RO-Net: Recurrent Neural Networks for Range-only SLAM using range-only measurements

Hyungtae Lim[1] , Junseok Lee[1], Changgyu Park[1], Ye Eun Kim[1],

***Abstract*— Range-only(RO) SLAM is a method for localizing a mobile robot and beacons by mainly utilizing distance measurements. Because range-only measurements have only magnitude so it has rank-deficiency. And distance is only measured by the time of flight(TOF), data is noisy.**

In this paper, we proposed a novel approach to range-only SLAM using multimodal bidirectional stacked LSTM models. Unlike the traditional probability-based range-only SLAM method, we present a novel approach using a recurrent neural network architecture that directly learns the end-to-end mapping between distance data and robot position.

We gathered our own dataset and tested in 2 cases exploiting eagle eye motion capturer camera. The multimodal bidirectional stacked LSTM structure exhibits the precise estimates of robot positions, but one case, it is less accurate than traditional SLAM algorithm.

I. INTRODUCTION

SLAM is widely used in autonomous vehicles, drones, intelligence field robots, and mobile phone applications. Thus, according to the smart city development plan, several technologies are required, and the importance and the necessity of SLAM are increasing together. Various kinds of sensors are utilized to SLAM, such as GPS, LiDAR, ultrasonic-based sensor, camera and distance sensor.Trilateration is a conventional algorithm for locating a vehicle in the metropolitan area by range measurements between the vehicle and fixed beacon sensors [1]. Due to the convenience of trilateration that estimates the position of a receiver of range sensors if one only knows range measurement, trilateration algorithm has been widely incorporated into robotics fields, especially utilized in the indoor environment to estimate the position of an object by distance measurements obtained from range sensors such as UWB, ultrasonic, laser-based beacon sensors [2]–[4]. Specifically, range-only Simultaneous Localization and Mapping(RO-SLAM) methods are utilized popularly, which not only estimate the position of the receiver of range sensors, but also localize the position of range sensors regarded as features on a map, and studies have been conducted continuously in terms of probability-based approach [5]–[8].

In the meantime, as deep learning age has come [9], various kinds of deep neural architectures have been proposed for many tasks related to robotics field, such as detection [10]–[12], navigation [13], [14], pose estimation [15], and so on. Especially, recurrent neural networks (RNNs), originated from Natural Language Process(NLP) area [16], have been shown to achieve better performance in case of dealing with time variant information, thereby RNNs are widely utilized such as not only speech recognition, but also pose estimation and localization [15], [17]–[20].

In this paper, we propose a deep learning-based SLAM method by multimodal stacked bidirectional Long Short-Term Memory(multimodal stacked Bi-LSTM) for more accurate localization of the robot. Using deep learning, our structure directly learns the end-to-end mapping between range measurements and robot position. This operation non-linearly maps the relationship not only considering the long-range dependence of sequential distance data by the LSTM, but also using the correlation of the backward information and the forward information of the sequence of each time step by virtue of its bidirectional architecture. Existing RO SLAM needs calibration before filtering, and then, range measurement undergoes outlier rejection, prediction and correction processes are needed. Furthermorme, it uses low dimensional data to perform localization, there is a disadvantage that estimation is difficult even if the value deviates slightly from the model. Therefore, we solve this complex algorithm with end-to-end based deep learning. This system overview is shown in the figure below.

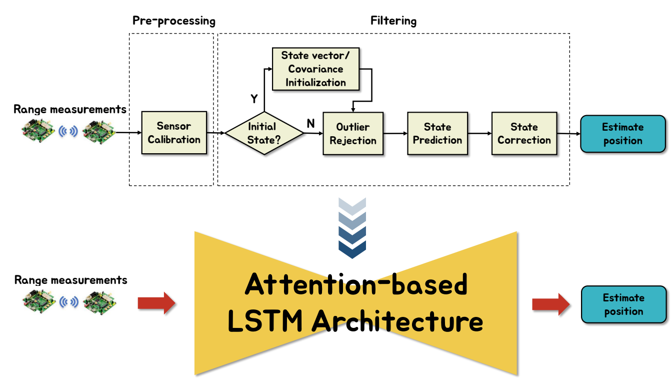


Fig. 1. System overview. A robot localizes its own pose through distance data and the derivative of distance data. We replace whole probabilistic approach with the deep learning-based ene-to-end mapping.

II. RELATED WORKS

In this section, we briefly survey previous researches closely focused on RO-SLAM, Long Short-Term Memory(LSTM) model and applications of LSTMs to solve domain problems.

[1]Hyungtae Lim, [1]Junseok Lee, [1]Changgyu Park, and [1]Ye Eun Kim are with the Urban Robotics Laboratory, Korea Advanced Institute of Science and Technology (KAIST) Daejeon, 34141, South Korea. {shapelim, ljs630, cpark, yeeunk}@kaist.ac.kr

1) RO-SLAM: SLAM is a technique for building the map information while localizing the position of the robot while moving. Localization of the SLAM predicts the current position of the robot using the landmark measured by the sensor, and mapping locates the terrain object based on the pose of the robot. Research on this technology has been actively carried out, and researches and techniques have been summarized. In 2006, the ad hoc sensor network consisting of range detection beacon was applied to SLAM technology for various ranges. This technology integrates node-to-node measurements to reduce drift and expedite node-map convergence [21] In 2008, the technique to consistently combine the observation information considering the uncertainty was studied through comparing the experimental data with the actual robot and simulation using Ultra Wide-Band (UWB) devices and Rao-Balckwellized Particle Filter (RBPF) [5]. In 2012, a simple and efficient algorithm for position recognition with high accuracy and low computational complexity was researched with ultrasonic sensors [22]. In recent years, 3-dimensional-based SLAM has also been under active research and development. In 2013, a localization mapping approach of a wireless sensor network (WSN) node was studied through a centralized EKF-SLAM-based optimization research [7]. In addition, in 2014, a method of minimizing noise and localizing Unmanned Aerial Vehicle (UAV) by using range-only measurement while simultaneously mapping the position of the wireless range sensors were proposed [23]. SLAM based on range measurement has been continuously researched and developed then applied to various fields. In this paper, we propose a novel technology that applying deep-learning to range-only SLAM that derives accurate range and robot position measurement through in-depth learning.

2) LSTM: LSTM is a type of Recurrent Neural Networks(RNNs) that has loops so that infer output based on not only the input data, but also the internal state formed by previous information. In other words, while the RNN deals with sequential data, the network has remembered the previous state generated by past inputs and might be able to output the present time step via internal state and input, which is very similar to filtering algorithms.

However, RNNs often have a *vanishing gradient problem*,i.e., RNNs fail to propagate the previous matter into present tasks as time step gap grows by. In other words, RNNs are not able to learn to store appropriate internal states and operate on long-term trends. That is the reason why the Long Short-Term Memory (LSTM) architecture was introduced to solve this long-term dependency problem and make the networks possible to learn longer-term contextual understandings [24]. By virtue of the LSTM architecture that has memory gates and units that enable learning of long-term dependencies [25], LSTM are widely used in most of the deep learning research areas and numerous variations of LSTM architecutres have been studied.

3) Localization with Deep Learning: There have been many approaches combining Simultaneous Localization and Mapping (SLAM) with deep learning, aiming to overcome the limitations on SLAM only technique such as difficulty on tuning the proper parameters in different environments and recovering an exact scale. Actually, those researches are showing the superior performance to the traditional SLAM approaches.

One of the popular SLAM techniques with deep learning is CNN-SLAM [26] which takes Convolutional Neural Networks (CNNs) to precisely predict the depth from a single image without any scene-based assumptions or geometric constraints, allowing them to recover the absolute scale of reconstruction. Another approach using deep learning for localization is Deep VO [27] In this method, Recurrent Convolutional Neural Networks (RCNNs) is utilized. Specifically, feature representation is learned by Convolutional Neural Networks and Sequential information and motion dynamics are obtained by deep Recurrent Neural Networks without using any module in the classic VO pipeline.

4) Applications of LSTMs: There are many variations of LSTM architecture. As studies of deep learning are getting popular, various modified architectures of LSTM have been proposed for many tasks in a wide area of science and engineering. Because LSTM is powerful when dealing with sequential data and infering output by using previous inputs, LSTM is utilized to estimate pose by being attached to the end part of deep learning architecture [18]–[20] as a stacked form of LSTM. In addition, LSTM takes many various data as input; LSTM is exploited for sequential modeling using LiDAR scan data [17], images [15], [18], IMU [28], a fusion of IMU and images [27]. Since existing RO-SLAM performs localization using low-dimensional data, it is difficult to estimate even if the value deviates slightly from the model. In addition, LSTM has the advantage of being able to solve long-term dependence problem of traditional RNN, and it is possible to model it by non-linear mapping through analyzing the current situation without modeling data characteristics separately. Therefore, we propose RO SLAM technology using deep learning based SLAM which applies the advantages of LSTM and deep learning to solve the disadvantages of RO SLAM.

5) Attention: Attention is powerful module nowadays and mostly improves performance of neural network. Originally neural networks treats information equally. But, using attention layer, neural networks can be ATTENDED what it should be examined closely. At the first time, attention is utilized at natural language processing area for improving translation performance [29]. But nowadays, attention layer is employed in many areas to improve the performance of the networks. For example, Jaderbeg *et al.* [30] introduced the attention layer to let the neural networks attend to spatial information. In addition, attention is even utilized to pose estimation and optimization [31], detection [32], and video captioning [33]

III. Our approaches

In this chapter, we introduce our proposed neural network model which is used for estimating the robot's pose and Landmarks' position when only the range sensor data is

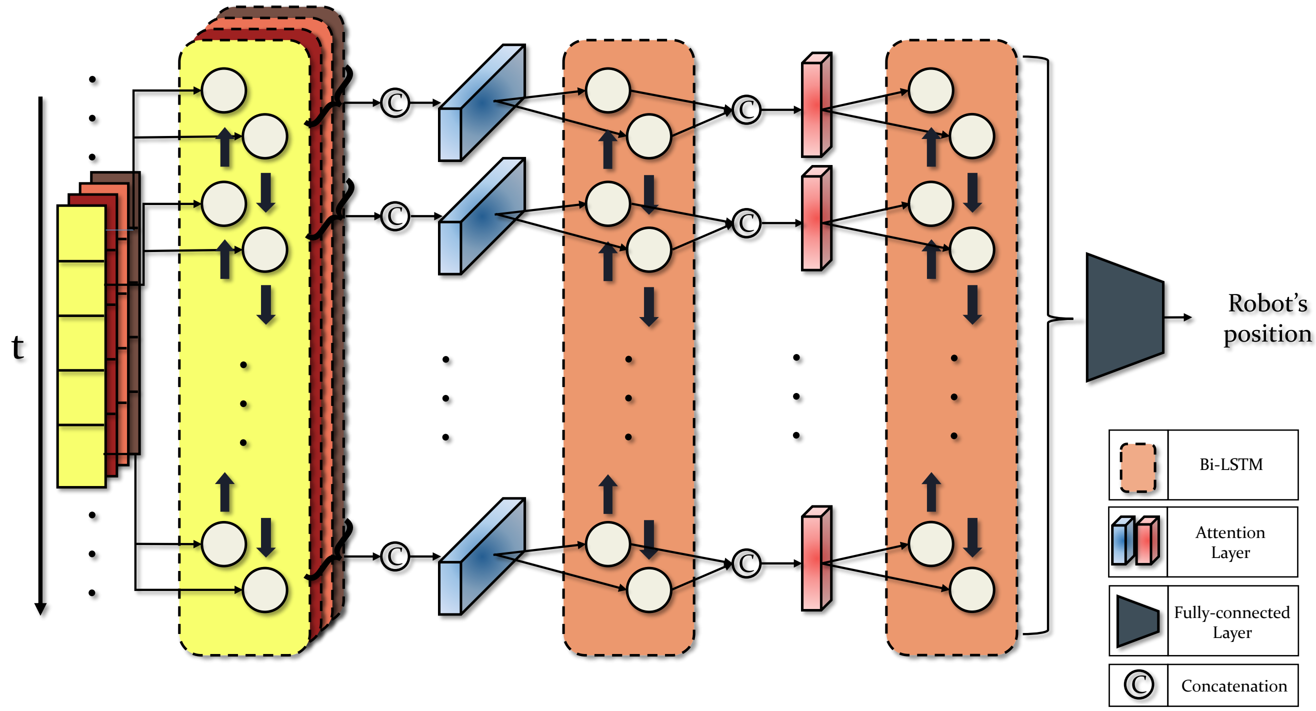


Fig. 2. Overall architecture of multimodal stacked Bi-LSTM.

given from each distance sensor. Firstly, the overall network architecture is provided. Then, the details of each part are explained.

A. Network Architectures

As it is illustrated in Fig. 2, our proposed stacked Bi-LSTM can be divided into 3 parts: (1) Input part, which accepts and preprocesses the sequences of the sensors with multiple bidirectional LSTM layers. (2) Hidden layer part consisting of attention modules and bidirectional LSTM layer (3) Output part where a fully-connected output layer gives Robot's pose and positions of landmarks as the network's final results.

B. Multimodal LSTM

To effectively accept the inputs collected from the multiple sensors, instead of using a single layer as an input layer, we use several LSTM layers, thinking that each single sensor represents a different modality. Each layer corresponds to the input of each distance sensor. In other words, if N sensors are used for measuring the distance of the robot, the number of input layers also would be N. And, the MTh layer accepts an input from the MTh sensor. So, the network is able to represent total N modalities. By doing so, we can further expect that the input layers can act as the sensor calibration process in traditional RO-SLAM, allowing the sensors to be tuned respectively with the input layer's parameters.

C. Bidirectional LSTM

As traditional RO-SLAM [5], [6] takes an odometry which is an accumulated data from the beginning to the present point, our network takes sensor data for the time period l. So, if the current time stamp is t, the input layers take the sensor data obtained from timestamp t-l+1 to t. For dealing with such sequential information, LSTM network which is one of the most appropriate network for sequential data is applied to our network and each LSTM layer is designed to have l cells. Furthermore, to take an advantage from the bidirectional time flow, normal time order and reverse time order, we place the bidirectional LSTM layers in the three different locations. Each bidirectional LSTM layer consists of 2 independent LSTM layers corresponding to normal and reverse time order repectively. Individual LSTM layers play a different role. The LSTM layers of input part take and preprocess the sequence of sensor data. LSTM layer placed between input and output layer takes a spatial information from a previous spatial attention layer and send it to another temporal Attention layer. Lastly, the LSTM layer at the end outputs the positions of robot and landmars.

D. Attention layer

To precisely estimate the Robot's pose and landmarks' position, it is important for the network to distinguish which is more meaningful information and which is less for preventing to focus on less significant information. So, we add the two different types of attention modules [29] which extract something more important and related to the task information by making the network to focus on different part of input sequence. The first attention modules placed between the input LSTM layers and the second LSTM layer are called "Spatial attention modules". The spatial attention modules are represented as blue blocks in Fig. 2. These attention modules can judge which sensor has more spatial information. The second attention modules corresponding to the red blocks in Fig. 2 are the "Temporal attention modules". These temporal attention modules can determine which time stamp has more useful information about time, allowing the network to attend that time stamp more.

E. Stacked Architecture

In deep learning, the number of layers stacked is getting large, intending to increase the non-linearity and correspondingly to improve the performance. Likewise, the multiple LSTM layers can be stacked as well [34], enabling more complex representation and higher performance. In stacked Bi-LSTM, total 3 LSTM layers are stacked in the series.

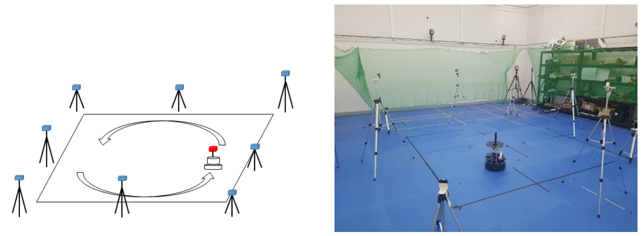
IV. Experiment

A. Experimental environment

Our experimental system consists of a UWB(ultra wide-band) sensor tag and eight anchors that have a UWB transceiver, the motion capture system with 12 cameras, a mobile robot and a small form-factor computer. UWB sensor anchors attached to landmarks become the end points of the range measurements. The anchors transmit the UWB signal. A UWB sensor tag attached to a robot becomes the opposite side end point of the measurements. The tag receives the signal and measures the range between two devices. Each UWB transceiver, DW1000 UWB-chip made by Decawave, supports 6 RF bands from 3.5 GHz to 6.5 GHz. It measures in centimeter-level accuracy . The motion capture system is Eagle Digital Realtime system of motion analysis corporation that operates with the principle of stereo pattern recognition that is a kind of photogrammetry based on the epipolar geometry and the triangulation methodology. The system has < 1mm accuracy. A mobile robot used in experiment is iClebo Kobuki from Yujinrobot that has 70 cm/s maximum velocity . The small form-factor computer is a gigabyte Ultra compact PC. Deep learning framework used for our network is pytorch 0.4.0 on python 3.6. The network inferences on the same setting. Fig. 3 shows the description of experimental environment. The UWB tag is attached to mobile robot that has a small compact computer. The UWB anchors are attached to stands that have two different heights. The anchors are positioned randomly in the square space. Inside of the space, a mobile robot goes on various random paths by experimenters. During the robot is going on, the data is saved in the computer. The distance data used for input data is measured by the UWB sensors. The global position data used for ground truth is measured by the motion capture system. After time synchronizing these two kinds of data, these are paired in a dataset. Each path has one dataset. All the paths are different.

B. Training/Test Dataset

In the computer there are two different thread. One receives UWB sensor data. The other one receives global position data. To synchronize time, we make an independent thread that concatenates and saves these data at the same time. The data is saved at 20Hz frequency. The description is shown in Fig. 4. After the experiment, we separate the entire dataset to two types, some are the training datasets and others are test datasets.



(a) Description. (b) Actual.

Fig. 3. Experimental system overview. In (a), anchors are attached to landmarks. A tag is attached to a mobile robot. The robot goes on random paths in square space. (b) shows actual experiment situation.

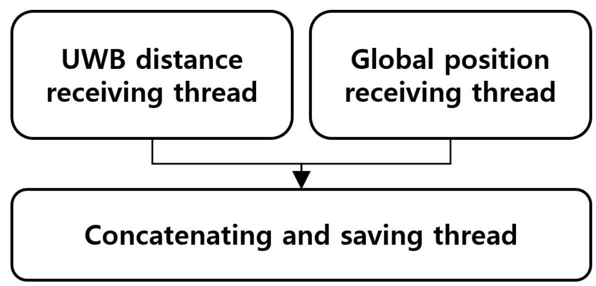


Fig. 4. Data syncronizing method. UWB distance data and the global position data are received to different thread. To syncronize time, another independent thread concatenates and saves these two kinds of data.

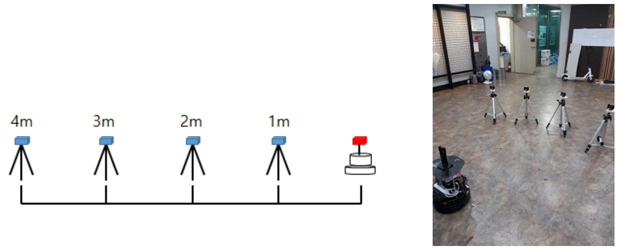
C. Sensor calibration

In addition, to use the distance data for traditional RO-slam we should calibrate the distance from each anchors. To calibrate it, we follow the method in the baseline paper. As you can see in Fig. 5(a), we measure the data from a tag to each anchors at the points where the actual distance was measured by 1m, 2m, 3m, 4m. Fig. 5(b) shows that four different anchors are measured at the same time. By using the linear regression, we compute the ratio between the measurement and the actual distance. And the ratios of each anchor are used to calibrate it.

D. Training Loss

The network is programmed by Tensorflow, which is deep learning library of python trained by using a GTX 1080 Ti and GTX Titan. The Adam optimizr is exploited to train the network during 1000 epochs with 0.0002 learning rate, 0.7 decay rate, and 5 decay step. Besides, Dropout is introduced to prevent the models from overfitting.

Let Θ be the parameters of our RNN model, then our final goal is to find optimal parameters Θ^* for localization by



(a) (b)

Fig. 5. Sensor calibration overview. (a) shows The formation between a tag and an anchor. (b) shows that four anchors are measured at the same time.

minimizing Mean Square Error (MSE) of Euclidean distance between ground truth position Y_k and estimated position $\hat{Y}_k$.

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{k=1}^{N} \| Y_k - \hat{Y}_k \|^2 \tag{1}$$

V. Results

To verify our proposal that RNNs can estimate the robot's position through varying range data, we trained our RNN-based multimodal architecture. Plus, to compare to previous traditional SLAM algorithm, we also estimate robot's position by particle filter(PF) based algorithm.

As illustrated in Experiment session, train data are our own data gathered by UWB sensors and motion capture camera, so neural networks take range-only measurements as input and output robot's position. Ground truth data is robot's position measured by eagle eye motion capturer, whose error is in mm units. The results of trajectory prediction are shown in Fig. 6 and Root-Mean-Squared Error (RMSE) are shown in Table I. Note that out experiment is conducted on mobile robot, so we can pre-estimates that z position of robot is almost similar while robot is running.

We set two test trajectory cases: an square path and zigzag path. an The results shows that it has better performance than established probabilistic-based approach. In both cases, performance of our networks is better that of particle filter. RMSE of our networks in test1 is 3.928cm and 4.119cm in test2.

We also verified effectiveness of attention layer. It was confirmed that the performance of the networks with the attention layer is improved compared to the networks without the attention layer.

The results of RMSE[cm]		
Model	Test1	Test2
Particle Filter-based w/o pre-estimates of z	11.253	9.195
Particle Filter-based	5.525	5.258
Multimodal(Ours)	4.225	4.311
Multimodal(Ours) + attention	**3.928**	**4.119**

TABLE I: Root mean squared error of each case

VI. Conclusion

In this paper, we proposed a novel approach to range-only SLAM using multimodal-based RNN models and tested our architectures in two test data.

Using deep learning, our structure directly learns the end-to-end mapping between distance data and robot position. The multimodal bidirectional stacked LSTM structure exhibits the precise estimates of robot positions. We set two test trajectory cases: an square path and zigzag path. an The results shows that it has better performance than established probabilistic-based approach. In both cases, performance of our networks is better that of particle filter. RMSE of our networks in test1 is 3.928cm and 4.119cm in test2. Therefore, we could check the possibility that our multimodal LSTM-based structure can substitute traditional algorithms

As a future work, because we conducted on just localization, this approach may not be operated when locations of sensors are changed. Therefore, the proposed method needs to be revised for precise estimates even though locations of anchors are changed.

References

[1] H. Staras and S. Honickman, "The accuracy of vehicle location by trilateration in a dense urban environment," *IEEE Transactions on Vehicular Technology*, vol. 21, no. 1, pp. 38–43, 1972.

[2] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Transactions on robotics*, vol. 21, no. 1, pp. 93–101, 2005.

[3] H. Cho and S. W. Kim, "Mobile robot localization using biased chirp-spread-spectrum ranging," *IEEE transactions on industrial electronics*, vol. 57, no. 8, pp. 2826–2835, 2010.

[4] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran, "Accurate mobile robot localization in indoor environments using bluetooth," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4391–4396.

[5] J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal, "A pure probabilistic approach to range-only slam." in *ICRA*. Citeseer, 2008, pp. 1436–1441.

[6] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. González, "Efficient probabilistic range-only slam," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 1017–1022.

[7] F. R. Fabresse, F. Caballero, I. Maza, and A. Ollero, "Undelayed 3d ro-slam based on gaussian-mixture and reduced spherical parametrization," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. Citeseer, 2013, pp. 1555–1561.

[8] N. S. Shetty, "Particle filter approach to overcome multipath propagation error in slam indoor applications," Ph.D. dissertation, The University of North Carolina at Charlotte, 2018.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

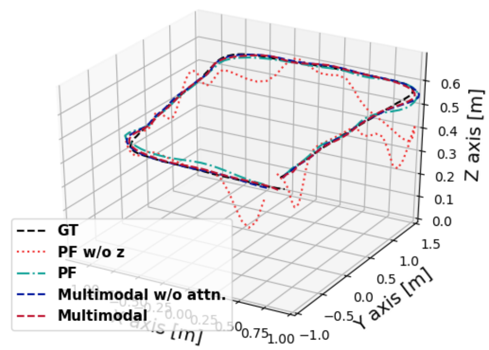
[10] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[11] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

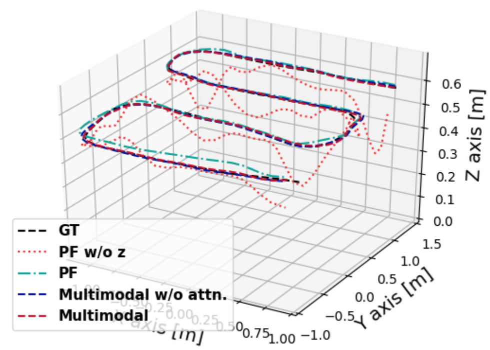
[12] H. H. Smith, "Object detection and distance estimation using deep learning algorithms for autonomous robotic navigation," 2018.

[13] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3357–3364.

[14] M. Hamandi, M. D'Arcy, and P. Fazli, "Deepmotion: Learning to navigate like humans," *arXiv preprint arXiv:1803.03719*, 2018.

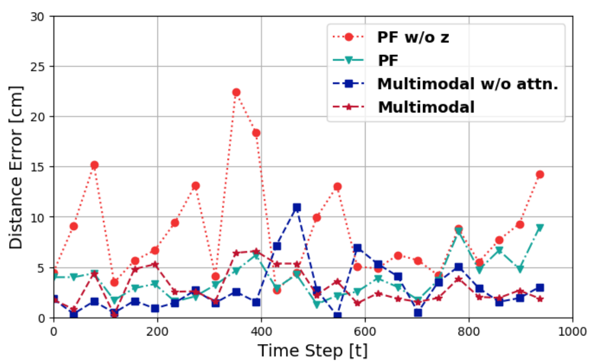


(a)

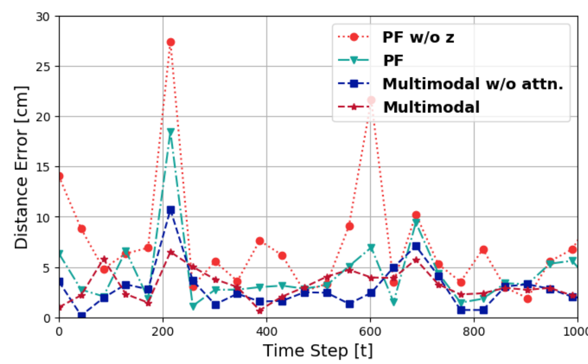


(b)

Fig. 6. Trajectories estimated by particle filter-based algorithm and our neural networks' architecture. (a)A trajectory of test1 data (b)A trajectory of test2 data



(a)



(b)

Fig. 7. The distance error graphes with time step. (a) The Distance error of test 1 data and (b)distance error of test 2 data

[15] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Int. Conf. Comput. Vis.(ICCV)*, 2017, pp. 627–637.

[16] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[17] S. Gladh, M. Danelljan, F. S. Khan, and M. Felsberg, "Deep motion features for visual tracking," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 1243–1248.

[18] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2043–2050.

[19] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[20] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti, "Deep endovo: A recurrent convolutional neural network (rcnn) based visual odometry approach for endoscopic capsule robots," *Neurocomputing*, vol. 275, pp. 1861–1870, 2018.

[21] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only slam for robots operating cooperatively with sensor networks," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2078–2084.

[22] P. Yang, "Efficient particle filter algorithm for ultrasonic sensor-based 2d range-only simultaneous localisation and mapping application," *IET Wireless Sensor Systems*, vol. 2, no. 4, pp. 394–401, 2012.

[23] F. R. Fabresse, F. Caballero, I. Maza, and A. Ollero, "Robust range-only slam for aerial vehicles," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE, 2014, pp. 750–755.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] W. Zaremba and I. Sutskever, "Learning to execute," *arXiv preprint arXiv:1410.4615*, 2014.

[26] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[27] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem." in *AAAI*, 2017, pp. 3995–4001.

[28] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[29] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[30] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[31] E. Parisotto, D. S. Chaplot, J. Zhang, and R. Salakhutdinov, "Global pose estimation with an attention-based recurrent network," *arXiv preprint arXiv:1802.06857*, 2018.

[32] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance

video object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7210–7218.
[33] J. Xu, T. Yao, Y. Zhang, and T. Mei, "Learning multimodal attention lstm networks for video captioning," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 537–545.
[34] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, "Transition-based dependency parsing with stack long short-term memory," *arXiv preprint arXiv:1505.08075*, 2015.

VII. Contribution

Hyungtae Lim: Implementing networks, drawing up a plan for experiment, planning experiment schedules, and extracting results.

Junseok Lee: Survey on related works, producing the parts needed experiments by 3D printer, operating Kobuki robot for gathering train data

Changgyu Park: Data preprocessing, writing code for result plotting, implementing the system that gathers dataset

Yeeun Kim: Survey on related works and established RO SLAM codes, study about LSTM.