A Light-Weight Monocular Depth Estimation With Edge-Guided Occlusion Fading Reduction

Kuo-Shiuan Peng, Gregory Ditzler and Jerzy Rozenblit

October 13, 2020

A Light-Weight Monocular Depth Estimation With Edge-Guided Occlusion Fading Reduction

Kuo-Shiuan Peng[1], Gregory Ditzler[1], and Jerzy Rozenblit[1,2]

[1]Department of Electrical & Computer Engineering
[2]Department of Surgery
University of Arizona
Tucson, AZ 85721 USA
{kspeng,ditzler}@email.arizona.edu,jerzyr@arizona.edu

Abstract. Self-supervised monocular depth estimation methods suffer occlusion fading, which is a result of a lack of supervision by the ground truth pixels. A recent work introduced a post-processing method to reduce occlusion fading; however, the results have a severe halo effect. This work proposes a novel edge-guided post-processing method that reduces occlusion fading for self-supervised monocular depth estimation. We also introduce Atrous Spatial Pyramid Pooling with Forward-Path (ASPPF) into the network to reduce computational costs and improve inference performance. The proposed ASPPF-based network is lighter, faster, and better than current depth estimation networks. Our light-weight network only needs 7.6 million parameters and can achieve up to 67 frames per second for 256 × 512 inputs using a single nVIDIA GTX1080 GPU. The proposed network also outperforms the current state-of-the-art methods on the KITTI benchmark. The ASPPF-based network and edge-guided post-processing produces better results, both quantitatively and qualitatively than the competitors.

Keywords: Monocular depth estimation · Atrous Spatial Pyramid Pooling · Edge-Guided post-processing.

1 Introduction

Depth estimation is a fundamental problem with a long history in computer vision, and it also serves as the cornerstone for many machine perception applications, such as 3D reconstruction, autonomous vehicles, industrial machine vision, robotic interactions, etc. Unfortunately, successful research in depth estimation is dependent on the availability of multiple observations in a target scene. The constraint of the multiple observations can be overcome by using supervised methods that are accelerated by deep learning [1]. These methods aim to directly predict the pixel depth from a single image by learning the given a large amount of ground truth depth data. Despite the promising results from monocular depth, these methods suffer from the limitation of the quality and availability of ground truth pixel depth. Hence, self-supervised approaches that learn depth information from a single image have received increasing attention recently.

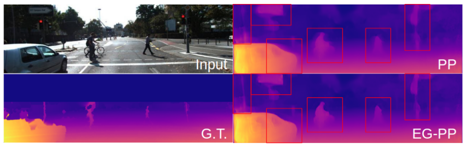


Fig. 1: Comparison between the conventional post-processing (PP) [2] and the proposed Edge-Guided post-processing (EG-PP) on KITTI dataset. Our method can reserve the sharp edge of the detected object depth and avoid the halo effect.

In the task of monocular depth estimation, self-supervised approaches only need supervision from stereo image pairs [2–4] or monocular video frames [5, 6]. In monocular depth estimation, the disparity is used as an intermediate product for depth estimation which can be converted to reconstruct the images with the inverse warping transform [7]. Recent works have introduced novel objective functions, such as the left-right consistency [2], correlational consistency [8], and adaptive global and local error [4]. Further, the high solution input solution has also been evaluated by [9] and [6] to detect the fine objects in images. Unfortunately, one major challenges with self-supervision is reducing false detections by using a compact network. It was shown in [2] that the deeper networks (e.g., Resnet50) can yield better depth estimates compared to a more compact network (e.g., VGG14). However, very deep networks are inefficient for real-time usage. Hence, a high performance light-weight network design for a depth estimation network is needed for real-time systems.

There are only a few works that focus on optimization of the network structure for self-supervision in real-time. Recently, a Light-Weight RefineNet was proposed for joint semantic segmentation and depth estimation [10]. This method was designed for supervision method. We have tested it and found out that its performance is limited when applying to the self-supervision. When we studied the multi-task network, we realized that the depth estimation and semantic segmentation can share the same feature representation in the network. Based on this finding, we argue that the semantic segmentation network structure can be used in the depth estimation network. In this paper, we introduce Atrous Spatial Pyramid Pooling (ASPP) module into our depth network from Deeplab semantic segmentation network [11]. We add forward-paths into the ASPP module and reduce the layers numbers of each Atrous convolutional layers to further optimize the network structure. We successfully designed a Light-weight DispNet that has only 20% size of the conventional depth network [2] but up to 55% faster in prediction. The prediction time of the proposed model can achieve 68 frames per second (15 msec per frame) using a single nVIDIA GTX1080 GPU.

Another limitation of self-supervision is the stereo dis-occlusion effect. Self-supervision relies on stereo image pairs to calibrate the estimation without the ground truth data. This self-supervision method inherits the stereo dis-occlusion effect from the objective function that uses stereo image pairs. Disparity ramps happen in the stereo dis-occlusion area of the estimated disparity and largely downgrade the estimation quality both quantitatively and qualitatively. The early researches in the occlusion detection used handcrafted the features to proceed the machine learning algorithms [12]. Recently, learning-based methods left-right symmetry [13, 14] have been proposed to estimate occlusions using the convolutional network. Then a post-processing method has been proposed to compensate the occlusion shading using a flip prediction alignment method [2], the compensated output suffers severe halo effect as shown in Fig. 1(PP). None of the current methods can fit the need of reducing the occlusion fading for a self-supervised depth estimation task. To address the occluding fading issue, we proposed an Edge-Guided post-processing (EG-PP) method to eliminate the occluding fading and halo effects in inference stage shown in Fig. 1(EG-PP). The proposed method effectively improves both the quantitative and qualitative results and can be applied to all the other self-supervision-based methods.

The main contributions of this paper are as follows: (1) We propose a Light-weight DispNet that is smaller, faster, and more accurate than the conventional DispNet. We have also proved that the last few dense feature layers of the encoder in DispNet are less efficient in extracting long-range features in our setting. (2) We a propose a novel Edge-Guided post-processing method to improve the performance. The occlusion fading is largely reduced with a minimized halo effect after applying our method. We also experimentally show that EG-PP is universal and can be applied to any other self-supervised method. (3) We evaluate our approach compared to the state-of-the-art on the KITTI dataset [15]. We fairly compare our model with priors using same conventional post-processing method to demonstrate that our method has fundamentally improved the network performance. (4) The proposed method is generalized to other unseen benchmark datasets. We test our method with the Make3d dataset [16] compared with other current state-of-the-arts quantitatively and qualitatively.

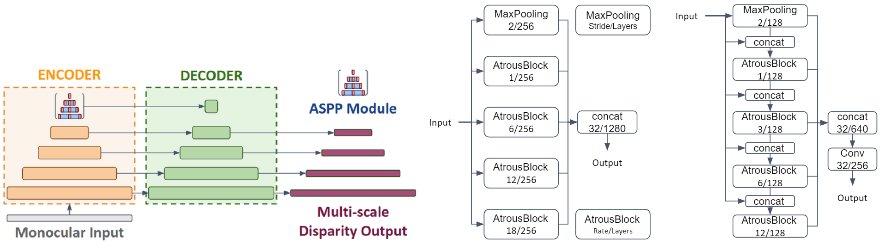
2 Methodology

Our model is inspired by the works of [11] and [2]. We first introduce the ASPP module [11] into our network design and optimize the network structure from the multiple conventional backbones. Then the objective function is directly adopted from [2]. The proposed Edge-Guided post-processing is explained in the last section.

2.1 Light-Weight Disparity Network

Many recent works designed their network by starting with DispNet [17], which is an autoencoder-based architecture. The multi-scale features from DispNet are exploited from the encoder, and the spatial resolution is recovered from the decoder. The recovered multi-scale spatial resolutions are the estimated disparities.

Since it was shown that depth estimation and semantic segmentation have common feature representations, they can share the base-network to perform



(a) Auto-Encoder (b) ASPP (c) ASPPF

Fig. 2: Light-Weight DispNet Structure, a) the proposed Auto-Encoder with ASPP module, b) the conventional ASPP module, and c) the proposed ASPPF module.

multi-task prediction [10]. Therefore, we use the network design concept of the semantic segmentation task. In the segmentation network, an effective module - Atrous Spatial Pyramid Pooling (ASPP) - was designed cascaded on top of the original network to detect long-range information [18]. We follow this design rule to modify the DispNet for depth estimation.

To further optimize our network, we analyzed the feature layers of the encoder, and we found that the last few convolutional blocks have a minor contribution to the estimation, especially after introducing the ASPP module shown in Fig. 2(b). Based on this observation, we simplify the DispNet by using the ASPP module to replace the last two convolutional blocks of the encoder. We also further use the maxpool to replace the convolutional block before the ASPP module. This design successfully reduces the network size of the network and produce a better performance than DispNet. We name this structure a Light-Weight DispNet. The proposed network structure is shown in Fig. 2(a). We here use [2] as a baseline example. If DispNet uses VGG14 as the backbone, the network parameters are about 31.6 million, and the inference time is about 19.11 msec. The corresponding Light-Weight DispNet only need 8.1 million (74% less) with inference time 14.74 msec (22.9% less).

Nevertheless, we further improve the network structure of the conventional ASPP module. Instead of using the Atrous Blocks in parallel, we add the forward-path for each Atrous Blocks (ASPPF) from the previous one which has the smaller dilation rate. The ASPPF can include more features from previous Atrous Block, but the computational cost would increase. Hence, we further reduce half of the number of the layers of each Atrous Block. A post convolutional block is also added after the concatenation of all the Atrous Blocks. We name this design as ASPP with Forward-Path (ASPPF) shown in Fig. 2(c). The proposed ASPPF modules has smaller size than the conventional ASPP module and the performance of the ASPPF design is better. The detailed analysis is elaborated in the section of Ablation Study.

2.2 Objective Function

We decide to adopt the objective function from [2] directly. There are several reasons. The most important consideration is that the aim of the left-right consistency function from [2] has demonstrated promising results among the recent works. The successors only have minor modifications. Besides, we would like to showcase that the proposed Light-Weight DispNet is substantially better than the conventional DispNet using the same objective function.

The objective function is a weighted sum of three terms: appearance (C_{ap}), disparity smoothness (C_{ds}), and left-right consistency (C_{cor}). The self-supervise total loss is defined as following:

$$C_s = \alpha_{ap} \times C_{ap} + \alpha_{ds} \times C_{ds} + \alpha_{lr} \times C_{cor} \tag{1}$$

The weights $(\alpha_{ap}, \alpha_{ds}, \alpha_{lr})$ are determined before optimization and set as $(1.0, 0.1, 1.0)$. The definition of each term can be found in [2].

The stereo dis-occlusion effect is one limitation of self-supervision for monocular depth estimation. Stereo dis-occlusion creates disparity ramps (occlusion fading) on both the left side of the image and the occluders. [2] proposed a post-processing method to reduce this effect. This form of post-processing estimates the disparity map d_l and the flipped disparity map d_l', which are from input image I and its horizontally flipped image I'. Then the flipped disparity map d_l' is flipped back as a d_l'' that aligns with d_l but where the occlusion fading is on the right of occluders as well as on the right side of the image. The final result is an average of d_l and d_l'', but assigning the first 2% on the left of the image using d_l and the last 2% on the right to the disparities from d_l.

The post-processing uses a mirror to generate a well-aligned projected disparity d_l'' that has right-side occlusion fading. The average of d_l and d_l'' can reduce the left-side occlusion fading because d_l'' has correct left-side estimation results. However, the right-side occlusion fading is also involved. This average process in the post-processing causes the halo effect in the final results, as shown in PP of Fig. 1. Instead of average, we propose an Edge-Guided weighted sum to suppress the occlusion fading of both d_l and d_l'' in the combination to reduce the halo effect, as shown in EG-PP of Fig 1.

2.3 Edge-Guided Post-Processing

The proposed Edge-Guided post-processing is depicted in Fig. 3. We follow the design concept of [2] to compute d_l and d_l'', but we add edge-aware weights (w, w'') in the final combination. Here we take w as an example to illustrate the algorithm. A right-edge detector is designed to extract the regional-edge confidence E. Instead of using Sobel detector, a wide-range horizontal gradient filter (f_{gx}) is used:

$$f_{gx} = \begin{bmatrix} 1 \;...\; 0 \; -1 \;... \\ 1 \;...\; 0 \; -1 \;... \\ 1 \;...\; 0 \; -1 \;... \end{bmatrix}_{3\times(2N)} /(3 \times (2N)) \tag{2}$$

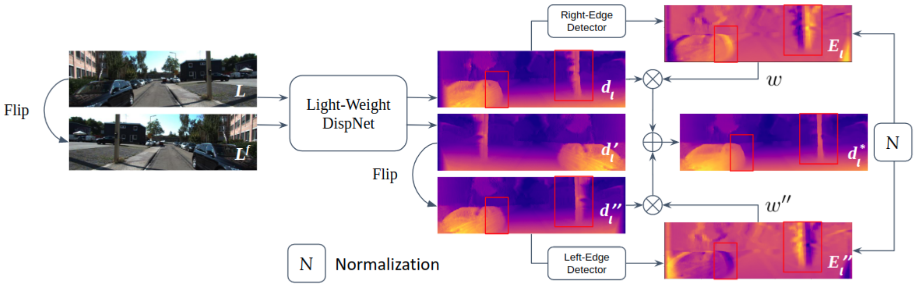


Fig. 3: Edge-Guided Post Processing

where N is the detection radius, whose default value is set to 10. After the convolution process ($\otimes$) of d_l and f_{gx}, we add an offset (b) and a gain (a) on the convolution result. Then a sigmoid function is applied:

$$E_l = \text{sigmoid}((d_l \otimes f_{gx} - b) * a) \tag{3}$$

where E_l is the right regional-edge confidence. The offset b and gain a are set as 0.5 and 32 to maximize the E_l in the range [0, 1]. In this equation, the right edge region has the confidence close to 1, while the left occlusion fading area has the confidence close to 0. The confidence of the flat area keeps around 0.5. The E_l'' is obtained in the same way but using the horizontal flipped f_{gx} as the left-edge detector. The last step is to normalize E_l and E_l'' to obtain w and w''. Then the final output $d_l^\star$ is a weighted sum of d_l and d_l'':

$$w = E_l/(E_l + E_l''), \qquad w'' = E_l''/(E_l + E_l'') \tag{4}$$

$$d_l^\star = w d_l + w'' d_l'' \tag{5}$$

Normalization is required to prevent overlap detection between E_l and E_l''. It ensures that the sum of w and w'' is 1 for each pixel and the final output $d_l^\star$ has no overhead compared to d_l and d_l''. There are no learning parameters and the computation cost is very low.

3 Experiments

Our benchmarks compare the performance of our approach to recent self-supervised monocular depth estimation methods. We selected Godard et al.'s work as our baseline and used the same benchmark configurations in [2]. We evaluated our approach on multiple aspects of KITTI dataset (i.e., both quantitative and qualitative). The ablation study is first conducted to prove the effectiveness of our approach using KITTI split. We then have a benchmark with the current start-of-the-art on Eigen split. We showcase the improvement of optimized network with the ASPP module by comparing to the priors. For fair comparison, we have all the methods with the conventional PP. In the last section of each scenario, we add the results applying the proposed EG-PP to demonstrate the effectiveness of EG-PP. We also generalized our method to other popular unseen data – Make3d.

3.1 Datasets, Metrics and Implementation

We evaluate the performance of our method on the KITTI benchmark [15]. We use two different test splits, KITTI and Eigen Split [1], of KITTI data to perform the ablation study for our method and the benchmark compared with the existing works. We follow the approach of [2] that uses 29k image pairs as the training set. We train our models by 8 batches and 100 epochs on the KITTI data. Furthermore, it has been shown by Godard et al. that pre-training with Cityscapes dataset can improve the performance on KITTI benchmark [2, 19]. We also include this strategy in the benchmark. In the combinational training on Cityscapes and KITTI dataset, we pre-train our models with an 8 batches and 50 epochs first on Cityscapes dataset and then on KITTI dataset. We use the evaluation metrics from Geiger et al. for depth estimation [15], which measures the error in meters from the ground truth and the percentage of depth that is within a threshold from the correct value. All of the reported error measurements represent the average error. Our methods were implemented in Tensorflow 1.15 [20] using Python 3.7 under the Ubuntu environment with a single NVIDIA GTX 1080 GPU. All input images are resized to 256×512 from the original size of the training image.

In the benchmarks, we show the experimental results of VGGASPPF (VGG8 Backbone) models. The VGGASPP (VGG14/VGG8 Backbone) model is included in the ablation study to show that the last three convolutional blocks are redundant. The computation costs of each backbone are also summarized in the Ablation section. We show that both our models have better performance than competitors in the benchmark studies. The our code for these experiments are publicly available [Github].

3.2 Results

Ablation Study In the ablation study, we analyze the quantitative performance improvement and the computational costs of our various designs using KITTI split on the KITTI dataset. For the quantitative performance improvement, we use VGG14 of the prior work [2] as the baseline. We first apply PP and EG-PP on the baseline to show the effectiveness of EG-PP. Then we start from VGGASPP with VGG14 to check up the improvement and the optimized VGGASPP/VGGASPPF with VGG8 are evaluated. Last, we include the results of pretrain C+K cases.

In the first section of Table 1, we show that the proposed EG-PP is effective to not only our model but also the baseline method [2]. This is particularly evident in terms of RMSE(log) and $\delta < 1.25$, which are the most challenging parts. In the remaining of the sections in Table 1, we can see that the proposed

Approach	Encoder	ASPP	PP	EG-PP	Train	ARD	SRD	RMSE	RMSE(log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
						Lower is better.				Higher is better.		
Baseline	VGG14				K	0.1240	1.3880	6.125	0.217	0.841	0.936	0.975
Our ASPP	VGG14	v			K	0.1183	1.2671	6.070	0.209	0.848	0.941	0.977
Our ASPP	VGG8	v			K	0.1134	1.1636	5.734	0.201	0.853	0.945	0.979
Our ASPPF	VGG8	v			K	0.1112	1.1263	5.693	0.201	0.859	0.946	0.979
Our ASPPF w/ PP	VGG8	v	v		K	0.1068	1.0033	5.460	0.193	0.861	0.949	0.981
Our ASPPF w/ EG-PP	VGG8	v		v	K	0.1062	0.9924	5.365	0.188	0.864	0.952	0.983
Our ASPPF w/ EG-PP	VGG8	v		v	C+K	**0.0992**	**0.9196**	**5.035**	**0.175**	**0.883**	**0.961**	**0.986**

Table 1: Quantitative results for different variants of our approach on the KITTI Stereo 2015 test dataset. We use our prior [2] as our baseline is shown in the first section. The training scenario is based on the KITTI training set (K), while the last section shows the results which are pre-trained by Cityscapes training sets (C+K). The best result in each subsection is shown in bold.

Approach	Blocks	Parameters	Predict(ms/FPS)
Baseline VGG	VGG14	31600072	19.11/52.32
Our VGGASPP	VGG14	38941384	22.03/45.4
Our VGGASPP	VGG8	8134344	14.74/67.83
Our VGGASPPF	VGG8	**7642440**	**14.5/68.97**
Baseline VGG+PP	VGG14	31600072	31.06/32.20
Our VGGASPPF+PP	VGG8	**7642440**	**21.93/45.6**
Our VGGASPPF+EGPP	VGG8	**7642440**	22.51/44.42

Table 2: Computational costs of different variants of our approach on the KITTI training dataset. The units of training of prediction are msec(ms)(lower is better) and frame per second(FPS) (Higher is better).

ASPP models have better performance than the baseline among all the metrics. The VGGASPP/VGG8 has an equivalent even better performance than VGGASPP/VGG14, which shows that the last few convolutional blocks in the VGG14 encoder are less effective when the ASPP module is applied. Last, VGGASPPF/VGG8 has better performance, although VGGASPP/VGG8 has a smaller computational cost.

In Table 2, we examine the computational costs of both our methods and the baseline. We included the VGG14 baseline to check the improvement rate. The results applying the post-processing are also included. The predictions happen in around 14.7ms/68FPS of the proposed VGGASPP/VGG8 model and 14.5ms/69 FPS of the proposed VGGASPPF/VGG8 model. Our VGGASPPF/VGG8 has only 24.2% of parameters but is 31.8% faster in the prediction compared to VGG14 model of the baseline. When the post-processing method is applied, the model's input becomes a batch of two images (left and flipped left images) and the prediction efficiency drops to around 22ms/45.6FPS of VGGASPPF/VGG8. When the proposed Edge-Guided post-processing is applied to the proposed VGGASPPF/VGG8 model, there is only 2.5% loss in computation time.

State-of-the-art comparison In the benchmark, we include the post-processing in the comparison. We only include VGGASPPF/VGG8 in the evaluation. From the training aspect of view, there are K only and C+K cases. An exceptional case is that [6] has ImageNet [21] as the pre-train dataset. Another special case

Approach	Train	Test	PP	ARD	SRD	RMSE	RMSE(log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
				Lower is better.				Higher is better.		
Monodepth [2]	K	E - 80m	Y	0.1480	1.3440	5.927	0.247	0.803	0.922	0.964
Fei et al. [22]	K	E - 80m	Y	0.1390	1.2110	5.702	0.239	0.816	0.928	0.966
Monodepth2 [6]	K	E - 80m	Y	0.1300	1.1440	5.485	0.232	0.831	0.932	0.968
Wong et al. [4]	K	E - 80m	Y	0.1264	0.9935	5.282	0.222	0.831	0.939	0.973
Ours	K	E - 80m	Y+	**0.1072**	**0.9079**	**4.877**	**0.202**	**0.862**	**0.945**	**0.975**
Monodepth [2]	C+K	E - 80m	Y	0.1140	0.8980	4.935	0.206	0.861	0.949	0.976
Fei et al. [22]	C+K	E - 80m	Y	0.1120	0.8360	4.8920	0.204	0.862	0.950	0.977
Monodepth2 [6]	I+K	E - 80m	Y	0.1090	0.873	4.960	0.209	0.864	0.948	0.975
Ours	C+K	E - 80m	Y+	**0.1015**	**0.7966**	**4.633**	**0.193**	**0.876**	**0.953**	**0.979**
Monodepth2 [6] (1024×320)	I+K	E - 80m	Y	0.1070	0.8490	4.764	0.201	0.874	0.953	0.977
Ours (1024×320)	C+K	E - 80m	Y+	**0.0999**	**0.7665**	**4.455**	**0.189**	**0.881**	**0.956**	**0.980**
Monodepth [2]	K	E - 50m	Y	0.1400	0.9760	4.471	0.232	0.818	0.931	0.969
Fei et al. [22]	K	E - 50m	Y	0.1320	0.8910	4.3120	0.225	0.831	0.936	0.970
Wong et al. [4]	K	E - 50m	Y	0.1202	0.7432	4.022	0.209	0.845	0.946	0.976
Ours	K	E - 50m	Y+	**0.1009**	**0.6480**	**3.656**	**0.190**	**0.875**	**0.952**	**0.979**
Monodepth [2]	C+K	E - 50m	Y	0.1080	0.6570	3.729	0.194	0.873	0.954	0.979
Fei et al. [22]	C+K	E - 50m	Y	0.1060	0.6150	3.697	0.192	0.874	0.956	0.980
Ours	C+K	E - 50m	Y+	**0.0959**	**0.5853**	**3.486**	**0.181**	**0.887**	**0.958**	**0.981**

Table 3: This table shows the additional benchmark specifically compared with recent methods. All the results use the crop defined by Garg et al. [3]. In the PP column, Y means using the conventional PP, while Y+ means using the proposed EG-PP. The results which are pre-trained with Cityscapes (C) or ImageNet (I) are evaluated as well. The high resolution results are also included for the comparison with [6].

is the high resolution input case of [9] and [6]. We also implement the same resolution input on our model. In test cases, we use Eigen-split with full and near distance under Garg et al. crop shown in Table 3 [3]. Our results with conventional post-processing are still better than the recent priors. When we apply EG-PP, our results become significantly better, and only the last accuracy term is slightly behind.

The improved performance of our method is not only quantitative, but also qualitative. The results are shown in Fig. 4. Our results have a much better ability to reproduce clear object shapes and edges in any size, especially the signs and trunks in the test images. The halo effects around objects (e.g., cars, signs, trunks,etc.) are largely reduced using the proposed EG-PP. In the visual evaluation, we provide more accurate and visually appealing images to viewers.

Edge-Guided Post-process Generalization The proposed EG-PP method is universal to be applied to any self-supervision depth estimation methods. We have prepared the experiment results of [2], [4], and ours with non-PP, PP, and proposed EG-PP in Table 4. The performance of the quantitative results are improved for both of the two methods except the ARD and SRD terms of [2] and [4]. Thus, this result shows that the proposed EG-PP method can be applied to other self-supervision methods to improve the performance.

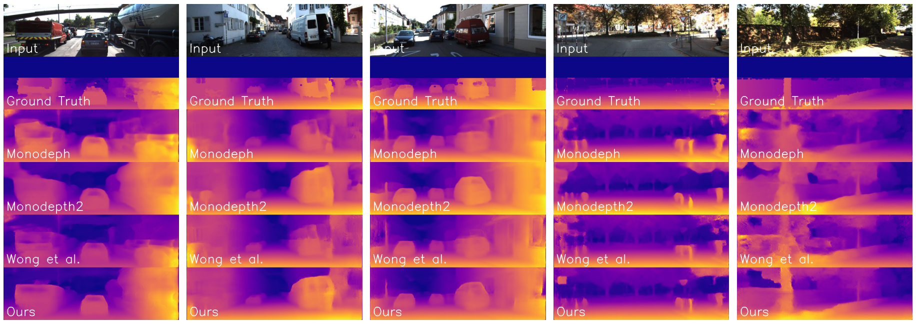


Fig. 4: Benchmark of qualitative results on KITTI dataset Eigen Split. We compare Ours with the priors - Monodepth [2], Monodepth2 [6], and Wong et al. [4]. Our VGGASPPF has applied the proposed Edge-Guided post-processing. Our results can capture much more clear object shapes, such as signs, cars, and trunks than priors. The halo effects are also effectively reduced in our results.

Approach	Train	PP	ARD	SRD	RMSE	RMSE(log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
			Lower is better.				Higher is better.		
Monodepth [2]	K	N	0.1240	1.3880	6.125	0.217	0.841	0.936	0.975
Monodepth [2]	K	Y	**0.1170**	**1.1773**	5.811	0.206	0.847	0.942	0.977
Monodepth [2]	K	Y+	**0.1170**	1.1873	**5.766**	**0.204**	**0.850**	**0.944**	**0.978**
Wong et al. [4]	K	N	0.1112	1.1350	5.682	0.202	0.854	0.946	0.979
Wong et al. [4]	K	Y	**0.1058**	**0.9811**	5.424	0.193	0.857	0.949	0.981
Wong et al. [4]	K	Y+	0.1074	1.0394	**5.417**	**0.191**	**0.861**	**0.950**	**0.982**
Ours	K	N	0.1112	1.1263	5.693	0.201	0.859	0.946	0.979
Ours	K	Y	0.1068	1.0033	5.460	0.193	0.861	0.949	0.981
Ours	K	Y+	**0.1062**	**0.9924**	**5.365**	**0.188**	**0.864**	**0.952**	**0.983**

Table 4: Quantitative results for proposed Edge-Guided Post-Processing method on the KITTI Stereo 2015 test dataset. The PP means using post-processing. N is no PP, Y is the conventional PP proposed by [2], and Y+ is the proposed Edge-Guided PP. The best performance of each metric in each section is bolded. The proposed Edge-Guided PP can effectively improve the performance especially the most challenging accuracy metric $\delta < 1.25$.

Dataset Generalization We further apply our method to another unseen dataset to verify the ability of the generalization. We follow the idea of [2] and [6] to evaluate Make3d dataset [16]. We use the same setting as these two priors that Cityscapes Dataset only trains our model, and we only consider less than 70 meters depth in evaluation. We also used the same evaluation code from [2] to generate the final results. The quantitative results are shown in Table 5. We have shown that our model has better results using stereo supervision. On the other hand, Fig. 5 shows the qualitative results. We compare our results to monodepth [2] and monodepth2 [6], where monodepth2 [6] is supervised by the monocular sequence. Our model provides better visual performance than monodepth [2] in clarity and competitive compared to monodepth2 [6].

Approach	abs_rel	sq_rel	rmse	rmse_log
Monodepth [2]	0.544	10.94	11.76	0.193
Fei et al. [22]	0.458	8.681	12.335	0.164
Wong et al. [4]	0.427	8.183	11.781	**0.156**
Ours	**0.365**	**5.073**	**8.135**	0.174

Table 5: Quantitative results on Make3d.

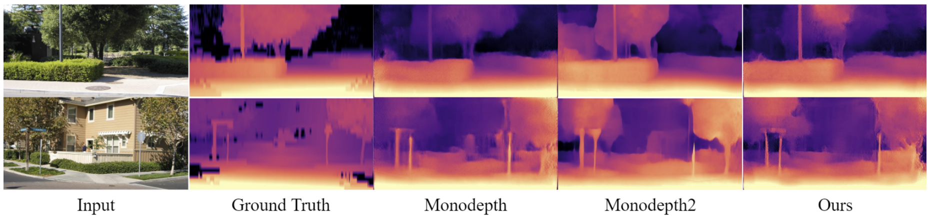


Fig. 5: Qualitative results of Make3d Dataset. We compare our method with monodepth [2] and monodepth2 [6]. The results of two references come from the source papers and codes.

4 Conclusion

We proposed a Light-weight DispNet and a novel Edge-Guided post-processing method to improve a self-supervised monocular depth estimator's performance. Our primary contribution is that the proposed Light-weight DispNet demonstrates the inherent capability to capture long-range features to estimate better the depth map with a much smaller network structure than the current commonly used DispNet. Another contribution of this work is that the Edge-Guided post-processing can resolve most occlusion fading effect of self-supervision methods. It can effectively reduce the halo effect that comes from the conventional post-processing to yield the object shape. The proposed EdgeGuided post-processing is suitable for all the self-supervised monocular depth estimators.

Acknowledgment

K. Peng and J. Rozenblit were supported by the National Science Foundation under #1622589. G. Ditzler was supported by the Department of Energy #DE-NA0003946 and National Science Foundation CAREER #1943552.

References

1. D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Neural Information Processing Systems*, 2014.
2. C. Godard, O. M. Aodha, and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conf Comp Vis Patt Recog*, 2017.

3. R. Garg, V. Kumar, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conf. on Comp. Vis.*, 2016.
4. A. Wong and S. Soatto, "Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction," in *IEEE Conf. Comp. Vis and Patt. Recog.*, 2019.
5. Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *IEEE Conf. Comp. Vis and Patt. Recog.*, 2018.
6. C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow., "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3828–3838, 2019.
7. M. Jaderberg, K. S. andrew Zisserman, and K. kavukcuoglu, "Spatial transformer networks," in *Neural Information Processing Systems*, 2015.
8. K.-S. Peng, G. Ditzler, and J. W. Rozenblit, "Self-supervised correlational monocular depth estimation using resvgg network," *In 7th IIAE International Conference on Intelligent Systems and Image Processing*, pp. 93–102, 2019.
9. S. Pillai, R. Ambruş, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *Int'l Conf. on Robotics and Automation*, 2019.
10. V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, "Real-time joint semantic segmentation and depth estimation using asymmetric annotations," *Int'l Conf. on Robotics and Automation*, pp. 7101–7107, 2019.
11. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv:1706.05587*, 2017.
12. A. Humayun, O. M. Aodha, and G. J. Brostow, "Learning to find occlusion regions," *In 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2161–2168, 2011.
13. A. Li and Z. Yuan, "Symmnet: a symmetric convolutional neural network for occlusion detection," *arXiv preprint arXiv:1807.00959*, 2018.
14. E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions and motion and depth boundaries with a generic network for disparity and optical flow or scene flow estimation," *European Conf. on Computer Vision*, pp. 614–630, 2018.
15. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE Conf Comp Vis Patt Recog*, 2012.
16. A. Saxena, M. Sun, and A. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Trans Patt Anal Mach Intell*, vol. 31, no. 5, pp. 824–840, 2009.
17. 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 *IEEE Conf. on Comp. Vis. and Patt. Recog*, 2016.
18. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *In Proceedings of the European conference on computer vision*, pp. 801–818, 2018.
19. 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 *IEEE conference on computer vision and pattern recognition*, 2016.
20. M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.
21. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, and Z. H. et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
22. X. Fei, A. Wong, and S. Soatto, "Geo-supervised visual depth prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1661–1668, 2019.