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— **In this study, a three-layered bidirectional Long Short-term Memory (Bi–LSTM) with residual attention, referred to as RONet, is proposed to achieve localization using range measurements. Accordingly, we acquired our own datasets and tested RONet using realistic conditions. It is shown that the RONet can estimate the position of the mobile robot in real time using the Nvidia Jetson AGX Xavier based only on range measurements.**

We also analyzed the sequence length of LSTM as a type of hyperparameters. We found that a sequence with a length equal to eight is optimal compared to sequences with lengths equal to two, three, five, and 12, given that construction of the network with the optimal sequence length estimates the position precisely and accounts for uncertainties.

As verified experimentally, RONet yields a better and more precise performance, and results in an increased robustness against outliers compared to a conventional range-only approach based on a particle filtering and the use of deep-learning-based approaches. We set three cases, reduced the number of anchors, and verified that the RONet was a robust solution. We also confirmed that it is the only solution that yields the smallest Root-Mean-Square Error (RMSE) values, equal to 4.466 cm, 3.210 cm, and 3.090 cm, in the cases where three, five, and eight anchors were implemented, respectively.

I. INTRODUCTION

Given the increasing demand in recent years to achieve localization in indoor environments with global positioning systems (GPS) attributed to the fact that signals from these systems could become imprecise, numerous researchers have proposed various methods for locating objects based on the use of magnetic fields, acoustic signals, or laser-based data [1]–[3]. Among them, beacon sensors based on the time-of-flight (TOF) have been extensively utilized by virtue of their characteristics, including their low cost, small size, accurate performance, and convenience of installation. As a result, these range measurement-based approaches have been suggested as a solution for localization in indoor [4], [5] and underwater environments [6], [7].

Specifically, these range-only approaches have addressed the problem of localization with sets of range-only measurements between object nodes that need to be localized, referred to as the tag nodes, and landmarks, referred to as anchor nodes. However, range measurements only represent distances between each landmark and the mobile robot. In other words, one-dimensional data (range-only observations) are associated with two problems: a) they tend to be nonlinear because TOF-based measurements are vulnerable to noise and are highly uncertain owing to the multipath fading channel (MPF) problem [8] in real-world applications, and b) they are associated with a *rank deficiency* problem [9]. Specifically, the single value used to represent the distance between each landmark and the mobile robot is deficient in describing the exact position or orientation of the landmark, thereby leading to a multimodal distribution [10].

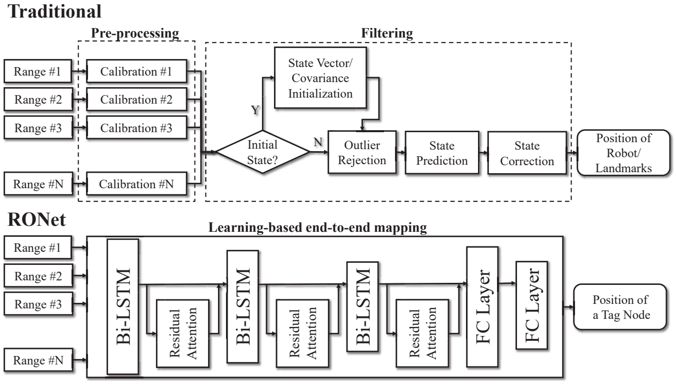


Fig. 1. Comparison between a conventional probabilistic-based range-only frameworks and our learning-based approach.

To alleviate these issues, many studies have been conducted based on probabilistic Bayesian inference frameworks and Monte-Carlo Bayesian filters. However, in recent years, numerous attempts have been expended to solve these problems based on neural-network-based approaches [11]–[14]. With nonlinear end-to-end mapping, prior research studies showed feasibility. However, in most cases, researchers only utilized multilayer perceptrons (MLP) that constitute the front-end architecture of deep learning fields [11]–[13]. In [14], a stacked bidirectional long short-term memory (Bi–LSTM) layer was implemented to account for the noise of range observations based on the network's characteristics that accepted temporal sequential values as input. However, the authors only tested this formulation using a simulated environment [11]–[13]. Additionally, none of the prior conducted studies investigated whether their learning-based approaches could be executed in real time or not.

In this study, we propose a robust stacked Bi–LSTM network with residual attention, referred to as RONet. To the best of our knowledge, it is the first approach that applies an LSTM-based architecture to localize a mobile robot in real

*This study was supported by the Ministry of Trade, Industry & Energy(MOTIE, Korea) based on the 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]Changgyu Park, and [3]Hyun Myung are with the Urban Robotics Laboratory, Korea Advanced Institute of Science and Technology (KAIST) Daejeon, 34141, South Korea. shapelim@kaist.ac.kr, cpark@kaist.ac.kr, hmyung@kaist.ac.kr

time using only range measurements in realistic applications. Unlike conventional probabilistic-based algorithms, it does not need any preprocessing modules, such as calibration, or outlier rejection modules. Besides, RONet yields a better and more precise performance, and results in an increased robustness against outliers compared to a conventional range-only approach based on a particle filtering and the use of deep-learning-based approaches.

Our contribution is threefold:

- We develop three-stacked Bi–LSTM layers on which residual attention layers are attached to allow the neural network be trained well so that the RONet yields the best performance compared to previous approaches
- We also analyze how the sequential length of the network affects performance and evaluate the robustness of the RONet with a minimum number of anchors
- We operate the RONet on Nvidia Jetson AGX Xavier and confirm on whether the inference frequency (which is approximately equal to 32 Hz) complies with real-time frequencies

The rest of the study is organized as follows: Section II overviews related, previously published studies. Section III describes our neural network in detail and defines the study's problem, and Section IV describes the experimental results. Finally, Section V summarizes our contributions and describes future work.

II. Related Works

A. Conventional Range-Only Localization

There are two conventional approaches to localize a mobile robot using range measurements: methods based on a) Kalman filtering (KF) or b) particle filtering (PF). However, unlike other sensors, it is difficult to approximate range measurements using linear models owing to MPF or Non-line-of-sight issues (NLOS). Thus, some authors insisted that PF-based approaches could be better than KF-based approaches because PF can cope with complex nonlinear models and also cover multimodal distributions [10], [15], [16].

Fig. 1 shows the general steps used by the conventional probabilistic approaches. First, each range of measurements needs to be calibrated. After initialization, the algorithm checks whether an input value is an outlier or not, and then eliminates unexpectedly large noise values. Subsequently, it predicts the present states, including the mobile robot's pose and anchor locations, and finally it corrects its prediction using range observations.

B. LSTM-based Sequential Modeling

In view of the emergence and rapid development of deep learning approaches [17], various types of deep neural architectures have been proposed for localization tasks [18]–[20]. Specifically, recurrent neural networks (RNNs) that originated from the natural language process (NLP) area [21] have been shown to achieve better performance in cases associated with time variant information.

Additionally, despite the fact that long short-term memory (LSTM) architecture solves the *long-term dependency* issue that is inherent to RNNs, it is unable to learn the relationship of sequential information as the time-sequential gap grows [22]. Accordingly, LSTM is actively introduced to learn longer-term contextual understandings. Therefore, many authors exploited LSTM for sequential modeling after feature extraction by Convolutional Neural Networks (CNN) [23]–[25].

C. Deep Learning for Range Only Localization

LSTM was also utilized to model low-dimensional sensor data by itself. In [26], the authors exploited LSTM for indoor localization with magnetic and light sensors. Additionally, in [27], the authors estimated two-dimensional (2D) Odometries via stacked Bi–LSTM that accepted only Inertial Measurement Unit (IMU) sensor data as input.

Regarding our objective for localization using range-only measurements, we note that many authors had previously employed neural network-based approaches in wireless sensor networks fields (WSNs) [11]–[13], yet most networks had been based on the MLP. Their approaches only mapped a set of range observations without temporal context to achieve localization. Accordingly, their approaches may be more potentially week to unexpected noise that are not included in the train data. Another study [14] implemented stacked Bi–LSTM for localization of a mobile robot that accepted sequential range measurements from anchors and tags. However, the authors only conducted simulations in which there were no MPFs or unexpected noise. Therefore, in this study, we conducted realistic experiments to verify the feasibility of the approaches to account for all the noise sources from all sensors with sequential range information followed by the comparison of these approaches.

D. Attention Layer

An attention layer is powerful module nowadays and mostly improves the performance of a neural network. Originally, neural networks treat information equally. However, using attention layers, neural networks can be examined closely. Accordingly, the attention layer assumes the role of a feature selector [28]. Initially, attention was utilized at the NLP areas for the improvement of the translation performance [29]. However, nowadays, the attention layer is employed in many areas to improve the performance of the networks.

III. RONet

In this chapter, we explain how our proposed residual attention-based stacked Bi–LSTM is implemented, as illustrated in Fig. 2. Specifically, we introduce the stacked Bi–LSTM and residual attention module that we choose for localizing the tag node, and we then compare the outcomes to those from other previous publications. Finally, we describe how to set the loss function of our neural network.

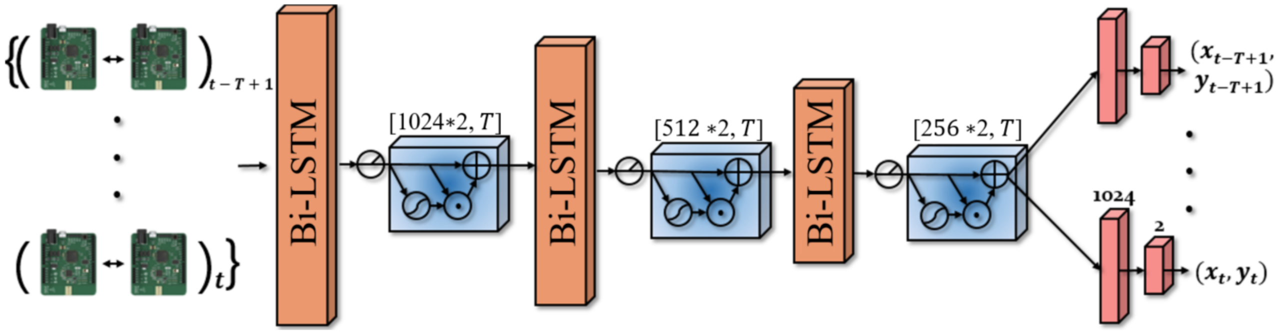


Fig. 2. Our networks consist of three elements: a) Bi–LSTM, b) the residual attention module (the blue cuboid), and c) the fully connected layer (FC layer). Features are input to the Bi–LSTM, which reduces the number of features in half from 2048 to 1024 to 512. Finally, extracted features are input to the FC layer to estimate the position corresponding to each time step.

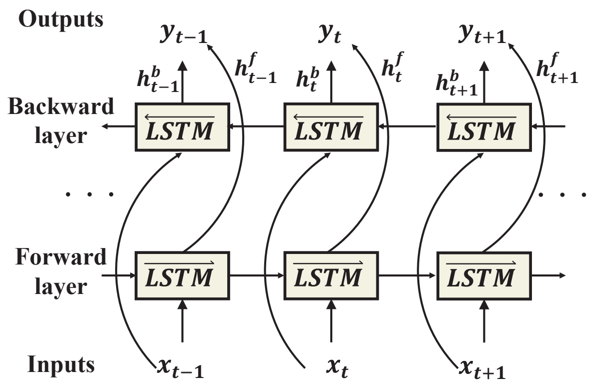


Fig. 3. Architecture of the bidirectional LSTM (Bi–LSTM). The bidirectional LSTM consists of two LSTMs: one forward (right arrow) and one backward (left arrow) LSTM layer.

A. *Long Short-Term Memory*

Unlike the RNN that only consists of a hidden state, in LSTM a cell state is added to the network. The cell state consists of three gates to preserve the previous information and control the cell state, i.e., the a) forget, b) input, and the c) output gates. The respective equations of these gates are as follows,

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

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

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

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

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

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

where σ_s is an activation function, known as *sigmoid*, f_t, i_t, and o_t, respectively denoting the forget, input, and output gates, while c_t denotes the cell states. Furthermore, $\odot$ denotes element-wise multiplication, referred to as the *Hadamard product*. All the gates are activated by the sigmoid function and the cell states are activated by the $\tanh$ function.

The forget gate layer, f_t, determines how much information to forget based on the previous hidden state, h_{t-1}, and the present input, x_t. Subsequently, the input gate, i_t, decides how much information to include when the cell state is updated. Accordingly, c_t is updated by the cell state layer based on f_t, i_t, and by the candidate cell state, $\tilde{c}_t$ (4). In addition, the output gate layer, o_t, serves as a filter, which means that o_t determines the values which will be output (5) in such a way that h_t is updated based on the updated cell state of o_t, c_t (6).

B. *Stacked Bidirectional LSTM*

Based on the fact that the deeper the architectures of the neural networks are, the better their performances [30], [31], many authors have analyzed variations of LSTM architectures and found that stacking multiple LSTM layers improves the performance for many tasks [32]–[34]. Furthermore, bidirectional RNNs are introduced [35] to extract well-described contexts. They consist of one forward LSTM, $\overrightarrow{LSTM}$, and one backward LSTM, $\overleftarrow{LSTM}$, running in reverse time so that the network exploits the previous forward context to update h_t and c_t, and the future backward context as well, as shown in Fig. 3.

For these reasons, we decided to implement the stacked Bi–LSTM architecture to model the system. By virtue of the increased nonlinearity caused by the number of stacked layers, the network could model more complex localization schemes by considering the UWB-ranging observations as the input that contain unexpected noise and MPF problems. Furthermore, we determined that Bi–LSTM would be more helpful in the production of more appropriate context by concurrently considering both the past and the future.

Therefore, we constructed our networks by stacking three Bi–LSTM networks to increase the nonlinearity. Note that stacking more than three LSTM layers does not result in additional performance improvements. We consider that this problem arises from the sigmoid and $tanh$ functions which are parts of the LSTM. These activation functions cause the *vanishing gradient problem* [36] according to which the networks fail to perform training owing to the fact that the gradient is getting closer to zero during the backpropagation. Consequently, we placed the Rectified Linear Unit (ReLu) function between the LSTMs to avoid the vanishing gradient problem [37] instead of stacking more LSTM layers to increase nonlinearity. In addition, experiments showed that reducing the hidden size of the next LSTM layer when the features are input into the LSTM layer slightly increases performance. In conclusion, we decided to set the respective sizes of the three layers to 1024, 512, and 128. Note that when we adopted the Bi–LSTM, the actual feature sizes were respectively 2048, 1024, and 256. At the end-part of the LSTM, fully connected layers were attached to predict the position of the mobile robot based on the sequential features processed by the LSTMs.

C. Residual Attention layer

To precisely estimate the position of the tag node, it is important for the network to distinguish the most meaningful context based on the time step T to help contextual understanding of our networks. The equation of the original attention mechanism is as follows,

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

where x denotes the output of the previous neural network layer, $H(x)$ denotes the output of the attention layer to be forwarded to the next neural network layer, and $M(x)$ denotes the attention mask. Based on the multiplication of x by $M(x)$ in an element-wise manner, the network weight was established by the attention layer as crucial information.

Despite the improvement of the performance, the attention layer is associated with potential risks in that it may dilute the features because the attention mask value ranges from zero to one. Thus, we adopted a residual attention layer to alleviate this problem as follows [28],

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

As shown by the blue cuboid shape in Fig. 2, this idea originated from ResNet [31] that contains skip connections in such a way so as to mitigate the aforementioned dilution problem and help the network to be properly trained. Similar to the ResNet, residual attention also contains other branches which are used to calculate how much attention is required. These branches are joined by the original feature vectors x. Each hidden state consists of a residual attention layer so that these attention modules can a) determine which time stamp is more meaningful and b) deliver the output to the next bidirectional LSTM.

D. Training loss

In this subsection, we describe the method for training our network. Let n be the number of anchor nodes of the dataset, L_t, and the measured values of each anchor node, tag node, and ground truth of the 2D positions, Y_t, be represented at each 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 distance between the i^{th} anchor and tag nodes. Note that our neural network does not only accept a set at the time t but accepts sets based on the sequential length of input to our network, T, as follows,

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

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

Consequently, neural networks could be optimized to be able to localize the mobile node by being trained using the training dataset $\mathbb{D}$ 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. Our final goal is to find optimal parameters Θ^* for precise localization by minimizing the L_2 loss term. The L_2 loss term denotes the mean-square-error (MSE) of the Euclidean distance between the normalized ground truth position $\mathfrak{N}(Y_k)$ and the estimated position $\hat{Y_k}$, as follows,

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

IV. Experimental Results and Discussion

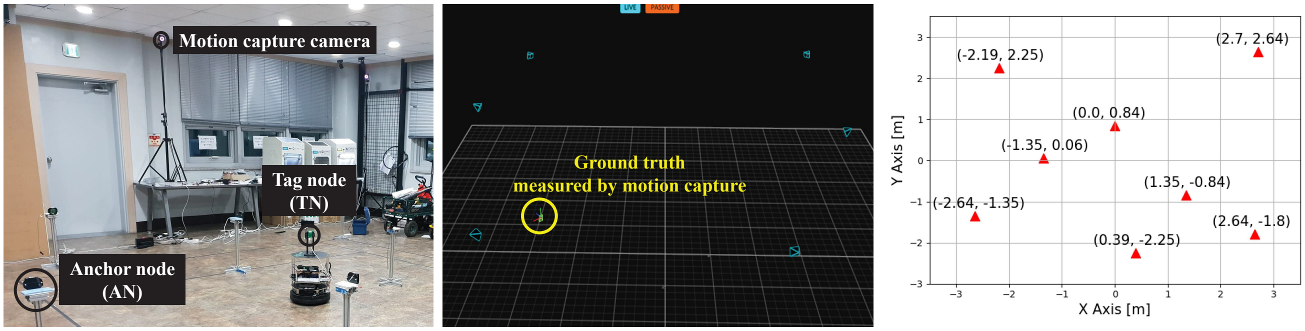
A. Experimental environment

Our experimental system consists of a Ultra-Wide Band (UWB) sensor tag node attached on the mobile robot platform and eight anchor nodes that act as a UWB transceiver, six Optitrack Prime 13 motion capture cameras, and an Nvidia Jetson AGX Xavier, which is a small-form-factor (SFF) computer that has a graphics processing unit (GPU). Fig. 4a shows our experimental environment and indicates how the anchor and tag nodes are attached. In addition, we used the mobile platform iClebo Kobuki from Yujinrobot.

The tag node receives the signal and measures the range between two devices based on the time-of-flight (TOF) and the received signal strength indication (RSSI). Each UWB transceiver, comprising a DW1000 UWB-chip made by Decawave, supports six radiofrequency (RF) bands which range from 3.5 GHz to 6.5 GHz. The measurement accuracies are at the cm level.

B. Acquisition of the Train/Test data

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



(a) (b) (c)

Fig. 4. (a) Experimental environment, (b) tracked pose from Optitrack motion capture, and (c) exact position of anchor nodes.

Note that these two datasets are transmitted at different frequencies: range measurements are obtained at a frequency of $\sim$ 27 Hz, yet the ground truth data are obtained at 120 Hz. Thus, we synchronized these two data based on the range measurements in Hz. Specifically, we set an independent thread so that it a) selected the ground truth data of the closest time instant based on the UWB range measurements, b) concatenated, and c) saved these data.

Moreover, the mobile robot moved within this space controlled using a keyboard manually. All the trajectories are thus different. After collecting complete datasets, we separated the entire dataset to three types: one of these refers to the training dataset, another refers to the validation dataset, and the third is the test dataset. In the case of the test dataset, only range measurements were conducted and used as input to the network.

C. Training the Network

To optimize our network, the Adam optimizer is exploited to train the network during 1200 epochs with a 0.001 learning rate, 0.9 decay rate, and five decay steps. Additionally, we found that the network was influenced by batch size. When the batch size was too large, the unexpected noise attributed to the sensors was reduced by the filter's overgeneralizing response. Conversely, when the batch size was too small, the network tended to overfit to train the specific noise patterns associated with the data. Therefore, we set moderate-sized batch sizes to a size of 6355.

D. Localization Results

1) Performance according to the sequence length: We also considered the sequence length as a type of hyperparameters. We investigated the effectiveness of the optimal size of the sequence length in three cases by changing the number of input data ranges. As fig. 5 shows, the performance is improved when an increased number of sensor values is used as input, but as the sequence becomes longer, the network improves the overall performance in accordance with more useful temporal information.

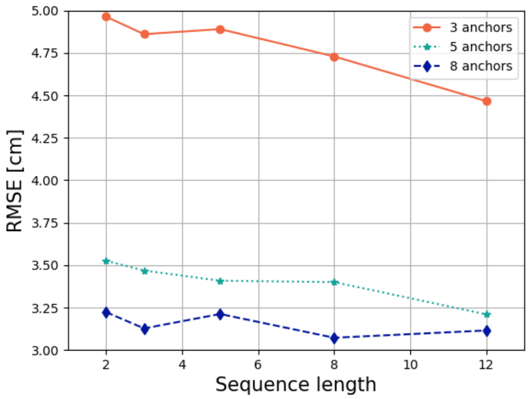
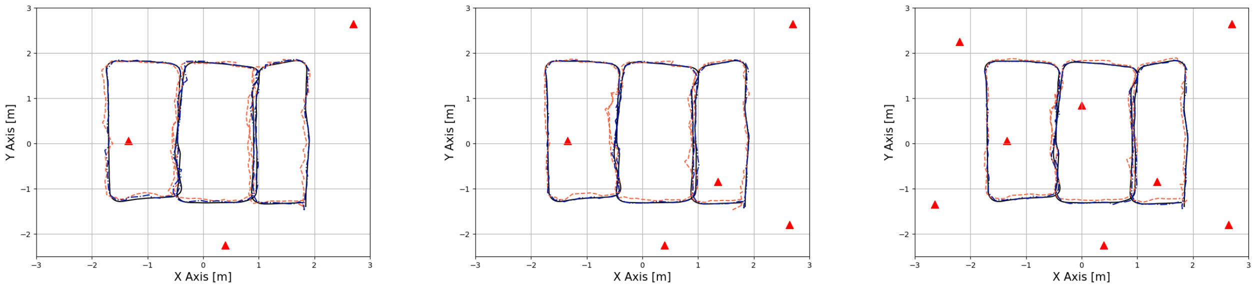


Fig. 5. Plot of root-mean-square error (RMSE) with respect to (w.r.t.) the sequence length for various numbers of anchors.

As a result, we found that there is a trade-off between robustness and the generalization performance according to the sequence length. The networks with longer sequence lengths tend to have smaller error variances and increased abilities to generalize the situation since they can utilize an extended range of temporal information. By doing so, the neural network becomes able to suppress the disturbance caused by noise.

However, note that the performance of the network is inaccurate when it is implemented to sequence lengths equal to 12 compared to sequence lengths equal to eight. This is because of the accumulations of different patterns of sensor noise levels as the number of anchors increases. In other words, the tendency of the domain values may vary owing to the accumulation of different patterns of sensor noise levels, even though a part of the test data path is similar to the path of the train data. This is the reason why range observations included in test data are not correctly or easily mapped following training as sequence lengths increase.

For these reasons, we set the optimal length sequence to be equal to eight. Accordingly, we ensured that we addressed uncertainty issues based on the use of additional temporal information, and that we estimated position precisely.



(a) (b) (c)

Fig. 6. Trajectories corresponding to various numbers of anchors: (a) three anchors, (b) five anchors, and (c) eight anchors. For clarity, only the PF-based (orange) and our RONet results (blue) are presented.

TABLE I: Root Mean Square Error of each algorithm w.r.t. the number of anchors

The results of RMSE [cm]				
Number of anchors	Particle Filter [10]	MLP [13]	Bi-LSTM [14]	Ours
3	8.722	5.485	5.051	**4.466**
5	8.286	4.546	4.418	**3.210**
8	7.650	4.235	4.290	**3.090**

2) Performance comparison of other algorithms: We also compared our network with recently presented learning-based approaches, MLP [13], Bi–LSTM [14], and the conventional PF-based approach [10], [15]. We implemented PF-based localization based on a prior publication of [10]. We tested various numbers of anchors to check if the algorithms worked well on the tested environment when the number of range sensors was small so the algorithms were more affected by the sensor noise.

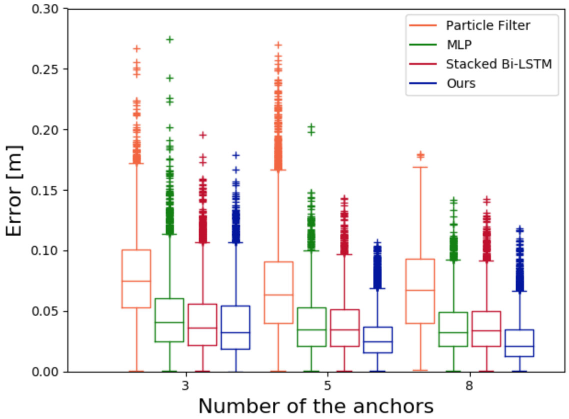


Fig. 7. Box plot results w.r.t. the number of anchors.

As Fig. 7 and Table I show, our proposed RONet exhibits the best performance among the conventional algorithms and previous deep-learning-based approaches in all cases. It yields the smallest RMSE values. Specifically, the RMSE values are 4.466 cm, 3.210 cm, and 3.090 cm, in the cases where three, five, and eight anchors are implemented, respectively. Furthermore, it also shows that our network estimates position with fewer outliers compared to other algorithms.

V. Conclusion

In this study, we proposed a robust three-stacked Bi–LSTM with residual attention, referred to as RONet to solve the problems of localization when using the range measurements. We tested our approach in a realistic manner and showed that it could estimate the position of the mobile robot in real time at an approximate frequency of 32 Hz using range-only measurements. Unlike the conventional probabilistic-based algorithms, the proposed approach did not need any preprocessing module, such as calibration and outlier rejection modules, because it mapped the range observation and position in an end-to-end manner.

In addition, we also analyzed the sequence length as a type of hyperparameters. We concluded that a sequence length that was equal to eight was optimal compared to lengths of two, three, five, and 12. This was attributed to the fact that when a network was built with the optimal sequence length, the ability of the network to deal with uncertainty following the use of additional temporal information was improved, and the position was estimated with increased precision.

Finally, we compared our RONet approach with other conventional probabilistic approaches and previously presented deep-learning-based algorithms. It was shown that RONet yielded the most precise estimates of robot positions. We set three cases, reduced the number of anchors, and confirmed that our RONet was robust, and yielded the smallest RMSE values of 4.466 cm, 3.210 cm, and 3.090cm, when three, five, and eight anchors were implemented, respectively.

In future work, this approach may not be used when the sensors are placed arbitrarily. Besides, thie approach does not consider NLOS situation when training RONet. Therefore, the proposed method of the loss term or architecture of the neural network needs to be revised to obtain precise estimates irrespective of the placement of the anchors or changes in their locations and train data will be acquired considering NLOS case.

References

[1] J. Jung, T. Oh, and H. Myung, "Magnetic field constraints and sequence-based matching for indoor pose graph slam," *Robotics and Autonomous Systems*, vol. 70, pp. 92–105, 2015.

[2] C. Medina, J. Segura, and A. De la Torre, "Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy," *Sensors*, vol. 13, no. 3, pp. 3501–3526, 2013.

[3] R. Li, J. Liu, L. Zhang, and Y. Hang, "Lidar/mems imu integrated navigation (slam) method for a small uav in indoor environments," in *2014 DGON Inertial Sensors and Systems (ISS)*. IEEE, 2014, pp. 1–15.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[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] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

[18] 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.

[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] 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.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.