ES^{3}Net: Accurate and Efficient Edge-based Self-Supervised Stereo Matching Network

I-Sheng Fang, Hsiao-Chieh Wen[†], Chia-Lun Hsu, Po-Chung Jen, Ping-Yang Chen, Yong-Sheng Chen
National Yang Ming Chiao Tung University, Hsinchu, Taiwan
{isfang.en09, wenxiaojie.cs08, chialun.cs10, jen1026.cs10, pingyang.cs08, yschen}@nycu.edu.tw

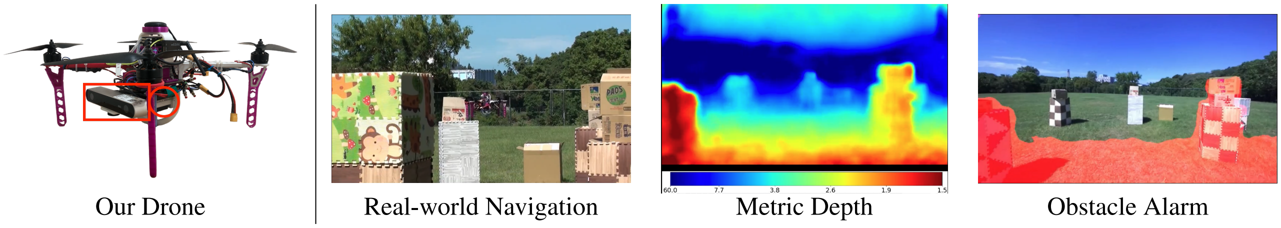

Figure 1. We deploy our efficient stereo depth estimation model, Edge-based Self-Supervised Stereo Matching Network (ES^{3}Net), to our self-made drone equipped with a ZED 2 stereo camera (red framed) and an NVIDIA Jetson TX2 (red circled) for drone obstacle avoidance and navigation [4]. The unit of depth map is meter.

Abstract

*Efficient and accurate depth estimation is crucial for real-world embedded vision applications, such as autonomous driving, 3D reconstruction, and drone navigation. Stereo matching is considered more accurate than monocular depth estimation due to the presence of a reference image, but its computational inefficiency poses a challenge for its deployment on edge devices. Moreover, it is difficult to acquire ground-truth depths for supervised training of stereo matching networks. To address these challenges, we propose **E**dge-based **S**elf-**S**upervised **S**tereo matching **N**etwork (ES^{3}Net), which efficiently estimates accurate depths without ground-truth depths for training. We introduce dual disparity to transform an efficient supervised stereo matching network into a self-supervised learning framework. Comprehensive experimental results demonstrate that ES^{3}Net has comparable accuracy with stereo methods while outperforming monocular methods in inference time, approaching state-of-the-art performance. More specifically, our method improves over 40% in terms of RMSE$_{\text{log}}$, compared to monocular methods while having 1500 times fewer parameters and running four times faster on NVIDIA Jetson TX2. The efficient and reliable estimation of depths on edge devices using ES^{3}Net lays a good foundation for safe drone navigation.*

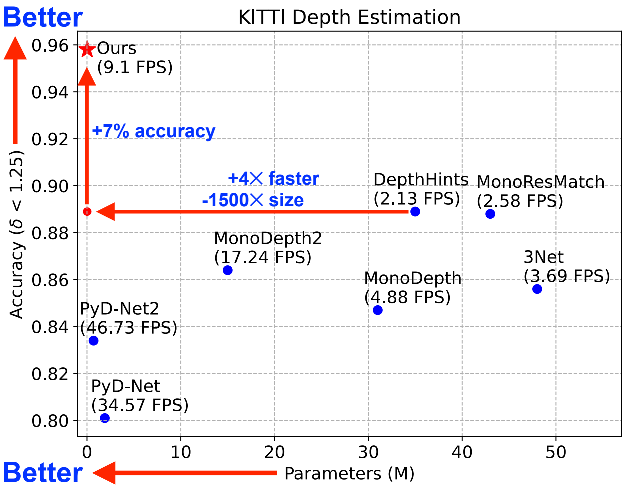

Figure 2. The comparison between model performance ($\delta < 1.25$) and complexity on KITTI 2012 [11]. Our ES^{3}Net outperforms SOTA self-supervised monocular methods [14, 15, 28–30, 33, 40], achieving significant improvements across multiple evaluation metrics while reducing model complexity by nearly 1500 times and running four times faster on NVIDIA Jetson TX2, comapred to DepthHints [40].

1. Introduction

Depth estimation receives growing attention owing to its numerous applications in the fields of autonomous driv-

[†] Hsiao-Chieh Wen is now with MediaTek Inc., Hsinchu, Taiwan.

ing [20, 27, 42], obstacle avoidance [10], 3D environment reconstruction [13, 44], and drone navigation [32], in which real-time responses are critical. However, edge computing devices such as mobile phones, VR headsets, and drones have limited hardware resources and battery life. To enable real-time operations on such devices, it is crucial to develop both accurate and efficient depth estimation models. As shown in Figure 1, our motivation is to estimate accurate depth on the edge device (NVIDIA Jetson TX2) for applications such as obstacle avoidance and drone navigation.

Accurate depth estimation is important for obstacle avoidance applications, allowing autonomous vehicles and drones to detect and avoid obstacles, reducing the risk of collisions and ensuring the safety of operations. The two mainstream approaches for depth estimation are monocular and stereo methods. Monocular depth estimation networks [14, 15, 28–30, 33, 40] are trained to estimate depths from a single image using the visual depth cues [19]. However, as these approaches rely on cues from a single image, they cannot reliably obtain metric depths. In contrast, stereo matching methods [5, 15, 21, 34, 38, 43, 47] estimate disparities between two cameras with calibrated camera parameters and obtain accurate and reliable results of metric depths. Nevertheless, these approaches suffer from high computational costs, which are not affordable for edge devices. Particularly, the typical approach for stereo depth estimation is the cost-volume method, which extracts features of stereo images through a convolutional neural network (CNN) and constructs a cost volume for the computation of disparity, resulting in high computation requirements.

Another challenge when training a depth estimation model for real-world applications is to obtain ground-truth depths for supervised learning. In this context, self-supervised learning of depth estimation provides a compelling and flexible alternative to traditional supervised learning methods. It offers a solution to the limitation of supervised learning and provides a promising approach for depth estimation. Godard *et al.* [15] proposed self-supervised learning via geometry supervision of stereo image pairs instead of ground truth depths to learn how to estimate depths without strong supervision. The model must simultaneously estimate the left and right disparities with stereo pair to satisfy left-right consistency which geometrically supervises the model training.

To address the aforementioned challenges, we extend the lightweight architecture proposed in [2] and reduce the training cost with left-right consistency [15] by introducing dual disparity without modifying the cost volume. Our approach achieves high-quality results in quasi-real-time, as shown in Figure 2, without requiring ground-truth depths or incurring additional computational costs. In summary, contributions of this work include:

- We propose a new **E**dge-Based **S**elf-**S**upervised **S**tereo Matching Network (ES^{3}Net), which is trained in self-supervised manner and can provide an effective solution for rapid and precise estimation of depths from stereo images. To the best of our knowledge, our proposed method is the first to achieve the high-speed self-supervised stereo depth estimation for edge computing. In addition to computational efficiency, our method is advantageous as it requires the fewest model parameters.
- Compared to the state-of-the-art (SOTA) self-supervised monocular method, our method significantly improves performance according to various evaluation metrics, including absolute relative error (AbsRel), squared relative error (SqRel), root mean squared error (RMSE), and root mean squared error of the logarithm (RMSE$_{\log}$), by 46.39%, 58.67%, 48.36%, and 41.94%, respectively. In addition, our method provides a reduction of model complexity by almost 1500 folds while running over four times faster on the NVIDIA Jetson TX2.
- The comprehensive experiments on multiple tasks have demonstrated that our method performs equally or better than its task-specific counterparts using the existing self-supervised stereo estimation methods, without leveraging optical flow.

2. Related Works

2.1. Stereo Matching in Depth Estimation

Cost-volume-based models have been the mainstream approach for stereo matching in depth estimation in recent years. Chang *et al.* [3] construct a 4D cost volume by concatenating left and right-view feature maps as well as 3D hourglass networks to estimate disparity. In wake of the high computational cost of PSMNet [3], Chang *et al.* [2] propose a real-time architecture that comprises an efficient backbone and attention-aware feature aggregation. Xu *et al.* [41] propose sparse points based representation for intra-scale cost aggregation to improve efficiency of self-supervised stereo matching. Wang *et al.* [37] take the L_1 distance between the left and right-view feature maps and propounded a cascaded 3D CNN architecture for undertaking disparity estimation. In this work, we employ diverse techniques to construct the cost volume and evaluate multiple architectures in order to demonstrate the robustness of our proposed model trained by self-supervised learning.

2.2. Self-supervised Depth Estimation

In recent years, self-supervised depth estimation has been extensively studied given the advantage of not requiring large ground-truth data as supervision. Godard *et* *al.* [15] treat depth estimation as a form of image reconstruction. Designed to estimate left-to-right as well as right-to-left disparities, their network utilizes the left-right disparity consistency loss to enforce consistency between both estimations. Zhong *et al.* [48] integrate self-supervised learning by image reconstruction with cost-volume stereo matching and RNN-based network. Lai *et al.* [21] bridge stereo and optical flow estimation with a two-stream architecture using spatiotemporal correspondences. Wang *et al.* [38] unify optical flow and stereo estimation in a self-supervised two-branch network. Chi *et al.* [6] use a hybrid loss function with shared encoders and separate decoders to enhance joint stereo and depth estimation. While existing methods depend on joint networks for stereo and optical flow estimation, or collaborative stereo and depth features, our proposed method eliminates the need for any additional network for joint estimation, reducing computational cost and simplifying the architecture.

2.3. Self-Supervised Lightweight Depth Estimation

As the usage of mobile and edge devices continues to rise, there is a growing demand for lightweight models. In previous related works, various monocular methods have been demonstrated to be effective to achieve self-supervised learning. Watson *et al.* [40] and Tosi *et al.* [33] obtain proxy labels with semi-global matching (SGM) [18] and then use proxy labels as alternative ground truths. Poggi *et al.* [28] and Poggi *et al.* [29] focus on efficient feature extractor and stacked depth estimators with pyramidal structure. Overall, these methods have demonstrated promising results in self-supervised monocular depth estimation. Although state-of-the-art methods have improved depth estimation accuracy, they still face limitations because the obtained depths are only relative. To the best of our knowledge, our proposed method is the first to deploy a self-supervised stereo matching network to edge computing.

2.4. Efficiency of Disparity Computation

To train a depth estimator using geometry supervision, the model must estimate the left and right disparity from a stereo pair in a manner that satisfies left-right consistency. The CBMV [1] constructs the cost volume with a bi-directional search to estimate the disparity at the expense of memory penalty by doubling the size of the cost volume. Rahnama *et al.* [31] propose a two-stage depth estimation architecture by combining R^{3}SGM [26] and ELAS [12] methods. The first stage entails estimating the disparity with R^{3}SGM [26]; then, the right disparity is flipped and left-right consistency is performed in order to derive the left view disparity. The second stage involves removing the outliers from the disparity with ELAS [12]. However, their depth estimation is premised on traditional methods, which cannot be considered a deep learning approach. The proposed dual transform overcomes this issue with minimal additional cost.

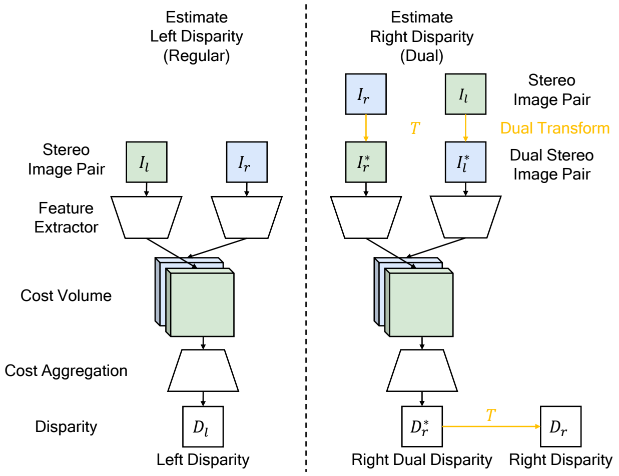

Figure 3. Illustration of the strategies for disparity estimation. Our approach with dual disparity is to estimate the right disparity with the same model for the purpose of estimating the left disparity.

3. Methods

We propose ES^{3}Net, a network architecture for stereo depth estimation in embedded edge computing. We discuss our choice of dual disparity transformation and combine three losses to improve self-supervised learning.

3.1. Dual Disparity

The right disparity D_r of stereo image pair is required in order to train our model with left-right consistency loss (Section 3.3). We introduce its *dual disparity* D_r^* to efficiently estimate the right disparity D_r with the same model f for the left disparity D_l. That is, the left disparity D_l is estimated by model f as follows:

$$D_l = f(I_l, I_r)\,, \tag{1}$$

where I_l and I_r denote the left and right images, respectively.

To estimate the right dual disparity D_r^*, we apply the dual transform T (such as horizontal flip and 180-degree rotation) that maps image pairs to the dual space, as shown in the right part of Figure 3. We define T as a self-inverse congruent transformation such that the right dual disparity D_r^* is computed from the right disparity $D_r^* = T(D_r)$ and vice versa ($D_r = T(D_r^*)$). Similarly, the dual counterpart for every image I is computed as $I^* = T(I)$. Therefore, the right dual disparity D_r^* is estimated by the same model f for the left disparity D_l as follows:

$$D_r^* = f(I_r^*, I_l^*)\,. \tag{2}$$

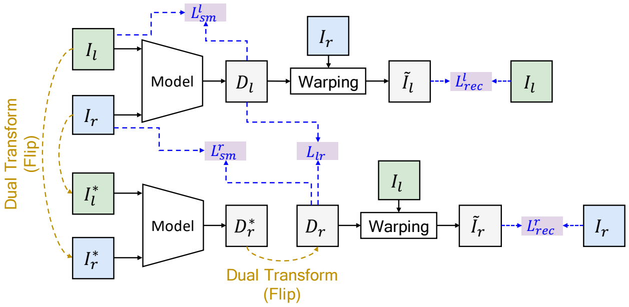

Figure 4. The overall structure for self-supervised depth estimation losses, including L_{rec}: Reconstruction Loss, L_{sm}: Smoothness Loss, and L_{lr}: Left-Right Consistency Loss.

As recommended in [23, 49], we use horizontal flip as the dual transform T. The flipped right disparity becomes the left disparity of the flipped right image and the flipped left image, as shown in Eq.(2). Therefore, we are able to use a single model to estimate both the left and right disparities. By doing so, the left disparity model is capable of estimating the right disparity. The dual disparity can be used to regularize the model with left-right consistency loss.

3.2. Overall Architecture

We present a simple technique for producing a disparity map of the right view by means of a geometry transform process. This method enables us to train the stereo-matching model end-to-end through self-supervised learning without requiring any additional parameters.

4D Cost Volumes: Congruent with the approach of PSMNet [3], we create a cost volume by combining the feature maps of the left view and their corresponding feature maps of the right view across each disparity level in the stereo image. This leads to a 4D cost volume for each stereo image, with dimensions of $H \times W \times D \times 2F$ ($height \times width \times disparity\ range \times feature\ size$).

Backbone: We select for the RealtimeStereo backbone [2] due to its advantages in mitigating computational challenges and accelerating training. RealtimeStereo features an attention-aware feature extraction method that leverages a number of stacked blueprint separable convolutions [17] to extract features with reduced computational cost, in addition to an attention-aware feature aggregation module to augment the representational capacity of features. We adopt the aforementioned method to construct the cost volume from the left-view feature maps and their corresponding right-view feature maps. Cost aggregation is then applied to regularize the cost volume and estimate the disparity, using the original stacked 3D hourglass architecture for 4D cost volume. Lastly, we utilize a cascaded architecture of 3D CNNs for estimating coarse-to-fine disparities.

Coarse-to-Fine Disparity: The multi-scale architecture of PSMNet [3] is designed to extract features for all image levels and to improve semantic and context information in a coarse-to-fine manner. However, the photometric objective for geometry supervision can lead to adverse effects on accuracy in self-supervised learning due to the interpolation blur problem [22] at different scales. To obviate this predicament, we adopt a single-scale architecture for training our backbones, PSMNet [3] and RealtimeStereo [2], using the final outputs of stacked hourglass 3D CNNs.

To elaborate on our method, we first make an initial estimation of a rough disparity map. We then use this map as input to iteratively refine the disparity estimation, resulting in a more accurate and precise map. After a similar strategy as proposed in [2], we begin by computing the complete disparity map at a lower resolution and subsequently calculating the disparity residuals at a higher resolution. This approach enables our model to reduce the search range, leading to a significant speedup in processing time. As a result, our self-supervised stereo-matching algorithm, which employs a lightweight CNN, is highly efficient, significantly lowering computational costs, and is capable of running in real-time on edge devices.

Summing up, our proposed pipeline is quite flexible and not confined to backbones [2, 3], 3D/4D cost volumes [3, 41], or single/multi-scale approaches [3, 22].

3.3. Self-Supervised Depth Estimation Losses

To learn disparity estimation through self-supervised learning, we extended the cost-volume-based stereo depth estimation model with geometry supervision, incorporating losses inspired by Godard *et al.* [15]. They are L_{rec}: Reconstruction Loss, L_{sm}: Smoothness Loss, and L_{lr}: Left-Right Consistency Loss. The overall structure for self-supervised depth estimation losses is shown in Figure. 4.

Reconstruction Loss: In this work, we follow the reconstruction loss proposed by Godard *et al.* [15]. Inspired by [45], they used a combination of L_1 loss [46] [16] and Structural Similarity (SSIM) [39] as their image reconstruction loss. Our model forms a reconstructed left-view image $\widetilde{I}_l$ by warping the right-view image I_r with the left disparity map D_l. We then compute the reconstruction loss L^l_{rec} for the left-view image by calculating the difference between the input image I_l and the reconstructed image $\widetilde{I}_l$.

$$
\begin{aligned}
L^l_{rec} = \frac{1}{N}\sum_{i,j}\Bigg(&\alpha\frac{1-\mathrm{SSIM}(I_l(i,j),\widetilde{I}_l(i,j))}{2} \\
&+(1-\alpha)\|I_l(i,j)-\widetilde{I}_l(i,j)\|\Bigg),
\end{aligned}
\tag{3}
$$

where N denotes the number of pixels, (i, j) represent the pixel coordinates, and α signifies the weight between L_1 loss and SSIM, which is 0.85 in this work. The reconstruction loss L^r_{rec} for the right-view image can be calculated in a similar way.

Smoothness Loss: We adapt the approach proposed in [36] to encourage both smoothness and edge preservation of the disparity map. Our algorithm uses the second-order derivative of the disparity map in the calculation of the smoothness loss. Given that depth discontinuities often appear on the image gradient, we utilize the image gradient to weight the smoothness loss, thus:

$$\begin{aligned} L^l_{sm} = \frac{1}{N} \sum_{i,j} \Big(&\|\partial^2_x D_l(i,j)\| e^{-\beta\|\partial_x I_l(i,j)\|} \\ &+ \|\partial^2_y D_l(i,j)\| e^{-\beta\|\partial_y I_l(i,j)\|} \Big) \ , \end{aligned} \tag{4}$$

where ∂D denotes the disparity gradient, ∂I denotes the image gradient, and β denotes the edge-weighted hyperparameter. The smoothness loss L^r_{sm} for the right disparity D_r can be calculated in a similar way. By minimizing the smoothness loss, our proposed method is able to align the disparity map with the edge structure of the input image while ensuring the smoothness of the disparity map.

Left and Right Consistency Loss: In this work, we aim to improve the accuracy of disparity maps and balance the performance of left and right estimation by ensuring the consistency between the left and right-view disparity maps. Following [21] [15] [35], we reconstruct the disparity map pair by warping them with each other and then comparing them with the original disparity maps to calculate the L_1 left-right consistency loss:

$$\begin{aligned} L_{lr} = \frac{1}{N} \sum_{i,j} (&\|D_l(i,j) - W(D_r(i,j), D_l(i,j))\| \\ &+ \|D_r(i,j) - W(D_l(i,j), D_r(i,j))\|) \ , \end{aligned} \tag{5}$$

where W represents the warping function. By doing so, we ensure that the left-view disparity map and the right-view disparity map are in consonance with each other, and improve the overall accuracy of the result.

4. Experiments and Analysis

We evaluate the ES^{3}Net against SOTA depth estimation methods on KITTI 2012 [11] and 2015 [25]. The assessment of computational efficiency is based on the number of frames processed per second (FPS) on the NVIDIA Jetson TX2 board. The board comes with a GP10B GPU and 2 CPUs, including a dual-core Denver 2 CPU and a quad-core ARM Cortex A57 2035Mhz. The maximum performance mode (Max-N) is enabled for this evaluation.

4.1. Datasets and Setting

We train our models on the Scene Flow [24] and KITTI raw dataset [11], and evaluate them on the train splits of KITTI 2012 [11] and KITTI 2015 [25].

Scene Flow dataset [24] comprises more than 35,000 stereo image pairs with ground truths.

KITTI raw dataset [11] is a real-world dataset containing over 42,000 stereo image pairs in various scenarios such as city, residential, road, campus, and person.

KITTI 2012 [11] contains 194 training stereo image pairs with ground truths and 195 testing stereo image pairs without ground truths.

KITTI 2015 [25] contains 200 training stereo image pairs with ground truths and 200 testing stereo image pairs without ground truths.

We follow the quantitative evaluation protocol of Monodepth [15] and evaluate all experimental results for each set by selecting 194 and 200 stereo image pairs with ground truths from the KITTI 2012 training set [11] and KITTI 2015 training set [25], respectively. We evaluate model performance not only on the scale-invariant metrics introduced by Eigen *et al*. [9] but also on the D1-all disparity error [11].

4.2. Implementation Details

PSMNet [3] in self-supervised learning setting: We did not follow the standard training settings on self-supervised learning for PSMNet [3] because of the considerable computation cost in the training procedure. Instead, we used the same configuration of PSMNet [3] to train our models from scratch on the Scene Flow dataset [24] before fine-tuning them on KITTI 2015 [25].

RealtimeStereo [2] in self-supervised learning setting: We follow the same training settings of most SOTA self-supervised learning methods for RealtimeStereo [2]. Notably, the RealtimeStereo [2] is trained on the KITTI raw dataset [11] with a learning rate of 0.0005. For the first 15 epochs of training, only the reconstruction loss is engaged. The smoothness and left-right consistency losses are added in the following 52 epochs. Within the training processes, the input images are randomly cropped to 288×576.

4.3. Comparison with SOTA Models

As shown in Tables 1 and 2, we compare the ES^{3}Net against other self-supervised SOTA depth estimation methods with respect to accuracy and efficiency. To begin with, we conducted a comparative analysis between the ES^{3}Net and other existing models [5, 15, 43, 47], using the KITTI 2015 dataset [25]. To ensure a fair comparison, we excluded the optical-flow-based estimator from our evaluation due to the absence of its efficiency scores. To demonstrate the computational efficiency and effectiveness of the proposed method, we compared the performance of our method with those of SOTA self-supervised monocular depth estimation methods [14, 15, 30, 33, 40], in addition to a lightweight method [28, 29], using the KITTI 2012 dataset [11].

Quantitative comparison for Stereo Models: Table 1 shows that under the existing self-supervised estimation

Method	AbsRel ↓	SqRel ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
EPC [43]	0.109	1.004	6.232	0.203	0.853	0.937	0.975
Zhong *et al.* [47]	0.075	1.726	4.857	0.165	**0.956**	0.976	0.985
Godard *et al.* [15]	0.068	0.835	4.392	0.146	0.942	0.978	0.989
Lai *et al.* (Stereo-only) [21]	0.078	0.811	4.700	0.174	0.918	0.965	0.983
Chi *et al.* (Stereo-only) [5]	0.063	**0.662**	4.312	0.140	-	-	-
UnOS (Stereo-only) [38]	**0.060**	0.833	4.187	**0.135**	0.955	**0.981**	**0.992**
ES^{3}Net	0.063	0.754	**4.096**	0.139	0.947	0.978	0.989

Table 1. Quantitative comparison with SOTA stereo-matching methods with reference to the KITTI 2015 dataset [25]. Bold-face and blue numbers indicate the first and second places, respectively.

Method	Training	Input Size	AbsRel ↓	SqRel ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑	Params	FPS
DepthHints [40]	I+K	1024×320	0.097	0.733	4.445	0.186	0.889	0.962	0.981	35M	2.13
MonoResMetach [33]	CS+K	1280×384	0.098	0.711	4.433	0.189	0.888	0.960	0.980	43M	2.58
MonoDepth [15]	CS+K	512×256	0.124	1.076	5.311	0.219	0.847	0.942	0.973	31M	4.88
3Net [30]	CS+K	512×256	0.117	0.905	4.982	0.210	0.856	0.948	0.976	48M	3.69
MonoDepth2 [14]	I+K	640×192	0.109	0.873	4.960	0.209	0.864	0.948	0.975	15M	17.24
PyD-Net [28]	CS+K	512×256	0.146	1.291	5.907	0.245	0.801	0.926	0.967	1.9M	34.57
PyD-Net2 [29]	CS+K	640×192	0.127	1.059	5.259	0.218	0.834	0.942	0.974	0.7M	46.73
ES^{3}Net (Stereo)	K	1248×384	0.045	0.303	2.299	0.108	0.958	0.984	0.993	0.023M	9.1

Table 2. Quantitative comparison with state-of-the-art monocular methods on the KITTI 2012 dataset [11] reveals that our method achieves excellent results without having to undergo additional training data. Here, ”K” denotes training on the KITTI raw dataset [11], ”CS” represents training on the Cityscapes dataset [7], and ”I” signifies pre-training on the ImageNet dataset [8]. Bold-face and blue numbers indicate the first and second places, respectively. All the performance of SOTA methods are directly taken from [29] using the Eigen split [9], while our ES^{3}Net is trained on KITTI raw dataset [11] and evaluated on KITTI 2012 training split [11]. The images are padded to 1248 × 384 to be adapted to our network.

methods, our ES^{3}Net performs at par or better than its task-specific counterparts without optical flow on those evaluation metrics with minimal parameter size (less than one thousand times) and high-speed (faster than ten times) (See Table 3). Moreover, our method can also be implemented on the embedded system NVIDIA Jetson TX2 (FPS≈ 9.1) for drone navigation and collision avoidance using **limited** computing resources.

Quantitative Comparison with Monocular Models: We conducted a further investigation into the trade-off between accuracy and speed between monocular methods and our proposed method. Table 2 presents a clear superiority of our method over DepthHints [40], which is currently the SOTA in monocular depth estimation. Our method achieves remarkable improvements in all evaluated metrics, including AbsRel, SqRel, RMSE, and $\text{RMSE}_{\log}$, with gains of 46.39%, 58.67%, 48.36%, and 41.94%, respectively. Furthermore, our method achieves a speed-up of over four times and a model complexity reduction of almost 1500 folds on the NVIDIA Jetson TX2. Although PyD-Net2 [29] achieves a higher frame rate (46 FPS), this comes at the expense of a significant drop in accuracy compared to our method, with a 64.57% decrease in AbsRel. Furthermore, our method achieves a model complexity reduction of over 20 folds compared to PyD-Net2 [29]. It is for this reason that our method is particularly well-suited for not only the embedded system but also wearable devices.

Qualitative Results: Figure 5 demonstrates the qualitative performance of our proposed method, in contrast to Godard *et al.* [15]. Although the competing methods were unsuccessful in producing the desired outcomes, our approach (third row) effectively estimated the position of the road railings. Furthermore, our method (first and second rows) outperformed the other methods in accurately identifying the shapes of vehicles and tree trunks. As per these findings, our approach has a distinct advantage in accurately predicting the position of objects, such as the road surface, which sets it apart from other methods.

4.4. Runtime Analysis on Different Architectures

Fig. 2 plots the comparison between the model parameter and $\delta < 1.25$ metrics for many evaluated models. It can be observed that our ES^{3}Net (red star in Fig. 2) achieves outstanding complexity-accuracy performance compared to other SOTA models.

Table 3 shows that our ES^{3}Net achieves either the first or second place in performance among all metrics in the stereo matching methods on KITTI 2015 [25]. Additionally,

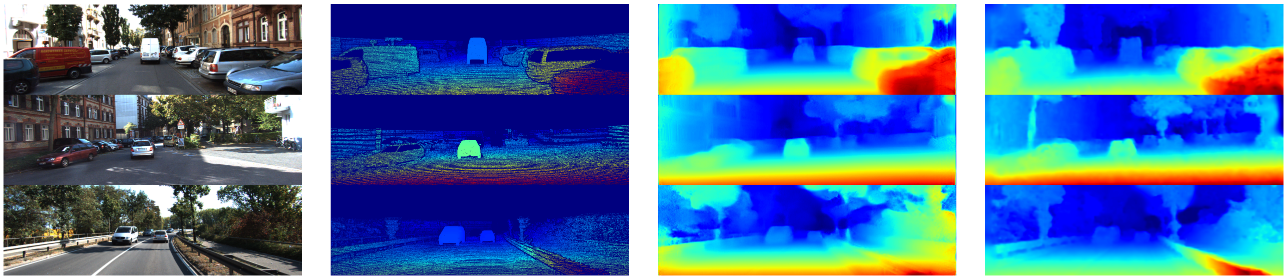
Input Ground Truth Godard *et al.* [15] Ours

Figure 5. Comparison of disparity estimation results regarding the KITTI 2015 dataset using the methods proposed by Godard *et al.* [15] and our ES^{3}Net.

Method	AbsRel ↓	SqRel ↓	RMSE ↓	RMSE$_{\log}$ ↓	Params	FPS
Stereo Method:						
Godard *et al.* [15]	0.068	0.835	4.392	0.146	31.6M	-
Lai *et al.* [21]	0.078	0.811	4.700	0.174	58.4M	-
Chi *et al.* [5]	0.063	**0.662**	4.312	0.140	> 31.6M	-
UnOS [38]	**0.060**	0.833	4.187	**0.135**	-	-
ES^{3}Net (Stereo)	0.063	0.754	**4.096**	0.139	**0.023M**	**9.1**
Monocular Method:						
DepthHints [40]	0.097	0.733	4.445	0.186	35M	2.13
MonoResMetach [33]	0.098	0.711	4.433	0.189	43M	2.58
MonoDepth [15]	0.124	1.076	5.311	0.219	31M	4.88
3Net [30]	0.117	0.905	4.982	0.210	48M	3.69
MonoDepth2 [14]	0.109	0.873	4.960	0.209	15M	17.24
PyD-Net [28]	0.146	1.291	5.907	0.245	1.9M	34.57
PyD-Net2 [29]	0.127	1.059	5.259	0.218	0.7M	**46.73**
ES^{3}Net (Stereo)	**0.045**	**0.303**	**2.299**	**0.108**	**0.023M**	9.1

Table 3. Quantitative comparison with SOTA methods. We obtained the runtime and parameter counts by processing images on a single NVIDIA Jetson TX2. Bold-face and blue numbers indicate the first and second places, respectively.

Method	D1-all ↓	AbsRel ↓	SqRel ↓	RMSE ↓	RMSE$_{\log}$ ↓
180° Rotation	8.687	0.064	0.778	4.106	0.140
Horizontal Flip	8.514	**0.063**	**0.754**	4.096	0.139
Zhong *et al.* [47]	**8.321**	0.064	0.833	**3.952**	**0.136**

Table 4. Quantitative results with dual transformation and cost-volume formation methods on the KITTI 2015 [25].

these methods could not run on Jeston NVIDIA TX2 successfully due to the model complexity and limited computational cost. However, our ES^{3}Net could perform SOTA performance for edge computing purposes. Notwithstanding that some monocular methods [14, 28, 29] can run in real-time (> 15 FPS) on NVIDIA Jeston TX2, their absolute relative errors are higher than 0.1, thus indicating that our ES^{3}Net outperforms them by over 55%. Even when compared to the SOTA method DepthHints [40], our ES^{3}Net consistently outperforms it across all evaluation metrics.

We highlight three advantages of our ES^{3}Net: (1) it achieves SOTA performance in comparison to stereo as well as monocular methods, making it a feasible choice for depth estimation tasks; (2) it can run in quasi-real-time on embedded systems like the NVIDIA Jetson TX2, enabling the integration of a stereo matching model for edge computing; and (3) it has a small parameter size, making it practical to deploy the model on wearable devices.

4.5. Ablation Study of Dual Disparity

We conducted the experiment to compare cost-volume formation method [47] and different dual transformations, such as horizontal flip and 180° rotation. Table 4 shows that the findings of all metrics as well as qualitative results (see Fig. 6) are relatively similar regardless of the form of the cost-volume. Consequently, we selected the flip, which has data augmentation benefits [15, 49], given that our dual disparity in the proposed method.

4.6. Comparisons of Single and Multi-scale

The Multi-scale in PSMNet [3] is designed to enhance the receptive field to extract information at the whole image level, thus enhancing the semantic and context information in a coarse-to-fine manner. However, when we adopted reconstruction loss for geometry supervision, we observed a decrease in accuracy due to pixel misalignment at different level scales [22]. We conducted several ablation studies on KITTI 2015 in order to verify the rationality and cogency of our method [25] so as to evaluate the impact on implementation in varying settings within a self-supervised manner. We analyzed 1) training with different backbones [2, 3]; 2) different construction methods of cost volume; and 3) single and multi-scale. Table 5 shows the ablation study of accuracy improvements when single scale is adopted. Despite making changes to any condition of 1) or 2), we were still able to focus on 3) to improve accuracy. As a result, the single scale method outperformed the multi-scale method in all metrics.

Architecture	Method	D	M	D1-all ↓	AbsRel ↓	SqRel ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
PSMNet [3]	Horizontal Flip	✓	✓	9.917	0.1081	2.1120	4.856	0.191	0.932	0.962	0.975
PSMNet [3]	Horizontal Flip	✓		**9.211**	**0.0929**	**1.6325**	**4.505**	**0.169**	**0.939**	**0.968**	**0.981**
RealtimeStereo [2]	Zhong *et al.* [47]		✓	9.401	0.097	2.244	5.555	0.198	0.934	0.967	0.980
RealtimeStereo [2]	Zhong *et al.* [47]			**8.321**	**0.064**	**0.833**	**3.952**	**0.136**	**0.950**	**0.981**	**0.990**
RealtimeStereo [2]	180° Rotation	✓	✓	9.153	0.070	0.999	4.504	0.151	0.942	0.976	0.988
RealtimeStereo [2]	180° Rotation	✓		**8.687**	**0.064**	**0.778**	**4.106**	**0.140**	**0.946**	**0.978**	**0.989**
RealtimeStereo [2]	Horizontal Flip	✓	✓	8.523	0.070	1.129	4.439	0.138	0.946	0.977	0.988
RealtimeStereo [2]	Horizontal Flip	✓		**8.514**	**0.063**	**0.754**	**4.096**	**0.139**	**0.947**	**0.978**	**0.989**

Table 5. Quantitative results with RealtimeStereo [2] and PSMNet [3] as the backbone on the KITTI 2015 [25]. ”M” represents multi-scale architecture and ”D” represents dual disparity.

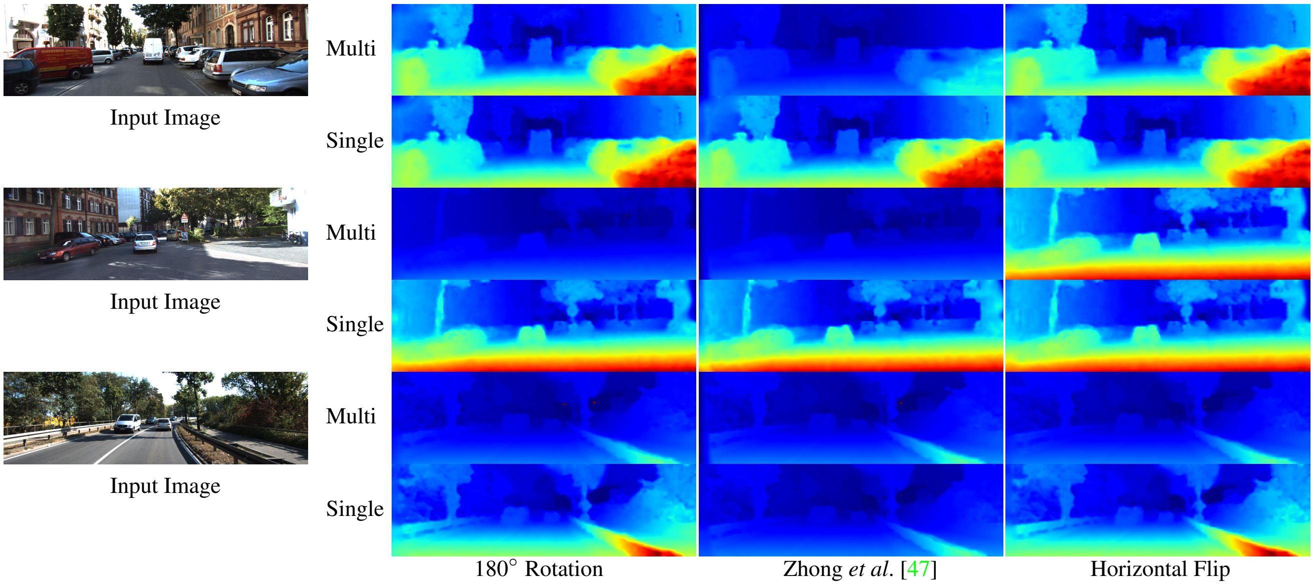

Figure 6. We compared the disparity estimation findings of the self-supervised RealtimeStereo [2] using the strategies including 180° Rotation, Zhong *et al.* [47], and Horizontal Flip. Additionally, we conducted a qualitative performance comparison using single and multi-scale architectures.

5. Conclusions

We introduce ES^{3}Net, a novel approach for efficient and accurate self-supervised stereo depth estimation with fast inference speed and small amount of model parameters. Our proposed method outperforms SOTA self-supervised monocular methods in terms of accuracy, achieving substantial improvements across multiple evaluation metrics while reducing model complexity by approximately 1500 times and speeding up over four times. Moreover, compared to existing self-supervised stereo estimation methods, ES^{3}Net achieves comparable performance in terms of accuracy.

Our work represents the pioneering application of self-supervised stereo matching in embedded vision, addressing the challenges of computational inefficiency and the lack of ground-truth depths. Our results demonstrate the potential of self-supervised stereo-matching as a valuable tool for enhancing safety and functionality in real-world applications of embedded vision. We also deployed ES^{3}Net to our drone to estimate absolute depth values for obstacle avoidance and navigation, showcasing its real-world applicability.

Acknowledgements

This work was partially sponsored by Qualcomm Technologies Inc. and National Science and Technology Council, Taiwan, with Grant 111-2634-F-A49-010. We are grateful to the National Center for High-performance Computing, Taiwan, for providing computational resources and facilities.

References

[1] Konstantinos Batsos, Changjiang Cai, and Philippos Mordohai. CBMV: A coalesced bidirectional matching volume for disparity estimation. In *CVPR*, 2018. 3

[2] Jia-Ren Chang, Pei-Chun Chang, and Yong-Sheng Chen. Attention-aware feature aggregation for real-time stereo matching on edge devices. In *ACCV*, 2020. 2, 4, 5, 7, 8

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018. 2, 4, 5, 7, 8

[4] Chao-Yu Chen. Autonomous obstacle avoidance for drone navigation based on real-time depth estimation. Master's thesis, National Yang Ming Chiao Tung University, 2021. 1

[5] Cheng Chi et al. Feature-level collaboration: Joint unsupervised learning of optical flow, stereo depth and camera motion. In *CVPR*, 2021. 2, 5, 6, 7

[6] Cheng Chi, Qingjie Wang, Tianyu Hao, Peng Guo, and Xin Yang. Feature-level collaboration: Joint unsupervised learning of optical flow, stereo depth and camera motion. In *CVPR*, 2021. 3

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014. 5, 6

[10] Thomas Eppenberger et al. Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking. In *IROS*, 2020. 2

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 1, 5, 6

[12] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010. 3

[13] Khang Truong Giang, Soohwan Song, and Sungho Jo. Curvature guided dynamic scale networks for multi-view stereo. In *ICLR*, 2022. 2

[14] Clément Godard et al. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019. 1, 2, 5, 6, 7

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7

[16] Xiaoyang Guo et al. Learning monocular depth by distilling cross-domain stereo networks. In *ECCV*, 2018. 4

[17] Daniel Haase and Manuel Amthor. Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets. In *CVPR*, 2020. 4

[18] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, 2005. 3

[19] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. In *ICCV*, 2019. 2

[20] Narsimlu Kemsaram, Anweshan Das, and Gijs Dubbelman. A stereo perception framework for autonomous vehicles. In *VTC2020-Spring*, 2020. 2

[21] Hsueh-Ying Lai, Yi-Hsuan Tsai, and Wei-Chen Chiu. Bridging stereo matching and optical flow via spatiotemporal correspondence. In *CVPR*, 2019. 2, 3, 5, 6, 7

[22] Kunming Luo et al. UPFlow: Upsampling pyramid for unsupervised optical flow learning. In *CVPR*, June 2021. 4, 7

[23] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *CVPR*, 2018. 4

[24] N. Mayer et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 5

[25] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 5, 6, 7, 8

[26] Rahnama Oscar et al. R3SGM: Real-time raster-respecting semi-global matching for power-constrained systems. In *FPT*, 2018. 3

[27] Wanli Peng et al. IDA-3D: Instance-depth-aware 3D object detection from stereo vision for autonomous driving. In *CVPR*, 2020. 2

[28] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *IROS*, 2018. 1, 2, 3, 5, 6, 7

[29] Matteo Poggi, Fabio Tosi, Filippo Aleotti, and Stefano Mattoccia. Real-time self-supervised monocular depth estimation without gpu. *IEEE T-ITS*, 2022. 1, 2, 3, 5, 6, 7

[30] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *3DV*, 2018. 1, 2, 5, 6, 7

[31] Oscar Rahnama, Tommaso Cavallari, Stuart Golodetz, Alessio Tonioni, Thomas Joy, Luigi Di Stefano, Simon Walker, and Philip HS Torr. Real-time highly accurate dense depth on a power budget using an FPGA-CPT hybrid soc. *IEEE TCAS-II*, 2019. 3

[32] Boitumelo Ruf, Sebastian Monka, Matthias Kollmann, and Michael Grinberg. Real-time on-board obstacle avoidance for UAVs based on embedded stereo vision. *ISPRS*, 2018. 2

[33] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *CVPR*, 2019. 1, 2, 3, 5, 6, 7

[34] Hengli Wang, Rui Fan, Peide Cai, and Ming Liu. PVStereo: Pyramid voting module for end-to-end self-supervised stereo matching. *IEEE RA-L*, 2021. 2

[35] Yang Wang et al. Joint unsupervised learning of optical flow and depth by watching stereo videos. *arXiv preprint arXiv:1810.03654*, 2018. 5

[36] Yang Wang et al. Occlusion aware unsupervised learning of optical flow. In *CVPR*, 2018. 5

[37] Yan Wang et al. Wang, yan and lai, zihang and huang, gao and wang, brian h and van der maaten, laurens and campbell, mark and weinberger, kilian q. In *ICRA*, 2019. 2

[38] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. UnOS: Unified unsupervised optical-flow

and stereo-depth estimation by watching videos. In *CVPR*, 2019. 2, 3, 6, 7

[39] Zhou Wang et al. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 4

[40] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *ICCV*, 2019. 1, 2, 3, 5, 6, 7

[41] Haofei Xu and Juyong Zhang. AANet: Adaptive aggregation network for efficient stereo matching. In *CVPR*, 2020. 2, 4

[42] Guorun Yang et al. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *CVPR*, 2019. 2

[43] Zhenheng Yang et al. Every pixel counts: Unsupervised geometry learning with holistic 3D motion understanding. In *ECCVW*, 2018. 2, 5, 6

[44] Yao Yao et al. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 2

[45] H Zhao et al. Is L2 a good loss function for neural networks for image processing? *arXiv preprint arXiv:1511.08861*, 2015. 4

[46] Hang Zhao, Orazio Gallo, Iuri Frosio, and Kautz. Loss functions for image restoration with neural networks. *IEEE TCI*, 2016. 4

[47] Yiran Zhong, Yuchao Dai, and Hongdong Li. Self-supervised learning for stereo matching with self-improving ability. *arXiv preprint arXiv:1709.00930*, 2017. 2, 5, 6, 7, 8

[48] Yiran Zhong, Hongdong Li, and Yuchao Dai. Open-world stereo video matching with deep RNN. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *ECCV*, 2018. 3

[49] Chaoyue Zou, Nan Li, Guanzhi Li, Chen Qian, and Ping Luo. SimpleRecon: 3D reconstruction without 3D convolutions. In *CVPR*, 2021. 4, 7