Stereo disparity estimation using the Argoverse stereo dataset and a dense synthetic stereo dataset collected from CARLA simulator

Luis Rosero and Fernando Osório
Institute of Mathematics and Computer Science, São Carlos, São Paulo, Brazil
University of São Paulo
{lrosero, fosorio}@usp.br

***Abstract*— Recent research has shown that stereo disparity estimation can be formulated as a supervised learning task, and many approaches use deep learning to solve this problem with impressive performance on most of the standard benchmarks. However, creating training datasets with a dense ground-truth in natural outdoor environments is still very challenging. The Argoverse team has released a dataset with a sparse ground-truth for training disparity algorithms and organized challenges. In this report, we present our solution for the 2022 Argoverse Stereo Estimation Challenge. Our approach uses a synthetic dataset extracted from the CARLA simulator for training, and then we perform fine-tuning on the Argoverse stereo dataset. Final tests are performed on the challenge leaderboard. We got third place and honorable a mention.**

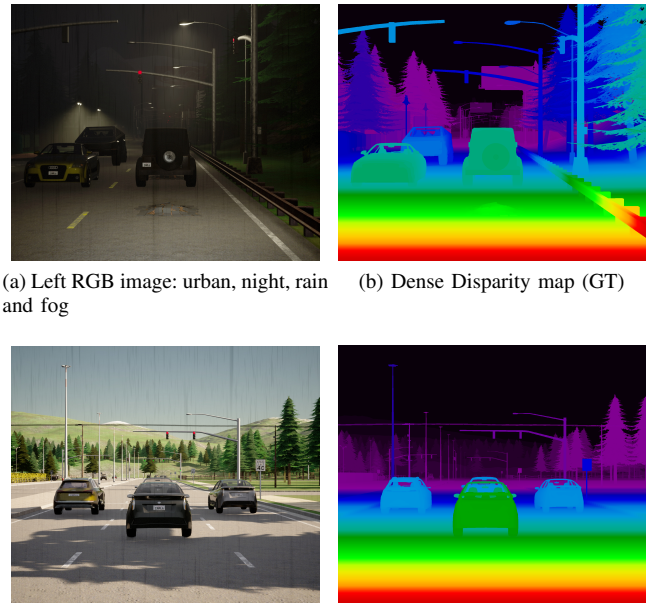
I. Introduction

Stereo matching aims to estimate the disparity map between a pair of rectified images. Disparity refers to the horizontal distance between a pair of corresponding pixels in the left and right images. The disparity ($disparity$) of a pixel can be converted to depth (Z) by $Z = Bf/disparity$. In this way, the accuracy of the depth improves with the prediction of the disparity. Disparity estimation is an essential task for computer vision applications, such as autonomous driving, 3D reconstruction, and robot navigation.

Significant progress of deep learning in the field of computer vision has been also extended to geometric problems such as stereo matching. Currently, deep learning based approaches have reached state of the art performance in most stereo disparity benchmarks, such as Middlebury, KITTI, ETH3D, and Argoverse.

However, scenarios of the real world not only require state of the art algorithms but also real time inference, and domain adaptation. Argoverse has released a stereo dataset in different lighting and weather conditions containing images at ten times the resolution relative to images in the KITTI dataset, and 16 times as many training frames, making it a much larger and more challenging dataset. The ground-truth depth is derived from LiDAR point cloud accumulation. The Argoverse team has organized the 2022 stereo challenge to motivate researchers to test algorithms that work well, run fast, and generalize to new scenes at the same time in the Argoverse stereo dataset.

For this challenge we chose PWC-Net, a CNN (Convolutional Neural Network) that achieves a good balance between good results in stereo disparity prediction and inference speed named. For training we create a dense synthetic stereo dataset from CARLA simulator [3]. Then, we fine-tune the model using the Argoverse stereo dataset. For evaluation, metrics from the KITTI stereo challenge are adopted.



(a) Left RGB image: urban, night, rain and fog

(b) Dense Disparity map (GT)

(c) Left RGB image: highway, day and rain

(d) Dense Disparity map (GT)

Fig. 1. Dense synthetic stereo dataset generated using the CARLA simulator.

II. Synthetic Dataset

A dense synthetic stereo dataset was created using the CARLA simulator. We configure a simulated agent using a setup similar to the one used in the Argoverse stereo dataset for data collection. We use two RGB cameras at a height of 1.7m on the roof with a baseline (distance between the two cameras) of 0.2986 m. Each camera has a FOV=30 deg, the images have a size of 2464×2056 pixels. In the same pose as the left camera we set a depth camera that provides raw data of the scene encoding the distance of each pixel to the camera (also known as depth buffer or Z-buffer) to create a depth map of the elements in the scene. To create our dense stereo dataset we use the depth (Z) values and the baseline value in pixels to create disparity maps for each left RGB

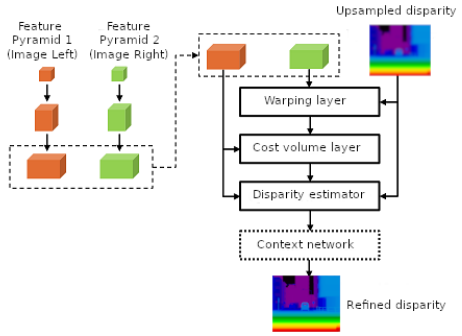


Fig. 2. PWC-Net for disparity estimation. Adapted from [7]

image as shown below.

$$disparity = \frac{Bf}{Z} \tag{1}$$

Where B is the baseline and f is the focal length of the camera (calculated from FOV), Z is depth. We collected approximately one hundred thousand frames.

Our dataset includes urban, residential and highway environments, as well as different weather and lighting conditions including rain, fog, day, night, etc. Figure 1 shows two examples of left RGB images and their respective dense disparity map captured from the CARLA simulator.

III. PWC-Net for stereo disparity estimation

According to the authors, PWC-Net is a CNN model for optical flow that has been designed according to simple and well-established principles: pyramidal processing, warping, and the use of a cost volume. Cast in a learnable feature pyramid, PWC-Net uses the current optical flow estimate to warp the CNN features of the second image. It then uses the warped features and features of the first image to construct a cost volume, which is processed by a CNN to estimate the optical flow [7]. PWC-Net outperforms optical flow methods on the MPI Sintel and KITTI 2015 benchmarks, running at about 35 fps on Sintel resolution (1024×436) images.

Considering two rectified RGB images coming from a calibrated stereo camera, we can have an epipolar line across the two images and the stereo matching is the pixel correspondence between the two images along the epipolar line in the horizontal direction on the x-axis. Thus, for this challenge we consider stereo matching as a special case of optical flow where disparities between the stereo pair can be modeled as optical flow on the x-axis of the image. So we can use models used for optical flow (x and y coordinates) to solve the disparity problem as optical flow only at the x coordinate along the epipolar line between the left and right image.

Figure 2 summarizes the key components of PWC-Net adapted for stereo estimation. Feature pyramid 1 and feature pyramid 2 correspond to learnable feature pyramids from a feature pyramid extractor feeded with the left and right RGB rectified images. A warping operation from the traditional optical flow approach is used as a layer in the network to estimate large motion. PWC-Net has a layer to construct a cost volume, which is then processed by CNN layers to estimate the flow (disparity). The warping and cost volume layers have no learnable parameters and reduce the model size. Finally PWC-Net uses a context network to exploit contextual information and refine the disparity.

IV. Implementation

Our adaptation of PWC-Net for disparity estimation is implemented in the MMFlow framework [1]. MMFlow is an open source pytorch based toolbox that is a part of the OpenMMLab project. MMFlow is the first toolbox that provides a framework for unified implementation and evaluation of optical flow algorithms.

V. Training

We use the same parameters used for training in [7] and the same loss proposed in FlowNet [2]. We use a search range of 4 pixels to compute the cost volume at each pyramid level. We first train the model using our synthetic stereo dataset using the S_{long} learning rate schedule introduced in [4], Starting from 0.0001 and reducing the learning rate by half at 0.4M, 0.6M, 0.8M, and 1M iterations. Finally, we fine-tune the model using Argoverse stereo dataset using the S_{fine} schedule [4]. Batch size 4 was used for all the training process.

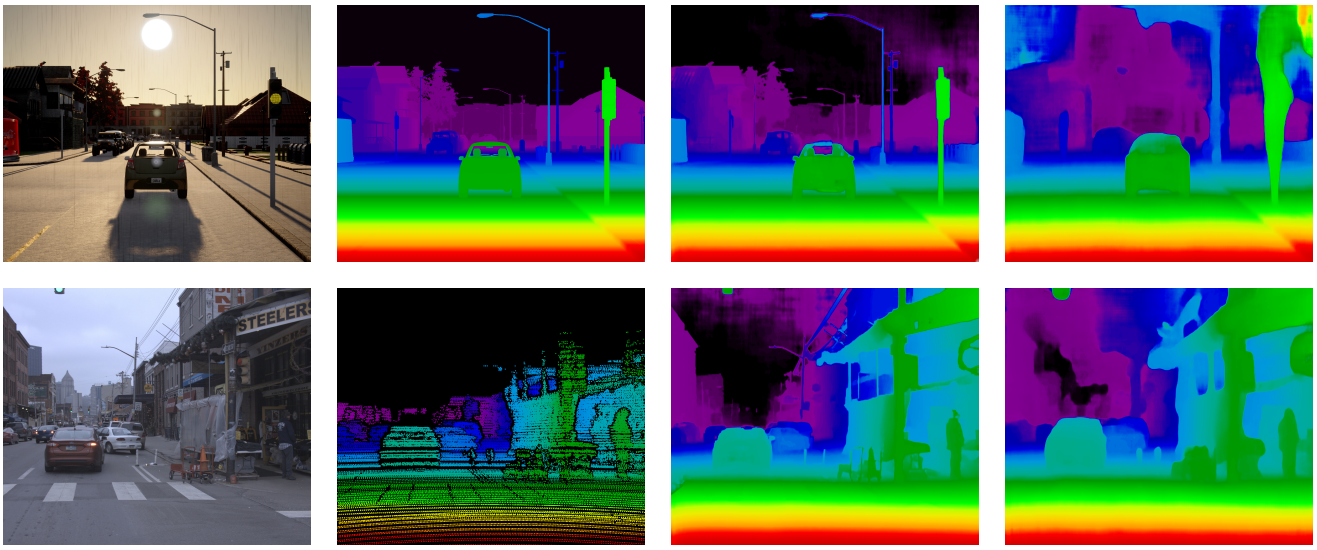
For data augmentation, we use a random crop (768 x 2432 patches). Inference is performed in full resolution.

VI. Results

Figure 3 shows two frames: one for our synthetic dataset and the other taken from the validation set of the Argoverse stereo dataset. Figure 3a shows the left RGB image, Figure 3b represents the ground-truth (dense for our dataset and sparse for Argoverse stereo dataset) and then disparity results: first we show inference results for a PWC-Net model trained only on our synthetic dataset (Figure 3c) and finally inference results for the same model but with fine-tuning performed on the Argoverse stereo dataset (Figure 3d).

Note that the ground-truth disparity released together with the Argoverse stereo dataset is sparse (Figure 3b below) and many pixels in the background and foreground do not have ground-truth, mainly in the upper parts of the image, for example: the tops of buildings, traffic lights very close to the camera and lamps. However, PWC-Net trained only on our dataset correctly calculates disparity in the upper parts (Figure 3c above and below). This is because the disparity ground-truth of our dataset is dense and available at training time for all pixels in the image. In the same way as testing on the synthetic dataset, inference on the Argoverse stereo dataset (Figure 3d) also benefited from prior training on the synthetic dataset. Disparity in the upper parts in the Argoverse dataset is correctly estimated.

An important result of training on our synthetic stereo dataset is domain adaptation. The model trained only using synthetic data obtains very satisfactory results on the test set of the Argoverse as shown in Figure 3c. In these figures



(a) Left RGB image (b) Ground-Truth (c) Inference: PWC-Net trained only using our CARLA stereo dataset (d) Inference: PWC-Net trained using our CARLA stereo dataset + fine-tuning in Agoverse stereo dataset.

Fig. 3. Comparison between two models (PWC-Net) tested in our synthetic stereo dataset (first row) and Argoverse stereo dataset (second row). The first column is the left RGB image, and the second column is the ground-truth. The firs model is trained only using our CARLA stereo dataset (inference results on third column) and the second model is the same model with further fine-tuning in Argoverse stereo dataset (last column).

we can see that the shape of the objects and their edges are well defined and thin objects are correctly differentiated, for example, we can clearly see power cables and other thin objects, while when fine-tuning is performed on the sparse dataset these details tend to disappear.

Argoverse Stereo Competition server computes the percentage of bad pixels averaged over all ground-truth pixels, similar to the KITTI Stereo 2015 benchmark [5] [6] for all 1,094 test disparities from 15 log sequences.

The disparity of a pixel is considered to be correctly estimated if the absolute disparity error is less than a threshold or its relative error is less than 10% of its true value. Three disparity error thresholds are defined: 3, 5, and 10 pixels. The leaderboard ranks all methods according to the number of bad pixels using a 10 pixels threshold (all:10 is the main metric). We compare our results in Table I using all:10, fg:10, and bg:10.

Acording to the online leaderboard[1], as shown in Table I the overall ten-pixel-error (all:10) for the PWC-Net is 2.47, we occupy the third place for all:10 and bg:10 metrics and second place for fg:10 metric. We surpassed by a wide margin the results of [8]. For the 2022 Argoverse stereo competition, methods that run in real time are desired and algorithms must run faster than 200 ms per disparity prediction (during forward pass). We achieve an average inference time of 191.60 ms on an Nvidia GeForce RTX 2080Ti GPU.

TABLE I

CARLA LEADERBOARD RANKING

Participant team	all:10	fg:10	bg:10
GMStereo	1.61	1.71	1.56
MSCLab (DEQ Stereo)	2.39	3.34	2.01
LRM (Our entry)	2.47	2.67	2.38
Odepth	3.78	4.57	3.46
ACVNet (Baseline) [8]	4.06	7.77	2.54

VII. CONCLUSIONS

We created a dense synthetic stereo dataset collected from the CARLA simulator, and we trained PWC-Net using this synthetic dataset, and we obtained good disparity estimation on test images from both our dataset and the Argoverse stereo dataset. Subsequently, fine tuning is performed on the target dataset (Argoverse stereo dataset). The estimated disparity maps achieve very good results in the Argoverse evaluation server and also demonstrate that pretraining in our synthetic stereo dataset improves disparity estimation in all regions of the image.

[1] https://eval.ai/web/challenges/challenge-page/1704/leaderboard/4066

ACKNOWLEDGMENT

The authors acknowledge the financial support provided by Programa Rota2030 Linha V (FUNDEP) - Projeto SegurAuto

REFERENCES

[1] MMFlow Contributors. MMFlow: Openmmlab optical flow toolbox and benchmark. https://github.com/open-mmlab/mmflow, 2021.

[2] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[4] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.

[5] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[6] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 2018.

[7] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[8] Gangwei Xu, Junda Cheng, Peng Guo, and Xin Yang. Attention concatenation volume for accurate and efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12981–12990, 2022.