RONet: Real-time Range-only Indoor Localization via Stacked Bidirectional LSTM with Residual Attention

Hyungtae Lim[1] , Changgyu Park[2], Hyun Myung[3], *Senior Member, IEEE*

***Abstract*— Range-only SLAM is a method for localizing a mobile robot and beacons by mainly utilizing distance measurements. 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. !!! 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. !!!!

As verified experimentally, this new proposal represents a significant improvement in accuracy, computation time, and robustness against outliers.

I. INTRODUCTION

In recent years, as demand for localization in indoor environments where the signals of the Global Positioning Systems(GPS) could become imprecise gradually increases, many researchers have conducted various methods for locating obejcts, e.g., magnetic fields, acoustic signals, and laser-based data. Among them, Time of Flight(TOF)-based range beacon sensors are widely utilzed by virture of characteristics of beacon sensors: low-cost, small-size, acceptively accurate performance, and convenience of being installed. As a result, these range measurement-based approaches have been suggested as a solution for localization not only on the indoor environment [1], [2], but also underwater environments [3], [4]

Specifically, these range-only approaches has addressed the problem of localization with sets of range-only measurements between object node that we want to localize, called tag node, and landmarks, called anchor node. However, range measurements that only represent distances between each landmark and mobile robot respectively, one the other words, a set of one-dimensional data have two problems: one is that it tend to be non-linear becasue TOF-based measurement is very vulnerable to noise and has huge uncertainties caused by multipath fading channel(MPF) problem [5] in real world, and the other one is that these range-only observations have *rank dificiency* problem [6]. To be specific, the single value to represent distance between each landmark and mobile robot respectively is deficient to describe exact position or orientation of the landmark so cause multimodal distribution [7].

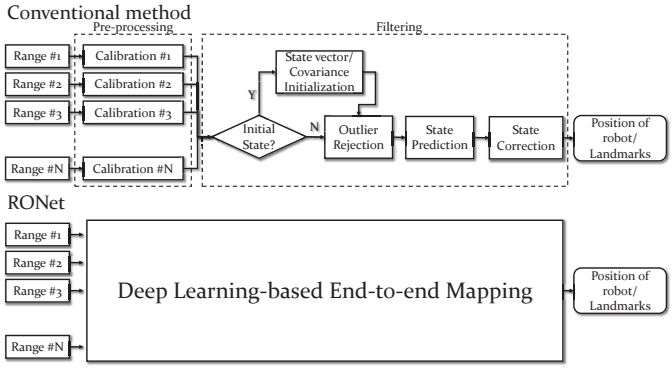


Fig. 1. Comparison between a standard range-only localization or SLAM framework and our learning-based approach.

To alleviate these issues, many studies have been conducted based on probabilistic Bayesian inference framework and Monte-Carlo Bayesian filter, but in recent years, there have been attempts to solve these problems based on neural-network-based approaches [8]–[11]. With non-linear end-to-end mapping, the authors show feasibility. But most cases, they just utilized Multilayer Perceptron(MLP), which is beginning architecture of deep learning fields [8]–[10]. In [11], they exploit stacked bidireactional Long Short-Term Memory(Bi-LSTM) to cover noise of range observation by utilizing the chracteristics of it that takes temporal sequential value as input, yet they tested on the simulated environment. And all of them are not checked whether their learning-based approaches are real-time or not.

In this paper, we propose a robust stacked BI-LSTM with residual attention, called RONet. To the best of our knowledge, it is a first approach to apply LSTM-based architecture to localize a mobile robot on the real-world in real-time using range-only measurement. Unlike conventional probabilistic-based algorithms, it does not need any preprocessing module,

*This material is based upon work supported by the Ministry of Trade, Industry & Energy(MOTIE, Korea) under Industrial Technology Innovation Program. No.10067202, 'Development of Disaster Response Robot System for Lifesaving and Supporting Fire Fighters at Complex Disaster Environment'.

[1]Hyungtae Lim, [2]Jungmo Koo, [3]Jieum Hyun, and [4]Hyun Myung are with the Urban Robotics Laboratory, Korea Advanced Institute of Science and Technology (KAIST) Daejeon, 34141, South Korea. {shapelim, jungmokoo, jimi, hmyung}@kaist.ac.kr

such as an calibration and outlier rejection.

Our contribution is threefold as follows:

- We develop 3-stacked Bi-LSTM layers and attach residual attention layer for both improving performance and let the neural network be trained well so that our RONet shows the best performance when comparing the previous approaches.
- We also analyzed how the sequential length of the network affects performance and check robustness of our RONet with minimal number of anchors.
- We operate RONet on the Nvidia Jetson AGX Xavier and check the inference Hz is Real-time, about 32Hz.

The rest of the paper is organized as follows: Section II overviews the related works. Section III describes our neural network in detail and defines the problem to be considered, and Section IV describes the experimental results. Finally, Section V summarizes our contributions and points to future work.

II. Related Works

A. Conventional Range-Only Localization

To localize a mobile robot using range measurements, there are two conventional approaches: Kalman Filter(KF)-based method and Particle Filter(PF)-based method.

EKF [4]

Besides, these measurements could have huge uncertainties caused by multipath fading channel(MPF) problem [5] in real world. To alleviate these issues, many studies based on Kalman Filter(KF) algorithm or probabilistic approaches are conducted how to estimate state of the mobile robot and landmarks more precisely with covering the uncertainties.

based on Ultra-Wide-Band(UWB), ultrasonic, laser-based beacon sensors. By virtue of this low-cost, small-size, acceptively accurate performance, and convenience of being installed, RO-SLAM

h due to the convenience of trilateration that estimates the position of a receiver of range sensors if one only knows range measurement. For that reasons, 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 [12]–[15]. communicate with other wiresless sensors efficiently and conveniently.

B. LSTM-based Sequential Modeling

As deep learning age has come [16], various kinds of deep neural architectures have been proposed for localization task related to robotics field [17]–[19]. Especially, recurrent neural networks (RNNs), originated from Natural Language Process(NLP) area [20], have been shown to achieve better performance in case of dealing with time variant information, thereby RNNs are widely utilized for sequential modeling such as not only speech recognition, but also pose estimation and localization.

And as Long Short-Term Memory (LSTM) architecture solves the *long-term dependency*, which is the inherent issue to RNNs that become unable to learn the relationship of sequential informations as the time-sequential gap grows [21], LSTM are actively introduced to learn longer-term contextual understandings. For example,in [22], the authors estimate the odometry by combination of RGB images and IMU sensor data and thy utilize the CNN unit for extraction of spatial information by images and LSTM unit for sequential modeling. in [23], [24], the architectures combined with CNN and LSTM are proposed for localization and odometry estimation tasks. LSTM unit extract sequential features from temporal outputs of CNN units. In [25], Walch *et al.* propose the architecture that include CNN and LSTM units to regress camera pose for indoor and outdoor scenes. LSTM unit extract the sequential feature in an output of CNN unit. In addition, in [26], the authors utilize multi-layered LSTM network for sequential action recognition by RGB-D images.

C. Deep Learning for Range Only Localization

Specifically, LSTM are also utilized to model low-dimentional sensor data by itself. In [27], they exploit LSTM for indoor localization with magnetic and light sensors. And in [28], they estimate 2D odometry via stacked bidirectional LSTM that takes only IMU sensor data as input.

Our target, localization using range-only measurement, many authors employee neural networks-based approaches in Wireless Sensor Networks Fields(WSNs) [8]–[10], yet most networks are based on The Multilayer Perceptron(MLP). Their approach only map a set of range measurment on time t to position so their approaches might have potential to unexpected sensor input. Other paper [11], they utilize stacked bidirectional LSTM for localization a mobile robot takes sequential rnage measurements from anchors and the tag, but they only simulated on a ideal case, that MPF or unexpected noises not occur.

Therefore in this paper, we conduct experiments on a real-world to verify the feasibility to cover all noises of sensors with seqeuntial range data and compare these approaches.

!!!!!!!!!!!!!!!!!!!!

An main advantage of neural networks-based approaches is that the neural networks enables themselves to recognize the position of the MNs quite accurately through the training even if there is no mathematical description. The neural networks also cover all noises that occurs when ANs and MNs interchange their signals. The authors show that their nerual networks-based approach have better performance than traditional approaches or machine learning-based approaches, yet most networks are based on The Multilayer Perceptron(MLP). In addition, the some authors just show the feasibility of their approaches, i.e., they trained and tested on the simulations situation. Plus, some authors let the neural networks infer the estimates on the two-dimensional space, but gap of the complexity of estimating position between on two-dimensional space and on three-dimensional using *rank-deficient* measurements cannot be negligible.

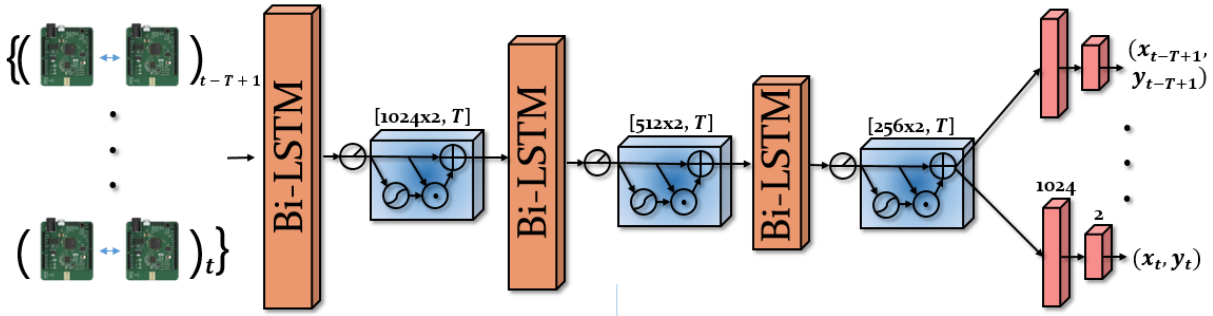


Fig. 2. Our networks

D. Residual Attention Layer

A Attention layer 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]

To precisely estimate the position of the tag node, it is important for the network to distinguish which is more meaningful context on time step T to help contextual understanding of our networks. So, we add the attention layer between the LSTM and the attention layer take on a role of a feature selector [34]. The equation of the attention machanism is as follows:

III. RONET

In this chapter, we explain how our proposed residual attention-based stacked Bi-LSTM is implemented, as illustrated in Fig. 2. In detail, we introduce the neural networks concepts that we choose for localizaing the tag node and then compare to those of other previous works. Finally, we describe how to set the loss function of our neural network

A. Long Short-Term Memory

Unlike RNN that consist only of hidden state, in LSTM, cell state is added on the network. The cell state consists of the 3 gates to preserve the previous information and control the cell state: forget gate, input gate, and output gate and equations of those are as follows:

$$f_t = \sigma_s\left(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f\right) \quad (1)$$

$$i_t = \sigma_s\left(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i\right) \quad (2)$$

$$\tilde{c}_t = \tanh\left(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c\right) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (4)$$

$$o_t = \sigma_s\left(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o\right) \quad (5)$$

$$h_t = o_t \odot \tanh\left(c_t\right) \quad (6)$$

where σ_s is a kind of activation function, called *sigmoid*, f_t, i_t, and o_t respectively indicates the forget gate, input gate, and output gates, and c_t denotes cell states. And $\odot$ denotes element-wise multiplication, called *Hadamard product*. Entire gates are activated by sigmoid function and cell states are activated by $\tanh$ function.

The Forget gate layer, f_t, decides how much information to forget based on previous hidden state, h_{t-1}, and present input, x_t. Next, the input gate, i_t, decides how much information to embrace when updating the cell state. After that, c_t is updated by the cell state layer based on f_t, i_t, and candidate cell state, $\tilde{c}_t$ (4). In addition, output gate layer, o_t, sereves as a filter, which means o_t determine what values are going to output (5) in such a way as that h_t, is updated based on o_t updated cell state, c_t (6).

B. Stacked Bidirectional LSTM

Like the fact that the deeper the architecture of neural networks, the better their performance [35], [36], many authors have analyzed variations of LSTM architecture and find out that stacking multiple layers of the LSTM improve the performance for many tasks [37]–[39]. Besides, Bidirectional RNNs are introduced [40] to extract well-described context. It has one forward LSTM, $\overrightarrow{LSTM}$, and one backward LSTM, $\overleftarrow{LSTM}$, running in reverse time so that the network exploits not only previous forward context to up update h_t and c_t but also future backward context as well, as FIGURE. 3 shown.

For these reasons, we adopt stacked bidirectional LSTM architecture to model the system. By virtue of these increased non-lineary caused by a number of stacked layers, the network could model more complex localization taking UWB-rangings as input including unexpected noise and MPF problems. And, we judge that Bi-LSTM would be more

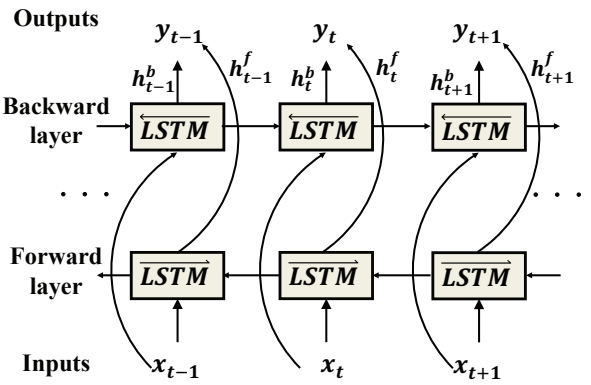


Fig. 3. Architecture of the Bidirectional LSTM(Bi-LSTM). bidirectional LSTM consist of 2 LSTMs: one forward LSTM layer

helpful to produce more appropriate context considering both past and future at the same time

Therefore, we construct our networks by stacking three LSTM to increase the non-linearity. Note that stacking over than three LSTM doesn't show the improvement of performance. We deem that this problem could comes from the sigmoid function and $tanh$ function that compose the part of LSTM. These activation functions cause the *vanishing gradient problem* [41], which the networks fail to training due to the fact that the gradient is getting closer to zero during the backpropagation. Consequently, we put the ReLu function between LSTMs to avoid the vanishing gradient problem [42], instead of stacking more LSTM to increas non-linearity. And experiments shows that reducing the hidden size of next LSTM layer when the features are fed into the LSTM layer slightly increases preformance, but reducing dramatically rather cause underfitting. In conclusion we decide to set the size of the layers as 1024-512-128. Note that we adopt Bi-LSTM, acutual feature size is 2048-1024-256. And end part of the LSTMs, fully connected layers are attached to predict the mobile robot's position based on the sequential features processed by the LSTMs.

C. Residual Attention layer

A Attention layer 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. In other words, attention layer take on a role of a feature selector [34]. To precisely estimate the position of the tag node, it is important for the network to distinguish which is more meaningful context on time step *T* to help contextual understanding of our networks. The equation of original attention machanism is as follows:

$$H(x) = M(x) \odot x \tag{7}$$

where x denotes the output of previous neural network layer, $H(x)$ denotes the output of the attention layer to be passed to the next neural network layer and $M(x)$ denotes the attention mask. By element-wise multiplying x by $M(x)$, attention layer makes the network weight more crutial information.

Despite of the improvement of the performance, the attention layer has potential risks that it may dilutes the features because attention mask ranges over 0 to 1. So we adopt residual attention layer to alleviate this problem as follows [34]:

$$H(x) = (1 + M(x)) \odot x \tag{8}$$

As blue cuboid shape in the FIGURE 2 shown, this idea is originated from the Residual Net(ResNet) [36] that has skip connection in such a way as to mitigate aforementioned dilution problem and help the network to be trained well. Likewise the ResNet, residual attention also has other branch to calculate how much to attend and the branch is joined with original feature vector x. Each hidden state has each residual attention layer so that these attention modules can determine which time stamp has more fruitful meaning and deliver the output to next bidirectional LSTM.

D. Training loss

In this section, we describe the method for training our network. Generally, let n be the number of anchor nodes, data set, L_t, measured by each anchor node and tag node and ground truth of 2D position, Y_t, are represented on the time step t as follows:

$$L_t = (l_1, l_2, ..., l_n)_t \tag{9}$$

$$Y_t = (x_t, y_t) \tag{10}$$

where l_i denotes the the distance between i^{th} anchor node and the tag node. Note that our neural network does not only take a set at the time t but takes sets based on the sequential length of input to our network, T as follows:

$$\mathbb{L}_t = \{L_{t-T+1}, L_{t-T+2}, ..., L_t\} \tag{11}$$

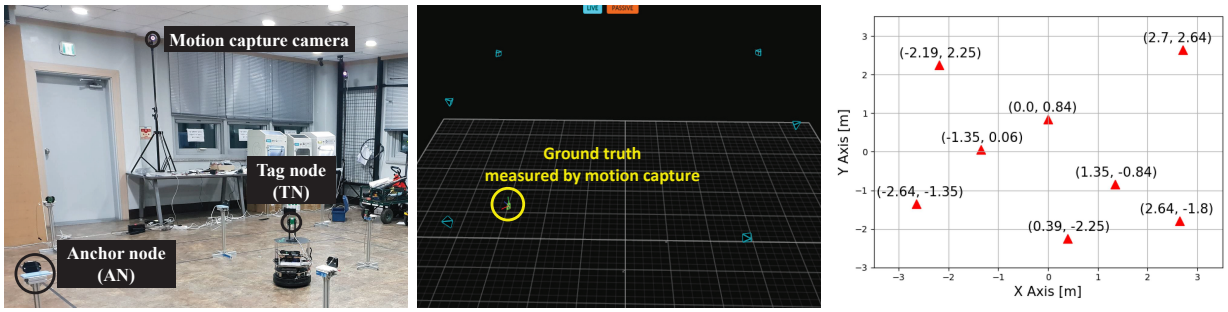
$$\mathbb{Y}_t = \{Y_{t-T+1}, Y_{t-T+2}, ..., Y_t\} \tag{12}$$

Conseqeuntly, neural network could be optimized to be able to localize the mobile node by being trained using the train data set $\mathbb{D}$ as follows:

$$\mathbb{D} = \{(\mathbb{L}_{T-1}, \mathbb{Y}_{T-1}), ..., (\mathbb{L}_t, \mathbb{Y}_t), ...\} \tag{13}$$

Therefore, Let Θ be the parameters of our network model and our final goal is to find optimal parameters Θ^* for precise localization by minimizing L_2 loss term. The L_2 loss term indicates mean square error(MSE) of Euclidean distance between the normalized ground truth position $\mathfrak{N}(Y_k)$ and estimated position $\hat{Y_k}$ as follows:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \frac{1}{N}\frac{1}{T} \sum_{k=T-1}^{N} \sum_{m=k-T+1}^{k} \parallel \mathfrak{N}(Y_m) - \hat{Y_m} \parallel^2 \tag{14}$$



(a) A gull (b) A tiger (c) A mouse

Fig. 4. Pictures of animals

IV. Experimental Results

A. Experimental environment

Our experimental system consists of a UWB sensor tag node attached on the mobile robot platform and eight anchor nodes that take roles of a UWB transceiver, 6 Optitrack Prime 13 motion capture cameras, a Nvidia Jetson AGX Xavier, which is a SFF(small-form-factor) computer that has a GPU. Fig. 4a shows our experimental environment briefly and how to the anchor nodes and the tag node are attatched. And we use the mobile platform, called iClebo Kobuki from Yujinrobot.

The tag node receives the signal and measures the range between two devices based on time of flight(TOF) and Received Signal Strengh Indication(RSSI). Each UWB transceiver, DW1000 UWB-chip made by Decawave, supports 6 RF bands from 3.5 GHz to 6.5 GHz. It measures in cm-level accuracy.

Deep learning framework used for our network is tensorflow-gpu 1.7.0 on python 3.5.2

B. Acquisition of the Train/Test data

UWB sensor anchors are installed randomly in the region where motion capture camaras are acceptable, as Fig. 4c shown. These anchor nodes transmit the UWB signal to tag node that is attached to the mobile robot and the Optitrack motion capture cameras also transmit the ground truth data to the SSF computer by utilizing Robot Operating System(ROS).

Note that these two dat are transmitted by different frequency: range measurements are gathered with a frequency of almost 27Hz, yet the ground truth data are 120Hz. So we synchronize these two data based on the range measurements' Hz. More specifically, we set an independent thread so that this thread select the ground truth data of the nearest time based on the UWB-range measurements, concatenates and saves these data at the same time.

And the mobile robot moves in this sapce by manually. All the trajectories are different. After collecting whole datasets, we separate the entire dataset to three types: one are the training datasets, another are for the validation datasets, and the other is for test dataset. On the test dataset, only range measurements are taken as input to the network.

C. Training the Network

To train our network, 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.

D. Localization Results

Pass

To improve the performance of our proposed networks, we check which seqeunce length is optimal for localization of MN. And then, we implemented and trained other previous networks to compare their performance when our real-world data are given as input. We set three test trajectory cases: a square path, a verical and horizontal winding path, and a diagonal winding path.

1) Performance according to the sequence length: Even though LSTM solve the long term dependency problem, we thought that it would not always make neural network perform better as sequence length become more longer. So, as part of turning hyperparameters, we modified sequence length of our network in a variety of numbers to find optimal sequence length.

As illustrated in FIGURE. **??**, The train data are gathered by the robot that arbitrary moves on the region where the motion capture camera cover. The neural network takes distances measured by each AN and a MN as input and outputs robot's position. The Root-Mean-Squared Error (RMSE) results of trajectory prediction with respect to sequence length are shown in FIGURE. 5 and specific figures are shown in Table I.

As a result, we found that there is a trade-off between robustness and improvement of general performance according to the sequence length. Figure. 5 show that as longer the sequence length, as more inaccurate the mean performance. This is because it is hard for the neural architecture to train the chracteristics of the sequential data. In other words, as sequence length become longer, the tendency of the values may be different due to accumulation of different patterns of system noises even if the data is acquired by moving to a similar path in the train data. Note that network with longer sequence length tend to have less error variance and more a ability to generalize the situation since they could utilize more extended temporal information. By doing so, the nerual

The results of RMSE [cm]				
# of anchors	EKF RO-SLAM	MLP	Bi-LSTM	Ours
3	1	2	3	4
5	a	b	c	d
8	A	B	C	D

network have the ability to suppress the disturbance caused by noises.

Fig. 5. Error bar graph of RMSE with respect to sequence length

TABLE I: Root mean squared error of each case

The results of RMSE[cm]					
	Sequence length				
	2	3	5	7	10
Test1	2.84	2.88	**2.78**	3.65	4.16
Test2	3.61	3.63	**3.57**	3.84	4.18
Test3	**2.93**	3.21	3.03	3.62	3.93

E. Performance comparison Result

The results of trajectory prediction are shown in Fig. 3(a) and Fig. 3(d) and Root-Mean-Squared Error (RMSE) are shown in Table 1. Performance is better in order of stacked Bi-LSTM, Bi-LSTM, LSTM and GRU. In case of GRU, it has only two gates which is less complex structure than LSTM [27]. However, due to GRU's less complexity, GRU has less number of neurons than LSTM so their non-linear mapping achieves less performance. Likewise, Bi-LSTM consists of two LSTMs to process sequence in two directions so that infer output using the correlation of the backward information and the forward information of the sequences of each time step with its two separate hidden layers. Thus, Bi-LSTM has better nonlinear mapping capability than LSTM. For similar reasons, stacked Bi-LSTM is the architecture that stacks two Bi-LSTMs, so inference performance is better than Bi-LSTM. As a result, the stacked Bi-LSTM showed the best performance among unit RNN architectures. Therefore, we can conclude that the performance improves as the non-linearity of the architecture increases.

!!!!!!!!!!!!!!!! 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. We also provide statistical analysis from simulations demonstrating that our new approach can cope with highly noisy sensors and reduces in one order of magnitude the average errors of the method proposed !!!!!!!!!!!!!!!!

V. 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] L. Peneda, A. Azenha, and A. Carvalho, "Trilateration for indoors positioning within the framework of wireless communications," in *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE*. IEEE, 2009, pp. 2732–2737.

[2] J. Jung and H. Myung, "Indoor localization using particle filter and map-based nlos ranging model," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5185–5190.

[3] P. Newman and J. Leonard, "Pure range-only sub-sea slam," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. Ieee, 2003, pp. 1921–1926.

[4] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006.

[5] J. Li, X. Yue, J. Chen, and F. Deng, "A novel robust trilateration method applied to ultra-wide bandwidth location systems," *Sensors*, vol. 17, no. 4, p. 795, 2017.

[6] F. R. Fabresse, F. Caballero, I. Maza, and A. Ollero, "An efficient approach for undelayed range-only slam based on gaussian mixtures expectation," *Robotics and Autonomous Systems*, vol. 104, pp. 40–55, 2018.

[7] J. González, J.-L. Blanco, C. Galindo, A. Ortiz-de Galisteo, J.-A. Fernández-Madrigal, F. A. Moreno, and J. L. Martínez, "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 496–507, 2009.

[8] M. S. Rahman, Y. Park, and K.-D. Kim, "Localization of wireless sensor network using artificial neural network," in *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*. IEEE, 2009, pp. 639–642.

[9] M. Abdelhadi, M. Anan, and M. Ayyash, "Efficient artificial intelligent-based localization algorithm for wireless sensor networks," *Journal of Selected Areas in Telecommunications*, vol. 3, no. 5, pp. 10–18, 2013.

[10] S. Kumar, R. Sharma, and E. Vans, "Localization for wireless sensor networks: A neural network approach," *arXiv preprint arXiv:1610.04494*, 2016.

[11] H. LIM, J. GOO, and H. Myung, "Effective indoor robot localization by stacked bidirectional lstm using beacon-based range measurements," in *International Conference of Robotics Intelligence and Applications (RiTA)*. Universiti Malaysia Pahang, 2018.

[12] 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.

(a) A gull (b) A tiger (c) A mouse

Fig. 6. Pictures of animals

[13] 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.

[14] 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.

[15] 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.

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

[17] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4762–4769.

[18] 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.

[19] 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.

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

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

[22] 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.

[23] M. Patel, B. Emery, and Y.-Y. Chen, "Contextualnet: Exploiting contextual information using lstms to improve image-based localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[24] 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.

[25] 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.

[26] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data." in *AAAI*, vol. 1, no. 2, 2017, pp. 4263–4270.

[27] X. Wang, Z. Yu, and S. Mao, "Deepml: Deep lstm for indoor localization with smartphone magnetic and light sensors," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[28] C. Chen, X. Lu, A. Markham, and N. Trigoni, "Ionet: Learning to cure the curse of drift in inertial odometry," *arXiv preprint arXiv:1802.02209*, 2018.

[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] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," *arXiv preprint arXiv:1704.06904*, 2017.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[38] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[39] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional lstm with cnn features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[40] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[41] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.