DeepSLAM: A Robust Monocular SLAM System with Unsupervised Deep Learning

Ruihao Li, Sen Wang, *Member, IEEE* and Dongbing Gu, *Senior Member, IEEE,*

***Abstract*—In this paper, we propose DeepSLAM, a novel unsupervised deep learning based visual Simultaneous Localization and Mapping (SLAM) system. The DeepSLAM training is fully unsupervised since it only requires stereo imagery instead of annotating ground-truth poses. Its testing takes a monocular image sequence as the input. Therefore, it is a monocular SLAM paradigm. DeepSLAM consists of several essential components, including Mapping-Net, Tracking-Net, Loop-Net and a graph optimization unit. Specifically, the Mapping-Net is an encoder and decoder architecture for describing the 3D structure of environment while the Tracking-Net is a Recurrent Convolutional Neural Network (RCNN) architecture for capturing the camera motion. The Loop-Net is a pre-trained binary classifier for detecting loop closures. DeepSLAM can simultaneously generate pose estimate, depth map and outlier rejection mask. We evaluate its performance on various datasets, and find that DeepSLAM achieves good performance in terms of pose estimation accuracy, and is robust in some challenging scenes.**

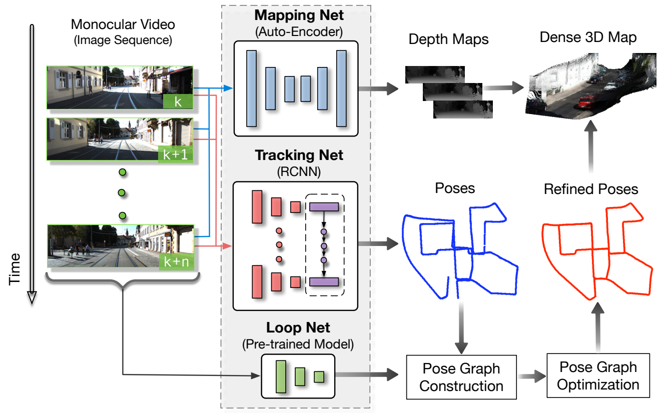
***Index Terms*—SLAM, unsupervised deep learning, depth estimation, RCNN, machine learning.**

I. Introduction

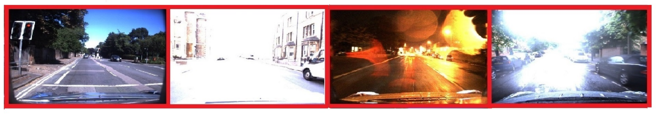
VISUAL Simultaneous Localization and Mapping (SLAM) is essential for robots operating autonomously with cameras, and is also the core of enormous vision based applications, e.g., virtual and augmented reality. Tremendous efforts have been made to visual SLAM in the robotics and computer vision communities. Especially, over the past decade several state-of-the-art visual SLAM systems have been designed based on sparse feature points [?], [?], [?], [?], [?] and photometric consistency of dense pixels [?], [?], [?].

However, since most of these methods are geometric model based, they cannot learn automatically from raw images or benefit from continuously increased datasets. Some of them are also fragile under challenging scenes. There increasingly arises a question, particularly when encountering large-scale dataset, that whether it is possible to understand and tackle the visual SLAM problem from a data-driven perspective and whether data-driven approaches are beneficial.

Recently, Deep Learning (DL) based methods have demonstrated a promising performance on pose and depth estimation [?], [?]. Most of them learn from raw images with limited consideration of geometric models, which have been well understood over these years and been recognized as the fundamentals of visual SLAM systems. It has been demonstrated, however, that the learning representation for depth estimation is more efficient if geometric constraints are respected [?]. Therefore, it is interesting to see how the learning representation could be effectively exploited for visual SLAM by seamlessly incorporating the knowledge accumulated over decades on geometric models. How to combine the geometric models and constraints with the network architecture and the loss function is still a hard and open problem. Geometric



(a) Testing framework of the proposed DeepSLAM.



(b) Challenging scenes DeepSLAM can deal with.

Fig. 1: (a) Testing framework of the proposed DeepSLAM. It takes monocular color images as input and produces depth maps, poses and point clouds as outputs by using Mapping-Net, Tracking-Net and Loop-Net. Mapping-Net is an auto-encoder architecture for depth estimation. Tracking-Net is a RCNN based architecture for pose estimation. Loop-Net is a CNN architecture for loop closure detection. Pose graph construction and optimization are implemented as the back-end. (b) DeepSLAM demonstrates a robust performance in some challenging scenes.

This project is supported in part by National Natural Science Foundation of China (Grant No. 61903377), and in part by EPSRC Robotics and Artificial Intelligence ORCA Hub (Grant No. EP/R026173/1), EU H2020 Programme under EUMarineRobots project (Grant ID 731103) and DeepField project (Grant ID 857339).

Ruihao Li is with Artificial Intelligence Research Center (AIRC), National Innovation Institute of Defense Technology (NIIDT), Beijing 100166, China. He is also with Tianjin Artificial Intelligence Innovation Center (TAIIC), Tianjin 300457, China. liruihao2008@gmail.com

Sen Wang is with the Edinburgh Centre for Robotics, Heriot-Watt University, Edinburgh, EH14 4AS, UK. s.wang@hw.ac.uk

Dongbing Gu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK. dgu@essex.ac.uk

constraints are related to the ego-motion and the structure of the environment, which implies that they could be exploited when designing spatio-temporal photometric loss and geometric loss for DL based methods. Inappropriate use of geometric constraints in the loss function could lead to poor performance of the estimation, or even worse, non-convergence of the learning process. On the contrary, an appropriate use could lead to an improved estimation accuracy in an unsupervised learning framework.

Most DL based methods are based on supervised learning schemes which require datasets with annotated ground-truth. However, labeling large amounts of data is difficult and expensive, which limits the potential application scenarios of DL based methods. This is particularly true in the context of visual SLAM because robots typically operate in completely unknown environments. Moreover, it is very interesting for a visual SLAM system to learn under an unsupervised scheme so that the performance could be continuously improved by the increased size of datasets without annotated ground-truth.

In this paper, we propose DeepSLAM, an unsupervised DL based monocular SLAM system. It takes monocular color images as input and outputs pose trajectory, depth map, and 3D point cloud simultaneously (see Fig. ??). Our main contributions are summarized as follows:

- A novel visual SLAM framework based on unsupervised DL is proposed. It exploits the combination of DL and geometric constraints.
- Deep Recurrent Convolutional Neural Network (RCNN) is designed to model the ego-motion by leveraging both spatial and temporal properties of a sequence of stereo images during training.
- DeepSLAM integrates the DL based tracking result and loop closure detection with a graph based optimization mechanism, forming a full visual SLAM system.
- Outlier rejection is handled by the uncertainty derived from error maps of both geometric and photometric consistencies. This improves the robustness performance of DeepSLAM in challenging scenes.

Note that although DeepSLAM uses a stereo setup for training, only monocular vision is required for testing. Therefore, DeepSLAM is a monocular visual SLAM system.

The rest of this paper is organized as follows. Section II reviews related work. The system architecture of the proposed DeepSLAM is provided in Section III, followed by the presentation of training losses and outlier rejection in Section IV. Section V introduces the construction and optimization of pose graph in our DeepSLAM. Section VI provides our experimental results. Finally, our conclusion is drawn in Section VII.

II. Related Work

A. Supervised Deep Learning for Pose Estimation

It is well known that Convolutional Neural Networks (CNNs) are successful in object classification. A novel idea is to use CNNs for pose regression. [?] proposed PoseNet to solve the pose regression problem with a CNN. It was trained with ground-truth poses and could be used for relocalization scenarios. [?] then extended PoseNet to include a depth network in order to enhance the relocalization performance in challenging environments. [?] incorporated a spatial LSTM module into PoseNet to improve the performance. In order to estimate the uncertainty of pose estimation, [?] proposed Bayesian PoseNet by considering the Dropout in the network as a means of sampling. [?] proposed to use a RCNN to implement the pose regression with video clips by taking the temporal information into consideration.

Apart from pose regression, the ego-motion between two image frames could be estimated by using DL inspired by stereo geometric models. [?] developed a CNN to estimate the ego-motion with supervised training. [?] adopted a CNN to estimate the transformation between two consecutive frames. [?] and [?] trained a RCNN to estimate the camera motion. [?] presented a relative camera pose estimation system with CNN. [?] constructed a metric net for ego-motion estimation and a topological net for topological location estimation. Then the predictions from these two networks were combined by a successive optimization. [?] proposed "DeMoN" to estimate ego-motion, image depth, surface normal and optical flow simultaneously, but the ground-truth data was required. [?] developed one network to predict the locations of feature points and another network to compute the homography with manually synthesized data. Instead of the above end-to-end pose estimation with CNNs, [?] employed a CNN to estimate the depth map, then used a geometric model-based SLAM system to perform the pose estimation. [?] combined a CNN-based depth prediction method with ORB-SLAM [?] to overcome the scale problem for monocular SLAM systems. [?] fused sparse map points from the SLAM system and dense predicted maps from CNN to deal with depth boundaries in 3D reconstruction. [?] proposed a 3D reconstruction system called Deepfusion. They introduced predicted dense depth maps into ORB-SLAM [?], and adopted the estimated depth gradients of keyframes as a constraint to ensure the global consistency of the 3D reconstruction.

B. Unsupervised Deep Learning for Depth and Ego-motion Estimation

The main problem in the supervised pose estimation systems is the requirement of a large amount of data with annotated ground-truth to train the networks. Currently, the size of datasets with annotated ground-truth is limited and they are costly to collect. This restrains the performance of the supervised learning systems from further improvement. Recently unsupervised DL methods were successfully applied for depth estimation, inspired by the image wrap technique "spatial transformer" [?]. [?] proposed an unsupervised depth estimation method by exploiting left-right photometric constraint in stereo image pairs. The network training was fully unsupervised in an end-to-end manner and it even outperformed some supervised methods in terms of accuracy of depth estimation. [?] proposed a 2D-to-3D video conversion method with unsupervised learning. [?] improved the method of Garg et al. by wrapping left and right images across each other. [?] proposed SfMLearner, which used a monocular image

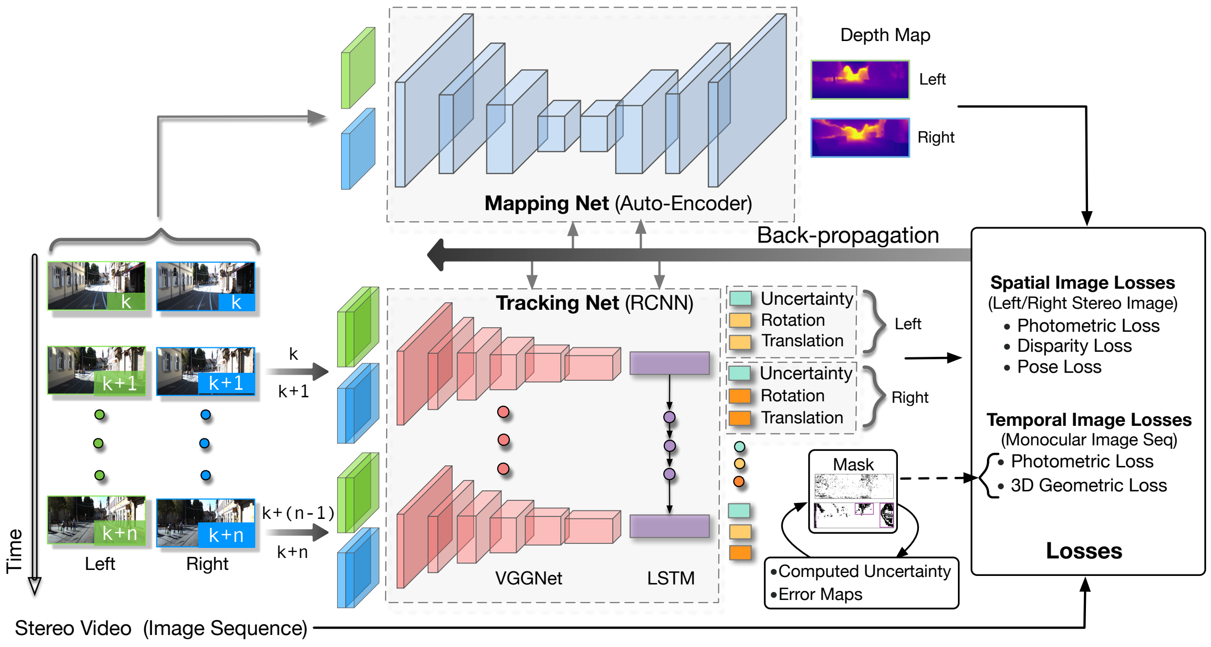


Fig. 2: Training scheme of the proposed DeepSLAM system. We use stereo images to train the system in order to recover the scale information of the environment which is similar to ones used in training. The spatial image losses between a stereo image pair and the temporal image losses between a sequence image pair are formulated to train the networks. The error map produced from the system is used as the loss masks for outlier rejection. The uncertainty produced from the system is also used to train the networks.

sequence for image alignment in order to estimate the depth and ego-motion simultaneously with unsupervised learning. However, the estimated depth map and ego-motion lacked scale information. [?] proposed SfM-Net which added motion masks to the photometric loss. It could estimate optical flow, depth map and ego-motion simultaneously. [?] proposed a visual odometry method based on unsupervised learning. It used stereo images to train and monocular images to test, which could recover the scale of estimated trajectories.

Outlier rejection, loop closure detection [?], [?] and pose graph optimization [?] are very important components for visual SLAM systems to reduce cumulative errors. However, only few papers introduced them into DL-based visual odometry systems. Recently, DL-based methods have also achieved a great success in place recognition and loop closure detection [?], [?]. It is essential to combine DL-based loop detection methods with DL-based visual odometry systems to improve the accuracy performance.

In summary, unsupervised DL techniques are promising a new research trend within the visual SLAM research field, potentially producing a new paradigm of visual SLAM systems and further improving their performance.

III. System Overview of DeepSLAM

According to the DeepSLAM testing framework in Fig. ??, the trained Tracking-Net, Mapping-Net and Loop-Net can be viewed as the front-end yielding a pose graph from a monocular image sequence. Specifically, Tracking-Net is a RCNN architecture constructed from a CNN part of the VGGNet [?] and a Recurrent Neural Network (RNN) to estimate poses and uncertainties, Mapping-Net is an encoder-decoder architecture

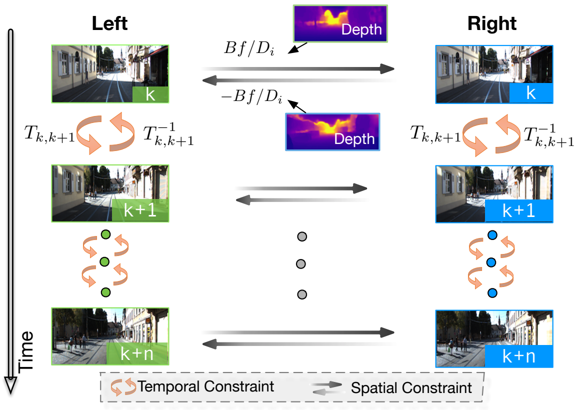


Fig. 3: Spatial and temporal constraints used to formulate the spatial and temporal losses.

to produce dense depth maps, and Loop-Net produces sparse feature vectors for loop closure detection. Meanwhile, the pose graph optimization is employed to refine the poses as the back-end.

The training scheme of DeepSLAM is shown in Fig. ??. Tracking-Net and Mapping-Net are trained unsupervisedly using stereo image pairs and geometric losses in Section ??. The purpose of using stereo image pairs instead of monocular ones for training is to recover the scale information of the environment. We found in our experiments that the scale information can be recovered when the environments of training and testing are similar. Loop-Net is a pre-trained CNN for identifying loop closures.

As shown in Fig. ??, we utilize both spatial and temporal geometric consistencies of stereo image sequences to formulate the loss function. The spatial geometric consistency represents the geometric projective constraint between the corresponding points in left-right image pairs, while the temporal geometric consistency represents the geometric projective constraint between the corresponding points in two consecutive monocular images. By using these constraints to construct the loss functions and minimizing them all together, the networks learn to estimate scaled 6-DoF poses and depth maps in an end-to-end unsupervised manner. We are now in the position to discuss the details of various loss functions used for training.

IV. Unsupervised Training based on Spatial and Temporal Geometric Consistencies

This section describes the losses designed for training Mapping-Net and Tracking-Net. In general, there are two kinds of losses to be minimized for training: spatial image loss and temporal image loss. The relationship between these two losses and a sequence of stereo images is shown in Fig. **??**.

A. Spatial Image Loss of a Pair of Stereo Images

The spatial image loss exploits the geometric constraints (shown in Eq. (1)) between stereo images to enable the Mapping-Net to produce meaningful depth maps which contain the scale information. For a pair of stereo images, every overlapped pixel i in one image can find its correspondence in the other image with a horizontal distance H_i [**?**]. Given its depth value D_i, the distance H_i can be calculated by

$$H_i = Bf/D_i \tag{1}$$

where B is the baseline of the stereo camera and f is the focal length. Therefore, by using the predicted depth map $\widehat{D}_i$ from the Mapping-Net, a distance map H can be generated for the whole image. Based on H, we can synthesize a new image by warping an image from the other through spatial transformer [**?**]. Assume I'_l and I'_r are the synthesized left and right images from original right image I_r and left image I_l, respectively.

The left-right photometric consistency losses can be constructed as

$$L^p_{l,r} = \sum \lambda_s f_s(I_l, I'_l) + (1-\lambda_s)\left\|I_l - I'_l\right\|_1 \tag{2}$$

$$L^p_{r,l} = \sum \lambda_s f_s(I_r, I'_r) + (1-\lambda_s)\left\|I_r - I'_r\right\|_1 \tag{3}$$

where $\|\cdot\|_1$ is the L1 norm, $f_s(\cdot) = (1 - SSIM(\cdot))/2$ and $SSIM(\cdot)$ is the Structural SIMilarity (SSIM) metric to evaluate the quality of a synthesized image [**?**], [**?**] with weight λ_s.

1) Disparity Consistency Loss: The disparity map is defined by

$$Q = H \times w \tag{4}$$

where w is the image width. Therefore, the estimated left and right disparity maps can also be constrained by H. Denote Q_l and Q_r as the left and right disparity maps, respectively. Similar to the photometric consistency loss, we can use H to synthesize Q'_l, Q'_r from Q_r, Q_l, respectively. By using these disparity maps, the disparity consistency losses can be constructed as

$$L^d_{l,r} = \sum \left\|Q_l - Q'_l\right\|_1 \tag{5}$$

$$L^d_{r,l} = \sum \left\|Q_r - Q'_r\right\|_1 \tag{6}$$

2) Pose Consistency Loss: If left and right image sequences are employed separately to estimate the 6-DoF transformations of camera motion through the Tracking-Net, ideally these relative translations should be approximate, and rotations should be exactly the same. Therefore, the differences between these two groups of pose estimates can be introduced as a left-right pose consistency loss as

$$L^o = \lambda_p \left\|\widehat{\mathbf{x}}_l - \widehat{\mathbf{x}}_r\right\|_1 + \lambda_r \left\|\widehat{\varphi}_l - \widehat{\varphi}_r\right\|_1 \tag{7}$$

where $[\widehat{\mathbf{x}}_l, \widehat{\varphi}_l]$ and $[\widehat{\mathbf{x}}_r, \widehat{\varphi}_r]$ are the estimated poses from left and right image sequences by the Tracking-Net, respectively. λ_p and λ_r are the position and rotation weights, and λ_p is much smaller than λ_r. Note that the length of image sequence can be variable thanks to the recurrent network in the Tracking-Net.

B. Temporal Image Loss of a Sequence of Monocular Imagery

The temporal image loss exploits the geometric constraints (namely ego-motions) among multiple views of a monocular image sequence to enable the Mapping-Net to produce meaningful depth maps and the Tracking-Net to estimate the camera motion.

As shown in Fig. **??**, the RCNN architecture enables correlation between two consecutive monocular images. It includes photometric consistency loss and 3D geometric registration loss.

1) Photometric Consistency Loss: Different from the previous photometric consistency loss of a pair of stereo images, the photometric loss here focuses on the temporal information among a monocular image sequence. For each image pair I_k, I_{k+1} with some scene overlaps, we can obtain their synthesized images I'_k and I'_{k+1} by using a spatial transformer network [**?**]. Specifically, for an overlapped pixel p_k in the kth frame, we can derive its corresponding pixel p'_{k+1} in the $(k+1)$th frame through

$$p'_{k+1} = K\widehat{T}_{k,k+1}\widehat{D}_k K^{-1} p_k \tag{8}$$

where K is the camera intrinsics matrix, $\widehat{D}_k$ is the pixel's depth estimated from the Mapping-Net, $\widehat{T}_{k,k+1}$ is the camera coordinate transformation matrix from the kth frame to the $(k+1)$th frame predicted by the Tracking-Net. Based on this, I'_k and I'_{k+1} can be constructed from I_{k+1} and I_k, respectively. Define a temporal photometric error map between an image I_k and its synthesized image I'_k as $E^k_p = I_k - I'_k$. Then, the photometric error maps for the k-to-$(k+1)$ and $(k+1)$-to-k consistencies are

$$E^k_p = I_k - I'_k, \quad E^{k+1}_p = I_{k+1} - I'_{k+1} \tag{9}$$

Then, the photometric losses of an image pair I_k, I_{k+1} from the monocular image sequence are

$$L^p_{k,k+1} = \sum M^k_p \left(\lambda_s f_s(I_k, I'_k) + (1-\lambda_s)\left\|E^k_p\right\|_1\right) \tag{10}$$

$$L^p_{k+1,k} = \sum M^{k+1}_p \left(\lambda_s f_s(I_{k+1}, I'_{k+1}) + (1-\lambda_s)\left\|E^{k+1}_p\right\|_1\right) \tag{11}$$

where M_p denotes the bitwise mask of the corresponding photometric error map. We will discuss the mask in Section **??**. Note that frames k and $k+1$ are not necessarily consecutive but they should have overlapped pixels. Since DeepSLAM has a RNN architecture, the photometric losses are determined by several pairs of images in the image sequence, which facilities the construction of local graph. See more details on local graph in Section **??**.

2) 3D Geometric Registration Loss: Geometric loss is used to constrain and estimate the transformations by considering 3D point clouds. It is similar to Iterative Closest Point (ICP), a well-known method to align point clouds. In DeepSLAM, we also use this loss for pose estimation.

Assuming P_k and P_{k+1} are the 3D point clouds in the kth and $(k+1)$th camera coordinations, and P'_k and P'_{k+1} are the transformed point clouds in these two coordinate frames. Then we construct the geometric losses in the monocular image sequence as

$$L^g_{k,k+1} = \sum M^k_g \left\| E^k_g \right\|_1 \tag{12}$$

$$L^g_{k+1,k} = \sum M^{k+1}_g \left\| E^{k+1}_g \right\|_1 \tag{13}$$

where $E^k_g = P_k - P'_k$ and $E^{k+1}_g = P_{k+1} - P'_{k+1}$ are the temporal geometric error maps, and M_g is the mask of the corresponding geometric error map. For P_k, P'_k, P_{k+1} and P'_{k+1}, they are all tensors with size $h \times w \times 3$, where h is the height of images, w is the width of images, and 3 denotes (x,y,z) in the camera coordination. E^k_g and E^{k+1}_g are obtained through element-wise point cloud subtraction.

With regards to existing works, [**?**] and [**?**] used left-right photometric loss to estimate depth map, [**?**] used left-right disparity consistency loss, and [**?**] used the photometric loss of image sequence to recover ego-motion and depth. However, little work has explored the combination of all these losses to estimate both scaled camera pose and depth map.

C. Uncertainty Estimation and Outlier Rejection

Uncertainty estimation and outlier rejection are essential in SLAM systems. For the uncertainty estimation of supervised pose regression methods with DL, they either adopt a sampling method by using Dropout [**?**] or add a balance factor to the network as a mixture model [**?**] to obtain the uncertainty of pose estimation. Different from those supervised methods, DeepSLAM can produce the projected photometric error maps E^k_p, E^{k+1}_p and the projected geometric error maps E^k_g, E^{k+1}_g for two consecutive images I_k and I_{k+1}. Assuming μ^k_p, μ^{k+1}_p, μ^k_g, μ^{k+1}_g are the mean of E^k_p, E^{k+1}_p, E^k_g, E^{k+1}_g, respectively. Then, the uncertainty of pose estimation and depth estimation with the kth frame and $(k+1)$th frame can be represented as

$$\sigma_{k,k+1} = 2 \times S(\mu^k_p + \mu^{k+1}_p + \lambda_e(\mu^k_g + \mu^{k+1}_g)) - 1 \tag{14}$$

where $S(\cdot)$ is the Sigmoid function and λ_e is the normalizing factor between the geometric and photometric errors. Because μ^k_p, μ^{k+1}_p, μ^k_g, μ^{k+1}_g are all positive, Sigmoid function here normalizes the uncertainty between 0.5 and 1 to represent the belief on the accuracy of pose estimate. We use $\sigma_{k,k+1}$ to train the uncertainty estimation $\widehat{\sigma}_{k,k+1}$ of the Tracking-Net:

$$L^u_{k,k+1} = \left\| \sigma_{k,k+1} - \widehat{\sigma}_{k,k+1} \right\|_1 \tag{15}$$

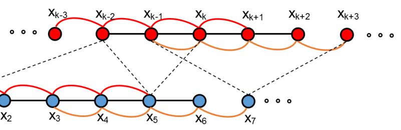


Fig. 4: Pose Graph with local and global connections. The dotted lines represent the global loops detected from Loop-Net, while the solid lines represent the local loops generated by Tracking-Net. The local graph with image sequence length 5 is shown here as an example.

$\widehat{\sigma}_{k,k+1}$ is estimated by the Tracking-Net and represents the uncertainties of estimated poses and depth maps. Intuitively, $\widehat{\sigma}_{k,k+1}$ is small when the estimated poses and depth maps are accurate enough to reduce the photometric and geometric errors.

In real-world environments, the photometric and geometric losses could be corrupted by dynamic objects. Therefore, we introduce the masks for the error maps in the previous temporal losses. We propose a novel method to construct bitwise masks to reject the outlier during training. According to the error values in the error maps, masks are constructed with a percentile q_{th} of pixels as 1 and a percentile $(1-q_{th})$ of pixels as 0. Specifically, based on the uncertainty $\sigma_{k,k+1}$, the percentile q_{th} of the pixels is determined by

$$q_{th} = q_0 + (1-q_0)(1-\sigma_{k,k+1}) \tag{16}$$

where $q_0 \in (0,1)$ is the basic constant percentile. Then, we can construct the bitwise masks M_p and M_g to filter out $(1-q_{th})$ proportion of the big errors (as outliers) in the error maps. The generated masks not only automatically adapt to the different percentage of outlier, but also can be used to infer dynamic objects in the scene. This will be discussed in detail in Section **??**.

V. Pose Graph Construction and Optimization

Pose graph optimization plays an important role in SLAM systems due to its ability to reduce the cumulative pose drifts. In our system, we also perform a pose graph optimization with both local and global pose connections. The local pose graph is built upon a short sequence of consecutive images as a direct result of the recurrent model of the Tracking-Net considering an image sequence, i.e. the local pose graph is constructed from consecutive image frames. The global loop closures are detected by the Loop-Net from the historical images, which are usually non-consecutive.

A. RCNN based Local Pose Graph

The RCNN architecture of the Tracking-Net is able to learn the relationship between CNN features over time as the camera moves, modeling the motion dynamics of the camera from an image sequence. Therefore, based on the structure of Tracking-Net, we can construct the local pose graph directly. Assuming the length of an image sequence is n, each time the Tracking-Net can estimate $(n-1)$ relative poses to build the local pose graph. An example of the pose graph built with sequence length 5 is shown in Fig. **??**. As the camera moves over time,

TABLE I: Tracking results on KITTI dataset with our proposed DeepSLAM system.

	Monocular												Stereo	
Seq.	DeepSLAM A+B (416×128)		DeepSLAM A (416×128)		ESP-VO [?] A (1242×376)		SfMLearner [?] A (416×128)		ORB-SLAM [?] (1242×376)		VISO2-M [?] (1242×376)		VISO2-S [?] (1242×376)	
	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$	$t_{\text{rel}}(\%)$	$r_{\text{rel}}(^\circ)$
03	7.66	**4.30**	7.15	5.13	**6.72**	6.46	15.82	7.03	0.67	0.18	8.47	8.82	3.21	3.25
04	**4.56**	**1.90**	5.22	2.27	6.33	6.08	5.25	2.32	0.65	0.18	4.69	4.49	2.12	2.12
05	**3.25**	**1.31**	4.04	1.40	3.35	4.93	13.47	4.41	3.28	0.46	19.22	17.58	1.53	1.60
06	**4.97**	**1.53**	5.99	1.54	7.24	7.29	30.32	7.47	6.14	0.17	7.30	6.14	1.48	1.58
07	4.71	**1.84**	4.88	2.14	**3.52**	5.02	15.08	5.96	1.23	0.22	23.61	29.11	1.85	1.91
10	**8.35**	**3.93**	10.77	4.45	9.77	10.2	18.45	6.77	7.27	1.00	41.56	32.99	1.17	1.30
mean	**5.58**	**2.47**	6.34	2.82	6.15	6.66	16.40	5.99	3.21	0.37	17.48	16.52	1.89	1.96

- t_{rel}: average translational RMSE drift $(\%)$ on length of 100m-800m.
- r_{rel}: average rotational RMSE drift $(^\circ/100m)$ on length of 100m-800m.
- A: KITTI sequence 00-02, 08, 09 as the training data.
- B: KITTI sequence 11-21 as the training data.

our system can gradually construct the pose graph with the local loops.

B. CNN based Global Loop Detection

For global pose graph, we use the Loop-Net to perform the place recognition and loop detection between non-neighboring frames. The Loop-Net in our DeepSLAM system is a CNN model pre-trained on the ImageNet dataset for object recognition as it has shown a good performance in learning representations. Note that there is no training involved for the Loop-Net. The Inception ResNet V2 architecture [**?**] is adopted here. The Loop-Net maps images into feature vectors for loop closure detection. We can then compute the cosine distance of two feature vectors from an image pair to detect loop closures:

$$d_{cos} = cos(\mathbf{v_1}, \mathbf{v_2}) \tag{17}$$

where $\mathbf{v_1}$, $\mathbf{v_2}$ are the feature vector representations of an image pair. When d_{cos} is smaller than a threshold d_{cos}^{th}, the image pair is treated as a loop.

After the Loop-Net detects global loops, we use our Tracking-Net to calculate the transformation between detected image pairs. Since the recurrent structure makes the Tracking-Net flexible to the length of image sequence, we can use the sequence length 2 to compute the pose transformation. Once a global loop detected, g2o [**?**] is used as the back-end for pose graph optimization.

VI. Experimental Evaluation

In this section, we demonstrate the tracking and mapping performance of the proposed DeepSLAM system. We conducted the evaluation on pose and depth accuracy separately in order to see how each network performs.

The proposed DeepSLAM was designed with the DL framework TensorFlow and trained on NVIDIA DGX-1 with Tesla P100. The Adam optimizer was employed to train the network for up to 20-30 epochs. The starting learning rate was 0.001 and decreased by half for every 1/5 of total iterations. The parameter β_1 is 0.9 and β_2 was 0.99. The sequence length of images feeding to the Tracking-Net was 5. The image size was 416×128.We also resized the output images to a higher resolution to compute the losses and fine-tuned the networks in the end. For testing, a laptop equipped with a NVIDIA GeForce GTX 980M GPU and Intel Core i7-6820HK 2.7GHz CPU was used. The GPU memory required for the Tracking-Net was less than 400MB with 40Hz real-time performance. For the Mapping-Net and Loop-Net, we performed the dense depth prediction and loop closure detection every 5 frames. It took about 48ms for Mapping-Net to predict a dense depth map per frame, and about 120ms for Loop-Net to encode an image to the corresponding feature vector. The Tracking-Net, Mapping-Net, Loop-Net, and the pose graph optimization unit were run in separate threads. For the whole system, it can run at about 20Hz.

To achieve a better training performance, some measures on data augmentation were taken, such as left-right image augmentation, rotational data augmentation and image color augmentation.

A. Pose Accuracy Performance on KITTI

We first evaluated the accuracy performance of our DeepSLAM system on KITTI Odometry dataset [**?**]. The full system of DeepSLAM includes the Tracking-Net with loop closure detection and graph optimization. The detailed quantitative results are listed in Table **??**. We used the standard evaluation method provided along with KITTI dataset: average translational root-mean-square error (RMSE) drift (%) and average rotational RMSE drift ($^\circ$/100m) on length of 100m-800m. We also added two data-driven learning methods (ESP-VO and SfMLearner) and three model-based methods (monocular ORB-SLAM, monocular VISO2-M and stereo VISO2-S) into the table for comparison. VISO2-M and monocular ORB-SLAM did not work with resolution 416×128, and we used input images with size 1241×376. For stereo methods, VISO2-S also used input images with size 1241×376. All learning based methods (DeepSLAM, ESP-VO and SfMlearner) used KITTI sequences 00-02, 08, 09 for network training, and KITTI sequences 03-07, 10 as the testing datasets. Our DeepSLAM is an unsupervised learning method and does not need the ground truth for training. In order to show the advantage of unsupervised learning and fully draw out the potential of DeepSLAM, we also used KITTI sequences 00-02, 08, 09, 11-21 to train the network. The best tracking results among learning methods are made in bold.

As shown in the table, our DeepSLAM outperforms ESP-VO and SfMLearner in terms of tracking accuracy. When compared with ESP-VO, we used more datasets (KITTI sequences 11-21) for network training as our DeepSLAM does not need

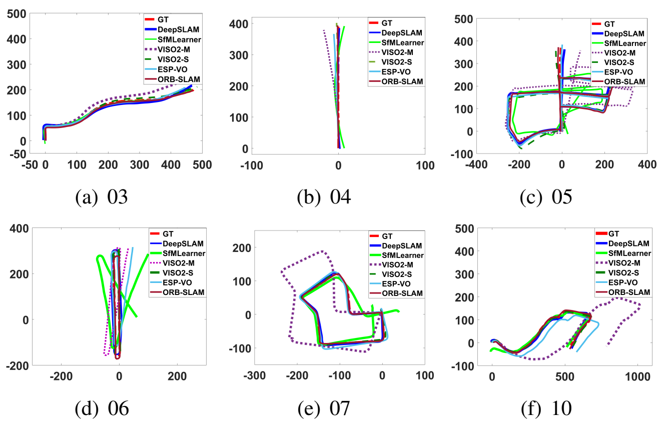


Fig. 5: Trajectories on KITTI sequences 03-07, 10 using our DeepSLAM system (best viewed in color). Trajectories with ESP-VO [?], SfMLearner [?], ORB-SLAM [?], VISO2-M [?] and VISO2-S [?] are also plotted for comparison.

TABLE II: Robustness performance on challenging scenes of RobotCar dataset. ✓ represents working well, ×represents losing tracking.

Methods	Dataset	Environments			
		Overcast	Sun	Rain	Night
LSD-SLAM [?]	RobotCar	×/✓	×	×	×
ORB-SLAM-S [?]	RobotCar	✓	×	×	×
DeepSLAM	RobotCar	✓	✓	✓	✓

datasets with annotated ground truth. ESP-VO is a supervised learning method and it can not use KITTI sequences 11-21 for training. The result indicates that unsupervised learning methods can use more datasets for training and make the benefit in performance from it. When compared with SfMLearner, the DeepSLAM system adopts more carefully designed spatial and temporal losses functions and takes RCNN as Tracking-Net architecture. The DeepSLAM also outperforms monocular VISO2-M, but its performance is not as good as ORB-SLAM and stereo VISO2-S as DeepSLAM cannot maintain the local map and global map like ORB-SLAM. The estimated trajectories on sequences 03-07, 10 with above methods are plotted in Fig. **??**. As shown in the figure, the trajectories from the DeepSLAM achieve good performance in terms of pose estimation.

In order to perform more experiments and comparisons, we also used KITTI sequences 00-08 for network training and the rest sequences for testing. There are no ground-truth provided for KITTI sequences 11-21, thus we plotted trajectories with stereo ORB-SLAM (ORB-SLAM-S) for reference. The trajectories of sequences 13, 15-19 are plotted in the figure. As shown in Fig. **??**, the trajectories produced by our DeepSLAM are similar with the ones produced by stereo ORB-SLAM. In order to highlight the role of the loop closure in localization, we have added the results of our proposed system without Loop-Net in Fig. 6. As shown in the figure, for sequences 13, 15, 16, 18, 19 which have loops, the system without Loop-Net can not achieve the same performance as with loop closure due to the fact that the accumulated errors can not be reduced.

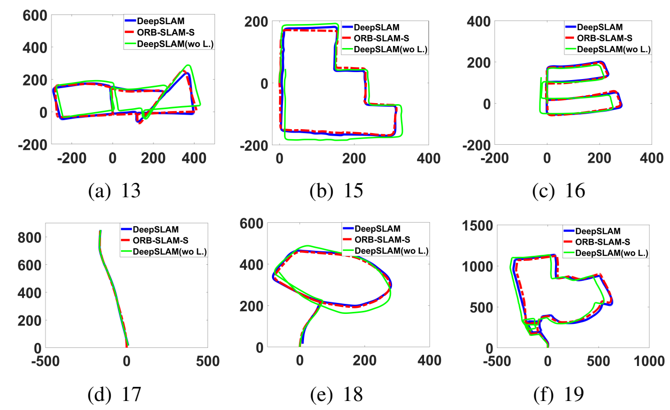


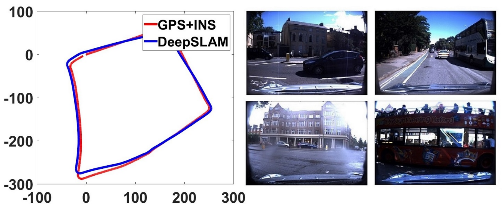
Fig. 6: Trajectories on KITTI dataset using our DeepSLAM system. The trajectories using our DeepSLAM system without Loop-Net are also given. There is no ground-truth for these trajectories. We plotted trajectories with stereo ORB-SLAM (ORB-SLAM-S) for reference. KITTI sequences 00-08 are used for network training.

B. Robustness Performance on Challenging Scenes

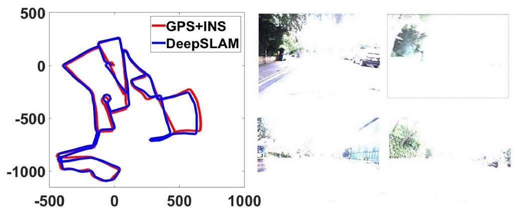
Robustness is a significant factor for wider applications of visual SLAM. We used the public RobotCar [?] dataset to test the robustness of the proposed DeepSLAM system. The RobotCar dataset was collected in different environments over 1 year. We chose some datasets collected in challenging environments to test our trained models. Dataset (b) in Fig. **??** was used for training the Tracking-Net and Mapping-Net.

As shown in Fig. **??**, the challenging scenes include image distortion, excessive exposure, night-time, bad white balance, and raining. These scenes are difficult for visual SLAM systems to perform accurately and robustly. Model-based SLAM methods (such as LSD-SLAM and ORB-SLAM) are sensitive to camera parameters, and not robust when performing feature extraction or transformation estimation under the above situations. So these methods are fragile when faced with challenging scenes. Fig. **??**(a) is the situation with image distortion. Fig. **??**(b) is the situation that there is excessive exposure in parts of the trajectory. Fig. **??**(c) is the situation that the data is collected at night while raining. Fig. **??**(d) is the situation that the images are collected while raining. No ground-truth is provided. We compared our results with the trajectories collected by GPS/INS. For Fig. **??**(c), the GPS signal was poor due to the rain. For Fig. **??**(d), the GPS/INS almost did not work due to the terrible weather, and we plotted the trajectory with our DeepSLAM in Google map for reference. As shown in Fig. **??**, the DeepSLAM demonstrated a resilient behavior when encountering these challenging scenes.

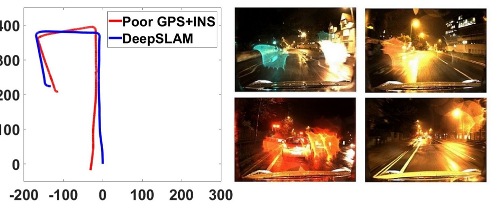
Further, the right columns of the images in Fig. 8 provide the estimated depth images for the RoboCar dataset. It can be seen that the Mapping-Net demonstrates a robust performance on depth estimation (i.e., reconstruction/mapping of the environments) even facing challenging scenes, such as night-time and raining with under- and over-exposure. For example, the depths of the cars on the right parts of both the 6th and the 7th images in Fig. 8 can be accurately estimated. Therefore, the Mapping-Net provides reliable scene



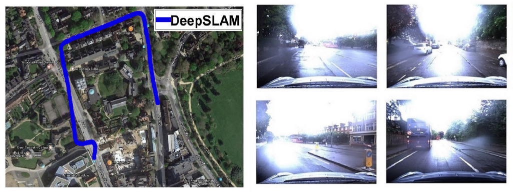
(a) 2014-06-26-08-53-56



(b) 2015-08-12-15-04-18



(c) 2014-11-21-16-07-03 (seq. 10)



(d) 2015-05-29-09-36-29 (seq. 06)

Fig. 7: Robustness performance of our DeepSLAM on some challenging environments in RobotCar dataset. The left part of each subfigure shows the trajectory produced from our DeepSLAM, and the right part is the testing images taken in different locations. (a) Images with distortion, (b) Images with excessive exposure, (c) Images collected at night while raining, (d) Images collected while raining. Note that the GPS and INS data is very poor in RoboCar dataset 2015-05-29-09-36-29 (seq. 06) due to raining.

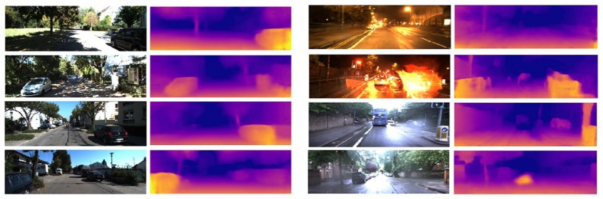


Fig. 8: The estimated depth maps with our Mapping-Net. The left two columns are the images from the KITTI dataset. The right two columns are the images from challenging scenes in the RobotCar dataset.

mapping which is used by the Tracking-Net to enhance the robustness for pose estimates during training. Meanwhile, as shown in Fig. 7, the Tracking-Net can keep estimating the 6-DoF poses when facing challenging environments in the RoboCar dataset, leveraging the accurate 3D reconstruction from the Mapping-Net. In summary, it is believed that the Tracking-Net and Mapping-Net both play an important role in improving the robustness and work in tandem when facing challenging scenes.

Table **??** shows the results of DeepSLAM, LSD-SLAM and stereo ORB-SLAM on challenging scenes of RobotCar dataset. When encountering the challenging scenes such as raining, night-time and bad white balance, LSD-SLAM and ORB-SLAM can hardly work, but our DeepSLAM works well by exploiting the prior knowledge learned through training.

C. Testing with a Low-cost Camera

We also used a low-cost stereo camera ZED to collect some data ourselves and tested our system. We used a laptop, a cheap GPS and a ZED camera to collect outdoor data. No other types of equipment were used, and we did not have the ground-truth. We used the GPS data to provide the reference. The trajectories from our DeepSLAM and GPS are plotted in Fig. **??**. The weather was cloudy when we collected the data, so the images are quite dim. As shown in Fig. **??**, the DeepSLAM works well with this low-cost camera.

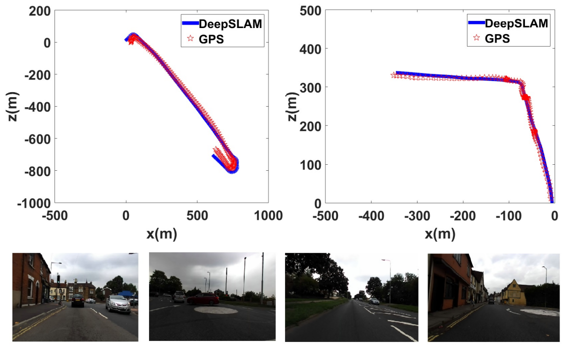


Fig. 9: Testing on our self-collected dataset using a low-cost ZED camera.

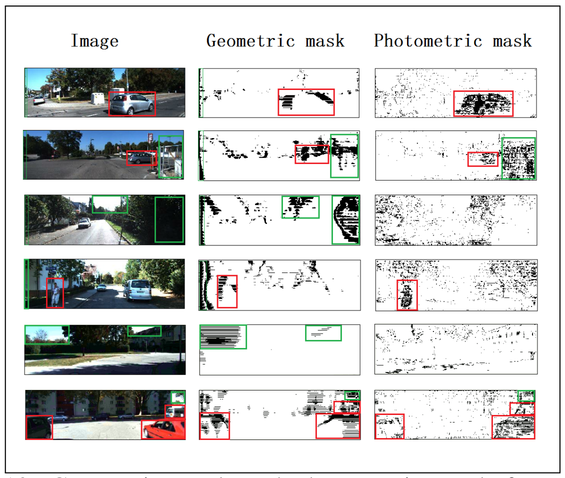
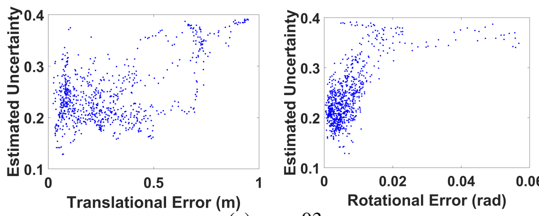


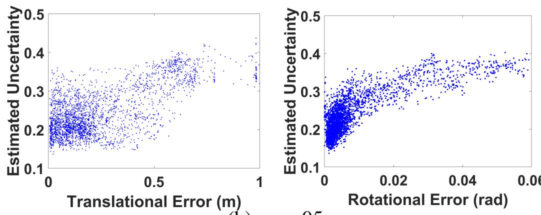
Fig. 10: Geometric mask and photometric mask for outlier rejection. The red boxes represent moving objects and the green boxes represent depth values with high uncertainty.

TABLE III: Depth estimation results on KITTI using the split of Eigen et al. [**?**].

Methods	Dataset	Input size	Supervision	Scale	Network	Capped depth	Error metric			
							Abs Rel	Sq Rel	RMSE	RMSE log
Eigen [**?**]	K (raw)	612×184	✓	✓	VGG	80m	0.214	1.605	6.563	0.292
Monodepth [**?**]	K (raw)	512×256	×	✓	ResNet-50	80m	0.148	1.344	5.927	0.247
SfMLearner [**?**]	K (raw)	416×128	×	×	VGG	80m	0.208	1.768	6.856	0.283
SfMLearner [**?**]	K (raw)	416×128	×	×	VGG	50m	0.201	1.391	5.181	0.264
DeepSLAM	K (odo)	416×128	×	✓	VGG	80m	0.1724	1.659	6.362	0.259
DeepSLAM	K (odo)	416×128	×	✓	VGG	50m	0.164	1.288	4.782	**0.204**
DeepSLAM	K (odo)	416×128	×	✓	VGG	30m	**0.147**	**0.834**	**3.031**	0.207



(a) seq. 03



(b) seq. 05

Fig. 11: The estimated uncertainty against the corresponding translational and rotational errors. It shows that the estimated uncertainty values are strongly correlated with both of them.

D. Outlier Rejection

As introduced above, We used the photometric error maps and geometric error maps (3D point-cloud registration error maps) from monocular image sequences to generate the loss mask and uncertainty. The loss mask can reject the outlier and refine the estimated poses. This is due to the fact that the 3D registration error map includes depth and pose estimation information, and the photometric error map includes photometric information, depth and pose estimation information. The uncertainty is related to the mean of the error map, which is used to automatically select the size of the mask.

Fig. **??** shows the 3D registration error mask and the photometric error mask. The red boxes in the figure are moving objects in the scenes, such as pedestrians and vehicles. The green boxes in the figure are depth values with high uncertainty. These values tend to be sky, extreme dark places, edges of objects or non-overlap areas of left-right images. It is very hard for the network to estimate the real depth value of these places, and therefore the estimated depth value has a high uncertainty. In our system, the photometric mask is directly related to moving objects, and the geometric mask is related to estimated depth values with high uncertainty. We also plot the estimated uncertainty against the corresponding translational and rotational errors in Fig. **??**. As shown in the figure, the estimated uncertainty values from our network are strongly correlated with both of them.

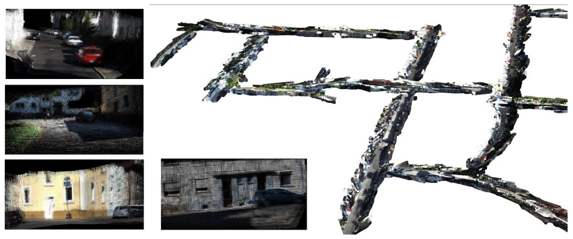


Fig. 12: Reconstructed 3D map with our DeepSLAM system.

E. Depth Estimation and 3D Reconstruction

The Mapping-Net of the DeepSLAM can also produce the scaled dense depth map. Fig. **??** shows some raw RGB images and the dense depth maps generated by the DeepSLAM. The left two columns are the estimated depth maps selected from KITTI dataset, and the right two columns are the estimated depth maps of the challenging scenes selected from RobotCar dataset. As shown in the figure, the depths of cars, trees and trunks are clearly visible.

The detailed quantitative depth estimation results are listed in Table **??**. RobotCar dataset does not provide the ground-truth for depth maps. We used KITTI dataset to evaluate the performance of the Mapping-Net quantitatively. As shown in the table, the DeepSLAM outperforms the supervised one [**?**] and the unsupervised one without scale [**?**], but performs not as good as [**?**]. This could be caused by a few reasons. First, we only used parts of KITTI dataset (KITTI odometry dataset) for training while all other methods use full KITTI dataset to train their networks. Second, [**?**] used higher resolution (512 × 256) input and a more sophisticated Deep Neural Network (ResNet-based architecture). Third, the temporal image loss we used could introduce some noise (such as moving objects) for depth estimation.

Exploiting the powerful ability of pose estimation and depth estimation with the DeepSLAM system, we can also reconstruct the dense 3D point-cloud of the scenes with a monocular camera. Fig. **??** shows some 3D dense maps generated by the DeepSLAM system.

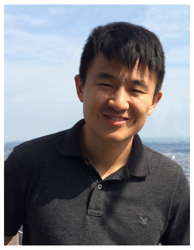
VII. Conclusions

Most state-of-the-art SLAM algorithms rely on geometric models and optimization engines to estimate the structure of environment and the motion of camera. This paper approaches to the problem from a data-driven perspective, i.e. training deep neural networks with existing data sets. The system architecture of DeepSLAM mimics that of model-based SLAMs. The important geometric models and constraints are respected and embedded into the network architecture and the loss function. This provides a guarantee for estimation accuracy.

Our evaluation results show the data-driven approach DeepSLAM achieves good performance in terms of accuracy and robustness. DeepSLAM falls within an unsupervised learning framework as no manual annotation is required for training. This distinguishes itself from supervised deep leaning approaches to SLAM. In the future, more unlabeled data sets will be made available as they are easy to collect. It is expected the DeepSLAM will have the opportunity to further improve the performance. Following this direction, our future work is planned to train DeepSLAM with more data.

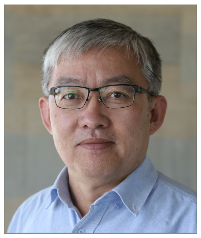
Acknowledgment

The authors would like to thank Robin Dowling and Ian Dukes for their support in collecting our dataset with ZED camera.



Ruihao Li received the B.Sc. degree in automation from Beijing Institute of Technology, Beijing, China, in 2012 and the M.Sc. degree in control science and engineering from National University of Defense Technology, Changsha, China, in 2014. In 2018, he received the Ph.D. degree in robotics from University of Essex, UK. Currently, he is an Assistant Professor at Artificial Intelligence Research Center, National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing, China. His research interests are in robotics, SLAM, deep learning, semantic scene understanding.

Sen Wang is an Assistant Professor in Robotics and Autonomous Systems at Heriot-Watt University and a faculty member of the Edinburgh Centre for Robotics. Previously, he was a postdoctoral researcher at the University of Oxford. His research focuses on robot perception and autonomy using probabilistic and learning approaches, especially autonomous navigation, robotic vision, SLAM and robot learning.



Dongbing Gu received the B.Sc. and M.Sc. degrees in control engineering from Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in robotics from University of Essex, Essex, UK. He was an Academic Visiting Scholar with the Department of Engineering Science, University of Oxford, Oxford, UK, from October 1996 to October 1997. In 2000, he joined the University of Essex as a Lecturer. Currently, he is a Professor with the School of Computer Science and Electronic Engineering, University of Essex. His current research interests include robotics, multiagent systems, cooperative control, model predictive control, visual SLAM, wireless sensor networks, and machine learning.