2019 IEEE 5th International Conference on Mechatronics System and Robots

SLAM for Robotic Navigation by Fusing RGB-D and Inertial Data in Recurrent and Convolutional Neural Networks

Ruixu Liu[1], Ju Shen[1], Chen Chen[2], Jianjun Yang[3]
[1] Interactive Visual Media (IVIDIA) Lab, Department of Computer Science, University of Dayton, Ohio, USA
[2]Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, USA
[3]Department of Computer Science and Information Systems, University of North Georgia, Georgia, USA
e-mail :liur05@udayton.edu; jshen1@udayton.edu; chen.chen@uncc.edu; jianjun.yang@ung.edu

Abstract—**Simultaneous localization and mapping (SLAM) is a key component for mobile robot navigation that enables many service robotic applications. The capacity of acquiring accurate 3D-map of an environment is critical for robots to perform various tasks with a high degree of autonomy. Due to the indoor environment complexity and sensor uncertainties, SLAM remains a challenging task in the domain of 3D reconstruction. In this paper, we propose a simple yet effective solution for RGB-D based SLAM by integrating an Inertial Measurement Unit (IMU) into a recurrent and convolutional neural network that leads to enhanced pose estimation and point cloud registration. The IMU data provide an advantage of fast rate inertial data measurement and drift error reduction. Specifically, by imposing additional constraints from the IMU device, an optimal long-short term memory LSTM) is trained to mitigate scale ambiguity thus improve the concatenated ego-motion estimation. Compared to existing SLAM techniques and recent effort of RNN based solutions for 3D reconstruction, we show that our approach is competitive with high accuracy and robustness.**

Keywords-Simultaneous localization and mapping, 3D Recon- struction, RGB-D Sensing, Recurrent Neural Network (RNN), Inertial Measurement Unit, Long Short-Term Memory (LSTM)

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is one of the most fundamental problems for mobile robots to safely navigate and interact with indoor environment with a high degree of autonomy. The capability of acquiring a compact 3D representation of the robot surroundings enables numerous applications, ranging from navigation and localization to de- livery robot to autonomous surveillance, and boosts the overall productivity of industrial robotics. For the last decade, various types of SLAM systems have been extensively investigated, from early probabilistic methods, e.g. Particle Filters, to depth-sensor based strategies by using global optimization techniques, e.g. KinectFusion based techniques [1], to deep learning approaches, e.g. long-short term memory (LSTM) [20]. Although today these techniques have demonstrated impressive results, most of the system either require strict calibration and synchronization settings, e.g. expensive stereo and LiDAR setups, or suffer from drift or consistency problems in long trajectories when dealing with higher dimensions or degrees of freedom.

Inspired by the recent success of recurrent and convolutional networks models for processing the captured data directly from sensors, we propose a novel RNN-based approach for 3D indoor scene reconstruction. A moving RGB-D camera is used to simultaneously obtain color and depth information per frame and incrementally build the 3D scene through sequential scanning. We choose to use the Kinect sensor as the monocular capture device due to its ubiquitous and low- cost advantage that can be flexibly used in common indoor environment settings such as office, lab, classroom. Instead of using handcrafted features, the RNN framework provides an inherent advantage of learning feature representation directly from the raw data that can be flexibly adapted to various scene settings. To capture long term dependencies in camera motion, sequence-models can be used to robustly estimate the pose transformation between frames. Here we use long-short term memory (LSTM) to provide smooth trajectories by reducing the sensor noises and uncertainties between intermediate transforms. In addition, an inertial measurement unit (IMU) is utilized to mitigate scene scale ambiguity thus increases the estimate accuracy. Our SLAM acquisition device setup is shown in Figure 1. The piece of blue chip mounted on the top of the Kinect is the IMU device, which is low-cost and portable (has a similar size to a quarter). Such features make it

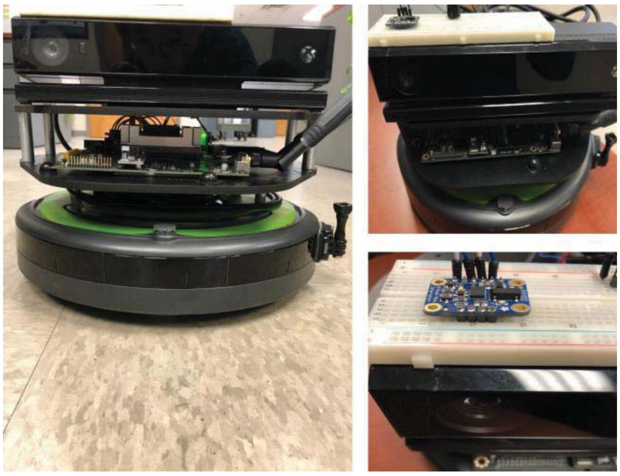


Figure 1. The SLAM system setup: a kinect and an IMU are mounted on a mobile robot.

978-1-7281-2223-6/19/$31.00 ©2019 IEEE

popularly adopted for robots and other mobile devices, e.g. smart phones.

Our major technical contribution is to develop a tailored recurrent and convolutional neural network to effectively reduce scale ambiguity and drift by fusing RGB-D and inertial data into the deep network. To verify the effectiveness of the proposed fusion technique, we use the Technical University of Munich (TUM) RGB-D dataset [3], which has three degrees of freedom (3-DoF) position and orientation estimation. The quantitative and qualitative results demonstrate our cross-modal fusion framework is competitive to state-of-the-art.

II. Related Work

In computer vision and robotics, the technical goal of SLAM systems is to construct a 3D scene of a given environment, which falls into the category of 3D reconstruction as other terminologies often being used alternatively, such as visual odometry (VO) [4], structure from motion (SfM) [5]. A general acquisition process is to use a moving sensor to scan the environment and obtain the ego-motion estimation incrementally, followed by a synchronization procedure to register the acquired 3D points into the same coordinate. One of the most well-studied algorithms for the 3D point cloud registration is the Iterative Closest Point (ICP) algorithm that identifies optimal correspondences between two frames by iteratively refining the rigid transformation matrices [6]. However, the alignment accuracy often depends on the scene content and trajectories. For example, when large planar surfaces present in the capture scene, the ICP methods may significantly compromise on the 3D estimate performance [7]. Such misalignment is caused by the failure of ICP in identifying correct correspondence between planar point clouds and limits its feasibility for globally-consistent alignment in long trajectories. To resolve this issue, numerous methods have been proposed to improve the accuracy by introducing complementary data from other sensors, e.g. fusing sensing depth with RGB values [15], integrating an inertial measurement unit (IMU) to create a visual-inertial odometry (VIO) [8]. The additional cues not only resolve the geometric ambiguity but also reduce the drift error for long trajectories [7]. Early efforts rely on feature detection and matching to enhance the odometry estimates, e.g. SIFT. One major challenge of such visual feature-based approaches is to build a stable representation that has good invariance properties and can be reliably associated.

Recently, a quickly evolving subset of SLAM systems utilizes deep learning solution to learn feature representations automatically instead of handcrafting them. It demonstrates impressive accuracy and robustness for processing raw, high-dimensional data, on which deep convolutional neural networks (CNNs) are employed to extract automated appearance features. Relative pose transformation and generic 3D representation can be reliably learned based on a hybrid CNN architecture [9]. To achieve better temporal consistency, recurrent neural networks (RNNs) are also widely explored for sequential learning and modeling [10]. RNNs has demonstrated incredible success in processing time-series data in many applications, e.g. speech recognition [11]. On the other hand, despite the advance of deep methods, existing SLAM still face two main technical challenges: failure to observe scene scales and long trajectory scale drift [2].

III. The Method

The central idea of our method is to fuse the inertial measurement unit data and the output from our RGB-D based CNN model into a recurrent neural network to leverage the underlying geometry by mitigating scale drift. The CNN model is trained to extract features from the concatenation of two consecutive monocular RGB-D images. It enables us to automatically learn a unique representation from the scanned data. In order to deliver a generic and flexible visual odometry for different indoor situations, we expect a desirable feature representation that can capture the geometric variation combined with texture information. Our CNN structure is inspired by the optical flow estimation network that successfully learns the features encoded with geometrical property from the input data [14]. In addition, since SLAM systems directly operate on sequential images from a moving sensor, transformation constraints arise naturally along with the associated poses. It is essential to enforce temporal and geometric consistency across time steps.

The Inertial Measurement Unit (IMU) is a commonly used electronic device comprised of accelerometers, gyroscopes and sometimes magnetometers. We integrate the IMU into our RNN model to mitigate noises by utilizing the accelerometer and gyroscope. The magnetometer (3-axis compass) is heavily influenced by nearby magnetic fields that cannot be easily modeled. The IMU contributes to speed up the estimation and obtain the velocity meanwhile the absolute scale information can help the depth sensor to estimate the scene depth more reliably. The data speed of IMU is usually 10 times as fast as the one from visual sensors, e.g. 100 Hz vs 10 Hz.

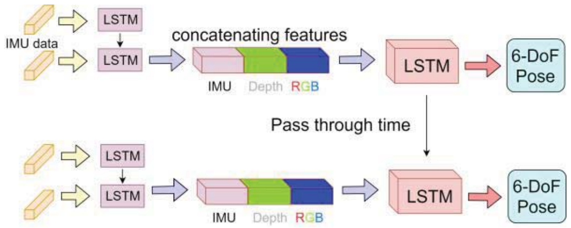


Figure 2. Our fusion pipeline on RGB-D and IMU data

RNNs perform sequential learning with arbitrary lengths that are different from CNN. RNN maintains memory of hidden states over time and has feedback loops, which formulate the current hidden state as a function of the output of its previous state. Hence it inherently encodes temporal constraints between neighbor states in the sequence. The architecture of our proposed RNN model for fusing cross-modal sensing data is shown in Figure 2. Given a convolutional feature X_t at time t, the hidden state ht updates by

$$h_t = \tanh(W_t X\, X_t + W_{hh} h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{yh} h_t + b_y, \quad (2)$$

where h_t and y_t are the hidden state and output at time t respectively, W terms denote corresponding weight matrices, b terms denote bias vectors, and tanh is an activation function hyperbolic tangent. Although the conventional RNN can learn sequences with arbitrary lengths in theory. To void the well-known vanishing gradient problem, it is limited to short sequences in practice. In order to find the correlations among images that are captured along different trajectories, we employ the Long Short-Term Memory (LSTM) [10], which is capable of learning long-term dependencies by introducing memory gates and units [20]. The internal structure of a LSTM unit along with an unfolded LSTM over time is shown in Figure 3.

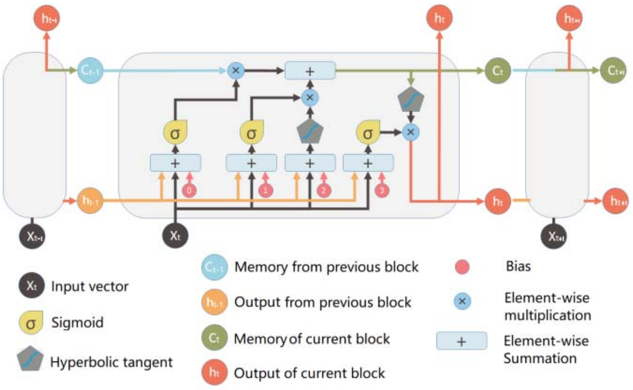

h t-1
h t
h t+1
C t-1
C t
C t+1
σ
σ
σ
+
×
0
1
2
3
h t-1
h t
h t+1
X t-1
X t
X t+1
Xt Input vector
σ Sigmoid
Hyperbolic tangent
C t-1 Memory from previous block
h t-1 Output from previous block
Ct Memory of current block
ht Output of current block
Bias
Element-wise multiplication
+ Element-wise Summation

Figure 3. Internal structure of the LSTM architecture.

It can be seen that the different memory gates control how information is obtained from previous states and passed to the subsequent ones. Given the convolutional feature X_t at time t, the hidden state h_{t-1} and the memory cell C_{t-1} of the previous LSTM unit, the LSTM updates at time t according to:

$$C_t = f_t \circ C_{t-1} + i_t \circ g_t, \quad (3)$$

$$h_t = o_t \circ \tanh(C_t), \quad (4)$$

where ◦ denotes the element-wise product of two vectors; C_t is cell state; h_t is hidden state; tanh is the hyperbolic tangent non-linearity. f_t, i_t, g_t, and o_t are forget gate, input gate, input modulation gate, and output gate at time t, respectively:

$$f_t = \sigma_g(W_{fX}\, X_t + W_{fh} h_{t-1} + b_f) \quad (5)$$

$$i_t = \sigma_g(W_{iX}\, X_t + W_{ih} h_{t-1} + b_i) \quad (6)$$

$$g_t = \tanh(W_{CX}\, X_t + W_{Ch} h_{t-1} + b_C) \quad (7)$$

$$o_t = \sigma_g(W_{oX}\, X_t + W_{oh} h_{t-1} + b_o) \quad (8)$$

σ_g is the sigmoid non-linearity, W terms denote corresponding weight matrices, b terms denote bias vectors.

IV. EXPERIMENTS

A. *Implementation and Architecture Configuration*

Our CNN architecture for multi-views RGB-D data fusion is explained in Figure 4. A sequence of RGB-D frames are taken as the input. At each time step, the RGB images are pre-processed by subtracting the mean RGB values of the training set; the depth images are pre-processed by scaling the depth of each pixel into a range d ∈ [0, 10]. To conveniently accommodate the input data to our CNN, both of the RGB and depth images are normalized into a new dimension 640×448. In such a way, along either dimension, the size is dividable by 64. The configurations of our CNN for processing RGB data is outlined in Table I. In contrast, the depth-based CNN only involves one time convolution.

TABLE I: THE CONFIGURATION OF OUR PROPOSED CNN STRUCTURE

Layers	Kernels size	Strides	Feature size	output channels
Conv1	7	2	320 × 224	64
Conv2	5	2	160 × 112	128
Conv3_1	5	2	80 × 56	256
Conv3_2	3	1	80 × 56	256
Conv4_1	3	2	40 × 28	512
Conv4_2	3	1	40 × 28	512
Conv5_1	3	2	20 × 14	512
Conv5_2	3	1	20 × 14	512
Conv6	3	2	10 × 7	1024

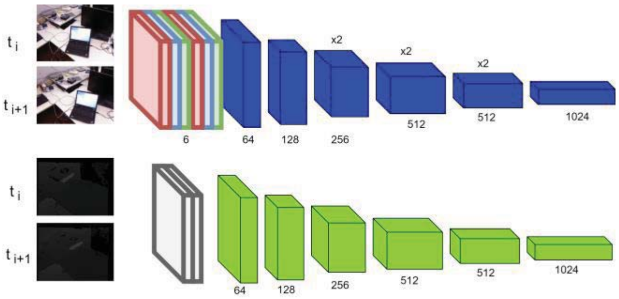


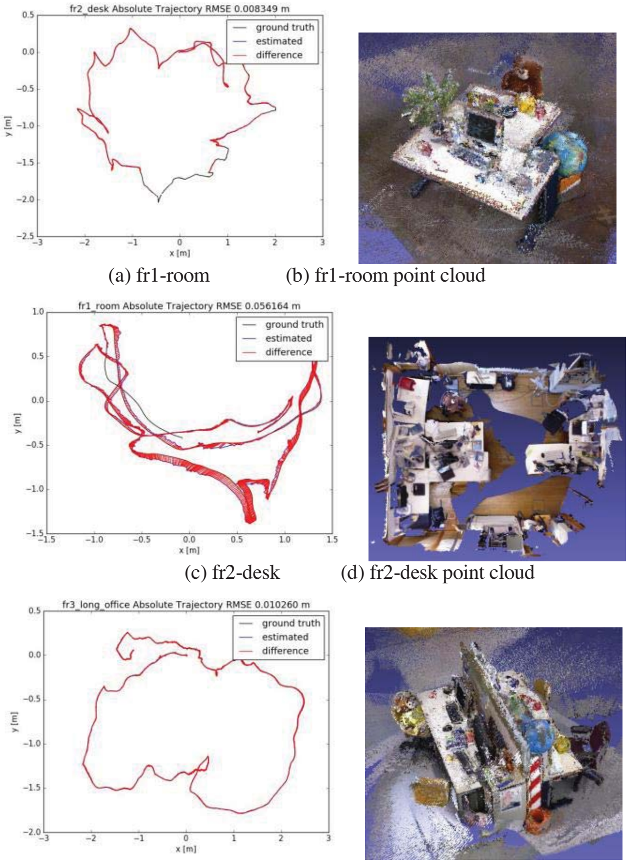
Figure 4. Detailed settings for each CNN layer.

The RGB-based CNN has 9 convolutional layers while the depth-based CNN has 6 layers. All of them are followed by a rectified linear unit (ReLU)activation except Conv6. The sizes of the receptive fields in the network gradually reduce from 7 × 7 to 5 × 5 until 3 × 3 to capture small interesting features. To preserve the spatial dimensions of the tensor after convolution, zero padding is applied, which also favors the receptive field configuration. The number of the channels, i.e., the number of filters for feature detection, increases to learn various feature representations that not only compress the original high-dimensional RGB-D images into a more compacted description, but also boost the successive sequential training procedure. Hence the last convolutional feature Conv6 is passed to the RNN for sequential modeling estimation. The network is implemented based on the TensorFlow tool with KERAS and trained by using a NVIDIA Titan Xp GPU. In order to minimize the training time and necessary data to recover the RGB, depth and IMU data are trained as separated model and also using the pretrained model parameter as the initial weights for the fusion system then concatenated all of features as a long vector to a pose LSTM. The Adam optimizer is employed to train the network for 200 epochs with the learning rate 0.001. Dropout and batch normalization tools are utilized to prevent the models from overfitting.

B. Performance Evaluation

TABLE II: THE QUANTITATIVE EVALUATION BY RMSE

Data Sequence	System Method (m)			
	DVO SLAM	RGB-D SLAM	ORB SLAM2	Ours
fr1-room	0.043	0.087	0.047	0.056
fr2-desk	0.017	0.039	0.009	0.008
fr3-office	0.035	0.032	0.010	0.010



(a) fr1-room (b) fr1-room point cloud

(c) fr2-desk (d) fr2-desk point cloud

(c) fr2-desk (d) fr2-desk point cloud

(e) fr3-office (f) fr3-office point cloud

Figure 5. Experiment result on 3 public datasets: the estimated trajectory against the ground truth (left), the reconstructed 3D scene (right)

To evaluate our approach, we present qualitative, as well as quantitative comparisons with state-of-the-art on the public dataset Technical University of Munich (TUM) [3]. To validate the proposed algorithm, we present the results by measuring the reconstruction errors and providing comparisons to existing methods. The absolute trajectory error (ATE) metrics is adopted to measure the accuracy based on the root-mean- square-error (RMSE) [3]. Since the IMU data is not available in TUM datasets, we use the camera trajectory's ground-truth to simulate inertial information by adding Gaussian noise as σ = 0.5. In Figure 5, we estimate the trajectories (left column) and reconstruct the scene (right column) from three different indoor scenes: f1: training dataset: fr1-360 (e.g. scene objects include floor, desk, desk2), testing dataset: fr1 room; f2: training dataset: fr2-360, test dataset fr2-desk; f3: test dataset fr3-office. It is worth mentioning we did not use the training data for f3 evaluation, as deep neural network has the ability to remember information from the training data. In order to verify the learnability of the proposed algorithm, we apply the training results from f1 and f2 to test f3 directly and shows its reliability to deal with a completely new environment.

In Figure 5, we first compare our estimated trajectory (in blue color) against the ground-truth (in black color) by plotting them in the 3D space, as shown in the left column of sub- figures. The corresponding points are paired up and connected using red line segments. The longer of the red line segment, the bigger of the offset between the correspondences. Though errors in f1 are more noticeable than those in f2 and f3, it overall has a reasonable alignment trajectory against the ground-truth with the average RMSE 0.056, indicated in Table II. For the testing dataset of f2 and f3, our method achieves satisfactory performance with the ground-truth and estimated trajectories almost overlapped leading to very thin red curves. To visually demonstrate the efficacy, we reconstruct the 3D scene together with color attached to the point clouds for as shown in the right column of sub-figures.

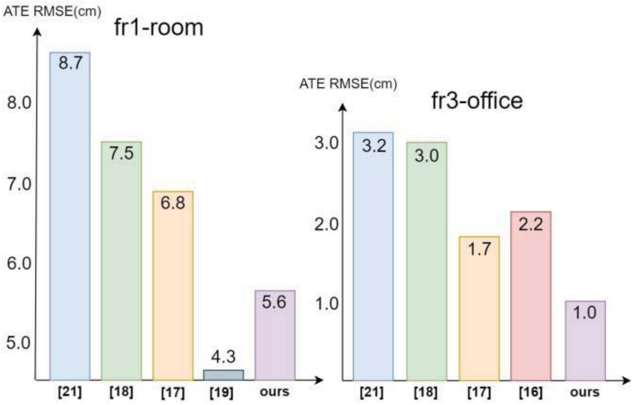


Figure 6. Comparison with state-of-the-art on ATE

In Table II, we give the quantitative evaluation in details by comparing the output with the ones from DVO-SLAM method [19] in term of RMSE. Admittedly, ORB-SLAM2 has achieved a slightly better result than ours in the f1 dataset (RMSE: 0.047 vs 0.056). We conjecture this is because ORB-SLAM system processes all the scanned frames for the 3D reconstruction, while ours only uses a subset from the capture sequence. For our RNN framework, we down sample the data steam rate for RGB and depth acquisition to 10 Hz to cooperate with the IMU (data 100 Hz) by following the integration model [8]. So the performance might be compromised when the scanned camera is moving at a fast speed, e.g. fr1-room, which causes missing details between any two computing frames. For the fr2-desk and fr3-office, a better trajectory estimation is achieved under a relatively slower camera motion. One can see our results outperform all the other approaches, as demonstrated in Figure 6. On the other hand, the reconstructed 3D scene seems degraded compared to the one from fr1- room (Figure 5). This is because loop-closure detection is not involved in out framework. If the camera moves back to its previously navigated areas, slow movement may cause the captured data overlapping to each other that affect the recon- structed 3D scene. So it is a trade-off between minimizing the absolute trajectory error (ATE) and reconstructing a visually satisfactory 3D scene. In addition, we apply our algorithm on different types of real-world objects as shown in Figure 7 to further validate the qualitative performance. We choose two types of objects: a research lab and a person with the former having substantial planar surfaces presence that is a major challenge for traditional ICP based schemes, and the later involving human face that is sensitive for 3D reconstruction. The recovered RGB-D results demonstrate satisfactory 3D surface with the fine details and preserved object contours, e.g. the human face.

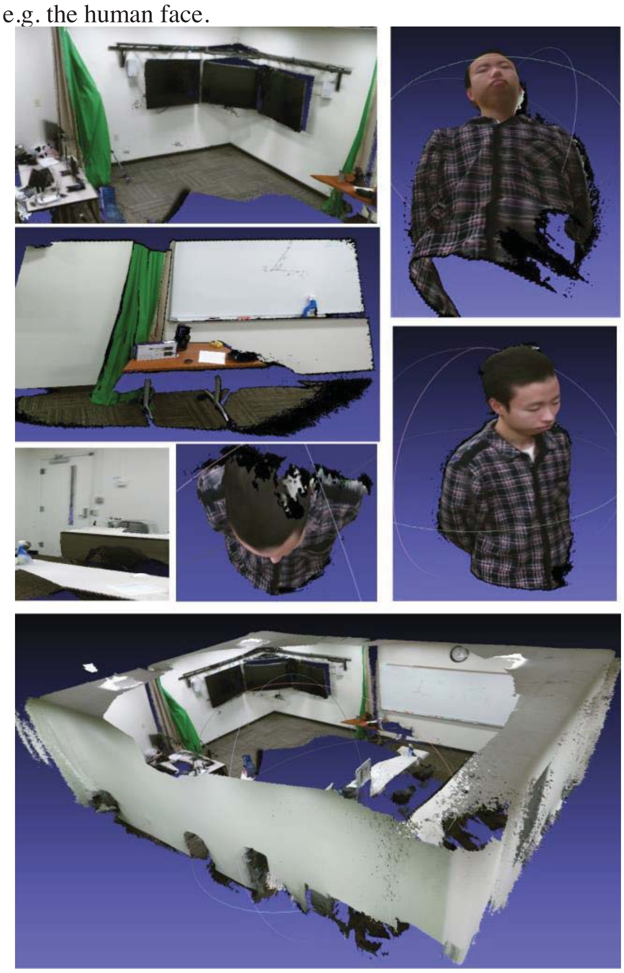


Figure 7. Qualitative evaluation on real-world data: indoor scene and human body 3D reconstruction

V. Conclusion

This paper presents an effective solution to simultaneous localization and mapping for mobile robot navigation. The proposed method leverages the complementary nature of the RGB-D visual data and the inertial measurement unit signals in a recurrent and convolutional neural network framework to achieve more robust pose learning and point cloud registration as compared to using a single modality data only. The experimental results on a benchmark dataset demonstrate the proposed approach has better performance than existing SLAM techniques.

References

[1] R. Liu, V. Asari. “3D indoor scene reconstruction and change detection for robotic sensing and navigation”, Mobile Multimedia/Image Processing, Security, and Applications 2017.

[2] R. Clark, S. Wang, H. Wen, A. Markham, N. Trigoni, “VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem”, Pro- ceedings of AAAI Conference on Artificial Intelligence, 2017.

[3] J. Sturm and N. Engelhard and F. Endres and W. Burgard and D. Cremers, “A Benchmark for the Evaluation of RGB-D SLAM Systems”, Proc. of the International Conference on Intelligent Robot Systems (IROS), 2012.

[4] K. Konda and R. Memisevic. ”Learning visual odometry with a convo- lutional network”, International Conference on Computer Vision Theory and Applications, VISAPP, 2015.

[5] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. ”Sfm-net: Learning of structure and motion from video”, CoRR, abs/1704.07804, 2017.

[6] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” International Conference on 3-D Digital Imaging and Modeling, 2001, pp. 145 –152, 2001.

[7] P. C. Su, J. Shen and S. C. S. Cheung, “A robust RGB-D SLAM system for 3D environment with planar surfaces,” IEEE International Conference on Image Processing, pp. 275-279, 2013.

[8] C. Forster, L. Carlone, F. Dellaert, and Scaramuzza, D. ”IMU prein- tegration on manifold for efficient visual-inertial maximum-a-posteriori estimation”, In Robotics: Science and Systems XI, 2015.

[9] I. Melekhov, J. Kannala and E. Rahtu. ”Relative camera pose estima- tion using convolutional neural networks”, In: Proceedings of advanced concepts for intelligent vision systems (ACIVS), 2017.

[10] J. Donahue, L. Hendricks, S. Guadarrama,”Long-term recurrent convolutional networks for visual recognition and description. IEEE Transactions on PAMI 39(4): 677–691, 2016.

[11] Graves A and Jaitly. ”Towards end-to-end speech recognition with re- current neural networks”, In: Proceedings of the international conference on machine learning (ICML), volume 14, pp. 1764–1772, 2014.

[12] R. Clark, S. Wang, H. Wen, A. Markham, N. Trigoni. ”VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem”, AAAI Conference on Artificial Intelligence, 2017.

[13] Mur-Artal, Raul, and Juan D. Tardo´s. ”Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras.” IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[14] A. Dosovitskiy, P. Fischer, et al. “Flownet: Learning optical flow with convolutional networks,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2758–2766.

[15] Liu, Ruixu, et al. ”Real-time 3D scene reconstruction and localization with surface optimization.” NAECON 2018-IEEE National Aerospace and Electronics Conference. IEEE, 2018.

[16] Dai, Angela, et al. ”Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration.” ACM Transactions on Graphics (TOG) 36.4 (2017): 76a.

[17] Whelan, Thomas, et al. ”ElasticFusion: Real-time dense SLAM and light source estimation.” The International Journal of Robotics Research 35.14 (2016): 1697-1716.

[18] Whelan, Thomas, et al. ”Real-time large-scale dense RGB-D SLAMwith volumetric fusion.” The International Journal of Robotics Research 34.4-5 (2015): 598-626.

[19] Kerl, Christian, Jurgen Sturm, and Daniel Cremers. ”Dense visual SLAM for RGB-D cameras.” Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on. IEEE, 2013.

[20] Hochreiter, Sepp, and Ju¨rgen Schmidhuber. ”Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.