International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 4
ISSN: 2088-5334

Outdoor Localization of 4-Wheels for Mobile Robot Using CNN with 3D Data

Hanan A. Atiyah [a,*], Mohammed Y. Hassan [a]

[a] *Department of Control and Systems Engineering, University of Technology-Iraq, Baghdad, Iraq*
Corresponding author: [*]cse.19.03@grad.uotechnology.edu.iq

***Abstract*— One of the possible problems for a mobile robot is the localization. This is due to GPS systems' difficulty in detecting the location of a moving robot and the effects of weathering on sensors, such as the light sensitivity of RGBs sensors. In addition, mapping techniques in severe environments requires time and effort. This research seeks to enhance the localization of mobile robots by merging 3D LiDAR data with RGB-D images and using deep learning techniques. The suggested method entails using a simulator to design a four-wheel mobile robot controlled by a LiDAR sensor and testing them in an outdoor environment. The proposed localization system works in three steps. The first step is the training step, in which the 3D point cloud LiDAR sensor scans the entire city and then uses the PCA method to compress the dimensions of the 3D LiDAR data to a 2.5D image. The testing data stage is the second step. First, the RGB and depth images have merged using the IHS technique to create a 2.5D fusion image. Next, Convolution Neural Networks are used to train and test these datasets to extract features from the images. Finally, the K-Nearest Neighbor method was used in the third step. The classification step allows high accuracy while also reducing training time. The experimental findings show that the suggested technique is better in yielding results up to an accuracy of 98.15 % and a Mean Square Error of 0.25, and the Mean Error Distance is 1.36 meters.**

***Keywords*—CNN; HIS; K-NN; LiDAR; outdoor localization; pca; mobile robot.**

Manuscript received 30 Aug. 2021; revised 8 Oct. 2021; accepted 17 Jan. 2022. Date of publication 31 Aug. 2022.

I. INTRODUCTION

A key challenge in autonomous mobile robot navigation is mobile robot localization, which involves estimating the location and orientation of a mobile robot in an unknown environment [1]. Localization is critical in the autonomous behavior of mobile robots because a set of tasks must be completed, where a robot must properly recognize its position as it navigates [2]. With the growing worldwide interest in robotics, mobile robots may replace humans in various sectors due to their capabilities in doing tedious works. At the same time, they have a wide range of applications, including medical care, personal care, industrial automation, and monitoring [3]. However, events such as disease outbreaks have the potential to change timelines. For example, the coronavirus illness (COVID-19) breakout in 2020 accelerated the process and created new autonomous robots and automation opportunities in various industries [4].

Because of the availability of low-cost, compact, and high-resolution cameras, vision-based localization is one of the most common approaches in a mobile robot. A camera captures a wide range of information about the surroundings, including color, texture, and shape [5]. Changes in lighting in outdoor scenes are a severe issue for visibility since illumination is highly dependent on the environment (sunshine, rain, clouds, etc.). However, RGB cameras alone may not be enough. Current sensors, such as portable Light Detection and Ranging (LiDAR) and RGB-Depth cameras, supplement RGB images with depth information, opening up new options for developing robust and practical applications [6], [7]. For example, Li et al. presented a strategy for increasing the precision of pose prediction of 3D point clouds using LiDAR and Iterative Closest Point (ICP) by precisely segmenting the surface point and point cloud. However, in challenging environments such as complex environments and highways, ICP cannot achieve accurate point-cloud registration efficiency, which is limited to meeting real-time requirements, mainly when dealing with large-scale point-cloud data, and computation is an expensive phase in the ICP algorithm [8].

There has been an increase in interest and effort in developing Deep Learning (DL) approaches for robotics systems that use computer vision. Atsuzawa et al. [9]

employed two suggested methods, the odometer-based and visual-assisted algorithms by Convolution Neural Network CNN. However, many factors can cause high orientation errors because the proposed localization algorithms were created and evaluated using small data. It is also probable that CNN's architecture and preparation choices are not yet well-suited to using the dataset for translation. To conduct successful robot localization, Sinha et al. suggested a CNN architecture that employs an Extended Kalman Filter (EKF). The existing framework has several flaws in integrating information from a single monocular camera's pictures with low-cost sensors from a mobile robot. It becomes more difficult to accurately estimate the robot's position in situations with repetitive sceneries or visual elements. Also, position regression performance drifts a lot during sharp corners [10].

Some approaches failed to properly categorize a mobile robot's position because of changing weather circumstances such as rain and snow. For example, Debeunne and Vivet [11] offered the Visual-LiDAR Simultaneous Localization and Mapping SLAM of a highly accurate environment of 3D information, relatively few research deals with 3D details. However, a disadvantage for Visual-LiDAR SLAM is that it does not work well under rain or in textured environments. Furthermore, mapping-based methods require high computational costs in complex environments such as complex highways [12].

Specific techniques fail to accurately determine the mobile robot's position due to various weather conditions such as rain and snow. In addition to the fact that some sensors, such as RGB, do not perform well in an outside setting because they are light-sensitive. The SALM technique and other systems cost time and effort to localize mobile robots in challenging environments, complicated surroundings, and roads. In the field of DL, if the CNN design isn't proper, it affects the robot's location accuracy.

As a basis, a proposed system based on 3D data from LiDAR and RGB-D with a Deep Learning algorithm to achieve precision and resilience in determining the right position of the mobile robot is suggested. The proposed work has three stages training, testing, and classification. These are just a few of the tasks that make up each level. The Principal Component Analysis (PCA) method transforms a 3D LiDAR point cloud scan into a 2.5D image during the training stage. To extract features from the 2.5D pictures, the (CNN) method has been employed. Image fusion has been accomplished at the testing stage by combining RGB and Depth (D) images into a single RGB-D image and then extracting features from the RGB-D image using CNN. Finally, the tested image has been classified using the K-Nearest Neighbors algorithm in the classification stage to find the position of the mobile robot.

II. Materials and Methods

A. The Training Stage Uses a LiDAR Sensor with PCA Method

LiDAR produces 3D models and maps of structures and surroundings to train the deep learning network and acquire the 3D dataset. As signals bounce off surfaces and return to the scanner, LiDAR identifies the shape. After processing and arranging the individual signals, LiDAR data produces point cloud data. 3D elevation points with X, Y, and Z coordinates comprise the point clouds [8]. However, rain, fog, dust, air particles, light scattering, and other influencing variables would often negatively affect signals generated by LiDAR sensors during transmission, producing noise in point cloud pictures. To solve this issue, a proposed noise reduction approach based on PCA has been developed to filter LiDAR point clouds [13].

PCA is a technique for extracting data's major feature components. The noise is frequently connected to the eigenvector corresponding to the weakest eigenvalue and eliminating this eigenvector can aid with noise reduction. Assuming that there are m 3D data points, they are first organized into a matrix $\boldsymbol{\xi}$ with three rows and m columns. By subtracting the means, each row of the matrix $\boldsymbol{\xi}$ is zero-centered, ensuring that the average value of each row is zero. The covariance matrix C is then obtained [14]:

$$C = \frac{1}{m}\xi\xi^T \tag{1}$$

The following equation must be completed once three eigenvectors are placed in the matrix E=(e1,e2,e3) [14]:

$$E^T C E = \Lambda \tag{2}$$

Λ is a diagonal matrix. As a result, three eigenvectors and their eigenvalues have been discovered. The eigenvectors have been placed in a matrix in decreasing eigenvalue rows, with the first two rows of eigenvectors being used to form matrix P. The third row has been deleted since it offers minimal information. The original 3D data has transformed to 2.5D using the following formula [14]:

$$Y_{2\times m} = P_{2\times 3} \times \xi_{3\times m} \tag{3}$$

These data points are then projected onto a new axis, the U-V coordinate system. The Z-values that represents the height or depth utilizing the 3D point cloud have been observed in the point cloud. Positive Z-value points are visible on the map, whereas negative Z-value points are invisible belowground. The PCA technique is used to rotate about the Z-axis, coordinate the point cloud around the X-axis, and eliminate negative Z-values in this case.

B. Testing Stage Using RGB-D Sensors and Intensity Hue Saturation (IHS)

Many robotic systems are now employing inexpensive RGB-D sensors. The depth data, in particular, offers extra information on object shape and is unaffected by changes in lighting or color. As a result, it may improve the performance of mobile robots in the challenging task of object detection. In addition, image fusion is a method for extracting valuable data from several input images and merging it into a new output image that is more descriptive and helpful than the input images alone. For example, the IHS color is created by mathematical formulae to transform RGB values into matching points [15]:
The Hue H is given by:

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases} \tag{4}$$

Where:

$$\theta = cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\} \quad (5)$$

The Saturation S:

$$S = 1 - \frac{3}{(R+G+B)}[min(R,G,B)] \quad (6)$$

The Intensity I is given by:

$$I = \frac{1}{3}(R+G+B) \quad (7)$$

C. Convolution Neural Network CNN

The CNN is a deep learning approach that combines artificial neural networks with deep learning techniques [16]. Various image detection tasks have been performed using this artificial neural network. CNN has lately caught the attention of experts worldwide, as it has extreme performance in a range of computer vision and machine learning challenges.

Convolution, pooling, and fully linked layers are the three types of layers that make up CNNs [17]. Convolution and pooling are the first two layers that extract features, whereas the third, a fully connected layer, transfers specific characteristics into the final output, such as classification. A convolution layer is a component of CNN that consists of a series of mathematical operations such as convolution, a type of linear operation [18].

A convolution, polling, and a fully connected layer were used to design a robust CNN network in extracting characteristics from images. It consists of 12 layers using classifier K-Nearest Neighbors in the last classification layer sensor to choose the correct position of the mobile robot, where Network design details is explained in Section II-K.

D. Classification Stage by K-Nearest Neighbors

The K-NN classification is one of the most commonly used distance-based techniques that evaluate the distances between the testing and the training data to determine the final classification performance. The Euclidian space calculates the distance between the test sample and all training samples in the normal K-NN classification method [19] :Euclidean Distance between two points [20].

$$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (8)$$

E. Open Source Webots Simulator

Cyberbotics Ltd developed Webots, an open-source three-dimensional mobile robot simulator [21]. Webots allows users to build dynamic landscapes for mobile robot simulations. The controller's activities in Webots are based on a continuous process. The inputs given by the sensors are gathered and then processed to create outputs delivered to the actuators at each iteration [22]. A mobile robot must promptly manage uncertain and inaccurate, or incomplete, information about the surroundings to travel safely in an unfamiliar area. A controller must be assigned to the robot to perform the specified behavior. The controller is programmed in various languages, including C, C++, Java, Python, and Matlab, and the source code is displayed in a window in the Webots simulator interface and can be modified [22].

F. Suggested Outdoor Localization System

To increase the efficiency and reliability of a localization system, the suggested technique uses 3D data to identify the position of a mobile robot. Fig.1 demonstrates how the proposed system collects the dataset for training and testing using two sensors (LiDAR and RGB-D) placed on a mobile robot. This paper proposed a technique that splits into three stages. These are training, testing, and classification stages. The LiDAR sensor performs a scan to acquire 3D point cloud data during training. Using the PCA technique, a 3D point cloud is converted to a 2.5D picture. The feature is extracted from 2.5D images using the CNN method. It uses a matrix format to contain all featured data, point cloud data, and pre-processing data. It employs the RGB and Depth sensors to obtain two pictures of the same place during the testing level. Merge two RGB and Depth (D) into an RGB-D fusion image using the IHS technique to generate a single 2.5D image. The features are retrieved from a 2.5D RGB-D picture using the CNN method. All of the feature data is then arranged in a matrix. To identify the proper position of the mobile robot, the test data is categorized with the training data stored in the classification level of the K-NN classifier. Its third stage will reduce training time and give high classification accuracy because this algorithm does not need training, so it is fast in locating the robot.

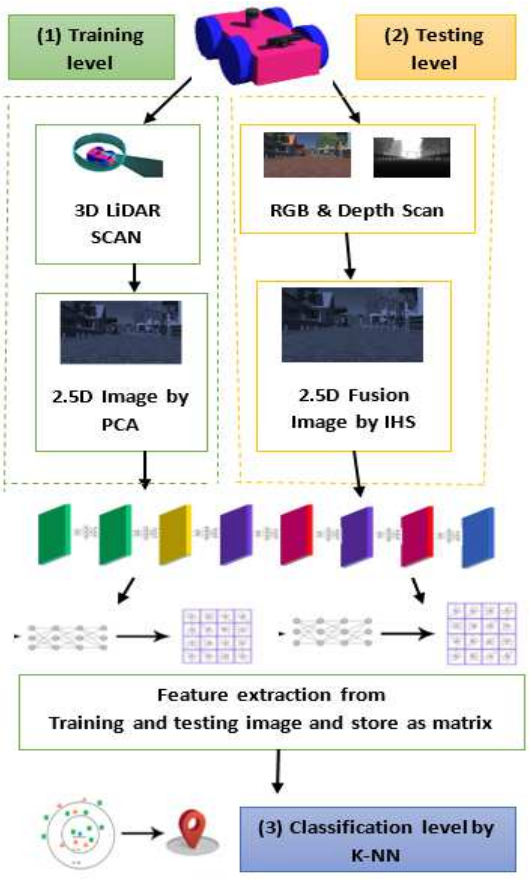


Fig. 1 Suggested outdoor localization system

G. Design proposed outdoor mobile robot

To realize efficient and smooth navigation for a mobile robot in an unfamiliar environment, a robot control algorithm based on the Braitenberg vehicle concept is developed. The robot has a box-shaped body, four cylindrical wheels, four

rotation motors, two sensors Kinect, and 360 Lidar. The details of the design dimensions are shown in Fig. 2.

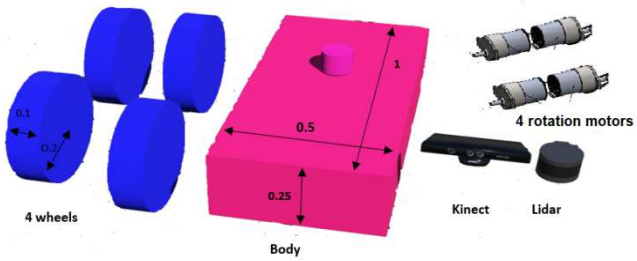


Fig.2 Designed Mobile robot with details in Webots simulator (open-source simulator [21])

H. Obstacle Avoidance Control by Lidar

The goal is to build a navigation system for the mobile robot based on the Braitenberg vehicle, allowing it to navigate in an unfamiliar area and avoid obstacles by replicating the Braitenberg 3b Vehicle's behavior [22]. The tuning technique is based on sensor data from the robot's Lidar sensor for this purpose. The angle coordinates and the object distance acquired from the sensing device are utilized as references in this system. However, there must be some sensor data processing because of the nature of the stimulus and detecting hardware between sensing and motor actuation [23].

After finishing setting up the world scene and the four wheels' mobile robot, there is a need to create the controller code to simulate the behavior of a Braitenberg 3b vehicle. Fig.3 explains the calculation of cumulative obstacles for both left and right sides. A specific threshold (d > 1m) is applied to the measured range values, and particular weights are assigned to the measured range values. In these calculations, obstacles farther than a certain threshold (d > 1m) are not considered [22].

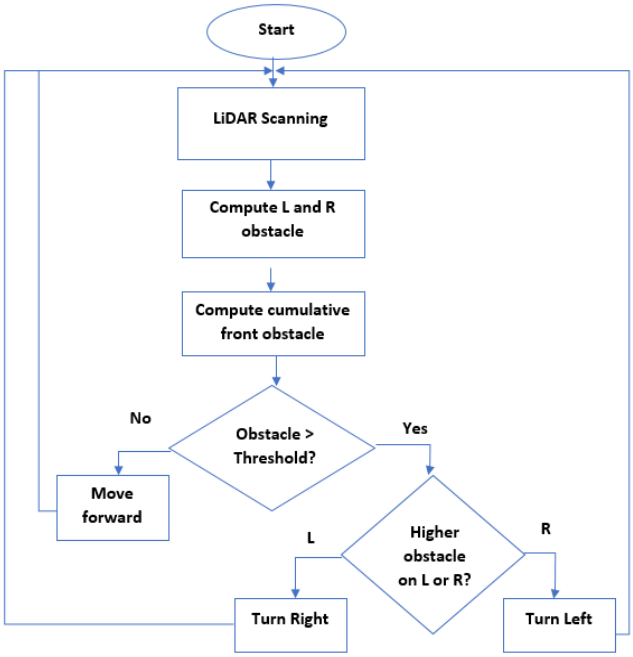


Fig. 3 Collision-avoidance algorithm flowchart [22]

I. Design for Outdoor Environment

An outdoor environment is built on the Webots simulation platform, where many items are placed, including buildings, trees, and a mobile robot, as seen in Fig. 4. The design area is about 10,000 m^2.

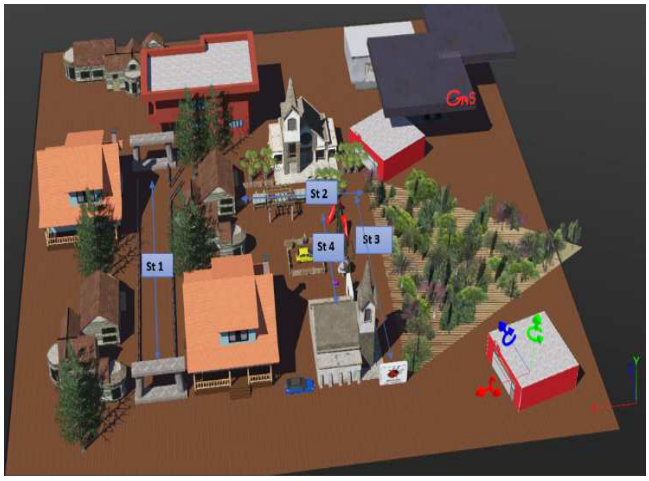
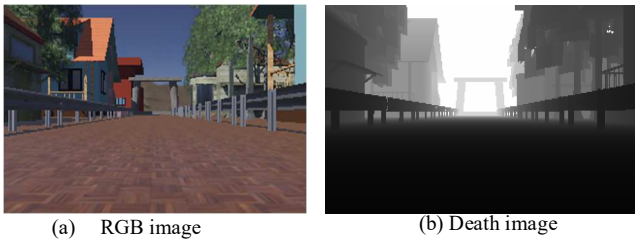


Fig. 4 Top view of outdoor proposed design environment with four streets by Webots simulator (open-source simulator [21])

The LiDAR sensor must scan the region in front of the robot, spanning a 360° horizontal field of view. Other technical parameters, such as the minimum and maximum ranges, the number of transmitted beams, and the sensor resolution, are initialized to mimic the characteristics of a simple device. Four streets shown in Fig. 4 have been identified where the robot is moving to collect data on these streets.

J. Collecting the Dataset

In most cases, supervised learning data is required to train a neural network. A dataset containing RGB images, depth images, and high-resolution LiDAR scans with associated mode designations is required to train and test the designed device. A mobile robot can be equipped with a Kinect sensor that captures images at a predefined frame rate. As shown in Fig. 5, the Kinect sensor provides RGB and depth pictures.



(a) RGB image (b) Death image

Fig. 5 Explain the RGB image and the Depth image (These images were saved when we ran the open-source simulation [21])

There were two sets of training data, each with 200 RGB and Depth images and 100 LiDAR frames, where the data was recorded and saved when the robot moved every minute by a Python program.

K. Design CNN Localization

Localization CNNs have 12 layers. Therefore, 12-layer CNN with 224 ×224 input pixels was structured for brief training to improve precision and minimize error. As a result, this network was optimized using Stochastic Gradient Descent with Momentum (SGDM). Fig. 6 and Table I analyzed the network design.

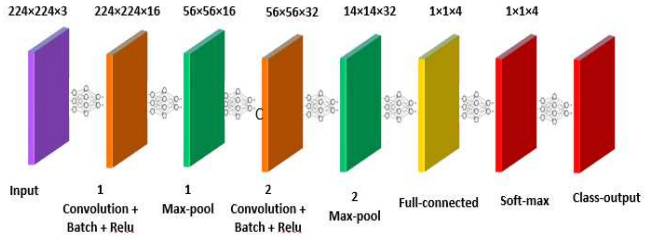


Fig. 6 Twelve layers and a K-NN classifier in CNN localization

TABLE I
ANALYZED NETWORK FOR CNN DESIGN

	Name	Activations	Learnables
1	Image input 224×224×3 image with zero center normalization	224×224×3	-
2	CONV_1 16 3×3×2 convolutions with stride [1 1] and padding [1 1 1 1]	224×224×16	Weights 3×3×3×16 Bias 1×1×16
3	Batchnorm_1 Batch normalization with 16 channels	224×224×16	Offset 1×1×16 scale 1×1×16
4	Relu_1	224×224×16	-
5	Maxpool_1 4×4 max pooling with stride [4 4] and padding [0 0 0 0]	56×56×16	-
6	CONV_2 32 3×3×16 convolutions with stride [1 1] and padding [1 1 1 1]	56×56×32	Weights 3×3×16×32 Bias 1×1×32
7	Batchnorm_2 Batch normalization with 32 channels	56×56×32	offset 1×1×32 scale 1×1×32
8	Relu_2	56×56×32	-
9	Maxpool_2 4×4 max pooling with stride [4 4] and padding [0 0 0 0]	14×14×32	-
10	FC 4 fully-connected layer	1×1×4	Weights 4×6272 Bias 4×1
11	Softmax	1×1×4	-
12	Class output	-	-

For training, a PC with Intel(R) Corei5-8250U CPU running at 1.60GHz and UHD Graphics 620 with 8GB of RAM is used, and the code of the program is written in MATLAB. The 100 LiDAR frames were processed throughout four epochs. Three iterations in each epoch, with a maximum of 12 iterations and a learning rate of 3*10-4. The data is divided into 70% for training and 30% for testing. This achieves 98.15% accuracy, as shown in Fig. 7.

To calculate the error in distance, Mean Error of Distance (MED) and Mean Square Error (MSE) is used. The error of Distance (ED) is defined as the difference between the estimated and actual sums S* and S [24]:

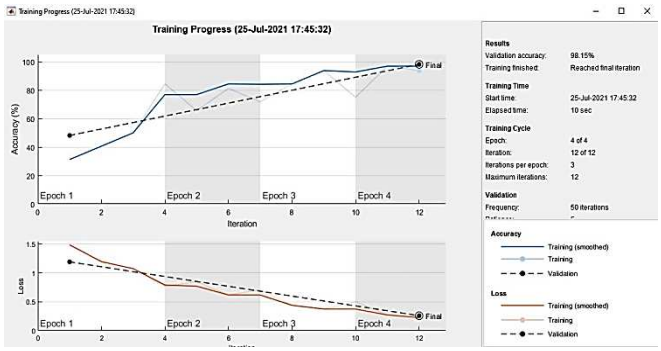


Fig. 7 Training progress with details of accuracy result

$$ED = |S^* - S| \tag{9}$$

The average of all error distances is the (MED). The total error distances are computed using the Mean Squares of Error (MSE) method [24]:

$$MED = E[ED] = \sum_{ED_i \in \Omega} ED_i P(ED_i) \tag{10}$$

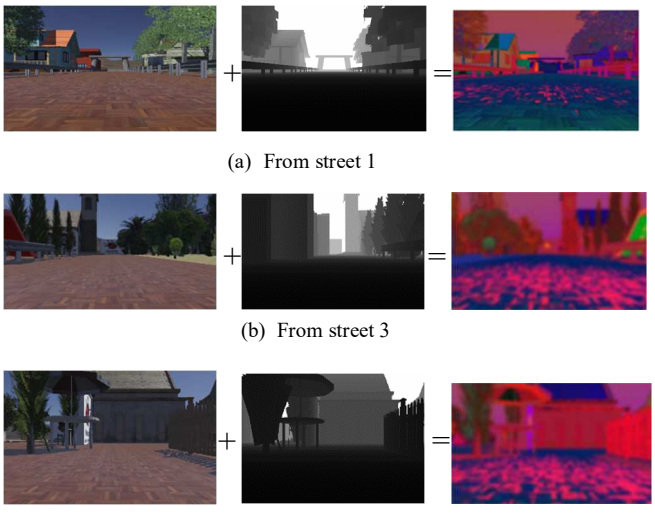
$$MSE = E[ED^2] = \sum_{ED_i \in \Omega} ED_i^2 P(ED_i) \tag{11}$$

Where Ω is the set of all errors.

III. RESULT AND DISCUSSION

A. Testing results by IHS Transformation

The IHS method has an influence when merging RGB and Depth images. An individual pixel's intensity fluctuates with its brightness in a depth image. Fig. 8 illustrates how Hue, Saturation, and Intensity are mixed in the 2.5d fusion image.



(a) From street 1

(b) From street 3

(c) From street 4

Fig. 8 Image created by combining RGB and Depth images (photos from open-source simulator [21])

B. Evaluation Results

A high-quality and accurate localization sample that can be compared with reality is required to test the performance of the PCA technique utilizing the K-NN algorithm. Four instances were tested at random places throughout the city, the correct location was known, and the Mean Square Error MSE calculated as an equation can be evaluated [25]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(\hat{\theta}_i - \theta_i\right)^2 \tag{12}$$

and are shown in Table II.

TABLE II
MSE FOR THE TEST IMAGES

Case	Predict the correct location	MSE
1	Street 1	0.3519
2	Street 2	0.2222
3	Street 3	0.3148
4	Street 4	0.1111
	Mean error	0.25

To calculate MED according to equation (10 and 11), Webots simulator recorded 60 Frame Per Second (FPS), then MED = 1.36 meter. Several research methodologies are compared to the proposed method, as listed in Table III.

TABLE III
COMPARISON BETWEEN THE PROPOSED METHOD WITH OTHER METHODS

Data	Algorithms	Accuracy	Mean Error	Amount of data
RGB + GPS+ Compass [26]	Faster R-CNN & FFNN	Not mention	28.47 m	1625 data
RGB-D and Conditional Random Field (CRF) [27]	ORB-SLAM2	91.2%	4.41 m	500 3D object models
Proposal work RGB-D & LiDAR	CNN + PCA +KNN	98.15 %	1.36 m	300 data

It can be noticed that the proposed method achieved high accuracy with less training time in addition to a low average error compared to the two methods mentioned in Table III. Even though the dataset included 100 LiDAR frames, the accuracy was 98.15%, MSE was 0.25, MED 1.36 m, utilizing the PCA technique and the K-NN classifier, and the IHS to correct the localization for a mobile robot. The K-NN classifier's findings show that it takes less time to train with higher accuracy and lower error rates.

IV. CONCLUSION

The problem of robot location loss in the outside environment is addressed by the mobile robot localization system presented in this work. The location is determined incorrectly due to a variety of elements in the environment that influence the sensors attached to the robot. Thus, the utilization of 3D sensors obtains better precision using Deep Learning algorithms. The three stages of the intended architecture are training, testing, and classification. PCA is used to decrease dimensions and rotate the point cloud 3D LiDAR, and the IHS technique is used to create the 2.5D RGB-D fusion signal image. The K-NN method provided high-accuracy results in a short amount of time. Findings are improved with 98.15% accuracy, MSE of 0.25, and MED equal to 1.36 m.

REFERENCES

[1] R. T. Kamil, M. J. Mohamed, and B. K. Oleiwi, "Path Planning of Mobile Robot Using Improved Artificial Bee Colony Algorithm," Eng. Technol. J., vol. 38, no. 9A, pp. 1384–1395, 2020, doi: 10.30684/etj.v38i9a.1100.

[2] X. Bai, Z. Zhang, L. Liu, X. Zhai, J. Panneerselvam, and L. Ge, "Enhancing localization of mobile robots in distributed sensor environments for reliable proximity service applications," *IEEE Access*, vol. 7, pp. 28826–28834, 2019, doi: 10.1109/ACCESS.2019.2899059.

[3] M. Y. Hassan and M. Z. A. Karam, "Design and Implementation of Rehabilitation Robot for Human Arm Movements," *IRAQI J. Comput. Commun. Control Syst. Eng.*, vol. 15, no. 2, 2015.

[4] N. A. K. Zghair and A. S. Al-Araji, "A one decade survey of autonomous mobile robot systems," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 4891–4906, 2021.

[5] T. D. Dung, D. Hossain, S. ichiro Kaneko, and G. Capi, "Multifeature image indexing for robot localization in textureless environments," *Robotics*, vol. 8, no. 2, pp. 1–11, 2019, doi: 10.3390/ROBOTICS8020037.

[6] N. C. Mithun, K. Sikka, H.-P. Chiu, S. Samarasekera, and R. Kumar, "Rgb2lidar: Towards solving large-scale cross-modal visual localization," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 934–954.

[7] X. Kang, J. Li, X. Fan, and W. Wan, "Real-Time RGB-D Simultaneous Localization and Mapping Guided by Terrestrial LiDAR Point Cloud for Indoor 3-D Reconstruction and Camera Pose Estimation," *Appl. Sci.*, vol. 9, no. 16, p. 3264, 2019.

[8] X. Li, S. Du, G. Li, and H. Li, "Integrate point-cloud segmentation with 3d lidar scan-matching for mobile robot localization and mapping," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010237.

[9] K. Atsuzawa, S. Nilwong, D. Hossain, S. Kaneko, and G. Capi, "Robot navigation in outdoor environments using odometry and convolutional neural network," In: IEEJ international workshop on sensing, actuation, motion control, and optimization (SAMCON) 2019.

[10] H. Sinha, J. Patrikar, E. G. Dhekane, G. Pandey, and M. Kothari, "Convolutional Neural Network Based Sensors for Mobile Robot Relocalization," in *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, 2018, pp. 774–779.

[11] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[12] S. Oishi, Y. Inoue, J. Miura, and S. Tanaka, "SeqSLAM++: View-based robot localization and navigation," *Rob. Auton. Syst.*, vol. 112, pp. 13–21, 2019.

[13] Y. Duan, C. Yang, H. Chen, W. Yan, and H. Li, "Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction," *Opt. Commun.*, vol. 482, no. 2019, 2021, doi: 10.1016/j.optcom.2020.126567.

[14] Y. Duan, C. Yang, H. Chen, W. Yan, and H. Li, "Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction," *Opt. Commun.*, vol. 482, p. 126567, 2021.

[15] C.-B. Hsu, J.-C. Lee, and T.-M. Tu, "Generalized IHS-BT framework for the pansharpening of high-resolution satellite imagery," J. Appl. Remote Sens., vol. 12, no. 04, p. 1, 2018, doi: 10.1117/1.jrs.12.046008.

[16] S. Agustin, H. Tjandrasa, and R. V. H. Ginardi, "Deep Learning-based Method for Multi-Class Classification of Oil Palm Planted Area on Plant Ages Using Ikonos Panchromatic Imagery," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 10, no. 6, p. 2200, 2020, doi: 10.18517/ijaseit.10.6.12030.

[17] A. A. Abdulhussein and F. A. Raheem, "Hand Gesture Recognition of Static Letters American Sign Language (ASL) Using Deep Learning," *Eng. Technol. J.*, vol. 38, no. 6A, pp. 926–937, 2020, doi: 10.30684/etj.v38i6a.533.

[18] L. Alzubaidi, J. Zhang,A. Humaidi,A. Al-Dujaili,Y. Duan, O. Al-Shamma, ... & L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", vol. 8, no. 1. Springer International Publishing, 2021.

[19] H. Naser, K. Al-behadili, and K. R. Ku-mahamud, "Hybrid K-Nearest Neighbour and Particle Swarm Optimization Technique for Divorce Classification," Int. J. Adv. Sci. Eng. Inf. Technol.vol. 11, no. 4, pp. 1447–1454, 2021.

[20] X. Gao and G. Li, "A KNN Model Based on Manhattan Distance to Identify the SNARE Proteins," IEEE Access, vol. 8, pp. 112922–112931, 2020, doi: 10.1109/ACCESS.2020.3003086.

[21] "Webots: robot simulator." https://cyberbotics.com/#cyberbotics (accessed Aug. 24, 2021).

[22] G. Belingardi, A. Valle, M. P. Cavatorta, and I. Singh, "Bilby Rover autonomous mobile robot in the context of Industry 4.0," PhD Thesis. Politecnico di Torino, 2021.

[23] N. Ahmed and W. J. Teahan, "Using Compression to Discover Interesting Behaviours in a Hybrid Braitenberg Vehicle," *IEEE Access*, vol. 9, pp. 11316–11327, 2021, doi: 10.1109/ACCESS.2021.3050882.

[24] Y. Wu, Y. Li, X. Ge, Y. Gao, and W. Qian, "An Efficient Method for Calculating the Error Statistics of Block-Based Approximate Adders," *IEEE Trans. Comput.*, vol. 68, no. 1, pp. 21–38, 2019, doi: 10.1109/TC.2018.2859960.

[25] T. S. Mahmoud, D. Habibi, M. Y. Hassan, and O. Bass, "Modelling self-optimised short term load forecasting for medium voltage loads using tunning fuzzy systems and Artificial Neural Networks," *Energy Convers. Manag.*, vol. 106, no. December, pp. 1396–1408, 2015, doi: 10.1016/j.enconman.2015.10.066.

[26] S. Nilwong, D. Hossain, S. I. Kaneko, and G. Capi, "Deep learning-based landmark detection for mobile robot outdoor localization," *Machines*, vol. 7, no. 2, 2019, doi: 10.3390/machines7020025.

[27] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, no. c, pp. 75545–75559, 2018, doi: 10.1109/ACCESS.2018.2873617.