Self-Supervised Correlational Monocular Depth Estimation using ResVGG Network

Kuo Shiuan Peng*, Ditzler Gregory, Jerzy Rozenblit

Department of Electrical and Computer Engineering
The University of Arizona
1230 E Speedway Blvd., Tucson, AZ, 85705, USA

*Corresponding Author: kspeng@email.arizona.edu

Abstract

Self-supervised monocular depth estimation (SMDE) has recently received significant attention in computer vision. Leveraging the development of deep learning approaches, SMDE provides a solution to the applications of automation, navigation, and scene understanding. In this paper, we propose a novel training objective and learning network to perform a single image depth estimation in our convolutional neural network without the ground truth depth data. The proposed training objective enables the learning network to learn the stereo image correlation in training and estimates the image depth from a single input image in prediction. The proposed learning network ResVGG is a hybrid structure of Resnet50 and VGG-16. The proposed ResVGG has a similar performance as Resnet50 but needs much less computational costs. We demonstrate that our proposed method has competitive accuracy comparing to the current state-of-the-art on KITTI dataset and achieves the frame rates of 32 frame per second (FPS) in prediction using a single NVIDIA GTX 1080 GPU. Furthermore, the proposed method can potentially support visual odometry depth estimation.

Keywords: monocular depth estimation, deep learning, self-supervised.

1. Introduction

Depth estimation is one of the fundamental problems with a long history in computer vision. It also serves as the cornerstone for many machine perception applications, such as 3D reconstruction, auto-driving system, industrial machine vision, robotics interaction, etc. However, most research is performed based on the availability of multiple observations in target scenes. The constraint of the multiple observations can be overcome by the supervised method because of the emerging deep learning technology [1-3]. These methods aim to directly predict the pixel depth from a single image by learning the given ground truth depth data of a large amount of dataset. Despite the promising results of the monocular depth prediction, these methods suffer from the limitation of the quality and availability of the collected ground truth pixel depth. Hence, the self-supervised approaches learning the depth information from a single image has received increasing attention in recent research.

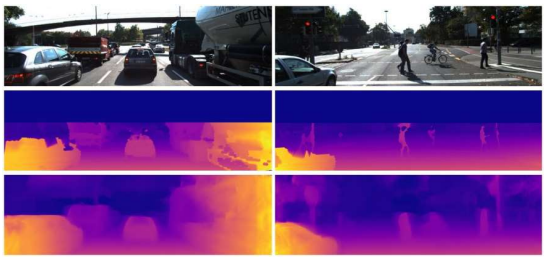


Fig. 1. The prediction results of our method on KITTI benchmark dataset. The first row shows the input image, the second row is ground truth disparities, and last row gives our results. Our method can successfully estimate the depth of different object structures, such as cars, bikes, signs, etc.

In the task of monocular depth estimation, the input source is a monocular image (e.g., a left image). Then the corresponding another view (e.g., the right image), can be reconstructed by the estimated right depth and the input left image (left) using a warping function[4]. Hence, the reconstructed right view is supervised by an actual right image. The estimated depth can also be calibrated in the regression for the reconstructed right view. If the stereo image pairs

DOI: 10.12792/icisip2019.019

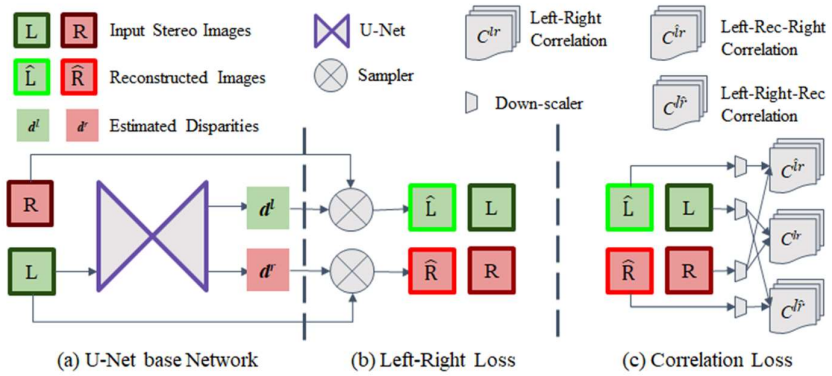


Fig. 2. Network design of the proposed method (a) the fundamental architecture of U-Net network, (b) Left-right loss (7), (c) the proposed correlation layer loss.

were available, the geometric cue between the stereo training image pairs can be further considered. To achieve this goal, the network first predicts the left and right depth simultaneously. Then the objective function is designed to learn the geometrical relationship, such as the stereo image correlation, between the reconstructed left and right images. Several existing methods use this idea, but there are some limitations. For example, the stereo image correlation is not explicitly defined in the training objective [5], or the learning network architecture is too deep, e.g. Resnet50[6], to achieve real-time application[7,8] on a consumer GPU. We overcome these limitations by introducing the exact stereo image correlation in the training object and a new network backbone architecture. The proposed Self- supervised Correlational Monocular Depth Estimation (CMODE) improve the prediction performance in both quantitative and qualitative aspects.

The example visual results of our work are presented in Fig. 1. The computational costs including parameters and computation time of the proposed method are much lower than the current existing method. Per prediction only needs around 30 milliseconds using a single NVIDIA GTX1080 GPU. The main contributions are as follows:

1. First, we propose a novel training objective function that the network can learn the stereo image correlation in training and estimate the depth from a single image without the ground truth data.

2. Second, a new network architecture is proposed to largely reduce computational costs including parameters and computation efficiency. The proposed ResVGG network performs similar to Resnet50 with much smaller architecture and lower computation time.

3. Third, we evaluate the our method and compare with the state-of-the-art on the challenging KITTI dataset.

2. Related Works

In the past few years, a large body of literature on depth estimation using learning-based methods have been proposed for either stereo view or single view. Most stereo estimation algorithms focus on calculating the similarity between each pixel in the left and right images. The depth estimation problem is generally converted to a 1D search problem for each pixel. The process of treating the matching process as a supervised learning problem and training an objective function to predict the correspondences has shown superior results compared to the conventional handcrafted similarity searching approaches[9,10]. A fully convolutional deep neural network[11] named DispNet[12] has been proposed to directly predict the correspondences between two images. Then the disparity of each pixel can be directly predicted by minimizing the regression training loss.

On the other hand, the monocular depth estimation has drawn more attention recently. An image depth estimation task is typically considered as a disparity estimation between the stereo image pairs. In the case of an object with deep depth in an image, the locations in the stereo image pair are almost identical because the object disparity between the left and right image planes are close to zero. A multi-scale deep network was firstly designed to directly estimate dense pixel depth from a single image[2]. Other advantages of this approach, such as using a more robust loss function[13], incorporating strong scene prior[14], changing the loss function from regression to classification [15], have been adopted in other works. These methods above still require high quality and pixel aligned ground truth depth at training time to estimate the depth from a single image. However, the ground truth depth information

is luxury to acquire in various real-world settings.

Recently, several methods not requiring ground truth at training have been proposed, such as a geometry constraint based single view estimator using an image reconstruction loss[5]. The image reconstruction of this approach, however, is not fully differentiable. One of the more successful algorithms is to use self-supervision in the form of synchronized stereo pairs. The training objective is to pose depth estimation as an image reconstruction process in training and perform monocular depth in prediction. The key of this approach is the warping function[4] that reconstructs one image from a stereo pair onto the other using the predicted depth. The learning process calibrates the reconstructed and actual image pairs. A successful example using left-right consistency and a bilinear sampler obtains a fully differentiable training loss[7] to generate an outstanding estimated depth. From this approach, there are several extensions, such as training with semi-supervised data[8] or involving temporal information[16].

Instead of using rectified stereo image pairs as the training target, another approach of the self-supervision is to train the network with the adjacent temporal frames from the input monocular video sequence. From unstructured video sequences, a set of learning network[17] is proposed to estimate the image depth and camera pose simultaneously. The estimated camera pose is used to help constrain the depth estimation network in training only. Inspired by this approach, several methods have extended the application of the temporal information to achieve the goal of the self-supervised monocular depth estimation, such as involving spatial and temporal dense information[18-20], structure-from-motion algorithm[21], or 3D Geometric Constraints[22]. A different approach offering monocular sequence is to estimate the optical flow to calculate pixel's 3D information and then the image depth can be retrieved from a pixel's 3D position. A recent approach yields prolific results using image correlation to estimate the image displacement of the optical flow[23] and the image depth[12,19]. Image correlation between left-right image pair can match the image features to extract object displacement and a set of dense layers are calculated. Learning the depth estimation can benefit from the image correlation dense layers to predict a better shape of the object depth. Unfortunately, the conventional approaches having contracted the image correlation based on the stereo images[12] is not feasible to be applied to the monocular application. A recent two-stage method[19] overcome the limitation of the stereo input required in computing image correlation by synthesizing the right view from the left view input, but it needs two separate models, resulting high computational costs. Currently no model is available for considering the image correlation from a single image in a single model.

In this work, we are interested in predicting the image depth from a single input image. We argue that the network can learn the image correlation information from the stereo training image set although image correlation is unavailable from the single input image in prediction. Based on this argument, we design a novel objective function to enable the depth network to learn the image correlation in training. Hence, the correlation information in stereo image pairs is embedded inside the learning network and the estimated depth from a single image can still retrieve the correlation information from the network. To further reduce the computation costs, we propose a new network structure which has much lower computational costs but outperforms the current commonly used network structure.

3. Methodologies

3.1 Depth Estimation Using Correlation

Our model is inspired by the works of (23, 7). We first review the image correlation, followed by the network design, and the overall object function.

The proposed method aims to involve the stereo correlation information into a monocular depth estimation task as shown in Fig. 2. The network of our proposal has two parts: prediction and training loss. The prediction part is the U-Net of Fig. 2(a), discussed in next section, and the training loss is the rest of the network across the Fig. 2(b) and (c). Except using the Left-right training loss[7], our work proposes a novel correlation loss shown in Fig. 2(c). The prediction network can learn the stereo correlation through the proposed correlation loss. In Fig. 2(c), we use the method of the correlation layer[23] to calculate the image correlation between the left and right-view image pairs. The main process of the correlation layer is to perform multiplicative patch comparisons between two images. Rather than computing the correlation between two feature maps as the conventional methods, we propose to measure the correlation between two RGB images directly.

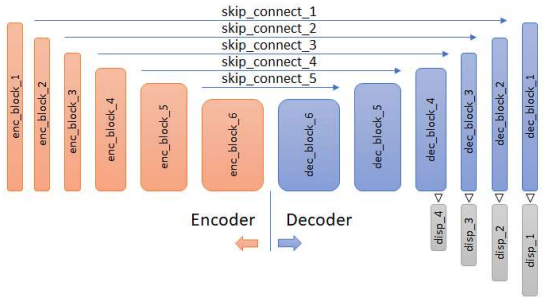


Fig. 3. Network design of the proposed method (a) the fundamental architecture of U-Net network, (b) Left-right loss[7], (c) the proposed correlation layer loss.

Given a set of image pair L and R with the dimension of width (w), height (h), and channel (c), we consider a single comparison of two patches within the image pairs (Fig. 2(c)). The definition of the correlation between two patches centered at x1 in L image and x2 in R image is given by

$$c(x_1,x_2)=\sum_{o\in[-k,k]\times[-k,k]}\langle L(x_1+o),R(x_2+o)\rangle \qquad (1)$$

which is a patch of a square shape of size $K := 2k + 1$ and the function $\langle\cdot,\cdot\rangle$ is a convolution between two patches as the one in neural network. However, the weights are not trainable in the correlation layer since the two image patches are known.

Theoretically, it costs $c\times K^2\times w\times h$ computations to compute $c(x_1, x_2)$ within the image size $w\times h$ when we search whole images to compute the correlation c. However, in depth estimation applications, the displacement, *disp*, of an object between the left and right image is confined in a range, determined by the stereo camera configuration.

Therefore, we only need to consider a neighborhood of size $D:=2\times disp+1$ as the total displacement range while computing the correlation c. The stride of the convolution is the unit step in the calculation. We only need the 1D correlation layer in this application because the depth estimation is a horizontal alignment problem. In short, the correlation layer yields the output size $w\times h\times D$ and is fully differentiable.

3.2 3Depth Estimation Network

The network architecture we propose is based on (7), which is a state-of-art depth estimator that considers the left-right consistency (LRC). The core network of (7) is an autoencoder architecture that is U-Net[24]. Based on the autoencoder architecture, the unique design of U-Net is the skip connections inside the multi-scale feature maps shown in Fig. 3. During the encoding process of the left side network, the output of each scale for the encoder feature layers is passed to the scale of the decoder feature layers, which is shown in the right-side network as the part of the input. The LRC method has the structure of U-Net and generates the multi-scale estimated depths from the output of each scale of the decoder feature layers.

Table 1. Proposed ResVGG encoder.

Modules	Layers	Ch_I/O	Scale
enc_block_1	conv_7x7	3/64	2
enc_block_2	maxpool_3x3	64/64	4
enc_block_3	Res50_blockx3	64/64	8
enc_block_4	Res50_blockx4	64/128	16
enc_block_5	VGG_block_3x3	128/256	32
enc_block_6	VGG_block_3x3	256/512	64

The most commonly used encoders in the selected UNet architecture are VGG-16[25] and Resnet50. The VGG backbone is adopted by (7) and performers well in both quantitative and qualitative result. Additionally, the computational efficiency of the VGG backbone in (7) can achieve over 30 FPS in our simulation setting. On the other hand, Resnet50 backbone was selected by (7) and (8) as well. The Resent50 backbone has better performance in both quantitative and qualitative results than the VGG backbone, but the computational efficiency is only 23 FPS and the training model is much larger. In other words, the Resnet50 backbone has a better ability to detect the features of the scene to predict the image depth but the network architecture is more complicated. On the contrary, the VGG backbone has a smaller and simpler network architecture to accomplish the feature detection process although the detection result of the VGG backbone is not as good as Resnet50. In our experiments, we found that the Resnet50 backbone converges the detection process in first two Resnet50 blocks and the rest of convolution layers are not necessary to keep using Resnet50 structure. Hence, we propose a hybrid architecture, named ResVGG, to combine Resnet50 in the first half part and VGG in the second half part network architecture. The proposed ResVGG architecture is shown in Table 1. In the simulation, the proposed ResVGG architecture has competitive results comparing to ResNet50 in the evaluation and has much less parameters than both ResNet50 and VGG-16. In our design, the encoding blocks (enc blocks) from 1 to 4 are the Resnet50 backbone, while the rest enc blocks are the VGG backbone.

Rather than directly learning from the ground truth depth data, (7) is supervised by training *L/R* and reconstructed $\hat{L}/\hat{R}$ stereo image pairs. The design scenario is shown in Fig. 2(b). Depth estimation methods typically predict the disparity first and then use the camera configuration to calculate the corresponding depth[26]. The disparity d can be converted back to the depth ^ d by the formula $\hat{d} = bf/d$, where b is given baseline distance between cameras and f is the camera focus length. After the disparities are predicted (d^l, d^r) then the reconstructed image pairs ($\hat{L}$, $\hat{R}$) are converted from (d^l, d^r) and the training image pairs (*L, R*) using a warping sampler *W*[4]. The assumption is that one side view image can be fully reconstructed by the other side view, which can be expressed by $\hat{L} = W(R, d^l)$ and $\hat{R} = W(L, d^r)$. In the end, the desired output is the estimated left disparities (d^l)of the input image (*L*). This method is considered as the self-supervised learning of the depth estimation because the learning process only relies on input images themselves and do not need ground truth.

The proposed approach introduces left-right image correlation into the loss function of the network as shown in Fig.2(c). We aim to minimize the error of the correlations between the reconstructed images and the training images. The correlation layer identifies the object-location correlation between images and is calculated by a correlation layer defined in the previous section. The ground truth Left-Right Correlation (C^{lr}) is generated using the training images. On the other hand, the left-reconstruction Left-Rec-Right Correlation ($C^{\hat{l}r}$) and the right-econstruction Left-Right-Rec Correlation are also generated ($C^{l\hat{r}}$) based on the source and reconstructed images. Then we calculate and minimize the errors of (C^{lr}, $C^{\hat{l}r}$)and (C^{lr}, $C^{l\hat{r}}$) pairs.

3.3 Objective Function

Although the U-net is not designed to learn the disparity, a well-designed loss function can help this network estimate the disparity successfully. There are four scales of the prediction output, and the corresponding loss C_s at each output scale is defined to form a total loss as $C = \sum_{s=1}^{4} C_s$. Following (7) and (27), Cs is a weighted sum of three terms: appearance (C_{ap}), disparity smoothness (C_{ds}), and left-right correlation (C_{cor}).

$$C_s = \alpha_{ap} \times C_{ap} + \alpha_{ds} \times C_{ds} + \alpha_{cor} \times C_{cor} \quad (2)$$

The proposed loss in Eq. 2 is a convex combination. Each term in the objective function is in the range [0,1] and consists of the left and right loss, e.g. left and right appearance loss (C_{ap}^l, C_{ap}^r), because the network is designed to predict the left and right disparities simultaneously. The weights $(\alpha_{ap}, \alpha_{ds}, \alpha_{cor})$ are determined during the optimization process.

In the first two terms, we follow the approach presented in (27) and make a change. The appearance loss (C_{ap}) is a combination of the L_1 difference and single scale SSIM[28], which is a method for evaluating the perceived image quality, between the training image pair *L*, *R* and the reconstructed image pair $\hat{L}$, $\hat{R}$ respectively.

$$C_{ap}^l = \frac{1}{N}\sum_{i,j} \alpha_{ssim} \frac{1 - SSIM\left(I_{ij}^l, \hat{I}_{ij}^l\right)}{2} + (1 - \alpha_{ssim})\left|I_{ij}^l - \hat{I}_{ij}^l\right| \quad (3)$$

The combination weights of the L_1 and SSIM is 0.15 and 0.85 (α_{ssim} = 0.85), and SSIM is designed based on a simplified 3×3 block filter. Disparity smoothness (C_{ds}) is the term that encourages smoothness of the disparity with a L_1 difference on the disparity gradients.

$$C_{ds}^l = \frac{1}{N}\sum_{i,j} \left|\partial_x d_{i,j}^l\right| e^{-\left|\partial_x I_{ij}^l\right|} + \left|\partial_y d_{i,j}^l\right| e^{-\left|\partial_y I_{ij}^l\right|} \quad (4)$$

This term is edge-aware and set for the depth discontinuities, which often occur at image gradients ∂I.

The last term, left-right correlation (C_{cor}), forces the network to learn the image correlation in the training process. We introduce the *L1* between the training and reproduction image pairs,

$$C_{cor} = \sum_k \left|C_k^{lr} - C_k^{\hat{l}r}\right| + \left|C_k^{lr} - C_k^{l\hat{r}}\right| \quad (5)$$

where k is the layer of the correlation layers in the displacement range *D*. The reconstructed left image is calibrated by the training image pair using the absolute correlation difference in the first term, while the reconstructed right image repeats the same process in the second term. The total correlation loss is composed of the correlation-calibration information of the reconstruction left and right images and network can learn this correlation to detect the object depth more accurate. To reduce the computational load, we size-down the images by the down-scaling factor f_{ds}.

4. Experiments

In this section, the implementation details are explained then the proposed method is compared with the state-of-the-art in either supervised, semi-supervised, or unsupervised monocular methods of the single view and single network. The benchmark is based on KITTI[29], and Make3D[30] datasets in quantitative and qualitative results. Additionally, we also demonstrate the qualitative results of

the Cityscapes[31] dataset. The ablation is then analyzed, followed by the discussion of the limitation and potential application.

Configuration Our algorithms were implemented in Tensorflow[32] using Python 3.5 under the Ubuntu environment with a single NVIDIA GTX 1080 GPU. All input images are resized to a 640_192 from the original size of the training image and the trainable weights are around 17.5 million using the proposed ResVGG encoder backbone in the network. Training takes around 7.5 hours by 8 batches and 20 epochs on the KITTI training dataset. The predictions happen in around 30 milliseconds (around 32 frames per second (FPS), which is fast and feasible for real-time applications.

Parameter Settings The parameters of the correlation layer are defined as follows: To reduce the computational cost, we define the down-scaling factor $f_{ds} = 8$ to size down the 640×192 input image of the network to 80×24. Then we assume that the maximum disparity d_{max} of the image is set as d_{max} = 0.3×image width based on the stereo camera configuration of the KITTI dataset. Based on this constraint, we chose the parameters $k = 1$, $d = 20$, $s1 = 1$, $s2 = 2$ for the correlation layer in the training loss function. Although the displacement *disp* is defined as 20, the effective search range is 40 by the $s2 = 2$. Then, the displacement range is $D = 41$. Furthermore, the weights in the objective function are determined during the optimization process. The α_{ap} and α_{cor} are both set as 1, and the α_{ds} is related to the output scale and defined as $0.5/r$, where r is the down-scaling factor of each layer in respect to the resolution of the input image. We use the Adam optimizer in Tensorflow[32]. The total training flow involves a batch size 8 and 20 epochs. In our design, the training converges after 15 epochs and the improvement after 20 epochs is minor. Our performance is competitive with the reference methods even the training epochs are much less. We start from the initial learning rate of $\lambda = 10^{-4}$ and decrease λ by half every ten epochs. During the training, the dataset is randomly augmented, including horizontal flipping, color/gamma/brightness adjustment by 50% chance.

Network Backbone In the evaluation, we show the simulation results in different kinds of backbones, including VGG-16, Resnet50, and proposed ResVGG, in U-Net. The computation costs of training and predicting process in each backbone are also summarized in the Ablation section. We show that the proposed ResVGG structure has competitive performance better than Resnet50 but less computational costs than VGG-16.

Post-Processing We use a post-processing method to reduce the single-side halo effect designed by (7) in the last step. The visual artifacts, including the halo effect and broken shape, are largely reduced.

5. Results

We evaluate the performance of the proposed method on the KITTI benchmark. We use two different test splits, KITTI and Eigen Split, of KITTI dataset to do the ablation analysis of the variants of our method and the benchmark compared with the existing works. The KITTI dataset contains 42,382 rectified stereo pairs in raw form from 61 scenes. The typical image size of the KITTI dataset is 1242×375 pixels and the stereo image pairs are well calibrated in the calibrated camera configuration. The depth labels have been collected from a Velodyne laser sensor and in the form of sparse 3D laser measurement. The parameter of the stereo setup is clearly defined in the dataset as well so the predicted disparities can be converted back to the corresponding depth. The evaluation metrics are from (24), which measure error in meters from the ground truth and the percentage of depth that are within some threshold from the correct value. We can expect that the error terms present the average error and lower value is better; the percentage of the correct value is an alias of accuracy and higher valuer is better.

5.1 Ablation Study

First, various design choices are evaluated to prove the improvement upon the proposed the training loss and network architecture. We first consider the baseline design using VGG-16 backbone and no correlation layer. Then results of the design of backbones with the correlation layer followed. All the models are trained from the scratch on KITTI training dataset using KITTI split and tested on the KITTI stereo 2015 test dataset. The quantitative results and the computational cost analysis are shown in Tables 2 and 3. First, the design with the correlation layer outperforms the baseline. However, the training time increase 10%. Then, the Resnet50 backbone largely improves the performances but also needs significant computational resources. The proposed ResVGG backbone has results close to Resnet50 and needs much less parameters than Resnet50 even VGG. The prediction frame rate of ResVGG remains over 33 FPS and is fast enough for real-time applications.

Table 2. Quantitative results of different variants of our approach on the KITTI Stereo 2015 test dataset. Best results are shown in bold, and seconds are in italic. The first row is the baseline design using VGG-16 backbone and no correlation layer; the following three rows are VGG-16/Resnet50/ResVGG with proposed correlation layer. The proposed correlational training object can effectively improve the performance, and the proposed ResVGG has very close performance as Resnet50.

Approach	ARD	SRD	RMSE	$RMSE_{(log)}$	δ<1.25	δ<1.25^2	δ<1.25^3
	Lower is better				Higher is better		
No Cor / VGG	0.1259	1.3718	6.341	0.227	0.828	0.93	0.971
Cor / VGG	0.119	1.0991	5.694	0.215	0.837	0.939	0.975
Cor / Resnet50	**0.1146**	**1.0065**	**5.46**	**0.203**	**0.848**	**0.945**	**0.98**
Cor / ResVGG	*0.1165*	*1.0371*	*5.535*	*0.206*	*0.844*	*0.944*	*0.978*

Table 3. Computational costs of different variants of our approach on the KITTI training dataset within 8 batches and 20 epochs. The unit of training time is hour and of prediction is FPS. Best results shown in bold, second best in italic.

Approach	Parameters	Train (hours)	Predict (FPS)
No_Cor_VGG	19806248	**7**	**36.5**
Cor_VGG	*19806248*	7.8	**36.5**
Cor_Resnet50	58452008	9.3	23.4
Cor_ResVGG	**17462824**	*7.2*	*33.4*

5.2 State-of-the-art comparison

In this section, the proposed method is compared to existing supervised and unsupervised works. The quantitative results are shown in Table. 4 and the qualitative samples are shown in Fig. 4. In the comparison, we use the Eigen test split[(2)] of 697 images which covers a total of 29 scenes. To fairly compare all the methods, we evaluate at the input image resolution and use the same crop as (2). The simulation results are given by (5, 7, 8) and we rerun the given models to get the results in an equivalent training condition, which is in a batch size 8 and 20 epochs. Our method of being pre-trained on the Cityscapes dataset has a competitive result comparing to the current state-of-the-arts but needs much less computation cost. Furthermore, our method of the low-resolution input image is still competitive among other methods. In the qualitative examination, the prediction examples in Fig. 4 also show that our method can better estimate the depth map.

5.3 Limitations and Potentials

Although the correlation layer in the objective function can successfully improve the image depth clarity, the proposed method still has some limitations in identifying the depth of the low-contrast object. The features of the low-contrast object are much less significant in the prediction. Without actual stereo images as the input, the low-contrast object is less likely to be predicted well. The possible solution is to involve the per-pixel minimum error (16) to optimize the error regression process in the low difference area.

On the other hand, the proposed method can potentially be generalized to the visual odometry based depth estimation because the correlation layer is originally designed to estimate the optical flow. Furthermore, the proposed ResVGG network architecture can be applied to other fields, such as object detection, image segmentation, etc., which originally use Resnet50 as the base network architecture.

6. Conclusions

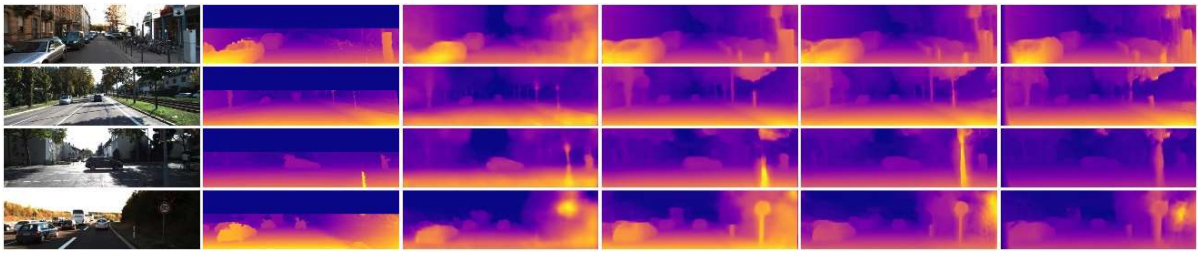
A novel model is presented in this work to deal with the self-supervised monocular depth estimation problem from a single image. We proposed a new loss function to enable our network to learn the stereo-image correlation and a new network architecture to reduce the computational costs. Our method is competitive comparing to the current state-of-the-art in quantitative and qualitative results with the much less computation cost. We demonstrate that our model is robust to the low-resolution input image and can still keep high quality output with 33 FPS in prediction under a single consumer GPU. Our future work is to extend our method to consider video sequence and support visual odeometry based depth estimation in real-time.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Number 1622589

Table 4. Quantitative results comparing with current state-of-the-art on KITTI eigen split. The prefix * indicates that the ground truth is needed in this method. The first half results are in the full distance (0-80m); while the second half results are in the near distance (1-50m). Best results are marked in bold. Our method is competitive comparing to the reference methods in the benchmark. Our low-resolution version is only for reference in the comparison, but it is still competitive among all the other methods.

Approach	ARD	SRD	RMSE	$RMSE_{(log)}$	δ<1.25	δ<1.25^2	δ<1.25^3
	Lower is better				Higher is better		
*Eigen et al.[2] Coarse	0.214	1.605	6.563	0.292	0.673	0.884	0.957
*Eigen et al.[2] Fine	0.203	1.548	6.307	0.282	0.702	0.89	0.958
*Yevhen et al. [8]	0.2645	1.859	5.902	0.28	0.632	0.916	0.97
Godard et al.[7] C+K	0.1648	3.5323	6.898	0.278	0.846	0.927	0.959
Godard et al.[7] pp C+K	0.1143	1.0651	5.031	0.216	**0.859**	**0.94**	0.971
Ours	0.1724	3.6001	7.071	0.284	0.829	0.92	0.957
Ours C+K	0.1643	3.4488	6.858	0.278	0.841	0.925	0.959
Ours pp C+K	**0.112**	**0.9524**	**4.927**	**0.212**	0.856	**0.94**	**0.972**
Ours pp C+K 128	0.115	1.046	5.17	0.217	0.849	0.938	0.971
*Yevhen et al.[8] cap 50m	0.2622	1.6213	4.962	0.274	0.638	0.92	0.972
Garg et al.[5] cap 50m	0.169	1.08	5.104	0.273	0.74	0.904	0.962
Godard et al.[7] C+K cap 50m	0.1478	2.0366	4.998	0.258	0.857	0.933	0.963
Godard et al.[7] C+K pp cap 50m	0.1078	0.7704	3.801	0.204	**0.87**	**0.946**	0.974
Ours cap 50m	0.1551	2.0865	5.149	0.264	0.841	0.927	0.961
Ours C+K cap 50m	0.1479	1.9966	5.005	0.258	0.853	0.931	0.962
Ours pp C+K cap 50m	**0.1062**	**0.7068**	**3.762**	**0.201**	0.868	**0.946**	**0.975**
Ours pp C+K 128 cap	0.1083	0.7587	3.911	0.204	0.861	0.945	**0.975**



Input Ground Truth Yevhen et al.[8] Godard et al.[7] Ours Ours 128

Fig. 4. Figure 4: Benchmark of qualitative result on KITTI dataset Eigen Split. We interpolate the sparse ground truth velodyne depth for visualization. Our method can capture more fine objects and maintain their shapes better. Even our low-resolution input version (Ours 128) can still perform well.

"Computer Guided Laparoscopy Training". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

(1) L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In Conference on Computer Vision and Pattern Recognition, 2014.

(2) D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in Neural Information Processing Systems, 2014.

(3) F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10):2024–2039, 2016.

(4) M. Jaderberg, K. Simonyan, A. Zisserman, and koray kavukcuoglu. Spatial transformer networks. In Advances in neural information processing systems, 2015.

(5) R. Garg, V. Kumar, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In European Conference on Computer Vision, 2016.

(6) K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Conference on Computer Vision and Pattern Recognition, 2016.

(7) C. Godard, O. M. Aodha, and G. Brostow. Unsupervised monocular depth estimation with left-right consistency. In Conference on Computer Vision and Pattern Recognition, 2017.

(8) Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In Conference on Computer Vision and Pattern Recognition, 2017.

(9) J. Xie, R. Girshick, and A. Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In European Conference on Computer Vision, 2016.

(10) J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research, 2016.

(11) E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.

(12) 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 Conference on Computer Vision and Pattern Recognition, 2016.

(13) L. Iro, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In International Conference on 3D Vision, pages 239–248, 2016.

(14) X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In IEEE Conference on Computer Vision and Pattern Recognition, 2015.

(15) Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. In IEEE Transactions on Circuits and Systems for Video Technology, volume 28, pages 3174–3182, 2018. 2

(16) C. Godard, O. M. Aodha, and G. Brostow. Digging into self-supervised monocular depth estimation. In arXiv:1806, 2018.

(17) T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In IEEE Conference on Computer Vision and Pattern Recognition, 2017.

(18) R. Li, S. Wang, Z. Long, and D. Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In IEEE International Conference on Robotics and Automation, 2018.

(19) Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin. Single view stereo matching. In Conference on Computer Vision and Pattern Recognition, 2018.

(20) H. Zhan, R. Garg, C. S.Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In IEEE Conference on CVPR, 2018. 2

(21) C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In IEEE Conference on Computer Vision and Pattern Recognition, 2018.

(22) R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In IEEE Conference on Computer Vision and Pattern Recognition, 2018.

(23) A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, 2015.

(24) D. Eigen and R. Fergus. Predicting depth, surface normal and semantic labels with a common multi-scale convolutional architecture. In IEEE international conference on computer vision, 2015.

(25) K. Simonyan and A. Zisserman. A very deep convolutional networks for large-scale image recognition. In arXiv:1409, 2014.

(26) R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, 2003.

(27) Z. Hang, O. Gallo, I. Frosio, , and J. Kautz. Loss functions for image restoration with neural networks. IEEE Transactions on Computational Imaging 3, (1):47–57, 2004.

(28) W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, (4):600–612, 2004.

(29) A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In Conference on Computer Vision and Pattern Recognition, 2012.
(30) A. Saxena, M. Sun, and A. Ng. Make3d: Learning 3d scene structure from a single still image. IEEE Trans Pattern Anal Mach Intell, 31(5):824–840, 2009.
(31) 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.
(32) M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.