21] and has been used in [15, 19, 21]. We perform online data augmentation, as done in the case of the NYU dataset. All the other parameters are the same. For evaluation, we upsample an estimated depth map to 345×460 and compare the result against the ground-truth depth map, which is also upsampled to 345×460. We only compute the errors in regions of depths less than 70m (C1 criterion).

Table 4 compares the proposed algorithm with conventional algorithms [12, 15, 17, 19, 21]. Again, the proposed WSM-Ref outperforms all conventional algorithms. Fig. 10 shows qualitative results. The proposed WSM-Ref yields faithful depth maps, and the reliability maps detect erroneous regions effectively. These experimental results indicate that the proposed algorithm is a promising solution to monocular depth estimation for both indoor and outdoor scenes.

5 Conclusions

In this work, we proposed a monocular depth estimation algorithm based on the WSM upsampling and the reliability-based refinement. First, we developed the WSM layers to exploit the horizontally or vertically flat characteristics of depth maps. We constructed the depth estimation network by stacking WSM upsampling blocks upon the ResNet-50 encoder. Second, we measured the reliability of each estimated depth, and exploited the information to refine the depth map



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

Fig. 10: Depth estimation of the proposed WSM-Ref on the Make3D dataset: (a) input, (b) ground-truth, (c) estimation result, (d) reliability map, and (e) error map. In (d) and (e), a bright color indicates a higher value than a dark one.

Table 4: Comparison of quantitative results on the Make3D dataset.

Methods	rel	$\log_{10}$	rms
Karsch et al. [12]	0.355	0.127	9.20
Liu et al. [21]	0.335	0.137	9.49
Liu et al. [15]	0.314	0.119	8.60
Li et al. [17]	0.278	0.092	7.19
Laina et al. [19]	<u>0.176</u>	<u>0.072</u>	**4.46**
WSM	0.185	0.073	4.85
WSM-Ref	**0.171**	**0.063**	**4.46**

through the CRF optimization. Experimental results showed that the proposed algorithm significantly outperforms the conventional algorithms on both indoor and outdoor datasets, while requiring a moderate number of parameters.

Acknowledgement

This work was supported partly by the Cross-Ministry Giga KOREA Project Grant funded by the Korean Government (MSIT) (development of 4D reconstruction and dynamic deformable action model based hyper-realistic service technology) under Grant GK18P0200, partly by the National Research Foundation of Korea Grant funded by the Korean Government (MSIP) under Grant NRF-2015R1A2A1A10055037 and Grant NRF-2018R1A2B3003896, and partly by NAVER LABS.

References

1. Yang, S., Maturana, D., Scherer, S.: Real-time 3D scene layout from a single image using convolutional neural networks. In: ICRA. (2016) 2183–2189
2. Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., Guo, B.: An interactive approach to semantic modeling of indoor scenes with an RGBD camera. ACM Trans. Graph. **31**(6) (2012) 136
3. Porzi, L., Buló, S.R., Penate-Sanchez, A., Ricci, E., Moreno-Noguer, F.: Learning depth-aware deep representations for robotic perception. IEEE Robot. Autom. Lett. **2**(2) (2017) 468–475
4. Kim, K.R., Koh, Y.J., Kim, C.S.: Multiscale feature extractors for stereo matching cost computation. IEEE Access **6** (May 2018) 27971–27983
5. Gupta, A., Efros, A.A., Hebert, M.: Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: Proc. ECCV. (2010) 482–496
6. Lee, D.C., Gupta, A., Hebert, M., Kanade, T.: Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In: Proc. NIPS. (2010) 1288–1296
7. Russell, B.C., Torralba, A.: Building a database of 3D scenes from user annotations. In: Proc. IEEE CVPR. (2009) 2711–2718
8. Ladicky, L., Shi, J., Pollefeys, M.: Pulling things out of perspective. In: Proc. IEEE CVPR. (2014) 89–96
9. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Proc. NIPS. (2005) 1161–1168
10. Saxena, A., Sun, M., Ng, A.Y.: Make3D: Learning 3D scene sctructure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5) (2009) 824–840
11. Liu, B., Gould, S., Koller, D.: Single image depth estimation from predicted semantic labels. In: Proc. IEEE CVPR. (2010) 1253–1260
12. Karsch, K., Liu, C., Kang, S.B.: Depth transfer: Depth extraction from video using non-parametric sampling. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11) (2014) 2144–2158
13. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Proc. NIPS. (2014) 2366–2374
14. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proc. IEEE ICCV. (2015) 2650–2658
15. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proc. IEEE CVPR. (2015) 5162–5170
16. Wang, P., Shen, X., Lin, Z., S.Cohen, Price, B., Yuille, A.: Towards unified depth and semantic prediction from a single image. In: Proc. IEEE CVPR. (2015) 2800–2809
17. Li, B., Shen, C., Dai, Y., van den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In: Proc. IEEE CVPR. (2015) 1119–1127
18. Chakrabarti, A., Shao, J., Shakhnarovich, G.: Depth from a single image by harmonizing overcomplete local network predictions. In: Proc. NIPS. (2016) 2658–2666
19. Laina, I., Rupprecht, C., Belagiannis, V.: Deeper depth prediction with fully convolutional residual networks. In: Proc. IEEE 3DV. (2016) 239–248
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE CVPR. (2016) 770–778

21. Liu, M., Salzmann, M., He, X.: Discrete-continuous depth estimation from a single image. In: Proc. IEEE CVPR. (2014) 716–723
22. Kim, H.U., Kim, C.S.: CDT: Cooperative detection and tracking for tracing multiple objects in video sequences. In: Proc. ECCV. (2016) 851–867
23. Jang, W.D., Kim, C.S.: Online video object segmentation via convolutional trident network. In: Proc. IEEE CVPR. (2017) 5849–5856
24. Lee, J.T., Kim, H.U., Lee, C., Kim, C.S.: Semantic line detection and its applications. In: Proc. IEEE ICCV. (2017) 3229–3237
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proc. NIPS. (2012) 1097–1105
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2012)
27. Lee, J.H., Heo, M., Kim, K.R., Kim, C.S.: Single-image depth estimation based on Fourier domain analysis. In: Proc. IEEE CVPR. (2018) 330–339
28. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proc. IEEE CVPR. (2009) 248–255
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. IEEE CVPR. (2015) 3431–3440
30. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Proc. NIPS. (2016) 4898–4906
31. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Proc. ECCV. (2012) 746–760
32. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: AAAI. (2016) 4278–4284
33. Matthies, L., Kanade, T., Szeliski, R.: Kalman filter-based algorithms for estimating depth from image sequences. Int. J. Comput. Vis. **3**(3) (1989) 209–238
34. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2) (2008) 228–242
35. Yang, J., Ye, X., Li, K., Hou, C., Wang, Y.: Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. IEEE Trans. Image Process. **23**(8) (2016) 3443–3458
36. Diebel, J., Thrun, S.: An application of markov random fields to range sensing. In: Advances in neural information processing systems. (2006) 291–298
37. Park, J., Kim, H., Tai, Y.W., Brown, M.S., Kweon, I.: High quality depth map upsampling for 3d-tof cameras. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1623–1630
38. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.: Caffe: Convolutional architecture for fast feature embedding. In: ACM Multimedia. (2014) 675–678
39. Li, J., Klein, R., Yao, A.: A two-streamed network for estimating fine-scaled depth maps from single rgb images. In: Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy. (2017) 22–29