

Navigable Area Detection and Perception-guided Model Predictive Control for Autonomous Navigation in Narrow Waterways

Jonghwi Kim, Changyu Lee, Dongha Chung, Jinwhan Kim

Abstract—This paper presents an integrated navigation and control strategy for an autonomous surface vehicle (ASV) to operate in narrow waterways without relying on GPS. The proposed method uses a camera and a light detection and ranging (LiDAR) sensor to detect navigable regions in the waterway. A deep learning-based semantic segmentation algorithm is applied to detect the navigable region in camera images, and the segmented region is projected onto the water surface using planar homography. A line-detection algorithm is also introduced to improve the reliability of detecting navigable regions from LiDAR measurements. A safe collision-free path for the ASV is generated within the navigable regions using model predictive control-based local path planning and control algorithms. The performance and practical utility of the proposed method were demonstrated through field experiments using a small cruise boat, modified as an autonomous surface vehicle.

I. INTRODUCTION

AUTONOMOUS surface vehicles (ASVs) have attracted significant research and development interest due to their safety and economic feasibility over crewed operations. To ensure the robust and safe ASV operation, reliable autonomous navigation technologies including state estimation, situation awareness, path planning and control are required, and these technological components must be integrated to operate the vehicle system. While the GPS is an essential tool for navigation, its accuracy is limited and unreliable in some areas such as under bridges or near large steel structures where signals are blocked or deteriorated. Therefore, perception sensors including cameras, radar, and light detection and ranging (LiDAR) that can actively recognize the surrounding environment are necessary for reliable navigation and local path-planning.

Different strategies are required for different areas encountered during ship operation, open sea, river, canal, and coastal regions. In open sea areas, GPS is the primary localization sensor, while marine radar helps detect objects and prevent collisions with other ships or obstacles. However, in more

confined spaces such as rivers, coastal regions, and canals, avoiding grounding becomes a critical concern. In these areas, cameras and LiDAR sensors are relied upon heavily, with GPS playing a complementary role in ensuring navigation reliability. It is worth noting that human pilots often rely solely on visual cues, disregarding GPS, particularly when navigating through narrow waterways. Therefore, in such waterway environments, an alternative perception sensor-based navigation approach that does not depend on GPS may be necessary and preferable.

Recent research has proposed the idea of robot navigation along paths generated by analyzing the navigable region in the look-ahead direction. Research on mobile robots and autonomous vehicles has primarily focused on analyzing traversability through camera and LiDAR-based methods, including classical image processing [1]–[3] and deep learning-based segmentation techniques [4]–[8]. Additionally, sensor fusion-based methods for detecting drivable terrain and roadways [9]–[13] have also been explored. To navigate using information about the navigable region, planning and tracking control algorithms must be integrated [14]–[25]. Recent studies have introduced navigation methods that utilize the position of traversable regions, projected from vision-based road segmentation results to 2D ground coordinates, and have validated their methods through real-world experiments [18], [24], [25]. To provide more accurate geometric information, depth data from stereo vision [20], RGB-D sensors [15], and LiDAR [22] have also been used. To handle the complex path and nonlinear dynamics of the vehicle, model predictive control (MPC) [14], [21] or deep reinforcement learning (DRL) with traversability results have been employed [16], [26].

However, researches in the maritime environment are relatively limited compared to the ground environment. ASVs must detect the water surface to find a navigable region, which differs from ground vehicles. Some studies have proposed vision- [27]–[34], LiDAR- [35], [36], and marine radar-based [37] segmentation algorithms for the water or navigable region detection in the maritime environment. Additional modules have been developed for these methods, utilizing inertial data [27], [30], temporal context [28], and millimeter-wave radar measurement [31]. In [35], [36], water surface and object detection methods by semantic segmentation of LiDAR point cloud and fitting the blank region by geometric model were proposed to detect the navigable region. Navigation of an ASV in a narrow water environment by detecting the navigable region by stereo cameras was addressed in [38]. However,

This paper was recommended for publication by Editor P. Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Avikus Corp. It was also supported by the 'Autonomous Ship Technology Development Program (20011722, Development of a Situational Awareness System for Preventing Collisions and Accidents of Autonomous Ships)' funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

The authors are with Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea. stkimjh@kaist.ac.kr; leeck@kaist.ac.kr; chungdongha@kaist.ac.kr; jinwhan@kaist.ac.kr

(Jonghwi Kim and Changyu Lee contributed equally to this work.) (Corresponding author : Jinwhan Kim)

Digital Object Identifier (DOI): see top of this page.

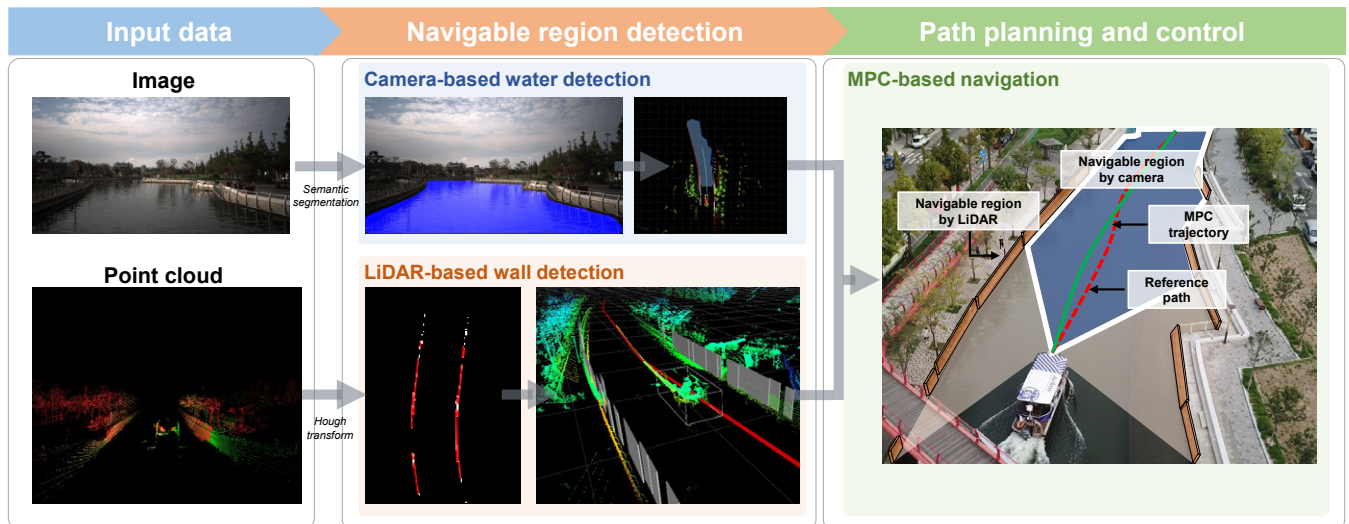


Fig. 1: An overall framework of the proposed method

most studies have focused on detecting the navigable region without linking the results to the navigation and control of an actual ASV.

To ensure safe autonomous navigation in narrow water areas, the control algorithms of ASVs must consider their underactuated and slow-motion characteristics. To address these issues, MPC-based approaches have been widely studied, and several studies have proposed collision avoidance path generation and following algorithms using MPC through simulations [39]–[43]. In recent studies, the trajectory tracking performance of ASVs has been verified through system identification and experiments [44]. The collision avoidance experiments using LiDAR-sensors and MPC have also been conducted [45], [46]. However, it should be noted that the previous studies mostly relied on localization and waypoint tracking using a prior map and GPS, and were limited to simulations that assumed accurate obstacle detection or lab-scale experiments with controllable experimental settings. MPC-based lane-keeping algorithms have also been studied, with applications similar to those of narrow canal navigation. In a recent experimental study, the effectiveness of a linear MPC algorithm that considers lane boundary constraints [47] and obstacle avoidance maneuvers during lane following was validated [48]. However, the autonomous navigation of ASVs discussed in this study poses unique challenges that are distinct from automotive applications. These challenges primarily arise from the irregular shapes of canal walls and the limited maneuverability due to the nonlinear dynamics of ships.

In this study, we propose an approach for autonomous navigation of an ASV in narrow waterways by detecting navigable regions using a camera and LiDARs. Instead of global path following using GPS, a front-view camera is used for local path planning, and LiDAR is additionally equipped on the ASV to leverage both appearance and geometrical information of canal boundaries. To detect the navigable region in the camera image, a deep learning-based segmentation algorithm is applied, and the detection result is projected

onto the water surface. A line-detection algorithm is used to enhance the performance of detecting waterway boundaries based on LiDAR measurements. Finally, an integrated MPC-based local path planning and control algorithm which enables navigating within the detected navigable regions is designed and implemented. Field experiments in the Pohang canal were conducted to verify the feasibility of the proposed method. The concept and overall framework of the proposed navigation approach are shown in Fig. 1. The main contributions of this study can be summarized into three points:

- A novel local path planning framework was proposed for ASV in canal environments, which addresses the issue of unreliable GPS signals. In contrast to conventional approaches that rely on globally set paths and positions obtained from GPS, our approach does not require a reference path annotated with GPS information.
- An integrated framework of perception, path planning, and control was developed. This framework detects the navigable areas using LiDAR and cameras in body-fixed coordinates, sets a reference path, and performs simultaneous collision avoidance and following control.
- This study conducted a field experiment in the Pohang canal using an autonomous cruise boat to test the proposed algorithm and integrated framework under real-world conditions, which provides valuable insights beyond the limited scope of previous simulations and lab-scale experiments.

II. NAVIGABLE REGION DETECTION

A. Vision-based Navigable Region Segmentation

We propose to use a front-view camera to detect the water area and determine a navigable path for the ASV, mimicking how humans navigate boats using their eyes. To detect the water area, we apply a semantic segmentation algorithm, which classifies pixels into predefined classes. We utilized the DeepLabv3 architecture [49], which incorporates atrous convolution and spatial pyramid pooling techniques to improve

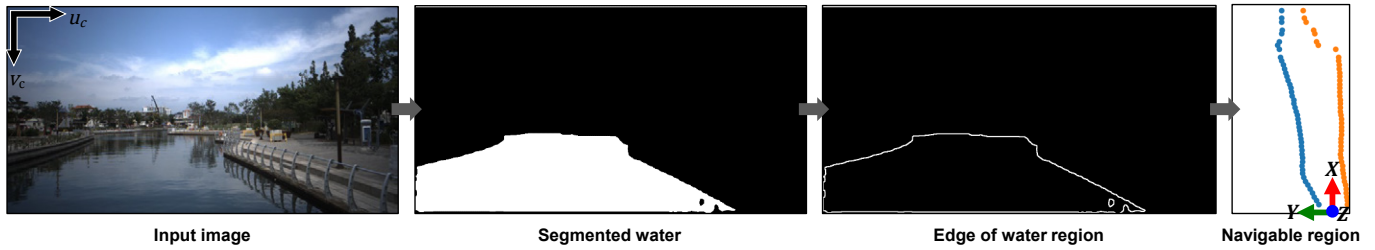


Fig. 2: Camera-based navigable region detection process and sample result

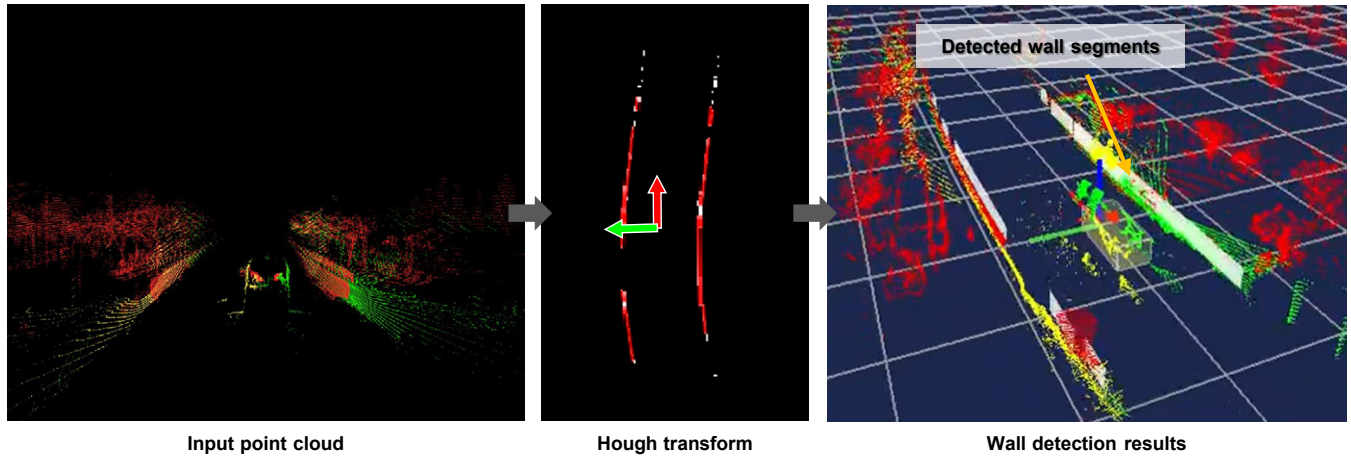


Fig. 3: LiDAR-based navigable region detection process and sample result. The detected wall segments are visualized as gray rectangles in the rightmost image.

the model's robustness to water reflections and wake disturbances. The front-view image is used as input, and binary cross-entropy loss is used to classify the pixels into water or other.

Once the water region is detected in the image, it is transformed into the ASV-fixed 2D water surface coordinate by camera geometry using the attitude measurement from the attitude heading reference system (AHRS). As all water surface pixels lie on the same plane, a homography matrix is used to formulate the projective transformation between the water pixels in the front-view image and the position of the water region in the top-view 2D world coordinate. It is formulated as follows:

$$s \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}, \quad (1)$$

where s is scale factor, u_c and v_c are the position of the water pixels in the image coordinate. f_x and f_y are the focal length, and c_x and c_y are the principal point in the u and v axes, respectively, and these are the intrinsic parameters of the camera. \mathbf{R} and \mathbf{t} are the rotation and translation matrix of the extrinsic parameter, respectively. X and Y are the positions of the water region in the water surface coordinate. The position of the water pixels in z direction are set to zero because water region is on the same surface. The intrinsic parameters (f_x , f_y , c_x , and c_y) are obtained by preliminary camera calibration, and the translation matrix \mathbf{t} is obtained by

measuring the position difference between the camera and the center of the ASV. The rotation matrix \mathbf{R} is determined by reflecting the transformation between the camera coordinate system and the ASV-fixed water surface coordinate system, and the ASV's attitude measurement from the AHRS sensor. Therefore, the unknown variables s , X , and Y are finally obtained by solving (1) using the fixed parameters and the water pixel position u and v from semantic segmentation results. To reduce the computational cost, the edges of the water pixels obtained in the segmented image are only used to determine the boundary of the navigable region. The projected points are then divided into left and right boundaries of the navigable water, as follows:

$$\begin{aligned} L(\mathbf{p}, \mathbf{q}) &= \{\mathbf{p} + t\mathbf{q} \mid \mathbf{p}, \mathbf{q} \in \mathbb{R}^2, t \in [0, 1]\}, \\ B_c^l &= \{L(\mathbf{p}_c^l(i), \mathbf{q}_c^l(i)), \text{ for } i = 1, 2, \dots, n\}, \\ B_c^r &= \{L(\mathbf{p}_c^r(i), \mathbf{q}_c^r(i)), \text{ for } i = 1, 2, \dots, n\}, \end{aligned} \quad (2)$$

where B_c^l and B_c^r denote the sets of n line segments that are determined by a pair of vectors consisting of projected points of the left ($\mathbf{p}_c^l, \mathbf{q}_c^l$) and right ($\mathbf{p}_c^r, \mathbf{q}_c^r$) boundaries in water surface coordinate, respectively, with n being the number of points. $L(\mathbf{p}, \mathbf{q})$ denotes the line segment with a finite length that is made by two endpoints \mathbf{p} and $\mathbf{p} + \mathbf{q}$. The process of the camera-based detection algorithm and sample result are shown in Fig. 2.

B. LiDAR-based Wall Detection

As LiDAR cannot see water due to its wavelength, the blank region below a certain height threshold is assumed to be water, and thus the side wall and fence of the waterway are used as boundaries of the navigable region. The point cloud within the predefined threshold is filtered and projected onto the water surface coordinate to create an image. The Hough transform is then applied to detect the line segment representing the side wall and fence of the waterway. The detected wall segments are expressed as follows:

$$\begin{aligned} B_l^l &= \{L(\mathbf{p}_l^l(i), \mathbf{q}_l^l(i)), \text{ for } i = 1, 2, \dots, m\}, \\ B_l^r &= \{L(\mathbf{p}_l^r(i), \mathbf{q}_l^r(i)), \text{ for } i = 1, 2, \dots, o\}, \end{aligned} \quad (3)$$

where B_l^l and B_l^r represent the sets of line segments representing the detected walls on the left and right sides, respectively, with a total of m and o segments in each set. Let $(\mathbf{p}_l^l, \mathbf{q}_l^l)$ and $(\mathbf{p}_l^r, \mathbf{q}_l^r)$ be the pair of vectors representing the left and right boundaries on the water surface coordinates, respectively. The LiDAR-based wall-detection procedure and sample result are shown in Fig. 3.

III. MPC-BASED NAVIGATION

The MPC method was employed to optimize the trajectory and control inputs for avoiding non-navigable areas detected by camera and LiDAR sensors. To design the MPC algorithm, knowledge of the mathematical model of the control target, the reference path for the objective function, and constraints are required.

A. Boat Dynamic Modeling

Because excessive control of ship speed and turning is discouraged in the Pohang canal for safety reasons, control is mostly performed at low speeds. To generate a safe and efficient trajectory with minimal model complexity while considering enough physical characteristics, the following model was employed. Let u be the speed and r be the yaw rate of the boat. Then the following equations can be obtained:

$$\begin{aligned} \dot{x} &= u \cos \psi, \quad \dot{y} = u \sin \psi, \quad \dot{\psi} = r, \\ \dot{u} &= (d_u u + F \cos \delta) / m, \\ \dot{r} &= (d_r r + l_y F \sin \delta) / I_r, \\ F &= c_T n_T^2, \quad \delta = c_s n_s, \end{aligned} \quad (4)$$

where d_u and d_r represent the linear drag coefficients, F and δ are the force and angle of the outer board, respectively, m is the mass, and I_r is the moment of inertia of the boat. n_T and n_s are the control inputs of thrust and steering estimated by the encoder of the attached motor, as shown in Fig. 5b. The coefficients of the control inputs are represented by c_T and c_s .

B. Trajectory Optimization and Control

To generate a reference path to compose the objective function of the MPC, the navigable region from Section II is used. The center points are calculated by taking the average of the left and right boundaries, and then a Bezier curve interpolation is applied to make a smooth path. Based on the target speed in

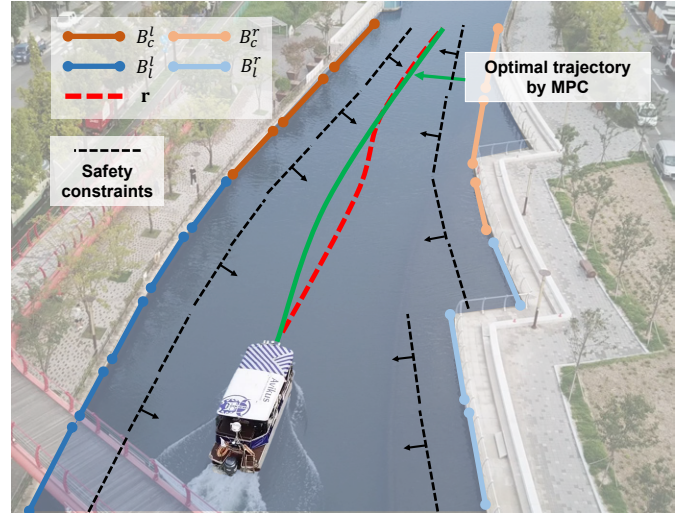


Fig. 4: Reference path and inequality constraints

the canal area, reference points are selected along the smooth path at intervals corresponding to the prediction horizon, as shown in Fig. 4. The obtained reference state can be expressed as follows:

$$\mathbf{r} = [x_r, y_r, \psi_r, u_r, 0, 0, 0]^T, \quad (5)$$

where x_r, y_r and ψ_r represent the desired position and heading in the ASV-fixed water surface coordinate, respectively, and u_r is the desired speed. The desired heading is determined by calculating the slope of the tangent line at the reference position.

The objective function has been formulated to avoid non-navigable areas and follow the reference path with minimal input effort, considering the dynamic equations of the boat. The state is represented by $\mathbf{x} = [x, y, \psi, u, r, n_T, n_s]^T$ and the input by $\mathbf{u} = [\Delta n_T, \Delta n_s]^T$. The function can be described as follows:

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \sum_{i=0}^{N-1} \ell(\mathbf{x}_i, \mathbf{r}_i, \mathbf{u}_i) + \ell_T(\mathbf{x}_N, \mathbf{r}_N) \quad (6a)$$

$$\text{s.t. } \mathbf{x}_0 - \mathbf{x}_{init} = 0, \quad (6b)$$

$$\mathbf{x}_{i+1} - f_d(\mathbf{x}_i, \mathbf{u}_i) = 0, \quad i = 0, \dots, N-1, \quad (6c)$$

$$g(\mathbf{x}_i, \mathbf{u}_i) \leq 0, \quad i = 0, \dots, N, \quad (6d)$$

$$h(\mathbf{x}_i) \leq 0, \quad i = 0, \dots, N, \quad (6e)$$

where i is the time index, $\mathbf{r}_i = [x_{r,i}, y_{r,i}, \psi_{r,i}, u_{r,i}, 0, 0, 0]^T$ is the reference state, \mathbf{x}_{init} is the initial state, ℓ is the stage cost function, ℓ_T is the terminal cost function, and N is the prediction horizon. The equality constraint for system dynamics is defined as (6c) and the inequality constraints for input and state are defined as (6d) and (6e), respectively. Let f_d denote the function obtained by discretizing (4) using the Euler method with a sampling time of T_s .

The stage and terminal cost functions penalize the error between the predicted states and the reference states as follows:

$$\begin{aligned} \ell(\mathbf{x}_i, \mathbf{r}_i, \mathbf{u}_i) &= (\mathbf{x}_i - \mathbf{r}_i)^T Q (\mathbf{x}_i - \mathbf{r}_i) + \Delta \mathbf{u}_i^T R \Delta \mathbf{u}_i \\ \ell_T(\mathbf{x}_{N_p}, \mathbf{r}_{N_p}) &= (\mathbf{x}_N - \mathbf{r}_N)^T P (\mathbf{x}_N - \mathbf{r}_N), \end{aligned} \quad (7)$$

where the matrices Q , R , and P represent the weighting matrices of the cost function that penalize errors in the states, the rate of change of the control input, and the terminal state error, respectively.

The inequality constraints for the control inputs and their rate of change $g(\mathbf{x}_i, \mathbf{u}_i) \leq 0$, as defined in (6d), are given as follows:

$$\begin{aligned} |n_T| &\leq n_{T,\max}, & |n_S| &\leq n_{S,\max} \\ |\Delta n_T| &\leq \Delta n_{T,\max}, & |\Delta n_S| &\leq \Delta n_{S,\max} \end{aligned} \quad (8)$$

where the subscript $(\cdot)_{\max}$ indicates the maximum value of the corresponding variables. The inequality state constraints for obstacle avoidance, as defined in (6e), are given as follows:

$$h(\mathbf{x}_i) = \begin{cases} a_j x_i + b_j y_i + c_j, & \text{for left boundary,} \\ -(a_j x_i + b_j y_i + c_j), & \text{for right boundary,} \end{cases} \quad (9)$$

for $j = 1, 2, \dots, n_l$,

where (a_j, b_j, c_j) represents the j -th linear inequality constraints on the left or right sides of the canal. To reduce the computational complexity, for each i -th reference point, the closest n_l line segments are considered as constraints, as shown in Fig. 4.

The real-time MPC problem, as formulated in 6, is solved using the real-time iteration algorithm. This algorithm is generated by the ACADO Code Generation Toolkit [50]. The nonlinear programming problem is solved using a sequential quadratic programming algorithm, while the quadratic program is solved using a parametric active-set algorithm [51].

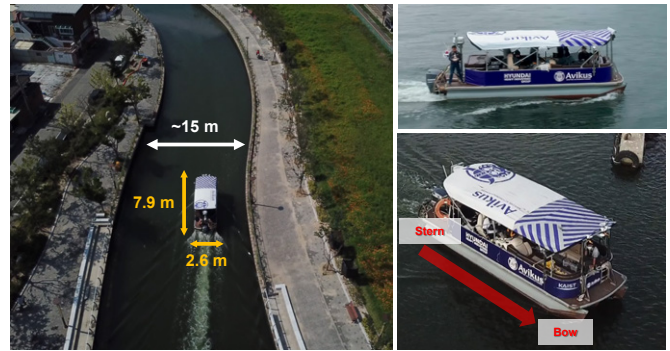
IV. FIELD EXPERIMENTS

A. Experimental Setup

To verify the feasibility of the proposed navigation approach, we conducted field experiments with an ASV developed by remodeling a 12-passenger cruise boat that had been operating in Pohang, South Korea. Among many sensors newly installed in the ASV, the left camera and three LiDARs (front, starboard, and port) were used for perception, and AHRS was used for measuring the attitude of the ASV. GPS was used for recording the ASV's trajectory during these experiments. For controlling the ASV, the motors were installed on the steering wheel and thruster bar, and the encoder was equipped to the thruster. Figure 5 shows the experimental platform and detailed hardware configuration. The experiment was conducted in the narrow waterway of Pohang canal, which has an average width and length of approximately 15 m and 1 km, respectively.

Approximately 2,000 images from water segmentation datasets [52]–[54] were used to fine-tune the DeepLabv3 model with a ResNet-101 backbone for semantic segmentation. We took advantage of the water images captured from the perspective of a boat for training similar experimental environments. Due to the computational capability of the computers equipped, the input image size was reduced to 1024×540 , and the camera-based navigable region detection algorithm was executed at a rate of 1 Hz.

To determine the model of the ship described in (4), experimental data from zigzag and turning maneuvers were gathered



(a) Overview of the experimental platform



(b) Sensor configuration and thruster control devices

Fig. 5: Experimental platform and hardware configuration

TABLE I: Parameters of the controller

Symbol	Value	Symbol	Value
m	1.9149e+03	d_u	-29.220
I_r	1.9351e+03	d_r	-2.1940e+03
c_T	1.3331e-05	c_s	4.3633e-03
N	25	T_s	1.0 sec
$n_{T,\max}$	100%	$n_{S,\max}$	100%
$\Delta n_{T,\max}$	10%/s	$\Delta n_{S,\max}$	40%/s
l_y	3.0 m	δ_{\max}	25°
Q	diag([1, 1, 500, 10, 1000, 0, 0])		
R	diag([0.0001, 0.0001])		
P	diag([1, 1, 500, 10, 1000, 0, 0]) $\times N$		

and utilized. The optimal model parameters P^* were estimated using a nonlinear programming approach, as follows:

$$P^* = \arg \min_P \sum_{i=0}^M (\mathbf{x}_i - \bar{\mathbf{x}}_i)^\top W (\mathbf{x}_i - \bar{\mathbf{x}}_i), \quad (10a)$$

$$\text{s.t. } \mathbf{x}_{i+1} = f_d(\mathbf{x}_i, \bar{\mathbf{u}}_i), \quad (10b)$$

where M is the number of data samples, W is the weight matrix, $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are the state and control input of the experimental data, respectively. The vector P consists of unknown parameters in (4). The estimated model parameters and MPC settings are summarized in Table I.

B. Experimental Results

Figure 6 shows the trajectory of the ASV recorded by GPS overlaid on a Google satellite map, and the sample images of experimental site. The ASV navigated the narrow waterway

with an average velocity of 4.08 knot, a path length of 1.05 km, and a time of 8 minutes and 15 secs. The green squares in Fig. 6 represent the sample locations shown in detail in Fig. 7.



Fig. 6: ASV trajectory and sample images of the environmental site

Figure 7 shows the sample results of camera and LiDAR-based navigable region detection and the corresponding MPC-based path planning. In the figures on the left, the segmented water region by camera is shown in blue pixels. The middle and right figures show the results in top-view and perspective-view, respectively. The point clouds from the front, port, and starboard LiDAR are represented by green, yellow, and red points, respectively. The camera-based navigable region detection results are represented as blue regions, and left and right boundaries are shown as red and blue spheres. The LiDAR-based wall detection results are shown as gray cuboids. The generated reference path is shown as blue lines, and the optimized path by MPC planning is shown as green lines.

Moreover, a heterogeneous sensor combination was necessary because they cover different detection ranges, as shown in Fig. 7. To emphasize the necessity of using both camera and LiDAR, we measured their limits of detection ranges, as shown in Table II.

TABLE II: Detection range limit of camera and LiDAR

Sensor	Avg. detection range limit	Max. detection range limit
Camera	130.3 m	340.1 m
LiDAR	29.8 m	56.5 m

Considering that the prediction horizon distance was approximately 50 m, as the prediction horizon was 25 seconds and the average speed was 4.08 knots, the use of both camera and LiDAR was necessary to cover the entire prediction horizon. As indicated in Table II, the camera was essential for long-range path planning in the MPC framework. However, LiDAR sensors were also required to provide a wider

horizontal field of view and the ability to detect close objects, which complements the limitations of the camera.

In summary, the ASV successfully navigated autonomously through the narrow waterway of 1.05 km in 8 minutes, using the camera and LiDARs without any localization method such as GPS or a prior map. The feasibility of the proposed navigation method has been demonstrated.

V. CONCLUSIONS

In this study, we proposed an autonomous navigation approach for surface vehicles operating in narrow waterways. A camera and LiDAR were used for detecting navigable regions. This enables the generation of a reference path in the body-fixed coordinate system, thereby overcoming the limitation of GPS-dependent path following. The deep learning-based semantic segmentation was used to detect water pixels in the image acquired from the camera. Their positions were transformed into the ASV-fixed water surface coordinate system and regarded as the navigable region. The point cloud from the LiDAR was projected onto the water surface coordinate system, and a line-extraction algorithm was applied to detect the side walls of the waterway. A reference path was generated from the water boundary from camera-based detection results, and an MPC framework was designed for path planning and control of the ASV considering reference path tracking, boundary avoidance, and ship dynamics. Field experiments were conducted to verify the feasibility of the proposed algorithm.

REFERENCES

- [1] D. Kim, S. M. Oh, and J. M. Rehg, "Traversability classification for UGV navigation: A comparison of patch and superpixel representations," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3166–3173.
- [2] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous robots*, vol. 18, no. 1, pp. 81–102, 2005.
- [3] S. Matsuzaki, K. Yamazaki, Y. Hara, and T. Tsubouchi, "Traversable region estimation for mobile robots in an outdoor image," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3, pp. 453–463, 2018.
- [4] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *European Conference on Computer Vision*. Springer, 2012, pp. 376–389.
- [5] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4885–4891.
- [6] S. Gong, H. Zhou, F. Xue, C. Fang, Y. Li, and Y. Zhou, "FastRoadSeg: Fast Monocular Road Segmentation Network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 505–21 514, 2022.
- [7] J.-Y. Sun, S.-W. Kim, S.-W. Lee, Y.-W. Kim, and S.-J. Ko, "Reverse and boundary attention network for road segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [8] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor," *IEEE transactions on intelligent transportation systems*, vol. 19, no. 12, pp. 3981–3991, 2018.
- [9] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3D-LIDAR and camera," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5631–5637.
- [10] F. Schilling, X. Chen, J. Folkesson, and P. Jensfelt, "Geometric and visual terrain classification for autonomous mobile navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2678–2684.

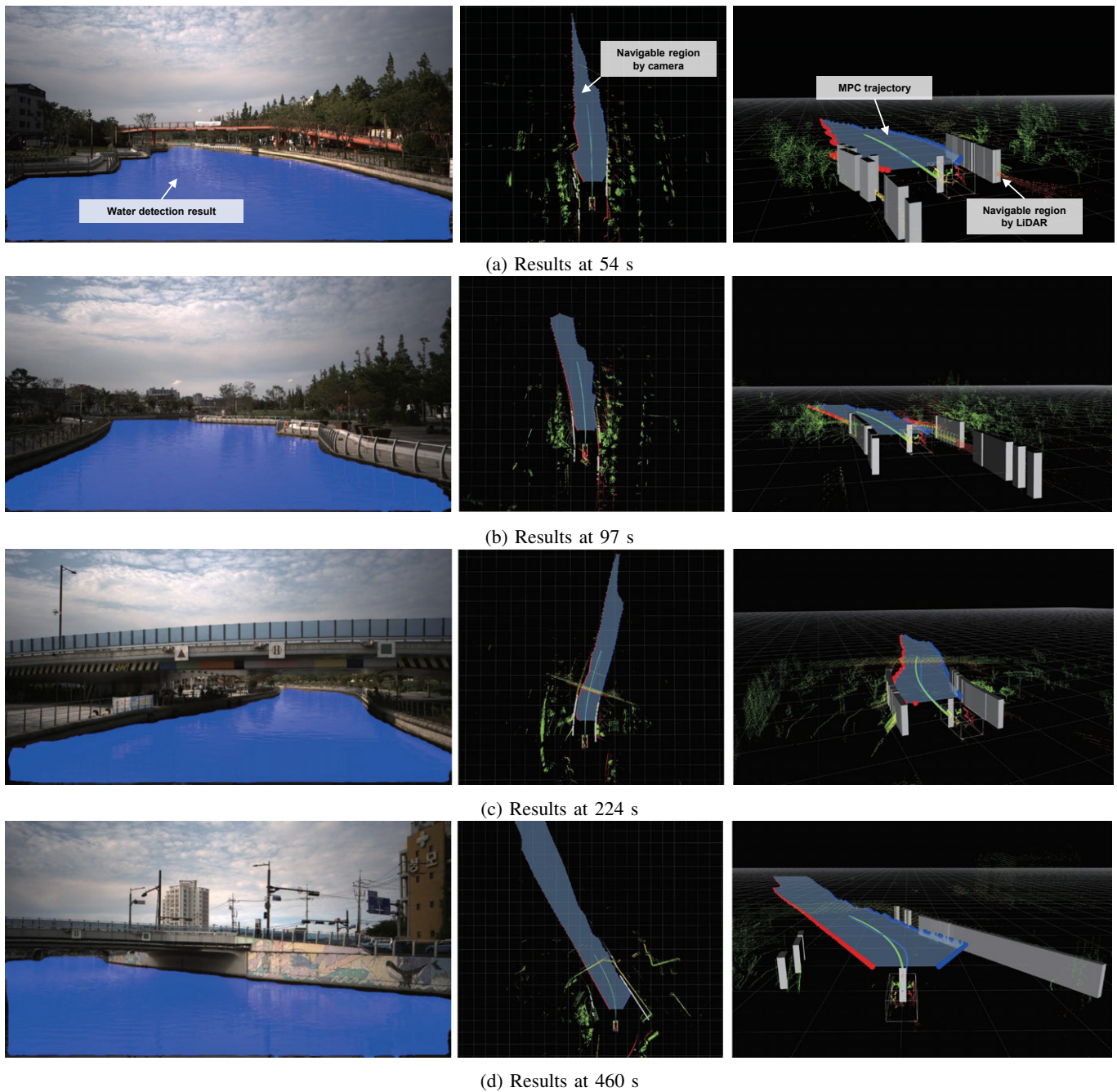


Fig. 7: Sample results of camera and LiDAR-based navigable region detection and corresponding MPC-based path planning in image (left), top-view (middle), and perspective-view (right)

[11] S. Gu, Y. Zhang, J. Yang, J. M. Alvarez, and H. Kong, "Two-view fusion based convolutional neural network for urban road detection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6144–6149.

[12] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LiDAR-camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.

[13] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nüchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 540–555, 2013.

[14] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. Velasquez, V. A. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "Wayfast: Navigation with predictive traversability in the field," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 651–10 658, 2022.

[15] S. Matsuzaki, H. Masuzawa, and J. Miura, "Image-Based Scene Recognition for Robot Navigation Considering Traversable Plants and Its Manual Annotation-Free Training," *IEEE Access*, vol. 10, pp. 5115–5128, 2022.

[16] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathymoorthy, K. Weerakoon, and D. Manocha, "GA-Nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[17] T. Guan, Z. He, R. Song, D. Manocha, and L. Zhang, "TNS: Terrain

- traversability mapping and navigation system for autonomous excavators," *Proceedings of Robotics: Science and Systems, New York City, NY, USA*, 2022.
- [18] X. Li, K. Guo, T. Jia, and X. Zhang, "Visual perception and navigation of security robot based on deep learning," in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2020, pp. 1216–1221.
- [19] K. Zhang, F. Niroui, M. Ficocelli, and G. Nejat, "Robot navigation of environments with unknown rough terrain using deep reinforcement learning," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2018, pp. 1–7.
- [20] J. Bao, X. Yao, H. Tang, and A. Song, "Detecting Untraversable Regions for Navigating Mobile Robot on Pedestrian Lanes," in *International Conference on Intelligent Robotics and Applications*. Springer, 2019, pp. 661–671.
- [21] J. Shin, D. Kwak, and K. Kwak, "Model predictive path planning for an autonomous ground vehicle in rough terrain," *International Journal of Control, Automation and Systems*, vol. 19, no. 6, pp. 2224–2237, 2021.
- [22] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.
- [23] R. de Silva, G. Cielniak, G. Wang, and J. Gao, "Deep learning-based Crop Row Following for Infield Navigation of Agri-Robots," *arXiv preprint arXiv:2209.04278*, 2022.
- [24] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot monocular vision navigation based on road region and boundary estimation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1043–1050.
- [25] R. Miyamoto, Y. Nakamura, M. Adachi, T. Nakajima, H. Ishida, K. Kojima, R. Aoki, T. Oki, and S. Kobayashi, "Vision-based road-following using results of semantic segmentation for autonomous navigation," in *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*. IEEE, 2019, pp. 174–179.
- [26] K. Weerakoon, A. J. Sathyamoorthy, U. Patel, and D. Manocha, "TERP: Reliable planning in uneven outdoor environments using deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9447–9453.
- [27] B. Bovcon and M. Kristan, "WaSR—A Water Segmentation and Refinement Maritime Obstacle Detection Network," *IEEE Transactions on Cybernetics*, 2021.
- [28] L. Žust and M. Kristan, "Temporal Context for Robust Maritime Obstacle Detection," *arXiv preprint arXiv:2203.05352*, 2022.
- [29] H. Kim, J. Koo, D. Kim, B. Park, Y. Jo, H. Myung, and D. Lee, "Vision-based real-time obstacle segmentation algorithm for autonomous surface vehicle," *IEEE Access*, vol. 7, pp. 179 420–179 428, 2019.
- [30] A. Finlinson and S. Moschogiannis, "Semantic Segmentation for Multi-Contour Estimation in Maritime Scenes," in *Proceedings of the 19th ACM SIGGRAPH European Conference on Visual Media Production*, 2022, pp. 1–10.
- [31] W. He, X. Jiang, and G. Jin, "RaRWS: A Radar-assisted Real-time Water Segmentation Network to Meet the Autonomous Navigation of USV in Inland Waterways," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 4168–4174.
- [32] W. Zhang, C. Wu, and Z. Bao, "SA-BiSeNet: Swap attention bilateral segmentation network for real-time inland waterways segmentation," *IET Image Processing*, vol. 17, no. 1, pp. 166–177, 2023.
- [33] L. Yao, D. Kanoulas, Z. Ji, and Y. Liu, "ShorelineNet: an efficient deep learning approach for shoreline semantic segmentation for unmanned surface vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5403–5409.
- [34] X. Chen, Y. Liu, and K. Achuthan, "WODIS: water obstacle detection network based on image segmentation for autonomous surface vehicles in maritime environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.
- [35] X. Yao, Y. Shan, J. Li, D. Ma, and K. Huang, "LiDAR based navigable region detection for unmanned surface vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3754–3759.
- [36] Y. Shan, X. Yao, H. Lin, X. Zou, and K. Huang, "Lidar-based stable navigable region detection for unmanned surface vehicles," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.
- [37] K. Kim and J. Kim, "Semantic segmentation of marine radar images using convolutional neural networks," in *OCEANS 2019-Marseille*. IEEE, 2019, pp. 1–6.
- [38] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz, "Stereo vision-based navigation for autonomous surface vessels," *Journal of Field Robotics*, vol. 28, no. 1, pp. 3–18, 2011.
- [39] G. Bitar, B.-O. H. Eriksen, A. M. Lekkas, and M. Breivik, "Energy-optimized hybrid collision avoidance for ASVs," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2522–2529.
- [40] M. Abdelaal, M. Fränzle, and A. Hahn, "Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances," *Ocean Engineering*, vol. 160, pp. 168–180, 2018.
- [41] S. Helling, C. Roduner, and T. Meurer, "On the dual implementation of collision-avoidance constraints in path-following MPC for underactuated surface vessels," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3366–3371.
- [42] M. Lutz and T. Meurer, "Optimal trajectory planning and model predictive control of underactuated marine surface vessels using a flatness-based approach," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 4667–4673.
- [43] M. Kosch, A. Elkhshap, P. Koschorrek, R. Zweigel, and D. Abel, "Hardware-in-the-loop trajectory tracking and collision avoidance of automated inland vessels using model predictive control," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 2251–2256.
- [44] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6189–6196.
- [45] A. Gonzalez-Garcia, I. Collado-Gonzalez, R. Cuan-Urquizo, C. Sotelo, D. Sotelo, and H. Castañeda, "Path-following and LiDAR-based obstacle avoidance via NMPC for an autonomous surface vehicle," *Ocean Engineering*, vol. 266, p. 112900, 2022.
- [46] W. Wang, B. Gheneti, L. A. Mateos, F. Duarte, C. Ratti, and D. Rus, "Roboat: An autonomous surface vehicle for urban waterways," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6340–6347.
- [47] E. Alcalá, V. Puig, J. Quevedo, and U. Rosolia, "Autonomous racing using linear parameter varying-model predictive control (LPV-MPC)," *Control Engineering Practice*, vol. 95, p. 104270, 2020.
- [48] J. P. Allamaa, P. Listov, H. Van der Auweraer, C. Jones, and T. D. Son, "Real-time nonlinear mpc strategy with full vehicle validation for autonomous driving," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 1982–1987.
- [49] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
- [50] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [51] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.
- [52] Y. Liang, N. Jafari, X. Luo, Q. Chen, Y. Cao, and X. Li, "WaterNet: An adaptive matching pipeline for segmenting water with volatile appearance," *Computational Visual Media*, vol. 6, no. 1, pp. 65–78, 2020.
- [53] B. Haipalmaa, N. Passalis, H. Zhang, M. Gabbouj, and J. Raitoharju, "High-resolution water segmentation for autonomous unmanned surface vehicles: A novel dataset and evaluation," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2019, pp. 1–6.
- [54] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.