Autonomous Vehicle Localization System for University Applications

1st Duc-Anh Le
Department of Automation, Faculty of Electrical & Electronics Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Vietnam.
anh.leanhle44@hcmut.edu.vn

2nd Huy-Hoang-Anh Tran
Department of Automation, Faculty of Electrical & Electronics Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Vietnam.
anh.tranhuyhoang03@hcmut.edu.vn

3rd Ngoc-Huy Tran
Department of Automation, Faculty of Electrical & Electronics Engineering
Ho Chi Minh City University of Technology
Ho Chi Minh City, Vietnam.
tnhuy@hcmut.edu.vn

***Abstract*—Autonomous vehicle localization is a promising research area with significant potential in transportation technology. This study explores the integration of 2D lidar sensors and depth cameras to enhance the mapping and localization capabilities of autonomous vehicles in university campus environments. While 2D lidar is commonly used for mapping and positioning, its data density limitations can hinder accuracy in outdoor settings, particularly in areas with vegetation. By combining depth cameras, we address these limitations and improve the quality of lidar data, providing richer depth information. This research develops a robust localization system that leverages the strengths of both sensor types, aiming for better situational awareness and safer navigation for autonomous vehicles on campus. We will validate the effectiveness of the integrated approach through experimental studies in real-world environments.**

Keywords— Depth camera, Mapping, Navigation, Lidar 2D.

I. INTRODUCTION.

In the context of the rapid development of autonomous vehicle technology, research on localization and environmental perception has become increasingly urgent. Autonomous vehicle systems not only promise to improve traffic efficiency and optimize transportation processes but also contribute to the safety and sustainability of mobility solutions.

To achieve this goal, integrating several types of sensors is crucial. Currently, 2D lidar is one of the most used technologies for collecting environmental data due to its ability to provide accurate and detailed distance information. However, 2D lidar faces certain limitations in outdoor scenarios, particularly in complex areas with structures such as trees, tall buildings, and other noise factors. These elements can affect the reliability and accuracy of localization data.

To address these limitations, integrating depth cameras with 2D lidar has emerged as a promising approach. Depth cameras provide rich depth information, allowing for the creation of more detailed 3D maps and enhancing object recognition capabilities in complex environments.

This study aims to develop an effective localization system for autonomous vehicles by combining 2D lidar and depth cameras, particularly in university environments. This system not only ensures accurate localization but also offers numerous practical applications, such as security, food delivery, and transportation services. The research will conduct experiments to demonstrate the feasibility and effectiveness of integrating 2D lidar and depth cameras, contributing to the development of safe and efficient autonomous vehicle applications in university settings.

II. HARDWARE SPECIFICATIONS

The autonomous vehicle positioning system in this study is built upon the integration of two primary sensor types: 2D lidar and a depth camera, aimed at enhancing accuracy and efficiency in mapping and environmental localization.

A. Lidar 2D

The lidar used in the system is the Hokuyo UTM-30LX, one of the most popular 2D laser scanning sensors available today.

TABLE I. MAIN SPECIFICATIONS OF HOKUYO UTM-30LX

Specification	Value
Operating Range	0.1 – 30 (m)
Scanning Angle	270 deg
Angular Resolution	0.25 deg
Scanning Frequency	40 Hz

The Hokuyo lidar provides precise information regarding the distance and shape of objects in the environment, aiding the autonomous vehicle system in determining its location and identifying obstacles [1]. However, in outdoor environments, particularly where there is dense vegetation or complex terrain, 2D lidar may encounter limitations in distinguishing between objects at varying heights.

B. Depth camera

To supplement the limitations of the lidar, the system employs the Microsoft Xbox 360 Kinect depth camera. Originally designed for gaming applications, Kinect has gained popularity in the robotics field due to its low-cost depth data acquisition capabilities [2].

TABLE II. MAIN SPECIFICATIONS OF KINECT

Specification	Value
Operating Range	0.5 – 4.5 (m)

Specification	Value
Infrared Sensor	Yes (for depth measurement)
RGB Camera	Yes (for capturing color images)
Resolution	640x480 pixels
FPS	30

Kinect has the capability to generate 3D mapping data, allowing the positioning system to differentiate between objects at varying heights and recognize more complex environments. Notably, in outdoor situations or areas with intricate terrain, such as university campuses, Kinect enhances the quality of the maps produced by the 2D lidar.

III. ALGORITHM

A. *PointCloud data fusion algorithm*

In this algorithm, the first step involves processing the input data. The data obtained from the lidar is in the form of a 2D PointCloud, which contains the (x, y) coordinates of the scanned points along with distance information. However, since it operates solely on a horizontal plane, the lidar does not provide height information for objects in the environment, leading to a lack of data in outdoor settings or in areas with complex terrain. In contrast, the data from the depth camera includes the (x, y, z) coordinates of points in space, as well as color values from the integrated RGB camera. However, Kinect has a shorter operating range and encounters limitations when collecting data in strong outdoor lighting conditions. To effectively integrate data from these two sensors, synchronization and data merging steps are necessary.

The first step is Temporal Synchronization. Both the lidar and Kinect operate at different scanning frequencies (lidar: 40 Hz, Kinect: 30 Hz). Therefore, the data from both sensors need to be synchronized before merging. This is achieved through the time synchronization system in ROS (Robot Operating System), ensuring that the PointClouds from the lidar and Kinect are recorded at the same moment in time.

The second step is Coordinate Transformation. Since the lidar and Kinect are mounted at separate locations (relative to the base_link), each sensor collects data based on its own coordinate system. Transforming the coordinate systems (using a 4x4 homogeneous transformation matrix) between the two sensors is necessary to bring all data points into a common coordinate system, typically the vehicle frame coordinate system (base_link).

The general formula for the transformation between two coordinate systems is:

$$P_{global} = T_{base\_to\_sensor} * P_{sensor} \quad (1)$$

P_{sensor} refers to the coordinates of the data point from the sensor, $T_{base\_to\_sensor}$ is the transformation matrix from the sensor's coordinate system to the common coordinate system of the vehicle, and P_{global} is the point's coordinates after being transformed into the common coordinate system.

Finally, there is the process of merging the PointCloud data from two sensors based on applying alignment algorithms (registration) to ensure that the data from the lidar and Kinect camera are accurately aligned. Below are the algorithms used for this purpose.

The first is the Iterative Closest Point (ICP). ICP is a common alignment algorithm for merging two PointCloud datasets based on the closest distance between points in the two clouds. ICP optimizes the transformation, including the rotation matrix R and the translation vector T, such that the distance between corresponding points in the two datasets is minimized. The objective function of ICP is expressed as follows:

$$E = min \sum_{i=1}^{n} \left\| p_i - (R_{q_i} + T) \right\|^2 \quad (2)$$

where p_i and q_i are corresponding points in the two point clouds, and n is the number of points in the dataset [3].

The second is the Generalized ICP (GICP) algorithm. This is an extension of ICP to improve accuracy when handling data with noise and different resolutions. GICP not only relies on finding the nearest point pairs but also utilizes information about the local structure of the surface, through covariance matrices Σ_p và Σ_q [4]. The objective function of GICP is expressed as follows:

$$E = min \sum_{i=1}^{N} (p_i - (R_{q_i} + T))^T (\Sigma_p + R\Sigma_q R^T)(p_i - (R_{q_i} + T) \quad (3)$$

The third algorithm is Normal Distributions Transform (NDT). NDT divides the 3D space into small cells (often represented as a voxel grid) and models each cell using a Gaussian distribution (characterized by its mean μ and covariance Σ). When applying NDT to register two sets of points, this method minimizes the distance between the points in the transformed point set and the Gaussian distribution of the cells in the reference point set. The optimization is performed by minimizing the objective function:

$$E = -min \sum_{i=1}^{m} \log(P(q_i | R(p_i) + T)) \quad (4)$$

where P denotes the probability density function of the Gaussian distribution associated with the cells in the reference point set, evaluated at the transformed points $R(p_i) + T$. [5]

Additionally, there are several other methods that can be used in specific situations, such as Go-ICP, Super4PCS, FGR, and CPD, but they are not as commonly used.

B. *The Full SLAM (Simultaneous Localization and Mapping) model*

The SLAM problem utilizes probabilistic algorithms to estimate the trajectory of an Automated Guided Vehicle (AGV) and construct a real-time map of the surrounding environment.

Assuming the AGV operates in an unknown environment, let t be the time under consideration, with $0 \le t \le T$. At this point in time, the position of the AGV is denoted as x_t. When the AGV moves on a plane, x_t can be described by a three-dimensional vector comprising two coordinate values of the AGV in 2D space and one rotational value. The set of AGV positions is represented as follows:

$$X_t = \{x_0, x_1, x_2, \dots, x_T\} \quad (4)$$

where x_o is the known initial position of the AGV, and T is the endpoint of the computation process.

Let u_t be the characteristic value (relationship) for the motion between two positions of the AGV or a sequence of geometric measurements obtained during the movement. The set of u_t is defined as:

$$U_t = \{u_1, u_2, u_3, \dots, u_T\} \tag{5}$$

Let z_t be the characteristic value representing location information through observation measurements from the environment m. The set of z_t is represented as:

$$Z_t = \{z_1, z_2, z_3, \dots, z_T\} \tag{6}$$

The SLAM model is categorized into two types: full SLAM and online SLAM. The online SLAM model computes the location and map data only at the current moment without retaining the entire movement process. Conversely, full SLAM focuses on computing the entire trajectory from the start to the current moment of the AGV, along with the environment's map. Specifically, the full SLAM model determines the set of all positions X_t and constructs the map of the environment mmm through geometric measurements U_t and environmental data Z_t. This is expressed as:

$$p(X_{0:T}, m | Z_{1:T}, U_{1:T}) \tag{7}$$

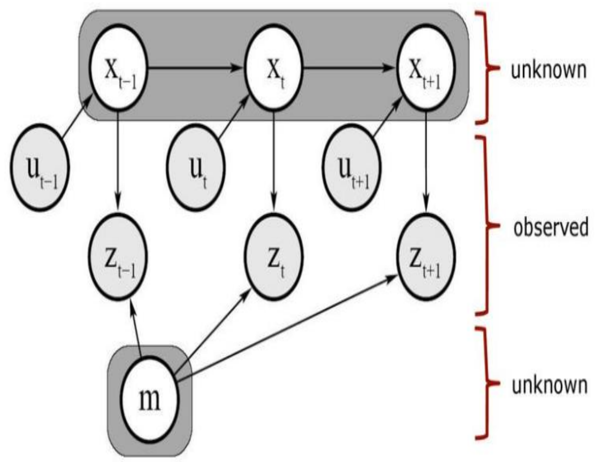

Fig. 1. The full SLAM model.

Thus, the full SLAM model is suitable for the localization problem outlined above.

C. AMCL (Adaptive Monte Carlo Localization)

AMCL is one of the most used localization algorithms in mobile robotic systems. This algorithm combines the Monte Carlo Localization model with optimization methods to improve the accuracy of determining the robot's position in a known environment. AMCL operates based on a set of particles, where each particle represents a hypothesis about the robot's position and orientation.

The localization process begins by initializing a set of particles distributed randomly in the space or based on an initial assumption of the robot's position. As the robot moves, the particles are updated according to the robot's motion model, allowing the algorithm to predict new positions for the particles. Then, sensor data, such as lidar or camera information, is used to assess how well each particle fits with the surrounding environment. Particles that match well with the sensor data are assigned higher weights, while particles that do not match are given lower weights.

Resampling is an essential part of AMCL, where particles with higher weights are more likely to be selected again, while particles with lower weights are discarded. This gradually improves the accuracy of the estimated position. Additionally, AMCL introduces random free particles, and the number of these particles is calculated based on long-term weight estimates ω_{slow} and short-term weight estimates ω_{fast}, which are expressed as follows:

$$\begin{cases} \omega_{slow} = \omega_{slow} + \alpha_{slow}(\omega_{avg} - \omega_{slow}), \\ \omega_{fast} = \omega_{fast} + \alpha_{fast}(\omega_{avg} - \omega_{fast}). \end{cases} \tag{8}$$

In which, ω_{avg} is the average weight of all particle points; α_{slow} and α_{fast} are parameters used to estimate the decay rate of the exponential filter for long-term and short-term ($\alpha_{fast} \gg \alpha_{slow} \geq 0$)).[6]

IV. Results and Discussion

This section evaluates the performance of the algorithms in processing and merging PointCloud data, as well as their capability in map scanning and localization experiments. The simulation results were obtained using the ROS Melodic platform.

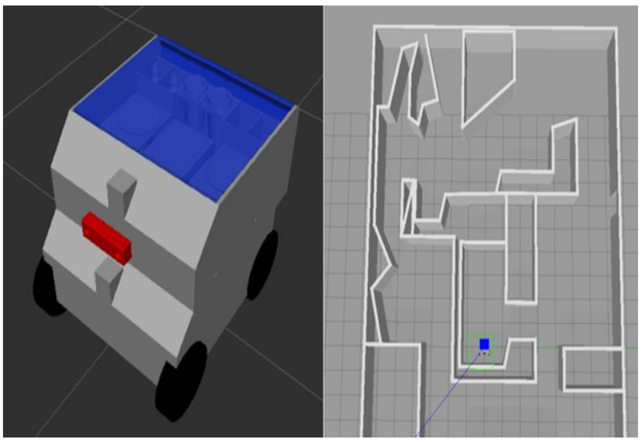

Fig. 2. System model and map in simulation.

The model has dimensions of 0.45x0.45x0.5m, with the depth camera and lidar simulated using the main specifications of Hokuyo and Kinect. The coordinates of the depth camera are (0.145, 0, 0.3) and those of the lidar are (0.13, 0, 0.23). Accordingly, we will have the transformation matrices for each device as follows.

The translation matrix of the depth camera: $\begin{bmatrix} 0.145 \\ 0 \\ 0.3 \end{bmatrix}$.

The translation matrix of the lidar: $\begin{bmatrix} 0.13 \\ 0 \\ 0.23 \end{bmatrix}$.

Since the sensors are not rotated, the transformation matrices are respectively:

From camera to base: $\begin{bmatrix} 1 & 0 & 0 & 0.145 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

From lidar to base: $\begin{bmatrix} 1 & 0 & 0 & 0.13 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.23 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

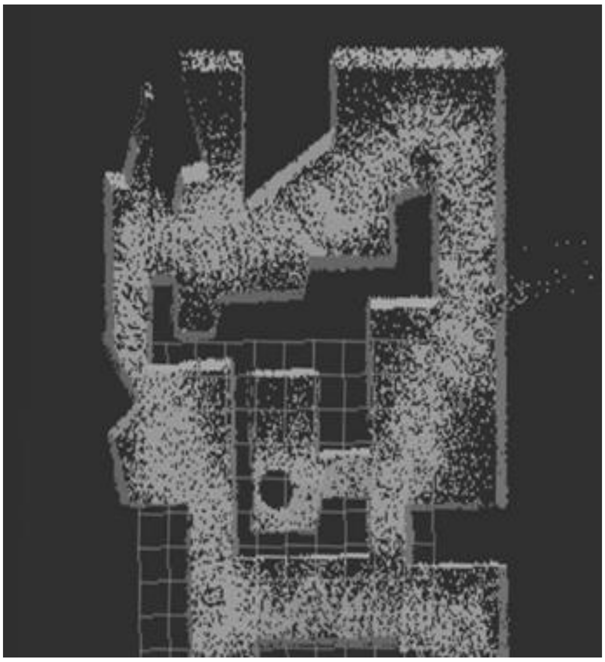

Fig. 3. The map after the scanning is completed (using NDT method).

After completing the processing and map registration, during the mapping process, each time a frame of the map is scanned, the quality of the registered point cloud is evaluated against the original point cloud. The evaluation of point cloud quality is based on two main standards.

The first is the overlap ratio. This is the percentage of points in the original point cloud that overlaps with other point clouds (NDT, ICP, GICP). The overlap ratio is calculated by counting the number of points from the original point cloud that have at least one nearest point in the other point clouds (using a K-d tree to find the nearest point) and dividing this by the total number of points in the original point cloud.

The second is mean deviation. This is the average of the Euclidean distances between corresponding points in the original point cloud and the other point clouds (NDT, ICP, GICP). This helps to determine the level of difference between the processed point clouds and the original point cloud. The deviation indicates the accuracy of the algorithms compared to the original data.

TABLE III. EVALUATION OF POINT CLOUD QUALITY ACCORDING TO DIFFERENT ALGORITHMS.

Applied algorithm	Overlap (%)	Mean Deviation (m)
ICP	0.851	0.105
GICP	0.884	0.098
NDT	0.961	0.067

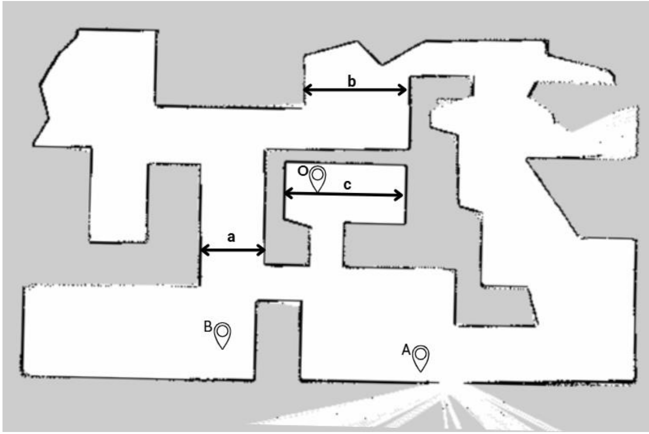

Fig. 4. The map after conversion to 2D.

TABLE IV. ERROR DATA OF THE OBTAINED MAP (SIMULATION)

Road section	Actual size on the simulation map (m)	Size on the map (m)
a	3	3.03
b	4.75	4.79
c	4.25	4.19

This is the result of localization coordinates on the indoor map (simulation) interpolated using the AMCL algorithm.

TABLE V. MOCALIZATION COORDINATES ON THE OBTAINED MAP (SIMULATION)

Point	*x*	*y*	OA (vehicle coordinates, m)	OA (Rviz tool, m)
O	0.002	-0.001		
A	-4.365	6.595	7.9106	8
B	6.806	5.649		

V. CONCLUSION

This study has completed the evaluation of the performance of PointCloud registration algorithms in a simulation environment, with the most outstanding result achieved from the NDT algorithm. Specifically, NDT demonstrated the highest performance with an Overlap value of 96.1% and a Mean Deviation of only 0.067 m, surpassing other algorithms such as ICP and GICP.

In addition, we successfully performed localization and map assessment in a real-world area using only lidar data. However, some work remains incomplete, including mapping with the support of depth cameras due to budget constraints. Future work will focus on integrating data from depth cameras into real-world environments, thereby evaluating its effectiveness on the accuracy and reliability of the system.

ACKNOWLEDGMENT

This research is funded by Ho Chi Minh city University of Technology (HCMUT), VNU-HCM under grant number SVKSTN-2024-ĐĐT- 14. We acknowledge Innovation FabLab, Ho Chi Minh University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] Hokuyo Automatic Co., "Hokuyo UTM-30LX," [Online]. Available: https://www.hokuyo-aut.co.jp/02sensor/05lidar/utm30lx_e.html. [Accessed: Aug. 19, 2024].

[2] Microsoft Corporation, "Kinect for Xbox 360," [Online]. Available: https://www.microsoft.com/en-us/kinectforwindows. [Accessed: Aug. 21, 2024].

[3] Besl, P. J., & McKay, N. D. (1992). "A method for registration of 3-D shapes." IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239-256.

[4] Segal, A., Haehnel, D., & Thrun, S. (2009). "Generalized-ICP." In Proceedings of Robotics: Science and Systems.

[5] M. Magnusson, "The Three-Dimensional Normal Distributions Transform - Theory and Applications," *Ph.D. dissertation*, *Örebro University*, Örebro, Sweden, 2009.

[6] L. X. Ding and W. J. Tao, "Design and implementation of localization navigation of indoor mobile robot in unknown environment," Ordnance Automation, vol. 3, 2018.

[7] Hao J. (2019) SLAM and navigation robot designs based on the Cartographer algorithm. Shandong university.

[8] L. X. Ding and W. J. Tao, "Design and implementation of localization navigation of indoor mobile robot in unknown environment," Ordnance Automation, vol. 3, 2018.

[9] Rusu, R. B., & Cousins, S. (2011). "3D is here: Point Cloud Library (PCL)." In IEEE International Conference on Robotics and Automation.