ML-Based Fast and Precise Embedded Rack Detection Software for Docking and Transport of Autonomous Mobile Robots Using 2-D LiDAR

Sunghoon Hong, *Student Member, IEEE*, and Daejin Park, *Member, IEEE*

***Abstract*—Autonomous mobile robots (AMRs) are widely used in dynamic warehouse environments for automated material handling, which is one of the fundamental parts of building intelligent logistics systems. A target docking system to transport materials, such as racks, carts, and pallets is an important technology for AMRs that directly affects production efficiency. In this letter, we propose a fast and precise rack detection algorithm based on 2-D LiDAR data for AMRs that consume power from batteries. This novel detection method based on machine learning to quickly detect various racks in a dynamic environment consists of three modules: first classification, secondary classification, and multiple-matching-based 2-D point cloud registration. We conducted various experiments to verify the rack detection performance of the existing and proposed methods in a low-power embedded system. As a result, the relative pose accuracy is improved and the inference speed is increased by about 3 times, which shows that the proposed method has faster inference speed while reducing the relative pose error.**

***Index Terms*—Low-power vision processing, machine learning, mobile robot, object detection.**

I. Introduction and Related Works

AUTONOMOUS mobile robots (AMRs) have become one of the important transportation methods in the intelligent logistics and manufacturing industries to reduce costs while improving efficiency. An AMR is a type of robot that can recognize its surroundings without external help, using only sensors equipped to the robot, and automatically move toward its destination [1].

Manuscript received 3 August 2024; accepted 9 August 2024. This work was supported in part by the BK21 FOUR Project under Grant 4199990113966; in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1A6A1A03025109 (10%) and Grant NRF-2022R1I1A3069260 (10%); and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government (MSIT, Metamorphic Approach of Unstructured Validation/Verification for Analyzing Binary Code, 30%) under Grant 2021-0-00944, (PIM Semiconductor Design Research Center, 20%) under Grant 2022-0-01170, and (Development of Flexible SW-HW Conjunctive Solution for On-edge Self-supervised Learning, 30%) under Grant RS-2023-00228970. This manuscript was recommended for publication by A. Shrivastava. *(Corresponding author: Daejin Park.)*

Sunghoon Hong is with the School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Republic of Korea, and also with the Next Software Design Team, Syscon ROBOTICS, Incheon 22851, Republic of Korea.

Daejin Park is with the School of Electronics Engineering, Kyungpook National University, Daegu 41566, Republic of Korea (e-mail: boltanut@knu.ac.kr).

Digital Object Identifier 10.1109/LES.2024.3442927

As an important part of intelligent logistics systems for automated material handling in dynamic warehouse environments, a target detection algorithm is a key technology for robust docking maneuvers to efficiently transport materials, such as racks, carts, and pallets. For example, AMRs perform material picking and delivery operations in material-transfer tasks. Material picking requires precise docking for the AMR to engage with the target, and precise docking requires continuous detection of the target.

A cascade classifier using Haar-like features is one of the object detectors for 2-D images [2], and various studies are currently being conducted to improve its performance [3], [4]. However, cameras are greatly affected by uneven lighting, and distance information is lost during the mapping from 3-D to 2-D space. Therefore, vision-based object detection can no longer satisfy the requirements of current industrial production.

To improve the accuracy of relative pose for precision target detection, most AMRs require time-of-flight sensors, such as LiDAR, with high range accuracy. The most widely used LiDAR-based relative pose estimation method involves point cloud registration (PCR) using iterative error minimization techniques to calculate the high-accuracy relative pose between two point clouds. The iterative closest point (ICP) algorithm is one of the most high-performing and well-known PCR methods and is still utilized in recent research [5].

For battery-powered AMRs, reducing hardware costs and power consumption is important. In this letter, we propose a fast and precise rack detection method based on machine learning for a robust target docking system using 2-D LiDAR in low-power embedded systems. The proposed method quickly and accurately detects various racks so that the AMR can engage with the target rack. The detection algorithm's computational resource cost can be reduced because the first classifier is implemented in data preprocessing. For robust target detection, multiple-matching-based 2-D PCR is also designed to precisely correct the relative pose between the AMR and the target.

We structure the rest of this letter as follows. Section II introduces the proposed methodology, including machine-learning-based fast object detection and relative pose correction, and Section III shows the performance of the proposed method. Finally, Section IV concludes of this letter.

II. Proposed Architecture

A. Relative Pose Estimation

Relative pose estimation is one of the important technologies used in computer vision and robotics to determine the position and orientation between the mobile robot and the target. This technology is essential in various applications, including object tracking, simultaneous localization and mapping, navigation, and target docking. The goal of relative pose estimation is to accurately calculate the rigid body transformation, $\boldsymbol{T}_{\mathrm{TR}} \in \mathrm{SE}(2)$, of the robot frame (denoted as R) in the imaged-based target frame (denoted as T). To calculate the transformation of relative pose, $\boldsymbol{T}_{\mathrm{TR}}$, the transformation matrix, $\boldsymbol{T}_{R'}$, in the image-based robot frame (denoted as R') is calculated on the tangent space as follows:

$$\boldsymbol{\xi}_{R'} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \boldsymbol{\xi}_R \tag{1}$$

where $\boldsymbol{\xi} = [\boldsymbol{t}^\top \, \boldsymbol{\omega}] \in \mathbb{R}^3$ is a tangent vector and $\boldsymbol{\omega} \in \mathfrak{so}(2)$ is the angular velocity (lie algebra of SO(2)) and $\boldsymbol{t} \in \mathbb{R}^2$. The problem of target detection is defined as finding the rigid body transformation, $\boldsymbol{T}_{\mathrm{TR}'} \in \mathrm{SE}(2)$, to precisely localize the pose of the image-based robot frame in the imaged-based target frame. The homogeneous transformation matrix, $\boldsymbol{T}_{\mathrm{TR}'}$ from $\{R'\}$ to $\{T\}$, can be defined as follows:

$$\boldsymbol{T}_{\mathrm{TR}'} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{TR}'} & \boldsymbol{t}_{\mathrm{TR}'} \\ 0 & 1 \end{bmatrix} \tag{2}$$

where $\boldsymbol{R}_{\mathrm{TR}'} \in \mathrm{SO}(2)$ and $\boldsymbol{t}_{\mathrm{TR}'} \in \mathbb{R}^2$ are the rotation matrix and the translation vector of the image-based robot frame relative to the target frame, with $\boldsymbol{t}_{\mathrm{TR}'} = [t_x, t_y]^\top$.

In 2-D LiDAR-based applications, the ICP algorithm is used to achieve point cloud fine registration. The problem of PCR is defined as finding the rigid body transformation, $\boldsymbol{T}_{\mathrm{TR}}$, that best aligns a *reading* point cloud of N_p points $\boldsymbol{P}_R \in \mathbb{R}^{2\times N_p}$ in the robot frame to a *reference* point cloud of N_q points $\boldsymbol{Q}_T \in \mathbb{R}^{2\times N_q}$ in the target frame.

The ICP algorithm has shown that, under ideal noise-free conditions, $\boldsymbol{T}_{\mathrm{TR}}$ is accurately calculated because point-wise correspondences are correctly matched. However, when noise is present, $\boldsymbol{T}_{\mathrm{TR}}$ is incorrectly calculated because point-wise correspondences are sometimes mismatched, increasing the probability of being trapped in a local minimum, such as when the scene is very ambiguous compared to the reference. Therefore, ICP-based relative pose estimation may not be an effective solution because the point cloud of the 2-D LiDAR has many unnecessary points to match with the point cloud of the reference image.

In this letter, we propose a novel detection method to quickly detect various targets in a dynamic environment using the template-matching-based first classification and machine-learning-based secondary classification. Also, for robust target detection, we designed a multiple-matching-based 2-D PCR using an ICP algorithm to precisely correct the relative pose in the region of interests (ROIs) that passed the 2nd classification and significantly reduce the computational cost of the ICP algorithm.

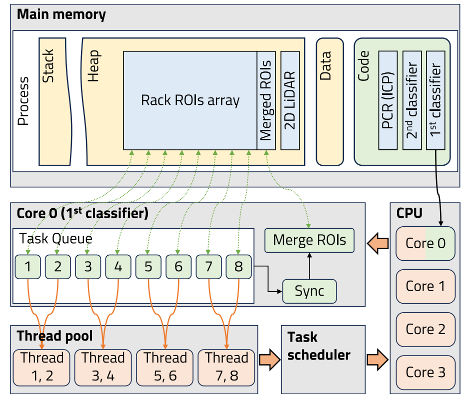


Fig. 1. Embedded software architecture for a target detection system.

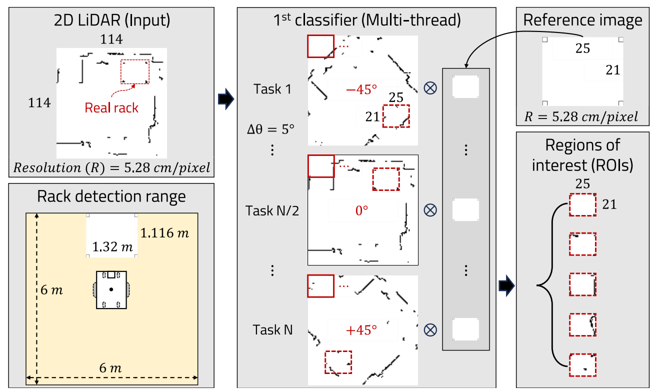


Fig. 2. Template-matching-based 1st classifier using multithreading.

B. Rack Detection

The main purpose of rack detection is to reduce the computational resource cost and the probability of being trapped in a local minimum by using the ICP algorithm. The rack detection algorithm we propose uses a thread-pool-based multithreading technique to efficiently use CPU resources, as Fig. 1 shows. The 1st classifier extracts candidates for rack ROI from each thread to detect the rack roughly and quickly. The sync function waits for all candidates to be extracted from each thread, and the merge ROIs function selects and merges ROIs suitable for the rack in the extracted ROI candidates. The 2nd classifier accurately detects the rack in the merged ROIs and estimates the relative pose between the AMR and the rack, and then PCR corrects the estimated relative pose using multiple reference images.

1) Template-Matching-Based 1st Classification: For fast rack detection, the template-matching-based 1st classifier uses the 2-D LiDAR reference image, as Fig. 2 shows. However, because template matching does not consider the rotation of the image, we perform rotation-based template matching using parallel processing of multithreading to find the relative pose, $\boldsymbol{T}_{R'T'}$, between the image-based robot frame and classifier-based target frame (denoted as T′). The image-based robot coordinate, ${}^{T}\tilde{\boldsymbol{p}}_{R'}$, in the target frame can be defined as follows:

$${}^{T}\tilde{\boldsymbol{p}}_{R'} = \boldsymbol{T}_{\mathrm{TR}'} \tilde{\boldsymbol{p}}_{R'} \tag{3}$$

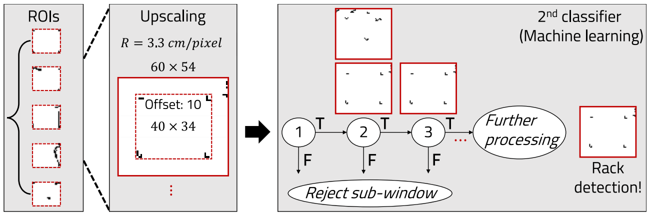


Fig. 3. Haar cascade-based 2nd classifier.

where $\tilde{\boldsymbol{p}} \in \mathbb{R}^3$ is the homogeneous vector, with $\tilde{\boldsymbol{p}} = [x, y, 1]^\top$. The 2-D LiDAR's point, $\boldsymbol{p}_{R'} \in \mathbb{R}^2$, in the image-based robot frame can be defined as follows:

$$
\begin{aligned}
\boldsymbol{D}_{\text{pixel}} &= \boldsymbol{D}_{\text{meter}}/\text{resolution} \\
\boldsymbol{p}_{R'} &= \boldsymbol{D}_{\text{pixel}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix}
\end{aligned} \tag{4}
$$

where $\boldsymbol{D}_{\text{pixel}}$ is the 2-D LiDAR distance in pixels, $\boldsymbol{D}_{\text{meter}}$ is the 2-D LiDAR distance in the real world, and resolution $=$ 0.0528 m/pixel is the resolution from pixel to meter. To calculate the template matching score for the 1st classifier, we use the reference point cloud instead of all coordinates in the reference image, as follows:

$$
\begin{aligned}
\boldsymbol{p} &= \boldsymbol{R}\boldsymbol{p}_{R'} + \boldsymbol{c} \\
S(\boldsymbol{R}, \boldsymbol{t}) &= \sum_{i=1}^{N} \left\{ \alpha \hat{\boldsymbol{B}}_{\boldsymbol{q}_i} - \beta \left(1 - \boldsymbol{B}_{\boldsymbol{t}+\boldsymbol{q}_i}\right) \right\}
\end{aligned} \tag{5}
$$

where $\boldsymbol{p} \in \mathbb{R}^2$ is a point in the image frame, N is the number of reference points, and $\boldsymbol{R} \in \mathrm{SO}(2)$ and $\boldsymbol{t} \in \mathbb{R}^2$ are the rotation matrix and translation vector to search for the target in the image frame. $\boldsymbol{c} \in \mathbb{R}^2$ is an image centroid coordinate, $\hat{\boldsymbol{B}} \in \mathbb{R}^{\hat{w} \times \hat{h}}$ is a binary image of the reference point cloud $\boldsymbol{Q} \in \mathbb{R}^{2 \times N_q}$, $\boldsymbol{B} \in \mathbb{R}^{w \times h}$ is a binary image of the image-based point cloud $\boldsymbol{P} \in \mathbb{R}^{2 \times N_p}$, and $S(\boldsymbol{R}, \boldsymbol{t})$ is a template matching score. $\hat{\boldsymbol{B}}_{\boldsymbol{q}_i}$ is always 1 because $\boldsymbol{q} \in \boldsymbol{Q}$ is the reference point, and $\boldsymbol{B}_{\boldsymbol{t}+\boldsymbol{q}_i}$ is 0 or 1.

2) Machine-Learning-Based 2nd Classification: For accurate rack detection, the machine-learning-based 2nd classifier uses Haar-like features. A Haar-like feature is the single weak classifier, and one weak classifier does not have enough information to detect a rack. Therefore, it is necessary to learn a stronger classifier using several meaningful weak classifiers. To extract only weak classifiers that are meaningful for rack detection, we use the Adaboost algorithm to learn Haar-like features [6].

Finally, the 2nd classifier is learned from numerous positive 2-D LiDAR images corresponding to racks and numerous negative 2-D LiDAR images to not racks in 25 × 21 regions passed through the 1st classifier, as Fig. 3 shows. The 2nd classifier is also implemented based on a bird's-eye view using 2-D LiDAR, so the detection speed is faster because it calculates only in a 25 × 21 area without considering the scale. The transformation matrix, ${}^{\text{2nd}}\boldsymbol{T}_{R'T'}$, of 25 × 21 regions passed through the 2nd classifier can be defined as follows:

$$
\begin{aligned}
{}^{\text{2nd}}\boldsymbol{T}_{R'T'} &= \mathrm{H}\left(\max_{\boldsymbol{R},\boldsymbol{t}} \{ S(\boldsymbol{R}, \boldsymbol{t}) \} \right) \\
&= \begin{bmatrix} \boldsymbol{R}^\top & \boldsymbol{R}^\top (\boldsymbol{t} - \boldsymbol{c}) \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{6}
$$

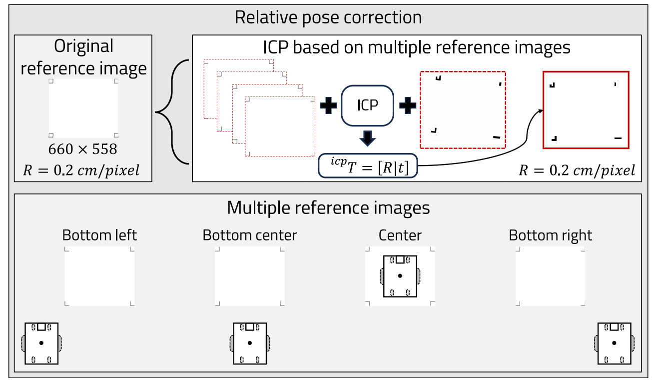


Fig. 4. Multiple reference images-based relative pose correction.

where ${}^{2nd}\boldsymbol{T}_{R'T'}$ is the transformation matrix of the classifier-based target frame relative to the image-based robot frame using the 2nd classifier, H.

3) Multiple-Matching-Based Point Cloud Registration: PCR using the ICP algorithm is based on minimizing the error function to calculate the rotation matrix, $\boldsymbol{R}$, and translation vector, $\boldsymbol{t}$. The ICP minimization problem with the point-point error function can be defined as follows:

$$
e(\boldsymbol{R}, \boldsymbol{t}) = \min_{\boldsymbol{R},\boldsymbol{t}} \sum_{i=1}^{N} \left\| \left(\boldsymbol{R}\boldsymbol{p}_{\mathrm{T}',i} + \boldsymbol{t} \right) - \boldsymbol{q}_{T,i} \right\|^2 \tag{7}
$$

where $e(\boldsymbol{R}, \boldsymbol{t})$ is the smallest average distance error, N is the number of point-wise correspondences, $\boldsymbol{p}_{\mathrm{T}'} \in \mathbb{R}^2$ is each *reading* point of the current point cloud $\boldsymbol{P}_{\mathrm{T}'}$ in the classifier-based target frame, and $\boldsymbol{q}_T \in \mathbb{R}^2$ is the closest *reference* point of the reference point cloud $\boldsymbol{Q}_T$ in the target frame using a *k-d tree* search for a correspondence search.

To improve the accuracy of PCR using ICP, the proposed method uses the multiple reference images, as Fig. 4 shows. If we use only the original reference image, point-wise correspondences are sometimes mismatched because the reference image has many similar features, so it is difficult to compare the point cloud of the 2-D LiDAR.

Therefore, to ensure accurate point-wise correspondence, reference images of various perspectives from which the mobile robot looks at the rack are used to reduce the probability of being trapped in a local minimum and improve the accuracy of the corrected relative pose, and the multiple-matching-based ICP minimization problem with multiple point-point error functions can be defined as follows:

$$
{}^{\text{icp}}\boldsymbol{T}_{\mathrm{T'T}} = \left[\min_{\boldsymbol{R},\boldsymbol{t}} e_i^N(\boldsymbol{R}, \boldsymbol{t}) \right]^{-1} \tag{8}
$$

where N is the number of reference images and $e_i^N(\boldsymbol{R}, \boldsymbol{t})$ is the smallest average distance error in the i-th reference image. ${}^{\text{icp}}\boldsymbol{T}_{\mathrm{T'T}}$ is the transformation matrix of the target frame relative to the classifier-based target frame using ICP. The corrected relative pose, $\boldsymbol{T}_{R'T}$, of the transformation matrix, ${}^{\text{2nd}}\boldsymbol{T}_{R'T'}$, can be defined as follows:

$$
\boldsymbol{T}_{R'T} = {}^{\text{2nd}}\boldsymbol{T}_{R'T'} \, {}^{\text{icp}}\boldsymbol{T}_{\mathrm{T'T}} \tag{9}
$$

where the rigid body transformation, $\boldsymbol{T}_{\mathrm{TR}'}$, is calculated by $\boldsymbol{T}_{R'T}^{-1}$.

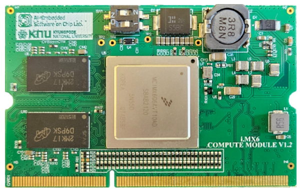


Fig. 5. i.MX6Q compute module. Powered by a single 12 V input, the compute module supports 2 GB DDR3 memory, 8 GB eMMC, a gigabit Ethernet PHY and a high-speed, high-density interconnect system.

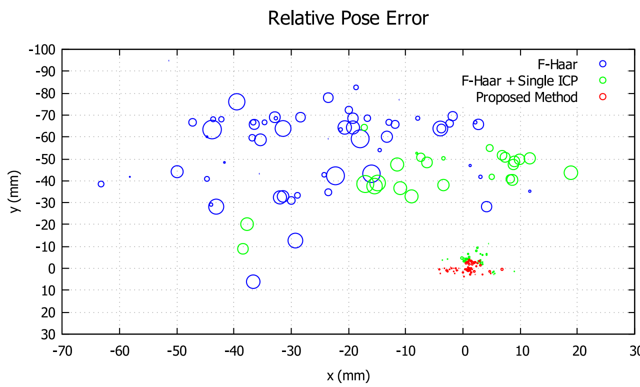
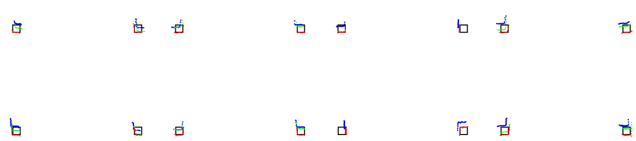


Fig. 6. Relative pose error results. Blue, green, and red circles are results using F-Haar, F-Haar + single ICP, and proposed method. The size of the circle is the magnitude of the relative heading angle error.



(a) (b) (c) (d)

Fig. 7. Visualization results of the detected rack pose. Black point cloud is reference points of the rack. Blue, green, and red point clouds are results using F-Haar, F-Haar + single ICP, and proposed method. (a) Bottom left. (b) Bottom center. (c) Center. (d) Bottom right.

III. Implementation and Experimental Result

To efficiently use CPU resources in low-power embedded systems, we implemented a rack detection algorithm using event-driven architecture. Also, we tested the proposed method's response speed and accuracy on the i.MX6Q compute module, as Fig. 5 shows. i.MX6Q is an application processor made by NXP and equipped with four 32-bit ARM® Cortex®-A9 processors, and the maximum operating speed per core is 1.2 GHz.

Fig. 6 shows the relative pose error result. The F-Haar method is a fast Haar cascade algorithm that uses the 2nd classifier in ROIs passed through the 1st classifier. We can see that the proposed method's relative pose accuracy is much higher than that of other methods.

Fig. 7 shows the visualization results of the detected rack pose. (a)–(d) are the visualization results when the AMR is located at the bottom left, bottom center, center, and bottom right of the rack. We can see that the point cloud of the proposed method is better aligned than with other methods.

TABLE I
Performance Comparison Results of the Traditional Method and the Proposed Method

Method	**FPS** (Avg)	**CPU** (Avg)	**MEM** (MB)	**Relative pose error (RMSE)** x (mm)	y (mm)	θ (°)
Traditional Haar	10	400%	67.6	33.13	51.04	1.81
F-Haar	33	19.8%	40.6	30.61	58.17	1.53
F-Haar + Single ICP	32	21.6%	43.2	9.90	29.16	1.15
Proposed Method	30	23.3%	48.9	2.17	1.85	0.21

Table I shows a comparison of performance with inference speed and relative pose error. We can see that the proposed method improves the inference speed by about three times compared to the traditional method. The F-Haar method we proposed is faster than the traditional Haar cascade method because it is performed in ROIs passed through the 1st classifier. Also, the proposed method reduces CPU usage by about 20.7 times compared to the existing method, thus helping reduce the hardware's power consumption and increase the AMR's travel time.

IV. Conclusion and Future Work

In this letter, we introduced a machine-learning-based fast and precise rack detection method for a robust target docking system using 2-D LiDAR. Unlike the traditional method, the proposed method has its computational cost in 25×21 regions passed through the 1st classifier, so it can reduce the computational costs of machine learning and the ICP algorithm. In addition, the response speed of the rack detection and accuracy of the relative pose are further improved by using the 1st classification, 2nd classification, and multiple-matching-based PCR techniques.

In the future, we plan to develop an efficient and robust target docking and transport system using a low-power embedded system-based AMR.

Acknowledgment

The EDA tool was supported by the IC Design Education Center (IDEC), South Korea.

References

[1] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2001, p. 1.

[3] S. Hong and D. Park, "Lightweight collaboration of detecting and tracking algorithm in low-power embedded systems for forward collision warning," in *Proc. 12th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2021, pp. 159–162.

[4] S. Hong and D. Park, "Runtime virtual lane prediction based on inverse perspective transformation and machine learning for lane departure warning in low-power embedded systems," in *Proc. IEEE Int. Conf. Imag. Syst Techn. (IST)*, 2022, pp. 1–6.

[5] T. Tuna, J. Nubert, Y. Nava, S. Khattak, and M. Hutter, "X-ICP: Localizability-aware lidar registration for robust localization in extreme environments," *IEEE Trans. Robot.*, vol. 40, pp. 452–471, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10328716

[6] Z. Hao, Q. Feng, and L. Kaidong, "An optimized face detection based on adaboost algorithm," in *Proc. Int. Conf. Inf. Syst. Comput. Aided Educ. (ICISCAE)*, 2018, pp. 375–378.