Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Simultaneous Monocular Visual Odometry and Depth Reconstruction with Scale Recovery*

Yong Luo, Guoliang Liu, Hanjie Liu, Tiantian Liu and Guohui Tian
School of Control Science and Engineering
Shandong University
Jinan, Shandong, China
{*liuguoliang*}*@sdu.edu.cn*

Ze Ji
School of Engineering
Cardiff University
Cardiff, CF24 3AA, UK
jiz1@cardiff.ac.uk

***Abstract*— In this paper, we propose a deep neural network that can estimate camera poses and reconstruct the full resolution depths of the environment simultaneously using only monocular consecutive images. In contrast to traditional monocular visual odometry methods, which cannot estimate scaled depths, we here demonstrate the recovery of the scale information using a sparse depth image as a supervision signal in the training step. In addition, based on the scaled depth, the relative poses between consecutive images can be estimated using the proposed deep neural network. Another novelty lies in the deployment of view synthesis, which can synthesize a new image of the scene from a different view (camera pose) given an input image. The view synthesis is the core technique used for constructing a loss function for the proposed neural network, which requires the knowledge of the predicted depths and relative poses, such that the proposed method couples the visual odometry and depth prediction together. In this way, both the estimated poses and the predicted depths from the neural network are scaled using the sparse depth image as the supervision signal during training. The experimental results on the KITTI dataset show competitive performance of our method to handle challenging environments.**

***Index Terms*— Visual odometry, depth prediction, deep learning, view synthesis**

I. Introduction

Visual odometry (VO) is a key technique for estimating camera poses through analyzing sequential camera images and has been used in a broad range of real-world applications of localization, mapping, and navigation for autonomous driving, robots, advanced driver assistance systems and augmented reality. Geometric VO estimates camera poses by minimizing the projection error of the three-dimensional (3D) points to consecutive image planes or minimizing the gradients of pixel intensities across consecutive images [1]. Previous works show that geometric VO has achieved great success in structured and controlled environments. However, it is sensitive to the camera parameters and, consequently, can result in decreased performance in challenging environments, such as low textures, motion blur, and illumination variations.

With the rapid advances of computing power and the emergence of large-scale visual datasets, deep learning based VO shows a great potential for enhancing the robustness in handling challenging images due to illumination variations, image noises, image motion blur and low textures. However, supervised deep learning for VO would require the ground-

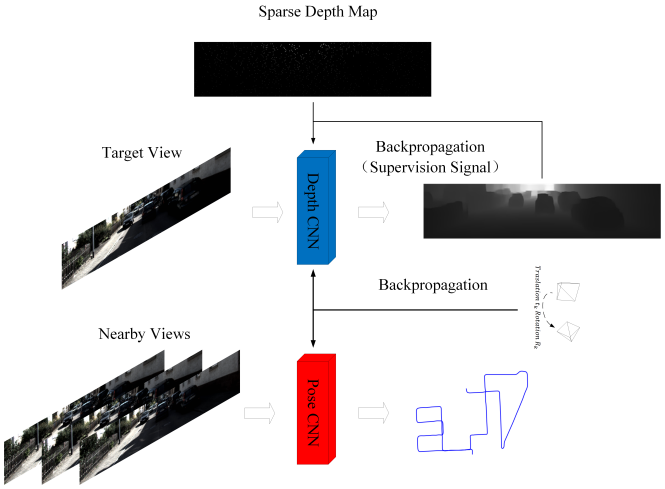

Fig. 1: The architecture of our system comprises the pose estimation convolution neural network (Pose CNN) and the depth prediction convolution neural network (Depth CNN). The input of the Depth CNN is the target image of current view, whereas the input of the Pose CNN consists of three consecutive images, with the target image located in the middle of the image sequence. The output of the Depth CNN is a per-pixel disparity map (inverse of predicted depth) of the target image, whereas the output of the Pose CNN is the relative pose between consecutive images. To recover the scale information of predicted depth and relative pose, the sparse depth images are used as supervision signals during the training step.

* This work is partially supported by the National Key R&D Program of China (#2018YFB1306504), National Natural Science Foundation of China (#61603213, #91748115), Young Scholars Program of Shandong University (#2018WLJH71), the Fundamental Research Funds of Shandong University, and the Taishan Scholars Program of Shandong Province. (Corresponding author: Guoliang Liu).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

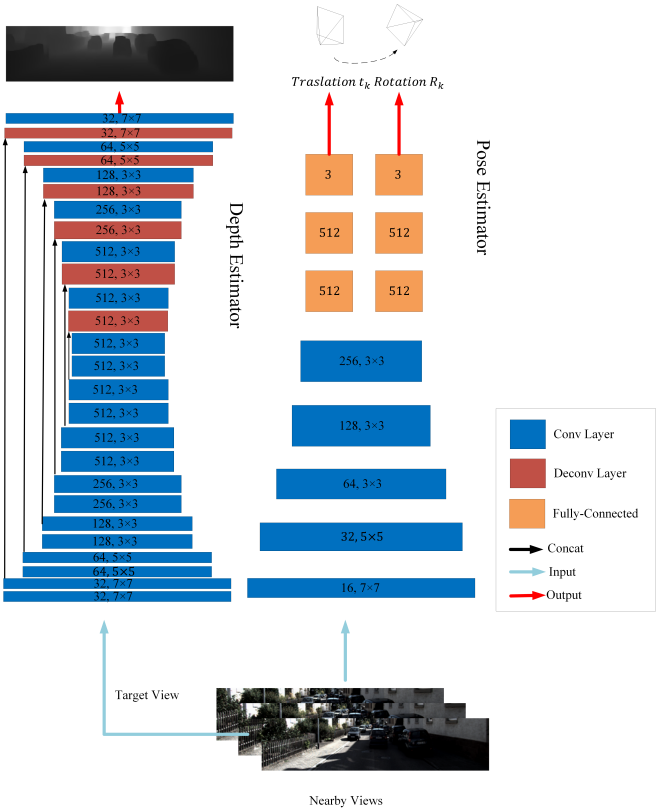

Fig. 2: The details of proposed neural network for testing. The neural network can simultaneously output visual odometry and depth prediction by the proposed pose network (pose estimator) and depth network (depth estimator), whereas the inputs of the system are consecutive monocular images. The depth network reconstructs the depth of the target view in full resolution based on the encoder-decoder structure. The pose network consists of 5 stride-2 convolutions followed by 3 fully-connected layers. Due to the high non-linearity of the rotation, we decouple the translation and the rotation with two separate groups of fully-connected layers after the last convolutional layer.

truth of the camera poses as the supervision data for training that imposes a challenge to acquire accurate labels. Recently, Zhou et al. [2] used an unsupervised learning method to estimate the camera motion and predict the depth using consecutive monocular images, which achieved impressive results since the view synthesis is employed to handle geometric constraints between the depth and relative pose. However, scale information from monocular images cannot be recovered from their system. Inspired by Ma et al. [3], we attempt to recover the scale information by using sparse depth measurements to augment the supervision data for training the network. In this work, as shown in Fig. 1, we propose a dual-network architecture that comprises a pose estimation convolution neural network (Pose CNN) and a depth prediction convolution neural network (Depth CNN). The input of the Depth CNN is the target image of current view, whereas the input of the Pose CNN consists of three consecutive images, with the target image located in the middle of the image sequence. The output of the Depth CNN is a per-pixel disparity map (inverse of predicted depth) of the target image, whereas the output of the Pose CNN is the relative pose between consecutive images. To recover the scale information of predicted depth and relative pose, the sparse depth images are used as supervision data during the training step.

In the work, a novel loss function is developed by fusing the view synthesis, depth difference between sparse depth and the predicted depth, and depth smoothness information. The sparse depth measurements can be easily obtained through various technologies, such as LiDARs, structured light cameras (e.g. Microsoft Kinect, Xtion, and RealSense), and stereo vision. In this paper, we generate depth maps by sparsely sampling the laser data, and projecting these depth samples to the image coordinates. The sparse depth map is used as supervision data for the depth network to recover the scale of the scene depth, and then this scale information is shared to the pose network by view synthesis, as shown in Fig. 1. Our main contributions are as follows:

- The proposed neural network can output scaled full resolution depth of the target view using monocular images as input. The novelty in solving the scale recovery of the depth is the employment of corresponding sparse depth images for supervised training of the proposed neural network.
- The visual odometry with the absolute scale can be estimated by sharing scale information with the depth neural network using view synthesis. Meanwhile, the view synthesis also improves the accuracy of dense depth reconstruction.

Finally, we evaluate the proposed method on the public KITTI dataset [4]. The experiment results show the effectiveness of our method for simultaneous depth reconstruction and visual odometry estimation compared to some state-of-the-art methods.

II. METHODOLOGY

In this section, we introduce the proposed neural network architecture of our system, which mainly includes the pose network for visual odometry and the depth network for full resolution depth prediction. We also present the sparse depth generation method and the novel loss functions by combining the view synthesis and the supervision sparse depth for training.

A. Network Architecture

As shown in Fig. 2, the proposed dual-network architecture comprises two parallel networks for the estimation of depths

TABLE I: Comparison of VO results on KITTI datasets

seq.	Ours		Liu et al. [5]		UnDeepVO[6]	
	Trans[%]	Rot[deg/m]	Trans[%]	Rot[deg/m]	Trans[%]	Rot[deg/m]
00	**3.90**	**0.0171**	5.14	0.0213	4.14	0.0192
02	5.52	0.0240	4.88	0.0226	5.58	0.0244
05	**3.12**	0.0152	3.84	**0.0129**	3.40	0.0150
07	**2.40**	0.0211	3.80	**0.0171**	3.15	0.0248
08	5.22	0.0189	**2.95**	**0.0158**	4.08	0.0179
mean	**4.03**	0.0192	4.122	**0.0179**	4.07	0.0202

and camera poses respectively. We aim to use monocular images as inputs to the proposed neural network, while can also recover the scale of the reconstructed depths and estimated camera poses, as a result of the deployment of the sparse depth measurements as supervision signals to train the network. The pose network consists of 5 stride-2 convolution layers followed by 3 fully-connected layers. The pose network takes three consecutive frames of unlabelled monocular images as input, and outputs camera relative poses between these frames. The middle frame of the nearby views is called the target view, and other two frames are source views. Similar to [6], we decouple the translation and the rotation with two separate groups of fully-connected layers after the last convolutional layer. The reason is that rotations have high non-linearity, which means it is usually difficult to train rotations compared to translations.

The depth network reconstructs the scene depth based on the encoder-decoder structure. The target view is the input of the depth network, and a per-pixel disparity map (inverse of the depth) is the output.

B. Sparse Depth Generation

Sparse depth measurements can be obtained using a low-resolution depth sensor, e.g., LiDAR, and can also be inferred from view differences in a pair of stereo images. In this paper, the sparse depth measurements are derived from a LiDAR device.

A homogeneous 3D spatial point coordinate X_s can be acquired from the LiDAR measurement that can be converted to the homogeneous coordinate in the camera coordinate system as:

$$X_c = T_c X_s, \tag{1}$$

where T_c denotes the 4×4 transformation matrix from the laser coordinate system to the camera coordinate system, which includes a rotation matrix and a translation vector. The parameters of the transformation matrix T_c can be calibrated after the laser and camera are set up [4]. Then, the non-homogeneous version of X_c is projected onto the camera imaging plane based on the pinhole camera model to get the homogeneous pixel coordinates u:

$$u = KX_c, \tag{2}$$

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where K indicates the camera intrinsic matrix, f_x and f_y denote the focal lengths of the camera on the x-axis and y-axis respectively, (c_x, c_y) is the principal point, the center position of the aperture of the camera. Applying this operation for each spatial point coordinate acquired by LiDAR, we can get a pixel set U on the depth image D^* which includes pixel coordinates u with real depth measurements from LiDARs. For these pixels that do not have depth measurement, we set them as zero on the depth image D^*.

To generate a sparse depth map, we use a Bernoulli probability $p = \frac{m}{n}$ [3], where m is the target number of sampled depth pixels, and n is the total number of valid depth pixels in set D^*. For each pixel (i, j), we have

$$D(i,j) = \begin{cases} D^*(i,j), & (i,j) \in U \ and \ w < p \\ 0, & otherwise \end{cases} \tag{4}$$

where w is a random number in the interval $[0, 1]$. Applying this operation for each pixel, we generate a sparse depth map D. In this way, we can create more training data given a depth image from LiDAR.

C. Loss Function

Our loss function for training the neural network includes three parts: view synthesis loss, depth loss and depth smoothness loss. The view synthesis is to generate a different view image of current image using the predicted relative pose. The generated image is then compared to the real image, such that we can define a loss function to minimize their difference. The main steps of view synthesis are as follows: we can project the pixel p_t in the target image I_t to $\tilde{p}_s$ in the source image I_s using the estimated relative pose transformation $\hat{T}$, predicted depth value $\hat{D}(p_t)$ and the internal camera parameter matrix K as:

$$\tilde{p_s} = K\hat{T}\hat{D}(p_t)K^{-1}p_t \tag{5}$$

The pixel value of $\tilde{p}_s$ on the source image can be found by using bilinear interpolation. Therefore, a warped source image $\tilde{I}_s$ in the target view can be obtained by performing

the projection operation on each pixel in the target image I_t. Finally, we adopt a robust image similarity measurement [7] for the photometric loss:

$$L_v = \alpha \frac{1 - SSIM(I_t, \tilde{I}_s)}{2} + (1 - \alpha) \left| I_t - \tilde{I}_s \right| \quad (6)$$

where SSIM defined in [8] denotes the structural similarity index with the weight α. However, using the loss function L_v, we cannot get the scale of the dense depth map and the relative pose. To solve such a problem, we propose to use a sparse depth map as a supervision signal for the depth network. Let p denote a pixel coordinate in the generated sparse depth image d, and $\hat{D}$ denote the predicted depth image. The depth loss L_{depth} can be calculated by

$$L_{depth} = \sum_p \left| \hat{D}(p) - d(p) \right|, d(p) > 0 \quad (7)$$

In addition, we also employ an edge-aware depth smoothness loss L_{smooth} [2] weighted by image gradients to our neural network as

$$L_{smooth} = \sum_p \left| \bigtriangledown \hat{D}(p) \right| \cdot (e^{-|\bigtriangledown I(p)|})^T \quad (8)$$

where $\bigtriangledown$ is the vector differential operator, and T denotes the transpose operator. Therefore, the final loss function is

$$L_{final} = L_v + \lambda_d L_{depth} + \lambda_s L_{smooth} \quad (9)$$

where λ_d and λ_s are the weights of the depth loss and smoothness loss respectively.

III. EXPERIMENTS

In this section, we first introduce the implementation details of the training process, followed by the performance evaluation of our system in comparison with some state-of-the-art algorithms.

We implement the proposed method using TensorFlow [9], which is trained on a PC with an Intel(R) E5-1650 v3 @3.50GHz CPU and a TITAN X (Pascal) GPUs with 12GB of memory. To ensure a fair performance comparison, we use the same training data from the KITTI odometry dataset [4] presented by Zhou et al. [2], Li et al. [6] and Liu et al. [5]. The KITTI odometry dataset includes driving sequences with ground truth data obtained from the IMU/GPS readings. Because LiDAR has no measurement for the upper area of the image, we only use the lower part of the image, which results in a fixed crop size of 228×1226. We generate a sparse depth image by projecting depth samples of the LiDAR measurements onto image plane as described in Section II-B. During training, the sparse depth images with real depth values are used as the supervision signals. Finally, we resize the RGB images and sparse depth images to 96×416 using a bilinear interpolation method for training our network. Since the size of the KITTI odometry dataset is relatively limited, two online data augmentation techniques are used to enlarge the training dataset, as follows:

- Scaling: The input monocular and depth images are scaled by a random number $s \in [1, 1.15]$, and depths are divided by s.
- Cropping: Randomly crop the color and depth images to meet the size requirement of the neural network.

We then input the consecutive monocular images with their sparse depth images into the pose estimation network and the depth prediction network respectively, and train the network from scratch. We here do not use any ground truth data of relative poses for training. In the experimental stage, we set the weights of the loss functions as $\alpha = 0.85, \lambda_s = 0.5/(l)$, $\lambda_d = 0.6$, where l corresponds to the downsampling scale of the predicted depth image. The Adam optimizer [10] is used to train the network for up to $300K$ iterations with learning rate of 0.0002 and mini-batch size of 4. We follow the recommended parameters of the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In addition, we use a batch normalization [11] method for all the layers except the output layer.

A. Performance Evaluation of Pose Estimation on the KITTI Odometry Dataset

For testing, our neural network takes consecutive monocular images as input, and directly generates scaled poses. To demonstrate the performance of our method, we compare our pose estimator with the work of Liu [5] and UnDeepVO [6] The method proposed by Liu, UNDeepVO are based on unsupervised deep learning, while our method is based on the semi-supervised deep learning.

We use the evaluation method suggested in [4] to compare the performance of various methods as shown in Table I, where the average translational root mean square error drift (in percentage, %) and the average rotational root mean square error drift (in deg/m) are used as the basic metrics. It is clear that our pose estimator has better performance than the SfMLearner. In addition, our pose estimation has comparable performance with UnDeepVo for the monocular inputs. It is be noted that we only use 600 sparse depth samples during training. As the number of samples increases, the performance of our system can be further improved. Compared to the work of Liu et al. [5], we require only external sparse depth information, while they require accurate dense depth measurements. However, from the evaluation results, our method has achieved comparable performance with Liu's work, even if only a few depth samples are used for our training step. This is attributed to the fact that the depth reconstruction module and visual odometry module are mutually optimized by view synthesis. For visual comparison, we also plot the trajectories of the proposed method compared to VISO_M and SfmLearner.

TABLE II: Comparison of depth prediction on KITTI datasets

Method	Error metric				Accuracy metric		
	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Zhou et al. [2]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Li et al. [6]	0.183	1.730	6.57	0.268	-	-	-
Ma et al. [3]	0.073	-	3.378	-	0.935	0.976	0.989
Ours	**0.069**	**0.169**	**1.240**	**0.116**	**0.940**	**0.981**	**0.994**

Fig. 3: Our depth prediction results on the KITTI dataset. The pictures in the left column are raw RGB images, and the pictures in the right column are the estimated dense depth images.

B. Performance Evaluation of Depth Estimation on the KITTI Dataset

We evaluate our depth predictor on the KITTI dataset as in [12]. During the testing, the network takes only a monocular image as input and directly generates a scaled scene dense depth image. Fig. 3 shows the depth reconstruction results of our model in several scenarios, where the depths of thin structures are successfully predicted, such as poles, signs, etc. The qualitative comparisons of depth estimation methods are summarized in Table II, where K denotes the KITTI dataset and CS denotes the Cityscapes dataset [13]. It is clear that our depth estimation method has significant improvement compared to the works proposed by Zhou et al. [2], Li et al. [6], Ma et al. [3]. For instance, we use 600 depth samples can reduce the RMSE (Root Mean Square Error) from 6.86 meters to 1.24 meters, and boosts δ_1 from 67.8% to 94.0% compared to SfmLearner proposed by Zhou et al [2]. UNDeepVO proposed by Li et al. [6] cannot achieve similar performance to our depth estimator, because of the inherent limitation of binocular depth estimation, i.e., maximum depth is limited by the baseline and the focal length. Sparse-to-Dense propsoed by Ma et al. [3] uses RGBd, where d denotes the sparse depth map, images for training and testing, whereas our method only takes monocular images as inputs for testing.

IV. Conclusion

In this paper, we propose a simultaneous monocular visual odometry and depth prediction method using semi-supervised deep learning. We use a sparse depth map as a supervision signal for training the depth prediction network to recover the scale of the dense depth map, which is then shared to the pose estimation network by the view synthesis. Extensive experiments have been carried out on the prevalent KITTI dataset. The proposed neural network demonstrates competitive performance, outperforming several state-of-the-art algorithms in both scaled dense depth reconstruction and ego-motion estimation using monocular images. Considering the benefit of directly estimating scaled depths and camera poses, we believe the work presented in this paper will be of interest to further advance the technologies in autonomous vehicles, 3D reconstruction, robot navigation, and so on.

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, Dec 2011.

[2] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6612–6619.

[3] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," *CoRR*, vol. abs/1709.07492, 2017. [Online]. Available: http://arxiv.org/abs/1709.07492

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3354–3361.

[5] Q. Liu, R. Li, H. Hu, and D. Gu, "Using unsupervised deep learning technique for monocular visual odometry," *IEEE Access*, vol. 7, pp. 18 076–18 088, 2019.

[6] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," *CoRR*, vol. abs/1709.06841, 2017. [Online]. Available: http://arxiv.org/abs/1709.06841

[7] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *2017 IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 6602–6611.

[8] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[9] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456.

[12] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2366–2374.

[13] 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," *CoRR*, vol. abs/1604.01685, 2016. [Online]. Available: http://arxiv.org/abs/1604.01685