Robust Feature-Based Camera-LiDAR Fusion for Global Localization of a Mobile Robot

Yasir Salam, **Li Yinbei**, **Yang Jiaqiang***, **Fan Wei**

College of Electrical Engineering, Zhejiang University, Hangzhou, 310027, China, e-mail: yasir.salam@zju.edu.cn, liyinbei@outlook.com, 22110012@zju.edu.cn
**Corresponding author: yangjiaq@zju.edu.cn*

Abstract: Real-time global localization of a mobile robot is a prerequisite to accomplishing the planning and navigation tasks. The adaptive Monte Carlo Localization (AMCL) algorithm is broadly applied to mobile robots' localization in indoor environments. The localization accuracy of the AMCL is limited due to the irregularity of the laser sensor, the random nature of particle sampling, and the problem of final pose inference. This paper addresses the issues of global localization in planar indoor environments by presenting a feature-based technique that synergizes the capabilities of an RGB-D camera and 2D LiDAR. We proposed a novel method by customizing the Extended Kalman Filter (EKF) using Encoders and Inertial Measurement Unit (IMU) data for initial pose prediction and subsequent correction with RGB-D and LiDAR observations. Our approach incorporates robust state vector fusion, considering sensor malfunctions for improved accuracy. The developed system is evaluated using a mobile service robot, and the results demonstrate enhanced accuracy, computational efficiency, and robustness over state-of-the-art techniques.

Keywords: Global localization, EKF, RGB-D, LiDAR

1. INTRODUCTION

Mobile service robots have become popular and valuable in various applications, including education (Aldás V. and Palacios-Navarro, 2024; Storjak et al., 2022), healthcare (Kebede et al., 2024; Mišeikis et al., 2020), hotels (Fuentes-Moraleda et al., 2020; Yang et al., 2018), agriculture(Unal et al., 2023), and public environments (Pereira et al., 2013; Heerphone, 2020; Amal, 2024; Wang et al., 2023). Real-time localization on a global map is one of the main challenges for mobile service robots because the robot must know its current position and orientation before accomplishing any task. AMCL is generally applied for mobile robot localization in indoor environments (Kebede et al., 2024; Chung and Lin, 2022; Wolf et al., 2005; Fox, 2003). It employs a probabilistic localization method to estimate the position and orientation of the robot on a known map. By continually adjusting the number and distribution of particles based on observations, AMCL handles uncertainties and provides accurate localization.

AMCL relies on laser range sensors for observation inputs. However, the inherent irregularity of laser sensing, such as noise and errors induced by surface reflections, can degrade AMCL's localization accuracy. Furthermore, AMCL depicts belief distributions using randomly sampled particles, raising uncertainties in its estimates, (Sahabuddin et al., 2022). Particularly, selecting the final pose from the particle filter has shown difficulties in symmetrical environments, where AMCL's maximum likelihood estimate may be inaccurate. These characteristics of laser sensors exemplify fundamental constraints of AMCL that encourage the exploration of refinements or alternative localization practices. Moreover, the accuracy of AMCL strongly depends on the count of updates and the number of particles used.

Considering the above constraints, (Chung and Lin, 2022) proposed an improved localization method by increasing the frequency of updates and the number of particles. Although the accuracy of AMCL increases by increasing the frequency of updates, it adds computational cost and still encounters inaccuracies in environments with perceptually vague features. Specifically, AMCL can produce false positioning estimates that do not match the robot's ground truth pose in situations such as long corridors with repetitive textures or symmetrical areas lacking unique terrains.

Radiofrequency identification (RFID) transmitters and receivers are reliable sources for the localization of mobile robots due to their consistent performance under varying conditions. Unlike visual cues, RFID tags provide unique IDs and radio signals resilient to environmental changes (Choi et al., 2011; Han et al., 2007; Luo et al., 2007). RFID-based systems use sensor networks composed of tags that store the location information. The mobile robot mounted with an RF reader uses each tag's unique identification (ID) for localization. They provide coarse-grained information about the distance and direction between the transmitter and receiver, which is insufficient for accurate localization. To overcome this problem, (Choi et al., 2011) proposed a hierarchical localization approach that integrates ultrasonic sensors with the RFID system and achieved satisfactory results. However, RFID-based systems consume more power and require many units to cover a big area, limiting their applications for mobile robots.

Conversely, feature-based techniques are widely applied for localization and can potentially handle perceptual changes in dynamic environments. These techniques rely on detecting and matching the distinctive features to estimate the position and orientation of the robot relative to these features. Markov-based localization technique uses temporal filtering for the sequential information of image sequences through a similarity matrix (T. Naseer and Burgard, 2017). This vision-based method uses a Bayes filter to determine joint probability densities for image matching and subsequently performs sequential filtering on the thresholded probabilities, efficiently handling intricate trajectories that might entangle redundant loop-closures. Priority commitment towards the size and state space is the excessive requirement of Markov-based localization (Thrun, 2005), limiting its applicability to real-time applications.

Scale-invariant feature transform descriptors (Kosecka et al., 2005; Nitsche et al., 2014; Paul et al., 2021), the line features (Chai et al., 2011), and Harris corner (Davison and Murray, 2002), impartial of variations in translation and rotation, are being used as unique features. Nevertheless, the high computational complexity of these techniques makes the utilization of the CPU most of the time. In contrast to the visual features, laser range sensors, i.e., 2D and 3D LiDAR, significantly reduce the complexity of feature extraction algorithms while detecting the walls or corners in the structured environment (Roumeliotis and Bekey, 2000; David and Kerstens, 1998). Although (David and Kerstens, 1998) used a higher-level post-processing module to combine the lines to extract the corner features, the computational cost is still lower than the visual feature descriptors. Extracting the geometric shapes from the 2D point cloud is challenging because identifying the corner, wall, and circular objects requires sophisticated computational techniques and mathematical modeling.

A promising direction for robust global localization is combining the visual and laser data in a complementary way, leveraging their strengths and weaknesses. Natural and artificial landmarks specify actual coordinates and are generally integrated with wheel odometers to minimize cumulative position error in an indoor environment (Amal, 2024). Image retrieval techniques fused with Monte Carlo Localization (Wolf et al., 2005) and object recognition and text retrieval systems (Wang et al., 2006) were presented for localization purposes. Robust localization fusing camera and LiDAR improves the positioning estimates under occlusion, but landmark information retrieval from the offline visual BoWs increase the computational overhead of the system (Kebede et al., 2024). QR codes are also used as artificial landmarks for pose estimation and navigation (Cao et al., 2019; Zhang et al., 2015; Lee et al., 2015). These can be identified swiftly in all directions and decoded efficiently using existing techniques, reducing computational time. Due to the high computational cost of visual descriptors and uncertainty in the precise shape extraction from 2D laser data, we synergistically combine the data from both sensors enabling the system to capitalize on the advantage of both.

Our approach utilizes QR codes and corner features as landmarks in planar indoor environments, such as those characterized by walls and doors. We uniquely combine QR codes as artificial landmarks with natural corner features in indoor environments. This approach is designed to be more adaptable and less labor-intensive than traditional landmark-based systems. Simultaneously, our system capitalizes on corner features, which are ubiquitous and naturally occurring in indoor settings. We significantly reduce the need for additional, labor-intensive landmark placement by leveraging these existing environmental features. We strive to balance dependency on landmarks and advanced sensor fusion techniques. This balanced approach enhances the accuracy of global localization and lessens the reliance on an extensive network of landmarks by incorporating the IMU and Encoder data.

Feature extraction techniques in our work differ from those mentioned above; we used the 3D camera to detect and track the relevant QR code's region of interest (ROI) to determine the position and orientation information in contrast to (Zhang et al., 2015; Lee et al., 2015). This is accomplished by determining the depth distance and the observation angle of the QR code from the RGB-D sensor, explained in Section 3.1. We did not use any separate post-processing module to extract the corner features from the 2D LiDAR point cloud. Also, we only processed the relevant data from LiDAR to reduce the computational cost explained in Section 3.2. The following are the contributions of this research:

1. We propose a new technique for extracting distinctive features from RGB-D images and 2D LiDAR data, reducing the computational cost.

2. We developed a measurement model to use the extracted features and employed customized EKF integrating robust state vector fusion, which maximizes the information content of features and minimizes their uncertainty for accurate and robust localization.

3. We demonstrate our work's experimental results in terms of accuracy, computational complexity, and robustness in an indoor environment using a real mobile robot.

The organization of this work is as follows: Section 2 briefly explains the proposed experimental hardware and the calibration of RGB-D and LiDAR. Section 3 describes the feature extraction algorithms and localization approach. Section 4 explains the proposed robust state vector fusion for global localization. Section 5 presents the experimental validation of the proposed system. Finally, section 6 concludes the work with some future directions.

2. SYSTEM DESCRIPTION

This section describes the hardware specifications of the proposed experimental setup and an overview of the calibration of 2D LiDAR and RGB-D sensors.

2.1 Robot Description

The hardware used for the localization experiment is a two-wheeled differential-driven indoor mobile robot named Ubot, as shown in Figure 1.

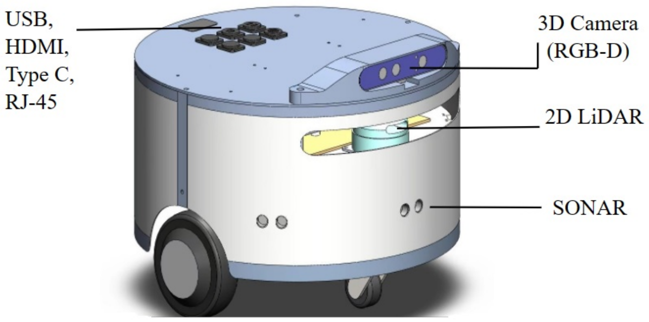

Fig. 1. Proposed mobile service robot.

Equipped with a built-in computer with a 2.4-GHz Intel I5 processor and 8-GB RAM, it runs Robot Operating System (ROS) endowed with navigation and visualization software packages. Its onboard sensors for mapping and navigation include two incremental wheel encoders with a resolution of 4096 pulses per revolution, a 2D laser sensor (RPLIDAR) with an angular range of 360^o and measuring distance of up to 16 $meters$, a $9-axis$ inertia measurement unit and an RGB-D camera (Orbbec Astro Pro) with FOV $60^o H \times 49.5^o V \times 73^o D$ and depth range of 0.6 to 8 $meters$. The intrinsic parameters of RGB-D are $f_{RDx} = 570.34$ and $f_{RDy} = 570.34$ for the focal lengths, while $p_{RDx} = 314.5$ and $p_{RDy} = 235.5$ for the principal point.

2.2 Extrinsic Calibration of 2D LiDAR and RGB-D

To determine the camera's position and orientation and its spatial relationship with the LiDAR sensor, we relied on the fixed mounting setup on our robotic platform. Both sensors were rigidly attached to the same base, providing a stable and consistent reference frame. This configuration was crucial for accurately aligning the RGB-D camera and LiDAR data, enabling precise localization. In our calibration process, extrinsic calibration is performed by using the algorithm (Zhang and Pless, 2004), and the intrinsic parameters of the RGB-D camera, including focal length and lens distortion coefficients, were obtained directly through the ROS framework. This approach ensured accurate retrieval of essential camera characteristics.

For a pinhole camera, mapping from the world frame $P = [X\ Y\ Z]^T$ to the image frame $p = [u\ v]^T$ can be described with the following expression (Song et al., 2016):

$$p \approx K(RP + t) \tag{1}$$

where K and R are the camera matrices representing the intrinsic parameters and orientation, respectively, and t is a vector that describes the position of the camera. The LiDAR point cloud represents the range data of the points in a plane parallel to the ground. The coordinate system of the 2D LiDAR is described with the origin at the sensor, and the scan plane is XZ with a zero height ($Y = 0$). A point P_c in the RGB-D reference system is positioned at the point P_l in the LiDAR reference system, and the rigid transform from RGB-D reference to LiDAR reference is represented by,

$$P_l = \psi P_c + \triangle \tag{2}$$

where $\triangle$ is a 3×1 translation vector describing the RGB-D sensor's position relative to the LiDAR reference frame, and ψ is a 3×3 orthonormal matrix representing the orientation of the RGB-D relative to the LiDAR reference frame. We determined the ψ and $\triangle$ matrices and adopted a unified coordinate system for the 3D vision and 2D laser sensors by omitting the one RGB-D coordinate representing height.

3. RGB-D AND LIDAR BASED LOCALIZATION

3.1 RGB-D Based Feature Extraction

RGB-D vision sensors use stereo vision or structured light to capture 3D information about the environment. QR codes are selected as artificial landmarks and generated using the online tool QRCode Monkey. Each QR code is encoded with different label information and detected with the coordinates of four edges and a center in the reference image frame. A bounding box is drawn around each recognized QR code and a circle on the center pixel (u, v), as shown in Figure 2. The center pixel (u, v) in the image reference frame is converted to camera frame of reference as follows,

$$\begin{aligned} X_s &= \frac{(u - p_{RDx}) \times d(u, v)}{f_{RDx}} \\ Y_s &= \frac{(v - p_{RDy}) \times d(u, v)}{f_{RDy}} \\ Z_s &= d(u, v) \end{aligned} \tag{3}$$

where, (p_{RDx}, p_{RDy}) is the principal point, (f_{RDx}, f_{RDy}) are the focal lengths of the RGB-D, and $d(u, v)$ evaluates the distance from the principal point of the RGB-D camera to the center of the recognized QR code in the image plane. To find the orientation information of the recognized QR code relative to RGB-D, we determined the angle of observation to the QR code's center in the RGB-D frame of reference as follows,

$$\theta_c = \cos^{-1}\left(\frac{CQ.CO}{\|CQ\|.\|CO\|}\right) \tag{4}$$

where, $C(X_c, Y_c, Z_c)$ is the position, $O(X_o, Y_o, Z_o)$ is the origin of the camera, and $Q(X_s, Y_s, Z_s)$ is the center of the recognized QR code in the camera coordinate frame.

3.2 LiDAR-Based Feature Extraction

LIDAR point clouds can detect obstacles and build a map of the environment, but they do not give information about objects' shape or orientation. In an indoor environment, we aim to extract the corners of walls and doors from the 2D LiDAR data. Figure 3 shows the corner feature extraction process. Initially, noise is removed from the raw LiDAR points by applying the Median filter. Median filtering eliminates attenuation and random variations in

Fig. 2. QR code detection

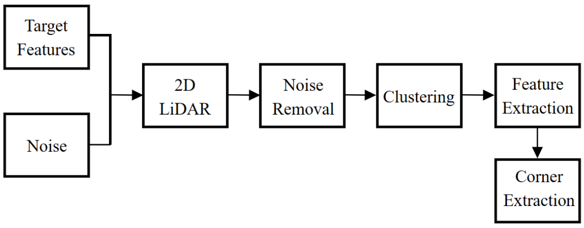

Fig. 3. Process of corner feature extraction

the LIDAR data (Lim et al., 2021). The raw data is then passed to the clustering stage, where 2D LiDAR measurements $(r_n, \theta_n), n \in R$ are separated into clusters based on the Euclidean distance. We use the Adaptive Break Point (ABP) algorithm (Borges and Aldon, 2004) for clustering, which compares the Euclidean distance between two consecutive LiDAR points p_n and p_{n-1} with the threshold distance D_{max} and assigns them to different clusters if the distance is greater than the threshold given by,

$$\|p_n - p_{n-1}\| > D_{max} \tag{5}$$

However, if equation (5) does not hold, p_n and p_{n-1} are clustered together. The threshold circle with a radius D_{max} is drawn around the point p_{n-1}, as shown in Figure 4. The fixed threshold D_{max} does not account for LiDAR's sparser measurements with increasing distance from the sensor. To address this, the ABP algorithm uses an adaptive threshold D_{max} based on the range d_n of the point p_n. This is achieved by drawing a virtual line through the point p_{n-1}, representing the most challenging scenario where the sensor detects an object at a specific angle of incidence. The virtual line forms an angle λ with the scanning angle θ_{n-1}. For the n-th hypothetical breakpoint p_n^h, the distance $\|p_n^h - p_{n-1}\|$ can be used as D_{max} for the breakpoint test and is determined as:

$$\|p_n^h - p_{n-1}\| = d_{n-1} \cdot \frac{\sin \triangle\theta}{\sin(\lambda - \triangle\theta)} \tag{6}$$

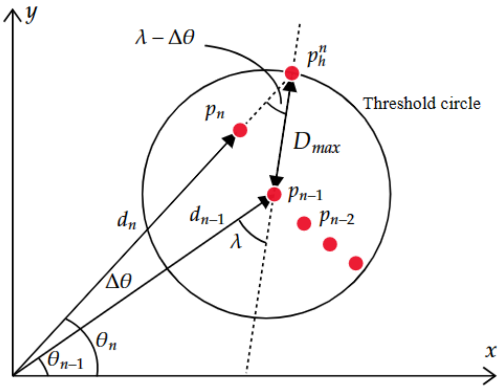

Fig. 4. Visualization of ABP algorithm

The adaptive threshold distance D_{max}, including the sensor error variance σ_r to account for inaccuracies when the range distance is small, is given as:

$$D_{max} = \|p_n^h - p_{n-1}\| + 3\sigma_r \tag{7}$$

After applying the ABP algorithm, the LiDAR point cloud $X \in \mathbb{R}^{n\times 2}$ is divided into segments $\{X_i\}$, where each $X_i \in \mathbb{R}^{n_i\times 2}$ contains n_i points. These segments $\{X_i\}$ are then passed to the feature extraction phase to extract geometric shapes. To locate corners in the environment, we apply the Search-Based Rectangle Fitting algorithm (Zhang et al., 2017), which scans all the LiDAR points and iterates in each possible direction to find rectangles. After each segment X_i is processed by the algorithm, L-shaped features, if present, are retrieved to determine the position of the corner points in the environment. The set of all the corner points at the k-th sampling period is given by,

$$C_k^i = \{C_k^1, C_k^2, ..., C_k^m\} \tag{8}$$

where, $C_k^i = (x_k^i, y_k^i), i = 1, ..., m$, is the corner point of the ith L shape feature. Figure 2 , shows that QR codes are strategically placed near the corners of the walls. Since equation (8) provides information about all the corner points in the environment, to ensure that the relevant corner (whose global coordinates are pre-known) is extracted, we calculated the coordinates of the center pixel (x_k^{qc}, y_k^{qc}) of the recognized QR code from equation (3) to determine the accurate location of the relevant corner C_k^{rc} from the RGB-D, such that:

$$C_k^{rc} = \underset{x_k^i, y_k^i \in C_k^i}{argmin} \sqrt{(x_k^i - x_k^{qc})^2 + (y_k^i - y_k^{qc})^2} \tag{9}$$

where, $C_k^{rc} = (x_k^{rc}, y_k^{rc})$ are the coordinates of relavent corner at the same sampling time k. Hence, the corresponding range r_k^L and angle θ_L measurements can be obtained from the (x_k^{rc}, y_k^{rc}).

3.3 Location Measurement

This section explains the proposed measurement model for the global pose estimation of Ubot by using the RGB-D

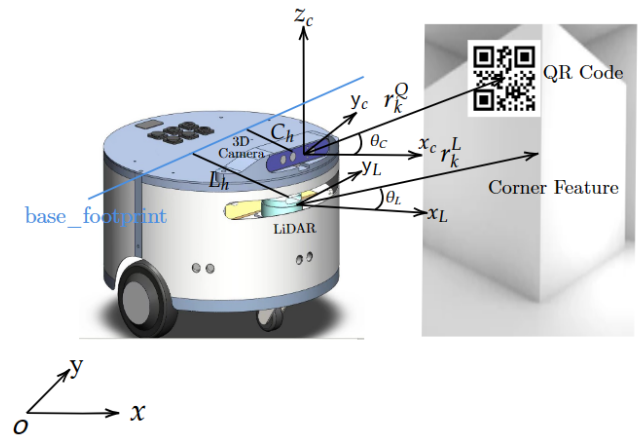

Fig. 5. Proposed measurement model

and LiDAR sensor measurements, as shown in Figure 5. The robot pose for the k-th sampling period is defined as, $P = [x_k \ \ y_k \ \ \theta_k]^T$, where (x_k, y_k) is the position and θ_k is the orientation of the robot.

The estimated robot pose $[\hat{x}_k \ \hat{y}_k \ \hat{\theta}_k]^T$, obtained from the LiDAR, can be expressed as follows,

$$\hat{x}_k = x_k^L - r_k^L \cos(\theta_L + \theta_k) - L_h \cos(\theta_k) \tag{10a}$$
$$\hat{y}_k = y_k^L - r_k^L \sin(\theta_L + \theta_k) - L_h \sin(\theta_k) \tag{10b}$$
$$\hat{\theta}_k = \theta_L + \theta_k \tag{10c}$$

where, (x_k^L, y_k^L) are the known global coordinates of the corner, r_k^L is the measured distance to the corner feature at the kth sampling period, θ_L is known angle of the corner feature, and L_h is the horizontal translation of LiDAR from the base-footprint (center of the robot). The robot pose $[\check{x}_k \ \check{y}_k \ \check{\theta}_k]^T$ at the same sampling period k acquired from the RGB-D camera is given by,

$$\check{x}_k = x_k^Q - r_k^Q \cos(\theta_c + \theta_k) - C_h \cos(\theta_k) \tag{11a}$$
$$\check{y}_k = y_k^Q - r_k^Q \sin(\theta_c + \theta_k) - C_h \sin(\theta_k) \tag{11b}$$
$$\check{\theta}_k = \text{atan2}(\check{y}_k - \check{y}_{k-1}, \check{x}_k - \check{x}_{k-1}) \tag{11c}$$

where, (x_k^Q, y_k^Q) are the known global coordinates of the QR code, r_k^Q is the measured distance to the center of the QR code at the k-th sampling period, θ_c is the angle of observation determined by the equation (4), and C_h is the horizontal translation of RGB-D from the base-footprint (center of the robot).

4. PROPOSED EKF-BASED GLOBAL POSE ESTIMATION

A customized EKF is applied for the global pose estimation of the robot, which predicts the robot's initial pose based on the data from Encoders and the IMU. Results from the equations (10) and (11) are employed in the correction step of EKF to obtain the state vectors (representing position and orientation) X_{dCam} and X_{Lidar} from RGB-D and 2D LiDAR, respectively. The EKF uses the same process noise covariance R in the prediction stage; however, measurement covariance matrices Q_{dCam} and Q_{Lidar} are defined according to the characteristics and noise levels of RGB-D and 2D Lidar, respectively. RGB-D and LiDAR perform well in moderate cases. However, both provide failed measurements when the RGB-D camera has difficulty estimating the QR code's surface normals and boundaries due to illumination problems or LiDAR having noise outliers or corner ambiguity.

We incorporated the adaptive switching rules to cope with these limitations to avoid false localization results. When both sensors perform well, the fusion algorithm finds the optimal scalar weights (Wang et al., 2010). The combined estimator X_C for the state vectors X_{dCam} and X_{Lidar} with the scaler weights is as follows:

$$X_c = \sum_{i=dCam,Lidar} \alpha^i X_i, \sum_{i=dCam,Lidar} \alpha^i = 1 \tag{12}$$

$$\alpha = \frac{R^{-1}e}{e^T R^{-1} e}, \alpha = [\alpha^{dCam} \ \alpha^{Lidar}]^T, e = [1 \ 1]^T \\ R = [tr(P_{dCam,Lidar})] \tag{13}$$

where X_i is the updated state vector, and α^i is the scalar weight for the ith sensor lies between 0 and 1, and $tr(P_{dCam,Lidar})$ is the trace of the associated cross-covariance matrix $P_{dCam,Lidar}$. The error variance matrix for X_c is P_c, given as follows,

$$P_c = \sum_{i,j=dCam,Lidar} \alpha^i \alpha^j P_{ij} \tag{14}$$

where $P_{ij}, i \neq j$ is the cross-covariance matrix for the error variance matrix of ith and jth sensors. In our case, P_{dCam} and P_{Lidar} are the error variance matrices for the state vectors X_{dCam} and X_{Lidar}, respectively. The error variance matrices are computed online as part of the EKF's iterative update process. The EKF framework is designed to handle this computation efficiently, updating the state vector and covariance matrix in real time based on the latest sensor measurements and their associated uncertainties. The estimated poses from equations (10) and (11) are fused with equations (12) and (14). Nevertheless, state vector fusion does not consider the failed measurements from the sensors. To ensure robustness, we modified the algorithm to assess the sensor malfunctions by comparing the position estimates from both sensors before calculating the weights for fusion.

If the Euclidean distance between the estimated positions $p_{Lidar} = [\hat{x}_k \ \hat{y}_k]^T$ and $p_{dCam} = [\check{x}_k \ \check{y}_k]^T$ is more than the threshold distance $D_{ther} = 200mm$ it means that the sensors disagree on the robot's position, and the estimate of a higher euclidean norm value is considered as the outcome of the failed estimate. Consequently, the weight of the failed measurement will become zero. Algorithm 1 explains all the meticulous steps, such as QR code detection, corner point extraction, and pose correction, culminating in a robust state vector fusion that determines the global pose of the robot. This description provides a

Algorithm 1 EKF-Based Global Localization

Require: 2D LiDAR and 3D Camera
Ensure: Global Pose Using equation (12)
Initialize EKF
if landmark detected **then**
    predict Pose Based on IMU and ODOM
    while $QR \neq 0$ **do**
        recognize QR Code and extract edges
        extract center pixel (u, v) and $d(u, v)$
        calculate 3D camera coordinates, equation (3)
        calculate observation angle (θ_c), equation (4)
        return $d(u, v)$ and θ_c
    end while
    while True **do**
        ABP-based clustering
        if $D < D_{th}$ **then**
            push in cluster C
        else
            start a new cluster
        end if
        Rectangle Fitting
        if criteria meet **then**
            extract L-shapes and corner points
            push in C_k, equation (8)
            find C_k^{rc}, equation (9)
        end if
    end while
    correct pose using equations (10), and (11)
    robust state vector fusion using equation (12)
    while True **do**
        calculate $d(p_{dCam}, p_{Lidar})$
        $= \sqrt{[p_{dCam} - p_{Lidar}]^T [p_{dCam} - p_{Lidar}]}$
        if $d(p_{dCam}, p_{Lidar}) > D_{ther}$ **then**
            if $\|p_{Lidar}\| > \|p_{dCam}\|$ **then**
                $\alpha^{dCam} = 1$ and $\alpha^{Lidar} = 0$
            else if $\|p_{Lidar}\| < \|p_{dCam}\|$ **then**
                $\alpha^{dCam} = 0$ and $\alpha^{Lidar} = 1$
            end if
        else
            find α^{dCam}, and α^{Lidar}, from equation (13)
        end if
    end while
else
    Rotate
end if

high-level overview of the algorithm, briefly touching on the key components and the overall objective.

Comparing only the position is a simple and fast way to check if the sensors agree on the robot's location. It can detect significant errors or failures that affect the robot's position and orientation. However, comparing the position only may not be sufficient for applications that require precise orientation control or navigation in complex environments. In such cases, comparing both the position and the orientation may be necessary to ensure accuracy and reliability.

5. EXPERIMENTAL RESULTS AND ANALYSIS

We conduct experiments on a real robot (Ubot) to verify the proposed robust global localization system in an indoor environment. First, we allow the Ubot to navigate around the experimental area, employing the gmapping algorithm as part of a SLAM (Simultaneous Localization and Mapping) process to perform the initial mapping. This approach enables the generation of a detailed 2D occupancy grid map of the environment. Utilizing this SLAM-generated map, we extract and register the global coordinates of corner and QR code features necessary for our subsequent feature-based localization task. To improve the accuracy and reliability of the generated map, we undertake a systematic modification process. Precise measurements of distances between walls and corners are manually obtained and used to refine the map, ensuring it accurately represents the actual physical environment. The modified map, with marked positions of artificial and natural landmarks, is shown in Figure 6, and the global coordinates of each landmark are presented in Figure 7.

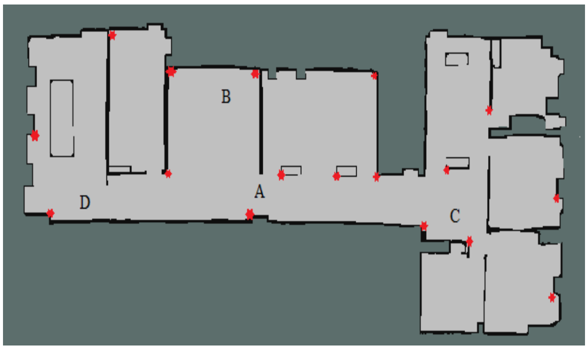

Fig. 6. SLAM map built by Ubot. The red marks indicate the location of landmarks, and the areas labeled A, B, C, and D indicate the localization test points.

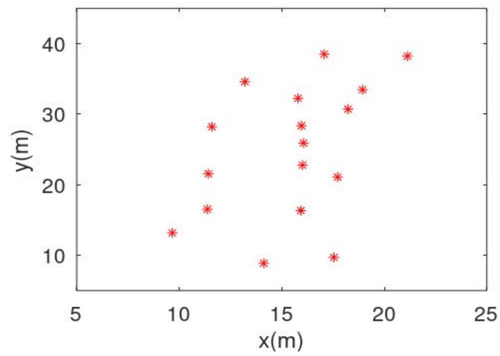

Fig. 7. Global coordinates of landmarks in map frame

5.1 *Localization Accuracy*

We evaluate the accuracy of our method through a comparative analysis with AMCL and the IAMCL(Chung and Lin, 2022) algorithms. IAMCL represents a recent advancement in localization techniques, providing a more contemporary benchmark for our evaluation. For a fair comparison, we employ AMCL and IAMCL on the same robotic platform (Ubot). Table 1 demonstrates the most correlated parameters used for both algorithms. We manually mark known reference positions on the experimental environment floor using grid markers as shown in Figure 8. To ensure reliable evaluation, we develop an accurate procedure to collect ground truth data tailored to our

Table 1. Configurations for the AMCL and IAMCL Method

Parameter	Specification
Motion Model	Differential Steering
Laser Model	Likelihood Mapping
Total Particles	10,000

environment. The robot pauses at each marked location and records the estimated pose using our localization system. This method, while labor-intensive, provides precise ground truth data for robust quantitative validation of the robot's localization performance across varied initial poses.

Figure 6 shows the location test areas A, B, C, and D, where we conduct the localization experiments by placing the Ubot and rotating with an angular velocity ($\omega = 0.3\ rad/s - 0.8\ rad/s$). The Ubot stops when it finds any landmark and returns the estimated pose relative to RGB-D and LiDAR features. To acquire the estimated pose from the AMCL and IAMCL algorithm, we performed roundtrip experiments from the marked positions by moving the Ubot along a straight line without obstacles in order to isolate the effects of odometry drift. The Ubot moves in the positive Y-axis direction by more than 1 meter and then returns to the initial position, providing the estimated pose. Subsequently, the Ubot moves in the negative Y-axis direction by 2 meters and is controlled to return to the initial position, giving the estimated pose. These actions are repeated five times to calculate the mean of the estimated positions.

We present the results of a specific experiment in Figure 9. The actual positions of Ubot, determined through ground truth measurements, are indicated by green points. The estimated positions obtained from the proposed EKF-based and IAMCL-based approaches are depicted by yellow and blue points, respectively. This visualization highlights the performance differences between the EKF and IAMCL methods in determining Ubot's position. The LiDAR scanned data, represented by the red point cloud, consistently aligns with the map, accurately reflecting Ubot's actual positions in the environment. Additionally, we calculate the mean absolute error (MAE) to assess localization error and measure the performance of our method. The MAE is defined as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|L_i - \hat{L}_i\right| \tag{15}$$

where L_i represents the actual location through ground truth and $\hat{L}_i$ represents the obtained location for the i-th experiment, and n is the total number of experiments. Figure 10 presents the MAE calculated over five experiments. Our method exhibits maximum localization errors of $0.052m$ in the X-axis and $0.105m$ in the Y-axis, which are lower than those of AMCL and IAMCL. IAMCL, an enhancement over standard AMCL, shows improved accuracy by iteratively executing the AMCL algorithm under specific conditions. While the original IAMCL work emphasized qualitative success rates, our implementation provides a quantitative evaluation of its performance in terms of MAE in both X and Y directions. Overall, EKF-based localization technique achieves higher accuracy than both AMCL and IAMCL. Irrespective of the local-ization accuracy, AMCL and IAMCL are prone to failure in environments with similar terrains (Chung and Lin, 2022).

Fig. 8. Experimental environment

Fig. 9. Comparison of localization results for Ubot at test locations. The actual positions (green points), estimated positions from the EKF-based method (yellow points), and estimated positions from the IAMCL-based method (blue points) are shown at Locations (a) A, (b) B, (c) C, and (d) D.

5.2 Computation Time and Robustness

We also evaluate the computation time of our method and compare it with that of AMCL and IAMCL. Table 2 presents the mean computation time of different algorithms. Computation time refers to the total time required to process from the initiation of feature extraction

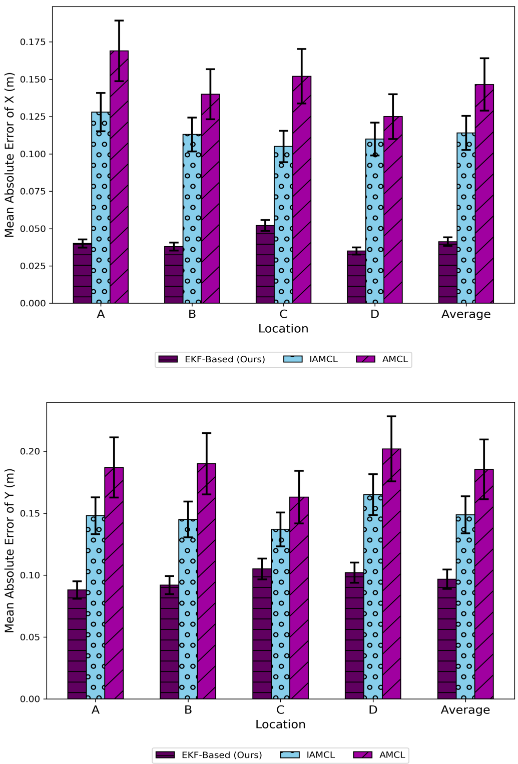

Fig. 10. Comparative analysis of Ubot localization errors in X and Y positions using EKF, AMCL, and IAMCL, with standard error bars.

Table 2. Configurations for the AMCL and IAMCL Method

Algorithm	**Time (s)**
EKF-Based (Ours)	0.036
AMCL	0.112
IAMCL	2.09

to the final output of estimated odometry. Our method demonstrates superior efficiency with a computation time of only 0.036 seconds, highlighting its efficiency for real-time applications. In contrast, the AMCL algorithm has a longer computation time of 0.112 seconds. This increase is due to its particle filter-based mechanism, which requires extensive simulations of multiple particles for position estimation. IAMCL, an extension of AMCL designed for enhanced accuracy, necessitates an even longer computation time of 2.09 seconds due to its iterative approach, where multiple AMCL processes are executed. Given these results, AMCL and IAMCL may not be suitable for time-critical or real-time applications if their computation times exceed acceptable performance or latency requirements.

The experiments demonstrated that the depth value of the recognized QR code varies slightly ($0 - 40mm$) depending on the viewing angle of the RGB-D sensor, indicating that depth estimation is not robust to changes in the orientation of Ubot. This depth error may result from inaccurate surface normal estimation, depth data completion artefacts, or changes in illumination. Additionally, irrelevant corners were extracted from the LiDAR data due to multiple adjacent corners (combination of L shapes) in the scenario, as shown in Fig. 11. The unrelated corners

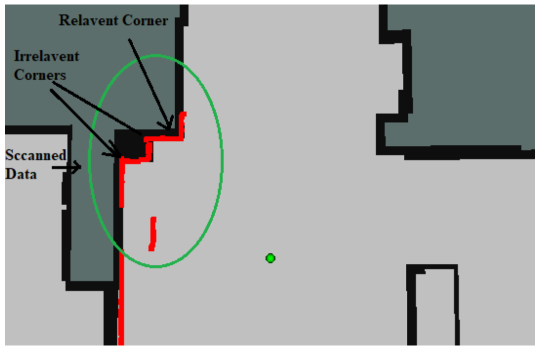

Fig. 11. Sensors malfunctioning at location C

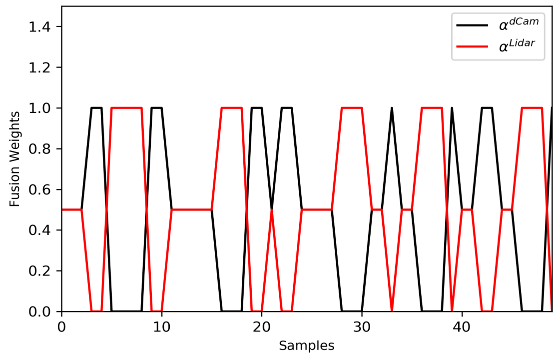

Fig. 12. Fusion weights at location C

are close and do not significantly affect the localization's accuracy. However, to evaluate the robustness of the state vector fusion, we considered the worst-case scenario, where the Ubot was placed near location C, as shown in Fig. 11. The proposed fusion method automatically rejected the failed measurements by switching the weights for either sensor at different time instances, as shown in Figure 12.

5.3 Dissecting Sensor Effects on EKF Accuracy

This section demonstrates the individual and synergistic effects of various sensor inputs on the accuracy of our method. We compare three sensor configurations: (1) IMU and encoder data (IMU+Enc), (2) IMU, encoder, and RGB-D feature measurements (IMU+Enc+RGB-D), and (3) IMU, encoder, and LiDAR data (IMU+Enc+LiDAR). Each configuration is evaluated through the roundtrip experiments discussed in Section 5.1, and the results are presented in Figure 13. The first configuration, IMU+Enc, demonstrates increased X and Y errors across all locations due to the limitations inherent in Encoder and IMU data, such as drift over time and lack of environmental interaction, which hinder accurate localization results. While the IMU+Enc+RGB-D configuration significantly reduces localization errors, it still faces challenges in depth perception, resulting in large errors compared to those presented in Figure 10. Conversely, the IMU+Enc+LiDAR setup illustrates improved localization accuracy due to LiDAR's superior range measurement capabilities. However, this setup lacks detailed feature recognition and fails to provide precise measurements at location C. As discussed in Section 5.2, these observations indicate that isolated sensor configurations cannot fully compensate for each other's

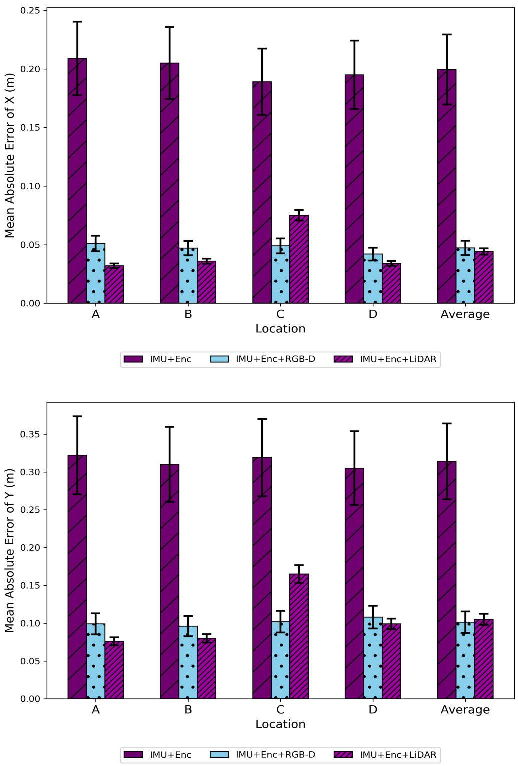

Fig. 13. Results of the sensor ablation study - analyzing the impact of different sensor configurations on localization accuracy, with standard error bars.

limitations, underscoring the necessity for a collaborative multi-sensor approach to achieve optimal performance.

6. CONCLUSION

This work aimed to improve the performance of a mobile robot's localization in terms of accuracy, computational complexity, and robustness by using distinctive features. Specifically, we used QR codes as artificial landmarks and corners as natural landmarks in the lab environment. Our framework extracted the QR codes from RGB-D and corner features from the LiDAR data by reducing the computational complexity compared to the existing feature-based techniques. Furthermore, we developed a measurement model that takes these features as input and returns the estimated global pose of the mobile robot. Finally, we performed robust state vector fusion by considering the sensor malfunctions, minimizing the features' uncertainty. Experimental results in accuracy, computational complexity, and feasibility demonstrate the presence of the presented EKF-based localization system over the AMCL and IAMCL algorithms.

Despite the sophistication of IAMCL, the results demonstrate that our EKF-based localization method outperforms IAMCL in critical aspects, i.e. accuracy and computation complexity. These results highlight the limitations of both existing methods for time-critical or real-time applications when computation time exceeds the desired performance or latency requirements. Ultimately, the presented technique showcases the efficacy and practicality of the developed system, paving the way for enhanced performance in various real-world applications. By introducing an EKF-based localization system, we provide a solid foundation for advancements in mobile robot localization. This novel approach can inspire researchers to explore new possibilities in optimizing mobile robot navigation, control, and path-planning strategies.

REFERENCES

Aldás V., J. and Palacios-Navarro, G. (2024). A ros-based open tool for controlling an educational mobile robot. *International Journal of Online and Biomedical Engineering (iJOE)*, 20(01), pp. 23–39.

Amal, B. (2024). Virtual navigation of mobile robot in v-rep using hybrid anfis-pso controller pdf brahami amal. *Journal of Control Engineering and Applied Informatics*, 26(2), 60–71.

Borges, G. and Aldon, M.J. (2004). Line extraction in 2d range images for mobile robotics. *Journal of Intelligent and Robotic Systems*, 40, 267–297.

Cao, X., Yang, Y., Lu, T., Fang, L., and Zhang, J. (2019). A fast pose estimation method based on new qr code for location of indoor mobile robot. *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, 257–262.

Chai, X., Wen, F., and Yuan, K. (2011). Fast vision-based object segmentation for natural landmark detection on indoor mobile robot. In *2011 IEEE International Conference on Mechatronics and Automation*, 2232–2237.

Choi, B.S., Lee, J.W., Lee, J.J., and Park, K.T. (2011). A hierarchical algorithm for indoor mobile robot localization using rfid sensor fusion. *IEEE Transactions on Industrial Electronics*, 58(6), 2226–2235.

Chung, M.A. and Lin, C.W. (2022). An improved localization of mobile robotic system based on amcl algorithm. *IEEE Sensors Journal*, 22(1), 900–908.

David, M. and Kerstens, A. (1998). Tracking naturally occurring indoor features in 2-d and 3-d with lidar range/ amplitude data. *The International Journal of Robotics Research*, 17, 907 – 923.

Davison, A. and Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 865–880.

Fox, D. (2003). Adapting the sample size in particle filters through kld-sampling. *The International Journal of Robotics Research*, 22(12), 985–1003.

Fuentes-Moraleda, L., Perez, P., Orea-Giner, A., Villace-Molinero, T., and Muñoz-Mazón, A. (2020). Interaction between hotel service robots and humans: A hotel-specific service robot acceptance model (sram). *Tourism Management Perspectives*.

Han, S., Lim, H., and Lee, J. (2007). An efficient localization scheme for a differential-driving mobile robot based on rfid system. *IEEE Transactions on Industrial Electronics*, 54(6), 3362–3369.

Heerphone, C. (2020). World robotics: Service robots 2020. *Internation Federation of robotics*.

Kebede, G., Gelaw, A., Andualem, H., and Tesfaye, A. (2024). Review of the characteristics of mobile robots for health care application. *International Journal of Intelligent Robotics and Applications*, 1–23.

Kosecka, J., Li, F., and Yang, X. (2005). Global localization and relative positioning based on scale-invariant keypoints. *Robotics Auton. Syst.*, 52, 27–38.

Lee, S.J., Tewolde, G., Lim, J., and Kwon, J. (2015). Qr-code based localization for indoor mobile robot with validation using a 3d optical tracking instrument. In *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 965–970.

Lim, H.b., Kim, E.s., Rathnayaka, P., and Park, S.Y. (2021). *Low-Resolution LiDAR Upsampling Using Weighted Median Filter*, 213–220.

Luo, R.C., Chuang, C.T., and Huang, S.S. (2007). Rfid-based indoor antenna localization system using passive tag and variable rf-attenuation. In *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, 2254–2259.

Mišeikis, J., Caroni, P., Duchamp, P., Gasser, A., Marko, R., Mišeikienė, N., Zwilling, F., de Castelbajac, C., Eicher, L., Früh, M., and Früh, H. (2020). Lio-a personal robot assistant for human-robot interaction and care applications. *IEEE Robotics and Automation Letters*, 5(4), 5339–5346.

Nitsche, M., Pire, T., Krajník, T., Kulich, M., and Mejail, M. (2014). Monte carlo localization for teach-and-repeat feature-based navigation. In M. Mistry, A. Leonardis, M. Witkowski, and C. Melhuish (eds.), *Advances in Autonomous Robotics Systems*, 13–24. Springer International Publishing, Cham.

Paul, S.K., Chowdhury, M.T., Nicolescu, M., Nicolescu, M., and Feil-Seifer, D. (2021). Object detection and pose estimation from rgb and depth data for real-time, adaptive robotic grasping. In H.R. Arabnia, L. Deligiannidis, H. Shouno, F.G. Tinetti, and Q.N. Tran (eds.), *Advances in Computer Vision and Computational Biology*, 121–142. Springer International Publishing, Cham.

Pereira, F., Vassallo, R., and Salles, E. (2013). Human–robot interaction and cooperation through people detection and gesture recognition. *Journal of Control, Automation and Electrical Systems*, 24.

Roumeliotis, S. and Bekey, G. (2000). Segments: a layered, dual-kalman filter algorithm for indoor feature extraction. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 1, 454–461 vol.1.

Sahabuddin, N.N.H.M., Yatim, N.M., Noh, Z.M., and Wahab, N.A. (2022). Performance analysis of particle filter localization algorithm for mobile robot. In K. Isa, Z. Md. Zain, R. Mohd-Mokhtar, M. Mat Noh, Z.H. Ismail, A.A. Yusof, A.F. Mohamad Ayob, S.S. Azhar Ali, and H. Abdul Kadir (eds.), *Proceedings of the 12th National Technical Seminar on Unmanned System Technology 2020*, 79–91. Springer Singapore, Singapore.

Song, H., Choi, W., and Kim, H. (2016). Robust vision-based relative-localization approach using an rgb-depth camera and lidar sensor fusion. *IEEE Transactions on Industrial Electronics*, 63(6), 3725–3736.

Storjak, I., Krzic, A.S., and Jagust, T. (2022). Elementary school pupils' mental models regarding robots and programming. *IEEE Transactions on Education*, 65(3), 297–308.

T. Naseer, B. Suger, M.R. and Burgard, W. (2017). Vision-based markov localization for long-term autonomy. *Robotics and Autonomous Systems*, 89, 147–157.

Thrun, S. (2005). *Probabilistic Robotics*. MIT Press, Cambridge, MA.

Unal, I., Kabas, O., and Eceoglu, O. (2023). Development of angular correction algorithm for movement of agricultural mobile robots in a straight line. *Journal of Control Engineering and Applied Informatics*, 25, 15–22.

Wang, J., Zha, H., and Cipolla, R. (2006). Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(2), 413–422.

Wang, M., Dong, X., Pan, W., Gao, S., and Wang, S. (2023). Robust path planning of obstacle avoidance for unmanned delivery robot. *Journal of Control Engineering and Applied Informatics*, 25(2), 60–71.

Wang, X., Zhu, Q., Jing, L., and Tao, L. (2010). Multisensor information fusion white noise deconvolution filter with colored noise. 1447–1453.

Wolf, J., Burgard, W., and Burkhardt, H. (2005). Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *IEEE Transactions on Robotics*, 21(2), 208–216.

Yang, P.Y., Chang, T.H., Chang, Y.H., and Wu, B.F. (2018). Intelligent mobile robot controller design for hotel room service with deep learning arm-based elevator manipulator. In *2018 International Conference on System Science and Engineering (ICSSE)*, 1–6.

Zhang, H., Zhang, C., Yang, W., and Chen, C.Y. (2015). Localization and navigation using qr code for mobile robot in indoor environment. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2501–2506.

Zhang, Q. and Pless, R. (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). volume 3, 2301 – 2306 vol.3.

Zhang, X., Xu, W., Dong, C., and Dolan, J.M. (2017). Efficient l-shape fitting for vehicle detection using laser scanners. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 54–59.