Monocular Depth Estimation Using Laplacian Pyramid-Based Depth Residuals

Minsoo Song, Seokjae Lim, and Wonjun Kim, *Member, IEEE*

***Abstract*— With a great success of the generative model via deep neural networks, monocular depth estimation has been actively studied by exploiting various encoder-decoder architectures. However, the decoding process in most previous methods, which repeats simple up-sampling operations, probably fails to fully utilize underlying properties of well-encoded features for monocular depth estimation. To resolve this problem, we propose a simple but effective scheme by incorporating the Laplacian pyramid into the decoder architecture. Specifically, encoded features are fed into different streams for decoding depth residuals, which are defined by decomposition of the Laplacian pyramid, and corresponding outputs are progressively combined to reconstruct the final depth map from coarse to fine scales. This is fairly desirable to precisely estimate the depth boundary as well as the global layout. We also propose to apply weight standardization to pre-activation convolution blocks of the decoder architecture, which gives a great help to improve the flow of gradients and thus makes optimization easier. Experimental results on benchmark datasets constructed under various indoor and outdoor environments demonstrate that the proposed method is effective for monocular depth estimation compared to state-of-the-art models. The code and model are publicly available at: |https://github.com/tjqansthd/LapDepth-release|.**

***Index Terms*— Monocular depth estimation, depth residuals, depth boundary, Laplacian pyramid, weight standardization.**

I. Introduction

DEPTH estimation from a single monocular image has been a critical task for a long time in many applications of real-world scenarios. For example, horizontal boundaries or location of the vanishing point can be efficiently estimated based on the statistics of the depth information, which are very useful to quickly understand a given scene. These clues often give remarkable advantages of interpreting the 3D geometrical layout, thus inferring the depth information has now become essential in the field of autonomous driving systems. Due to such plentiful possibilities, many researchers have devoted a great deal of efforts to resolve the problem of monocular depth estimation.

Manuscript received July 20, 2020; revised October 13, 2020 and December 7, 2020; accepted January 2, 2021. Date of publication January 8, 2021; date of current version October 28, 2021. This work was supported by Konkuk University in 2020. This article was recommended by Associate Editor W. Liu. *(Corresponding author: Wonjun Kim.)*

The authors are with the Department of Electrical and Electronics Engineering, Konkuk University, Seoul 05029, South Korea (e-mail: tjqansthd@konkuk.ac.kr; hgg08@konkuk.ac.kr; wonjkim@konkuk.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3049869.

Digital Object Identifier 10.1109/TCSVT.2021.3049869

In early days, features based on the human perception have been widely exploited. For example, the tendency of edge orientations and the distribution of frequency coefficients, which are strongly relevant to perceive the depth characteristic, are aggregated from local regions of a given image [1]. To precisely extract the statistical information from those features, image segmentation is often adopted as the pre-processing step [2], [3]. On the other hand, the ensemble scheme of global and features has been importantly considered for scene recognition [4], [5] and also employed for depth estimation. For example, several studies have attempted to select appropriate depth values for a given color image based on both global and local structural similarity with other scenes, which are followed by the optimization process for refining the visibility of the estimated depth image [6], [7]. Even though recent methods utilizing such well-designed features have shown the significant progress in estimating the depth information, they still lack the ability to predict the complicated relation between color and depth values only with a single image.

Owing to the great success of the generative model using the deep neural network (DNN), many researchers have started to formulate the problem of depth estimation as the problem of image translation, i.e., translation from the color image to the depth one. To extract underlying features relevant to the depth information, the convolution neural network (CNN) has been widely adopted as the backbone architecture of the generative model. Based on large-scale datasets including various real-world environments, e.g., KITTI driving dataset [8] and NYU depth dataset [9], the relation between color and depth values can be well encoded through deeply stacked architectures. In general, the depth information scanned by using 3D sensors (e.g., LiDAR, Kinect, etc.) is employed as the ground truth for supervised learning approaches. On the other hand, there also have been several attempts to utilize stereo inputs for monocular depth estimation in an unsupervised manner [10], [11]. Even though DNN-based approaches show a great ability to reveal the depth layout without any domain knowledge, they still struggle with ambiguities occurring at the depth boundary. Specifically, existing methods mostly utilize features extracted from well-known encoders, e.g., VGG, ResNet, etc. These latent features are simply up-sampled back to their original size by the decoding process in the symmetric architecture and lastly converted into the depth map. This conversion process has a difficulty to consider the depth boundaries of objects at

Fig. 1. From top to bottom: input color images, ground truth, and estimation results by the proposed method. Note that left two samples are from the KITTI dataset [8] while right ones belong to the NYU Depth V2 dataset [9].

various scale levels, thus it probably yields inaccurate depth values between object boundaries.

To solve these drawbacks of previous approaches, we propose a novel yet simple method for monocular depth estimation. The key idea of the proposed method is to precisely interpret the relation between encoded features and the final output for monocular depth estimation by exploiting the Laplacian pyramid-based decoder architecture. Laplacian has been used in various fields of scene understanding because of its ability to preserve the local information of the given data [12]. Our idea is inspired by the ability of the Laplacian pyramid in successfully emphasizing the difference across the scale spaces, which is highly relevant to object boundaries. Specifically, the encoded features are fed into stacked convolution blocks to generate sub-band depth residuals at each pyramid level. The depth map is progressively restored from coarse to fine scales by combining with the depth residuals at each pyramid level. This restoration process is greatly helpful to improve the performance of predicting depth boundaries. Instead of only repeating upsampling operations to recover back to the original resolution, we propose to guide the decoding process with residuals of the input color image, which are computed from different levels of the Laplacian pyramid, and combine predicted results (i.e., depth residuals) progressively from coarse to fine scales to reconstruct the final depth map. Based on this multi-level decoding scheme of depth residuals, we can utilize encoded features more efficiently to estimate the depth information in the complex scene. Moreover, we also propose to apply weight standardization to pre-activation convolution blocks, which is significantly effective to improve the flow of gradients and makes the convergence stable without loss of the performance. Examples of depth estimation by the proposed method are shown in Fig. 1. The main contributions of the proposed method can be summarized as follows:

- We propose to adopt the Laplacian pyramid for resolving the problem of monocular depth estimation. By recovering depth residuals from encoded features in different levels of the Laplacian pyramid and summing up those predicted results progressively, the proposed method successfully restores local details, e.g., depth boundary, as well as the global layout.
- By applying weight standardization to the pre-activation convolution blocks, which are the basic module of our decoder architecture, the flow of gradients is efficiently improved and thus the proposed network can be stably trained for estimating depth residuals whose values are mostly zero, i.e., sparse.
- We demonstrate various experimental results on benchmark datasets constructed under complicated indoor and outdoor environments, and show the efficiency and robustness of the proposed method compared to state-of-the-art methods.

The remainder of this paper is organized as follows. A brief review of related works is presented in Section II. The proposed method is explained in Section III. Experimental results on benchmark datasets and an ablation study are reported in Section IV. The conclusions follow in Section V.

II. Related Work

In this Section, we present a comparative review of previous studies for monocular depth estimation, which can be divided into two main groups, i.e., handcrafted feature-based methods and deep learning-based methods.

A. Handcrafted Feature-Based Methods

Early works mainly exploited statistical features acquired from a given color image for monocular depth estimation. As the first step, Torralba and Oliva [1] explored properties of spectral magnitudes according to the depth variation. Saxena *et al.* [3] predicted the depth value by using the planar layout including 3D location and orientation, which are estimated based on the Markov random field (MRF) with several types of textural features, e.g., edge orientations, chromatic values, etc. Chun *et al.* [13] estimated the depth map from a single indoor scene by utilizing the position information of the ground region, e.g., the relative distance from the highest floor point. More recent approaches have focused on finding appropriate depth values of the given image by computing the structural similarity with other scenes, which already have the real depth information. Karsch *et al.* [6] proposed to find the candidate depth by checking the similarity of spectral coefficients and refined it with the warping technique (i.e., SIFT flow [14]). In [15], authors focused on depth gradients, which are applied to the Poisson-based depth reconstruction, instead of directly finding optimal candidates (i.e., captured depth images) in training samples. Herrera *et al.* [16] attempted to resolve the problem of selecting the optimal depth from training samples in a coarse-to-fine manner by exploiting the cluster-based learning scheme. However, patch-based aggregation strategies often fail to clearly warp geometrical structures of the complicated scene, which makes the estimation result blurry.

B. Deep Learning-Based Methods

Based on the great capability of the deep neural network (DNN) for scene understanding, e.g., image classification and segmentation, DNN-based approaches have been started to be adopted for monocular depth estimation. In the beginning stage, Eigen *et al.* [17] firstly devised a two-stage DNN-based model. Specifically, they first predicted the coarse result of the depth image based on the deeply stacked convolution neural network (CNN) and refine local details by employing the coarse result and the original color image as the input of the second CNN stream. Even though the estimation result appears blurry due to pooling operations

repeated many times during the encoding process, it shows the possibility of the DNN-based approach for monocular depth estimation. After the performance by [17] was announced, various encoder-decoder architectures have been developed for inferring the relation between color and depth values in a given image more accurately. In particular, the conditional random field (CRF), which allows for the affinity between local regions in a pair-wise manner, has been employed with the superpixel-based segmentation technique to refine the visibility of the estimated result [18], [19]. Gan *et al.* [20] also embedded the affinity layer into the encoder-decoder architecture for considering both local and global contexts more efficiently. Xu *et al.* [21] proposed a deep architecture which fuses the complementary information derived from multi-scale CNN outputs by integrating a cascade of multiple CRFs. On the other hand, several researchers have explored unsupervised or semi-supervised learning schemes for monocular depth estimation by using the disparity-based consistency, which is computed via the stereo reconstruction loss [10], [11], [22]. Specifically, Garg *et al.* [11] predicted a disparity map through the deep CNN and conducted simple inverse warping with another view to compute the reconstruction loss. Godard *et al.* [10] proposed consistency loss using both left and right images warped from the predicted disparities. Kuznietsov *et al.* [22] attempted to improve the estimation result by using a direct image alignment loss even with sparse ground-truth depth. Most recently, Fu *et al.* [23] utilized the ordinal regression to estimate the depth boundary with features densely extracted by the atrous spatial pyramid pooling (ASPP) scheme [24]. Cao *et al.* [25] proposed to cast the depth estimation problem to classification by discretizing the continuous depth value and get the confidence of the predicted depth map in the form of probability distribution. In [26], authors proposed a learning strategy to pre-train the deep network on the relative depth dataset, which is constructed by using the stereo matching algorithm, and subsequently fine-tune the model with the ground truth depth. Mohaghegh *et al.* [27] introduced a data-driven approach that extracts the global form of the depth map from the pre-trained model and refines the depth map by mapping from image patches to depth values. Zuo *et al.* [28] applied the multi-scale intensity guidance to the global and local residual learning schemes for depth enhancement. In addition to the series of methods for predicting the depth map from a given 2D image, the direct reconstruction of 3D objects also has been actively studied. Ma *et al.* [29] proposed the separated channel-spatial convolution with attention modules, which adaptively fuse the channel information and the spatial one, to extract the abundant representation of the object.

Although DNN-based generative models have accomplished the significant performance improvement for monocular depth estimation, they still do not fully exploit underlying properties of well-encoded features due to the inefficient decoding scheme, which often leads to blurring artifacts at the depth boundary. Those methods mostly focus on predicting the depth information in a coarse-to-fine manner by adopting various approaches. However, they still struggle to clearly preserve the depth boundaries of objects at various scale levels since the depth map is estimated only from the final spatial resolution. Different from those approaches, we propose to apply the Laplacian pyramid to the decoding process for progressively restoring depth boundaries via various scale spaces.

III. Proposed Method

Our proposed method aims to successfully restore local details (i.e., depth boundaries) as well as the global layout of the depth map by applying the Laplacian pyramid-based decomposition technique to the decoding process. Specifically, Laplacian residuals of the input color image guide encoded features to be generated as the depth residual containing local details, which appropriately represent depth properties of different scale spaces. To make such decoding process more efficient, we also apply weight standardization to the pre-activation convolution block, which gives a great help for estimating depth residuals whose values are mostly zero. In this Section, we first introduce the overall architecture of the proposed decoder for monocular depth estimation. Then, the whole decoding process with the effect of weight standardization will be presented in detail. Lastly, we will explain loss functions used for training the proposed architecture.

A. Architecture Details

The overall architecture of the proposed method is shown in Fig. 2. Our network consists of the pre-trained encoder and the proposed decoder for restoring depth residuals. The encoder part can be set by using any architecture, e.g., VGG [30], ResNet [31], DenseNet [32], etc. In our implementation, we adopt ResNext101 [33], which is pre-trained for image classification. At the encoder side, the input color image is highly compressed as latent features through deeply stacked convolution blocks. The spatial size of such features becomes very small (1/16 scale of the original resolution in our implementation), however, those compactly contain the relation between color and depth values in the embedding space, which is learned from diverse scene geometries. To obtain the contextual information more densely, we adopt the DenseASPP technique [34] with four dilation rates, i.e., 3, 6, 12, and 18, for the convolution block of the encoder.

The proposed decoder is divided into multiple branches of the Laplacian pyramid. One branch, which is in charge of the highest level of the Laplacian pyramid (see the Layer4 in Fig. 2), performs a decoding task to restore the global layout of the depth map. Other branches generate the depth residuals (i.e., from R_4 to R_1 in Fig. 2) based on latent features, which are guided by Laplacian residuals of the input color image at the corresponding scale (i.e., from L_4 to L_1 in Fig. 2). This depth residual is progressively combined with the intermediate depth map, which is the output obtained from the higher level of the Laplacian pyramid, by using the point-wise addition. As shown in Fig. 2, we utilize the five-level Laplacian pyramid for the decoding process. The filter size of all convolution layers in the decoder is set to 3×3. The architecture details of the proposed method is also shown in Table I. The whole decoding process will be explained in the following subsection.

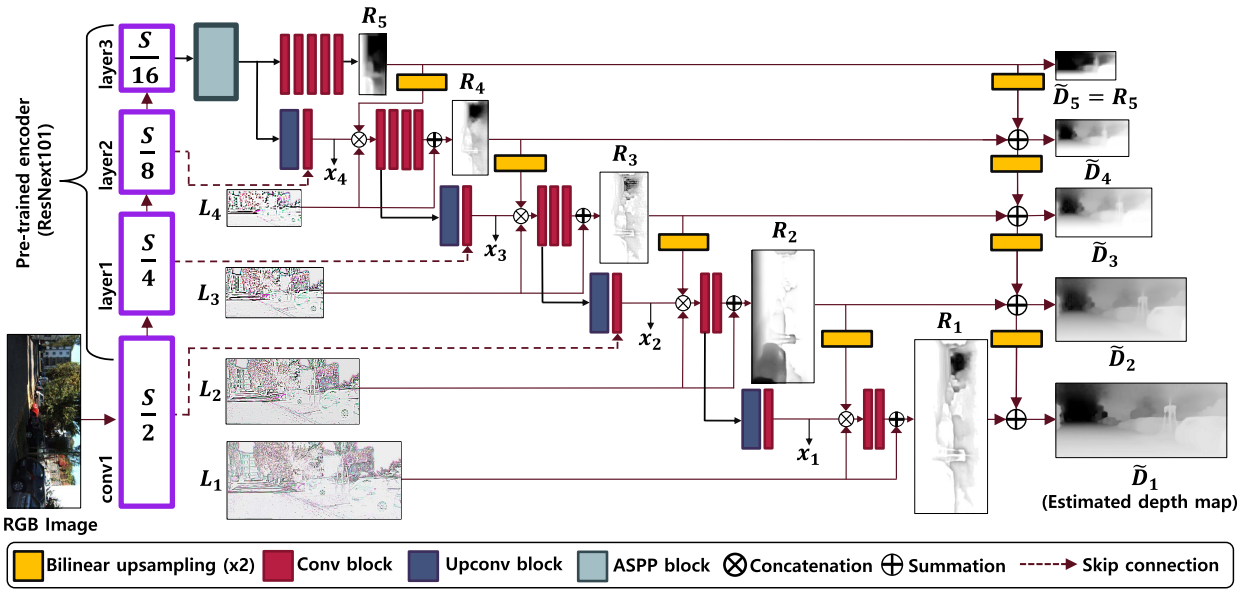

Fig. 2. The overall architecture of the proposed method for monocular depth estimation. S indicates the spatial resolution of the input image. The depth residual restored from the highest level of the Laplacian pyramid, i.e., R_5, is up-sampled ($\times 2$) and subsequently combined with the depth residual of the finer scale by using the point-wise addition. Note that all values of images except the input color image are inverted in this figure for the better visualization.

TABLE I

DETAILED ARCHITECTURE OF THE PROPOSED METHOD (**UP**: UPSAMPLING FACTOR, **CHANNEL**: THE NUMBER OF INPUT AND OUTPUT CHANNELS FOR EACH BLOCK, **IN** AND **OUT**: SPATIAL RESOLUTION OF THE INPUT AND THE OUTPUT, **INPUT**: INPUT OF EACH BLOCK, **LEV**: THE LEVEL INDEX OF THE LAPLACIAN PYRAMID)

Encoder							
Block	**Filter size**	**Stride**	**Channel**	**In**	**Out**	**Input**	
conv1	7×7	2	3/64	S	S/2	input color image	
maxpool	3×3	2	64/64	S/2	S/4	F(conv1)	
layer1	3×3	2	64/256	S/4	S/4	F(maxpool)	
layer2	3×3	2	256/512	S/4	S/8	F(layer1)	
layer3	3×3	2	512/1024	S/8	S/16	F(layer2)	
Decoder							
Block	**Filter size**	**Up**	**Channel**	**In**	**Out**	**Input**	**Lev**
reduction	1×1	1	1024/512	S/16	S/16	F(layer3)	-
ASPP	3×3	1	512/512	S/16	S/16	F(reduction)	-
dec5	3×3	1	512/1	S/16	S/16	F(ASPP)	5th
dec4up	3×3	2	512/256	S/16	S/8	F(ASPP)	4th
dec4reduc	1×1	1	768/252	S/8	S/8	F(layer2 $\oplus$ dec4up)	4th
dec4bneck	3×3	1	256/256	S/8	S/8	F(dec4reduc $\oplus$ R_4 $\oplus$ dec5*)	4th
dec4	3×3	1	256/1	S/8	S/8	F(dec4bneck)	4th
dec3up	3×3	2	256/128	S/8	S/4	F(dec4bneck)	3rd
dec3reduc	1×1	1	384/124	S/4	S/4	F(layer1 $\oplus$ dec3up)	3rd
dec3bneck	3×3	1	128/128	S/4	S/4	F(dec3reduc $\oplus$ R_3 $\oplus$ dec4*)	3rd
dec3	3×3	1	128/1	S/4	S/4	F(dec3bneck)	3rd
dec2up	3×3	2	128/64	S/4	S/2	F(dec3bneck)	2nd
dec2reduc	1×1	1	128/60	S/2	S/2	F(conv1 $\oplus$ dec2up)	2nd
dec2bneck	3×3	1	64/64	S/2	S/2	F(dec2reduc $\oplus$ R_2 $\oplus$ dec3*)	2nd
dec2	3×3	1	64/1	S/2	S/2	F(dec2bneck)	2nd
dec1up	3×3	2	64/60	S/2	S	F(dec2bneck)	1st
dec1bneck	3×3	1	64/64	S	S	F(dec1up $\oplus$ R_1 $\oplus$ dec2*)	1st
dec1	3×3	1	64/1	S	S	F(dec1bneck)	1st

Note : $\oplus$ and * denote concatenation and up-sampling ($\times 2$), respectively. S indicates the spatial resolution of the original image. F(B) denotes the output of the corresponding block B.

B. Decoding of Depth Residuals

First of all, we compute the Laplacian residual of the input color image, i.e., L_k, as follows:

$$L_k = I_k - Up(I_{k+1}), \quad k = 1, 2, 3, 4, \tag{1}$$

where k denotes the level index in the Laplacian pyramid. I_k is obtained by down-sampling the original input image to $1/2^{k-1}$ scale. $Up(\cdot)$ denotes the function of up-sampling ($\times 2$) and we adopt the bilinear interpolation for all the resizing procedures in the proposed method. Now, let R_k be the depth residual obtained from the k-th pyramid level and this depth residual is generated as follows: first, the latent feature x_k is concatenated with L_k as well as the up-sampled version of the depth residual, which is obtained from the $(k+1)$-th

Fig. 3. The example of the depth residual restored at each pyramid level of the proposed decoder. All pixel values of depth residuals are inverted for visualization. First row: input color image, R_4, R_3, R_2, and R_1. Second row: R_5, $\tilde{D}_4$, $\tilde{D}_3$, $\tilde{D}_2$, and $\tilde{D}_1$ (final depth map).

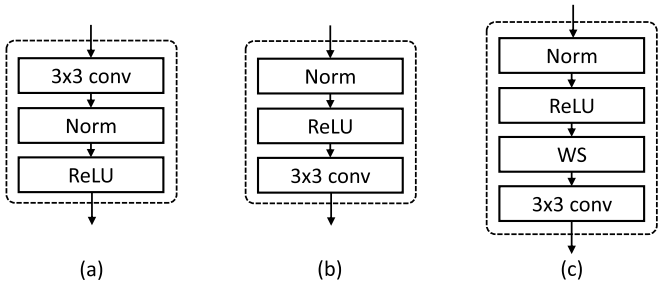

Fig. 4. (a) Typical convolution block. (b) Pre-activation convolution block [35]. (c) Pre-activation convolution block with weight standardization. Note that WS denotes weight standardization.

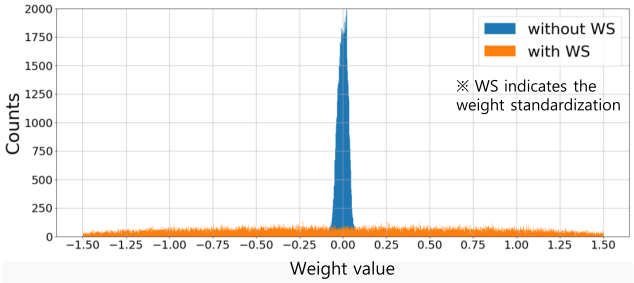

Fig. 5. Weight distributions in the last conv block used for estimating R_1 shown in Fig. 2. Note that other conv (or deconv) blocks show the similar distributions of weight values.

level of the Laplacian pyramid (see Fig. 2). Subsequently, such concatenated features are fed into stacked convolution blocks and the corresponding output is pixel-wisely added to L_k once again. This process can be formulated as follows:

$$R_k = B_k([x_k, L_k, Up(R_{k+1})]) + L_k, \quad k = 1, 2, 3, 4, \quad (2)$$

where $[x_k, L_k, Up(R_{k+1})]$ refers to concatenation of x_k, L_k, and $Up(R_{k+1})$. B_k, which consists of stacked convolution blocks, produces the one-channel output whose spatial resolution is the same as L_k. It is noteworthy that L_k guides the decoding process to accurately restore local details of various scale spaces and thus gives a great help to reveal depth boundaries without blurring artifacts. Finally, the depth map is progressively reconstructed from the highest level of the Laplacian pyramid as follows:

$$\tilde{D}_k = R_k + Up(\tilde{D}_{k+1}), \quad k = 1, 2, 3, 4. \quad (3)$$

Note that $\tilde{D}_5$ is set equal to R_5, which contains the global layout of the depth map at the highest pyramid level, as shown in Fig. 2. By iteratively computing (3) with the order of $k = 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$, $\tilde{D}_1$ is computed as the final depth map. Examples of depth residuals generated at the k-th pyramid level and the final depth map are shown in Fig. 3. As can be seen, depth properties according to the scene geometry are well revealed in depth residuals predicted at different scales.

To make the decoding process for monocular depth estimation more efficient, we also propose to conduct weight standardization in the pre-activation convolution block, which is the basic module of the proposed decoder as shown in Fig. 4(c). Since the depth map is reconstructed based on the iterative summation of depth residuals (see (3)), it is advisable for the predicted depth residual to contain negative and positive values in balance for stably and accurately estimating the depth information. However, the typical convolution block, which is composed of convolution→normalization→activation as shown in Fig. 4(a), discards most negative values in terms of the nonlinear property of the rectified linear unit (ReLU) activation at the last step. Even though the pre-activation convolution block [35] is adopted to alleviate this problem (see Fig. 4(b)), weight values of the convolution filter still tend to be zero with small variations since depth residuals are sparse (i.e., most parts of depth residuals have zero values as shown in Fig. 3), which probably leads to the gradient vanishing problem during training.

By simply putting the module of weight standardization [36] before performing the convolution operation (see Fig. 4(c)), the proposed decoder is successfully able to improve the flow of gradients by normalizing them during backpropagation, which is computed from each level of the Laplacian pyramid. This is fairly desirable for keeping the stability of the color-to-depth translation based on the residual information. Figure 5 shows the effect of the weight standardization for monocular depth estimation. It is easy to see that weight values are widely and evenly distributed in the proposed scheme whereas most weight values stay around zero without weight standardization. Again, previous convolution blocks are often not helpful for estimating depth residuals in that regard. By taking this advantage with the Laplacian pyramid-based decomposition scheme, it is thought that the proposed method makes it possible to successfully learn the complicated relation between color and depth values.

C. Loss Function

The trainable parameters of the proposed network are optimized based on our loss function L_t, which is composed

of two components, i.e., data loss L_d and gradient loss L_g, as follows:

$$L_t = \begin{cases} \alpha L_d(y, y^*), & \text{if epoch} < 30, \\ \alpha L_d(y, y^*) + \beta L_g(y, y^*), & \text{otherwise,} \end{cases} \quad (4)$$

where y and y^* denote the predicted depth map and the ground truth, respectively. α and β denote balancing factors for L_d and L_g, which are set to 10 and 0.1 respectively through extensive experiments. Note that the gradient loss is computed after 30 epochs since training in the KITTI dataset tends to be unstable when using both data and gradient losses at the beginning of the training due to the sparsity of the ground truth. Specifically, depth gradients computed from the interpolated depth map, which is slightly different from the original one, interfere with the convergence of the data loss. To alleviate this problem, the gradient loss is additionally computed after the depth map is properly restored by only using the data loss.

1) Data Loss: In general, the depth data is densely collected from close areas whereas that taken from a distance is very sparse due to limitation of the 3D sensor. To alleviate the problem of imbalance, we adopt the square root of the loss function introduced in [17] as the data loss L_d, which computes the difference between predicted depth values and the ground truth in the log space as follows:

$$L_d(y, y^*) = \sqrt{\frac{1}{n}\sum_{i \in V}^{N_V} d_i^2 - \frac{\lambda}{n^2}\left(\sum_{i \in V}^{N_V} d_i\right)^2}, \quad (5)$$

where $d_i = \log y_i - \log y_i^*$ and V is a set of valid pixels in the depth map. N_V denotes the total number of valid pixels. The balancing factor λ is set to 0.85 in the same way used in [37].

2) Gradient Loss: In order to enhance local details particularly at depth boundaries, we exploit gradients of the depth map for the loss function. Since the sparse depth data of the ground truth gives a difficulty to accurately compute gradients both in horizontal and vertical directions, the Matlab toolkit provided by [9], which has been popularly employed in previous approaches, is used for the point interpolation. The gradient loss is formulated as follows:

$$L_g(y, y^*) = \frac{1}{N}\sum_{i}^{N} |y_{h,i} - m(y^*)_{h,i}| + |y_{v,i} - m(y^*)_{v,i}|, \quad (6)$$

where $m(\cdot)$ denotes the interpolation function used in [9]. $y_{h,i}$ and $m(y^*)_{h,i}$ denote the i-th gradient value of the estimated depth map and the interpolated ground truth in the horizontal direction, respectively. Similar to this, $y_{v,i}$ and $m(y^*)_{v,i}$ are defined in the vertical direction. N is the total number of pixels contained in the estimated depth map. It is noteworthy that our gradient loss has an effect to force local edges to be accurately aligned across multi-level pyramids and thus makes depth boundaries to be clearly revealed in the final depth map. The effect of the gradient loss is shown in Fig. 6. As can be seen, the depth boundary of the far-away object is successfully revealed in the depth map predicted with the gradient loss.

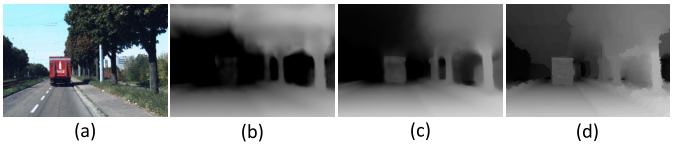

Fig. 6. Visual comparison of depth estimation results according to different combinations of loss functions. (a) Input color image. (b) L_d. (c) $L_d + L_g$. (d) Ground truth. Note that the gradient loss contributes to yield more reliable results on depth boundaries.

IV. Experimental Results

In this Section, we evaluate the performance of the proposed method through various experiments conducted on two widely-employed benchmark datasets, i.e., KITTI [8] and NYU Depth V2 [9] datasets, which are constructed under diverse indoor and outdoor environments.

A. Training

The proposed method is implemented on the PyTorch framework [38]. All the parameters in the proposed decoder (i.e. weights of the network) are initialized based on the strategy introduced in [39]. Each layer of the proposed decoder contains group normalization, which is known to be independent to the batch size. The proposed network is trained from scratch for 50 epochs with a batch size of 16 using the AdamW optimizer [40] where power and momentum are set to 0.9 and 0.999, respectively. The weight decaying factor is set to 0.0005 for the encoder and zero for the proposed decoder. The learning rate is firstly set to 10^{-4} and decreased until 10^{-5} using a polynomial decay with the power of 0.5. It takes 16 hours for training the proposed network with four NVIDIA GeForce Titan Xp GPUs. We adopt ResNext101 [33] as the encoder for feature extraction whose parameters are initialized with the pre-trained model based on the ILSVRC [41] dataset. We fix parameters of the first few layers since those are well-trained to extract low-level features (e.g., edges, corners, etc.) by using diverse natural images. Parameters of all the batch normalization layers in the encoder are also fixed as pre-trained values. The parameter size of the encoder and the proposed decoder are 58M and 15M, respectively.

In the training phase, online data augmentation is conducted to avoid the overfitting problem. Specifically, training samples are randomly cropped to 704 $\times$ 352 pixels for the KITTI dataset and 512 $\times$ 416 pixels for the NYU Depth V2 one, followed by randomly rotating them in the range of $[-3, 3]$ degrees. Input images are also horizontally flipped with the probability of 0.5. Moreover, brightness, color, and gamma values of the input color image are randomly adjusted with the scale factor selected from the range of [0.9, 1.1].

B. Benchmark Datasets

1) KITTI: The KITTI dataset [8] contains various road environments acquired from autonomous driving scenarios. The resolution of acquired images is 1242 $\times$ 375 pixels. For the performance comparison, we adopt the split strategy introduced by Eigen *et al* [17]. According to this scheme, the

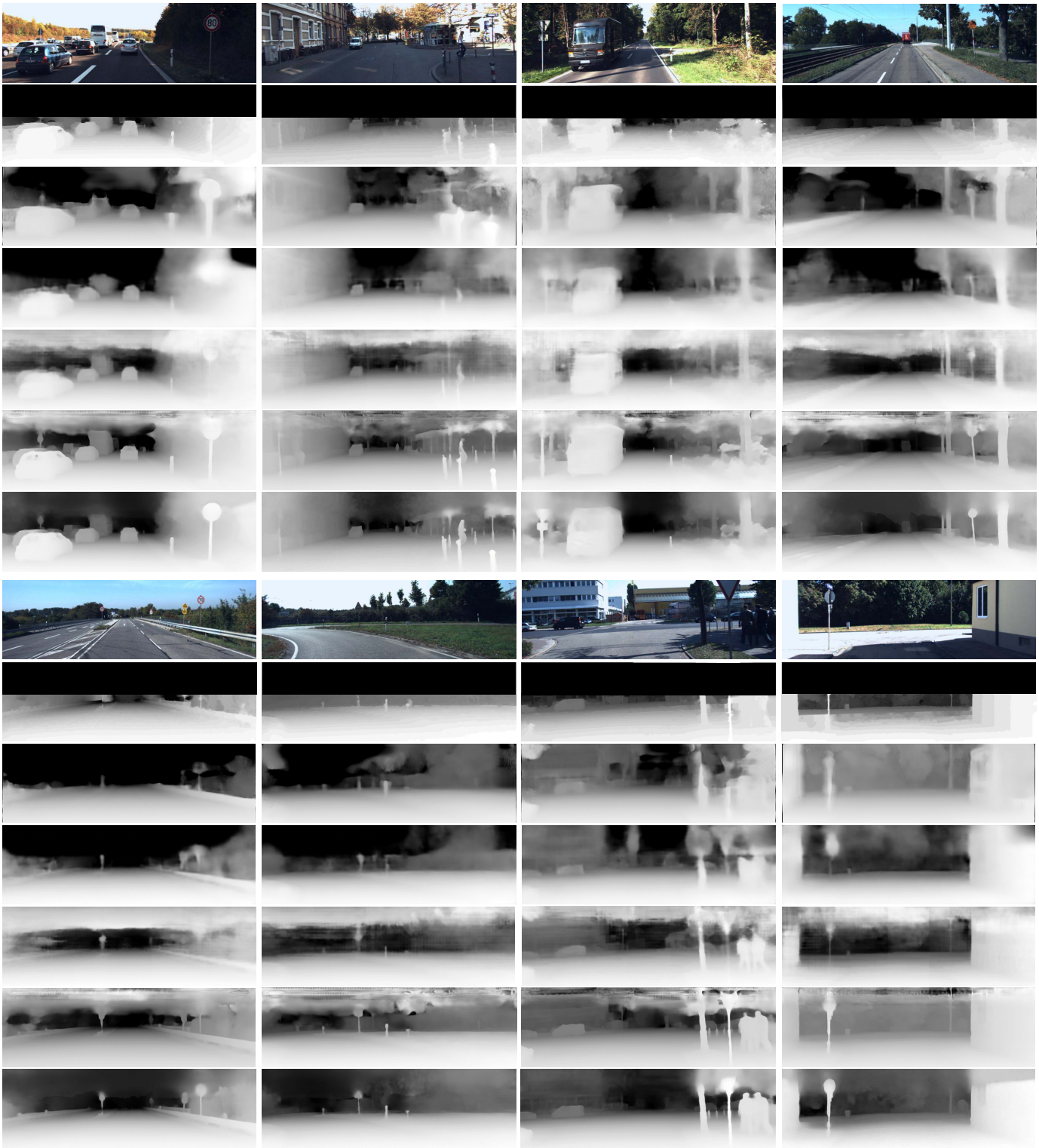

Fig. 7. Results of depth estimation on the KITTI dataset [8]. 1st and 8th rows: input color images. 2nd and 9th rows: ground truth. 3rd and 10th rows: results by Godard *et al.* [10]. 4th and 11th rows: results by Kuznietsov *et al.* [22]. 5th and 12th rows: results by Fu *et al.* [23]. 6th and 13th rows: results by Lee *et al.* [37]. 7th and 14th rows: results by the proposed method.

test set contains 697 images selected from 29 scenes while the training one is composed of 23,488 images from remaining 32 scenes. The maximum value of our predicted output is limited to the order of 80 meters in the test phase, as explained in the guideline of the KITTI dataset. We also adopt the central cropping scheme used in [11] for the performance evaluation.

2) NYU Depth V2: The NYU Depth V2 dataset [9] consists of 120K pairs of RGB and depth images which are captured under 464 indoor scenes with the resolution of 640 × 480 pixels by using the Microsoft Kinect sensor. We apply the previous train/test split, which contains 249 scenes for training and 654 images from remaining

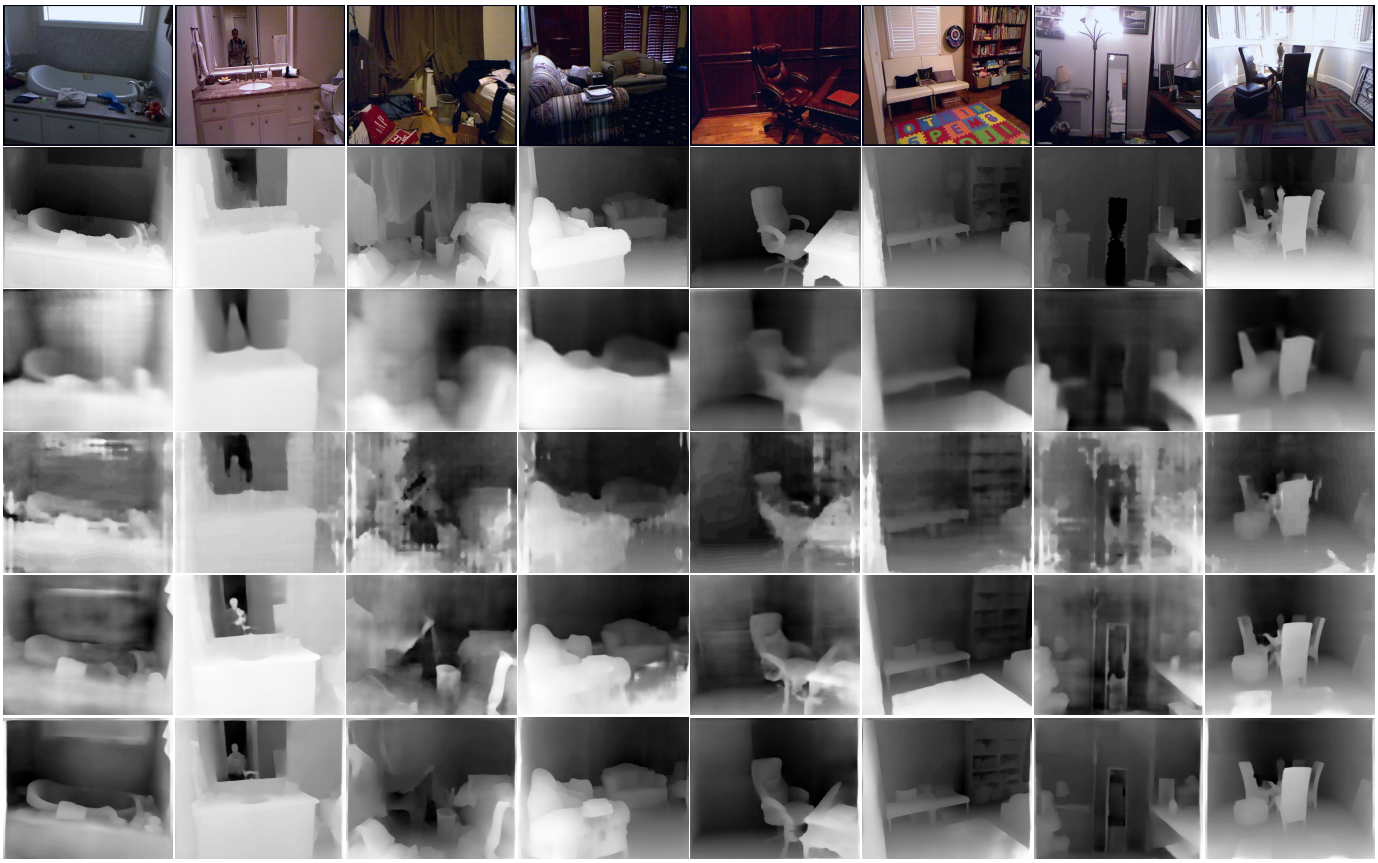

Fig. 8. Results of depth estimation on the NYU Depth V2 dataset [9]. 1st row: input color images. 2nd row: ground truth. 3rd row: results by Laina *et al.* [45]. 4th row: results by Fu *et al.* [23]. 5th row: results by Lee *et al.* [37]. 6th row: results by the proposed method.

215 scenes for testing, as introduced in [17]. Since the RGB image and the corresponding depth map are not perfectly synchronized, we abundantly select 36,253 samples from 249 scenes for training. Depth maps predicted by the proposed method are center-cropped to 561 $\times$ 427 pixels (as introduced in [8]) to be compared with existing methods.

C. Performance Evaluation

To demonstrate the efficiency and robustness, the performance of the proposed method is evaluated on two benchmark datasets, i.e., KITTI [8] and NYU Depth V2 [9] datasets. First, several results of qualitative comparisons with state-of-the art methods are shown in Fig. 7 and 8. Specifically, most previous methods fail to estimate the accurate boundary of thin objects, e.g., traffic signs and pillars on the sidewalk in Fig. 7. The shapes of vehicles are often revealed blurry in some methods [22], [23]. Even though the object boundaries are quite well estimated in [37], this method still suffers from the ambiguity occurring at the relatively high location (see the result of the third example in Fig. 7). In contrast, the proposed method reliably provides depth maps with clear depth boundaries in various road environments. In the case of indoor environments, there are many objects at a short distance, thus the depth boundaries are strongly related to the object boundaries as shown in Fig. 8. Most previous methods often yield unexpected depth variations and make the estimation result blurry at the complicated boundaries of various objects. In particular, previous models fail to maintain the uniformity of depth values in the same planes as shown in the first and third examples of Fig. 8. Moreover, complex textures of background lead to incorrect predictions in previous methods (see the ground region in the last example of Fig. 8). Compared to previous approaches, we can see that the proposed method successfully preserves the depth boundaries even with complex object shapes.

For the quantitative evaluation, we use six metrics introduced by Eigen *et al.* [17], which have been most widely employed for the performance evaluation of monocular depth estimation, defined as follows:

$$\textbf{Abs Rel} = \frac{1}{|T|}\sum_{y\in T}|y-y^*|/y^*,$$
$$\textbf{Sq Rel} = \frac{1}{|T|}\sum_{y\in T}||y-y^*||^2/y^*,$$
$$\textbf{RMSE} = \sqrt{\frac{1}{|T|}\sum_{y\in T}||y-y^*||^2},$$
$$\textbf{RMSE}_{\log} = \sqrt{\frac{1}{|T|}\sum_{y\in T}||\log y-\log y^*||^2},\ \textbf{Accuracy} = \%\ \text{of}\ y_i\ \text{s.t.}\ \max\left(\frac{y}{y^*},\frac{y^*}{y}\right)=\delta<\text{th},$$
$$\textbf{log10} = \frac{1}{|T|}\sum_{y\in T}|\log y-\log y^*|,$$

where y and y^* denote the predicted depth map and the ground truth, respectively. T is the total number of valid pixels in the ground truth. Based on such metrics, we compared ours with state-of-the-art methods on KITTI [8] and NYU

TABLE II

QUANTITATIVE EVALUATIONS ON THE KITTI DATASET [8] USING THE TEST SPLIT OF EIGEN *et al.* [17] FOR DIFFERENT CAPS. $*$ DENOTES THAT THE PERFORMANCE IS EVALUATED USING THE OFFICIAL ANNOTATED GROUND TRUTH (DEFAULT: USING RAW VELODYNE DATA). NOTE THAT WE USE THE CROPPING STRATEGY INTRODUCED BY GARG *et al.* [11]

Method	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log
	Higher value is better			Lower value is better			
Eigen *et al.* [17]	0.692	0.899	0.967	0.190	1.515	7.156	0.270
Liu *et al.* [19]	0.647	0.882	0.961	0.217	1.841	6.986	0.289
Godard *et al.* [10]	0.861	0.949	0.976	0.114	0.898	4.935	0.206
Kuznietsov *et al.* [22]	0.862	0.960	0.986	0.113	0.741	4.621	0.189
Gan *et al.* [20]	0.890	0.964	0.985	0.098	0.666	3.933	0.173
Fu *et al.* [23]	0.897	0.966	0.986	0.099	0.593	3.714	0.161
Lee *et al.* [37]	0.904	0.967	0.984	0.091	0.555	4.033	0.174
Proposed method	**0.939**	**0.972**	**0.988**	**0.082**	**0.427**	**3.203**	**0.158**
Godard *et al.**	0.916	0.980	0.994	0.085	0.584	3.938	0.135
Kuznietsov *et al.**	0.906	0.980	0.995	0.138	0.478	3.610	0.138
Amiri *et al.** [42]	0.923	0.984	0.995	0.078	0.417	3.464	0.126
Fu *et al.**	0.932	0.984	0.994	0.072	0.307	2.727	0.120
Lee *et al.**	0.950	0.993	**0.999**	0.064	0.254	2.815	0.100
Proposed method*	**0.962**	**0.994**	**0.999**	**0.059**	**0.212**	**2.446**	**0.091**
0 - 80m cap							
Garg *et al.* [11]	0.740	0.904	0.962	0.169	1.080	5.104	0.273
Godard *et al.* [10]	0.873	0.954	0.979	0.108	0.657	3.729	0.194
Kuznietsov *et al.* [22]	0.875	0.964	0.988	0.108	0.595	3.518	0.179
Gan *et al.* [20]	0.898	0.967	0.986	0.094	0.552	3.133	0.165
Fu *et al.* [23]	0.906	0.968	0.986	0.096	0.503	2.902	0.155
Lee *et al.* [37]	0.914	0.970	0.986	0.088	0.437	3.127	0.165
Proposed method	**0.924**	**0.975**	**0.989**	**0.076**	**0.382**	**2.764**	**0.140**
Godard *et al.**	0.861	0.949	0.976	0.114	0.898	4.935	0.206
Fu *et al.**	0.936	0.985	0.995	0.071	0.268	2.271	0.116
Lee *et al.**	0.959	0.994	**0.999**	0.060	0.182	2.005	0.092
Proposed method*	**0.967**	**0.995**	**0.999**	**0.056**	**0.161**	**1.830**	**0.086**
0 - 50m cap							

Note : the best performance is highlighted as a bold text.

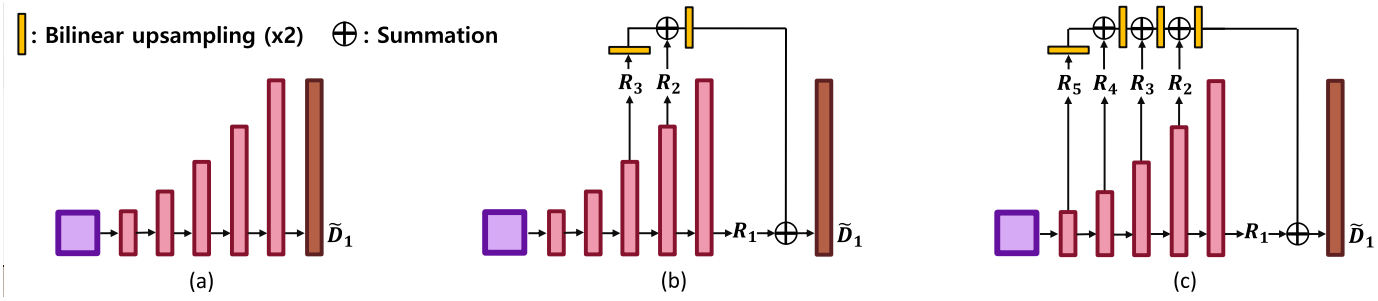

Fig. 9. Variations of the decoder architecture according to the number of pyramid levels. (a) Non-pyramid (b) Three-level. (c) Five-level (proposed method). Note that $\tilde{D}_1$ denotes the final depth map.

Depth V2 [9] datasets, and corresponding results are shown in Table II and III, respectively. Note that the performance of the proposed method is evaluated for both raw velodyne data and the annotated ground truth data recently released in the KITTI dataset. We use 652 test images for evaluating the annotated ground truth (45 images which do not have the corresponding ground truth are excluded for test). It is easy to see that our results achieve the best performance for all of metrics at the caps of both 50m and 80m as shown in Table II. Moreover, the proposed method also provides the reliable estimation results on the NYU Depth V2 dataset (see Table III). Therefore, it is thought that our Laplacian pyramid-based depth residuals are effective to accurately estimate the depth information from the color image acquired from various indoor and outdoor environments. Moreover, the processing speed of the proposed method is about 32 fps for the resolution of 1242×375 pixels and thus can be applied to various real-time applications.

D. Ablation Study

In this subsection, comparative experiments are conducted on the KITTI dataset to verify the effectiveness of the proposed architecture, i.e., Laplacian pyramid-based decoder and pre-activation convolution block based on weight standardization. Firstly, performance variations are evaluated according to the number of the pyramid levels adopted in the decoder architecture. The change of the decoder architecture according to different numbers of pyramid levels is shown in Fig. 9. Note that the Non-pyramid decoder as shown in

TABLE III
QUANTITATIVE EVALUATIONS ON THE NYU DEPTH V2 DATASET [9]

Method	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	log10	RMSE
	Higher value is better			Lower value is better		
Eigen *et al.* [17]	0.611	0.887	0.971	0.215	0.095	0.641
Li *et al.* [18]	0.621	0.886	0.968	0.232	0.094	0.821
Liu *et al.* [19]	0.650	0.906	0.976	0.213	0.087	0.759
Eigen *et al.* [43]	0.769	0.950	0.988	0.158	0.067	0.641
Chakrabarti *et al.* [44]	0.806	0.958	0.988	0.149	0.062	0.620
Laina *et al.* [45]	0.811	0.953	0.988	0.127	0.055	0.573
Qi *et al.* [46]	0.834	0.960	0.990	0.128	0.057	0.569
Hao *et al.* [47]	0.841	0.966	0.991	0.127	0.053	0.555
Fu *et al.* [23]	0.828	0.965	0.992	0.115	0.051	0.509
Lee *et al.* [37]	0.882	**0.979**	**0.995**	0.112	**0.047**	**0.352**
Proposed method	**0.885**	**0.979**	**0.995**	**0.110**	**0.047**	0.393

TABLE IV
PERFORMANCE ANALYSIS OF THE PROPOSED METHOD ON THE KITTI DATASET ACCORDING TO THE NUMBER OF PYRAMID LEVELS

Architectures	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log
	Higher value is better			Lower value is better			
Non-pyramid (Baseline)	0.921	0.972	0.992	0.080	0.315	3.174	0.123
Three-level	0.950	0.989	0.997	0.065	0.239	2.695	0.107
Five-level (Proposed)	0.961	0.994	0.999	0.060	0.215	2.449	0.092
Six-level	0.964	0.994	0.999	0.058	0.208	2.418	0.090
Seven-level	0.964	0.995	0.999	0.058	0.206	2.396	0.089

Note : All the architectures were trained with the smaller batch size (16→8) due to the GPU memory limit of architectures, which are deeper than the five-level Laplacian pyramid.

Fig. 9 and Table IV does not generate residuals as well as the Laplacian pyramid. As shown in Table IV, using more number of levels for decomposing features into the Laplacian pyramid is advantageous for accurately restoring local details as well as the global layout of the depth map. To conduct the experiment under the same conditions, all the architectures were trained with a smaller batch size (16→8) since the cases of six-level and seven-level Laplacian pyramids have the limitation of GPU memory. As can be seen in the sixth and seventh rows of Table IV, the performance is slightly improved, however, the performance has become almost saturated as pyramid levels increase more than five. Therefore, it is thought that the global layout of the depth map is sufficiently restored at the five-level Laplacian pyramid. For the six-level and seven-level pyramids, the parameter sizes of the corresponding decoders increase by 10M and 18M, respectively, compared to the five-level pyramid whereas the performance improvement is minimal. In addition, variants of the convolution block are tested and the corresponding results are shown in Fig. 10 and Table V. By applying weight standardization to the pre-activation convolution block, the decoding process is significantly improved in terms of loss convergence as well as the estimation accuracy. We can see that the improvement for the flow of gradients is important to correctly predict the depth residual, which is quite sparse (see the Table V). Specifically, the training loss of the proposed Laplacian pyramid-based decoder is stably converged without significant oscillations by using our convolution block (i.e., pre-activation + weight standardization) as shown in Fig. 10. Moreover, the test loss as well as the test accuracy are also converged in a similar trend with the training phase as shown in Fig. 11. Note that the test accuracy is measured for the case of "$\delta < 1.25$", which is used in the quantitative evaluation for the KITTI dataset. We also conducted various benchmark experiments by changing backbone encoders with six mainstream frameworks (i.e., MobileNetV2, VGG19, InceptionV3, ResNet-101, DenseNet-161, and ResNext-101) while other settings remained, and the corresponding results are shown in Table VII. As can be seen, the proposed decoder shows reliable results regardless of the encoder structure. In particular, the MobileNetV2-based model shows only 2.4% drop for the accuracy "$\delta < 1.25$" whereas it contains 20% number of parameters and shows 6 fps speed-up compared to the model based on the ResNext-101. The parameter size of the proposed decoder is determined about 15M, thus it can be widely applied to various backbone encoders. Lastly, the performance analysis is conducted according to changes of the network architecture in terms of pyramid types. The decoder based on the Gaussian pyramid attempts to restore the layouts of the depth map at each corresponding scale. This shows a better performance than the case where the decoder simply generates the final depth map at the original scale without using the pyramid structure. However, this structure lacks the ability to connect the high-frequency information between scale spaces. The feature pyramid network (FPN) [48]-based decoder contributes to more precise prediction of depth values by extracting feature maps which include both semantically strong features and low-level features. Although the FPN-based decoder improves the prediction performance compared to the Gaussian-based approach, it still does not reflect the scale variance due to

TABLE V

PERFORMANCE ANALYSIS OF THE PROPOSED METHOD ON THE KITTI DATASET ACCORDING TO VARIANTS OF THE CONVOLUTION BLOCK

Architectures	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log
	Higher value is better			Lower value is better			
w/o WS and pre-activation	0.928	0.976	0.994	0.077	0.296	2.975	0.115
w/o pre-activation	0.939	0.982	0.996	0.071	0.262	2.886	0.109
w/o WS	0.943	0.984	0.996	0.064	0.258	2.816	0.107
Proposed method	**0.962**	**0.994**	**0.999**	**0.059**	**0.212**	**2.446**	**0.091**

Note : WS denotes weight standardization

TABLE VI

PERFORMANCE ANALYSIS OF THE PROPOSED METHOD ON THE KITTI DATASET ACCORDING TO THE CHANGE OF THE DECODER STRUCTURE

Architectures	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log
	Higher value is better			Lower value is better			
Non-pyramid (Baseline)	0.922	0.972	0.992	0.079	0.313	3.176	0.124
Gaussian-based	0.931	0.979	0.993	0.075	0.305	3.013	0.112
FPN-based	0.945	0.986	0.995	0.068	0.258	2.894	0.105
Laplacian-based (Proposed)	**0.962**	**0.994**	**0.999**	**0.059**	**0.212**	**2.446**	**0.091**

Note : all settings except decoder structures are same (e.g., backbone encoder, training scheduling, hyperparameters, etc.).

TABLE VII

PERFORMANCE ANALYSIS OF THE PROPOSED METHOD ON THE KITTI DATASET ACCORDING TO THE CHANGE OF THE BACKBONE ENCODER

Architectures	Params	Speed(fps)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Abs Rel	Sq Rel	RMSE	RMSE log
			Higher value is better			Lower value is better			
MobileNetV2 [49]	15.7M	38.6	0.941	0.990	0.998	0.072	0.284	2.760	0.109
VGG19 [30]	23.9M	34.1	0.950	0.992	0.998	0.066	0.242	2.619	0.098
InceptionV3 [50]	17.7M	36.7	0.955	0.993	0.999	0.063	0.232	2.563	0.095
ResNet-101 [31]	43.1M	35.6	0.958	0.993	0.999	0.063	0.225	2.494	0.094
DenseNet-161 [32]	33.5M	33.8	0.963	0.994	0.999	0.060	0.221	2.476	0.092
ResNext-101 [33]	73.5M	32.1	0.964	0.994	0.999	0.059	0.212	2.446	0.091

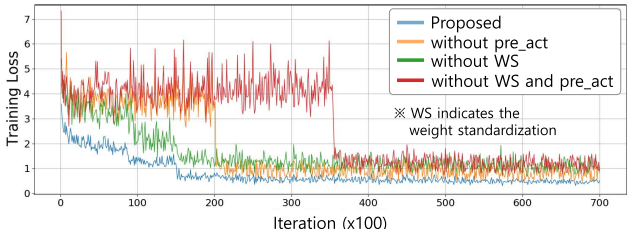

Fig. 10. Comparison of loss convergence according to various settings of the convolution block in the proposed decoder (best view in colors).

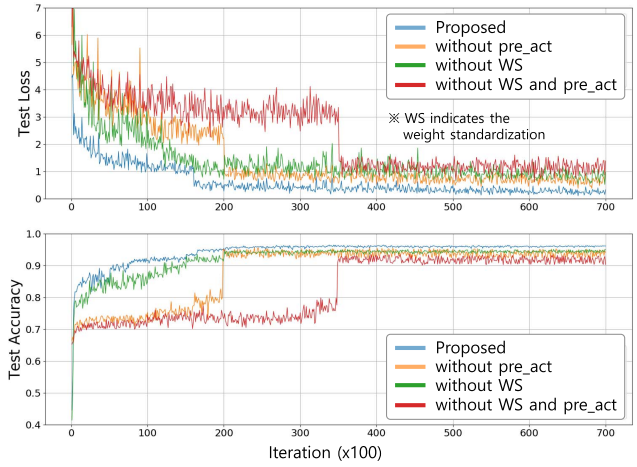

Fig. 11. Convergence of the test loss and accuracy according to various settings of the convolution block in the proposed decoder (best view in colors).

the absence of the direct connectivity between the scale spaces of the depth map. Different from those methods, the proposed decoder generates depth residuals at each level of the Laplacian pyramid and exploits the difference across the scale spaces, which is highly relevant to object boundaries. As a result, the proposed method outperforms other approaches based on different pyramid structures as shown in Table VI. Consequentially, experimental results show that the proposed Laplacian pyramid-based depth residuals paves the way for monocular depth estimation.

V. CONCLUSION

A novel method for depth estimation from a single monocular image has been proposed in this paper. The key idea of the proposed method is to decompose the decoding process by exploiting the Laplacian pyramid for fully utilizing underlying properties of well-encoded features. Furthermore, to make such decoding process more efficient, we propose to apply weight standardization to the pre-activation convolution block of the decoder, which improves the flow of gradients and thus contributes to generate the reliable depth map with the clear depth boundaries. Experimental results on benchmark datasets constructed under various indoor and outdoor environments show that the proposed method is effective for monocular depth estimation.

REFERENCES

[1] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1226–1238, Sep. 2002.

[2] D. Hoiem, A. N. Stein, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from a single image," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[3] 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.

[4] Z. Niu, G. Hua, X. Gao, and Q. Tian, "Context aware topic model for scene recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2743–2750.

[5] D. Tao, Y. Guo, Y. Li, and X. Gao, "Tensor rank preserving discriminant analysis for facial recognition," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 325–334, Jan. 2018.

[6] K. Karsch, C. Liu, and S. B. Kang, "Depth extraction from video using non-parametric sampling," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2014, pp. 775–788.

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

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.

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

[10] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.

[11] R. Garg, G. Carneiro, B. G. V. Kumar, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 740–756.

[12] W. Liu, X. Ma, Y. Zhou, D. Tao, and J. Cheng, "p-Laplacian regularization for scene recognition," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2927–2940, Aug. 2019.

[13] C. Chun, D. Park, W. Kim, and C. Kim, "Floor detection based depth estimation from a single indoor scene," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 3358–3362.

[14] 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.

[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] J. L. Herrera, C. R. del-Blanco, and N. Garcia, "Automatic depth extraction from 2D images using a cluster-based learning framework," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3288–3299, Jul. 2018.

[17] 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.*, Dec. 2014, pp. 2366–2374.

[18] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1119–1127.

[19] 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.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[20] Y. Gan, X. Xu, W. Sun, and L. Lin, "Monocular depth estimation with affinity, vertical pooling, and label enhancement," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 232–247.

[21] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 161–169.

[22] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2215–2223.

[23] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.

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

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

[26] Y. Cao, T. Zhao, K. Xian, C. Shen, Z. Cao, and S. Xu, "Monocular depth estimation with augmented ordinal depth relationships," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2674–2682, Aug. 2020.

[27] H. Mohaghegh, N. Karimi, S. M. R. Soroushmehr, S. Samavi, and K. Najarian, "Aggregation of rich depth-aware features in a modified stacked generalization model for single image depth estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 3, pp. 683–697, Mar. 2019.

[28] Y. Zuo, Y. Fang, Y. Yang, X. Shang, and Q. Wu, "Depth map enhancement by revisiting multi-scale intensity guidance within coarse-to-fine stages," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4676–4687, Dec. 2020, doi: 10.1109/TCSVT.2019.2962867.

[29] J. Ma, H. Zhang, P. Yi, and Z. Wang, "SCSCN: A separated channel-spatial convolution net with attention for single-view reconstruction," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8649–8658, Oct. 2020.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, May 2015, pp. 1–14.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[34] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 630–645.

[36] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, "Micro-batch training with batch-channel normalization and weight standardization," 2019, *arXiv:1903.10520*. [Online]. Available: http://arxiv.org/abs/1903.10520

[37] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," 2019, *arXiv:1907.10326*. [Online]. Available: http://arxiv.org/abs/1907.10326

[38] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. Neural Inf. Process. Syst.*, Dec. 2017, pp. 1–4.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[40] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*. [Online]. Available: http://arxiv.org/abs/1711.05101

[41] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[42] A. J. Amiri, S. Y. Loo, and H. Zhang, "Semi-supervised monocular depth estimation with left-right consistency using deep neural network," 2019, *arXiv:1905.07542*. [Online]. Available: http://arxiv.org/abs/1905.07542

[43] 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. (ICCV)*, Dec. 2015, pp. 2650–2658.

[44] 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.*, Dec. 2016, pp. 2658–2666.

[45] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (DV)*, Oct. 2016, pp. 239–248.

[46] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "GeoNet: Geometric neural network for joint depth and surface normal estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 283–291.

[47] Z. Hao, Y. Li, S. You, and F. Lu, "Detail preserving depth estimation from a single image using attention guided networks," in *Proc. Int. Conf. 3D Vis. (DV)*, Sep. 2018, pp. 304–313.

[48] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[49] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

Minsoo Song received the B.S. degree from the Department of Electronics Engineering, Konkuk University, Seoul, South Korea. He is currently a Graduate Student with the Department of Electronic, Information, and Communication Engineering, Konkuk University. His research interests include scene understanding and image enhancement, including depth estimation, super-resolution, and object tracking.

Seokjae Lim received the B.S. degree from the Department of Electronics Engineering, Konkuk University, Seoul, South Korea. He is currently a Graduate Student with the Department of Electronic, Information, and Communication Engineering, Konkuk University. His research interests include image enhancement, computer vision, deep learning, and biometrics.

Wonjun Kim (Member, IEEE) received the B.S. degree from the Department of Electronic Engineering, Sogang University, Seoul, South Korea, in 2006, the M.S. degree from the Department of Information and Communications, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2008, and the Ph.D. degree from the Department of Electrical Engineering, KAIST, in 2012. From September 2012 to February 2016, he was a Research Staff Member of the Samsung Advanced Institute of Technology (SAIT), Suwon, South Korea. Since March 2016, he has been with the Department of Electrical and Electronics Engineering, Konkuk University, Seoul, South Korea, where he is currently an Associate Professor. His research interests include image and video understanding, computer vision, pattern recognition, and biometrics, with an emphasis on background subtraction, saliency detection, and face and action recognition. He has served as a regular reviewer for over 30 international journal articles, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE ACCESS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CYBERNETICS, IEEE SIGNAL PROCESSING LETTERS, and *Pattern Recognition*.