Monocular Human Depth Estimation via Pose Estimation

JINYOUNG JUN\textsuperscript{1}, JAE-HAN LEE\textsuperscript{1}, CHUL LEE\textsuperscript{2}, (Member, IEEE), AND CHANG-SU KIM\textsuperscript{1}, (Senior Member, IEEE)

\textsuperscript{1}School of Electrical Engineering, Korea University, Seoul 02841, South Korea
\textsuperscript{2}Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Chang-Su Kim (changsukim@korea.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Government of Korea (MSIT) under Grant NRF-2018R1A2B3003896, Grant NRF-2019R1A2C4069806, and Grant NRF-2021R1A4A1031864.

\textbf{ABSTRACT} We propose a novel monocular depth estimator, which improves the prediction accuracy on human regions by utilizing pose information. The proposed algorithm consists of two networks — PoseNet and DepthNet — to estimate keypoint heatmaps and a depth map, respectively. We incorporate the pose information from PoseNet to improve the depth estimation performance of DepthNet. Specifically, we develop the feature blending block, which fuses the features from PoseNet and DepthNet and feeds them into the next layer of DepthNet, to make the networks learn to predict the depths of human regions more accurately. Furthermore, we develop a novel joint training scheme using partially labeled datasets, which balances multiple loss functions effectively by adjusting weights. Experimental results demonstrate that the proposed algorithm can improve depth estimation performance significantly, especially around human regions. For example, the proposed algorithm improves the depth estimation performance on the human regions of ResNet-50 by 2.8% and 7.0% in terms of $\delta_1$ and RMSE, respectively, on the proposed HD + P dataset.

\textbf{INDEX TERMS} Monocular depth estimation, human pose estimation, human depth estimation, loss rebalancing strategy.

\section{I. INTRODUCTION}

Depth estimation, which predicts the distance of a scene point corresponding to each pixel in a 2D image, is one of the fundamental tasks in computer vision for inferring the 3D geometry of the scene. The estimated depth map can be used to reconstruct 3D volumetric data, including point cloud or triangle mesh [1]. Traditionally, 3D information has been recovered by estimating depth maps from multiple images based on geometric constraints. In particular, stereo matching-based depth estimation, which computes the disparity between the images observed from two viewpoints by simulating human eyes, has been developed most actively [2]–[4]. Although the geometry-based algorithms can estimate an accurate depth map, they require calibrated image pairs or sequences. Furthermore, it is generally difficult to estimate accurate disparities in textureless regions. In contrast, monocular depth estimation uses only a single image to estimate a depth map and does not require additional equipments for capturing and calibrating multiple images.

Because only a single camera is available in most scenarios, monocular depth estimation has been employed in many computer vision applications, such as 2D-to-3D image and video conversion [5], augmented reality [6], autonomous driving [7], surveillance [8], and 3D model generation [9]. Due to its practical importance, monocular depth estimation has recently been researched actively, and various techniques have been developed to improve its accuracy [10]–[19].

Since different 3D scenes can be projected onto the same 2D image, estimating the depth map from a single image is ill-posed. Early approaches inferred depths by modeling the relation between regions based on prior assumptions on depth maps [11], [12], [21]. However, as they are based on manually defined features, they may provide unreliable results, especially in regions with small objects or ambiguous colors. Recently, with the advances in deep learning, many monocular depth estimation algorithms using convolutional neural networks (CNNs) have been developed to yield more accurate results. For example, multi-scale architecture [13],
robustness to global depth scales [13], ordinal regression of depths [15], Fourier domain analysis [16], whole strip masking [17], relative depth estimation [18], and reliable boundary processing [19] have been developed or exploited for accurate monocular depth estimation.

Although a lot of researches have been carried out to increase the accuracy of monocular depth estimation, little effort has been made to consider the characteristics of specific objects in a scene. In particular, humans are the main objects of interest in many practical applications, such as gaming and surveillance. However, the depth estimation of humans in a scene is more difficult than that of other objects due to flexible body parts, diverse clothing, and different poses. Figure 1 shows examples, in which a conventional algorithm yields less reliable depths for human regions. Thus, it is important to develop a monocular depth estimator that predicts the depths for human regions, as well as other regions, reliably and accurately by exploiting the characteristics of humans.

In this paper, we propose a novel monocular depth estimator, called HDEP (human depth estimation via pose), to improve the prediction accuracy on human regions by utilizing pose information. The proposed algorithm is composed of two networks: PoseNet, which predicts heatmaps of human skeletal keypoints, and DepthNet, which estimates a depth map for an entire image. To improve the depth estimation performance on human regions, DepthNet uses the pose information from PoseNet. To this end, we develop the feature blending block to fuse the features extracted from DepthNet with those from PoseNet. Finally, we develop a novel joint training scheme using partially labeled datasets, which balances multiple loss functions for depths and poses by adjusting the weights.

![FIGURE 1. Examples of monocular human depth estimation. The proposed algorithm improves the depth estimation accuracy on human regions, as compared to the baseline Chen et al. [20].](image1)

PoseNet and DepthNet. We then develop an effective joint training scheme for the HDEP algorithm using partially labeled data.

- We construct the HD + P dataset, in which an image is annotated with a depth map and a set of human keypoint heatmaps. We will make it publicly available.
- We experimentally demonstrate that the proposed HDEP algorithm provides excellent monocular depth estimation performance qualitatively and quantitatively on multiple datasets, especially on human regions.

The rest of this paper is organized as follows. Section II reviews related work on monocular depth estimation and pose estimation. Section III describes the proposed HDEP algorithm. Section IV discusses experimental results and analyzes the impacts of the key components in HDEP on the results. Finally, Section V concludes the paper.

**II. RELATED WORK**

**A. LEARNING-BASED MONOCULAR DEPTH ESTIMATION**

With the recent advances in deep learning and the construction of big datasets [21], [24]–[26], extensive researches have been conducted on learning-based monocular depth estimation. These researches have focused on designing better network architectures [13]–[15], [20], [27], [28] and formulating more effective loss functions [14], [19], [26], [29], [30]. In addition, attempts have been made to improve the performance by applying ordinal regression to quantized depths [15], enforcing geometric constraints in the 3D space [31], separating low- and high-level features [32], or balancing weights for multiple loss functions [33].

Depth information provides useful clues about closely related information in a scene, such as surface normals and semantic segmentation labels. Based on this observation, several algorithms [29], [34]–[36] have been developed to solve the depth estimation task jointly with other related tasks. However, if two tasks are closely related, the features extracted for each task may be different, so an incorrect prediction in one task may adversely affect the other task. In this work, we aim to improve the depth estimation performance by employing pose features additionally. Note that pose...
estimation is weakly related to depth estimation in that the former localizes only the keypoints of a human, whereas the latter predicts the depth value of every pixel in an image.

B. HUMAN POSE ESTIMATION
Human pose estimation is the task of locating human joints — also known as keypoints — in an image. It can be divided into 2D and 3D pose estimation according to the dimensions of estimated coordinates of joints. In 2D pose estimation, multiscale features have been adopted to deal with humans of various sizes in images [37]–[40]. The simplest approach to 3D pose estimation is to lift estimated 2D poses to 3D poses [41]–[43]. Alternatively, the likelihood of a joint in each voxel can be directly estimated using a 3D voxel model [44], or three marginal 2D heatmaps in the xy-, yz-, zx-plains can be predicted [45].

Datasets for 3D pose estimation have been constructed in indoor studios with markers on human joints to collect 3D ground-truth coordinates [46]–[48]. However, these datasets provide the depth information of only human regions or only human keypoints in simple and controlled environments. We cannot use these datasets, because we aim at estimating the depth of every pixel in an image, which contains humans with cluttered and real backgrounds.

C. HUMAN DEPTH ESTIMATION
Several algorithms have been developed for human depth estimation. Geometric details of human objects, such as cloth wrinkles and folds, have been considered in [22], [49]. Specifically, Tang et al. [22] estimated the shape of a human from an RGB image by combining segmentation results with 3D pose information. Tan et al. [49] proposed a self-supervised human depth estimation framework from monocular videos. In addition, Lin and Lee [23] developed a multi-person 3D pose estimation algorithm in the camera coordinate space, which predicts the overall distance of each person from the camera. Figure 2 shows how the proposed algorithm is different from these conventional algorithms. Specifically, our objective is to estimate the pixelwise depths of both human and other regions instead of extracting the 3D shape of humans only [22], [49], or the distance of each person from the camera [23].

III. PROPOSED ALGORITHM
Figure 3 shows an overview of the proposed HDEP algorithm, in which DepthNet estimates a depth map \( \hat{D} \) and PoseNet outputs a set of heatmaps \( \hat{H} \) for human skeletal keypoints. Features extracted by the PoseNet encoder are fed into the DepthNet encoder through the feature blending block. The pose information is estimated to improve the depth estimation performance; the pose estimation in itself is not an objective of this work. To train the networks, we define the overall loss \( \ell_{all}(\hat{D}, \hat{H}, D, H) \) between the estimated \( \hat{D} \) and \( \hat{H} \) and their ground-truth \( D \) and \( H \), which is a weighted sum of the eight loss functions in Table 1 for different aspects of depth and pose information.

A. NETWORK ARCHITECTURE
In Figure 3, both DepthNet and PoseNet have the same encoder-decoder architecture, where the encoders extract deep features for depths and human poses, respectively. For each encoder, we employ ResNet-50 [50], which has been commonly used in recent depth estimators [14], [19]. However, note that any conventional monocular depth estimator can be used as DepthNet, as will be discussed in Section IV-C. The decoders take the aforementioned features as input and...
yield a single depth map and multiple keypoint heatmaps. Since the PoseNet decoder estimates heatmaps for human skeletal keypoints, it has more output channels than DepthNet, which yields a single-channel depth map. More specifically, PoseNet outputs 16 channels as specified in the MPII dataset [51], and each channel represents a heatmap for each keypoint. Note that, since both DepthNet and PoseNet have the same architecture and a pretrained PoseNet is unavailable in general, we train both networks as will be described in Section IV-B.

Recently, multi-task learning has been adopted to estimate monocular depths jointly with surface normals, edge contours, and semantic segmentation labels [29], [34], [52]–[54]. Such strategies enable a network to learn various cues to understand the geometry of a scene by exploiting commonalities and differences among tasks. In contrast, pose estimation and depth estimation are less related; pose estimation aims to locate skeletal keypoints only, while depth estimation aims to predict a dense depth map of an entire scene. Thus, if depth and pose are trained jointly in a fully-shared network, they may fail to share common features effectively and the training of one network may make the training of the other more difficult or even divergent.

We transfer PoseNet features unidirectionally to DepthNet to exploit human skeletal information and improve the depth estimation performance on the corresponding human region. To this end, we develop the feature blending block, which enables DepthNet to exploit the pose information from PoseNet. As illustrated in Figure 3, both DepthNet and PoseNet reduce the spatial resolutions of feature maps in five stages. From the second to fourth decimation stage, the feature blending block adds the decimated features of PoseNet to those of DepthNet and then feeds the sum into the next layer of DepthNet. Specifically, let \( F^\text{out}_D \) and \( F^\text{out}_P \) denote the decimated output features of DepthNet and PoseNet, respectively. Then, the input feature \( F^\text{in}_D \) of the next layer of DepthNet is given by

\[
F^\text{in}_D = \kappa_P F^\text{out}_P + \kappa_D F^\text{out}_D \tag{1}
\]

where \( \kappa_P \) and \( \kappa_D \) are the weights for \( F^\text{out}_P \) and \( F^\text{out}_D \) to control the relative contributions of the two features. Note that, instead of the addition in (1), various blending strategies can be adopted. We will discuss the effects of different blending strategies on the performance in Section IV-D3.

### B. LOSS FUNCTIONS

To train DepthNet and PoseNet jointly, we employ the eight losses in Table 1, which affect the training of the networks in different aspects. Each loss is computed using a depth map, a set of heatmaps, or both.

The first five losses \( \ell_d, \ell_n, \ell_m, \ell_x, \) and \( \ell_y \) are computed using a depth map only. First, the depth loss \( \ell_d \) is the \( L_1 \)-norm between the ground-truth depth \( D \) and its estimate \( \hat{D} \). Second, the normal loss \( \ell_n \) is obtained from the cosine similarities between the normal vectors \( n_i \) and \( \hat{n}_i \) of pixel \((i, j)\), which are approximated from \( D \) and \( \hat{D} \) using depth gradients as done in [19]. Third, the mean-removed loss \( \ell_m \) measures the \( L_1 \)-norm between mean-removed \( D \) and \( \hat{D} \) to compare relative depths with respect to the average depth of a scene [18]. Fourth, the gradient losses \( \ell_x \) and \( \ell_y \) are the \( L_1 \)-norms between gradients of \( D \) and \( \hat{D} \), where \( \nabla_x \) and \( \nabla_y \) are the horizontal and vertical gradient operators, respectively. The gradient losses make the networks focus on high-frequency contents and thus reconstruct depth boundaries more sharply.

Next, the heatmap loss \( \ell_h \) is defined as the average \( L_1 \)-norm between the ground-truth heatmaps \( H_j \in H \) and their estimates \( \hat{H}_j \). \( \mathcal{J} \) is a set of human skeletal keypoints. We detect 16 kinds of keypoints as specified in [51], and thus \( |\mathcal{J}| = 16 \). The heatmap loss enables PoseNet to extract useful features for locating keypoints and identifying human poses.

Finally, we design two losses \( \ell_g \) and \( \ell_p \) to consider the accuracy of both depth and pose estimation jointly. They compute depth prediction errors around keypoints. More specifically, we define \( \ell_g \) as the average \( L_1 \)-norm between \( D \) and \( \hat{D} \) near the ground-truth keypoints, given by

\[
\ell_g = \frac{1}{|\mathcal{J}|HW} \sum_{j \in \mathcal{J}} \| (D - \hat{D}) \circ T_\tau(H_j) \|_1 \tag{2}
\]

where \( T_\tau(\cdot) \) denotes the binarization with threshold \( \tau \) and the operator \( \circ \) denotes the element-wise multiplication. \( \ell_p \) is defined similarly to \( \ell_g \) in (2). However, because \( \ell_g \) cannot be computed on a dataset without keypoint labels, we define \( \ell_p \) using the predicted keypoints. More specifically, predicted keypoints in \( \hat{H}_j \) are used instead of ground-truth heatmaps in \( H_j \). In this work, we fix \( \tau \) to 0.01.

### C. TRAINING BASED ON LOSS REBALANCING

The overall loss \( \ell_{\text{all}} \) is defined as a weighted sum of the eight loss functions in Table 1, given by

\[
\ell_{\text{all}} = \sum_{k \in \{d,n,m,x,y,h,g,p\}} w_k \ell_k \tag{3}
\]

where \( w_k \) is the weight for \( \ell_k \). As different losses are in different scales, each loss contributes to the overall loss differently. Also, the losses interact with one another and tend to fluctuate during training. Hence, it is necessary to

| Loss Type       | Expression                                      |
|-----------------|-------------------------------------------------|
| Depth loss      | \( \ell_d = \frac{1}{\|\mathbf{D}\|_1} \| \mathbf{D} - \mathbf{\hat{D}} \|_1 \) |
| Normal loss     | \( \ell_n = 1 - \frac{1}{\|\mathbf{D}\|_1} \sum_{i=1}^h \sum_{j=1}^w \frac{n_{ij} \cdot \hat{n}_{ij}}{\|n_{ij}\| \|\hat{n}_{ij}\|} \) |
| Mean-removed loss | \( \ell_m = \frac{1}{\|\mathbf{D}\|_1} \| (\mathbf{D} - \mu) - (\mathbf{\hat{D}} - \mu) \|_1 \) |
| Gradient losses | \( \ell_x = \frac{1}{\|\mathbf{D}\|_1} \| \nabla_x \mathbf{D} - \nabla_x \mathbf{\hat{D}} \|_1 \) \( \ell_y = \frac{1}{\|\mathbf{D}\|_1} \| \nabla_y \mathbf{D} - \nabla_y \mathbf{\hat{D}} \|_1 \) |
| Heatmap loss    | \( \ell_h = \frac{1}{|\mathcal{J}|HW} \sum_{j \in \mathcal{J}} \| \mathbf{H}_j - \mathbf{\hat{H}}_j \|_1 \) |
| Joint losses    | \( \ell_g = \frac{1}{|\mathcal{J}|HW} \sum_{j \in \mathcal{J}} \| (\mathbf{D} - \mathbf{\hat{D}}) \circ T_\tau(H_j) \|_1 \) \( \ell_p = \frac{1}{|\mathcal{J}|HW} \sum_{j \in \mathcal{J}} \| (\mathbf{D} - \mathbf{\hat{D}}) \circ T_\tau(H_j) \|_1 \) |
balance their contributions to the overall loss. To address this balancing issue, we develop a loss rebalancing scheme, inspired by [33], which adjusts loss weights $w_k$ adaptively in a periodic manner based on the rate at which each loss decreases during training.

Loss rebalancing consists of two steps: weight initialization and weight rebalancing. First, in the initialization, we equalize all weights so that each loss contributes equally to the overall loss. Then, in the rebalancing, the weight $w_k^t$ for the $k$th loss at the $t$th period is updated from $w_k^t-1$ by monitoring how fast the $k$th loss is reduced. In this work, one fourth of an epoch is defined as a period. The ratio $P_k^t$ of the $k$th loss in the overall loss is defined as

$$P_k^t = \frac{\ell_k^t}{\ell_{all}^t} \tag{4}$$

where $\ell_k^t$ and $\ell_{all}^t$ are the averages of $\ell_k$ and $\ell_{all}$, respectively, over the $t$th period. Then, the weight is updated as

$$w_k^t = w_k^{t-1} \left( 1 - \lambda \frac{\Delta P_k^t}{P_k^t} \right) \tag{5}$$

where $\Delta P_k^t = P_k^t - P_k^{t-1}$. The hyper-parameter $\lambda$ controls the rebalancing strategy. If $\lambda = 0$, loss weights are unchanged during training. At $\lambda < 0$, the weight for a slowly decreasing loss gets larger to make all losses balanced. This is because, if a certain loss is more difficult to reduce and thus decreases more slowly than the other losses, we have $\Delta P_k^t = P_k^t - P_k^{t-1} > 0$ and $w_k^t > w_k^{t-1}$. On the contrary, at $\lambda > 0$, the weight for a difficult loss gets smaller. To summarize, the training focuses on easy (or difficult) losses when $\lambda$ is positive (or negative).

Both depth and keypoint coordinates are needed to train the networks using the eight losses. However, most datasets have either depth or keypoint labels only. While the DIH dataset [55] contains both depth and keypoint labels, its images lack diversity in backgrounds and human poses. Thus, we use other datasets that contain either depth or keypoint labels for training. However, the overall loss $\ell_{all}$ in (3) and the weight $w_k^t$ in (5) for some $k$ cannot be computed using such partially labeled datasets, for the corresponding losses are undefined. For example, $\ell_d$ is undefined if depth labels are unavailable, while $\ell_k$ is undefined if keypoint labels are unavailable. Moreover, the joint losses $\ell_g$ and $\ell_p$ demand both depth and keypoint labels.

To address the aforementioned challenge on the joint training of the networks using partially labeled datasets, we develop an effective training strategy. First, when depth or keypoints are unavailable, some losses cannot be computed. We set such undefined losses to 0 when computing the overall loss $\ell_{all}$ in (3) during training — i.e., in the backpropagation process. Second, we augment the loss rebalancing scheme to consider partially labeled data. As done for the overall loss, an undefined loss $\ell_k$ is set to 0. Then, the weight in (5) cannot be computed. To address this problem, we replace the undefined loss $\ell_k$ with the average value over valid samples. Specifically, let $\ell_{k,i}$ denote the $k$th loss for the $i$th training sample and $\mathbb{I}(\cdot)$ be the indicator function

$$\mathbb{I}(x) = \begin{cases} 1, & \text{if } x \neq 0, \\ 0, & \text{if } x = 0. \end{cases} \tag{6}$$

Then, we define the pseudo loss $\tilde{\ell}_k$ as

$$\tilde{\ell}_k = \frac{\sum_i \ell_{k,i}}{\sum_i \mathbb{I}(\ell_{k,i})} \tag{7}$$

Next, we replace the undefined loss $\ell_k$ with the pseudo loss.

Figure 4 compares weighted loss trends over epochs during training. In the equal weighting in Figure 4(a), the overall loss is dominated by large losses, such as $\ell_d$ and $\ell_m$, whereas the contributions of small losses, such as $\ell_g$ and $\ell_n$, are negligible. In Figure 4(b), the proposed rebalancing scheme effectively equalizes the contributions of all losses to the overall loss by adjusting the weights adaptively. This is possible even with partially labeled data.
IV. EXPERIMENTAL RESULTS

A. DATASETS

We use four datasets for training: NYUv2 [24], DIH [55], MPII [51], and HD + P. Figure 5 shows examples of RGB images and their depth and/or keypoint labels in each dataset, and Figure 6 shows examples of heatmap representation of different joints. Table 2 summarizes available labels in each dataset.

1) DATA PREPROCESSING

As mentioned earlier, the datasets have different numbers and kinds of keypoints. Furthermore, depth maps contain invalid regions at occlusions or transparent or reflective objects. Therefore, we perform data preprocessing to match pose and depth labels from different datasets. For keypoints, MPII [51] and DIH [55] provide 16 and 17 labels, respectively, as listed in Figure 8(a). We match the labels in DIH to those in MPII. Specifically, we remove “Head,” “Left eye,” and “Right eye” from DIH and interpolate “Pelvis” and “Thorax” from the midpoints of both hips and both shoulders, respectively. Since the “Head” label in MPII corresponds to the upper region of a head, we use “HeadUp” in DIH instead of “Head” in MPII.

For depth maps, we first identify invalid regions for each dataset. Specifically, for the DIH dataset, we remove depth values outside the range in the hardware specification of Kinect 2, whereas we mask invalid regions manually for the HD + P dataset. Then, we fill in invalid regions using the colorization scheme [56]. Figure 9 shows examples of raw depth maps and the corresponding preprocessed depth maps.

B. IMPLEMENTATION DETAILS

1) NETWORK ARCHITECTURE

We employ ResNet [50] as the encoder backbones of both DepthNet and PoseNet. An input image is resized to 304 × 228, and the size of each encoder output is 10 × 8 × 2048. The encoder consists of four layers with three, four, six, and three bottleneck blocks, each of which has three convolutional layers with kernel sizes 1, 3, and 1 with

| TABLE 2. Ground-truth labels available in each dataset. |
|----------------|----------------|----------------|
|                  | Training set   | Test set       |
| NYUv2 [24]      | Depth          | Depth          |
| MPII [51]       | Keypoints      | Keypoints      |
| DIH [55]        | Depth & Keypoints | Depth & Keypoints |
| HD+P            | Depth          | Depth          |

| TABLE 3. Comparison of the numbers of scenes and humans in the DIH [55] and HD + P datasets. |
|--------------------------------|--------|--------|
|                                | DIH    | HD+P   |
| # of scenes Training           | 7      | 119    |
| Test                          | 5      | 26     |
| # of humans Training          | 9      | 11     |
| Test                          | 1      | 2      |

FIGURE 6. Examples of RGB images in MPII and several keypoint heatmaps.

FIGURE 7. Examples of challenging images in the HD + P dataset.

151449
J. Jun et al.: Monocular Human Depth Estimation via Pose Estimation

FIGURE 8. (a) Keypoint labels in the MPII [51] and DIH [55] datasets. (b) Annotated keypoint labels in a DIH image. Unchanged keypoints are in green, removed ones in red, and generated ones in blue.

FIGURE 9. Examples of raw and preprocessed depth maps.

| Index | MPII/HD+P | DIH |
|-------|-----------|-----|
| 1     | R. ankle  | R. ankle |
| 2     | R. knee   | R. knee |
| 3     | R. hip    | R. hip |
| 4     | L. hip    | L. hip |
| 5     | L. knee   | L. knee |
| 6     | L. ankle  | L. ankle |
| 7     | R. wrist  | R. wrist |
| 8     | R. elbow  | R. elbow |
| 9     | R. shoulder | R. shoulder |
| 10    | L. shoulder | L. shoulder |

2) TRAINING

We train the proposed HDEP algorithm in three steps. First, we train PoseNet using the MPII and DIH datasets, which contain keypoint labels. We use the Adam optimizer [57] for 140 epochs with an initial learning rate of $10^{-3}$. We decrease the learning rate by a factor of 0.1 at the 70th and 110th epochs, respectively. In this step, only the heatmap loss $\ell_h$ is used. Second, we train DepthNet only using the NYUv2, DIH, and HD + P datasets containing depth labels. We perform the same training as the first step, except that the depth loss $\ell_d$ is used instead of $\ell_h$. Finally, we retrain (fine-tune) both DepthNet and PoseNet jointly for 60 epochs using all four datasets with the overall loss $\ell_{all}$ in (3). For the loss rebalancing, we gradually reduce the hyper-parameter $\lambda$ in (5) from 3 to −3 as done in [33]. Also, in the feature blending block, $\kappa_P$ and $\kappa_D$ in (1) are set to 0, 0, 0.1, and 0, 1, 1 at the first, second, and third steps, respectively.

C. DEPTH ESTIMATION RESULTS

1) EVALUATION METRIC

We assess the depth estimation performance using two protocols. The first protocol (Entire) evaluates the estimation accuracy on an entire depth map, which is commonly used for assessing depth estimation performance. However, the performance evaluated using this protocol is strongly affected by large background regions, therefore this protocol may fail to faithfully reflect the accuracy on human regions. Therefore, to assess the depth estimation performance on human regions, we develop another protocol (Human-oriented) that measures the accuracy only on the regions containing humans. Specifically, we compute the accuracy only within the bounding boxes that contain all 16 ground-truth keypoints. For both evaluation protocols, three evaluation metrics for depth estimation performance are employed, which are defined as

$$\delta_n = \frac{1}{N} \left\{ \left| d_i \right| \max \left\{ \frac{d_i}{d^*_i}, 1 \right\} < 1.25^a \right\}$$  \hspace{1cm} (8)

$$\text{RMSE} = \left( \frac{1}{N} \sum_i (d_i - \hat{d}_i)^2 \right)^{0.5}$$  \hspace{1cm} (9)

$$\text{ARD} = \frac{1}{N} \sum_i \left| \frac{\hat{d}_i - d_i}{d_i} \right|$$  \hspace{1cm} (10)

where $d_i$ and $\hat{d}_i$ denote the ground-truth and predicted depths of pixel $i$, respectively, and $N$ is the number of pixels in a depth map.

2) BASELINES

As mentioned in Section III-A, any monocular depth estimator can be used with the proposed HDEP algorithm. We test four conventional depth estimators: Chen et al. [20], Hu et al. [19], ResNet-50 [50], and EfficientNet-B3 [58]. Note that ResNet and EfficientNet are commonly used as backbone networks for both depth estimation [14], [59], [60].
TABLE 4. Comparison of the depth estimation results on the DIH and HD + P datasets using the two evaluation protocols. For δ scores, a higher score indicates a better result, while lower RMSE and ARD values are better results. The scores in red indicate that the proposed algorithm improves the performance of each conventional estimator, while those in blue indicate that it degrades the performance.

|           | DIH                          | HD + P                      |
|-----------|------------------------------|-----------------------------|
|           | δ₁ (†) δ₂ (†) δ₃ (†) RMSE (↓) | ARD (↓)                    |
| Entire    | 0.753 0.920 0.982 0.403      | 0.184 0.721 0.937 0.988     | 0.594 0.560 0.170 |
| + Proposed| 0.772 0.930 0.985 0.390      | 0.170 0.752 0.953 0.990     | 0.560 0.170      |
| Hu et al. [19] + Proposed | 0.804 0.930 0.982 0.392 | 0.162 0.728 0.940 0.987 | 0.605 0.183      |
|            | 0.793 0.920 0.985 0.392 | 0.171 0.753 0.945 0.988 | 0.577 0.169      |
| ResNet-50 [50] + Proposed | 0.770 0.915 0.973 0.400 | 0.177 0.672 0.910 0.973 | 0.693 0.214      |
|            | 0.762 0.923 0.975 0.395 | 0.180 0.684 0.913 0.976 | 0.671 0.206      |
| EfficientNet-B3 [58] + Proposed | 0.777 0.919 0.983 0.383 | 0.169 0.773 0.953 0.989 | 0.552 0.165      |
|            | 0.761 0.924 0.984 0.379 | 0.170 0.779 0.953 0.990 | 0.541 0.164      |

TABLE 5. Results of the Wilcoxon signed-rank test against Chen et al.’s algorithm on the HD + P datasets.

|           | δ₁    | δ₂    | δ₃    | RMSE  | ARD   |
|-----------|-------|-------|-------|-------|-------|
| Entire    | 0.331 | 0.309 | 0.202 | 0.665 | 0.669 |
| Human-oriented | 0.297 | 0.175 | 0.069 | 0.692 | 0.713 |

TABLE 6. Comparison of the depth estimation results on the NYUv2 dataset.

|           | δ₁ (†) δ₂ (†) δ₃ (†) RMSE (↓) | ARD (↓) |
|-----------|------------------------------|---------|
| Chen et al. [20] + Proposed | 0.744 0.943 0.985 0.649 | 0.164 0.156 |
| Hu et al. [19] + Proposed | 0.758 0.954 0.989 0.615 | 0.157 0.159 |
| ResNet-50 [50] + Proposed | 0.690 0.916 0.973 0.741 | 0.190 0.193 |
| EfficientNet-B3 [58] + Proposed | 0.768 0.954 0.988 0.636 | 0.153 0.150 |

3) QUANTITATIVE AND QUALITATIVE EVALUATION

Table 4 compares the quantitative performances on the DIH [55] and HD + P datasets. In the Entire protocol, the proposed algorithm provides comparable or better estimation results than the backbone networks. Specifically, on the DIH dataset, the proposed algorithm improves the performance of Chen et al.’s estimator, is comparable to ResNet-50 and EfficientNet-B3, and slightly worsens Hu et al.’s estimator. This inconsistency is partly due to the lack of data diversity in the DIH dataset. In contrast, on the HD + P dataset, the proposed algorithm outperforms the backbone networks in all cases. Moreover, in the Human-oriented protocol, the proposed algorithm outperforms the backbone networks in almost all cases — in many cases by large margins. For example, the proposed algorithm provides a 0.044 higher δ₁ score than Chen et al.’s estimator on the DIH dataset and yields a 0.041 lower RMSE value than ResNet-50 on the HD + P dataset. These results indicate that the proposed algorithm improves the depth estimation performance on human regions significantly, while providing comparable or even better performance on entire images.

We also conducted a statistical test to determine the performance improvement using the Wilcoxon signed-rank test [62]. The null hypothesis is that the proposed algorithm degrades the estimation performance over the baseline for each evaluation metric. The signed-rank sum is normalized by the rank sum so that it lies in the range [−1, 1]. A positive number indicates the performance improvement over the baseline, while a negative number implies the performance degradation. Table 5 shows the normalized signed-rank sums for comparing the proposed algorithm against Chen et al.’s. The obtained p-values are p < 0.01 in all metrics except for δ₃ in the Human-oriented protocol, where p = 0.11. The results imply that all the null hypotheses are rejected for all metrics, so the proposed algorithm indeed improves the estimation performance of the baseline.

and pose estimation [61]. For Chen et al. [20] and Hu et al. [19], we use ResNet-34 [50] as the backbones and train them as specified in the respective papers.
Figure 10 qualitatively compares the depth estimation results of the proposed algorithm with those of the baseline networks and visualizes their error maps. The proposed algorithm improves the prediction accuracy and reduces errors compared to the baselines in most cases. In particular, the proposed algorithm yields significantly better results on human regions. For example, the head regions are erroneous in the baselines in the third, sixth, and seventh columns. The proposed algorithm estimates depths in those regions more accurately and reconstructs more faithful head outlines.
Table 6 compares the quantitative performance on the NYU dataset [24] using the Human-oriented protocol, most images of which do not contain humans. The proposed algorithm improves the performance of Chen et al.’s estimator and EfficientNet-B3, while slightly worsening Hu et al.’s estimator and ResNet-50. Because the pose information from PoseNet for images without humans does not provide useful information to DepthNet, the proposed algorithm provides the depth estimation performance comparable to the baselines.

To demonstrate the generalization capability of the proposed algorithm, we compare the performance on the validation set in MPII [51] and the test set in the Human3.6M dataset [46], which contain more diverse scenes and humans than DIH and HD + P, in Figures 11 and 12, respectively. In this test, we use Chen et al.’s estimator [20]. It can be seen that the proposed algorithm improves the estimation performance of the baseline, especially in terms of shape and depth consistency on the human regions, thereby reconstructing human silhouettes faithfully. Figure 11 also shows that the proposed algorithm can improve the estimation performance consistently regardless of the sizes of humans.

Finally, we compare the depth estimation performance according to various factors affecting the difficulty of estimation. To this end, we split the test set of the HD + P dataset into three groups: images with complex poses, occluded humans, and normal scenes. Table 7 compares the estimation performance. These results indicate that the proposed algorithm improves the depth estimation performance on
human regions consistently regardless of the difficulty of the scene.

D. ANALYSIS

We conduct ablation studies to analyze the contributions of the key components in the proposed algorithm: loss functions, weight rebalancing scheme, and feature blending strategy. All experiments are performed using ResNet-50 [50] as the baseline, and the Human-oriented protocol is used for comparison on the HD + P dataset, unless specified otherwise.

1) LOSS FUNCTIONS

To analyze the effectiveness of losses, we train the proposed networks to estimate depth maps using different combinations of losses. Table 8 compares the results quantitatively. First, \((l_n, l_m, l_x, l_y)\) improves the estimation performance. Second, \(l_h\) slightly decreases the depth estimation performance, for it only estimates pose information without considering depth maps. Third, joint training of pose information using \(l_h\) and the joint losses \(l_g\) and \(l_p\) improves the depth estimation performance significantly.

In addition, we analyze how each loss helps to train the network. Figure 13 visualizes some error maps for the eight losses in Table 1 on both the DIH and HD + P datasets. Note that \(l_g\) cannot be computed because ground-truth keypoints are unavailable in the training set of HD + P. Each error map focuses on different aspects of a depth map. For example, \(l_d\) and \(l_m\) attend to overall depth maps while \(l_g\) and \(l_p\) focus on human regions. Also, \(l_x\) and \(l_y\) are activated on human silhouettes where depths vary abruptly.

2) LOSS REBALANCING STRATEGIES

To analyze the effectiveness of the proposed loss rebalancing algorithm, we train the proposed networks with three different weighting strategies: equal weighting without rebalancing, Lee and Kim’s scheme [33], and the proposed rebalancing scheme. Table 9 compares the performances. The proposed algorithm provides significantly better depth estimation performance than the other two schemes by effectively equalizes the contributions of all losses to the overall loss.

In addition, we visualize the weighted sum of error maps for the equal weighting and the proposed rebalancing strategies in Figure 13. We see that the proposed rebalancing scheme focuses more on human regions than the baseline. This confirms the effectiveness of the proposed rebalancing strategy on the depth estimation performance on human regions. That is, as shown in Figure 4, if the weight for each loss is identical, all weighted losses are in different scales, thus the overall loss is dominated by large losses. On the other hand, with the proposed rebalancing scheme, weighted loss values contribute equally to the overall loss.

3) FEATURE BLENDING STRATEGIES

We analyze the effects of different feature blending strategies and the number of feature blending layers. Table 10 compares the quantitative results on different configurations obtained
by using Chen et al. [20] as the baseline. Using more layers in the feature blending block significantly increases the depth estimation performance, especially for concatenation and addition, by exploiting human pose information at various scales. Also, we see that the addition strategy provides better estimation performance than the other two blending strategies.

Finally, we compare an extreme scenario in Table 11, where a single encoder is used with two decoders (ResNet-50-2D). However, this setting degrades the performance, since sharing an encoder between depth and pose hinders the training. This result indicates that the performance improvement is brought by the effective use of pose information.

V. CONCLUSION

We proposed a monocular depth estimation algorithm that exploits pose information for images with humans. The proposed algorithm is composed of two networks, PoseNet and DepthNet, which estimate keypoint heatmaps and a depth map, respectively. We jointly trained the two networks by adopting the feature blending block and the loss weight rebalancing scheme. In addition, we constructed a new dataset for human depth estimation, called HD + P, in which an image is paired with a ground-truth depth map and a set of keypoint heatmaps. Experimental results demonstrated that the proposed algorithm improves depth estimation performance, especially on human regions. However, the proposed algorithm is limited to improve the depth estimation performance on only images containing humans. Thus, an important direction for future work is to develop adaptive schemes for feature transfer from PoseNet to DepthNet by adapting the network architecture according to the existence of humans in images.

REFERENCES

[1] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, “Registration with the point cloud library: A modular framework for aligning in 3-D,” IEEE Robot. Autom. Mag., vol. 22, no. 4, pp. 110–124, Dec. 2015.
[2] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: Theory and experiment,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, no. 9, pp. 920–932, Sep. 1994.
[3] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo matching using belief propagation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 7, pp. 787–800, Jul. 2003.
[4] I.-L. Jung, Y.-T. Chung, J.-Y. Sim, and C.-S. Kim, “Consistent stereo matching under varying radiometric conditions,” IEEE Trans. Multimedia, vol. 15, no. 1, pp. 56–69, Jan. 2013.
[5] J. Xie, R. Girshick, and A. Farhadi, “Deeplab: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks,” in Proc. Eur. Conf. Comput. Vis., Amsterdam, The Netherlands, Oct. 2016, pp. 842–857.
[6] C. Liu, J. Yang, D. Ceylan, E. Yamashita, and Y. Furukawa, “PlaneNet: Piecewise planar reconstruction from a single RGB image,” in Proc. CVPR, Salt Lake City, UT, USA, Jun. 2018, pp. 2579–2588.
[7] 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., Honolulu, HI, USA, Jul. 2017, pp. 270–279.
[8] H. Kim, S. Lee, Y. Kim, S. Lee, D. Lee, J. Ju, and H. Myung, “Weighted joint-based human behavior recognition algorithm using only depth information for low-cost intelligent video-surveillance system,” Expert Syst. Appl., vol. 45, pp. 131–141, Mar. 2016.
[9] H. Izadinia, Q. Shan, and S. M. Seitz, “IM2CAD,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 5134–5143.
[10] Y. Ming, X. Meng, C. Fan, and H. Yu, “Deep learning for monocular depth estimation: A review,” Neurocomputing, vol. 438, pp. 14–33, May 2021.
[11] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in Proc. Adv. Neural Inf. Process. Syst., vol. 18, Vancouver, BC, Canada, Dec. 2005, pp. 1–8.
[12] M. Liu, M. Salzmann, and X. He, “Discrete-continuous depth estimation from a single image,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, Jun. 2014, pp. 716–723.
[13] 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., vol. 27, Montreal, QC, Canada, Dec. 2014, pp. 2366–2374.
[14] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deepper depth prediction with fully convolutional residual networks,” in Proc. 4th Int. Conf. 3D Vis. (3DV), Stanford, CA, USA, Oct. 2016, pp. 239–248.
[15] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Glasgow, U.K., Jun. 2018, pp. 2002–2011.
[16] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, “Single-image depth estimation based on Fourier domain analysis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 330–339.
[17] M. Heo, J. Lee, K.-R. Kim, H.-U. Kim, and C.-S. Kim, “Monocular depth estimation using whole strip masking and reliability-based refinement,” in Proc. Eur. Conf. Comput. Vis., Munich, Germany, Sep. 2018, pp. 36–51.
[18] J.-H. Lee and C.-S. Kim, “Monocular depth estimation using relative depth maps,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA, Jun. 2019, pp. 9729–9738.
[19] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, “ReviseSing single image depth estimation: Toward higher resolution maps with accurate object boundaries,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Waikoloa Village, HI, USA, Jan. 2019, pp. 1043–1051.
[20] X. Chen, X. Chen, and Z.-J. Zha, “Structure-aware residual pyramid network for monocular depth estimation,” in Proc. Int. Joint Conf. Artif. Intell., Macao, Asia, Aug. 2019, pp. 694–700.
[21] 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.
[22] S. Tang, F. Tan, K. Cheng, Z. Li, S. Zhu, and P. Tan, “A neural network for detailed human depth estimation from a single image,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, South Korea, Oct. 2019, pp. 7750–7759.
[23] J. Lin and G. H. Lee, “HDNet: Human depth estimation for multi-person camera-space localization,” in Proc. Eur. Conf. Comput. Vis., Glasgow, U.K., Aug. 2020, pp. 633–648.
[24] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. Eur. Conf. Comput. Vis., Florence, Italy, Oct. 2012, pp. 746–760.
[25] 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, 2013.
[26] W. Chen, Z. Fu, D. Yang, and J. Deng, “Single-image depth perception in the wild,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, San Diego, CA, USA, Dec. 2016, pp. 730–738.
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.*, Honolulu, HI, USA, Jul. 2017, pp. 5354–5362.

R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2021, pp. 12179–12188.

D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proc. IEEE ICCV*, Santiago, Chile, Dec. 2015, pp. 2650–2658.

C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Seoul, South Korea, Oct. 2019, pp. 3828–3838.

W. Yin, Y. Liu, C. Shen, and Y. Yan, “Enforcing geometric constraints of virtual normal for depth prediction,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 5684–5693.

X. Ye, S. Chen, and R. Xu, “DPNet: Detail-preserving network for high quality monocular depth estimation,” *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107578.

J.-H. Lee and C.-S. Kim, “Multi-loss rebalancing algorithm for monocular depth estimation,” in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K., Aug. 2020, pp. 785–801.

D. Xu, W. Ouyang, X. Wang, and N. Sebe, “PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 675–684.

Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, “Pattern-affinitive propagation across depth, surface normal and semantic segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4106–4115.

M. Klingner, I.-A. Termohlen, J. Mikolajczyk, and T. Fingscheidt, “Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance,” in *Proc. ECCV*, Glasgow, U.K., Aug. 2020, pp. 582–600.

A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 483–499.

S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 4724–4732.

Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, “Cascaded pyramid network for multi-person pose estimation,” in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 7103–7112.

K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 5693–5703.

J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A simple yet effective baseline for 3D human pose estimation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Jun. 2017, pp. 2640–2649.

L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, “Semantic graph convolutional networks for 3D human pose regression,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 3425–3435.

H. Wu and B. Xiao, “3D human pose estimation via explicit compositional depth maps,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, New York, NY, USA, Feb. 2020, pp. 12378–12385.

G. Pavlakos, Z. Zhou, K. G. Derpanis, and K. Daniilidis, “Coarse-to-Fine volumetric prediction for single-image 3D human pose,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7025–7034.

A. Nibali, Z. He, S. Morgan, and L. Prendergast, “3D human pose estimation with 2D marginal heatmaps,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.* (WACV), Waikoloa Village, HI, USA, Jan. 2019, pp. 1477–1485.

C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

L. Sigal, A. O. Balan, and M. J. Black, “HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion,” *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, pp. 4–27, 2010.
JAE-HAN LEE received the B.S. degree from the School of Electrical Engineering and the B.S. degree from the School of Financial Engineering, Korea University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include monocular depth estimation and multi-task learning.

CHUL LEE (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2003, 2008, and 2013, respectively. He was with Biospace Inc., Seoul, from 2002 to 2006, where he was involved in the development of medical equipment. From 2013 to 2014, he was a Postdoctoral Scholar with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA, USA. From 2014 to 2015, he was a Research Scientist with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong. From 2015 to 2019, he was an Assistant Professor with the Department of Computer Engineering, Pukyong National University, Busan, South Korea. In March 2019, he joined the Department of Multimedia Engineering, Dongguk University, Seoul, where he is currently an Assistant Professor. His current research interests include image processing and computational imaging with an emphasis on restoration and high dynamic range imaging.

CHANG-SU KIM (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Seoul National University with the Distinguished Dissertation Award, in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. From 2001 to 2003, he coordinated the National Research Laboratory for 3D Visual Information Processing, 3D Data Compression Group, SNU. From 2003 to 2005, he was an Assistant Professor with the Department of Information Engineering, The Chinese University of Hong Kong. In September 2005, he joined the School of Electrical Engineering, Korea University, where he is currently a Professor. He has published more than 300 journal articles and conference papers. His research interests include image processing, computer vision, and machine learning. He is a member of the Multimedia Systems and Application Technical Committee (MSATC) of the IEEE Circuits and Systems Society. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award from Journal of Visual Communication and Image Representation (JVCI). He served as an Editorial Board Member for JVCI, and an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON MULTIMEDIA. He is a Senior Area Editor of JVCI. He was also an APSIPA Distinguished Lecturer, for the term 2017–2018.

***