IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 6, 2021, accepted November 13, 2021, date of publication November 19, 2021,
date of current version December 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129628

Self-Supervised Monocular Depth Estimation With Extensive Pretraining

HYUKDOO CHOI
Department of Electronics and Information Engineering, Soonchunhyang University, Asan 31538, Republic of Korea
Department of Electronic Materials and Devices Engineering, Soonchunhyang University, Asan 31538, Republic of Korea

e-mail: hyukdoo.choi@sch.ac.kr

This work was supported in part by the Soonchunhyang University Research Fund under Grant 20180404, and in part by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT) of the Korea government under Grant 2018R1C1B5086360.

ABSTRACT Although depth estimation is a key technology for three-dimensional sensing applications involving motion, active sensors such as LiDAR and depth cameras tend to be expensive and bulky. Here, we explore the potential of monocular depth estimation (MDE) based on a self-supervised approach. MDE is a promising technology, but supervised learning suffers from a need for accurate ground-truth depth data. Recent studies have enabled self-supervised training on an MDE model with only monocular image sequences and image-reconstruction errors. We pretrained networks using multiple datasets, including monocular and stereo image sequences. The main challenges posed by the self-supervised MDE model were occlusions and dynamic objects. We proposed novel loss functions to handle these problems in the form of min-over-all and min-with-flow losses, both based on the per-pixel minimum reprojection error of Monodepth2 and extended to stereo images and optical flow. With extensive pretraining and novel losses, our model outperformed existing unsupervised approaches in quantitative depth estimation and the ability to distinguish small objects against a background, as evaluated by KITTI 2015.

INDEX TERMS Monocular depth estimation, depth prediction, convolutional neural networks, self-supervised learning, unsupervised learning.

I. INTRODUCTION

Three-dimensional (3D) vision involves inferring 3D geometric information from two-dimensional (2D) images. Monocular depth estimation (MDE) produces a dense depth map from a single image. A depth map provides valuable information for any vehicle in motion to perceive the current situation and make plans for the future. As depth information is important to moving vehicles, we used driving datasets such as KITTI, Cityscapes, Waymo and A2D2 [1]–[4].

Depth maps are typically obtained from complex sensors such as LiDAR, depth cameras, or stereo cameras. One promising alternative source, MDE, is often criticized for the lack of concrete depth clues in a single image. However, a number of researchers are developing MDE because depth-acquisition sensors are costly, bulky, and come with high power demands.

MDE has been developed in the three steps. The first step trains depths with hand-crafted features in a supervised way [5], [6]. Features are extracted by convolution with manually selected kernels. Only the weights to be multiplied are optimized to predict true depths during training. Their results are relatively inaccurate, with average relative depth error rates of greater than 30%.

In the second step, convolutional neural network (CNN) models are applied and trained with supervision [7]–[10]. With a CNN, numerous kernels are automatically adjusted for accurate depth prediction, and less pre- and post-processing and regularization are required. While these methods produce the most accurate results [7], [9], with a relative depth error rate below 10%, they require depth-labeled datasets. Capturing a ground-truth depth map also requires expensive sensors and complex calibration, which reduces the number of available trainable datasets.

To relax depth-label constraints, techniques have been developed to train depth-estimate models in a self-supervised manner [11]–[16]. The core of self-supervision is photometric loss, which pairs temporally adjacent or stereo images and synthesizes one image from the other using an estimated depth map and the relative pose between them. The difference

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

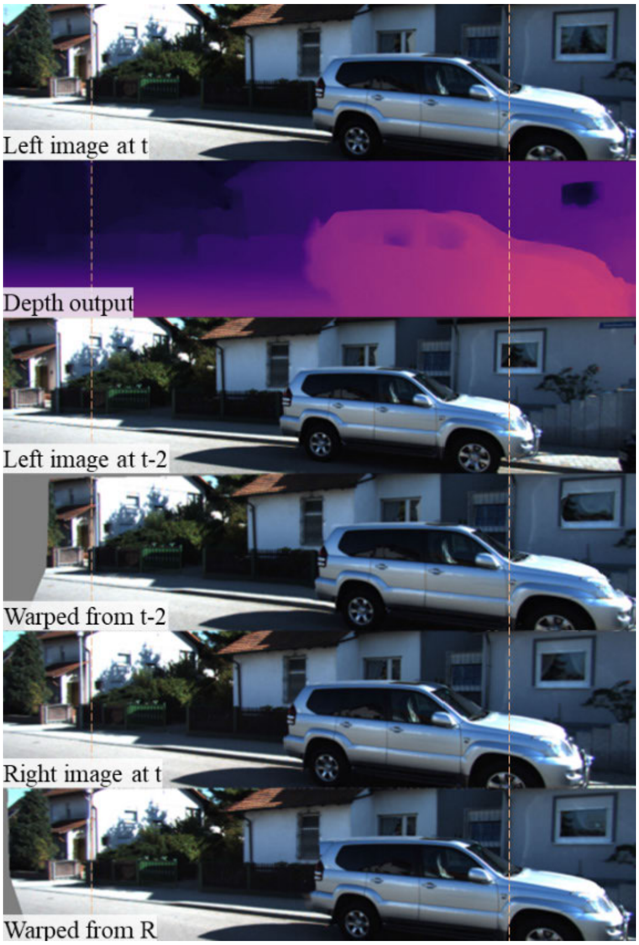

FIGURE 1. Depth estimation and image reconstruction of different views. From the top: the target image at t from the left camera and the estimated depth output; the left image at t-2; the target image reconstructed from the third image; the right image at t; and the target image reconstructed from the fifth image.

between the synthesized and original images represents the depth and pose estimation error, which is termed photometric loss. Reducing photometric loss does not theoretically guarantee accurate depth prediction, but the technique works empirically. Although large datasets are available for training, these methods only achieve accuracy levels comparable to those of supervised training methods, rather than exceeding such levels. However, self-supervision is worth pursuing because it is trained by low-cost datasets and has a greater potential for improvement.

Figure 1 provides examples of depth estimate and image reconstruction. The first and the second images are the target image and estimated depth map, respectively. The fourth image is warped from the third image using depth and pose estimation to reconstruct a target image. The last image is warped from the fifth using the estimated depth and known stereo extrinsic parameters. The warped images successfully reconstructed the target images, but the fourth image has more reconstruction error than the final image because it is affected by additional pose estimation error.

Here, we report progress in the effort to improve a self-supervised MDE model. Our contributions are fourfold. First, our model is trained by multiple large-scale datasets: KITTI, Waymo, Cityscapes, and A2D2 [1]–[4]. We find that depth accuracy can be enhanced by extensive pretraining with multiple datasets. Second, we present a novel concept, a *training plan*, to describe the training process. In our system, multiple models are trained collaboratively by multiple datasets and various losses following the training plan. The plan consists of multiple *periods*, with each period specifying the training model, loss combination, dataset, learning rate, and training epochs. Third, novel loss functions are proposed, called min-over-all (MoA) and min-with-flow (MiF) loss to handle both dynamic objects and occlusions. Lastly, we opened the source codes at https://github.com/goodgodgd/xpt-mde-2021. It is highly organized to easily replace models and losses and include useful utilities such as multiple dataset readers and depth-map generators from unordered point clouds.

The rest of this paper is organized as follows. Previous research is reviewed in detail in the next section. Section 3 presents the architecture of our model. Loss functions and the training plan are provided in Section 4, including the details of losses and the training process with multiple datasets. Experimental results and conclusions are presented in the final two sections.

II. RELATED WORKS

We reviewed previous MDE research from self-supervised methods skipping hand-crafted features [5], [6] and supervised methods [7]–[10]. Self-supervised approaches began with stereo datasets. The earliest models were proposed by Deep3D [12] and Garg *et al.* [11]. Deep3D predicts probabilistic distributions over discretized disparities for each pixel in a left-side image. A right-side image is then synthesized by combining copies of the left image shifted by discrete disparities of weights of the predicted probabilities. The model is trained using photometric errors between the synthesized image and the right-side image. Garg *et al.* [11] presented a similar model that predicts disparities directly and is trained by smoothness loss as well as photometric loss. MonoDepth [13] advanced this scheme by predicting disparity maps for both sides and synthesizing images for both sides from opposite-side images. Both left-to-right and right-to-left disparity maps are induced to be the same by incorporating left-right consistency loss. This significantly boosts the depth accuracy from Garg *et al.* [11].

While depth learning with stereo-image pairs can be effective, it is constrained to stereo datasets. SfmLearner [14] invented a new scheme that trains a Depth CNN and a Pose CNN simultaneously using a monocular temporal image sequence. In stereo training, a known baseline of a stereo camera is used to warp images, but SfmLearner replaces the baseline with a relative pose between two images predicted by a Pose CNN. Images are warped by both predicted depth and pose, and then trained to reduce temporal photometric loss. Subsequent studies developed this approach in various ways. Many continue to use monocular training [13], [14], [17]–[20], while others use both monocular and stereo learning [21]–[23].

Zhan *et al.* [23] adopted the idea that the same local image structure results in the same feature vector in a CNN's feature map. They reconstructed not only images but feature maps of another view and trained the models to reduce feature reconstruction loss. Mahjourian *et al.* [18] proposed using 3D geometric consistency between point clouds generated from predicted depth maps; if the predicted depths are correct, the point clouds should be perfectly aligned by iterative closest points (ICP). The residual errors from ICP can be exploited to train a depth estimation model. Recently, PDANet [24] claimed the most advanced depth accuracy by tackling color fluctuation problem but their loss functions do not counteract dynamic objects and their implementation is not open.

Departing from conventional CNN, new technologies are being introduced. Adversarial learning of Generative Adversarial Network (GAN) is applied to train MDE models [25], [26]. Depth and pose prediction models are used as a generator to synthesize a target image, and a discriminator distinguishes real images from synthesized images. As self-attention [27] has produced successful results in computer vision tasks [28]–[31], Johnston *et al.* [32] report applying self-attention to a depth prediction network to extract the global context of an image. In addition, discrete disparity volume can be incorporated into the network to estimate the probabilities for discrete disparities. However, GAN or self-attention techniques are known to take longer time to train than conventional CNN models.

Temporal photometric loss enables learning by monocular datasets, but it has trouble with occlusions and dynamic objects because it relies on the assumption of a static environment. Later studies focused on how to handle these issues [19]–[21]. GeoNet [19] treats 2D disparity as optical flow. In this method, rigid flow is first computed by a predicted depth map and pose to track the static background, and then ResFlowNet infers residual flow to track dynamic objects.

Monodepth2 [21] focuses on occlusions caused by parallax. Some pixels that cannot be reconstructed from another view, usually at geometrically discontinuous edges, can confuse training signals. To handle such invisible pixels, the per-pixel minimum reprojection loss is proposed. It takes only the minimum photometric loss per pixel over multiple reconstructions from different view images.

Ranjan *et al.* [20] proposed a new training scheme known as collaborative competition. In a collaborative stage, rigid networks (depth and camera motion) and an optical flow network are trained by static and dynamic regions, respectively, and the regions are divided by the motion segmentation network. In the competitive stage, the motion segmentation network is trained to select lower photometric loss per pixel between the rigid networks and the optical flow network. As the two training stages are repeated, the networks converge to reduce overall error.

In this paper, we propose new losses, MoA and MiF, to handle both occlusions and dynamic objects. The losses are inspired by the per-pixel minimum reprojection error of Monodepth2, but it deals only with occlusions, not dynamic objects. Like Ranjan *et al.* [20], we adopted an optical flow predicting network to track moving objects for the MiF loss but our method does not require motion segmentation network or complex training scheme.

Our approach is designed to minimize model complexity and instead rely on the power of data by expanding both quantity and diversity of data.

Some previous works [14], [19], [20] pretrained models by the Cityscapes dataset and then fine-tuned by the KITTI dataset. We extended this pretraining to extra datasets. Pretraining is a kind of transfer learning [33], [34]. The purpose of transfer learning is to transfer knowledge from the source to target domain to improve the performance of a target task. In this case, the pretraining datasets represents source domains, and the fine-tuning dataset is a target domain. By pretraining or transfer learning, knowledge from source domains is embedded into convolutional layers to create more generalized feature representation for depth estimation. Based on the knowledge, the model can yield improved performance in a target task.

III. SYSTEM ARCHITECTURE

Our system consists of the three CNN models, PoseNet, and FlowNet, and DepthNet. Our major focus is on DepthNet but the other two models are required to help train it. This section presents a system overview and the details of the network architectures.

A. SYSTEM OVERVIEW

Whenever a target image, T, is fed to DepthNet, it estimates a corresponding depth map, D, which has the same resolution as the target image. PoseNet takes both target and source images concatenated in the channel dimension and predicts a relative pose, M, between the target and source camera frames. PoseNet outputs a six-degrees-of-freedom, rigid-body motion, in the form of twist coordinates.

$$M = (\upsilon, \omega) \in se(3) \quad \text{where } \upsilon \in \mathbb{R}^3, \omega \in so(3) \qquad (1)$$

Source images are either temporally adjacent to a target image or stereo right images for which target images are left images. Like UnDeepVO [22], both temporal image sequences and stereo image pairs are exploited for training by photometric loss, which is defined in the next section. DepthNet and PoseNet are collectively termed RigidNet as they are used for rigid warping.

FlowNet takes both target and source images and estimates an optical flow map, F. The resulting map consists of two channels, du and dv, and has a quarter the resolution of the original image. These quarter-resolution images are then expanded to the original resolution using bilinear interpolation.

By combining a target depth map (D), and a relative pose (M), a source image (S) is warped into a target frame by Rigid Warping. The reconstructed view is termed a rigid reconstruction image, R^{Rigid} and can be generated by

$$R^{Rigid}\left(p_t\right) \leftarrow S\left(K\Lambda_{t\rightarrow s}D\left(p_t\right)K^{-1}p_t\right) \tag{2}$$

where p_t is the target image pixel coordinate, K is the camera projection matrix, and $\Lambda_{t\rightarrow s}$ is the transformation matrix corresponding to M. When source-image pixel coordinates corresponding to p_t are out of the image boundary during warping, the pixel values at p_t are set to zero. Those pixels are also omitted for loss computation like Monodepth2 [21].

A target image can also be synthesized from S using an optical flow map, F, regardless of dynamicity of the scene. The resulting image is called a flow reconstruction image, R^{Flow}. The difference between T and R^{Rigid} or between T and R^{Flow} is the photometric loss through which the models are trained.

B. NETWORK ARCHITECTURE

The network architectures of the DepthNet and PoseNet in Figure 2 are inherited from GeoNet [19] but with modifications. DepthNet follows DispNet [35] style, which has an encoder-decoder structure with skip connections. The encoder outputs feature maps at five scales, from 1/2 to 1/32 of the input resolution. Although GeoNet uses VGG or ResNet50 architectures for the encoder, we applied EfficientNet [36]. The decoder structure is the repetition of 2× upsampling, convolution, concatenation with the skip connection, and convolution. DepthNet outputs depth maps in multiple resolutions (1, 1/2, 1/4, and 1/8 of the input resolution) to learn geometric features in various scales.

Previous research typically used a feature-map resize operation before concatenation. As the input image size is not exactly a power of 2, an up-sampling 2× in the decoder does not always restore the original resolution. Decoder feature maps are supposed to have the same resolution as encoder feature maps at the same level for concatenation. Decoder feature maps are sometimes resized to fit the encoder feature map by increasing the resolution by one pixel. To remove this unnecessary operation and preserve spatial information, input image resolutions are restricted to integer multiples of 64 (128, 512).

PoseNet has a structure similar to that of GeoNet. It comprises strided convolution layers, reducing resolutions to extract high-level information for poses of six degrees of freedom. When training with higher-resolution images, a few strided convolutions are inserted to ensure additional resolution reduction and enlarge the receptive field of the final feature map. The last layer is drawn from the global average pooling to yield an output with a fixed dimension, $6N_s$, where N_s is the number of source images in the input. Each pose is represented by six dimensions, three for translation and three for rotation in the twist coordinates.

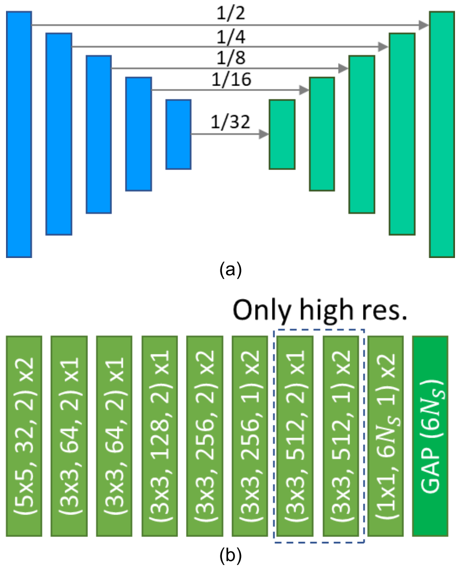

FIGURE 2. The network architectures. (a) DepthNet has the encoder-decoder structure with skip connections like DipsNet. (b) PoseNet is comprised of strided convolutions. The numbers in the parenthesis represents kernel size, output channels and a stride of convolution.

For FlowNet, we adopted PWCNet [37], following the CC [20] implementation. FlowNet also outputs flow maps at four different scales.

IV. TRAINING METHODOLOGY

While other studies focused on changing the network architecture, we were interested in the training process. Training with multiple datasets with a novel loss function constitutes our main contribution. The datasets used in training include both monocular and stereo datasets. Stereo datasets are more qualitatively effective, but monocular datasets are still quantitatively helpful. We proposed new training losses to handle both dynamic objects and occlusions. The entire training process follows a training plan that includes an order of datasets, learning rates, and loss functions. The next section supplies definitions of loss functions.

A. TRAINING LOSSES

1) BASIC LOSSES

The components of RigidNet are mainly trained by common photometric loss, which represents the difference between a target and a reconstructed image. The difference computed by either the L1 norm or SSIM and their combination are defined by the following equations.

$$\mathcal{L}_{L1}^{photo}\left(T,R\right) = T - R \tag{3}$$

$$\mathcal{L}_{SSIM}^{photo}\left(T,R\right) = \frac{1 - SSIM\left(T,R\right)}{2} \tag{4}$$

$$\mathcal{L}^{photo}\left(T,R\right) = \alpha\mathcal{L}_{L1}^{photo}\left(T,R\right) + \left(1-\alpha\right)\mathcal{L}_{SSIM}^{photo}\left(T,R\right) \tag{5}$$

where $\mathcal{L}$ represents a loss map that contains a loss value per pixel. The losses are averaged over all valid pixels. The SSIM refers to the structural similarity index [38], which measures the statistical similarity of two windowed images. The value is the mixture ratio of the two types of photometric losses. In previous reports, α was a small number such as 0.15 [19], but we found its value produced no significant differences in the experiments. Here, it is set at 0.5.

In our system, five consecutive images, called a *snippet*, are fed to PoseNet, where the center image is a target, and the others are sources. As the PoseNet generates four relative poses for the four source images, the target image is reconstructed four times. The temporal photometric losses from a snippet are summed. If a dataset provides stereo images, photometric losses are added in both left and right snippets and the photometric loss is computed from stereo image pairs. The following equations describe the temporal photometric losses from left and right snippets and the stereo photometric loss.

$$\mathcal{L}^{photo}_{temp,L} = \sum_{k=-2,k\neq 0}^{2} \mathcal{L}^{photo}\left(T_{L,t}, R^{Rigid}_{L,t+k\rightarrow t}\right) \tag{6}$$

$$\mathcal{L}^{photo}_{temp,R} = \sum_{k=-2,k\neq 0}^{2} \mathcal{L}^{photo}\left(T_{R,t}, R^{Rigid}_{R,t+k\rightarrow t}\right) \tag{7}$$

$$\mathcal{L}^{photo}_{stereo} = \mathcal{L}^{photo}\left(T_{L,t}, R^{Rigid}_{R\rightarrow L,t}\right) \tag{8}$$

where $T_{L,k}$ is the original left target image at time k, and $R^{Rigid}_{R\rightarrow L,t}$ and $R^{Rigid}_{L,t+k\rightarrow t}$ are the left target image reconstructed from the left source image at time t+k and from the right target image, respectively. $\mathcal{L}_{temp,R}$ and $\mathcal{L}_{stereo}$ are activated only when using stereo datasets.

DepthNet is also trained by the smoothness loss. It constrains geometric smoothness between adjacent pixels while allowing for abrupt changes on photometric edges. It is defined by

$$\mathcal{L}^{smooth} = \lambda_{dis} \nabla P e^{\lambda_{img} \nabla I} \tag{9}$$

where ∇P and ∇I are gradients of a disparity map and an input image, respectively, and λ_{dis} and λ_{img} are their respective weights. In previous research, the weights were set to 1 [13], [19], [23], [32], but we found that increasing both weights boosted depth prediction performances. Larger values for the weights drive DepthNet to make sharper depth discontinuities at photometric edges and smoother depths in low-textured regions.

Although there are no depth or pose annotations, the stereo calibration brings about the relative pose between two cameras. A known relative pose between two images can generate a supervision signal to PoseNet. In addition, PoseNet can learn metric scales from the known pose. To use the stereo calibration results, PoseNet predicts a relative pose from a stereo image pair. The loss is the mean squared error (MSE) between the predicted pose and the calibrated extrinsic pose. The loss is computed in both directions, left to right and right to left, and their sum becomes the stereo pose loss.

$$\mathcal{L}^{pose} = MSE\left(M^{pred}_{L\rightarrow R}, M^{gt}_{L\rightarrow R}\right) + MSE\left(M^{pred}_{R\rightarrow L}, M^{gt}_{R\rightarrow L}\right) \tag{10}$$

FlowNet also attempts to reconstruct the target image using optical flow and is trained by an independent photometric loss defined by

$$\mathcal{L}^{photo}_{flow} = \sum_{k=-2,k\neq 0}^{2} \mathcal{L}^{photo}\left(T_{L,t}, R^{Flow}_{L,t+k\rightarrow t}\right) \tag{11}$$

where $R^{Flow}_{L,t+k\rightarrow t}$ is the left target image reconstructed from the left source image at t+k.

Following PWCNet implementation, the FlowNet results are regularized by L2 regularization loss $\mathcal{L}^{L2reg}_{flow}$. As both the depth map and flow map are predicted at multiple scales, all losses mentioned above are summed over the scales.

2) ADVANCED LOSSES

Temporal photometric loss is a major loss function in all previous reports, but naïve temporal photometric loss is often corrupted by dynamic objects and occlusions. Here, we propose two loss functions to address this challenge. The first, MoA loss, is inspired by Monodepth2 [21], in which the per-pixel minimum reprojection loss takes the minimum over photometric losses per pixel from multiple source frames to handle occlusions. The MoA loss can handle dynamic objects by using stereo images. As stereo image pairs are captured synchronously, stereo photometric loss is not affected by dynamic objects. The MoA loss takes the minimum over not only temporal losses but stereo losses, formulated as follows.

$$\mathcal{L}^{photo}_{MoA,L}(p) = \min\left\{\min_{k\in\{-2,-1,1,2\}} \mathcal{L}^{photo}_{temp,L,k}(p), \mathcal{L}^{photo}_{stereo}(p)\right\} \tag{12}$$

where $\mathcal{L}^{photo}_{temp,L,k} = \mathcal{L}^{photo}\left(T_{L,t}, R^{Rigid}_{L,t+k\rightarrow t}\right)$ and $p = (u, v)$ as pixel coordinates. The loss is also computed from the right image snippet, resulting in $\mathcal{L}^{photo}_{MoA,R}$.

The second loss is MiF loss. As optical flow can track dynamic objects, the difference between temporal and flow photometric losses provides information about the location of dynamic parts. The MiF loss takes the temporal photometric loss only at the pixels for which the temporal photometric loss is lower than the flow photometric loss. It is defined as

$$\begin{aligned} \mathcal{L}^{photo}_{MiF,L} &= \sum_{k=-2,k\neq 0}^{2} M^{MiF}_{k} \otimes \mathcal{L}^{photo}\left(T_{L,t}, R^{Rigid}_{L,t+k\rightarrow t}\right) \\ M^{MiF}_{k} &= \left[\mathcal{L}^{photo}\left(T_{L,t}, R^{Rigid}_{L,t+k\rightarrow t}\right) < \mathcal{L}^{photo}\left(T_{L,t}, R^{Flow}_{L,t+k\rightarrow t}\right)\right] \end{aligned} \tag{13}$$

where [ ] is the Iverson bracket and $\otimes$ is the element-wise multiplication operator. Similarly, MiF loss is computed from the right snippet, $\mathcal{L}^{photo}_{MiF,R}$.

Both losses are designed to handle dynamic objects, but they have substantial differences. The MoA loss is computed from the outputs of RigidNet, while the MiF loss needs FlowNet as well as RigidNet. Training with the MiF loss therefore requires more computational power but the networks trained by the MiF loss outperform those using MoA loss.

B. LOSS COMBINATION

Various combinations of loss functions used in the training is termed as loss combination, denoted by $\mathcal{C}$. Four types of loss combinations are used in the training plan. The first is an initialization combination, used in the early training and defined as

$$\mathcal{C}_{init} = \mathcal{L}^{photo}_{temp,L} + \mathcal{L}^{photo}_{temp,R} + 0.01 \cdot \mathcal{L}^{photo}_{stereo} + \mathcal{L}^{smooth} + \mathcal{L}^{pose}. \tag{14}$$

In our experiments, the stereo photometric loss tends to drive depth predictions to diverge in early training. We set its weight to low value, 0.01, for stable training. The second one is the pretraining loss combination, which is used for multi-dataset pretraining. It increases the weights for both the stereo photometric loss and the smoothness loss.

$$\mathcal{C}_{pret} = \mathcal{L}^{photo}_{temp,L} + \mathcal{L}^{photo}_{temp,R} + \mathcal{L}^{photo}_{stereo} + 20 \cdot \mathcal{L}^{smooth} + \mathcal{L}^{pose} \tag{15}$$

In pretraining, the stereo photometric loss is omitted when using a monocular dataset. The third is the flow loss combination to independently train FlowNet.

$$\mathcal{C}_{flow} = \mathcal{L}^{photo}_{flow} + 4 \times 10^{-7} \cdot \mathcal{L}^{L2reg}_{flow} \tag{16}$$

RigidNet is fine-tuned by the latter two loss combinations. They replace temporal photometric losses with the MoA and MiF losses from the pretraining loss combination.

$$\mathcal{C}_{MoA} = \mathcal{L}^{photo}_{MoA,L} + \mathcal{L}^{photo}_{MoA,R} + \mathcal{L}^{photo}_{stereo} + 20 \cdot \mathcal{L}^{smooth} + \mathcal{L}^{pose} \tag{17}$$

$$\mathcal{C}_{MiF} = \mathcal{L}^{photo}_{MiF,L} + \mathcal{L}^{photo}_{MiF,R} + \mathcal{L}^{photo}_{stereo} + 20 \cdot \mathcal{L}^{smooth} + \mathcal{L}^{pose} \tag{18}$$

C. TRAINING PLAN

The training plan is a series comprising a dataset, a learning rate, the number of epochs, and a loss combination. The models are trained extensively with multiple datasets. The datasets are either monocular or stereo; KITTI, Cityscapes, and A2D2 are stereo while Waymo is monocular. With stereo datasets, the training is more robust to dynamic objects because stereo and MoA photometric losses are not affected by dynamic objects. Even if a stereo dataset demands more expensive devices and extrinsic calibration, it is too beneficial to abandon. Fortunately, several large-scale stereo or multi-camera datasets are available [1]–[4]. The datasets for self-driving research provide images from multiple front-mounted cameras and accurate extrinsic calibration results.

TABLE 1. The standard training plan.

Period	Dataset	Epochs	Learning rate	Loss
1	KITTI raw	5	1e-5	$\mathcal{C}_{init}$
2	KITTI raw	10	1e-4	$\mathcal{C}_{pret}$
3	A2D2	10	1e-4	$\mathcal{C}_{pret}$
4	Waymo	10	1e-4	$\mathcal{C}_{pret}$
5	KITTI odom	10	1e-4	$\mathcal{C}_{pret}$
6	Cityscapes	10	1e-4	$\mathcal{C}_{pret}$
7	KITTI raw	5	1e-4	$\mathcal{C}_{pret}$
8	KITTI raw	10	1e-4	$\mathcal{C}_{MiF}$
9	KITTI raw	10	1e-5	$\mathcal{C}_{MiF}$
10	KITTI raw	5	1e-6	$\mathcal{C}_{MiF}$

By comparison, a monocular dataset allows only single-sided temporal photometric loss, which is vulnerable to dynamic objects. The predicted depths on cars tend to be much larger than the truth when the model is being trained by the monocular dataset Waymo. A car driving in front of a camera usually keeps a constant distance from the camera. Because pixels on frontal cars look static, they are considered distant in the manner of clouds and mountains. However, if a dataset is quantitatively enough, it can help predict more accurate depth maps for static objects.

A standard training plan is presented in Table 1. The plan is a sequence of periods during which the training configurations are set as written in the corresponding row. In the first period, the models are warmed up with a low learning rate and an initialization loss combination in short epochs. The models are then pretrained extensively with five datasets up to the seventh period. The model matures by experiencing various datasets. In the last three periods, the models are fine-tuned for the evaluation dataset. For a comparison with previous research results, a KITTI raw dataset with the Eigen split was selected for evaluation. The fine-tuning loss combination can be replaced by $\mathcal{C}_{MoA}$ and the results are compared in Table 5.

The design principle of the training plan is as follows: initial training is focused on stability with a low learning rate. The stereo photometric loss in (8) is omitted in $\mathcal{C}_{init}$ for stable initialization. In the pretraining periods, training should be stopped before the loss is saturated, in order not to forget all the knowledge from the previous datasets. During the fine-tuning periods, the most advanced loss functions are used with decaying learning rate until the loss converges.

In the standard training plan, only RigidNet is trained and not FlowNet. As FlowNet is required by $\mathcal{C}_{MiF}$, it should be independently trained in advance. FlowNet is trained by another plan, the flownet training plan in Table 2. After learning by the flownet training plan, FlowNet is fixed during the standard training plan.

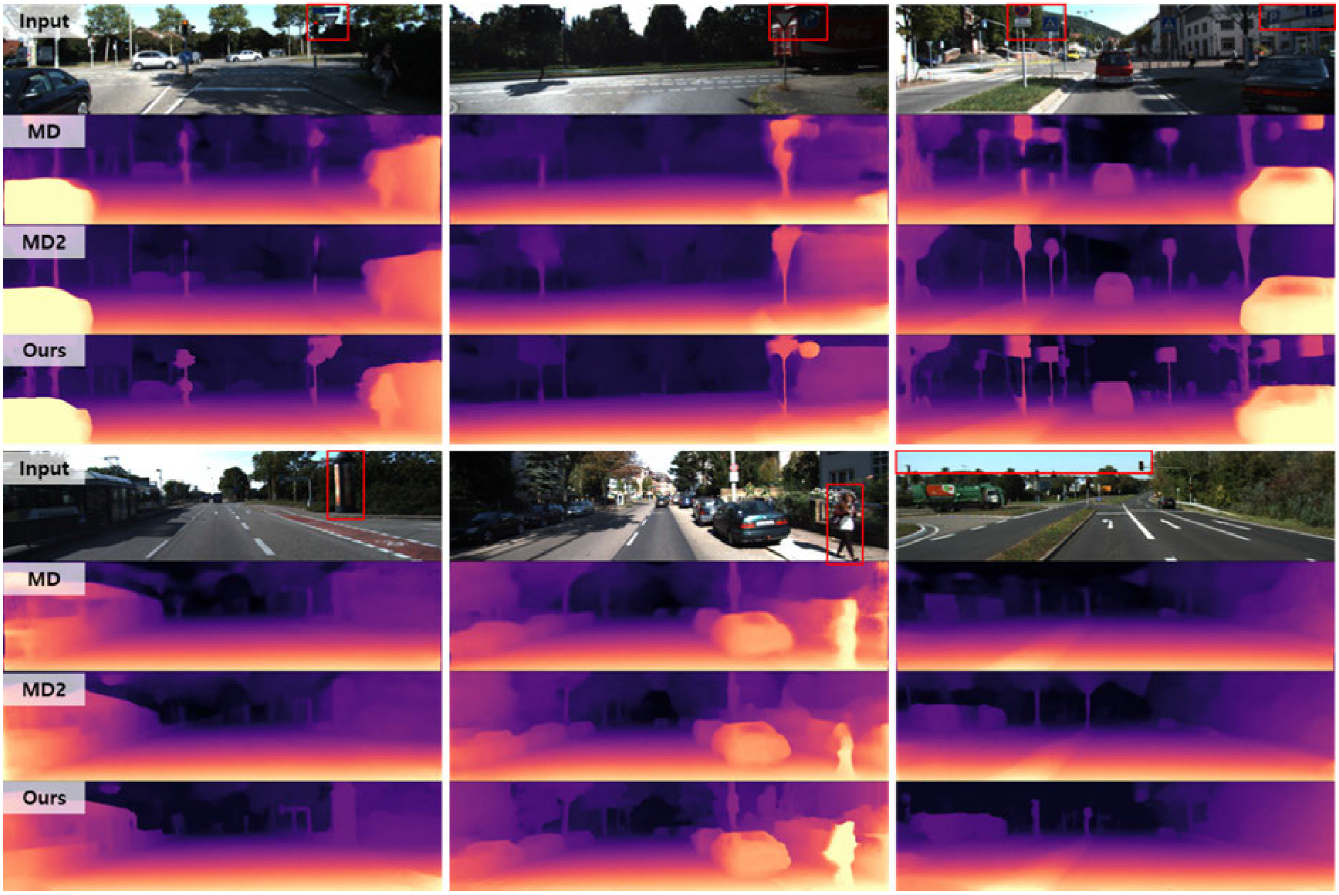

FIGURE 3. Qualitative comparison on the test set of the KITTI Eigen split. Each image is composed of an input image, and results of Monodepth [13] and Monodepth2 [21], and ours. Our depth maps do not miss small objects and show their shape more correctly. The regions to be noted are enclosed by the red boxes in the input images.

TABLE 2. The Flownet training plan.

Period	Dataset	Epochs	Learning rate	Loss
1	KITTI raw	5	1e-5	$\mathcal{C}_{flow}$
2	KITTI raw	10	1e-4	$\mathcal{C}_{flow}$
3	A2D2	7	1e-4	$\mathcal{C}_{flow}$
4	Waymo	7	1e-4	$\mathcal{C}_{flow}$
5	KITTI odom	10	1e-4	$\mathcal{C}_{flow}$
6	Cityscapes	6	1e-4	$\mathcal{C}_{flow}$
7	KITTI raw	5	1e-4	$\mathcal{C}_{flow}$
8	KITTI raw	5	1e-5	$\mathcal{C}_{flow}$
9	KITTI raw	5	1e-6	$\mathcal{C}_{flow}$

V. EXPERIMENTAL RESULTS

A. DATASETS

The models were trained by multiple calibrated datasets from different sensors in different formats. Training required only monocular or binocular image sequences. The properties of the datasets are summarized in Table 3, including the number of frames used for training, the original image resolution, the low and high input resolutions, and the camera type.

TABLE 3. The properties of the training datasets.

Dataset	# Frames	Orig. res.	Low res.	High res.	Camera
KITTI raw	21,827	375, 1242*	128, 512	256, 1024	stereo
KITTI odom	39,062	370, 1226*	128, 512	256, 1024	stereo
A2D2	42,704	1208, 1920	192, 384	384, 768	stereo
Waymo	76,263	1080, 1920	256, 384	512, 768	mono
Cityscapes	115,327	1024, 2048	192, 512	384, 1024	stereo

The resolution of the KITTI datasets varies little by sequences.

The KITTI vision benchmark suite [1], which is popularly used for computer vision and autonomous driving research, provides a pack of datasets containing stereo images and LiDAR data for various purposes. We chose two datasets: "raw" and "odometry," as they were used in most previous studies. Following the convention of the previous reports [7], [14], [19], the KITTI raw dataset was used for the main evaluation, while the other datasets were used for pretraining.

Cityscapes [2], A2D2 [4], and Waymo [3] are also popular datasets for self-driving studies, but they are much larger than KITTI in both diversity and quantity. The first two have multiple frontal cameras while Waymo has just one. Cityscapes provides various kinds of datasets; we used the sequential image dataset to generate snippet images. As Waymo offers

TABLE 4. Depth estimation performance metrics.

Method	Train	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen [7]	D	0.203	1.548	6.307	0.282	0.702	0.890	0.890
Liu [8]	D	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Klodt [39]	D*M	0.166	1.490	5.998	-	0.778	0.919	0.966
AdaDepth [40]	D*	0.167	1.257	5.578	0.237	0.771	0.922	0.971
Kuznietsov [41]	DS	0.113	0.741	4.621	0.189	0.862	0.960	0.986
DVSO [42]	D*S	0.097	0.734	4.442	0.187	0.888	0.958	0.980
SVSM FT [43]	DS	0.094	0.626	4.252	0.177	0.891	0.965	0.984
Guo [10]	DS	0.096	0.641	4.095	0.168	0.892	0.967	0.986
DORN [9]	D	**0.072**	**0.307**	**2.727**	**0.120**	**0.932**	**0.984**	**0.994**
Zhou [14]†	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang [17]	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian [18]	M	0.151	0.949	**4.383**	0.227	0.802	0.935	0.974
GeoNet [19]†	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DDVO [15]	M	0.148	1.187	5.496	0.226	0.812	0.938	0.975
DF-Net [44]	M	0.146	1.182	5.215	0.213	0.818	0.943	0.978
LEGO [45]	M	0.159	1.345	6.254	0.247	-	-	-
Ranjan [20]	M	0.139	1.032	5.199	0.213	0.827	0.943	0.977
EPC++ [46]	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth [47]	M	0.109	0.825	4.750	0.187	0.874	0.958	**0.983**
Monodepth2 [21]	M	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Johnston [32]	M	**0.106**	**0.861**	4.699	**0.185**	**0.889**	**0.962**	0.982
Garg [11]†	S	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Monodepth R50 [13]	S	0.108	**0.657**	**3.729**	**0.194**	0.873	**0.954**	0.979
StrAT [25]	S	0.128	1.019	5.403	0.227	0.827	0.935	0.971
3Net [48]	S	0.111	0.849	4.822	0.202	0.865	0.952	**0.978**
SuperDepth (pp) [49]	S	0.112	0.875	4.958	0.207	0.852	0.947	0.977
Monodepth2 [21]	S	**0.107**	0.849	4.764	0.201	**0.874**	0.953	0.977
UnDeepVO [22]	MS	0.183	1.730	6.57	0.268	-	-	-
Zhan FullNYU [23]	D*MS	0.128	0.815	4.204	0.216	0.835	0.941	0.975
EPC++ [46]	MS	0.127	0.936	5.008	0.209	0.841	0.946	0.979
Monodepth2 [21]	MS	0.106	0.806	4.630	0.193	0.876	0.958	0.980
Ours (128 × 512)	MS	0.105	0.644	4.080	**0.177**	0.887	**0.963**	**0.984**
Ours (256 × 1024)	MS	**0.099**	**0.629**	**3.919**	**0.177**	**0.903**	**0.963**	0.981

Our method is compared with the existing methods on the KITTI raw dataset using the Eigen split. The results are categorized by training methods and data types. The best results are typed in bold for each category. The meaning of the letters in the Train column: D - Depth supervision, D* - Auxiliary depth supervision, S - Self-supervised by stereo images, M – Self-supervised by monocular images. In the Method column, '†' represents newer results available on github. Metrics indicated by orange better be low, and blue metrics better be high.

environmental information, rainy or night frames can be filtered out. We implemented a simple algorithm to skip static snippets and applied it to all the datasets except for KITTI raw.

Images were resized and cropped to be model inputs to meet two conditions. First, static parts of images such as a car bonnet or the sky were cropped to avoid disturbing the training. Second, the input resolutions were restricted to multiples of either 64 or 128 for low and high resolutions, respectively. This removed redundant resize layers in DepthNet. Although input resolutions varied by dataset, the fully convolutional model was able to handle the diversity.

B. EVALUATION RESULTS

Depth prediction performance was evaluated mainly by the Eigen split [7] of the KITTI dataset, as was the case in most previous reports. Many of those studies involved training using monocular images; depth predictions were therefore evaluated at relative scales. DepthNet trained with stereo images may be able to predict depths at a metric scale, but depth evaluation at a metric scale includes overall scale error as well as relative depth error. Our model was evaluated at a relative scale to permit fair comparisons with other studies. In relative-scale evaluations, a predicted depth map is scaled such that the medium value of the predicted depth map is equal to that of the true depth map.

The performance of our model is compared with others in Table 4 where the KITTI raw dataset and the Eigen split is used for evaluation. The table compares various depth prediction networks categorized by training methods and dataset types. The best results for each category are in bold. The table refers to Johnston [31] but the numbers are updated. We collected the best results from the available papers or GitHub resources regardless of pretraining, depth clipping (usually at 50 m or 80 m), and post-processing, whereas Johnston [31] and Monodepth2 [21] compared only pure DepthNet outputs. Evaluation results at both low and high resolutions are presented in Table 4. The low-resolution results are the more accurate than the existing self-supervised methods and the high-resolution results are even better.

The predicted depth maps are compared in Figure 3. Our high-resolution results are compared with prediction produced by Monodepth [13] and Monodepth2 [21], for which depth predictions are available in their GitHubs. Monodepth2

TABLE 5. Depth estimation performances with different loss combinations.

Loss Comb.	Abs Rel	Sq Rel	RMSE	RMSE log
$\mathcal{C}_{pret}$	0.1065	0.6619	4.1332	0.1797
$\mathcal{C}_{MoA}$	0.1058	0.6527	4.1191	**0.1771**
$\mathcal{C}_{MiF}$	**0.1048**	**0.6439**	**4.0798**	0.1772

TABLE 6. Depth estimation performances with different training plans and input resolutions.

Dataset	Res.	Abs Rel	Sq Rel	RMSE	RMSE log
KITTI only	Low	0.1230	0.7910	4.5807	0.1988
Stereo only	Low	0.1059	0.6907	4.1792	0.1789
Std. plan	Low	**0.1048**	**0.6439**	**4.0798**	**0.1772**
KITTI only	High	0.1124	0.7594	4.3334	0.1908
Std. plan	High	**0.0988**	**0.6290**	**3.9186**	**0.1773**

'KITTI only' means that the model is trained by only the KITTI raw dataset. 'Stereo only' represents the model trained without any monocular dataset. 'Std. plan' is the standard training plan. 'Low' and 'High' indicate input resolutions: (128, 512) and (256, 1024) respectively.

TABLE 7. Depth estimation performances with different backbone networks.

Backbone	# params	Abs Rel	Sq Rel	RMSE	RMSE log
NasNet Mobile	8.0M	0.1352	1.0784	5.0422	0.2074
MobileNetV2	8.0M	0.1354	0.9569	4.9842	0.2103
Efficient B0	10.1M	0.1351	0.9375	4.9545	0.2096
Efficient B3	17.8M	0.1238	0.8351	4.6384	0.1970
Efficient B5	37.3M	0.1230	0.7910	4.5807	0.1988
Efficient B7	74.7M	**0.1185**	**0.7663**	**4.4799**	**0.1938**

presented the clean and sharp edges but missed some small objects or showed only parts of them. Our results appear to be more blurred on the large edges but do not miss any small objects, such as traffic signs or humans. The objects that show differences are highlighted by the red boxes in the input images. Our model is more capable of distinguishing small objects against background detail, which is a crucial task for self-driving vehicles, which must be able to identify traffic signs and signals, humans, and pets to ensure safety and compliance with the law. The last example shows that our model predicted a distant sky while Monodepth2 predicted a closer sky. Although the upper regions in depth maps are not evaluated, they can be critical to real-world applications.

C. ABLATION STUDY

To verify our proposals in Section 3 and 4, we performed an ablation study by changing loss, training plan, and backbone networks. The effect of different loss combinations is presented in Table 5, which compares the three loss combinations. The final row supplies the results by the standard training plan in Table 1. The first two results are yielded by replacing $\mathcal{C}_{MiF}$ with $\mathcal{C}_{pret}$ and $\mathcal{C}_{MoA}$, respectively, in the training plan. While $\mathcal{C}_{pret}$ is based on the conventional photometric loss, $\mathcal{C}_{MiF}$ and $\mathcal{C}_{MoA}$ include the MoA and MiF losses, respectively, which are proposed in this paper. The overall performance is best with $\mathcal{C}_{MiF}$, but consumes more computation power and memory because it needs FlowNet. With help of FlowNet, which handles dynamic objects, DepthNet can be trained using only static regions. However, $\mathcal{C}_{pret}$ and $\mathcal{C}_{MoA}$ require only RigidNet and therefore consume less computation power and memory during training. While $\mathcal{C}_{pret}$ has difficulty dealing with dynamic objects, $\mathcal{C}_{MoA}$ can handle them adequately, producing superior results compared with $\mathcal{C}_{pret}$.

Our main contribution is extensive pretraining using multiple datasets. To see the effect of pretraining, the models are trained with different levels of pretraining, and the results are summarized in Table 6. "KITTI only" means that the model is trained by only the KITTI raw dataset. "Stereo only" represents the model trained without any monocular dataset. "Std. plan" is the standard training plan in Table 3. "Low" and "High" indicate input resolutions: (128 × 512) and (256 × 1024), respectively. The standard plan using all the available datasets produced the best results in both input resolutions. In addition, "Std. plan" is superior to "stereo only," which indicates that a single monocular dataset among multiple stereo datasets can have a positive effect on the results.

The results of the tests of various backbone networks for DepthNet are presented in Table 7. The models are trained with only the KITTI raw dataset. The second column presents the number of trainable parameters of DepthNet models. The different scale levels of EfficientNet [36] models are used primarily to reveal the relationship between model complexity and performance. The popular ResNet and VGG families were not trained well but compact networks such as MobileNetV2 [50] and NasNet mobile [51] worked. The training followed the KITTI-only plan in Table 6. As expected, the performance improves with larger models. The Efficient B5 model is used for our main results but larger models have the potential to produce superior results with full pretraining.

VI. CONCLUSION

We have presented a novel training loss and plan for a monocular depth estimation network with assistance from pose and optical flow estimation networks. The network architectures achieved results that differ only marginally from those of existing models [19], [21], but effective loss functions and extensive pretraining enhanced depth estimation performance. Two novel losses are proposed to handle occlusions and dynamic objects. The MoA loss can achieve state-of-the-art performance in low-specification GPUs, while the MiF loss needs higher-specification GPUs but leads to beyond state-of-the-art performance. Extensive pretraining is a key contribution to improving depth estimation accuracy. With all differences of resolutions, aspect ratios, focal lengths, environments, and data types (mono or stereo), datasets other than KITTI raw helped training. While previous studies employed pretraining using only Cityscapes [2] we applied expanded datasets and checked the positive effect. Qualitatively, our model can separate small objects from the background but occasionally suffers from horizontal blurring in depth estimation.

In the future, we first plan to further improve depth estimation to make sharper edges. We trained three networks but focused only on depth estimation. Further efforts should therefore be made to achieve state-of-the-art performance in both visual odometry and optical flow estimation by changing loss functions and applying extensive pretraining. Moreover, pretraining based on the heuristically designed training plan can be advanced by modern domain adaptation techniques [52]. It will help transfer knowledge from the source domains to the target domain more effectively by learning domain-invariant feature representation. On the other hand, the multi-dataset training plan can be used for lifelong learning. The current implementation does not care about the catastrophic forgetting. However with lifelong learning techniques such as elastic weight consolidation [53], it does not have to forget and give up the performances from previous datasets. The model can possibly be trained to predict the correct depths for all datasets.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[3] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2446–2454.

[4] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi autonomous driving dataset," 2020, *arXiv:2004.06320*.

[5] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1161–1168.

[6] 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.

[7] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.

[8] 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.

[9] 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.

[10] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 484–500.

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

[12] J. Xie, R. Girshick, and A. Farhadi, "Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 842–857.

[13] 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.

[14] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[15] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2022–2030.

[16] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050.

[17] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry from videos with edge-aware depth-normal consistency," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7493–7500. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/issue/view/301

[18] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5667–5675.

[19] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[20] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12240–12249.

[21] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3828–3838.

[22] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7286–7291.

[23] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 340–349.

[24] H. Gao, X. Liu, M. Qu, and S. Huang, "PDANet: Self-supervised monocular depth estimation using perceptual and data augmentation consistency," *Appl. Sci.*, vol. 11, no. 12, p. 5383, Jun. 2021.

[25] I. Mehta, P. Sakurikar, and P. J. Narayanan, "Structured adversarial training for unsupervised monocular depth estimation," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 314–323.

[26] A. C. Kumar, S. M. Bhandarkar, and M. Prasad, "Monocular depth prediction using generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 300–308.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[28] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, "Attention augmented convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3286–3295.

[29] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 68–80. [Online]. Available: https://academic.microsoft.com/paper/2970389371/citedby/search?q=Stand-Alone%20Self-Attention%20in%20Vision%20Models&qe=RId%253D2970389371&f=&orderBy=0

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

[31] J. Beal, E. Kim, E. Tzeng, D. Huk Park, A. Zhai, and D. Kislyuk, "Toward transformer-based object detection," 2020, *arXiv:2012.09958*.

[32] A. Johnston and G. Carneiro, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4755–4764.

[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[34] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[35] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4040–4048.

[36] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, May 2019, pp. 6105–6114.

[37] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8934–8943.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[39] M. Klodt and A. Vedaldi, "Supervising the new with the old: Learning SFM from SFM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 698–713.

[40] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, "AdaDepth: Unsupervised content congruent adaptation for depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2656–2665.

[41] 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. 6647–6655.

[42] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 817–833.

[43] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, "Single view stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 155–163.

[44] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 36–53.

[45] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "LEGO: Learning edge with geometry all at once by watching videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 225–234.

[46] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3D holistic understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2624–2641, Oct. 2020.

[47] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 8001–8008.

[48] M. Poggi, F. Tosi, and S. Mattoccia, "Learning monocular depth estimation with unsupervised trinocular assumptions," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 324–333.

[49] S. Pillai, R. Ambrus, and A. Gaidon, "SuperDepth: Self-supervised, super-resolved monocular depth estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9250–9256.

[50] 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.

[51] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[52] W. Mei and D. Weihong, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Jul. 2018.

[53] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, and K. Milan, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

HYUKDOO CHOI received the B.S. and Ph.D. degrees in electrical and electronics engineering from Yonsei University, Seoul, South Korea, in 2009 and 2014, respectively.

From 2014 to 2017, he was a Senior Research Engineer at LG Electronics. Since 2018, he has been an Assistant Professor with the Department of Electronics and Information Engineering, Soonchunhyang University, Asan, South Korea. His research interests include simultaneous localization and mapping (SLAM), visual odometry, computer vision, object detection, machine learning, deep learning, and unsupervised learning.