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 paper, a three-layered Bidirectional Long Short-Term Memory(Bi-LSTM) with residual attention, called RONet, is proposed. We gathered our own dataset and tested RONet in the real-world. Our RONet shows that it could estimate a position of the mobile robot in real-time, almost 32Hz on Nvidia Jetson AGX Xavier, using range-only observations.**

We also analyze sequence length of LSTM as a kind of hyper-parameters. We find that 8 sequence length is optimal among 2,3,5,8, and 12 sequence lengths in the way that building the network with 8 sequence length consider both covering uncertainty with more temporal information and estimating position precisely.

As verified experimentally, our RONet shows better precise performance and robustness against outliers compared to a conventional range-only approach based on Particle Filter and deep learning-based approaches. We set three cases, reducing the number of anchors and check that our RONet is not only robust but also shows the least RMSE, 4.466cm, 3.210cm, 3.090cm in order of 3 anchors, 5 anchors, and 8 anchors are implemented respectively.

I. INTRODUCTION

In recent years, as demand for localization in indoor environments where the signals of Global Positioning Systems(GPS) could become imprecise gradually increases, many researchers have conducted various methods for locating objects, e.g., using magnetic fields, acoustic signals, or laser-based data. Among them, Time of Flight(TOF)-based range beacon sensors are widely utilized by virtue of characteristics of beacon sensors: low-cost, small-size, 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 a object node that we want to localize, called tag node, and landmarks, called anchor nodes. However, range measurements that only represent distances between each landmark and the mobile robot respectively. In other words, a set of one-dimensional data have two problems: one is that range-only observations tend to be non-linear because TOF-based measurement is very vulnerable to noise and has huge uncertainties caused by the multipath fading channel(MPF) problem [5] in the real-world, and the other one is that these range-only observations have *rank deficiency* problem [6]. To be specific, the single value to represent the distance between each landmark and the mobile robot respectively is deficient to describe the exact position or orientation of the landmark so cause multimodal distribution [7].

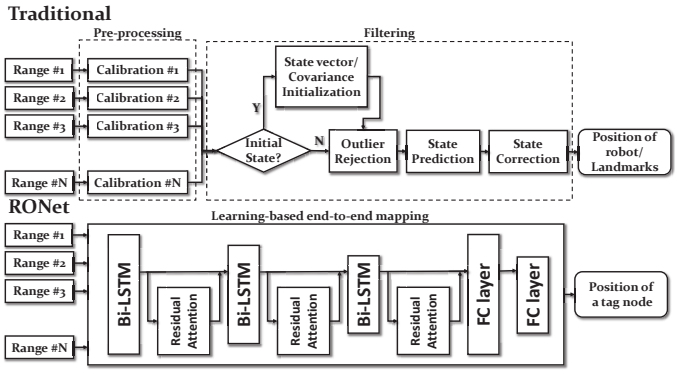


Fig. 1. Comparison between a conventional probabilistic-based range-only frameworks and our learning-based approach.

To alleviate these issues, many studies have been conducted based on probabilistic Bayesian inference frameworks and Monte-Carlo Bayesian filters, 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, the authors just utilized Multilayer Perceptron(MLP), which is beginning architecture of deep learning fields [8]–[10]. In [11], stacked bidirectional Long Short-Term Memory(Bi-LSTM) is implemented to cover the noise of range observation by utilizing the characteristics of it that takes temporal sequential value as input, yet they tested on the simulated environment. Besides, 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, such as a calibration and outlier rejection.

Our contribution is threefold as follows:

*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]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

- 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 a minimal number of anchors.
- We operate RONet on 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. However, unlike other sensors, range measurements are hard to linear approximation because of MPF or Non-line of Sight issues(NLOS). So some authors insist that PF-based approaches could be better than KF-based approaches because PF can cope with complex non-linear model and also cover multimodal distributions [7], [12], [13].

Fig. 1. shows that general steps for conventional probabilistic approaches. First, each range measurements are need to be calibrated respectively. Then, after initialization, the algorithm check whether input value is outlier or not, and then eliminate this unexpectedly large noise. Next, they predict present states, including the mobile robot's pose and anchors' location and finally they correct their prediction using range observation.

B. LSTM-based Sequential Modeling

As deep learning age has come [14], various kinds of deep neural architectures have been proposed for localization task [15]–[17]. Especially, recurrent neural networks (RNNs), originated from the Natural Language Process(NLP) area [18], have been shown to achieve better performance in case of dealing with time variant information.

Besides, 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 information as the time-sequential gap grows [19], LSTM are actively introduced to learn longer-term contextual understandings. Therefore, many authors exploit LSTM for sequential modeling after feature extraction by CNN [20]–[22]

C. Deep Learning for Range Only Localization

Specifically, LSTM are also utilized to model low-dimensional sensor data by itself. In [23], they exploit LSTM for indoor localization with magnetic and light sensors. And in [24], they estimate 2D odometries via stacked Bi-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 MLP. Their approach only map a set of range observations on time t to position so their approaches might have potential to unexpected sensor input. Other paper [11], stacked bi-LSTM is implemented for localization a mobile robot takes sequential range measurements from anchors and the tag, but they only conducted simulation environment, which MPF or unexpected noises do not occur. Therefore, in this paper, we conduct experiments on a real-world to verify the workability to cover all noises of sensors with sequential range information and compare these approaches.

D. 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, taking on a role as a feature selector [25]. On the first time, attention is utilized at the NLP areas for improving translation performance [26]. But 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. In detail, we introduce the stacked Bi-LSTM and residual attention module that we choose for localizing 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 to the network. The cell state consists of 3 gates to preserve the previous information and control the cell state: the forget gate, input gate, and output gate and equations of those are as follows:

$$f_t = \sigma_s\big(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f\big) \tag{1}$$
$$i_t = \sigma_s\big(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i\big) \tag{2}$$
$$\tilde{c}_t = \tanh\big(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c\big) \tag{3}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4}$$
$$o_t = \sigma_s\big(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o\big) \tag{5}$$
$$h_t = o_t \odot \tanh\big(c_t\big) \tag{6}$$

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*. The

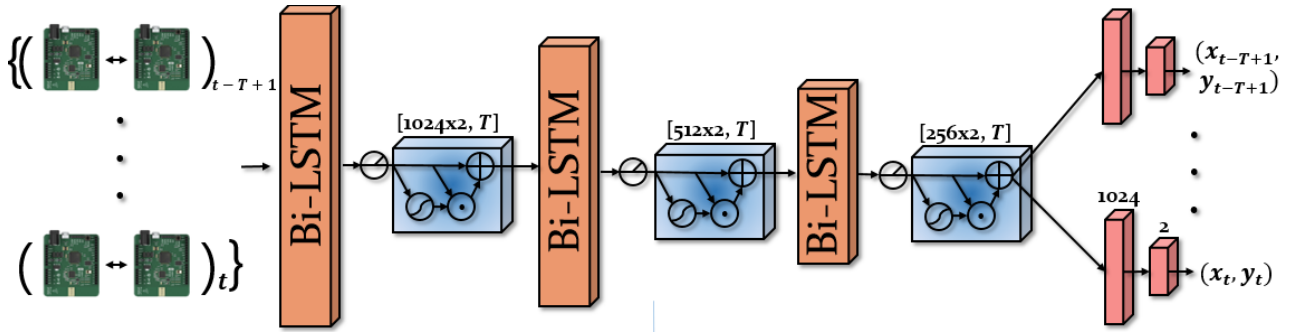


Fig. 2. Our networks consists of three elements: Bi LSTM, the residual attention module(the blue cuboid), and fully-connected layer(FC layer). Features are fed in to Bi-LSTM and Bi-LSTM reduce feature in half, as 2048-1024-512. Finally, extracted features are fed in to FC layer to estimate position corresponding to each time step

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 the 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

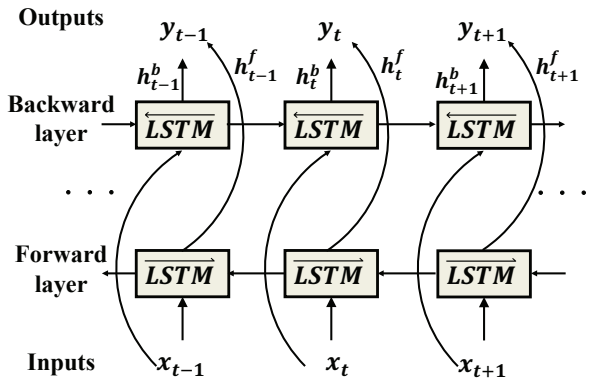


Fig. 3. Architecture of the Bidirectional LSTM(Bi-LSTM). bidirectional LSTM consist of 2 LSTMs: one forward LSTM layer

Like the fact that the deeper the architecture of neural networks, the better their performance [27], [28], many authors have analyzed variations of LSTM architecture and find out that stacking multiple layers of the LSTM improve the performance for many tasks [29]–[31]. Besides, Bidirectional RNNs are introduced [32] 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 the 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 decide to implement the stacked Bi-LSTM architecture to model the system. By virtue of the increased non-linearity caused by a number of stacked layers, the network could model more complex localization taking UWB-ranging observations as input, which includes unexpected noise and MPF problems. Furthermore, we judge that Bi-LSTM would be more 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 come from the sigmoid function and $tanh$ function that compose the part of LSTM. These activation functions cause the *vanishing gradient problem* [33], 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 Rectified Linear Unit(ReLu) function between LSTMs to avoid the vanishing gradient problem [34], instead of stacking more LSTM to increases non-linearity. In addition, experiments shows that reducing the hidden size of the next LSTM layer when the features are fed into the LSTM layer slightly increases performance, 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, actual feature size is 2048-1024-256. The 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

To precisely estimate the position of the tag node, it is important for the network to distinguish which is a more meaningful context on time step *T* to help contextual understanding of our networks. The equation of 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 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 crucial 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 [25]:

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

As blue cuboid shape in the FIGURE 2 shown, this idea is originated from the Residual Net(ResNet) [28] 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:

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

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 attached. 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 Strength 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.

B. Acquisition of the Train/Test data

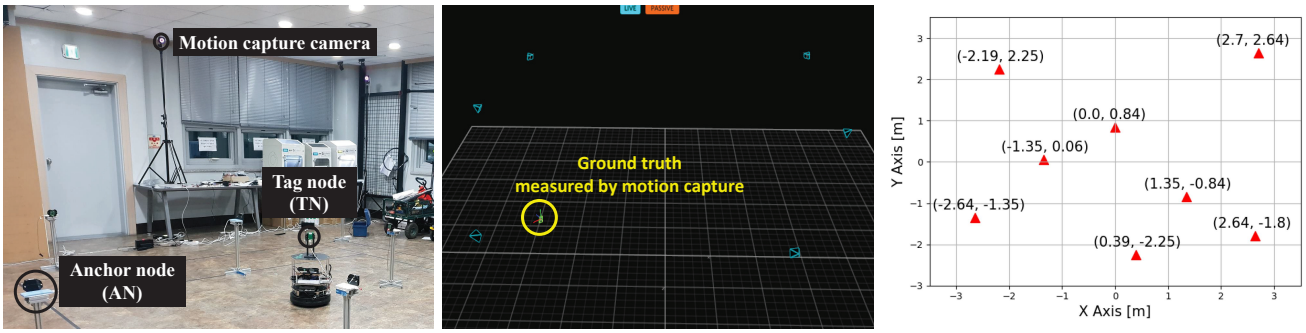
UWB sensor anchors are installed randomly in the region where motion capture camaras are acceptable, as Fig. 4c shown. These anchor nodes transmit the UWB signal to tag node that is attached to the mobile robot and the Optitrack motion capture cameras also transmit the ground truth data to the SSF computer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

Note that these two data 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 space 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 optimize our network, the Adam optimizer is exploited to train the network during 1200 epochs with 0.001 learning rate, 0.9 decay rate, and 5 decay step. And we found that network is influenced by batch size: when the batch size is too large, the performance of filtering unexpected noise of sensors is reduced by its over-generalizing. On the other hand, when the batch size is too small, the network tend to be overfitted to train data's specific noise pattern. Therefore, we set moderated-sized batch size as 6355.



(a) (b) (c)

Fig. 4. (a): entire experimental environments, (b): tracked pose from Optitrack motion capture, and (c): exact position of anchor nodes.

D. Localization Results

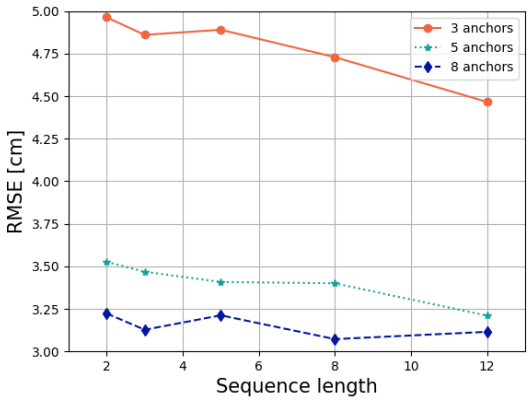


Fig. 5. RMSE graph of RMSE w.r.t. the sequence length by changing the number of anchors

1) The Performance according to the sequence length: We also considered the sequence length as a kind of hyper-parameters. We checked the effectiveness of optimal size of the sequence length in three cases by changing the number of range input data. As fig. 5 shown, there's no surprising that performance is improved when the more number of sensors value is taken as input, but as the sequence becomes longer, the network improves overall performance through more fruitful temporal information.

As a result, we found that there is a trade-off between robustness and generalization performance according to the sequence length. The network with longer sequence length tends to have less error variance and more an ability to generalize the situation since they could utilize more extended temporal information. By doing so, the neural network gets the ability to suppress the disturbance caused by noises.

However, note that the performance when the network is implemented to 12 sequence length is rather a little bit inaccurate than that of 8 sequence length. This is because of accumulations of different patterns of sensor noises as the number of anchors increases. In other words, the tendency of domain values may become different due to accumulations of different patterns of sensor noises even though a part of the test data path is a similar to the path in the train data. That's the reason why as sequence length become longer, range observations included in test data are hard to be mapped correctly based on the training.

For these reasons, we set optimal sequence length as 8 with considering these two aspects, abilities that both cover uncertainty with more temporal information and estimate position precisely.

2) The Performance comparison of Other Algorithms: We also compared our network with recently presented learning-based approaches, MLP [10] and Bi-LSTM [11], and the conventional PF-based approach. We implemented PF-based localization by referring [7]. We tested on various number of anchors in such a way as to check if the algorithms work well on the environment that the number of range sensors is small so the algorithms are more affected by the sensor noise.

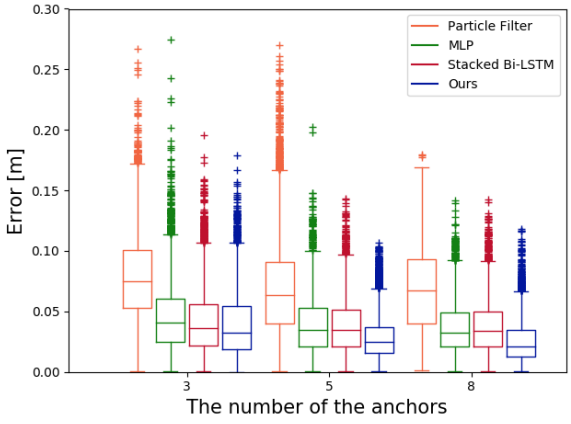
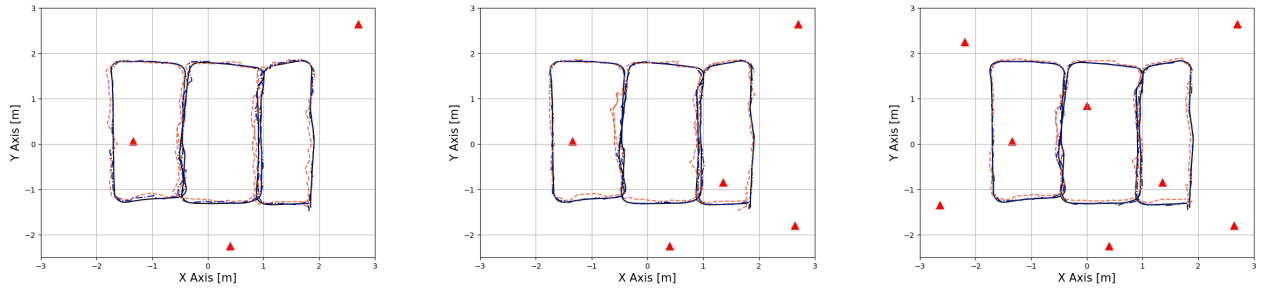


Fig. 6. Box plot results with respect to the number of anchors.

As Fig. 6 and Table. I are shown, our RONet show the best performance among both the conventional algorithm and previous deep learning-based approaches in all cases. It shows the least Root Mean Square Error(RMSE), which RONet's RMSE is 4.466cm, 3.210cm, 3.090cm in order of 3 anchors, 5 anchors, and 8 anchors are implemented



(a) (b) (c)

Fig. 7. Trajectories results with respect to the various number of anchors: (a): a trajectory with 3 anchors, (b): a trajectory with 5 anchors, (c): a trajectory with 8 anchors. For clarity, we just drawed PF-based results(organge) and our RONet results(blue).

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

The results of RMSE [cm]				
# of anchors	Particle Filter [7]	MLP [10]	Bi-LSTM [11]	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**

respectively. Furthermore, it also shows that our network estimate position with smaller outliers than other algorithms.

V. Conclusion

In this paper, we propose a robust 3-stacked BI-LSTM with residual attention, called RONet. We tested our approaches in real-world and it shows that it could estimate a position of the mobile robot in real-time, almost 32Hz, using range-only measurement. Unlike conventional probabilistic-based algorithms, it does not need any preprocessing module, such as a calibration and outlier rejection because it maps between range observation and position by end-to-end.

In addition, We also analyzed the sequence length as a kind of hyper-parameters. We conclude that 8 sequence length is optimal among 2,3,5,8, and 12 sequence lengths in the way that building the network with 8 sequence length compromise both abilities that cover uncertainty with more temporal information and estimate position precisely.

Finally, we compare other conventional probabilistic approach and previously presented deep learning-based algorithms with our RONet and RONet exhibits the most precise estimates of robot positions. We set three cases, reducing the number of anchors and check that our RONet is not only robust but also shows the least RMSE, 4.466cm, 3.210cm, 3.090cm in order of 3 anchors, 5 anchors, and 8 anchors are implemented respectively.

As a future work, because we conducted on just localization, this approach may not be operated when locations of sensors are arbitrary placed. Therefore, the proposed method, e.g., loss term or architecture of the neural network 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.

[13] 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.

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

[15] 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.

[16] 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.

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

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

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.