A WEIGHT-SHARED CNN 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
In most of the CNN-based methods, the predicted depth map usually can’t be clearly recognized. In this paper, an end-to-end CNN is proposed to predict depth and estimate camera pose from pairs of consecutive frames. There are two branches in the network, which share weights to increase the connection between dense map and camera motion. The network is designed and trained in an unsupervised manner with a newly proposed feature error to improve the depth accuracy of the environment structure. The feature error is computed by the intermediate feature maps to reduce the impact of the photometric noise. Experimental results on the KITTI datasets demonstrate that the proposed method can achieve better performance of dense map prediction and camera pose estimation compared 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). SLAM can be applied in many autonomous applications, such as mobile robots, unmanned aerial vehicle [1, 2], virtual and augmented reality [3], etc.

Most of the applications require for a dense map to depict the environment with more details. Some approaches get dense map through depth sensors, such as RGB-D cameras and stereo cameras. However, these cameras are easy to loss precise in outdoor environment, 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, the SLAM approaches can be divided into two categories. One is feature-based approaches, which adopt hand-crafted features to build data association [7] between frames. Another one named direct methods are based on the photometric consistence hypothesis [8, 9] involving all the pixels in the images. In feature-based approaches, the number of map points are decided by 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 approaches provide much more detailed maps that can reconstruct the environment structure. However, the maps predicted by the direct approaches are still too sparse for machine to make further use of. In the previous monocular approaches, lots of previous monocular approaches usually abandon the maps for their lack of rich environment information, nevertheless they can be further used in navigation and path planning.

Considering the remarkable performance of Convolution Neural Network (CNN) in dealing with pixel-wise tasks, many researchers incorporate SLAM with deep learning in recent years. Some approaches individually estimate the camera pose or depth map totally through CNN. The PoseNet [10] regressed the camera motion from a single image with an end-to-end CNN, which is robust to difficult lighting, motion blur and different camera intrinsics because of the high level feature generated by the network. Eigen et al. [11] employed two deep network stacks to predict the depth map from coarse to fine, whose depth map can be used to illustrate the rough structure of the environment. Liu et al. [12] treated the depth prediction as a continuous random field (CRF) learning problem. However, the limitation of these methods is that training the CNNs depends on the known camera motion, which is costly to assemble. Besides, in most of the SLAM applications, the camera motion cannot be accurately measured. To solve this challenge, there are several methods [13–15] utilizing the photometric constraints between consecutive frames to replace the known camera motion. 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. Ravi et al. [16] extended the loss function by L2 regulation to deal with the photometric error caused by photometric noise, which can predict the map more
smoothly. As a result, this method can predict the map more smoothly. Although these unsupervised approaches can be applied more flexible, the environment structure depicted by the depth maps predicted 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.

In this paper, an unsupervised end-to-end CNN is proposed to predict dense depth and camera pose from pairs of consecutive frames. The proposed network concludes two branches, which are with the same encoder to increase the connection among the dense depth prediction and camera pose estimation. 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. 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

![Network architecture for depth and camera motion prediction](image)

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 \( T_{t,t+1} \) 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.

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 depth map and estimate the camera pose from pairs of successive frames. To clearly describe the structure of the network, we divide the network into several parts. The Pre-layers are shared for camera pose 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 and the camera poses. 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 6DOF camera pose. The detailed explanations will be shown in the following.

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 \( \text{ReLU} \ max(0, x) \) 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.

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 \( F_t \) and \( F_{t+1} \) of \( I_t \) and \( I_{t+1} \). 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 \( T_{t,t+1} \)
between the input frames \( I_t \) and \( I_{t+1} \).

2.2. Loss function for unsupervised training

The loss function is used as a forward-backward consisten-
cy constraint between consecutive frames when training the
networks. There are two parts in the loss function: the pho-
tometric error and the newly proposed feature error. The pho-
tometric 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 CN-
N 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 er-
or usually brings out photometric noise because of the pixel-
wise calculation. Among successive images, some remark-
able 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:

\[
L_{t,t+1} = \sum_s L^p_s + \gamma \sum_s L^f_s, \tag{1}
\]

where \( L_{t,t+1} \) is the loss between the \( t \)th and the \( t+1 \)th image.
The subscript \( s \) denotes the scales. The four scales of predict-
ed depth maps are all calculated here with the correspond-
ence scaled images to achieve scale invariance. The loss \( L^p_s \)
and \( L^f_s \) 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 bound-
aries, thus the feature error is smaller than the photometric
error. The parameter \( \gamma \) 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:

\[
L^p = \sum_i \| I_t(p_i) - I_{t+1}(\omega(p_i, d_{t+1}, T_{t,t+1})) \|, \tag{2}
\]

where \( p_i \) is the intensity of the \( i \)th pixel in frame \( I_t \), while
\( d_{t+1} \) and \( T_{t,t+1} \) are the predicted depth and camera transla-
tion, which will be updated to minimize the loss. And for-
mla \( \omega() \) 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:

\[
(\begin{array}{c}
x' \\
y' \\
z'
\end{array}) = T(\begin{array}{c}
p_x/d \\
p_y/d \\
1/d
\end{array}), \tag{3}
\]

where \( (p_x, p_y) \) denotes the pixel coordinates in the image,
while \( (x', y', z') \) denotes the transformed point in the 3D
world, and \( d \) is the depth of the 3D point in the camera co-
ordinates.

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:

\[
L^f = \sum_j \| F_t(f_j) - F_{t+1}(\omega(f_j, d_{t+1}, T_{t+1,t}))) \|, \tag{5}
\]

where \( F_t \) means the intermediate feature map of frame \( I_t \).
And \( f_j \) is the intensity value of the \( j \)th 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 trans-
lation 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 [11] is used to
calculate the correct depth threshold, the absolute relative dif-
fERENCE, 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 unsup-
vised methods have potential to be widely applied in depth
prediction.
Table I. Single view depth evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>Train</th>
<th>Abs</th>
<th>Rel</th>
<th>Sq</th>
<th>RMSE</th>
<th>RMSE (log)</th>
<th>δ&lt;1.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigen et al [11]</td>
<td>depth</td>
<td>0.40</td>
<td></td>
<td>5.53</td>
<td>8.71</td>
<td>0.40</td>
<td>0.59</td>
</tr>
<tr>
<td>Godard et al [19]</td>
<td>stereo</td>
<td>0.15</td>
<td>1.34</td>
<td>5.92</td>
<td>0.25</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>Liu et al [12]</td>
<td>depth</td>
<td>0.20</td>
<td>1.61</td>
<td>6.52</td>
<td>0.28</td>
<td>0.68</td>
<td></td>
</tr>
<tr>
<td>Zhang et al [20]</td>
<td>stereo</td>
<td>0.14</td>
<td>1.39</td>
<td>5.87</td>
<td>0.24</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>SfM-Net [14]</td>
<td>stereo</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SIM-Net [14]</td>
<td></td>
<td>0.22</td>
<td>2.23</td>
<td>7.53</td>
<td>0.29</td>
<td>0.68</td>
<td></td>
</tr>
<tr>
<td>SIM-learner [15]</td>
<td></td>
<td>0.21</td>
<td>2.11</td>
<td>6.67</td>
<td>0.29</td>
<td>0.73</td>
<td></td>
</tr>
<tr>
<td>ours</td>
<td></td>
<td>0.21</td>
<td>2.11</td>
<td>6.67</td>
<td>0.29</td>
<td>0.73</td>
<td></td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Seq</th>
<th>ours</th>
<th>ORB</th>
<th>SFM-learner</th>
<th>Mean Odometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.020</td>
<td>0.014</td>
<td>0.021</td>
<td>0.032</td>
</tr>
<tr>
<td>10</td>
<td>0.018</td>
<td>0.012</td>
<td>0.020</td>
<td>0.028</td>
</tr>
</tbody>
</table>

The SiM-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, 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 [11], 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

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.

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.

4. CONCLUSION

In this paper, we propose a CNN-based method to predict the depth map from single view image and the camera pose between pairs of consecutive images. A VGG-style CNN is adopted in our end-to-end network which is trained in an unsupervised manner with the photometric error and an newly proposed feature error. The feature error is based on the feature maps that strengthen the environment structure, which deduces the impact of photometric noise brought out by the photometric error. In the meanwhile, the strategy to share weights for both depth prediction and camera pose estimation is demonstrated to be effective to synchronously improve the accuracy of the camera pose estimation. The experimental results verify that our method can predict the dense depth more accurately for both textured and structured environments, and can estimate the camera pose with a comparable performance with the state-of-the-art methods. In the future work, we would like to study more about the feature maps to improve the performance of the SLAM methods.
5. REFERENCES


