A WEIGHT-SHARED DUAL-BRANCH CONVOLUTIONAL NEURAL NETWORK FOR UNSUPERVISED DENSE DEPTH PREDICTION AND CAMERA MOTION ESTIMATION

Hong Liu, Yaofeng Dong, Weibo Huang

Key Laboratory of Machine Perception (Ministry of Education)
Shenzhen Graduate School, Peking University
{hongliu, yaofengdong, weibohuang}@pku.edu.cn

ABSTRACT

Convolutional Neural Network (CNN) can be used to indiscriminately predict dense depth and camera motion from images, however, ignoring the relationship between depth map and camera motion increases the computational burden to label the datasets and limits the accuracy of the results. In this paper, an end-to-end unsupervised dual-branch CNN is proposed to predict a pixel-wise depth map and simultaneously estimate camera pose. In particular, a weight sharing strategy for two branches is designed to increase the connection between depth map and camera motion. Besides, to reduce the impact of photometric noise, the intermediate feature maps are utilized to compute feature errors. Experimental results on the KITTI datasets demonstrate that our method achieves better performance on dense map prediction and camera pose estimation comparing with the state-of-the-art approaches.

Index Terms— Depth Prediction, Pose Estimation, Feature Map, Weight-shared

1. INTRODUCTION

Perceiving the surrounding environment and ego motion are preconditions for moving agents. The technique to achieve these two tasks is named Simultaneous Localization and Mapping (SLAM), which can be applied in many autonomous applications, such as mobile robots, unmanned aerial vehicle [1, 2], virtual and augmented reality [3], etc.

Relation to prior work: Most of the applications require for a map dense enough to depict the environment with more details. One method to obtain dense map is to utilize depth sensors, such as RGB-D cameras or stereo cameras. However, these sensors requires for very expensive computational expense, and they are not as ubiquitous as monocular color cameras [4, 5]. Therefore, it is essential to study the SLAM approaches using a monocular camera.

Traditional monocular SLAM contains three modules: front-end image operations, back-end optimization and loop closure [6]. According to the different front-end image operations, SLAM approaches can be divided into two categories. One is feature-based method, which adopts hand-crafted features to build data association between frames [7]. Another one named direct method is based on the photometric consistence hypothesis, which directly deals with the pixel intensity [8, 9]. In feature-based method, the number of map points depends on the amount of the features in the image, and the number of the features is a result of the trade off between the performance and computational complexity. As a result, the feature-based approaches are good at tracking camera pose, but the map is too sparse to depict the environment. Compared with the feature-based approaches, the direct method reconstructs the environment with a denser map, since the amount of the high intensity pixels is much more than the features. However, the maps predicted by the direct approaches are still too sparse to make further use of.

Attributing to the remarkable performance of Convolution Neural Network (CNN) in dealing with pixel wise task, many researchers incorporate SLAM with deep learning in recent years. With a coarse to fine idea, Egien *et al.* [10] employed two deep networks to predict a depth map, which can used to illustrate the rough structure of the environment. Liu *et al.* [11] treated the depth prediction as a continuous random field (CRF) learning problem. However, these methods assume that the camera motion is known in advance. Although the camera motion can also be regressed by an end-to-end CNN from a single image, such as the PoseNet [12], which is robust to different lighting and motion blur, the camera motion accuracy is worse than the traditional methods [7, 8].

In order to adapt to the monocular SLAM targets, which is to predict the dense map and estimate the camera motion only with consecutive frames, there are several methods [13–15] utilizing the photometric constraints to achieve unsupervised training. Nikolaus *et al.* [13] took optical flow constraints between frames as loss function to train the CNN, named DispNet. Sudheendra *et al.* [14] detected and tracked the objects in the image to improve the depth prediction accuracy. In [16], to smoothly predict depth map, the photometric noise caused by photometric error is regulated by L2 loss function. As a result, this method can predict the map more smoothly. Although these unsupervised approaches can be applied more flexible, the depth map depicted by these methods are not so

clear as the one predicted by the supervised methods. Moreover, these methods ignore the connection between camera pose and depth map both in the aspect of image and 3D reconstruction, which limits the accuracy of camera motion and depth map.

To deal with these problems, an end-to-end unsupervised dual-branch CNN is proposed to predict dense depth map and camera motion from pairs of consecutive frames. The proposed network concludes two branches, which share weights in the encoder to increase the connection among the dense depth prediction and camera motion estimation, since the depth and the camera motion accuracy can promote each other during the observation. Unsupervised training is applied to train the proposed network with the constraints between the consecutive frames. The training loss is designed to contain two parts, one is the photometric error and another is the proposed feature error. Motivated by the front-end image procedure in the direct SLAM method, the feature error is computed by the intermediate feature maps aiming to reduce the impact of the photometric noise. Experiments based on the KITTI datasets demonstrate that our method performs well in both depth prediction and camera pose estimation both in the structured and the textured environments.

2. UNSUPERVISED STRUCTURED DEPTH AND MOTION LEARNING MODEL

2.1. Unsupervised CNN architecture

The architecture of our network is shown in Fig. 1. An end-to-end convolutional neural network is proposed to predict the dense depth map and estimate the camera motion from pairs of successive frames. To clearly describe the structure of the network, we divide the network into several parts. The Pre-layers share weights for camera motion and depth map prediction to ensure the feature maps are calculated in the same manner. Therefore, the same landmarks in the real world can be presented the same, which strengthen the connections between the depth map and the camera motion. To reduce the impact of the photometric noise, the feature map and the original image are combined in the loss to train the network in an unsupervised manner. The final outputs of the networks conclude the scaled depth maps and the 6 DOF camera pose. Detailed explanations are shown as follows.

A. Pre-layers for encoding and feature map

The Pre-layers works as part of the encoder in the end-to-end network. Batch normalization [17] and ReLU $max(0, x)$ are conducted after each convolution layer. Then, the inputs are sub-sampled after a max-pooling layer with stride 2. On the one hand, max-pooling is used to reduce the computational burden. On the other hand, the remarkable landmarks are strengthened to reduce the photometric noise. At the same time, each layer in the Pre-layers can produce a set of feature maps. The feature maps before every max-pooling layer are preserved for decoder to achieve scale and translation invariance.

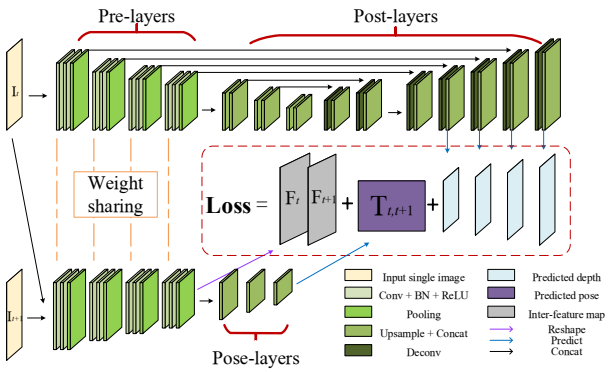

Fig. 1. Network architecture for depth and camera motion prediction. The width and height of the cubes denote the output spatial dimensions of the corresponding layer, and the size change in the successive layer is 2, increase or decrease. (a) The kernel size of the encoder layers before the Pre-layers is 7, and the number of the output channels is 64. The kernel size of the Post-layers is 3. (b) The pose networks share the encoder layers of the depth prediction layers. The input is single view image in successive while the output is the 6-DOF camera translation $\mathbf{T}_{t,t+1}$ between two adjacent frames. (c) The intermediate feature maps and the predictions are combined in the loss function for the unsupervised training of the networks.

B. Post-layers for scaled depth prediction

The rest of the layers for depth prediction are to achieve the encoder and decoder structure based on the Pre-layers. The Post-layers is based on the architecture of DispNet [13], which is used for stereo video depth prediction, and the kernel sizes are adjusted to suit the Pre-layers. The encoder layers in this part are convolution layers followed by the batch normalization and ReLU layers. The decoder upsamples the feature maps which are memorized before max-pooling in the Pre-layers. The de-convolution layers convolve the upsampled feature maps to predict the depth. There are four scales of depth maps predicted from each single view image when training the networks. The four scales of depth maps are all estimated to achieve the scale invariance.

C. Pose-layers for camera motion estimation

The camera pose is also estimated based on the feature maps of the Pre-layers. Differently from the depth prediction branch, the inputs of the Pose-layers are pairs of reshaped feature maps in the continuous timestamps, such as the reshaped feature maps $\mathbf{F}_t$ and $\mathbf{F}_{t+1}$ of $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$. Since the camera pose is a translation from one state to another, the inputs have to contain at least two single view images taken at different camera poses. The output is a 6-DOF camera motion $\mathbf{T}_{t,t+1}$ between the input frames $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$.

2.2. Loss function for unsupervised training

The loss function is used as a forward-backward consistency constraint between consecutive frames when training the networks. There are two parts in the loss function: the photometric error and the newly proposed feature error. The photometric error reflects the relationships among the successive frames, the camera translation and the depth map. Thus the photometric error can work as labels in the unsupervised CNN to train for the camera pose and the depth map. Therefore, the two branches can be trained at the same time to make the training process effective. However, the photometric error usually brings out photometric noise because of the pixel-wise calculation. Among successive images, some remarkable elements should be emphasized, and the smooth pixels which will bring out photometric noise should be reduced. To address this problem, we increase the ratio of the feature error to strengthen the environment structure. The total loss function is defined as follows:

$$\mathcal{L}_{\mathbf{t},\mathbf{t+1}} = \sum_s \mathcal{L}_{\mathbf{s}}^{\mathbf{p}} + \gamma \sum_s \mathcal{L}_{\mathbf{s}}^{\mathbf{f}}, \tag{1}$$

where $\mathcal{L}_{t,t+1}$ is the loss between the tth and the $t+1$th image. The subscript s denotes the scales. The four scales of predicted depth maps are all calculated here with the correspondence scaled images to achieve scale invariance. The loss $\mathcal{L}_{\mathbf{s}}^{\mathbf{p}}$ and $\mathcal{L}_{\mathbf{s}}^{\mathbf{f}}$ are the photometric error and feature error, respectively. Compared with the pixel-wise photometric error, the feature error only measures the error around the remarkable boundaries, thus the feature error is smaller than the photometric error. The parameter γ is used to adjust the feature error to the same order of the magnitude with the photometric error to ensure both errors contribute to the network training.

A. Pixel-wise photometric error

The photometric error function is defined as follows:

$$\mathcal{L}^{\mathbf{p}} = \sum_i \|\mathbf{I}_t(\mathbf{p}_i) - \mathbf{I}_{t+1}(\omega(\mathbf{p}_i, \mathbf{d}_{t+1}, \mathbf{T}_{t,t+1}))\|, \tag{2}$$

where $\mathbf{p}_i$ is the intensity of the ith pixel in frame $\mathbf{I}_t$, while $\mathbf{d}_{t+1}$ and $\mathbf{T}_{t,t+1}$ are the predicted depth and camera translation, which will be updated to minimize the loss. And formula $\omega()$ is defined as the 3D projection warp function, which projects the 3D point into the target image from the holding image. The mathematical definition of $\omega()$ can be expressed as:

$$\omega(\mathbf{p}, \mathbf{d}, \mathbf{T}) = \begin{pmatrix} \mathbf{x}'/\mathbf{z}' \\ \mathbf{y}'/\mathbf{z}' \\ 1/\mathbf{z}' \end{pmatrix}, \tag{3}$$

$$\begin{pmatrix} \mathbf{x}' \\ \mathbf{y}' \\ \mathbf{z}' \\ 1 \end{pmatrix} = \mathbf{T} \begin{pmatrix} \mathbf{p}_x/\mathbf{d} \\ \mathbf{p}_y/\mathbf{d} \\ 1/\mathbf{d} \\ 1 \end{pmatrix}, \tag{4}$$

where $(\mathbf{p}_x, \mathbf{p}_y)$ denotes the pixel coordinates in the image, while $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$ denotes the transformed point in the 3D world, and $\mathbf{d}$ is the depth of the 3D point in the camera coordinates.

B. Feature error for environment structure

The feature error is calculated from the feature map generated by the Pre-layers. This function is with the assumption that the same identity in the successive images is also presented the same in the successive feature maps. It can be regarded as a feature space version of the photometric error function, and it is proposed aiming to take advantage of the robust feature map. The feature error is defined as follows:

$$\mathcal{L}^{\mathbf{f}} = \sum_j \|\mathbf{F}_t(\mathbf{f}_j) - \mathbf{F}_{t+1}(\omega(\mathbf{f}_j, \mathbf{d}_{t+1}, \mathbf{T}_{t+1,t}))\|, \tag{5}$$

where $\mathbf{F}_t$ means the intermediate feature map of frame $\mathbf{I}_t$. And $\mathbf{f}_j$ is the intensity value of the jth pixel in the feature map. The definition of the wrap function $\omega()$ is similarly to the one in function (3).

For the amount of the effective pixels in the feature map is small, we ignore the distribution of the features and directly measure the L_1 distance of all the features. Since the feature maps preserve the photometric consistence, the L_1 distance between the feature maps can also be minimized by the translation between the input images.

3. EXPERIMENTAL RESULTS

The performance of our method is evaluated on the KITTI [18] dataset, which is composed of several outdoor scenes captured while driving with car-mounted cameras and depth sensor. The evaluation of depth prediction in [10] is used to calculate the correct depth threshold, the absolute relative difference, the squared relative difference, the linear RMSE and the $\log$ RMSE compared with the groundtruth of pixel-wise depth and the pose. The camera pose prediction is measured by the Absolute Trajectory Error (ATE).

3.1. Quantitative Results

We compare our depth prediction results with some state-of-the-art methods based on CNN. As shown in Table I, these methods in the first part train the networks supervised with ground-truth of the camera pose or the depth. The rest of the methods are trained with the photometric constraints. The comparisons between these results suggest that the unsupervised methods have potential to be widely applied in depth prediction.

The Sfm-Net has three versions with the same network architecture which are trained in different manner. The log RMSEs of the three versions are 0.31, 0.45, 0.77, respectively, which means that the networks trained with ground-truth have advantage over the unsupervised methods. While,

Table I. Single view depth evaluation

Method	Train	Error				Accuracy
		Abs Rel	Sq Rel	RMSE	RMSE (log)	δ<1.25
Eigen et al [10]	depth[1]	0.40	5.53	8.71	0.40	0.59
Godard et al [19]	stereo[2]	0.15	1.34	5.92	0.25	0.80
Liu et al [11]	depth	0.20	1.61	6.52	0.28	0.68
Zhang et al [20]	stereo	0.14	1.39	5.87	0.24	0.80
SfM-Net [14]	stereo	-[3]	-	-	0.31	-
SfM-Net [14]		-	-	-	0.45	-
SfM-Net [14]		-	-	-	0.77	-
SfM-learner [15]		0.22	2.23	7.53	0.29	0.68
ours		0.21	2.11	6.67	0.29	0.73

[1] The "depth" means that method is trained with depth ground-truth as supervision.
[2] The "stereo" means the model is trained with pairs of images with known disparities.
[3] Blank "-" in this row means the method is unsupervised.

Table II. Comparison of translation RMSE error (m)

Seq	ours	ORB	SfM-learner	Mean Odometry
9	0.020	0.014	0.021	0.032
10	0.018	0.012	0.020	0.028

the log RMSEs of our method and the sfm-learner [15] are 0.29, which is better than the method proposed by Eigen *et al* [10], 0.40, and the Sfm-Net, whose result is 0.31. The results demonstrate that our methods performs well to predict dense depth map from single images.

To evaluate the performance of our method in camera pose estimation, we measure the Absolute Trajectory Error (ATE) as shown in Table II. Our performance is better than sfm-learner [15], which demonstrates that our strategy to strengthen the connections between the camera pose and depth map works well.

3.2. Qualitative Results

We intuitively compare our performance of the single view depth prediction with sfm-learner [15] to analysis the advantages in environment structure description of our method. As shown in Fig. 2, our depth map is much clearer than the sfm-learner no matter in the textured or structured environment. In the first two input frames, there are not too much textures to be depicted, the predicted depth of sfm-learner is blurry. While in our depth map, the messy textures are ignored and the main structure is clearly predicted. In the following images, there are only fuzzy outlines of the environment in the sfm-learner's results. In the meanwhile, our method is also shown to be good at depicting the rigid bodies, such as the cars, walls.

3.3. Feature error efficiency

The reshaped feature map generated by the Pre-layers is shown in Fig. 3. Different color is utilized to distinguish different feature. The feature map mainly depict the structure of the environment. The disturbance around the margin are ignored after dealing with the feature error. Therefore, the dense depth map can be more accurate to describe the environment both textured and structured.

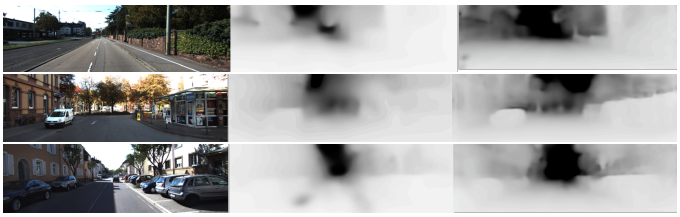

Fig. 2. Single-view depth prediction. The images in the first column are the original input images, and the depth maps in the second column is the results of sfm-learner, while the last column of depth maps are predicted by our method.

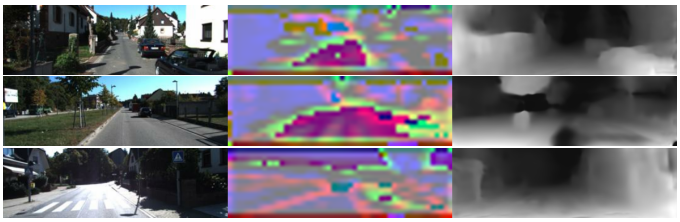

Fig. 3. Intermediate feature maps when training. The images in the first column are the inputs, and in the second column are the intermediate feature maps. The image in the final line is the depth map of the input.

4. CONCLUSION

In this paper, we propose an ene-to-end dual-branch CNN method to predict the dense depth map and the camera motion between pairs of consecutive images. Our method is trained in an unsupervised manner, while the loss function contains both photometric error term and feature error term. The strategy to share weights between the two branches promotes the relationship between depth map and camera motion, and it can effectively synchronously improve the accuracy of the camera motion and dense depth map. The feature error is based on the feature maps that strengthen the environment structure. At the same time, the feature error deduces the impact of photometric noise caused by the photometric error. The experimental results verified that our method can predict a dense depth map more accurate in both textured and structured environments, and the camera pose estimated by our method achieves comparable accuracy comparing with state-of-the-art method. In the future work, we would like to study more about the feature maps to improve the performance of the SLAM methods.

5. REFERENCES

[1] W. Huang and H. Liu, "Online initialization and automatic camera-imu extrinsic calibration for monocular visual-inertial slam," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 5182–5189.

[2] H. Liu, W. Huang, and Z. Wang, "A novel re-tracking strategy for monocular slam," in *IEEE Int. Conf. Acoust., Speech, Signal Process.* IEEE, 2017, pp. 1942–1946.

[3] J. Polvi, T. Taketomi, G. Yamamoto, A. Dey, C. Sandor, and H. Kato, "Slidar: A 3d positioning method for SLAM-based handheld augmented reality," *Computers & Graphics*, vol. 55, pp. 33–43, 2016.

[4] J. Stckler J. Engel and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1935–1942.

[5] Y. Lu and D. Song, "Robust RGB-D odometry using point and line features," *Proc. IEEE Int. Conf. Comput. Vision*, pp. 3934–3942, 2015.

[6] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, Ian D.Reid, and John J.Leonard, "Past, Present and Future of Simultaneous Localization And Mapping: Toward the Robust-Perception Age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2017.

[7] J.M.M Montiel R. Mur-Artal and J.D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," vol. 40, no. 99, pp. 611–625, 2017.

[9] T. Schöps J. Engel and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," *Proc. IEEE Europ. Conf. Comput. Vision*, pp. 834–849, 2014.

[10] E. David, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *International Conference on Neural Information Processing Systems*, 2014, pp. 2366–2374.

[11] F. Liu, C. Shen, G. Lin, and I Reid, "Learning depth from single monocular images using deep convolutional neural fields," vol. 38, no. 10, pp. 2024–2039, 2016.

[12] M. Grimes A. Kendall and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," *Proc. IEEE Int. Conf. Comput. Vision*, pp. 2938–2946, 2015.

[13] P. Hausser P. Fischer D. Cremers A. Dosovitskiy N. Mayer, E. Ilg and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow and scene flow estimation," *Proc. IEEE Int. Conf. Comput. Vision. Pattern. Recog.*, pp. 4040–4048, 2016.

[14] C. Schmid S. Rahul S. Vijayanarasimhan, R. Susanna and F. Katerina, "SfM-Net: Learning of structure and motion from video," *Proc. IEEE Int. Conf. Comput. Vision. Pattern. Recog.*, 2017.

[15] N. Matthew L. Snavely T. Zhou, M. Brown and G. David, "Unsupervised learning of depth and ego-motion from video," *Proc. IEEE Int. Conf. Comput. Vision. Pattern. Recog.*, pp. 6612–6619, 2017.

[16] G. Carneiro R. G, KBG. Vijay and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," *Proc. IEEE Europ. Conf. Comput. Vision*, 2016.

[17] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," pp. 448–456, 2015.

[18] P. Lenz A. Geiger and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *Proc. IEEE Int. Conf. Comput. Vision. Pattern. Recog.*, pp. 3354–3361, 2012.

[19] C. Godard, A. O. Mac, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," *Proc. IEEE Int. Conf. Comput. Vision. Pattern. Recog.*, pp. 6602–6611, 2017.

[20] H. Zhan, R. Garg, W. C. Saroj, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," June 2018.