Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

WSNNet: Stacked Bidirectional LSTM with Residual Attention for Indoor Localization of Wireless Sensor Network

HYUNGTAE LIM[1], (Student Member, IEEE), CHANGGYU PARK[1], (Student Member, IEEE), AND HYUN MYUNG.[1], (Senior Member, IEEE)

[1]Urban Robotics Laboratory, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea.

Corresponding author: Hyun Myung (hmyung@kaist.ac.kr).

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'.

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

INDEX TERMS Enter key words or phrases in alphabetical order, separated by commas. For a list of suggested keywords, send a blank e-mail to keywords@ieee.org or visit http://www.ieee.org/organizations/pubs/ani_prod/keywrd98.txt

I. INTRODUCTION

SIMULTANEOUS Localization and Mapping(SLAM) is widely used in autonomous vehicles, drones, intelligence field robots, and mobile phone applications. Thus, according to the smart city development plan, several technologies are required in such a way that the demand and the necessity of SLAM increase together. Various kinds of sensors are utilized to SLAM, such as GPS, LiDAR, ultrasonic-based sensor, camera and distance sensor. Especially, trilateration algorithm has been widely incorporated into robotics fields, especially utilized in the indoor environment to estimate the position of an object by distance measurements obtained from range sensors such as UWB, ultrasonic, laser-based beacon sensors [1]–[3] 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 [4]–[7].

In the meantime, as deep learning age has come [8], various kinds of deep neural architectures have been proposed for many tasks related to robotics field, such as detection [9]–[11], navigation [12], [13], pose estimation [14], and so on. Especially, recurrent neural networks (RNNs), originated from Natural Language Process(NLP) area [15], 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 [14], [16]–[19].

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

Various kinds of sensors have been utilized to localize a object using range measurement sensors, such as GPS, ultrasonic-based sensors. ultra-wideband(UWB) sensors. However, almost distance measured by range measurement sensors are based on Time of Flight(TOF), Time of Arriaval(TOA) [20], or Time of Differential Arrival(TDOA) in such a way as to consist of the 1-D data composed by the distance between landmarks and robot. This is the main issue dealing with range measurments, called *rank-dificiency* problems. Besides, only mangitudes could represent the range measurement, deflection, reflection, and refraction and ic Because range measurements consist of

In contrast to other SLAM, RO SLAM suffer from "rank deficiency problem", which means range measurement is 1D data so it is too deficient to describe position or orientation as you guys knows, it only has magnitude. As this figure shown, in 3D, posiibility of location of sensor is distributed over sphere / since range measurement doesn't contain direction information! To solve this problem, various type of RO SLAM has been studied. RO SLAM is generally divided into two approaches; PF RO SLAM and KF based RO SLAM

. 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

The rest of the paper is organized as follows. Section 2 describes relevant localization methods. Section 3 introduces principals of neural networks. The experiments by which these methods will be compared are given in Section 4. The results will be discussed in Section 5, and concluding comments will be made in Section 6

fixed or calculated during initialization stage [7]. For range-based methods, the distance information can be obtained by analyzing time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), or received signal strength indicators (RSSI) [8]. TOA algorithm calculates the distance on the basis of known transmission time and signal propagation speed. It requires highresolution clocks to be installed at sensor nodes. In case of AOA algorithm, the sensor node needs several narrow beam receivers or an antenna array to determine the direction of the received signal. TDOA uses two transmission signals of different propagation speeds. Therefore, it requires two different transmitters and receivers on each node. The above range-based localization techniques have little practical use in WSNs due to the necessity of additional hardware, which increases cost, size, and energy consumption of sensor nodes. RSSI algorithms estimate the node-to-node distances by using a signal propagation model. However, for real world

ference nodes. Currently, there is a considerable research interest in developing fingerprint localization methods based on artificial neural networks (ANNs) [10]. An important advantage of this approach is that the ANN enables accurate recognition of node location in case of noisy RSSI measurements. When using ANNs, the detailed information about indoor environment and locations of the reference nodes is not necessary. ANN interpolates the data collected in the fingerprint database to approximate a mapping between the multidimensional fingerprints space and the coordinates of nodes. In training phase, the collected RSSI vectors are used to tune weights of connections between neurons in the ANN. Although training can be time-consuming, the localization process is much faster than analytical estimation of the node location. In this paper a method is proposed that improves localization accuracy of the ANN-based fingerprinting. According to the introduced method, the entire localization area is divided into regions by clustering the fingerprint database. A separate ANN is trained for each region by using only those fingerprints that belong to this region (cluster). During clustering, a prototype RSSI vector is determined for each region. When localization process starts, those prototypes are selected that are most similar to the vector of current RSSI measurements. The ANNs that correspond to the selected prototypes are used to estimate the node coordinates. Final estimation of the location is obtained by fusion of the coordinates delivered by ANNs. Further improvement of the localization accuracy as well as speedup of learning process was achieved by employing fully connected neural networks

We propose a novel range-free localization algorithm for wireless sensor networks that is robust against the anisotropic signal attenuation

++++++

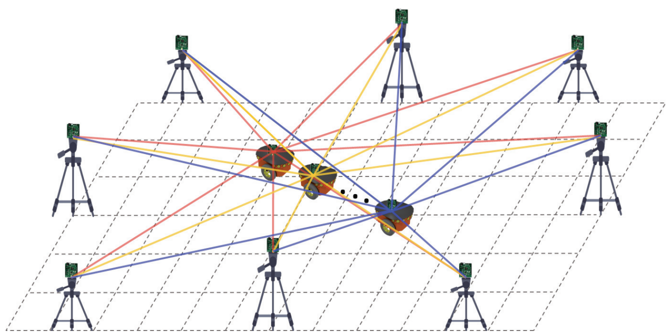


FIGURE 1: Figures from experiment (a)The anchor and tag nodes (b)Four examples of the trajectory (c) the process that makes dataset

II. RELATED WORKS

In the past few years, some researchers have conducted the studies for wireless sensor networks to improve the performance of their algorithms by reducing computational complexity or localizing a mobile node more precisely. Also, many machine learning techniques have been introduced:

one authors utilized support vector machine(SVM) for localizaiton, [21]–[24], other author developed method support vector regression(SVR) for localiztion [25], [26]. In [21], 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 [23], [24], [27]. Samadian *et al.* [28] 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 [25], [26]

Especially, to localize nodes of the range measurement sensors on the indoor space while covering range measurements' uncertainties using neural networks, several fascinating works have been studied. 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 [29]. However, they did not provide numerical analysis, so Shareef *et al.* did [30] and conducted their experiment in the real-world. They conlcuded Multi-Layer Perceptron(MLP) may be the best option among the suggested Kalman filter models and RNN.

Similarly, many researchers also have achived considerable improvement to localize position of mobile node by exploiting MLP [31]–[35] in WSN fields. Rahman *et al.* [31] 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 [32], 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 nodes in given arbitrarily spread sensor nodes test data set. Abdelhadi *et al.* [33] presented two artificial intelligence techniques: Sugeno-type fuzzy system and neural networks system. In addition, the authors conducted experiment on three-dimensional(3D) space in such a way as verified the feasibility of localization by utilizing nueral networks in 3D space. Kumar *et al.* [34] 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, [35] 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.

The contributions of this paper can be summarized as follow:

The paper is divided into five sections. Besides this introductory section, the section II develops the VLC system model deployed in the AoA and RSS estimators which are described in subsections II-C and II-D respectively. In Section III, the 3-D hybrid estimator obtained is applied to the SO-OFDM multiplexing scheme with DCO-OFDM. In Section IV numerical simulation results are considered aiming at corroborating the quality of the 3-D location estimations for the proposed scheme. Finally, in Section V the conclusions are offered.

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 [27], [30]–[32], [35]. Because the simulation situation is alomst ideal in the point that the multipath caused by reflection and refraction does not occur. In other words, the data generated in the simulation environment has less noise than that of real-world necessarily. These factors make the sensor values more highly unstable in turn have a bad influence on accuracy of localization directly. In case of virtual environment, even though the authors artificially design the non line of sight(NLOS) situation or mix the noise into the measured value and shows quite accurate localization results, 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.

Finally, It must be noted that the RSSI values obtained are highly unstable and turn to vary under environmental noise and mobility of sensor nodes. A neural network offers the advantage that prior knowledge of the environment and noise distribution is not necessary. Moreover, higher accuracies are achieved by neural networks compared to other techniques such as the Kalman filter [3]. The trade-off between the accuracy and memory requirements of the MLP neural network is the best when compared with other types of neural networks, thus it has been chosen to be used in this research.

++++ Unlike range-based algorithms, range-free methods only utilize the connectivity information for the positioning purpose. These approaches do not need non-anchors to have specific hardware for measuring distances. The researchers consider these techniques as a simple and cost-effective solution than range-based algorithms for the localization problem. The non-anchor nodes obtain the connectivity information of hop count distances from anchors and estimate their positions by this information. In recent years some research works exploit machine learning methods such as neural networks to improve the performance of sensor networks on given tasks, for instance forest fire detection [10], air quality monitoring [10], intelligent lighting control in the smart building [10], localization [11] and providing full coverage of the area using dynamic deployment [12]. The machine learning methods can be applied to both range-based and range-free localization algorithms. In range-free algorithms, the connectivity information is utilized for training of neural networks. After that, obtained neural network model is

Localization method	Dimension	Type of input data	Train data	Test data mobility	Implementation environment
MLP [30]	2D	RSSI	Grid	Dynamic nodes ✓	Real-world ✓
MLP [31]	2D	RSS	Grid	Static nodes	Simulation
MLPNN [27]	2D	Hop count	Grid	Static nodes	Simulation
MLBPN [32]	2D	TDOA	Grid	Static nodes	Simulation
MLP [33]	3D ✓	Distance	Spread	Static nodes	Real-world ✓
Clustering-based FCNNs [36]	2D	RSSI	Spread	Static nodes	Real-world ✓
MLP [34]	2D	RSSI	Grid	Static nodes	Real-world ✓
LPSONN [35]	2D	Hop cound	PSO ✓	Static nodes	Simulation
Ours	3D ✓	TOF	PSO on the mobile robot ✓	Dynamic nodes ✓	Real-world ✓

sent to the network for localization of non-anchor nodes. In this paper, we present a range-free localization method that uses neural networks for positioning of non-anchor nodes. The method uses hop count distances between anchor nodes for the training of the neural network. Particle swarm optimization (PSO) algorithm is used to optimize the count of neurons in the hidden layers of the neural network. An objective function is defined to optimize the neural network based on the localization accuracy and storage space that is needed for storing the weights of the neurons in the hidden layers. The contribution of this paper is that we use PSO to optimize the neural network based on the storage cost and localization accuracy, simultaneously. Furthermore, in this objective function, we consider both of average error of estimated positions of all beacons and the maximum error of estimated positions among beacons. The optimized neural network model is sent to the network and is used for localizing the non-anchor nodes. +++

Note that their In case of [30]. They let MLP learn the relationships between range measurement and position of mobile node, yet MLP could not learn sequential modeling. MLP just learn the relationship like generating finger print map.

In traditionally connected ANNs, such as the MLP or RBF, neurons are organized in layers and connections are introduced from one layer to the next layer. The FCNNs have additional connections across layers In [37] it was demonstrated that when comparing FCNNs with traditionally connected ANNs the latter ones require about twice as many neurons to perform a similar task. With connections across layers in FCNNs,

RSS is the actual signal strength received at the receiver and the unit of measurement can be in dBm, dB, milli Watt, Watt. So RSS will always have a unit.

In this multihop connectivity-based localization algorithm, the distance between the two nodes is calculated in terms of the shortest path between them, which is expressed in hop-counts. The beacon nodes send their locations to ordinary sensors by sending messages that are propagated hop by hop

Incidentally, There are many variations of LSTM architecture. As studies of deep learning are getting popular, various modified architectures of LSTM have been proposed for many tasks in a wide area of science and engineering. Because LSTM is powerful when dealing with sequential data and infering output by using previous inputs, LSTM is utilized to estimate pose by being attached to the end part of deep learning architecture [17]–[19] as a stacked form of LSTM. In addition, LSTM takes many various data as input; LSTM is exploited for sequential modeling using LiDAR scan data [16], images [14], [17], IMU [38], a fusion of IMU and images [39]. Since existing RO-SLAM performs localization using low-dimensional data, it is difficult to estimate even if the value deviates slightly from the model. In addition, LSTM has the advantage of being able to solve long-term dependence problem of traditional RNN, and it is possible to model it by non-linear mapping through analyzing the current situation without modeling data characteristics separately. Therefore, we propose RO SLAM technology using deep learning based SLAM which applies the advantages of LSTM and deep learning to solve the disadvantages of RO SLAM.

First, In case of particle filter based RO SLAM, it is more robust than kalman filter based approach, As figure illustrated, you can observe how the Kalman filter based approach performs poorly / when the uncertainty in the beacon position becomes excessively large. And In PF filter based RO SLAM, they exploit Rao-Blackwellization. Rao-blackwellization is a mathematical method. By dividing one hidden states into two variable, it proves that variance can be reduced.

So they utilize rao-blackwellized particle filter, called RBPF, so many authors separate all states / into states of landmarks and state of robot. But in many cases, they just consider almost annular ambiguity or projected spherical ambiguity, not spherical ambiguity!.

On the other hand, kalman filter based approach is steadily studied, and they make efforts to reduce the number of hidden state variables. In case of 3D RO SLAM, there are two angles to be estimated, one is the azimuth angle that indicates angle on horizontal plane, and the other is elevation angle which indicates amount of elevation literally. On state of the art paper about 3D RO SLAM based on EFK, they dramatically reduce the number of hidden states by expressing the hypothesis as multiplication of probability about azimuth angle and elevation angle as this figure shown.

Besides, not only for the indoor environment, also on the underwater environment, Olson *et al.* suggest a method for localize a autonomous underwater vehicle(AUV) using extended Kalman Filter(EKF) [40].

Especially, deep learning-based approaches are also implemented to reduce noise of the san

First, it's very noisy, so it can occur errors easily. Second, the measurement is very ambiguous because each measurement is defined as the probability density of the sensor's potential position. The last problem is that the landmark location estimations may converge to multi-modal densities. Especially, trilateration algorithm has been widely incorporated into robotics fields, especially utilized in the indoor environment to estimate the position of an object by distance measurements obtained from range sensors such as UWB, ultrasonic, laser-based beacon sensors [1]–[3] 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 [4]–[7].

In robotics fields, Blanco SLAM is a technique for building the map information while localizing the position of the robot while moving. Localization of the SLAM predicts the current position of the robot using the landmark measured by the sensor, and mapping locates the terrain object based on the pose of the robot. Research on this technology has been actively carried out, and researches and techniques have been summarized. In 2006, the ad hoc sensor network consisting of range detection beacon was applied to SLAM technology for various ranges. This technology integrates node-to-node measurements to reduce drift and expedite node-map convergence [41] In 2008, the technique to consistently combine the observation information considering the uncertainty was studied through comparing the experimental data with the actual robot and simulation using Ultra Wide-Band (UWB) devices and Rao-Balckwellized Particle Filter (RBPF) [4]. In 2012, a simple and efficient algorithm for position recognition with high accuracy and low computational complexity was researched with ultrasonic sensors [42]. In recent years, 3-dimensional-based SLAM has also been under active research and development. In 2013, a localization mapping approach of a wireless sensor network (WSN) node was studied through a centralized EKF-SLAM-based optimization research [6]. In addition, in 2014, a method of minimizing noise and localizing Unmanned Aerial Vehicle (UAV) by using range-only measurement while simultaneously mapping the position of the wireless range sensors were proposed [43]. SLAM based on range measurement has been continuously researched and developed then applied to various fields. In this paper, we propose a novel technology that applying deep-learning to range-only SLAM that derives accurate range and robot position measurement through in-depth learning.

A. DEEP LEARNING FOR LOCALIZATION

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

One of the popular SLAM techniques with deep learning is CNN-SLAM [44] which takes Convolutional Neural Networks (CNNs) to precisely predict the depth from a single image without any scene-based assumptions or geometric constraints, allowing them to recover the absolute scale of reconstruction. Another approach using deep learning for localization is Deep VO [39] In this method, Recurrent Convolutional Neural Networks (RCNNs) is utilized. Specifically, feature representation is learned by Convolutional Neural Networks and Sequential information and motion dynamics are obtained by deep Recurrent Neural Networks without using any module in the classic VO pipeline.

III. WSN NET

In this chapter, we explain how our proposed residual attention-based stacked Bi-LSTM is implemented, as illustrated in Fig. 2. In detail, we introduce the neural networks concepts that we choose for localizaing the tag node and 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 [45]. 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:

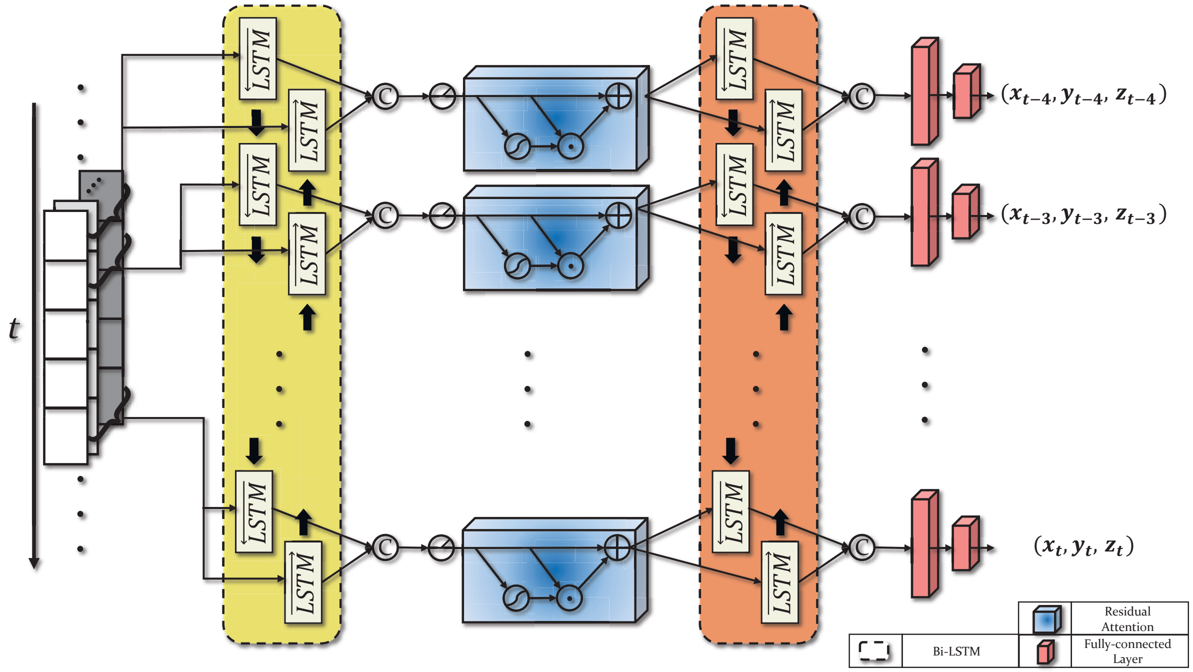


FIGURE 2: Figures from experiment (a)The anchor and tag nodes (b)Four examples of the trajectory (c) the process that makes dataset

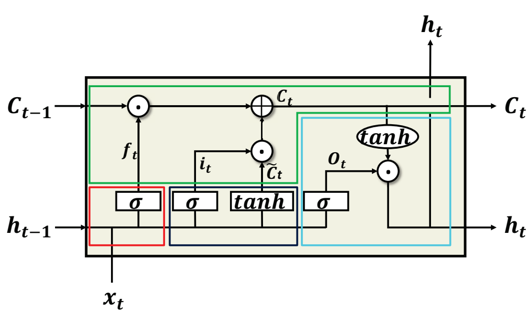


FIGURE 3: It introduce 3 gates, forget gate, input gate, and output gate. And output gate is divided into cell state layer(Green) and output gate layer(cyan)

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

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

The Forget gate layer, f_t, decides how much information to forget. 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

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 [46] 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 sequential systems [47]–[49].

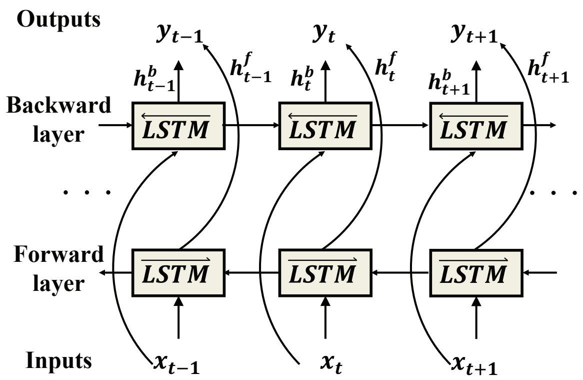


FIGURE 4: Bidirectional LSTM

As FIGURE. **??** shown, bidirectional 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:

$$h^f_t = \overrightarrow{LSTM}\left(x_t, h^f_{t-1}\right) \tag{7}$$
$$h^b_t = \overleftarrow{LSTM}\left(x_t, h^b_{t+1}\right) \tag{8}$$
$$y_t = \sigma_R\left(W_{h^f y} \cdot h^f_t + W_{h^b y} \cdot h^b_t + b_y\right) \tag{9}$$

where σ_R denotes activation function called ReLu. Note that in our case, we concatenate h^f_t and h^b_t 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 [6]. 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 [50], [51] 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 [49], [52], [53]. 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* [54], 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 [55], instead of stacking LSTM.

D. RESIDUAL ATTENTION LAYER

A Attention layer is powerful module nowadays and mostly improves performance of neural network. Originally, neural networks treats information equally. But, using attention layer, neural networks can be ATTENDED what it should be examined closely. At the first time, attention is utilized at natural language processing area for improving translation performance [56]. But nowadays, attention layer is employed in many areas to improve the performance of the networks. For example, Jaderbeg *et al.* [57] introduced the attention layer to let the neural networks attend to spatial information. In addition, attention is even utilized to pose estimation and optimization [58], detection [59], and video captioning [60]

To precisely estimate the position of the tag node, it is important for the network to distinguish which is more meaningful context on time step *T* to help contextual understanding of our networks. So, we add the attention layer between the LSTM and the attention layer take on a role of a feature selector [61]. The equation of the attention machanism is as follows:

$$H(x) = M(x) \odot x \tag{10}$$

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

Despite of the improvement of the performance, the attention layer has potential risks that it may dilutes the features because attention mask ranges over 0 to 1. To alleviate this problem, residual attention layer is introduced in our network as follows [61]:

$$H(x) = (1 + M(x)) \odot x \tag{11}$$

As blue cuboid shape in the FIGURE 2 shown, this idea is originated from the Residual Net(ResNet) [51] 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 second bidirectional LSTM.

E. TRAINING LOSS

In this section, we describe the method for training our network. Generally, let n be the number of mobile nodes and m be the number of the anchor nodes, data set are represented as follows:

$$\{(l_{11}, l_{12}, ..., l_{1m}, P_1), ..., (l_{n1}, l_{n2}, ..., l_{nm}, P_n)\} \tag{12}$$

where l_{ij} denotes the the distance between i^{th} mobile node and j^{th} anchor node, P_i denotes the position of mobile node, which consist of 2D (x and y), or 3D (x,y, and z). In other words, data consist of set of distance data corresponding to the position of mobile nodes. Conseqeuntly, neural network could be optimized to be able to localize the mobile node when take distance set as input.

However, in previous studies!!!!!!!!!!!!!!!!, it has a possibility of overfitting because the authors generate the grid-map to estimate the moible node 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 though the position of mobile node to be tested is quite far from the grid. As a reusult, neural netorks may have a tendancy only to infer similar position that are included in train data when taking set of distance $(l_{i1}, l_{i2}, ..., l_{im})$ as input. Therefore their grid map train impedes the optimization of nueral netorks to cover all over the region.

To return, our neural network does not only take a set of distance data but takes sets of distacne data on the time step T where T indicates sequential length of input to our network. And in our case, one mobile node is only placed on the robot. Therefore, data are formulated as follows:

$$\mathbb{L} = (L_t, Y_t) \tag{13}$$

where $L_t = \{(l_1, l_2..., l_m)_t\}$ denotes input range measurement between a tag node end each anchor node at the time t. We omit the part of subscript that indicates i^{th}mobile node because we have only one mobile node. Y_t denotes the ground truth of the robot's 3D position, which is denoted as $Y_t = \{x_t, y_t, z_t\}$.

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

$$P(Y_t|L_{t-T+1}, L_{t-T+2}, ..., L_t) = \\ p((x_t, y_t, z_t)|(l_1, ..., l_m)_{t-T+1}, (l_1, ..., l_m)_{t-T+2}..., (l_1, ..., l_m)_t) \tag{14}$$

Note that other studies only consider the input on time t, yet our approach consider temporal information about the range measurement data. 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 ground truth position Y_k and estimated position $\hat{Y}_k$ as follows:

$$\Theta^* = \underset{\Theta}{\text{argmin}} \frac{1}{N} \sum_{k=1}^{N} \parallel Y_k - \hat{Y}_k \parallel^2 \tag{15}$$

F. WHY ON THREE-DIMENSIONAL?

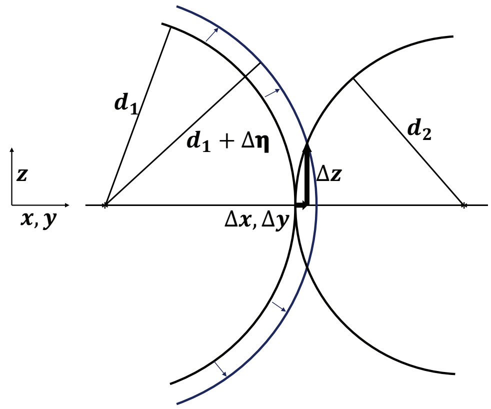


FIGURE 5: Figures from experiment (a)The anchor and tag nodes (b)Four examples of the trajectory (c) the process that makes dataset

One may ask a question that why we infer the robot's position on the three dimensional space even test are being conducted on the mobile robot. It is true that position of the z varys very little. However, we found that localizing the mobile node on the three dimensional space using range measurement data is very week to noise. In more detail, let assume that 4 anchor node are placed on the ground and form square with similar height. Let x_i, y_i, z_i, and d_i be the position of i^th anchor node and range measurement. Then equations on the 3-D space are as follows:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = {d_1}^2 \tag{16}$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = {d_2}^2 \tag{17}$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = {d_3}^2 \tag{18}$$

$$(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2 = {d_4}^2 \tag{19}$$

where x, y, and z is the unkown position of the mobile node. And we can rewrite these equation by substracting (19) from (16), (17), and (18)

$$A_{3D}X_{3D} = b_{3D} \tag{20}$$

where X_{3D} indicates $[x, y, z]^T$ and A_{3D} and b_{3D} are as follows:

$$A_{3D} = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & 2(z_2 - z_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 2(z_3 - z_1) \\ 2(x_4 - x_1) & 2(y_4 - y_1) & 2(z_4 - z_1) \end{bmatrix} \tag{21}$$

$$b_{3D} = \begin{bmatrix} ({d_1}^2 - {d_2}^2) - ({x_1}^2 - {x_2}^2) - ({y_1}^2 - {y_2}^2) - ({z_1}^2 - {z_2}^2) \\ ({d_1}^2 - {d_3}^2) - ({x_1}^2 - {x_3}^2) - ({y_1}^2 - {y_3}^2) - ({z_1}^2 - {z_3}^2) \\ ({d_1}^2 - {d_4}^2) - ({x_1}^2 - {x_4}^2) - ({y_1}^2 - {y_4}^2) - ({z_1}^2 - {z_4}^2) \end{bmatrix} \tag{22}$$

Unlike the case of 2D, A_{3D} consists of z components on the third column. Anchor nodes are placed in a less scattered on the z direction than x and y axis. In other words, it is hard to put the anchor nodes with exactly same height in real-world, which means $z_1 \approx z_2 \approx z_3 \approx z_4$. Consequently, values on the third column of A_{3D} converge to zero. Let assume that A_{3D} be full rank in such a way as to $\exists A_{3D}^{-1}$, then values of third row of A_{3D}^{-1} relatively have considerable nubmers. As a result, this make z value very unstable in such a way as to cause huge error with respect to z adirection.

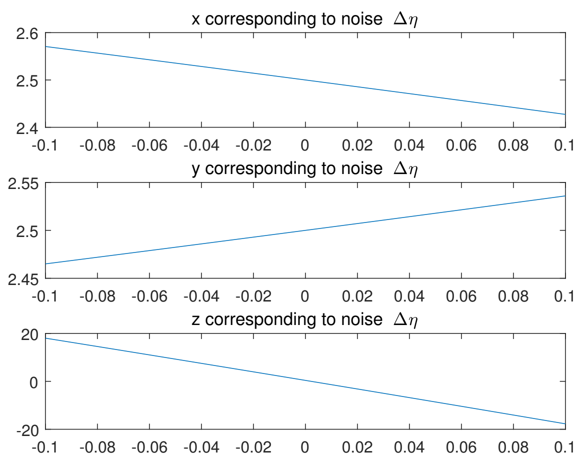


FIGURE 6: (-0.01,0.001, 0.24), (5.02, 0.01, 0.2), (5, 5.01, 0.21), (0.01, 4.999, 0.23) true: (2.5, 2.5,0.4)

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 small-form-factor(SFF) 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. 1 shows anchor and tag nodes.

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 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 small form-factor computer is a gigabyte Ultra compact PC. Deep learning framework used for our network is pytorch 0.4.0 on python 3.6. The network inferences on the same setting.

The UWB tag is attached to mobile robot that has a small compact computer. The UWB anchors are attached to stands that have two different heights. The anchors are positioned randomly in the square space. As you can see in Fig. 1(b), a mobile robot manually goes on various random trajectories by experimenters.

During the robot is going on, the data is saved in the computer. The distance data used for input data is measured by the UWB sensors. The global position data used for ground truth is measured by the motion capture system. These two kinds of data are paired in a dataset. 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. 1(c) 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.

B. DATA SYNCRONIZATION FOR TRAIN/TEST DATA

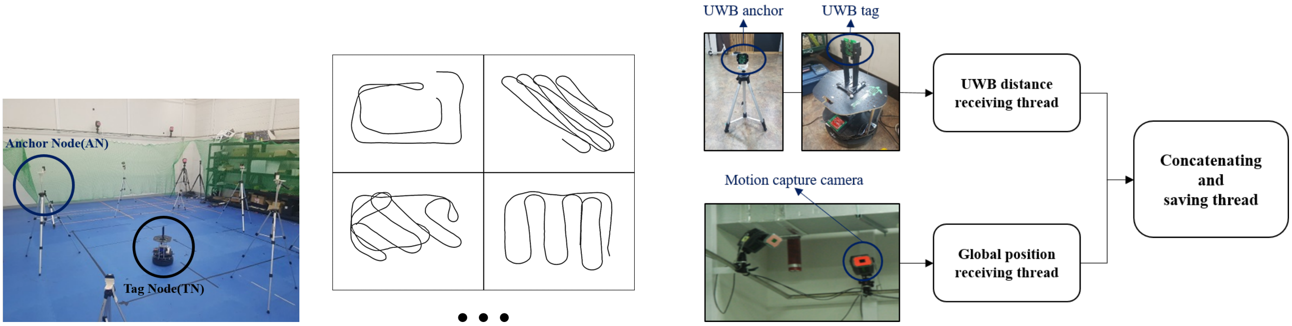
C. TRAINING THE NETWORKS

V. RESULTS

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

As illustrated in Experiment session, train data are our own data gathered by UWB sensors and motion capture camera, so neural networks take range-only measurements as input and output robot's position. Ground truth data is robot's position measured by eagle eye motion capturer, whose error is in mm units. The results of trajectory prediction are shown in Fig. **??** and Root-Mean-Squared Error (RMSE) are shown in Table 1. Note that out experiment is conducted on mobile robot, so we can pre-estimates that z position of robot is almost similar while robot is running.

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



(a) (b) (c)

FIGURE 7: Figures from experiment (a)The anchor and tag nodes (b)Four examples of the trajectory (c) the process that makes dataset

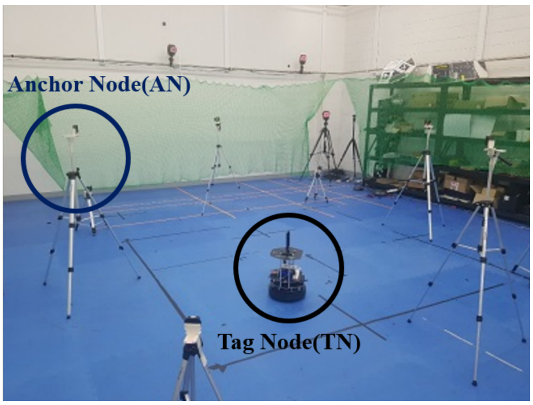


FIGURE 8: (a)The anchor and tag nodes

RMSE of our networks in test1 is 3.928cm and 4.119cm in test2.

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

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

TABLE 1: Root mean squared error of each case

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.

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.

.

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments. Avoid expressions such as "One of us (S.B.A.) would like to thank . . . ." Instead, write "F. A. Author thanks . . . ." In

most cases, sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

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

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

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

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

[14] 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.

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern recognition letters, vol. 31, no. 8, pp. 651–666, 2010.

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

[39] 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.

[40] 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.

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

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

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

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

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

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

[47] 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.

[48] 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.

[49] 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.

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

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

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

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

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

[59] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards high performance video object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7210–7218.

[60] J. Xu, T. Yao, Y. Zhang, and T. Mei, "Learning multimodal attention lstm networks for video captioning," in Proceedings of the 2017 ACM on Multimedia Conference. ACM, 2017, pp. 537–545.

[61] 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.

FIRST A. AUTHOR (M'76–SM'81–F'87) and all authors may include biographies. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year the degree was earned. The author's major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (publisher name, year) similar to a reference. Current and previous research interests end the paragraph. The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, it should be of good quality, and professional-looking. Following are two examples of an author's biography.

SECOND B. AUTHOR was born in Greenwich Village, New York, NY, USA in 1977. He received the B.S. and M.S. degrees in aerospace engineering from the University of Virginia, Charlottesville, in 2001 and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, PA, in 2008.

From 2001 to 2004, he was a Research Assistant with the Princeton Plasma Physics Laboratory. Since 2009, he has been an Assistant Professor with the Mechanical Engineering Department, Texas A&M University, College Station. He is the author of three books, more than 150 articles, and more than 70 inventions. His research interests include high-pressure and high-density nonthermal plasma discharge processes and applications, microscale plasma discharges, discharges in liquids, spectroscopic diagnostics, plasma propulsion, and innovation plasma applications. He is an Associate Editor of the journal Earth, Moon, Planets, and holds two patents.

Dr. Author was a recipient of the International Association of Geomagnetism and Aeronomy Young Scientist Award for Excellence in 2008, and the IEEE Electromagnetic Compatibility Society Best Symposium Paper Award in 2011.

THIRD C. AUTHOR, JR. (M'87) received the B.S. degree in mechanical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2004 and the M.S. degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently pursuing the Ph.D. degree in mechanical engineering at Texas A&M University, College Station, TX, USA.

From 2008 to 2009, he was a Research Assistant with the Institute of Physics, Academia Sinica, Tapei, Taiwan. His research interest includes the development of surface processing and biological/medical treatment techniques using nonthermal atmospheric pressure plasmas, fundamental study of plasma sources, and fabrication of micro- or nanostructured surfaces.

Mr. Author's awards and honors include the Frew Fellowship (Australian Academy of Science), the I. I. Rabi Prize (APS), the European Frequency and Time Forum Award, the Carl Zeiss Research Award, the William F. Meggers Award and the Adolph Lomb Medal (OSA).