Disentangling Object Motion for Self-supervised Depth Estimation

Jinxi Xiao, Kecheng Ye, Yi'ang Ju, Panfeng Jiang, Haoyu Wu
ShanghaiTech University

Abstract

Estimating monocular depth via unsupervised learning has emerged as a promising approach in many fields. However, one of the main difficulties would be occlusions caused by the motion of dynamic objects within the monocular inputs. In order to solve this problem, we propose some improvements based on established ideas.

1. Introduction

Depth is vital when constructing local 3D environment, but it is definitely not an easy task to obtain this information. Traditional methods use a stereo pair to capture both the left and right image of the same scene, followed by a procedure to compute the disparities of this certain image pair and retrieve depth information. With the emergence of deep learning and the urge to reduce the number of used sensors, people then begin to come up with the idea that only use a monocular to estimate depth within a self-supervised fashion.

Temporal and spatially continuous images are available in most real-world scenarios, and due to the self-supervised fashion it is easy to obtain inputs of the system, compared with many other methods. However, some general algorithms and designs, like re-projection loss and cost volume construction are all based on the assumption that the whole scene is static. But in fact there are flow of motions of objects in the sequence-input images, this case is shown in Figure 1

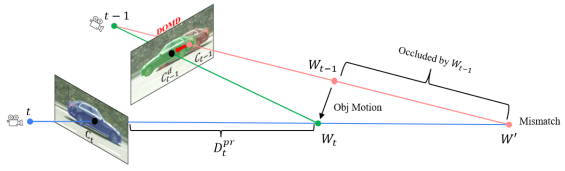

Figure 1. Dynamic Object Motion Disentanglement

The motion disentanglement will greatly affect the re-projection loss, since the unmatched area will mislead the back propagation of the gradient and further lead the system to a wrong direction. Thus, it is of great importance to find the dynamic objects and exclude them from the loss computation phase.

Thus, based on the this situation, and in order to overall improve the whole system, we propose the following improvements on the framework presented in Dynamic-Depth [1]:

- We introduce two loss functions to the Dynamic-Depth framework: L1 normal loss and normal ranking loss.
- Introduce pose cycle consistency constraint to enhance the ability of Pose Net to predict poses.
- Create a weighted scheme based on masks. Automatically adjust the weight of each datum according to the size of its dynamic object mask.
- Add a Flow Net into the framework of Dynamic-Depth, which predicts the optical flow of two images.

2. Related Work

Early works like [2] used a self-supervised CNN network to estimate depth information in a stereo system. Then monodepth2 [3] proposed a general architecture that is widely used in almost all latter papers. Dynamic-Depth [1] and SC-Depth [4] both used a prior monocular depth estimation system to help produce a more accurate result, while the former added a DOMD model to mask all dynamic objects and the latter one focused on geometry features like surface normal. On the other hand, DS-Depth [5] used a *Flow Net* to predict optical flow and thus able to predict the dynamic components in the image-sequence.

3. Methods

The general pipeline of our work is presented in Figure 2. And in the following sections we will explain some key components in the pipeline.

3.1. Pose Cycle Consistency

Pose network is an important part in self-supervised depth estimation, the network is trained to predict the relative poses between adjacent frames which is later used in

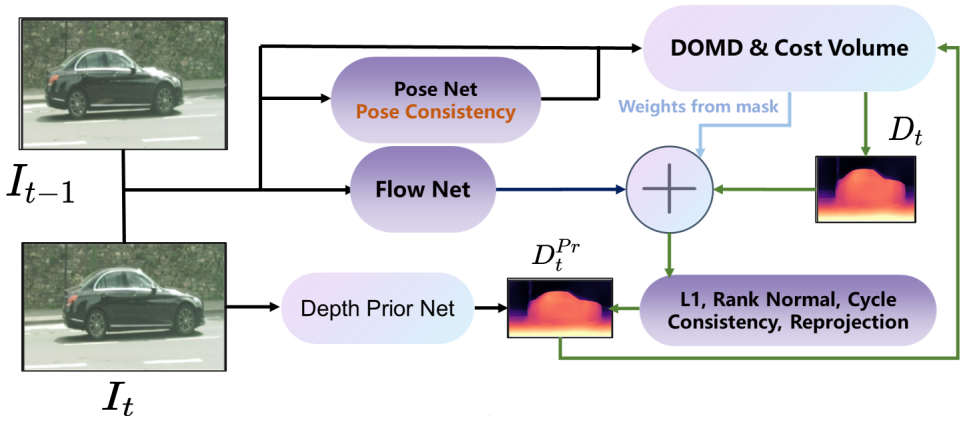

Figure 2. The total framework of our design. Based on Dynamic-Depth, we first introduce a pose cycle consistency constraint on Pose Net to help predict a better pose motion. Besides, we give each instance a weight based on the size of its dynamic object mask, thus we can avoid dynamic situation to some extent. Moreover, a Flow Net is add to predict a flow on back-projected 3D point cloud to correct the bias caused by object movement. At last, we use L1, Rank normal loss to learn better depth. Though not all designs works well, there are designs demonstrate remarkable improvements.

project function. Since the projection with relative pose is a hard association in a sequential process from predicted depth to the reconstructed image, the accuracy of pose net is highly required. Thus we introduce pose cycle consistency, a constraint to improve the performance of the pose net. The idea based on the fact that the matrix multiplication of translation matrix that from A to B and that from B to A should be a identity matrix. Specifically, the origin method input frame combinations that are [0,1],[-1,0] and use the inverse matrix to get the pose motion of [0,-1]. In our method, we input with all four possible combinations, though only [0,1],[0,-1] are used later for projection, other two are use to calculate the errors with regard to the translation cycle. We multiply the predicted result pairly and compare them with the identity matrix, we use L1-norm to turn the difference to loss.

$$L_{pose} = \sum |T_{t-1\to t} T_{t\to t-1} - I_4| \tag{1}$$

3.2. Auto weight from Dynamic Object mask

Dynamic object is always a problem in depth estimation. In Dynamic-Depth, they use a DOMD module to alleviate the error during the motion of camera and object. In the process, they use the a pretrained network to segment dynamic positions such as human and car, which we think that can be a measurement of how much is the dynamic part in a single image. So, we may have two guesses: 1. What if we increase the weight of images with less dynamic object like avoiding occlusions, to make the model perform better on standard statistic scenes. 2. What if we increase the weight of images with more dynamic object to make the model understand and adapt to dynamic problems, which may achieve better estimation. After experiment, which is show in Section 4, we found that increase the weight of images with less dynamic object is better and better than Dynamic-Depth. Specifically, we count the size of the mask of the dynamic object and get the weight simply by dividing the size of the image which is 192×512 and get the weight range from 0 to 1.

$$w_i = 1 - [count(M_i)/(192 \times 512)] \tag{2}$$

where w_i is the weight for the i-th instance, M_i is the dynamic object mask of the i-th instance(image). After calculating the re-projection loss for each instance, we apply corresponding weight and merge the loss in a batch.

3.3. Flow Net

Dynamic-Depth use DOMD to deal with dynamic object based on calibration matrix and projection, which is hard association. However, due to inaccuracy of the predicted relative poses and poor image condition which may caused by lighting, a hard association may lead to mistakes and accumulate errors. So, we add a Flow Net which estimate 3D flow to adjust the point cloud after the image and predicted depth is back-projected. Specifically, the net input images from two frames and output a vector which represent 3D flow for each pixel.

$$FlowNet(I_{t-1}, I_t) = F \tag{3}$$

where $I_{t-1}, I_t \in R^{3\times W\times H}$ are input images, and $F \in R^{W\times H\times 3}$ is the estimated flow. Through this, we have a two step correction for dynamic object:1.DOMD by Dynamic-Depth that is hard associated by projection formulas and in 2D. 2.Our Flow Net that if soft associated(neural network) and in 3D. The combination corrects dynamic problem from different angles.

Methods	abs_rel	sq_rel	rms	log_rms	a1	a2	a3
L1 normal	0.1134	1.1123	6.1344	0.1670	0.8730	0.9689	0.9901
L1 + ranking	0.1125	1.1128	6.2354	0.1673	0.8751	0.9694	0.9904
L1 + ranking(weighted)	0.1075	1.0613	6.1773	0.1634	0.8804	0.9707	0.9909
Pose cycle consistancy	0.1066	1.0660	6.0124	0.1599	0.8893	0.9724	0.9910
Auto doj mask weight(more dynamic less loss)	0.1056	1.0620	6.0105	0.1595	0.8898	0.9725	0.9911
Auto doj mask weight(more dynamic more loss)	0.1088	1.1310	6.0790	0.1624	0.8858	0.9712	0.9901
Flow Net	0.1091	1.1080	6.0390	0.1635	0.8842	0.9707	0.9905
Dynamic-Depth	0.1073	1.0856	6.0305	0.1605	0.8875	0.9720	0.9910

Table 1. These are experiments on our designs. Dynamic-Depth is our baseline. Two of our designs shows advancement: Auto dynamic object mask weight and Pose cycle consistency.

3.4. Normal matching loss

In our experiments, we find that the depth prior has the potential of generating good depth smoothness. In this section, we propose to leverage such attributes to regularize the self-supervised depth. Our idea is to: (i) constrain the surface normals that are derived from predicted depths and prior depth to be matched; and (ii) constrain two depth maps to be consistent w.r.t. relative normal angles of sampled point pairs around edges. The loss is computed using a library called Kornia, who provides a method to convert depth map into normals per pixel using a depth map and camera intrinsic matrix.
The l1 loss(with weight of 0.1 in practical) is as follows:

$$L_N = \frac{1}{N}\sum_{i=1}^{N} ||n_i - n_i^*||_1 \tag{4}$$

where n_i is the surface normal derived from the predicted depth and n_i^* is the normal derived from depth prior. N stands for the total number of pixels in the image.
The edge-aware relative normal loss(with weight of 0.1 in practical) is as follows:

$$L_{ERN} = \frac{1}{N}\sum_{i=1}^{N} ||n_{Ai} \cdot n_{Bi} - n_{Ai}^* \cdot n_{Bi}^*||_1 \tag{5}$$

where n_A denotes the normal of a sample point from the predicted depth, and * denotes depth prior. Combining the edge-guided sampling and relative normal loss, we can efficiently constrain the depth estimation on object boundary regions.

4. Experiments

We mainly test our models and methods on *CityScape Dataset* [6]

After adding pose cycle consistency constraint on pose net, as shown in Table 1, the model preforms better on all metrics, it is because the constraint forces the pose net to predict pose reasonably. Besides, we add a weight for each instance based on size of its dynamic object mask. As is described in Section 3.2, we have two guesses that more dynamic object in an image should lead to more loss or lead to less loss. After experiments, we found that give image that have more dynamic object less loss weight is better, we think that our design help the model focus on the standard static depth estimation. However, it also shows the fact that, even after dynamic correction like DOMD, the modified images still have a large gap toward standard static condition and remain a stubborn problem in depth estimation.

Additionally, we introduce Flow Net to modify the 3D point cloud to improve robustness towards the inaccuracy from predicted poses which was inherited by operations like fixed projection formulas. However, because all other networks are pre-trained, train the Flow Net was found to be time-consuming and found that perform not well when we stop training.

What's more, we have tried the *normal matching loss* introduced in Section 3.4, which explores the geometry information in the depth images. However, we have found that the introduction of this loss function leads the model to learn things even worse. We think that the reason might be that normals it predicts are in fact not correct, and we do not have any further implementations to constrain the predicted normals. As a result, this loss might mislead the whole system.

Some visual results are shown in Figure 3.

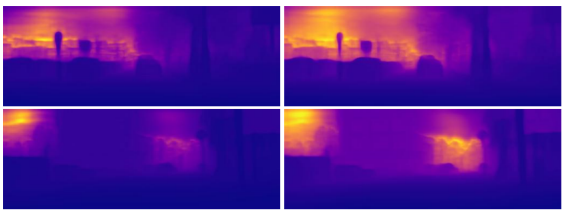

Figure 3. Depth map

5. Conclusions

In summary, we have tried a few different methods based on the established framework, and the method of *Auto doj mask weight* does improve the whole system. However, this problem is far from being well solved, for the overall system still requires a comparative long time to train, and we heavily rely on some of the pre-trained networks. As a result, a lot of work needs to be done and this project inspire our team to keep digging into the field of depth estimation.

6. Contribution

We want to thank for Kecheng Ye, for the most advanced work: the *Auto doj mask weight* and *pose constraint* methods are introduced by him. Thank for Yi'ang Ju's work in implementing *normal match loss* in section 3.4. Thank for Jinxi Xiao's idea of *flow net* and making poster. Thank for Panfeng Jiang and Haoyu Wu's work in some written materials.

References

[1] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li, "Disentangling object motion and occlusion for unsupervised multi-frame monocular depth," *arXiv preprint arXiv:2203.15174*, 2022.

[2] R. Garg, V. K. B.G., G. Carneiro, and I. Reid, *Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue*, p. 740–756. Springer International Publishing, 2016.

[3] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," October 2019.

[4] L. Sun, J.-W. Bian, H. Zhan, W. Yin, I. Reid, and C. Shen, "Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.

[5] X. Miao, Y. Bai, H. Duan, Y. Huang, F. Wan, X. Xu, Y. Long, and Y. Zheng, "Ds-depth: Dynamic and static depth estimation via a fusion cost volume," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[6] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," in *CVPR Workshop on The Future of Datasets in Vision*, 2015.