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, advancement in micro-electo-mechanical systems(MEMS) contributes the improvement of sensors' performance and miniaturization at the same time in such a way as to have enabled the industiral robots to have mobility. Accordingly, studies about techniques for localizing a mobile robot and recognizing surrounding environments, called Simultaneous Localization and Mapping(SLAM) [1], become more important and has been conducted actively. Specifically, Range-only SLAM(RO-SLAM) have addressed the problem of SLAM with a set of range-only measurements between a mobile robot and landmarks obtained from range sensors such as UWB, ultrasonic, laser-based beacon sensors. By virtue of this low-cost, small-size, acceptively accurate performance, and convenience of being installed, RO-SLAM have been suggested as a solution for localization on the indoor environment [2], [3] where the signals of the Global Positioning Systems(GPS) cannot be received, e.g., indoor environment [2]–[4] and underwater environment [5], [6]

The range measurements are usually measured by Time-of-Flight(TOF) and only consist of single value to represent distance between each landmark and mobile robot respectively. This cause two problems: one is that it is very vulnerable to noise because the distance are measured by only one-dimensionl information and the other one is *rank dificiency* problem [7]. In other words, the single value to represent distance between each landmark and mobile robot respectively is deficient to describe exact position or orientation of the landmark. Besides, these measurements could have huge uncertainties caused by multipath fading channel(MPF) problem [8] 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.

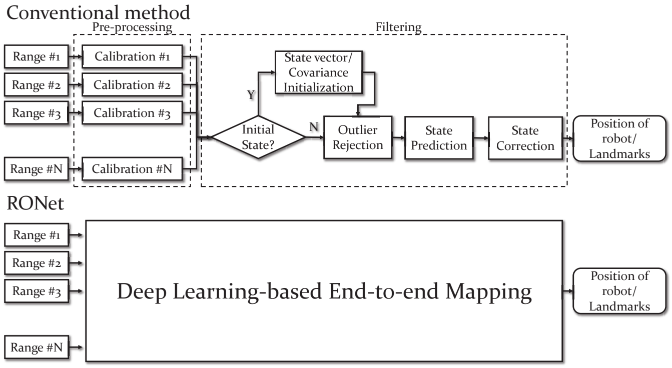

Fig. 1. Comparison between a standard range-only localization or SLAM framework and our learning-based approach.

In this paper, we propose a robust stacked bidirectional Long Short-Term Memory(stacked Bi-LSTM) with residual attention for more accurate localization of the robot. To the best of our knowledge, it is a first approach to apply LSTM-based architecture to localize the mobile robot on the real-world in real-time using range-only measurement. Since it is achieved in an end-to-end mapping between range measurement and robot position, it does not need any preprocessing module used in the conventional RO-SLAM or range-only localization, such as an calibration and outlier rejection. Our contribution is threefold as follows:

Using LSTM-based architecture, our structure directly learns the end-to-end mapping between range measurements and robot position. This operation non-linearly maps the

*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

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.

- Our structure directly learns the end-to-end mapping between sequential range measurements and robot position. Robust with minimal range sensors.
- Using neural networks, performance is higher than traditional method.
- Real-time on the real world

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.

[9] !!!! In the meantime, as deep learning age has come [10],

In this paper, we propose a stacked bidirectional Long Short-Term Memory(stacked Bi-LSTM) with residual attention for more accurate localization of a MN. Our contributions can be summarized as follows. First of all, Using deep learning, our structure directly learns the end-to-end mapping between range data measured by ANs-to-MN distances and the position of MN on three dimensional space. Unlike other approaches whose authors implement MLP-based neural networks, we exploit LSTM architecture in such a way as not only to take present range data but also take previous temporal data as input for estimating the MN's position. Second, we verify that effect of the residual attention in the way that the residual attention layer makes the network better understand context of the feature when dealing with low-dimensional input. Finally, we implement the networks presented at related works and compare which neural architecture perperforms better when real-world data are given as input. As a result, and our networks perform the best performance among other proposed approaches. Our system overview is shown in the figure

II. Related Works

A. Conventional RO SLAM

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 [4], [11], [12] 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 [13]–[16]. communicate with other wireless sensors efficiently and conveniently.

In the past few years, researchers have conducted the studies using machine learning-based approaches to reduce computational complexity and localize a mobile node more precisely. One authors utilized support vector machine(SVM) for localizaiton, [17]–[20], other author developed method support vector regression(SVR) for localiztion [21], [22]. In [17], authors suggested two SVMs for localization, called LSVMs, one LSVM infers x-dimension and the other LSVM infers y-dimension. To employeeing LSVMs, they divide the field into *M-l* x-classes and *M-l* y-classes, like grid, and this deployment has had an impact on succeeding studies [19], [20], [23]. Samadian *et al.* [24] introduced probablistic support vector machine for localization and they showed that probablistic vector machine has better performance than LSVM. In terms of SVR, Lee *et al.* suggested various types of SVR for localization [21], [22]

Especially, several works have been studied using neural networks to localize nodes of the range measurement sensors on the indoor space while covering range measurements' uncertainties. Regarding previous proposals, Chenna *et al.* first shows the suitability that Kalman filter could be replaced with the RNN when estimating states and tracking nodes [25]. However, they did not provide numerical analysis, so Shareef *et al.* did and conducted their experiment in the real-world. They concluded Radial Basis Function(RBF) may be the best option among the suggested Kalman filter models and RNN [26].

B. Deep Learning and LSTM-based Sequential Modeling

As deep learning age has come [10], various kinds of deep neural architectures have been proposed for many tasks related to robotics field. Especially, recurrent neural networks (RNNs), originated from Natural Language Process(NLP) area [27], 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 [28]–[32].

C. Deep Learning for Low-dimensional Sensor Data

WSNs fields also have introduced various kinds of neural network architectures [33]–[37]. 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.

Similarly, many researchers also have achived considerable improvement to localize position of mobile node by exploiting MLP in WSNs fields [26], [33]–[38]. In case of range-based method, Rahman *et al.* [33] have considered the neural networks for mapping between RSS and corresponding position of sensor nodes and let neural networks be trained by the train data gathered by the sensor nodes that are eqaully spaced over x-axis and y-axis. In [34], Singh *et al.* compared that performance of Multilayer Back propagation Network Model(MLBPN) and Radial Basis Function Network Model(RBFN) and the authors show that RBFN performs better than MLBPN when the number of the sensor nodes is larger than 220. Abdelhadi *et al.* [35] presented two artificial intelligence techniques: Sugeno-type fuzzy system and neural networks system. In addition, the authors conducted experiment on three-dimensional space in such a way as verified the feasibility of localization by utilizing nueral networks in 3D space. Kumar *et al.* [36] also introduced the neural networks and evaluated five different training techniques,e.g., Levenberg-Marquardt (LM), Bayesian Regularization (BR), Resilient Back-propagation (RP), Scaled Conjugate Gradient (SCG) and Gradient Descent (GD), to find optimal way to train neural networks with the best accuracy. Recently, [37] have proposed the neural networks with novel training technique, called Particle Swarm Optimization(PSO) and prove thire nerwork, called LPSONN, has better localization accuracy than previous machine learning method, soft computing method, and previously proposed network.

However, there are some points that could have been better. First of all, in some cases, their networks were trained by range measurement data corresponding position of mobile node in simulation environment [23], [26], [33], [34], [37]. The simulation situation is almost ideal in the point that the NLOS or MPF problem does not occur. Even though in [33], they artificially generate NLOS data, but the gap of complexity between real-world and artificially generated data exists, since the generated data has less noise than those of real-world necessarily. For these reasons, it is hard to say that their networks also works well on real world situation. Therefore, to test on whether it is possible for neural networks to estimate position with covering all disturbance, the experiment should be conducted on real-world. Second, some studies was conducted on the two-dimensional space to simplify their problem definition [26], [33], [34], [36]–[38]. However, gap of the complexity of estimating position between on two-dimensional space and on three-dimensional using *rank-deficient* measurements cannot be negligible. Finally, in previous studies [26], [33], [34], [36], it has a possibility of overfitting because the authors generate the grid-map as train data in such a way as to restrict their ground truth region. That is to say, their finite groud truth indicates where the sensors are placed at the equal distance interval so that nerual networks may recognize the only locations included in the grid are correct even when the position of MN to be tested is quite far from the grid. Therefore their grid map train impedes the optimization of nueral netorks to cover all over the region.

III. RONet

In this chapter, we explain how our proposed residual attention-based stacked Bi-LSTM is implemented, as illustrated in Fig. **??**. In detail, we introduce the neural networks concepts that we choose for localizaing the tag node and the describe the reason why we let the neural network infer in three-dimensional space even though experiment is conducted on the mobile robot. Finally, we explain how to set the loss function of our neural network and then compare to those of other previous works.

A. long short-Term Memory

Recurrent Neural Networks(RNN) is a special artificial neural networks in the way that it has a loop, so that RNN can deal with temporal information for sequential modeling. It originally used in the natural language processing, speech recognition, and image captioning area. By virtue of a loop, RNN can remember past data and past situation and respond appropriately to the present situation based on these past information.

But unfortunately, as the time-sequential gap grows, RNNs become unable to learn the relationship of these sequential information. This issue is called the problem of *Long-Term Dependency*,which fail to propagate the previous matter into present tasks so that long-term dependency lead to a failure of learning. 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 is introduced to solve this long-term dependency problem and make the networks possible to learn longer-term contextual understandings [39]. That's why LSTM have been actively studied for many tasks in a wide area of science and engineering.in most of the deep learning research areas and numerous variations of LSTM architecutres have been studied.

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:

TABLE I: Comparison of previous studies with our approach

Localization method	Dimension	Type of input	Train data	Test data mobility	Implementation environment
RBF [26]	2D	Temporary	Grid	Dynamic nodes ✓	Real-world ✓
MLP [33]	2D	Temporary	Grid	Static nodes	Simulation
MLBPN [34]	2D	Temporary	Grid	Static nodes	Simulation
MLP [35]	3D ✓	Temporary	Spread ✓	Static nodes	Simulation
c-FCNNs [38]	2D	Temporary	Spread ✓	Static nodes	Real-world ✓
MLP [36]	2D	Temporary	Grid	Static nodes	Real-world ✓
LPSONN [37]	2D	Temporary	Spread ✓	Static nodes	Simulation
Ours	3D ✓	Sequential ✓	Spread ✓	Dynamic nodes ✓	Real-world ✓

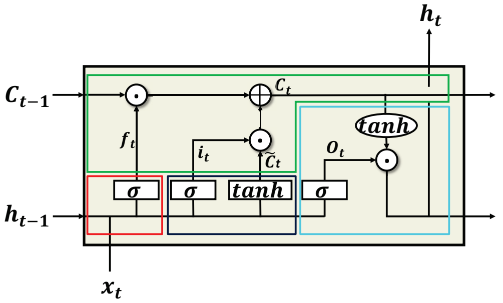

Fig. 2. Architecture of the LSTM. It consists of 3 gates, forget gate(the part inside the red box), input gate(the part inside the blue box), and output gate. And output gate is divided into cell state layer(the part inside the green box) and output gate layer(the part inside the cyan box)

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

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

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

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

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

$$h_t = o_t \odot \tanh\big(c_t\big) \tag{6}$$

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. The sigmoid layer, which is the activation function of f_t, takes previous hidden state, h_t, and present input, x_t and outputs a number between 0 and 1. Note that 1 indicates "totally keep the previous cell state, C_{t-1}" and 0 indicate "totally forget C_{t-1}" (1). Next, the input gate, i_t, decides how much information to embrace when updating the cell state. i_t are also from the sigmoid function layer (2) and $tanh$ generates the new candidate cell state, $\tilde{c}_t$, which ranges from -1 to 1 (3). After that, c_t is updated by the cell state layer based on f_t, i_t, and $\tilde{c}_t$ (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 present hidden state, h_t, is updated based on o_t updated cell state, c_t (6).

B. Bidirectional LSTM

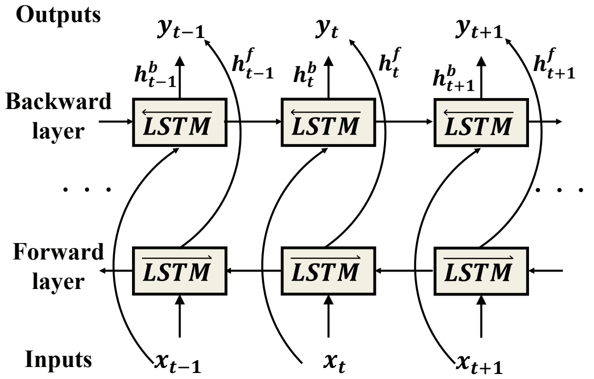

Fig. 3. Architecture of the Bidirectional LSTM(Bi-LSTM)

One shortcoming of conventional RNNs is that they only exploit previous context to update h_t and c_t. However, in many cases dealing with sequential data, it could be efficient to extract well-discribed context by utilizing future context as well. Bidirectional RNNs are introduced [40] for that reason and bidirectional RNNs process the data in both directions with two separate hidden layers. Especially, bidirectional LSTM, which we employee, has one forward LSTM and one backward LSTM running in reverse time and their features are combined at the output layer, y_t. As a result, bidirectional LSTM can produce more appropriate context considering both past and future at the same time. By virtue of this characteristics, bidirectional LSTM is popularly utilized for many tasks to model their sequentional systems [41]–[43].

As FIGURE. 3 shown, bidirectional $\overleftrightarrow{\text{LSTM}}$ consist of 2 LSTMs: one forward LSTM layer, $\overrightarrow{LSTM}$, and one backward LSTM layer, $\overleftarrow{LSTM}$. Let assume the hidden state of $\overrightarrow{LSTM}$ at the time step t be h^f_t and $\overleftarrow{LSTM}$ at the time step t be h^b_t, the hidden states and output sequence, y_t are calcaulated as follows:

$$h_t^f = \overrightarrow{LSTM}\big(x_t, h_{t-1}^f\big) \tag{7}$$

$$h_t^b = \overleftarrow{LSTM}\big(x_t, h_{t+1}^b\big) \tag{8}$$

$$y_t = \sigma_R\big(W_{h^f y} \cdot h_t^f + W_{h^b y} \cdot h_t^b + b_y\big) \tag{9}$$

where σ_R denotes activation function called ReLu. Note that in our case, we concatenate h_t^f and h_t^b to preverse their contexts seperately as our range measurement, which is gathered by the tag node and each anchor node, suffer from the *rank-dificiency*,which means range-based measurement consist of one-dimensional data [15]. Hence, we judge that it would be more helpful to increase the number of features naturally by concatenating two hidden states rather than adding them and when the network infer the position .

C. Stacked Architecture

Recently, researchers show that the deeper the architecture of neural networks, the better their performance [44], [45] and their demonstrations has opened a deep learning area. Likewise, many authors have analyzed variations of LSTM architecture and find out that stacking multiple layers of the LSTM improve the performance for many tasks [43], [46], [47]. In other words, as the number of stacked layers is getting large, the more activation functions which rise the non-linearity within the networks are stacked in such a way as that complexity of networks increases. As a results, networks could model more complex system by virture of these increased non-linerity.

Therefore, we also construct our networks by stacking two LSTM to increase the non-linearity. Note that stacking more than three LSTM doesn't show the improvement of performance. We suppose that activation funtions within the LSTM cause the *vanishing gradient problem* [48], which the networks fail to training due to the fact that the gradient is getting closer to zero during the backpropagation. We deem that this problem could comes from the sigmoid function and $tanh$ function that compose the part of LSTM. Consequently, we put the ReLu function between LSTMs to avoid the vanishing gradient problem [49], instead of stacking 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 measured by each anchor node and tag node are represented on the time step t as follows:

$$L_t = (l_1, l_2, ..., l_n)_t \tag{10}$$

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 of distance data at the time t but takes sets of distacne data based on the sequential length of input to our network, T. In other words, train data consist of T sets of distance data corresponding to the position of mobile nodes. Let $\mathbb{L}_t$ be

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

Conseqeuntly, neural network could be optimized to be able to localize the mobile node when take distance sets as input. Train data $\mathbb{D}$ are formulated as follows:

$$\mathbb{D} = (\mathfrak{N}(\mathbb{L}_t), \mathfrak{N}(Y_t)) \tag{12}$$

where Y_t denotes the ground truth of the robot's 2D position, which is denoted as $Y_t = (x_t, y_t)$ and $\mathfrak{N}$ denotes normalization function that transforms data to min-max scaling 0 to 1. Note that range measurements mesured by each range sensor are processed by each min-max scaling because each range sensor has own characteristic, as one calibrate range sensors respectively.

Let Θ be the parameters of our network model and assume that the trained network model could be expressed as conditional probability as follows:

$$P(\mathfrak{N}(Y_t)|\mathfrak{N}(\mathbb{L}_t)) = p((x_t, y_t)|\,\{L_{t-T+1}, L_{t-T+2}, ..., L_t\}) \tag{13}$$

Note that our approach takes sequential range data on the sequence length T so that extract temporal features to map sets of the range measurements to position and also learn how to cover the noises and outliers, which temporarily caused by the MPF problem or unstable signal conditions. Then, our final goal is to find optimal parameters Θ^* for 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:

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

IV. Experiments

A. Experimental environment

Our experimental system consists of a UWB(ultra wideband) sensor tag node attached on the mobile robot platform and eight anchor nodes that have a UWB transceiver, the motion capture system with 12 cameras, and a SFF(small-form-factor) computer.

UWB sensor anchors are attached to landmarks. These become the end points of the range measurements. The anchor nodes transmit the UWB signal. A UWB sensor tag is attached to a robot. It becomes the opposite side end point of the measurements. The tag node 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. Fig. 4 shows our experimental environment briefly and how to the anchor nodes and the tag node are attatched.

We inference the position of a robot with our network. To train the network and test the results, the ground truths are needed. We get the ground truth by using the motion capture system. The system is Eagle Digital Realtime system of motion analysis corporation that operates with the principle

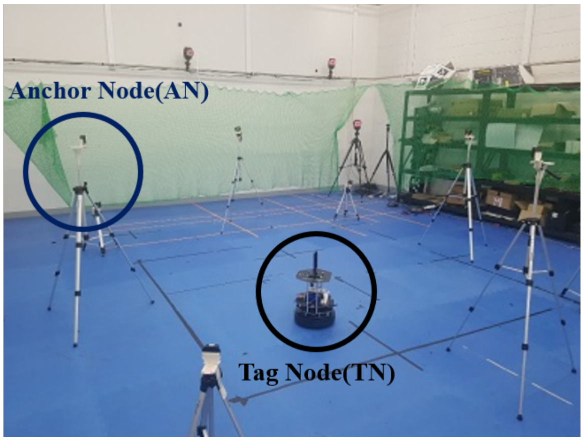

Fig. 4. our experimental environment briefly and how to the anchor nodes and the tag node are attatched.

of stereo pattern recognition that is a kind of photogrammetry based on the epipolar geometry and the triangulation methodology. We attach four markers to a robot. The system gives us the location of these markers and has < 1mm accuracy.

A mobile robot used in experiment is iClebo Kobuki from Yujinrobot that has 70 cm/s maximum velocity.The SFF 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.

The UWB tag is attached to mobile robot that has a SFF computer. The UWB anchors are attached to stands that have two different heights. The anchors are positioned randomly in the square space where the motion capture system can measure the position. The robot moves in this sapce also. As you can see in Fig. 5, a mobile robot manually goes on various random trajectories by experimenters.

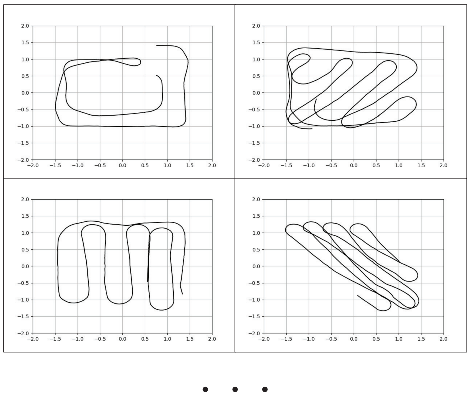

Fig. 5. Four example of the trajectories

B. Data syncronization for Train/Test data

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. These two kinds of data are paired in a dataset. Because these two data are transmitted by different frequency, we need to syncronize these. This process is conducted in a SFF computer. The computer receives these two kinds of data respectively and syncronizes these by time. To synchronize, we make an independent thread that concatenates and saves these data at the same time. The data is saved at 20Hz frequency. Each trajectory becomes one dataset. All the trajectories are different. Fig. 6 shows this process. After collecting whole datasets, we separate the entire dataset to two types, some are the training datasets and others are test datasets.

Ground truth data is robot's position measured by eagle eye motion capturer, whose error is in mm units.

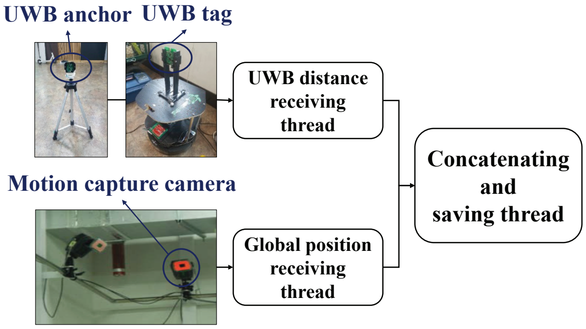
UWB anchor
UWB tag
UWB distance receiving thread
Motion capture camera
Global position receiving thread
Concatenating and saving thread

Fig. 6. the process that makes dataset

C. Training the Networks

Pass

V. Results

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.

A. 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. 5, 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. 7 and specific figures are shown in Table II.

As a result, we found that there is a trade-off between robustness and improvement of general performance according to the sequence length. Figure. 7 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 network have the ability to suppress the disturbance caused by noises.

Similarily, as sequence length become shorter, it is less difficult for the neural architecture to understand the chracteristics of the sequential data by virtue of the less accumulation of different patterns of system noises. Note that in most cases, the network constructed by the shorter sequential length tend to estimate the position more precisely than those with longer sequential lengths. However, it becomes a double-edged sword because temporal information is also reduced accordingly in such a way as to week to the noises, having larger error variance.

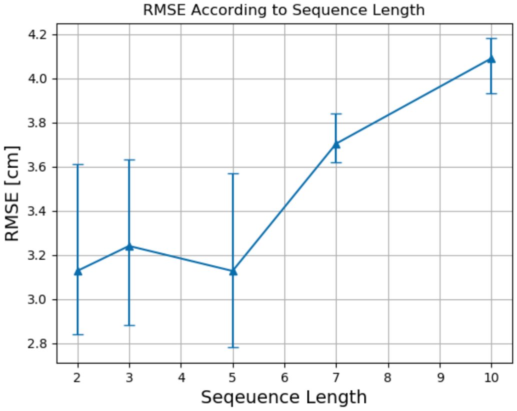

Fig. 7. Error bar graph of RMSE with respect to sequence length

TABLE II: 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

B. 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 !!!!!!!!!!!!!!!!!

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] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] 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.

[3] 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.

[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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 981–994, 2008.

[18] R. Huan, Q. Chen, K. Mao, and Y. Pan, "A three-dimension localization algorithm for wireless sensor network nodes based on svm," in *Green Circuits and Systems (ICGCS), 2010 International Conference on*. IEEE, 2010, pp. 651–654.

[19] V.-s. Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 505–512, 2012.

[20] S. Afzal and H. Beigy, "A localization algorithm for large scale mobile wireless sensor networks: a learning approach," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 98–120, 2014.

[21] J. Lee, W. Chung, and E. Kim, "A new kernelized approach to wireless sensor network localization," *Information Sciences*, vol. 243, pp. 20–38, 2013.

[22] J. Lee, B. Choi, and E. Kim, "Novel range-free localization based on multidimensional support vector regression trained in the primal space," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 7, pp. 1099–1113, 2013.

[23] A. Chatterjee, "A fletcher–reeves conjugate gradient neural-network-based localization algorithm for wireless sensor networks," *IEEE transactions on vehicular technology*, vol. 59, no. 2, pp. 823–830, 2010.

[24] R. Samadian and S. M. Noorhosseini, "Probabilistic support vector machine localization in wireless sensor networks," *ETRI Journal*, vol. 33, no. 6, pp. 924–934, 2011.

[25] S. K. Chenna, Y. K. Jain, H. Kapoor, R. S. Bapi, N. Yadaiah, A. Negi, V. S. Rao, and B. L. Deekshatulu, "State estimation and tracking problems: A comparison between kalman filter and recurrent neural networks," in *International Conference on Neural Information Processing*. Springer, 2004, pp. 275–281.

[26] A. Shareef, Y. Zhu, and M. Musavi, "Localization using neural networks in wireless sensor networks," in *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2008, p. 4.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] P. Singh and S. Agrawal, "Tdoa based node localization in wsn using neural networks," in *Communication Systems and Network Technologies (CSNT), 2013 International Conference on*. IEEE, 2013, pp. 400–404.

[35] 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.

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

[37] S. S. Banihashemian, F. Adibnia, and M. A. Sarram, "A new range-free and storage-efficient localization algorithm using neural networks in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 1, pp. 1547–1568, 2018.

[38] M. Bernas and B. Płaczek, "Fully connected neural networks ensemble with signal strength clustering for indoor localization in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 12, p. 403242, 2015.

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

[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] F. Zhang, C. Hu, Q. Yin, W. Li, H.-C. Li, and W. Hong, "Multi-aspect-aware bidirectional lstm networks for synthetic aperture radar target recognition," *IEEE Access*, vol. 5, pp. 26 880–26 891, 2017.

[42] W. Li, W. Nie, and Y. Su, "Human action recognition based on selected spatio-temporal features via bidirectional lstm," *IEEE Access*, vol. 6, pp. 44 211–44 220, 2018.

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.