A Stacked LSTM-Based Approach for Reducing Semantic Pose Estimation Error

Rana Azzam, Yusra Alkendi, Tarek Taha, Shoudong Huang, *Senior Member, IEEE*, and Yahya Zweiri, *Member, IEEE*

***Abstract*— Achieving high estimation accuracy is significant for semantic simultaneous localization and mapping (SLAM) tasks. Yet, the estimation process is vulnerable to several sources of error, including limitations of the instruments used to perceive the environment, shortcomings of the employed algorithm, environmental conditions, or other unpredictable noise. In this article, a novel stacked long short-term memory (LSTM)-based error reduction approach is developed to enhance the accuracy of semantic SLAM in presence of such error sources. Training and testing data sets were constructed through simulated and real-time experiments. The effectiveness of the proposed approach was demonstrated by its ability to capture and reduce semantic SLAM estimation errors in training and testing data sets. Quantitative performance measurement was carried out using the absolute trajectory error (ATE) metric. The proposed approach was compared with vanilla and bidirectional LSTM networks, shallow and deep neural networks, and support vector machines. The proposed approach outperforms all other structures and was able to significantly improve the accuracy of semantic SLAM. To further verify the applicability of the proposed approach, it was tested on real-time sequences from the TUM RGB-D data set, where it was able to improve the estimated trajectories.**

***Index Terms*— Deep learning, localization error, long short-term memory (LSTM), measurement uncertainty, semantic simultaneous localization and mapping (SLAM), sensor noise.**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) is one of the most prevalent research problems in the robotics community. It is defined as the problem of estimating the trajectory of a robotic vehicle and incrementally constructing a map of its surroundings, provided with measurements perceived from the environment [1]. SLAM serves as a key enabler of a wide range of applications in mobile robotics, such as search and rescue [2]–[4], autonomous navigation [5], and augmented reality [6]. Semantic SLAM [1] relies on visual measurements obtained by a vision sensor. It exploits understanding of the surrounding structure to build highly expressive maps that are easy for human operators to understand. It started to captivate a tremendous amount of attention, especially after the deep learning breakthrough, which led to advancements in object detection and tracking techniques [7]. The accuracy of the localization is a critical success factor in robotic tasks, particularly those involving interaction with humans. Examples of such tasks are search and rescue, autonomous driving, and elder care. Due to its infancy, semantic SLAM is yet to achieve more robustness in the presence of noisy measurements, such as those occurring due to inaccurate object pose estimation with respect to the vision sensor.

The uncertainty of SLAM estimates might arise due to measurement errors that differ based on the adopted approach to SLAM. In the case of object-based semantic SLAM, errors mostly occur when postprocessing the sensory data to determine the poses of the observed features relative to the sensor in the environment. This process starts with detecting the landmark in the environment and determining its bounding box and then computing its centroid. The centroid of the landmark is then utilized to compute the relative pose between the feature and the vision sensor. Furthermore, occlusions have a significant impact on the accuracy of the estimated object pose [8]. Occlusions happen when part of the object is observed in an image, while the rest is either hidden by other objects in the scene or is out of the field of view of the vision sensor. Due to the advancements in deep learning-based object detectors, occluded objects can still be detected and correctly labeled in an image. Hence, if they are not properly accounted for, the estimated pose of an occluded object can be far from the true one and may consequently cause severe accuracy degradation. In addition, the limitations of the sensors used to perceive the environment introduce another primary source of uncertainty.

The approach proposed in this article aims at reducing the joint effects of several sources of errors on the accuracy of semantic SLAM estimates. These errors might arise from limitations of the software and hardware components used to perform semantic SLAM, from external environmental conditions or unpredictable noise. Formulating a noise model that

Manuscript received May 23, 2020; revised September 16, 2020; accepted September 21, 2020. Date of publication October 22, 2020; date of current version December 22, 2020. This work was supported by the Khalifa University of Science and Technology under Award RC1-2018-KUCARS. The Associate Editor coordinating the review process was Lihui Peng. *(Corresponding author: Rana Azzam.)*

Rana Azzam and Yusra Alkendi are with the KU Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates (e-mail: rana.azzam@ku.ac.ae; yusra.alkendi@ku.ac.ae).

Tarek Taha is with the Robotics Lab, Dubai Future Foundation, Dubai, United Arab Emirates (e-mail: tarek.taha@dubaifuture.gov.ae).

Shoudong Huang is with the University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: shoudong.huang@uts.edu.au).

Yahya Zweiri is with the KU Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates, and also with the Faculty of Science, Engineering and Computing, Kingston University London, London SW15 3DW, U.K. (e-mail: yahya.zweiri@ku.ac.ae).

Digital Object Identifier 10.1109/TIM.2020.3031156

accounts for all such errors is challenging, especially because some errors occur unexpectedly during data collection and/or processing. Hence, a stacked LSTM-based neural network is proposed in this article to learn and capture the error patterns associated with the trajectory estimates of semantic SLAM. By comparing the trajectory estimates to the corresponding ground truth, the network is trained to reduce the error and, hence, enhance the accuracy of semantic SLAM.

The proposed approach is general; it can be used for any SLAM system since it operates on trajectory estimates rather than raw measurements. It targets trajectories of ground vehicles that are usually expressed using three degrees of freedom in the 2-D space: the vehicle's position (x, y) and orientation (ϑ). The approach is applicable to any 2-D SLAM problem. However, the employed neural network was trained on data obtained using semantic SLAM and, hence, is intended to improve the accuracy of semantic SLAM trajectory estimates.

The proposed approach can be used in applications that require accurate localization of the robotic vehicle. For example, a more accurate estimate of the trajectory estimated by semantic SLAM will result in a meaningful and more accurate map of the environment. Another practical use-case scenario of the approach presented in this article is in search-and-rescue applications. If the robots that are employed as first responders after a particular catastrophe have the ability to accurately pinpoint their location, it will expedite the process of rescuing victims, if any, or locating areas that need immediate help.

The contributions of this article are listed as follows.

1) We developed a novel stacked-LSTM-based approach to identify and reduce pose estimation error in object-based semantic SLAM. The approach alleviates the combined effect of predictable and unpredictable noise on the accuracy of trajectory estimates.
2) We developed an automated search approach to select the architecture and hyperparameters of the proposed stacked-LSTM neural network.
3) We extensively tested the proposed approach on simulated and real-time experiments, where its superiority compared with shallow neural networks (SNNs), deep neural networks (DNNs), support vector machines (SVMs), and semantic SLAM was proven.

The rest of this article is organized as follows. Section II presents recent related research work from the literature. The proposed approach is introduced in Section III, followed by experimental validation in Section IV. Finally, the conclusions of this work are drawn in Section V.

II. Related Work

A. Deep Neural Networks

Neural networks are trained to exhibit a particular behavior, suited for the problem at hand, when fed with data. During training, the internal parameters of the network, referred to as weights, are adjusted to minimize the discrepancy between the network's prediction and the desired output [9]. An SNN is a network with an input layer, one hidden layer, and an output layer. Networks with two or more hidden layers are referred to as DNNs. DNNs are much more efficient than SNNs with regards to the required number of computational units, especially when modeling a complex problem. This is attributed to the nonlinear nature of the activation functions occurring at several layers in the DNN [10].

Furthermore, recurrent neural networks (RNNs) are artificial neural networks that are capable of informing knowledge from a context. This is attributed to the use of loops, which allows information to be fed back to the network after being processed. However, such networks might suffer from vanishing gradients, which motivated the need for long short-term memory (LSTM) cells [11]. LSTM cells enable RNNs to retain information that is essential and discard them otherwise. This functionality cannot be realized when using conventional neural networks. DNNs and LSTMs have been exhibiting state-of-the-art performance in a multitude of various applications, including computer vision [12]–[14] and robotics [15]–[17].

B. SLAM and the Intervention of Deep Learning

A rich body of the literature has addressed the SLAM problem, and a wide range of algorithms exhibiting varying levels of performance in terms of reliability, accuracy, and efficiency has been proposed [18]–[20]. The utilization of deep learning approaches has been witnessed in a substantial share of these approaches in the past few years [1], and their capability to outperform the classical approaches has been demonstrated [17], [21]–[24]. In addition, deep learning-based object detection techniques [25]–[27] promoted the advancement of object-based semantic SLAM, which relies on observations of landmarks that can be semantically labeled in the environment, such as the approaches presented in [28] and [29]. Obtaining a reliable observation of a landmark in the environment and accurately pinpointing its position with respect to the sensor remain a challenge. On a different note, much less research effort was made in the area of employing deep learning approaches to improve the accuracy of state estimation, as discussed in the next section.

C. Enhancing SLAM Estimation Accuracy

The accuracy of state estimation in SLAM applications is vulnerable to the effects of several error sources. Such errors occur in one or more stages in the SLAM pipeline, such as data collection, data processing, and optimization. Most of the existing works in the literature assume that noise models always follow fixed distributions that can be mathematically formulated [30]. Nonetheless, this is not always the case in practical applications and might lead to severe degradation in estimation accuracy.

When visual measurements and dead reckoning are used together to estimate the state of a system, estimation uncertainty may result from visual sensor noise [31], [32], landmark detection and localization accuracy [33], odometry drift [34], or failure to arrive at a globally optimum estimate due to measurement noise. The effect of unpredictable nonuniform noise and external environmental conditions is also inevitable [35]. To enhance the accuracy of localization, the solutions found in

the literature can be classified into: 1) controlling the environment under investigation [36]; 2) sensor data fusion [37], [38]; 3) improving measurement covariane estimation [30], [35], [39], [40]; or 4) correcting measurement errors, which can be further classified into classical [41]–[44] and learning approaches [16], [17], [34], [45]–[47].

The work presented in [36] studies the placement of passive tags, used as landmarks, in the environment, to always keep the localization accuracy within a particular range. In another vein, the robustness of indoor localization was supported by accumulating sensory data, which compensated for the limitations of the employed sonar sensors, as presented in [37]. Another example of measurement fusion can be found in [38] where the measurements recorded by multiple IMUs along with other extroceptive sensors were integrated to improve localization accuracy. Instead of assuming a fixed measurement noise model, the work proposed in [39] predicts the noise model based on raw measurements by means of a DNN. The DNN was able to accurately predict the covariance of measurements obtained by light and vision sensors. However, the approach assumes that noise models follow a zero-mean Gaussian distribution that does not hold all the time, putting in doubt the generality of the approach. Similarly, the work presented in [30] relaxes the assumption of a fixed measurement noise model for dead reckoning and QR code detections by employing a tailored extended H_∞ filter. The approach is general and more computationally expensive than the extended Kalman filter yet achieves higher accuracy. Similarly, the approach proposed in [35] improves the accuracy of SLAM estimates by employing an adaptive Gaussian particle filter whose job is to compensate for bias in measurements. More particularly, the approach targets unmodeled, unpredictable noise patterns that are experienced in marine environments. A recent noise model learning approach was proposed in [40] where a DNN was trained to estimate the covariance of inertial measurements, which are then used in an extended Kalman filter to perform localization. Evaluation results demonstrated the applicability of the approach, yet, in its current version, it works for inertial sensors only.

Several previous studies have addressed the correction of measurements using classical or learning approaches. In [41], visual sensor limitations were accounted for by superimposing camera oscillations to improve the accuracy of visual SLAM. The work presented in [42] utilizes probabilistic fuzzy logic to reduce measurement uncertainties occurring due to stochastic and nonstochastic disturbances. This approach handles dead-reckoning and range measurements and was proven to outperform ordinary fuzzy logic in terms of improving the accuracy of positioning and mapping estimates. The work presented in [43] was able to cope with the occasional failure of inertial sensors during localization, by means of a discrete-time H_∞ filter. Localization was supported by a reference wireless sensor network. In [44], a novel ultrasonic sensor with self-configuration abilities was developed to cope with collisions of ultrasonic waves and, hence, enhance localization accuracy. The developed algorithm is also capable of handling topological changes in the environment in a real-time manner.

In [16], a deep learning-based approach is employed to improve the altitude estimation of a flying robotic vehicle. Moreover, in [45], the accuracy of the odometry of a wheeled cart, calculated using its dynamic equations, was improved using an SNN. The network was trained to compute an estimate of the vehicle's traveled distance and orientation. However, since the network is composed of a single hidden layer only, it might not capture all patterns of estimation errors and, hence, cannot generalize well. Correction of odometry measurements was also addressed in [34] and [46] where the Gaussian processes were trained based on the discrepancies between the odometry model and ground truth. The scalability and accuracy of the model proposed in [46] were achieved through deep kernel learning. Furthermore, the approach proposed in [17] improves the accuracy of stereo visual localization. The authors proposed a loss function based on the Lie group $SE(3)$ geodesic distance and used it to train a DNN to more accurately estimate the relative transformation between subsequent images. The advantages of the proposed approach over classical visual odometry were demonstrated through several experiments. However, the DNN operates on images, which makes the approach computationally expensive and requires a large DNN structure to achieve the sought performance. In addition, an end-to-end learning approach to visual odometry was presented in [15] where sequential learning was employed to improve the pose certainty. The DNN operates on raw images, from which it infers the uncertainty of the poses. Operating on large amounts of data requires the hardware to have high computational capabilities. The work proposed in [47] utilizes deep learning to improve depth estimation, which is then used to perform dense monocular SLAM. In a classical factor graph approach, the authors propose to use multiple objective functions, or factors, addressing several types of errors to further improve estimation accuracy. These factors are the photometric, reprojection, and sparse geometric factors. Combining these factors has resulted in robust motion estimation when tested on several real-time sequences.

The proposed stacked-LSTM-based approach has the following advantages over the aforementioned methods.

1) It alleviates the effects of all the possible disturbances experienced while performing SLAM, including measurement errors, sensor failure, data processing faults, or any other unpredictable noise.
2) It operates on a trajectory rather than raw sensor measurements, such as images. Hence, it can be used to reduce pose estimation errors irrespective of the employed sensors.
3) It is efficient since the input to the neural network is a short segment of the trajectory, which takes much less time than images to be processed.
4) It does not require any particular arrangement of the environment. More specifically, it does not depend on the number, geometry, or placement of landmarks in the environment.
5) The stacked nature of the LSTM and dense layers along with the nonlinear activation functions facilitate identifying complex error patterns that could be challenging to model mathematically.

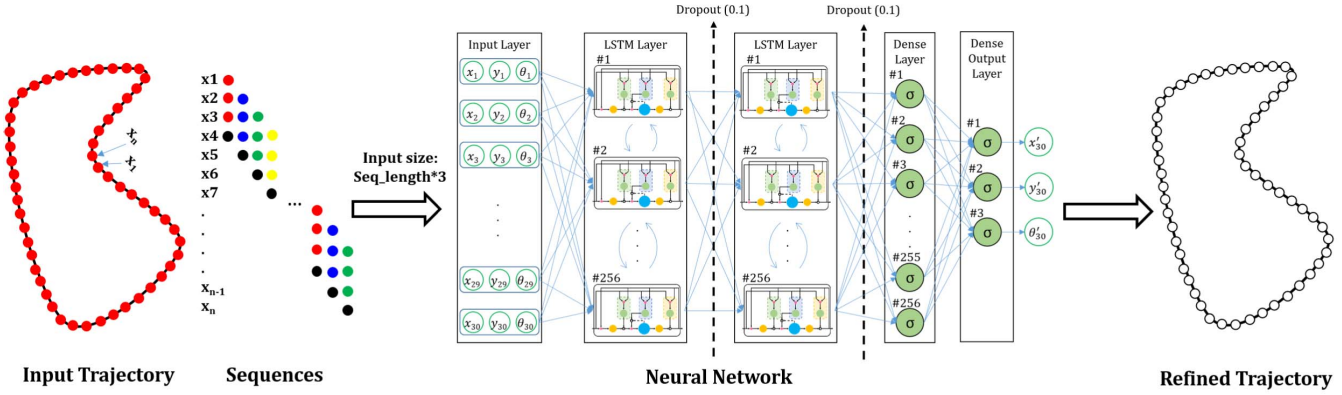


Fig. 1. Proposed deep learning approach.

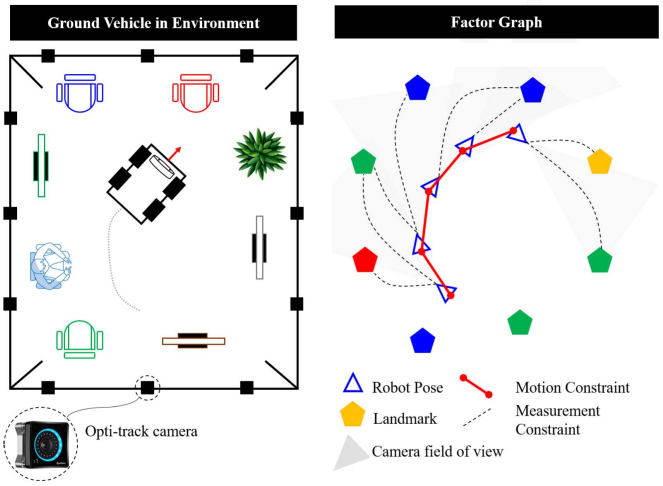


Fig. 2. Semantic SLAM.

III. Proposed Approach

The deep learning approach proposed in this article is depicted in Fig. 1. In general, a ground vehicle's trajectory, estimated using semantic SLAM, is passed to a neural network, which will identify and reduce possible pose estimation errors. The semantic SLAM algorithm will be described in Section III-A along with the error sources that contribute to reducing the accuracy of pose estimation. Section III-B details the deep learning-based pose estimation error reduction approach.

A. Semantic SLAM

The adopted semantic SLAM is designed for ground vehicles and is performed based on measurements from the vehicle's wheel encoders and an RGB-D camera that is mounted on top. A factor graph is used to model the problem, as shown in Fig. 2, where robot poses and map features (landmarks) are represented as nodes. The graph contains two types of edges: solid edges between every two consecutive pose nodes, to represent a spatial constraint (denoted as e_{mot}), and dashed edges between a pose node and a landmark node, to represent an observation of the landmark at that pose (denoted as e_{meas}). Each edge models a nonlinear quadratic constraint, which can be mathematically formulated, as shown in the following equation:

$$e_{\text{mot}} = (x_t - g(u_t, x_{t-1}))^T R_t^{-1} (x_t - g(u_t, x_{t-1})) \quad (1)$$

where g is the motion model, u_t is the control command, x_t is the robot pose at time t, and R_t is the covariance matrix of the motion noise, which is assumed to be Gaussian

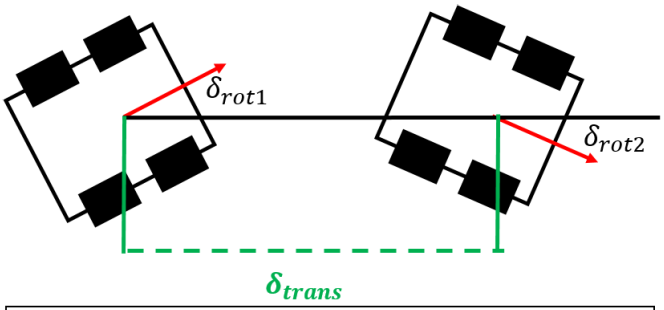
$$e_{\text{meas}} = \left(z_t^j - h(x_t, m_j)\right)^T Q_t^{-1} \left(z_t^j - h(x_t, m_j)\right) \quad (2)$$

where z_t^j are measurements, h is the measurement function, m_j is a landmark, and Q_t is the covariance matrix of the measurement noise, which is assumed to be Gaussian.

The goal of graph SLAM is then to find a configuration of the nodes that minimize the error introduced by the constraints. In our approach, this is done by means of the incremental smoothing and mapping algorithm, iSAM2 [48].

1) Landmark Pose Estimation and Data Association: To perform semantic SLAM, the vehicle's relative position to the observed landmarks must be computed. This is done using input RGB-D frames. RGB images are passed to the object detector, you only look once (YOLO) [25]. For each detected object in an image, YOLO predicts a label and a bounding box. The relative position between the camera and the detected object is then computed as the distance between the camera and the centroid of the object, as proposed in [29]. Briefly, the input depth image that corresponds to the input RGB image is converted to a point cloud. The point cloud is segmented in order to extract the cluster of points that belong to the detected landmark using a kd-tree search. The geometric centroid of the cluster is then computed.

Detected objects are associated with landmarks in the map based on their label, which is predicted by YOLO. Since multiple objects of the same category might exist in the environment, the observation is associated with the closest landmark within a particular distance threshold. If no landmarks exist within that threshold, a new instance is inserted in the map.



$\delta_{rot1}, \delta_{rot2}$, and δ_{trans} represent the true motion.

Motion corrupted with noise is computed as:

$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1|\delta_{rot1}|+\alpha_2|\delta_{trans}|}$$
$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3|\delta_{trans}|+\alpha_4|\delta_{rot1}+\delta_{rot2}|}$$
$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1|\delta_{rot2}|+\alpha_2|\delta_{trans}|}$$

Where $\alpha_1, \alpha_2, \alpha_3$ and α_4 are the motion noise model parameters

Fig. 3. Odometry noise model.

2) Measurement Uncertainty: There are several factors that contribute to reducing the estimation accuracy when performing semantic slam. Starting from the inputs, the sensors used to perceive the environment suffer from limitations that decrease the accuracy of the obtained measurements. For example, the uncertainty of RGB-D measurements might be caused by: 1) axial noise [49] that increases when the distance to the detected object increases; 2) lateral noise [49] that increases near the image corners; 3) multipath interference [50]; 4) flying pixels [50]; and 5) the scene's characteristics, such as color variations, temperature [50], and illumination conditions. In addition, odometry drift is affected by the accuracy of wheel encoders [34], wheel materials, floor flatness, and materials. Fig. 3 describes the model that was used to simulate odometry noise and add it to simulated odometry, which is considered to be perfect with no error.

Furthermore, object detection might result in incorrect labels or bounding box predictions. If such false detections are not treated as outliers, the accuracy of pose estimation and data association is severely affected. Given an accurate object label and bounding box, the pose estimation module is yet error-prone. Depending on the structure of the environment under investigation, object classes, camera position, illumination conditions, and, most importantly, object occlusions, the accuracy of segmentation and clustering can be significantly reduced.

Modeling such errors can be extremely challenging; therefore, guaranteeing a maximum likelihood estimate using the employed incremental smoothing and mapping technique is difficult.

The input to the neural network is the trajectory estimated by the semantic SLAM system. The estimation is based on observations that may sometimes be inaccurate due to several sources of error affecting data acquisition and/or processing. For example, the aforementioned types of RGB-D sensor noise may degrade the quality of the acquired RGB-D frames, which will consequently affect the accuracy of the information obtained from such images. Also, object detection, labeling, and segmentation are subject to errors that may negatively impact the corresponding measurement constraints. In simulated experiments, the noise was simulated and added to the measurements to mimic the real noise.

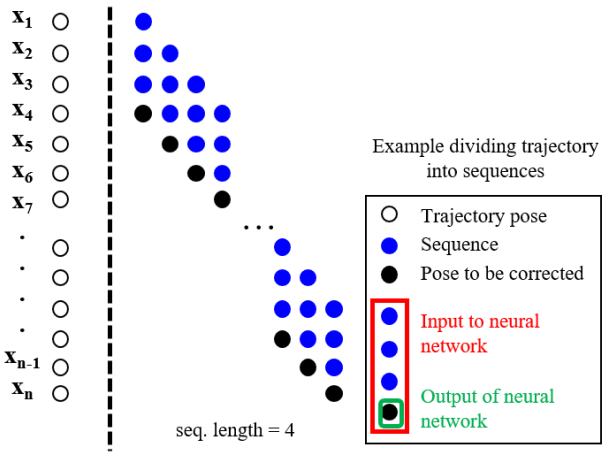


Fig. 4. Example dividing a trajectory to sequences of length 4.

The motion measurements and observations are passed to the optimization algorithm, in the SLAM back end, along with an estimate of the measurement noise model. The optimization algorithm then estimates the trajectory to find a configuration of the poses that minimizes the overall error along the trajectory. The resulting estimate of the robot trajectory still suffers from estimation errors since it was based on observations inferred from inaccurate measurements that propagate along the SLAM pipeline.

The neural network is trained to identify the error patterns in the final estimate of the robot trajectory by comparing it to the corresponding ground truth. For every pose, the neural network exploits a segment of the trajectory that precedes that pose to determine the pose estimation error and reduce it accordingly.

B. Stacked LSTM-Based Noise Reduction Approach

To find the best-suited neural network architecture, a systematic search was done in a pool of neural networks of varying types, depths, and activation functions. Three types of LSTM networks were explored: simple vanilla LSTM, stacked LSTM, and bidirectional LSTM. Moreover, a set of shallow and deep fully connected feedforward neural networks were investigated. A hybrid of LSTM and fully connected layers was also considered. The performance of all the tested networks was evaluated using a data set containing data generated from the simulated trajectories. The training and validation results were compared using the absolute trajectory error (ATE). More specifically, the Euclidean distance between the ground-truth pose and the corrected pose by the network is computed for all poses along the trajectory, and the mean error is used to compare the performance of the different networks.

Each of the tested neural networks takes in a segment of the trajectory, consisting of the current 2-D pose, along with a number of previous poses. The length of this segment will, hereinafter, be referred to as sequence length. An illustrative example of how a trajectory is divided into sequences of length 4 is depicted in Fig. 4. Each pose in this segment is denoted

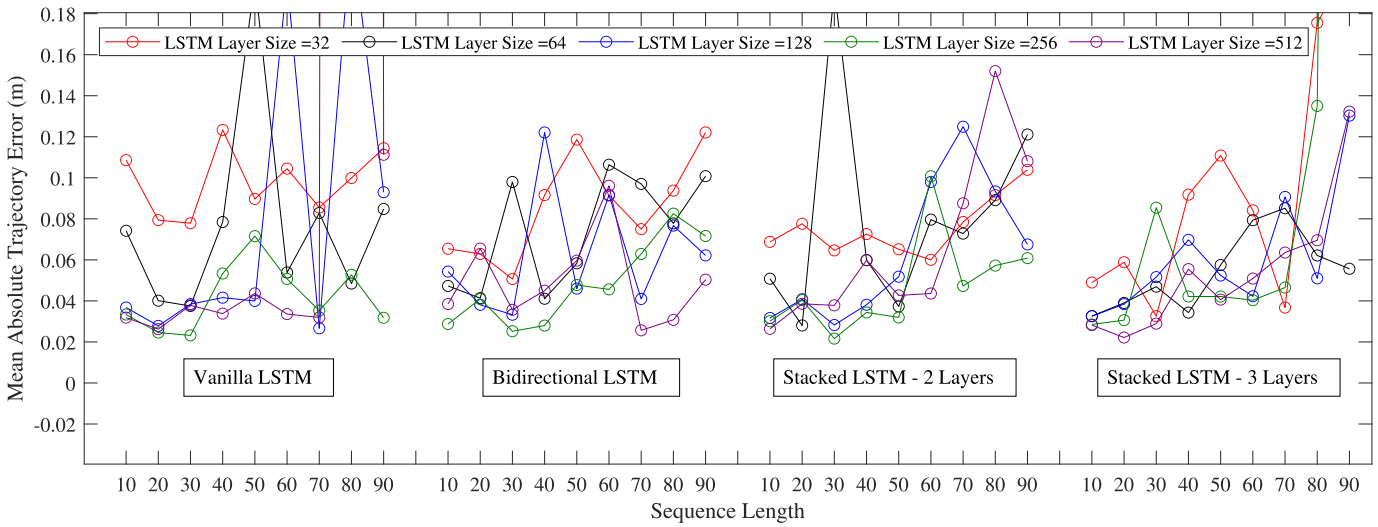


Fig. 5. Mean ATE obtained on the training and validation data sets by different LSTM network architectures.

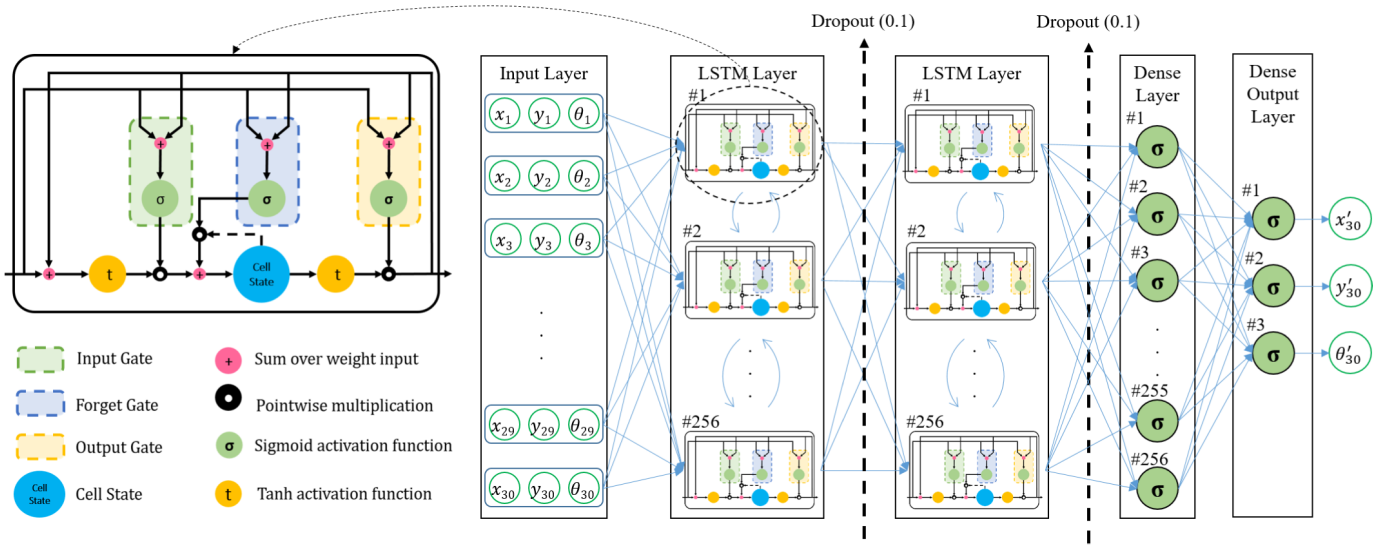


Fig. 6. Proposed neural network architecture.

as $X = [x_m, y_m, \vartheta_m]^T$, and hence, the input is a 2-D array of poses. It is worth mentioning that the network is not expected to predict a new pose following the input segment. Rather, it learns to correct the last pose based on the previous poses in the segment. Therefore, the output of the neural network is a three-tuple that represents the pose and is obtained by means of a dense layer of size three, activated using sigmoid for all the tested neural networks.

First, a search was conducted to determine the most suited type of LSTM networks; vanilla, bidirectional, or stacked. Several LSTM networks with a varying number of units and sequence lengths were trained and evaluated. The number of units in each LSTM layer was set to 2^m for $m \in [5, 9]$. The sequence length was varied between 10 and 90, in increments of 10. In addition, stacked LSTM was tested with two and three LSTM layers. The mean ATE obtained on the training and validation data sets by all the tested LSTM network architectures are shown in Fig. 5. Several architectures performed well and were able to improve the accuracy of the estimated trajectory. However, the architecture with two stacked LSTM layers, each with 256 units, with a sequence length of 30 exhibited the highest performance among all the considered LSTM networks in terms of reducing ATE.

In an attempt to further improve the results, one, two, and three dense layers were added after the LSTM layers, and various sizes and activation functions were tested, including sigmoid, swish [51], tanh, ReLU, and linear. Adding a dense layer of size 256 with a sigmoid activation function resulted in smoother trajectories with lower mean ATE. The mean ATE achieved by the other hybrid architectures did not improve. To aid generalization and overcome overfitting, dropout layers were added to the architecture.

Fig. 6 depicts the architecture of the adopted stacked LSTM neural network. The network accepts trajectory segments of length 30. Each of the poses in a segment consists of a three-tuple, $X = [x_m, y_m, \vartheta_m]^T$, representing the position and orientation of the vehicle at a time instance. The segment

is then passed to two stacked LSTM layers, separated by a dropout layer, with a dropout probability of 0.1. Each LSTM layer consists of 256 units. Then, after another dropout layer, a fully connected layer with 256 neurons, activated by sigmoid, is added. Finally, a dense layer, activated by sigmoid, is used to predict the improved pose.

The computational complexity of the proposed model per time step is $O(W)$, where W is the size of the weight space. This is attributed to the fact that the time complexity to update a single weight is $O(1)$. The size of the weight space is a function of the input size, hidden units, and output size [11], which were detailed earlier. The total number of trainable parameters in the proposed model is around 850k parameters. The architecture of the proposed model and its hyperparameters, such as the batch size and the number of training epochs, are fixed. Hence, for N training samples, the complexity becomes $O(N)$ since one training epoch runs in $O(1)$.

The proposed neural network will be compared with shallow and deep fully connected neural networks (SNN and DNN, respectively) and SVMs. Hence, a pool of SNNs, DNNs, and SVMs was investigated to search for the best structure from each paradigm for our problem. The parameters that were varied for SNN and DNN are the number of neurons per layer and the activation functions. Different depths of DNNs were also attempted. As for SVMs, a set of variables, such as the kernel and its corresponding parameters, were changed, and the SVM that resulted in the lowest mean ATE across the training and validation data sets was selected. The search for the most suited SNN, DNN, and SVM was done to ensure the fairness of our comparisons.

IV. Experimental Validation

In this section, the proposed approach is validated through a set of simulated and real-time sequences from publicly available data sets. The performance of the stacked LSTM neural network is then compared against other regression techniques, including SVM, SNN, and DNN, where it proved to outperform them.

As mentioned earlier, the proposed approach can be applied to trajectories computed using any 2-D SLAM. However, the data sets used here were obtained using semantic SLAM.

The rest of this section is organized as follows. The experimental setup used to record the training data set is presented in Section IV-A. In Section IV-B, the structure of the training data sets is described, followed by details about the training process. After that, the performance of the proposed approach is analyzed and compared with that of SNN, DNN, and SVM in Section IV-C. Finally, in Section IV-D, the proposed approach is tested on a set of simulated and real-time experiments, including three SLAM sequences from the TUM RGB-D data set [52].

A. Experimental Setup

A simulated pioneer 3AT robot with an RGB-D camera mounted in a front-forward position was used to navigate in several simulated environments and collect the sensory data required performing semantic SLAM. A sample of the simulated environments is shown in Fig. 7 with the simulated robotic vehicle at its starting point. This environment is 13×10 m^2 and is populated with 37 object instances of two different categories: 19 TV monitors and 18 bottles. Other simulated environments were also used where the object categories include potted plant, table, and person (who are assumed to be static while recording the experiment).

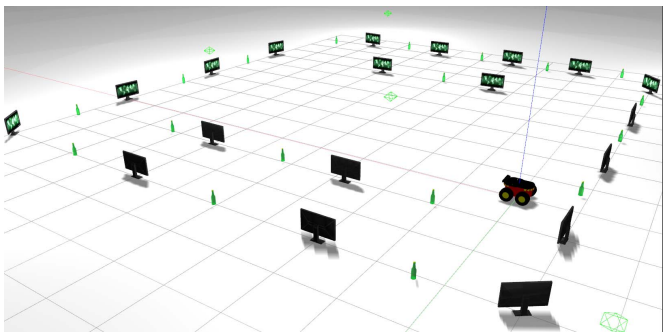


Fig. 7. Simulated environment setup with ground vehicle at its starting point.

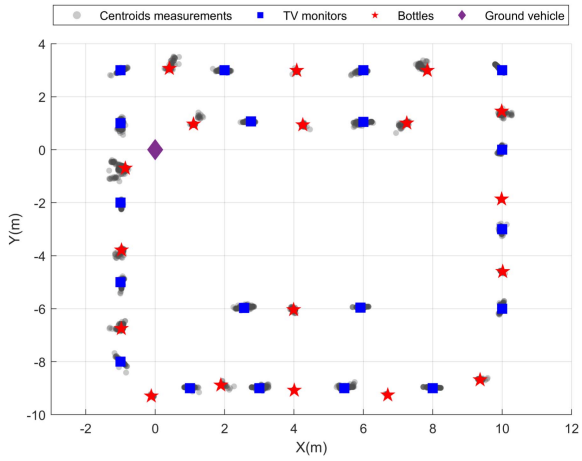


Fig. 8. Distribution of object measurements from some recorded simulated experiments in the environment shown in Fig. 7.

Several simulated trajectories were recorded in these simulated environments and then passed to the semantic SLAM algorithm to generate training data. Since the odometry measurements obtained from simulations are perfect, the odometry noise model described in Fig. 3 was used to simulate the noise and add it to the recorded measurements. Another source of error was observed when passing the RGB frames to the object detector, YOLO [25], and then performing segmentation to determine the centroid of the observed object. The centroids of the detected objects were seen to deviate from their true positions. This was mainly due to object occlusions since YOLO was able to detect an object even if part of it is occluded. Consequently, only the visible part of the object was used to compute the centroid of that instance, causing an error in the measurement. The error varies depending on the size of the object, and hence, the standard deviation of the noise associated with each object observation was set based on the object's dimensions. Object measurements from some recorded experiments in the previously described simulated environment are shown in Fig. 8.

TABLE I
SUMMARY OF TUM RGB-D SEQUENCES USED FOR EVALUATING THE PROPOSED APPROACH

Name	Tajectory Length	Trajectory Dimensions
Freiburg_pioneer_slam	40.38m	5.50m x 5.94m
Freiburg_pioneer_slam2	21.735m	4.98m x 5.34m
Freiburg_pioneer_slam3	18.135m	5.29m x 5.25m

TABLE II
HARDWARE SPECIFICATIONS

Computer type:	ASUS STRIX laptop
Processor:	with Intel core i7-6700HQ @ 2.60GHz × 8
System type:	64-bit operating system
Operating System:	Linux – Kubuntu 16.04 distribution

Real-time experiments were taken from the TUM RGB-D data set [52], where a Pioneer 3AT robot with a Kinect RGB-D sensor mounted in a front forward position was joysticked in a large hole. Multiple instances of chairs and tables appeared in the environment and were used as observations to perform semantic SLAM. Recordings of experiments are provided with the corresponding ground truth, which was used during the training process. Table I summarizes the details about the three trajectories taken from the data set.

The specifications of the computer used to conduct the semantic SLAM experiments are listed in Table II.

The semantic SLAM algorithm was implemented using the robot operating system (ROS) [53] Kinetic on Ubuntu 16.04. The communication between the simulated/real hardware and ROS was performed through RosAria and OpenNI for the ground vehicle and the RGB-D sensor, respectively. The system software was implemented in C++, where gtsam [54] and its iSAM2 [48] implementation was used to perform incremental smoothing and mapping, YOLO [25] was used for object detection, openCV [55] and depth_image_proc were used for processing RGB and depth images, respectively, and point cloud library (PCL) [56] was used to process point clouds.

B. Data Set Preparation

A total of 18 different simulated trajectories were generated and used to construct the data set. Every estimated trajectory was divided into smaller overlapping segments of length 30, as described in Fig. 4. The ground truth, corresponding to the last pose in each segment, is set as the target for that input segment and is referred to as $T = [x, y, \vartheta]^T$. In simulated experiments, the recorded odometry measurements, before adding simulated noise, are set as the target. In real-time experiments, ground-truth data were obtained from the data set online.

The output of the network is an improved estimate of the robot's 2-D poses along the trajectory, where each pose is denoted as $Y = [x_e, y_e, \vartheta_e]^T$. It is worth mentioning that, in the current version of the system, the reduction of pose estimation error is done offline.

For the neural network to perform well, the data sets need to be rescaled to a common range. To that end, all the collected data were normalized to the range $[0.05, 0.95]$. The reason why this particular range is selected is to avoid the problem of vanishing gradients that occurs when the neurons saturate, i.e., reach the minimum or maximum value of the activation function (0 and 1, respectively, for sigmoid), and, hence, the derivative of the function at that point drops to values close to zero. Using the same normalization parameters, predictions were rescaled to the original data range.

Backpropagation [57] was used to train the network in a supervised manner. The Adam optimizer was employed, with a learning rate of 0.0001, to minimize the mean absolute error, over 1000 training epochs. The batch size was set to 100. The building, training, and testing of the different neural networks were done using Keras [58] with a Tensorflow back end (version 2.0).

C. Performance Evaluation

In this section, the performance of the proposed stacked LSTM will be compared with DNN, SNN, and SVM using a set of simulated trajectories. The data sets were randomly split into two parts: 80% for training and 20% for validation to aid the model's regularization.

To ensure fairness when comparing the proposed approach to DNN, SNN, and SVM, a similar search strategy was adopted to find the most suited architecture using the same training data set. Fully connected SNNs with varying activation functions, including sigmoid, swish, ReLU, linear and tanh, and layer sizes set to 2^m for $m \in [5, 9]$, were examined. Along the same lines, fully connected DNNs with depths varying from two to six layers, layer sizes set to 2^m for $m \in [5, 9]$, and activation functions, including sigmoid, swish, ReLU, linear, and tanh, were investigated. The SNN and DNN that achieved the lowest mean ATE were selected to be compared with the proposed approach. The SNN that performed the best in terms of reducing pose estimation error had 512 neurons in its single hidden layer and was activated using ReLU. The DNN, on the other hand, had six hidden layers, each of size 256 neurons, and activated using sigmoid. The third regression technique that will be compared against the proposed approach is SVM. More particularly, varying structures of the epsilon support vector regression model [59] were explored. Several kernels, including linear, sigmoid, and polynomial with varying regularization and epsilon parameters, were tested. The best performing SVM out of the tested pool was of a 5° polynomial kernel. The predictions of these three models were compared with that of the proposed stacked-LSTM approach, as will be presented in the following.

Fig. 9 depicts nine different trajectories, along with which the performance of the proposed approach is evaluated and compared with the other alternatives. It is evident that the proposed stacked LSTM was capable of identifying error patterns in the input trajectories and significantly improving them along all the depicted trajectories. DNN predictions have also shown substantial improvements to the trajectories, yet there are segments of the trajectories where DNN predictions still suffered from errors. SNN predictions demonstrated improvements at times, but, especially after turns in the

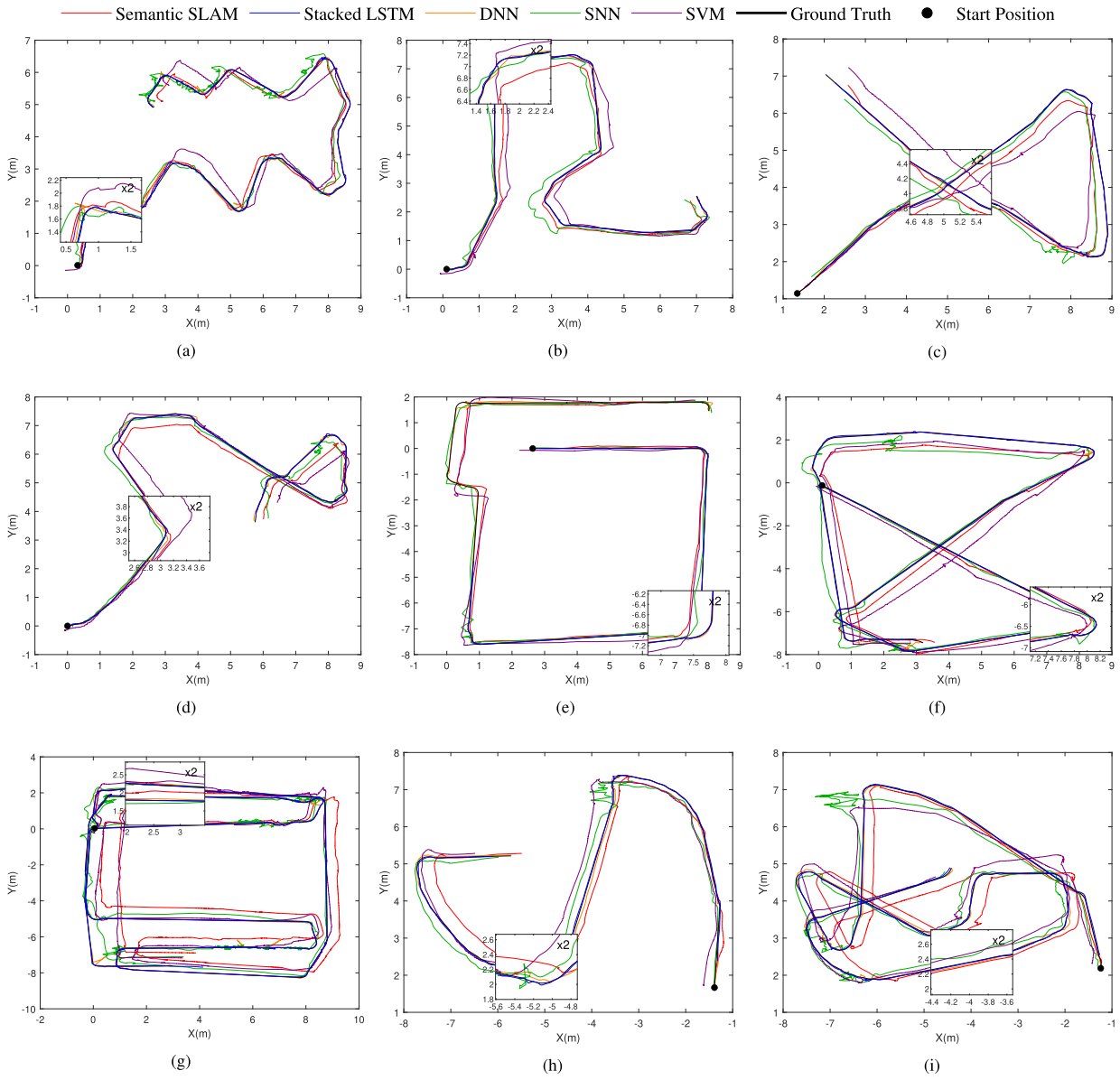


Fig. 9. Training and validation results on simulated trajectories. (a) Trajectory 1. (b) Trajectory 2. (c) Trajectory 3. (d) Trajectory 4. (e) Trajectory 5. (f) Trajectory 6. (g) Trajectory 7. (h) Trajectory 8. (i) Trajectory 9.

trajectory, predictions exhibited high fluctuations and, hence, large ATE. SVM predictions were mostly less accurate than the input trajectories generated by semantic SLAM, and hence, no improvement to the ATE was observed.

Table III lists the mean ATE achieved by the stacked LSTM network, SNN, DNN, and SVM along each trajectory. SVM predictions have not shown any improvement to the mean ATE along any of the nine trajectories. Stacked LSTM, SNN, and DNN, on the other hand, were able to reduce the mean ATE along all the trajectories. However, the proposed approach has clearly outperformed all the other alternatives, by achieving the lowest mean ATE along all the trajectories.

TABLE III

COMPARISON OF MEAN ATE (m) ACHIEVED ALONG THE TRAJECTORIES IN FIG. 9

	Semantic SLAM	Stacked LSTM	SNN	DNN	SVM
Trajectory 1	0.20	**0.025**	0.19	0.048	0.68
Trajectory 2	0.20	**0.021**	0.16	0.035	0.59
Trajectory 3	0.30	**0.025**	0.16	0.034	0.68
Trajectory 4	0.26	**0.024**	0.15	0.040	0.58
Trajectory 5	0.17	**0.019**	0.095	0.034	0.57
Trajectory 6	0.32	**0.024**	0.26	0.037	0.75
Trajectory 7	0.71	**0.022**	0.22	0.042	0.86
Trajectory 8	0.24	**0.024**	0.15	0.047	0.42
Trajectory 9	0.32	**0.025**	0.17	0.042	0.48

D. Performance Analysis on Publicly Available Data Sets

To further verify the applicability of the proposed approach, the training data set was extended to include more simulated and real-time experiments. The latter was taken from publicly available data sets that are used as a benchmark by the robotics community, particularly, Freiburg2_Pioneer_SLAM, Freiburg2_Pioneer_SLAM2, and Freiburg2_Pioneer_SLAM3 from the TUM RGB-D data set [52]. These public

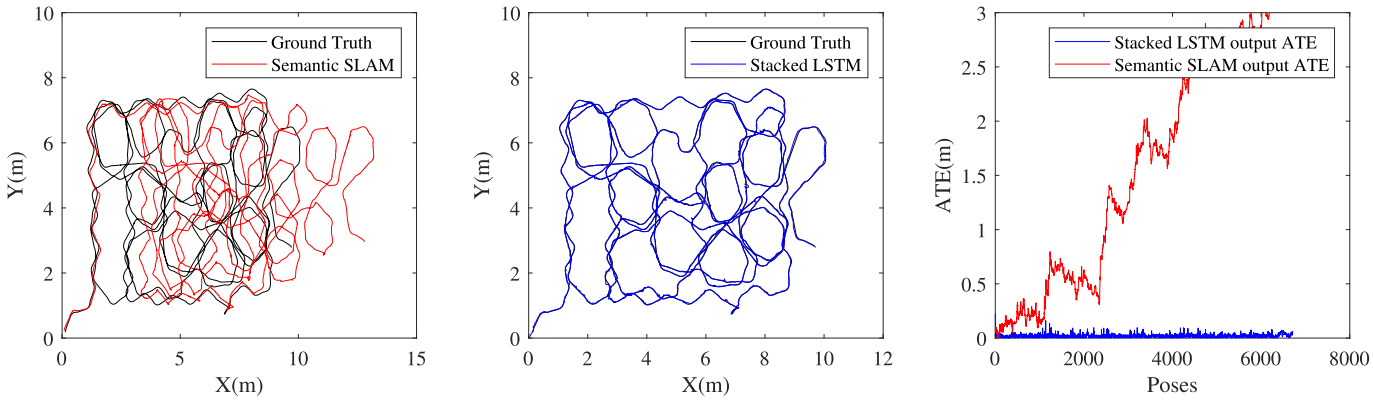


Fig. 10. Simulated experiment—sample 1.

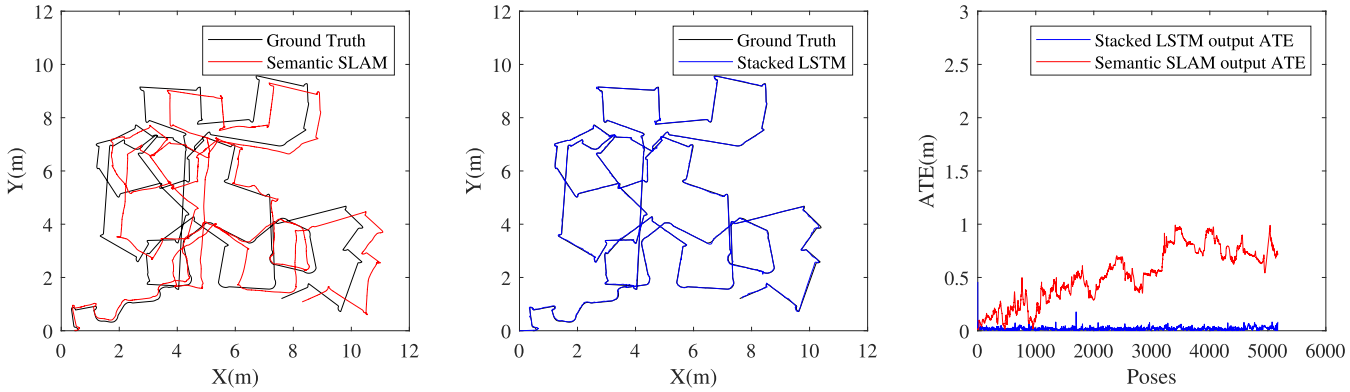


Fig. 11. Simulated experiment—sample 2.

data sets resemble a practical use-case scenario where a ground vehicle performs a maneuver in an indoor environment, populated with objects. The vehicle is equipped with wheel encoders and an RGB-D sensor mounted in a front-forward position, as described in Section IV-A. Vision measurements and the concurrent odometry are passed to the semantic SLAM system, which estimates the vehicles' trajectory in the environment. The process of acquiring and processing data is vulnerable to several sources of error that hinder the accuracy of the estimated trajectory. To reduce such inaccuracies, the trajectory is passed to the stacked-LSTM neural network, which, in turn, identifies and reduces pose estimation errors. The trajectory is divided into small overlapping segments, as depicted in Fig. 4 and described in Section III-B. Each pose along the trajectory is corrected based on the vehicles preceding poses. In the current version, the correction is done offline. The same process is applicable to any 2-D SLAM estimate.

The training data set, including all the simulated and real-time trajectories, was divided into three parts: 70% for training, 15% for validation, and 15% for testing. The training set will be used to optimize the weights of the network during the training process. The performance of the neural network on the validation set will be used to further update the network's weights after every training epochs. Finally, an unbiased evaluation of the network will be obtained using the testing set. Table IV lists the mean ATE that the proposed approach achieved on the training, validation, and testing sets. It is evident that the stacked LSTM was able to identify and significantly reduce the error patterns along the trajectories in the data set. The mean ATE on the testing data set, which was not exposed to the network during training, dropped from 65 to 2 cm. This proves the validity of the proposed approach on simulated and real-time experiments. Examples of trajectories from simulated and real-time experiments are depicted in Figs. 11–15. The leftmost plot in each figure shows the ground-truth trajectory and the trajectory estimated by semantic SLAM. The middle plot shows the output of the proposed approach compared with the ground-truth trajectory. The rightmost plot depicts the ATE along the trajectory for both the input to the network and its output. Fig. 15 depicts the regression plot of the variable ϑ for the sequence depicted in Fig. 14. Pearson's regression coefficient R is equal to 0.99996, as shown in the plot. The regression plot for ϑ in other sequences is very similar.

TABLE IV

MEAN ATE (m) ACHIEVED BY THE PROPOSED APPROACH ON THE TRAINING, VALIDATION, AND TESTING DATA SETS

	Training set	Validation set	Testing set
Input mean ATE (m)	0.65	0.66	0.66
Output mean ATE (m)	0.021	0.026	0.025

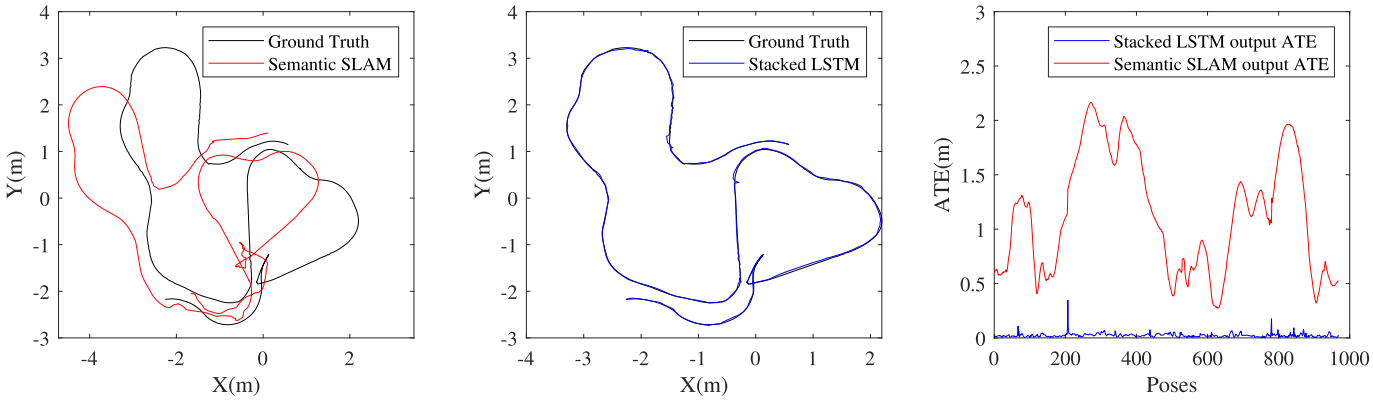


Fig. 12. Real-time experiment—Freiburg2_Pioneer_SLAM sequence.

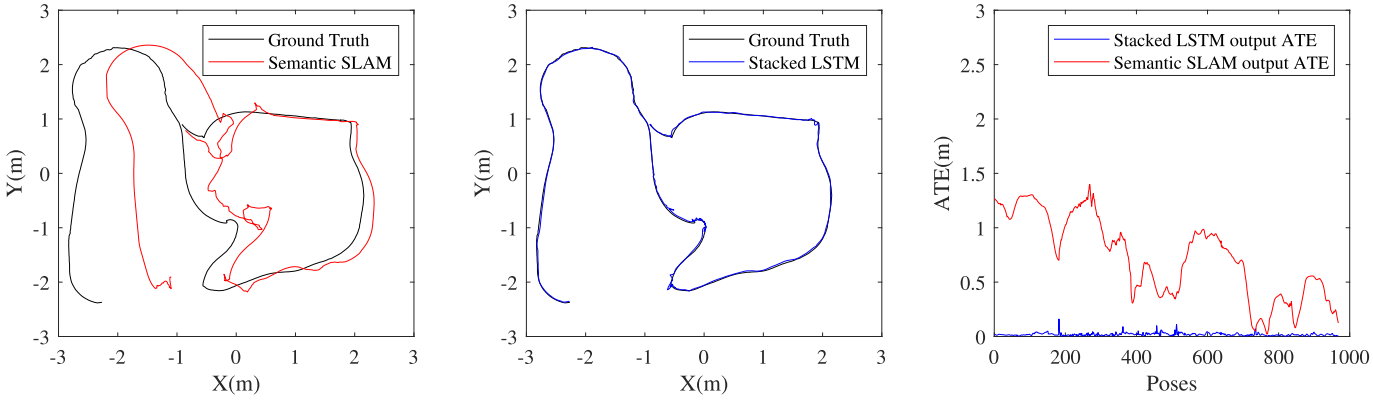


Fig. 13. Real-time experiment—Freiburg2_Pioneer_SLAM2 sequence.

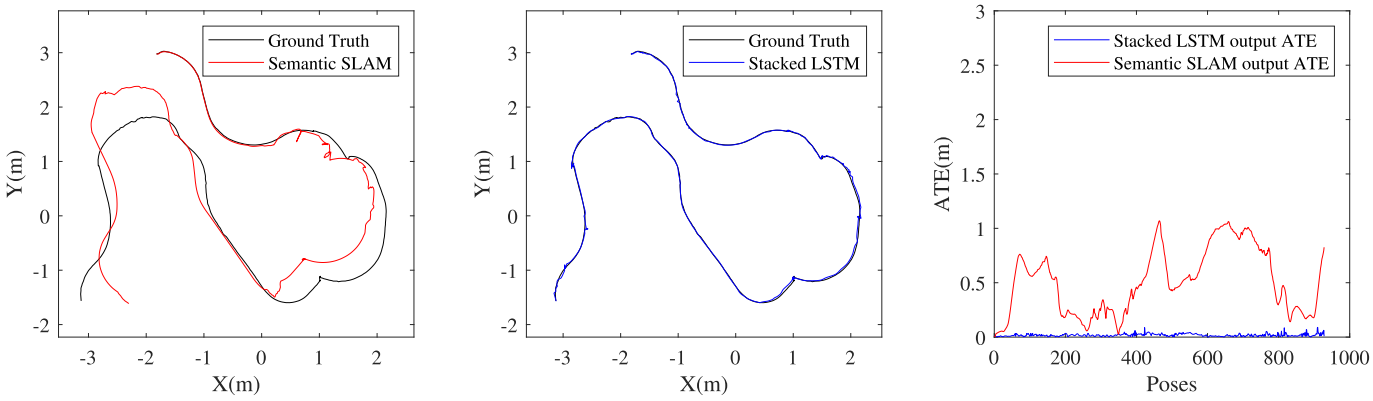


Fig. 14. Real-time experiment - Freiburg2_Pioneer_SLAM3 sequence.

The proposed stacked-LSTM-based approach can generalize well and is robust to input perturbations. However, the network may be fine-tuned to learn new noise models that were never exposed to the network during training and, hence, be able to recognize a wider variety of pose estimation errors. A portion of the data obtained from the new environment can be used to fine-tune the network, which will then be able to reduce pose estimation error along trajectories recorded under the same conditions. If the training data set consists of a wide range of error patterns, the neural network will have more potential to perform error reduction along previously unseen trajectories.

A possible use-case scenario of the proposed approach is in search-and-rescue applications. A robot's mission could be to find victims in a collapsed structure and then notify the rescue teams of the victim's location. While performing semantic SLAM, the robot can navigate in the environment, and once a victim is found, the robot's trajectory is communicated to the rescue team. This trajectory is first refined, using the proposed stacked-LSTM-based noise reduction approach, to pinpoint the robot's position accurately. This will help the human responders to arrive at the location in a shorter time.

TABLE V
INPUT AND OUTPUT MEAN ATE (m)

	Input mean ATE (m)	Output mean ATE (m)
Simulated Trajectory 1 (Fig. 10)	1.59	0.051
Simulated Trajectory 2 (Fig. 11)	0.53	0.041
Freiburg2_Pioneer_SLAM Trajectory (Fig. 12)	1.11	0.057
Freiburg2_Pioneer_SLAM2 Trajectory (Fig. 13)	0.69	0.048
Freiburg2_Pioneer_SLAM3 Trajectory (Fig. 14)	0.52	0.042

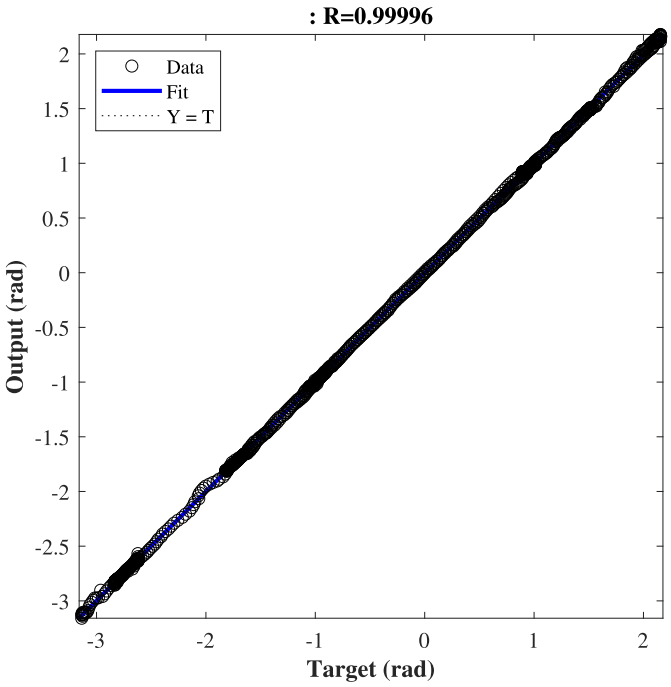


Fig. 15. Regression plot for the orientation variable ϑ for the Freiburg_ Pioneer_SLAM3 sequence.

Another use-case scenario of the proposed approach can be seen in applications that require the robot to map its surrounding environment. After the robot gathers the required visual information from the environment, its trajectory is estimated using semantic SLAM and then passed to the proposed stacked-LSTM-based neural network for possible error identification and reduction. Reprojecting visual observations along the corrected trajectory will result in a more accurate and reliable map of the environment compared with that obtained directly from the solver.

V. CONCLUSION

The error in estimating vehicle and landmark poses significantly hiders the success of semantic SLAM and its usability in high-accuracy critical applications. Several predictable and unpredictable sources of uncertainties contribute to forming such error, including landmark local pose estimation, object detection, incorrect data association, visual sensor noise, and odometry drift. The work proposed in this article employs a novel, general, and efficient deep learning approach to enhance the robustness of semantic SLAM, by reducing the combined effect of such errors on the trajectory estimation. A stacked LSTM-based neural network was developed after conducting an extensive search among different neural network types and hyperparameters. The architecture was adopted based on the network's ability to capture various error patterns and significantly decrease the estimation error. Simulated and real-time experiments, including three sequences from the TUM RGB-D data set, were used to measure the performance of the proposed approach. The results have proven the ability of the proposed approach to successfully identify and reduce pose estimation errors resulting from multiple factors in the semantic SLAM pipeline. The performance of the neural network was quantified using the mean ATE. It was compared with that of SNN, DNN, and SVM on several test sets, and the maximum estimation error reduction was evidently achieved by the proposed approach.

The capability of the proposed approach to generalize well to new semantic SLAM trajectory estimates relies heavily on the training data set. The data set should include samples collected from different environments, under varying conditions, and using various sets of sensors. If the network is exposed to a wide range of error patterns during training, it will have a higher potential to perform error reduction in previously unseen trajectories.

The work presented in this article can be extended to 3-D SLAM that applies to a wider range of robotic vehicles, such as aerial vehicles. In such a case, the neural network will be expected to predict more variables and account for various noise models. It can also be integrated into the online SLAM algorithm, and hence, error reduction will be performed right after computing a new SLAM estimate, making it more efficient.

REFERENCES

[1] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[2] X. Chen, H. Zhang, H. Lu, J. Xiao, Q. Qiu, and Y. Li, "Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 41–47.
[3] A. Denker and M. C. Iseri, "Design and implementation of a semi-autonomous mobile search and rescue robot: SALVOR," in *Proc. Int. Artif. Intell. Data Process. Symp. (IDAP)*, Sep. 2017, pp. 1–6.
[4] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 33, no. 3, pp. 367–385, Jun. 2003.
[5] A. Pfrunder, P. V. K. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2601–2608.
[6] D. Ramadasan, M. Chevaldonne, and T. Chateau, "Real-time SLAM for static multi-objects learning and tracking applied to augmented reality applications," in *Proc. IEEE Virtual Reality (VR)*, Mar. 2015, pp. 267–268.
[7] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019, *arXiv:1905.05055*. [Online]. Available: http://arxiv.org/abs/1905.05055
[8] S. Soetens, A. Sarris, and K. Vansteenhuyse, "Pose estimation errors, the ultimate diagnosis," *Eur. Space Agency, (Special Publication) ESA SP*, vol. 1, no. 515, pp. 181–184, 2002.
[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[10] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, no. 1, pp. 153–160, 2007.
[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[14] A. Garcia-Perez, F. Gheriss, D. Bedford, A. Garcia-Perez, F. Gheriss, and D. Bedford, "Going deeper with convolutions," in *Designing and Tracking Knowledge Management Metrics*. 2019, pp. 163–182.

[15] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 513–542, Apr. 2018.

[16] M. K. Al-Sharman, Y. Zweiri, M. A. K. Jaradat, R. Al-Husari, D. Gan, and L. D. Seneviratne, "Deep-learning-based neural network training for state estimation enhancement: Application to attitude estimation," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, pp. 24–34, Jan. 2020.

[17] V. Peretroukhin and J. Kelly, "DPC-net: Deep pose correction for visual localization," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2424–2431, Jul. 2018.

[18] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[19] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct monocular SLAM," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Lecture Notes in Computer Science), vol. 8690. 2014, pp. 834–849.

[20] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noel, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[21] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with CNNs for frame-to-frame ego-motion estimation," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 18–25, Jan. 2016.

[22] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[23] C. Cadena, A. Dick, and I. D. Reid, "Multi-modal auto-encoders as joint estimators for robotics scene understanding," in *Robotics: Science and Systems*, vol. 12. 2016.

[24] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, Jun. 2016, pp. 779–788.

[26] Y. Konishi, Y. Hanzawa, M. Kawade, and M. Hashimoto, *SSD: Single Shot MultiBox Detector*, vol. 1. Springer, 2016, pp. 398–413.

[27] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," 2018, *arXiv:1809.10790*. [Online]. Available: http://arxiv.org/abs/1809.10790

[28] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1722–1729.

[29] B. Mu, S.-Y. Liu, L. Paull, J. Leonard, and J. P. How, "SLAM with objects using a nonparametric pose graph," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4602–4609.

[30] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor localization of mobile robots through QR code detection and dead reckoning data fusion," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 6, pp. 2588–2599, Dec. 2017.

[31] P. Ozog and R. M. Eustice, "On the importance of modeling camera calibration uncertainty in visual SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3777–3784.

[32] J. H. Park, Y. D. Shin, J. H. Bae, and M. H. Baeg, "Spatial uncertainty model for visual features using a Kinect sensor," *Sensors*, vol. 12, no. 7, pp. 8640–8662, 2012.

[33] N. Sünderhauf *et al.*, "The limits and potentials of deep learning for robotics," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 405–420, Apr. 2018.

[34] J. Hidalgo-Carrio, D. Hennes, J. Schwendner, and F. Kirchner, "Gaussian process estimation of odometry errors for localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5696–5701.

[35] A. Rao and W. Han, "An adaptive Gaussian particle filter based simultaneous localization and mapping with dynamic process model noise bias compensation," in *Proc. IEEE 7th Int. Conf. Cybern. Intell. Syst. (CIS) IEEE Conf. Robot., Autom. Mechatronics (RAM)*, Jul. 2015, pp. 210–215.

[36] V. Magnago, L. Palopoli, R. Passerone, D. Fontanelli, and D. Macii, "Effective landmark placement for robot indoor localization with position uncertainty constraints," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 11, pp. 4443–4455, Nov. 2019.

[37] H. Liu, F. Sun, B. Fang, and X. Zhang, "Robotic room-level localization using multiple sets of sonar measurements," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 1, pp. 2–13, Jan. 2017.

[38] M. Zhang, X. Xu, Y. Chen, and M. Li, "A lightweight and accurate localization algorithm using multiple inertial measurement units," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1508–1515, Apr. 2020.

[39] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning Gaussian noise models for state estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1436–1443.

[40] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Trans. Intell. Vehicles*, early access, Mar. 13, 2020, doi: 10.1109/TIV.2020.2980758.

[41] M. Heshmat, M. Abdellatif, and H. Abbas, "Improving visual SLAM accuracy through deliberate camera oscillations," in *Proc. IEEE Int. Symp. Robotic Sensors Environments (ROSE)*, Oct. 2013, pp. 154–159.

[42] S. Chen and C. Chen, "Probabilistic fuzzy system for uncertain localization and map building of mobile robots," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 6, pp. 1546–1560, Jun. 2012.

[43] H. Hur and H. S. Ahn, "Unknown input H_∞ observer-based localization of a mobile robot with sensor failure," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 6, pp. 1830–1838, Dec. 2014.

[44] J.-W. Yoon and T. Park, "Maximizing localization accuracy via self-configurable ultrasonic sensor grouping using genetic approach," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 7, pp. 1518–1529, Jul. 2016.

[45] J. Toledo, J. Piñeiro, R. Arnay, D. Acosta, and L. Acosta, "Improving odometric accuracy for an autonomous electric cart," *Sensors*, vol. 18, no. 2, p. 200, Jan. 2018.

[46] M. Brossard and S. Bonnabel, "Learning wheel odometry and IMU errors for localization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 291–297.

[47] J. Czarnowski, T. Laidlow, R. Clark, and A. J. Davison, "DeepFactors: real-time probabilistic dense monocular SLAM," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 721–728, Apr. 2020.

[48] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[49] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3D reconstruction and tracking," in *Proc. 2nd Int. Conf. 3D Imag., Model., Process., Vis. Transmiss.*, Oct. 2012, pp. 524–530.

[50] O Wasenmüller and D. Stricker, "Comparison of kinect v1 and v2 depth images in terms of accuracy and precision," in *Proc. Asian Conf. Comput. Vis.*, vol. 1, 2017, pp. 277–289.

[51] B. Zoph and Q. V. Le, "Searching for activation functions," in *Proc. 6th Int. Conf. Learn. Represent., ICLR Workshop Track*, 2018, pp. 1–13.

[52] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[53] A. Koubaa, Ed., *Robot Operating System (ROS): The Complete Reference* (Studies in Computational Intelligence), vol. 3. Springer, 2018, p. 778.

[54] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Tech. Rep., 2012, pp. 1–26.

[55] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, to be published.

[56] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[57] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006. [Online]. Available: https://science.sciencemag.org/content/313/5786/504

[58] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://keras.io

[59] M. H. Law and J. T. Kwok, "Bayesian support vector regression," in *Proc. 8th Int. Workshop Artif. Intell. Statist.*, Key West, FL, USA, 2001, pp. 239–244.

Rana Azzam received the B.Sc. degree in computer engineering and the M.Sc. degree in research in electrical and computer engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree in electrical and computer engineering with a focus on robotics with the KU Center for Autonomous Robotic Systems (KUCARS).

Her current research interests include artificial intelligence and deep learning, and mobile robots simultaneous localization and mapping (SLAM).

Yusra Alkendi received the M.Sc. degree in mechanical engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2019, where she is currently pursuing the Ph.D. degree in aerospace engineering with a focus on robotics with the Khalifa University Center for Autonomous Robotics Systems (KUCARS).

Her current research is focused on the application of artificial intelligence (AI) in the fields of dynamic vision for perception and navigation.

Tarek Taha received the M.Eng. degree in computer control and the Ph.D. degree from the Centre of Excellence for Autonomous Systems (CAS), University of Technology Sydney, Ultimo, NSW, Australia, in 2004 and 2012, respectively.

He worked as a Senior Mechatronics Engineer in a Sydney-based engineering research and development company from 2008 to 2013, before joining Khalifa University, Abu Dhabi, United Arab Emirates, in 2014. He then led the Autonomous Aerial Lab, Algorythma, before joining the Dubai Future Foundation, Dubai, United Arab Emirates, to lead the Robotics Lab. His research interests include autonomous exploration, navigation, and mapping; machine vision; human–robot interaction; and assistive robotics and artificial intelligence.

Shoudong Huang (Senior Member, IEEE) received the bachelor's and master's degrees in mathematics and the Ph.D. degree in automatic control from Northeastern University, Shenyang, China, in 1987, 1990, and 1998, respectively.

He is currently an Associate Professor with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include mobile robots simultaneous localization and mapping (SLAM), exploration and navigation, and nonlinear system state estimation and control. He has published more than 150 articles in the robotics and control area.

Dr. Huang has been serving as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS and an Editor for the IEEE/RSL International Conference on Intelligent Robots and Systems (IROS) Conference Paper Review Board.

Yahya Zweiri (Member, IEEE) received the Ph.D. degree from the King's College London, London, U.K., in 2003.

He was involved in defense and security research projects in the last 20 years with the Defence Science and Technology Laboratory, King's College London, and the King Abdullah II Design and Development Bureau, Amman, Jordan. He is currently the School Director of the Research and Enterprise, Kingston University London, London, U.K. He is also an Associate Professor with the Department of Aerospace, Khalifa University, Abu Dhabi, United Arab Emirates. He has published over 100 refereed journal articles and conference papers and filed six patents in the USA and U.K. in the unmanned systems field. His research interests include interaction dynamics between unmanned systems and unknown environments by means of deep learning, machine intelligence, constrained optimization, and advanced control.