An Adaptive Image Thresholding Algorithm Using Fuzzy Logic for Autonomous Underwater Vehicle Navigation

I-Chen Sang and William R. Norris, *Member, IEEE*

***Abstract*—Breakthroughs in autonomous vehicle technology have ignited diverse topics within engineering research. Among these, the focus on conducting inspections through autonomous underwater vehicles (AUVs) stands out as particularly influential, owing to the substantial investments directed towards offshore infrastructures. Leveraging the capabilities of onboard sensors, AUVs hold the potential to adeptly trace and examine pipelines with high levels of accuracy. However, the complicated and varying underwater environment presents a formidable challenge to ensuring the robustness of the localization and navigation framework. In response to these challenges, this study introduces a novel GPS-denied, adaptive, vision-based navigation framework tailored specifically for AUV inspection tasks. Different from conventional approaches involving manual parameter tuning, this framework dynamically adjusts contrast enhancement and edge detection functions based on incoming frame data. Fuzzy inference systems (FIS) have been harnessed within both image processing and the navigation algorithm, strengthening the overall robustness of the system. The verification of the proposed framework took place within a simulation environment. Through the implemented algorithm, the AUV adeptly identified, approached, and traversed the pipeline. Additionally, the framework distinctly showcased its capacity to dynamically adjust parameters, reduce processing time, and uphold consistency amid diverse illuminations and levels of noise.**

***Index Terms*—Autonomous underwater vehicle, navigation, image processing, adaptive, fuzzy logic.**

NOMENCLATURE

Abbreviations

CDF	Cumulative density function.
DR	Dynamic range.
FIS	Fuzzy inference system.
PDF	Probability density function.
PE	Plateau equalization.

Symbols

μ	Membership functions.
A	Incoming image.
COA	Center of area of defuzzification function.
D	Total image derivative.
D_x	Image derivative on x-axis.
D_y	Image derivative on y-axis.
G	Gray level value before contrast enhancement.
G_{New}	Gray level value after contrast enhancement.

I. INTRODUCTION

DUE to rapid developments in autonomous system technology, the research community is actively delving into the diverse applications of autonomous vehicles, including underwater exploration [1], data collection [2], [3], and surveillance. One of the most challenging topics is conducting the inspection of underwater pipelines with autonomous underwater vehicles (AUVs). With the onboard sensors and computer, AUVs exhibit remarkable potential to conduct the inspection autonomously, thus alleviating the need for extensive human involvement [4], [5] [6], [7] and significantly enhancing accuracy [8], [9].

The sensors used in AUVs are crucial for the inspection task. [10] Given the limited GPS access underwater, AUVs' navigation relies on Doppler Velocity Log (DVL), array sonar, and cameras. Among all sensor candidates, cameras of various wavebands are the most common option due to their competitive price and the amount of information obtained. A wide variety of research has been published on camera-based AUV navigation. Previously non-learning-based methods were widely explored. Studies [11], [12], [13], [14] used traditional edge detection algorithms to detect landscapes. Authors in [15] adopted SIFT to match features in figures and conducted visual odometry calculation.

Traditional methods usually require manual parameter tuning. Given the highly dynamic environment underwater, the adaptability of the algorithms became an important topic. This could be resolved by adopting learning-based methods. Algorithms proposed in [16] and [17] used learning-based algorithms to identify the target that the AUV needed to follow. The study in [1] adopted supervised learning to detect cave lines. The work proposed in [18] proposed path planning algorithms using deep reinforcement learning with consideration of complicated environment parameters.

Though learning-based methods achieved satisfactory results, the generalizability of models and the difficulty in collecting datasets posed extra challenges to the aforementioned networks.

Manuscript received 31 August 2023; revised 14 April 2024; accepted 5 July 2024. Date of publication 11 July 2024; date of current version 2 September 2024. This work was supported by TechnipFMC plc through the University of Illinois Urbana-Champaign. The guest editor coordinating the review of this article and approving it for publication was Dr. Wenbo Ding. *(Corresponding author: I-Chen Sang.)*

The authors are with the Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: ichens2@illinois.edu; wrnorris@illinois.edu).

Digital Object Identifier 10.1109/JSTSP.2024.3426484

The adaptive variations of classical approaches were widely studied.

Both preprocessing and line detection algorithms rely heavily on tunable parameters. For example, Plateau Equalization [19] has been shown to greatly improve the contrast of the images [20]. But its performance is determined by a threshold value used in normalizing the cumulative density function of the gray levels. Further adaptive variations of these approaches were proposed in various pieces of literature [21], [22], [23].

The threshold values are even more commonly used in line detection algorithms. The threshold values differentiate the edge/non-edge pixels and the line/non-line segments. These parameters can vary largely across different images. Manual calibration was required for different illumination or noise levels. The tuning process is especially difficult in a complex underwater environment where visibility could change significantly. Therefore, an automated parameter tuner can provide significant benefits for the adaptability and reliability of the system.

An expert system-based algorithm is proposed in this research to resolve the problem faced in manual adjustments of parameters. Expert systems combine the knowledge of professionals and the noise tolerance capability of fuzzy logic to achieve better system performance. Fuzzy logic has been widely used in the control field, as well as in image processing. For instance, studies [24] and [25] leveraged fuzzy entropy theory and fuzzified graylevel histograms to enhance the contrast of the images. Methods proposed in [26], [27], [28], and [29] used fuzzy set theory to improve the thresholding process. The fuzzy inference system (FIS) has been applied to image filters [30], [31] and edge/lane detection [32], [33] procedures.

In this study, an algorithm that navigates the AUV along the pipeline under various environmental parameters is proposed. The control system was integrated with image processing to increase the system's robustness. Instead of calculating the thresholds directly, the values underwent a relative adjustment according to the deviation of several key indicators. With two independent fuzzy inference systems, the algorithm automatically tuned the parameters in the contrast enhancement module and line detection module. The fuzziness of the system increased the system's adaptability to the environment. With this proposed framework, the AUV, which was released from the underwater pipeline, could successfully approach and cruise along the pipeline under different environmental noise levels. This experiment verified the capability of the navigation algorithm under various visibility conditions, thereby demonstrating its adaptability to changing environments.

This paper presents four sections arranged in the order of introduction, method, experimental results, and conclusion. All experimental setups and the algorithm design are elaborated in the method section. The experimental results, along with their statistical analysis, are listed in the result section. Finally, the summary and future works of the research are presented in the conclusion section.

II. Methodology

The navigation framework proposed in this paper enabled the AUV to approach and follow the underwater pipeline with increased robustness. This work was verified in a realistic simulation environment [34] under a closed-loop scheme. With the measurement collected from the simulation platform, the proposed algorithm selected maneuver strategies accordingly and sent the command to the simulator.

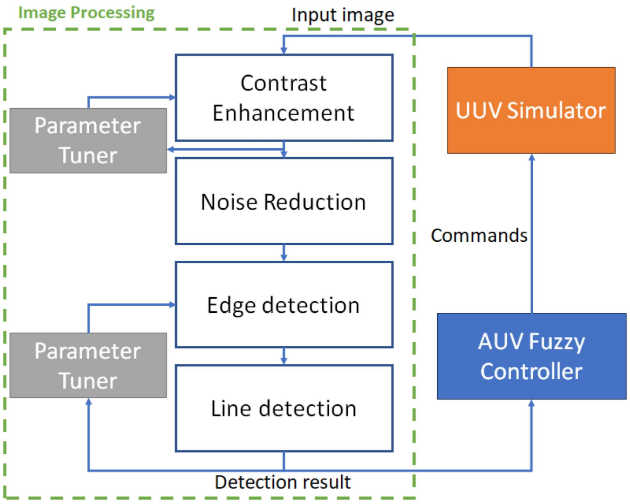


Fig. 1. Flow diagram of the proposed image processing algorithm.

The flow chart for the proposed method is shown in Fig. 1 with the image processing functions highlighted. First, the image's contrast was adjusted to expand the dynamic range. Noise reduction filters were then applied to suppress the noise level. Finally, an edge detector and a line detector were used to extract the position of the underwater pipeline. In addition, fuzzy logic was applied in this process to automatically tune the intrinsic parameters, increasing the system's adaptability over various environments.

This section explains the proposed algorithm and verification process in detail. It also presents the setup of the simulation, the theory of fuzzy logic, and the algorithm's design sequentially.

A. Fuzzy Inference System (FIS)

The fuzzy inference system contributed to the main structure of the proposed algorithm in the image processing module and navigation module. Fuzzy inference systems are also referred to as expert systems because their rulebases are constructed using the experience derived from professional knowledge. In contrast to the "black box" system derived from learning-based algorithms, the expert systems can be well explained by the rulebase.

Fuzzy inference systems (FIS) were originally applied in the control field. The inputs (navigational errors) were categorized using probability distribution functions. Based on the magnitude of the navigational errors, the pre-determined fuzzy rulebase decides the output magnitude category of the FIS. Finally, by reversing the categorization process, the numerical output commands were derived according to the output probability distribution functions.

The AUV navigation algorithm in this work was operated by a FIS. In addition, an innovative combination was proposed by applying a FIS to image parameter tuning. After identifying an

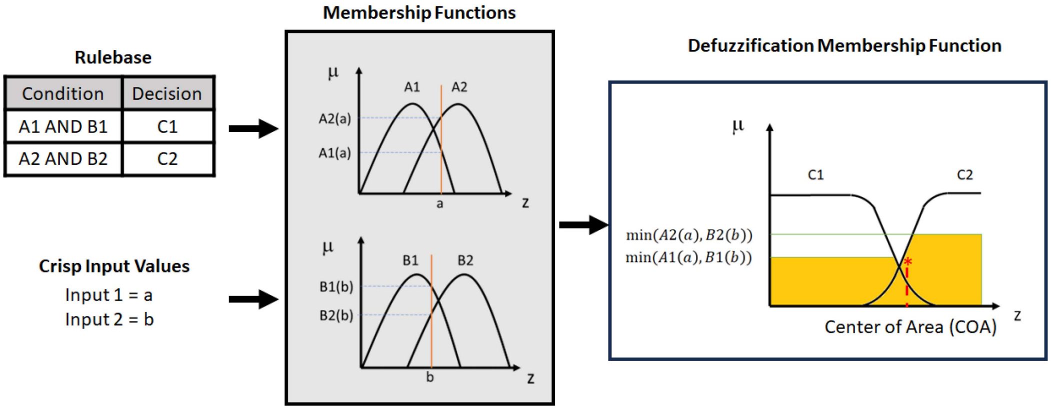


Fig. 2. Block diagram of a sample fuzzy inference system.

"observable parameter" (e.g., the brightness of the image, the standard deviation of the pixels, and the entropy of the images.), the FIS determined the direction and magnitude where the "tunable parameters" (e.g., threshold values) should be tuned.

Rather than meticulously crafting precise input-output configurations for control and parameter tuning systems, the probability nature of the FIS effectively prevented the commands from oscillating with the varying environment, thereby averting potential instabilities.

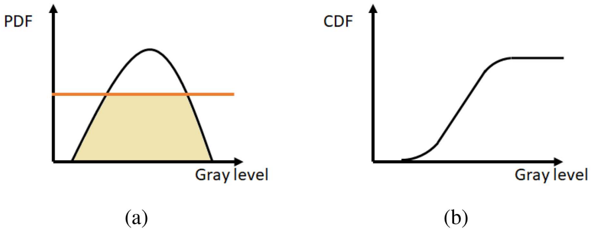
A brief introduction of the FIS is presented in this section to elaborate the decision process. The block diagram of an example fuzzy inference system is shown in Fig. 2. A Mamdani system with 'min' function used in the AND operator of the rulebase. According to the crisp input of the system, both parameters were fuzzified with the membership functions. The minimal value of the output from the membership functions was adopted for referencing the output value. The 'max' function was used in the aggregation process among the rules. The area representing the fuzzy output can be formed by combining the outputs from each rule with the maximal value. Finally, the centroid method was used to defuzzify the output value. Using the integration in (1), the centroid of the fuzzy output corresponds to the defuzzified output value.

$$COA = \frac{\int_Z \mu_A(z) z dz}{\int_Z \mu_A(z) dz} \tag{1}$$

B. Image Processing Algorithms

The image processing algorithm proposed in this study consisted of three modules – the pre-processing module (contrast enhancement and noise reduction), the edge detection module, and the line detection module. The input images were directly subscribed from the ROS topic of the front camera on the AUV. After being processed by the proposed image processing algorithm, the position of the underwater pipeline was recorded and processed by the navigation module proposed in [35].

1) Pre-Processing: The contrast between the objects was the most important indication for pipeline detection. Therefore, the images were first transformed from an RGB channel to gray level, then processed with a histogram equalization approach.



(a)

(b)

Fig. 3. Figure description of the plateau equalization algorithm. (a) The PDF of the image. The orange line (threshold value) determines the reserved area of the PDF. (b) The CDF derived from the reserved area in (a).

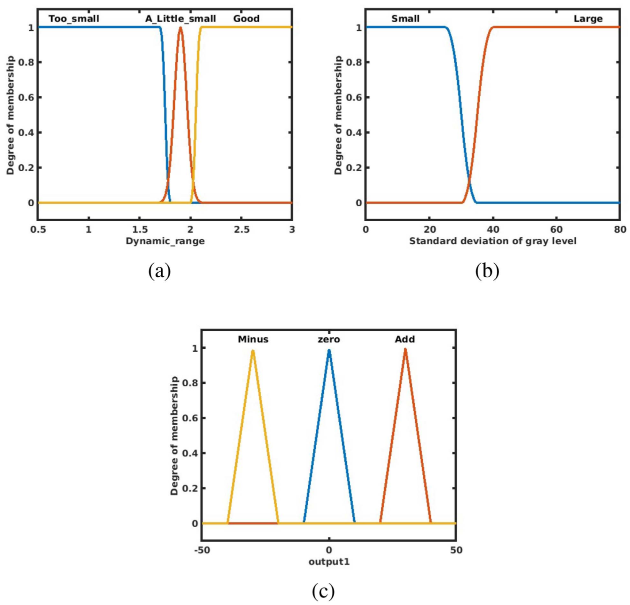
Plateau Equalization was adopted to broaden the contrast of the images. It has been shown that Plateau Equalization worked well in enhancing the contrast of grayscale images, such as thermal images [20]. A threshold value was added to the process of the classical histogram equalization algorithm. When plotting the gray-level probability density function (PDF), any count that exceeded the threshold value was cropped. The cumulative density function (CDF) was calculated according to the reserved area of the PDF. Finally, the new gray level of a pixel was

$$G_{New} = 255 \times CDF(G) \tag{2}$$

where G_{New} is the new gray level value, and G is its value in the original image. The detailed process of Plateau Equalization is shown in Fig. 3.

This process was designed to prevent the histogram equalization process from being biased by the dominating gray levels. Typically, the threshold was tuned according to various illumination and contrast conditions. To automate the threshold tuning process, the fuzzy inference system was introduced in the contrast enhancement module.

The fuzzy threshold tuner measured the output images' dynamic ranges and determined the variation value that would be applied to the threshold in the next frame. The dynamic range (DR) of an image is the logarithmic value of the ratio between the maximal and minimal gray level in the image. To equalize the contribution of all gray levels in the images, the algorithm was initiated with a lower threshold. However, when the pixels were concentrated within a few gray level values, more



(a)

(b)

(c)

Fig. 4. Membership function and linguistic parameters of (a) the dynamic range (b) the standard deviation of the graylevel. And (c) the defuzzification membership function of the output.

TABLE I
RULE BASE OF THE FUZZY CONTROLLER USED IN CONTRAST ENHANCEMENT

Dynamic Range	Stdev of Gray Level	Threshold control
Good		Minus
A Little Small		Add
Too Small	Small	Add
	Large	Minus

discontinuity would be found in the CDF, causing a reduced DR in the image. With a typical 8-bit gray level image, the highest dynamic range value achievable is 2.41. Therefore, the dynamic range was divided into three linguistic variables according to their difference with 2.41. The complete membership functions are shown in Fig. 4(a).

On the other hand, the standard deviation of the gray level counts also contributed to the threshold tuning process. When the standard deviation value was large, a variation in the adjusted gray level could be ensured even when the output image might not have a high dynamic range. As a result, the membership functions of the standard deviation with the linguistic variables are shown in Fig. 4(b). The hierarchical rule base reduction (HRBR) method [36] was used to reduce the size of the rule base according to the importance of the inputs and thereby accelerating the processing speed. With the effective HRBR method, the fuzzy rule base used in this study is shown in Table I. The adjustment value of the threshold was defuzzified using the membership function shown in Fig. 4(c)

After the fuzzy contrast enhancement process, the images underwent a 3×3 median filter to avoid the strong salt-and-pepper noise from influencing the edge detection result.

2) Edge Detection: The edge detection module proposed in this study was based on the Sobel edge detector [37], further optimized by a well-designed fuzzy inference system. The block diagram in Fig. 1 shows the workflow of the module. The preprocessed image underwent an edge-detecting process by the Sobel edge detector under default threshold values. The detected edge pixels were then processed by the line detection module to identify the candidates that best represented the pipeline. Finally, the line detection result was used as the input to the fuzzy parameter tuner in order to determine the threshold value for the next frame.

The Sobel edge detector used the standard Sobel kernels. The kernel was convoluted with the input image to calculate the derivatives of the pixels in the x and y directions, respectively. Given that the target of detection was the pipeline, the AUV was likely to have a smaller angle with the pipeline most of the time. Therefore, the derivative of the x-axis in the image was weighted more than that of the y-axis. The calculation of the overall derivative is shown with the equation below

$$D_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \tag{3}$$

$$D_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{4}$$

$$D = 1 \times D_x^2 + 0.5 \times D_y^2 \tag{5}$$

where A is the incoming image, D represents the overall pixel-wise derivative, while D_x and D_y represent the derivative on the x/y axis.

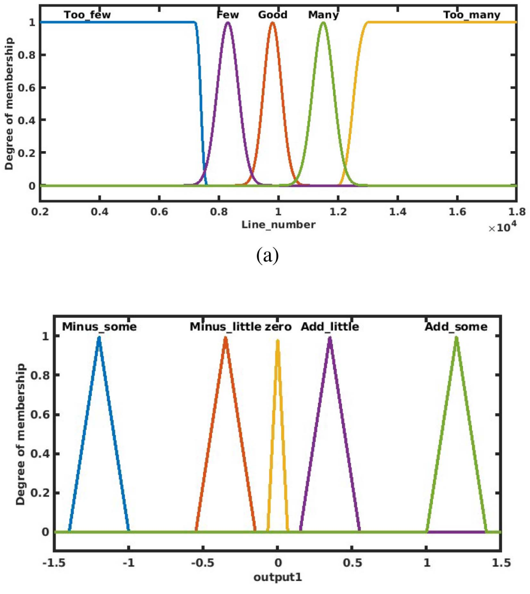
The thresholding process was crucial in the edge detection module as only the pixels with a gradient value higher than the threshold value were regarded as effective edge pixels. To adapt the algorithm to various images, the cutoff threshold was assigned referring to the pixel-wise derivatives obtained in the previous convolution process. Specifically, the product of a cutoff parameter and the mean derivative of the whole image was adopted using the threshold value. The initial value of the cutoff parameter was set to 10 in order to keep almost all candidate edges.

However, the number of candidate edge pixels can greatly influence the computation effort of the line detection module. A larger count in edge pixels resulted in a longer processing time. On the other hand, with too few edge pixels, the contour of the pipeline could not be appropriately plotted and therefore led to confusing inline detection. The cutoff parameter was adjusted by a fuzzy inference system in order to solve this issue. The input of the FIS was the number of lines detected with the Hough Transform, and the output was the variation value that would be applied to the next frame. The rulebase of this FIS is shown in Table II. When too many lines were detected, the cutoff variable was raised to eliminate pixels with smaller derivatives, and vice versa. The fuzzy membership functions of the linguistic parameters were determined by experiments, and the projections are shown in Fig. 5(a) and (b).

3) Line Detection: The widely-used line detector, Hough Transform, was adopted in this module. In the Hough Transform,

TABLE II
RULE BASE OF THE FUZZY CONTROLLER USED IN EDGE DETECTION

Detected line number	Threshold control
Too few	Minus some
Few	Minus little
Good	Zero
Many	Add little
Too many	Add some



(a)

(b)

Fig. 5. (a) Membership function and linguistic parameters of the line number. (b) Defuzzification membership function of the output.

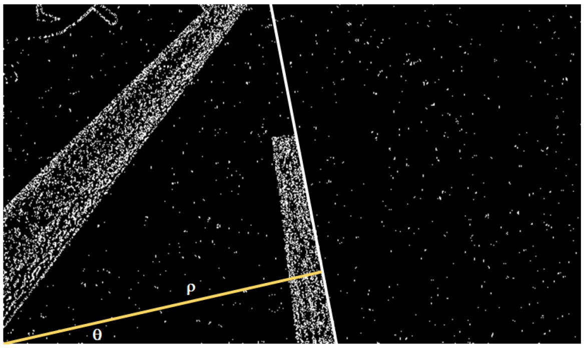


Fig. 6. Illustration of Hough Transform line detection.

virtual lines of all angles centering at every edge pixels are projected the edge pixels into the Hough space with the equation below.

$$\rho = xsin\theta + ycos\theta \tag{6}$$

where ρ is the shortest distance from the origin to a candidate line and θ is the corresponding angle of the line (see Fig. 6). The x and y shown in the equation is the coordinate of the edge pixel. Each line in the image space corresponds to a sine curve in the Hough space. If several edge pixels share the same straight line, their sine curves intersect at the same (ρ, θ) point. Therefore, by identifying the most intersected points in the Hough space, the lines that pass through the most edge pixels could be listed. These lines, therefore, have the highest probability of representing the pipeline.

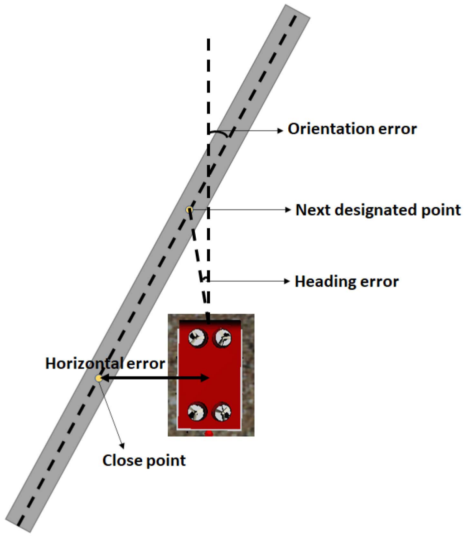


Fig. 7. Definition of the navigation errors used in the proposed model.

Since the Hough Transform iterates through all candidate edge pixels, its performance is highly dependent on the number of candidates. A proper number of candidates was maintained With the aforementioned fuzzy logic, and thereby a balance between computation resources and detection accuracy.

C. Navigation Framework

A navigation framework [35] that is based on an independent FIS was adopted in this work. The navigation error was interpreted from the incoming image data. And the FIS made the navigation decision according to the heading error, orientation error, and horizontal error.

1) Navigation of the FIS: A coordinate-based navigation framework using the three errors mentioned above was developed in [38]. In this framework, the AUV was kept at a constant speed of 0.15 m/s. By analyzing heading error, orientation error, and horizontal error, the FIS made the decision on the angular velocity applied to the vehicle.

The definition of the three errors is shown in Fig. 7. Assuming that the pipeline was marked with several coordinate points, the heading error was the angle between the AUV's heading direction and the next coordinate point on the pipeline. The orientation error was the difference between the AUV's heading and the pipeline's path. Finally, the horizontal error was defined by the closest distance between the AUV and the pipeline.

The errors were transformed into steering decisions by a hierarchical rulebase shown in Table III. And the linguistic variables with their membership functions for all parameters are shown in Fig. 8, where (a)–(c) are membership functions of the front angle error, horizontal error, and orientation error. The output defuzzification membership functions are shown in

TABLE III
RULE BASE OF THE FUZZY CONTROLLER IN THIS STUDY

Horizontal error	Heading error	Orientation error	Steering
Far left	Far left		Right3
Far left	Near left		Right2
Far left	Close left		Right1
Far left	Zero		Zero
Far left	Close right		Left1
Far left	Near right		Left2
Far left	Far right		Left3
Close left		Far left	Right3
Close left		Close left	Right2
Close left		Zero	Right1
Close left		Close right	Right1
Close left		Far right	Right1
Zero		Far left	Right2
Zero		Close left	Right1
Zero		Zero	Zero
Zero		Close right	Left1
Zero		Far right	Left2
Close right		Far left	Left3
Close right		Close left	Left2
Close right		Zero	Left1
Close right		Close right	Left1
Close right		Far right	Left1
Far right	Far left		Left3
Far right	Near left		Left2
Far right	Close left		Left1
Far right	Zero		Zero
Far right	Close right		Right1
Far right	Near right		Right2
Far right	Far right		Right3

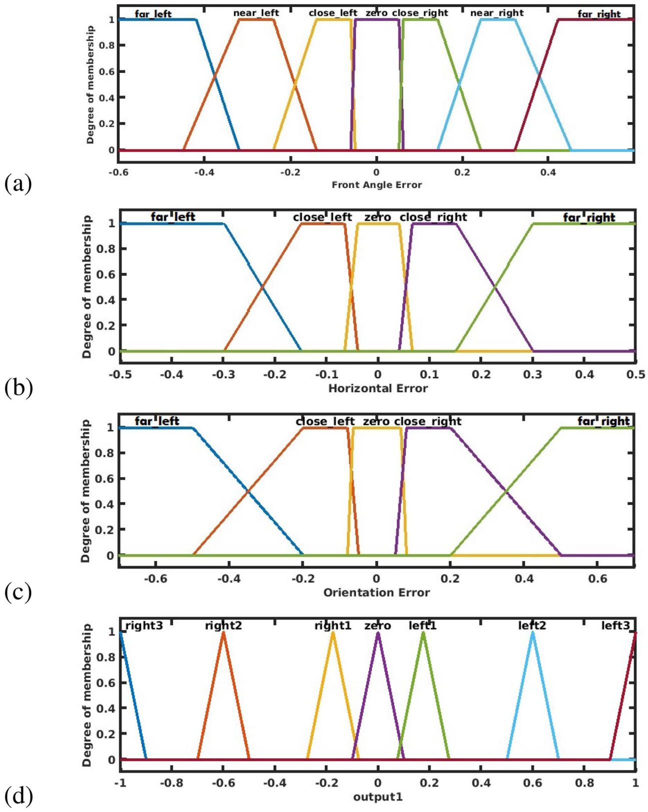


Fig. 8. Linguistic variable membership functions of the (a) heading error, (b) horizontal error, (c) orientation error, and (d) output.

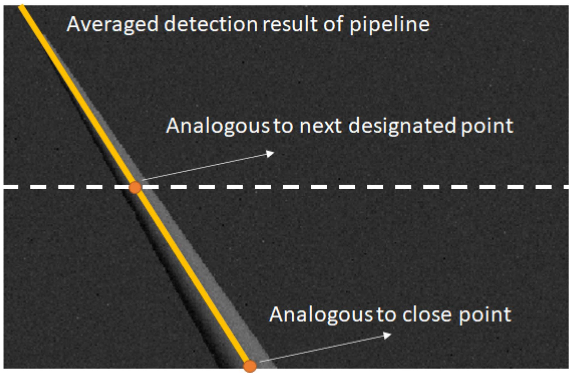

Averaged detection result of pipeline
Analogous to next designated point
Analogous to close point

Fig. 9. Image interpretation of the navigation errors used in the proposed model.

Figure 8(d). This fuzzy navigation framework has been proven effective in [35] with the presence of underwater currents.

2) Image Interpretation: In [35], the aforementioned fuzzy navigator was extended using computer vision and proximity sensors, alleviating the system's reliance on GPS signals. By interpreting the information provided by the onboard camera, the navigation errors were calculated.

The concept of the interpretation from image to navigation errors is shown in Fig. 9. Instead of having discrete coordinate points on the target path, the pipeline shown in the incoming images was continuous. Therefore, two target coordinates needed to be specifically defined.

The intersection between the detected pipeline and the bottom of the image was defined as the "close point." And the intersection between the pipeline and the center line of the image was defined as a "next designated point." Since the position and orientation of the camera relative to the AUV were known, the relative position between the AUV and these points was derived from coordinate transformation calculations. The horizontal error was defined by the relative distance between the AUV and the "close point." The heading error was defined by the difference in orientation between the AUV's heading and the vector pointing from the AUV to the "next designated point." Finally, the orientation of the pipeline was defined with the vector formed by the "next designated point" and the "close point." The orientation error was derived by measuring the angular difference between the AUV's heading and the orientation of the pipeline. The detailed image interpretation illustration is shown in Fig. 9.

III. RESULTS

Several simulation experiments were conducted to verify the algorithm's capabilities in optimizing the image and reducing navigation errors. The screenshot of the simulation software and the block diagram demonstrating the framework of the simulation are shown in Fig. 10. The detail of the experimental setup is presented below. The experiment results demonstrate the image

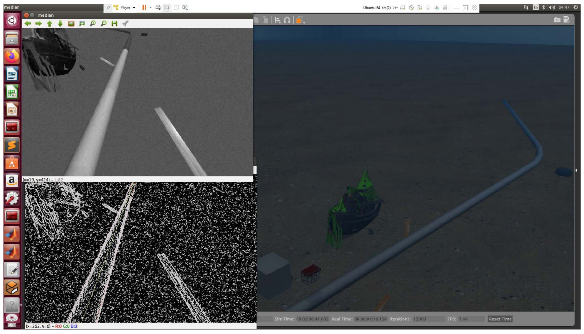


Fig. 10. Screenshot of the simulation environment.

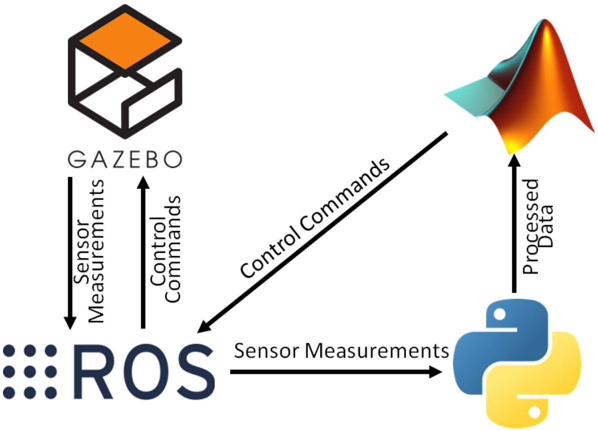


Fig. 11. Flow diagram of the proposed image processing algorithm.

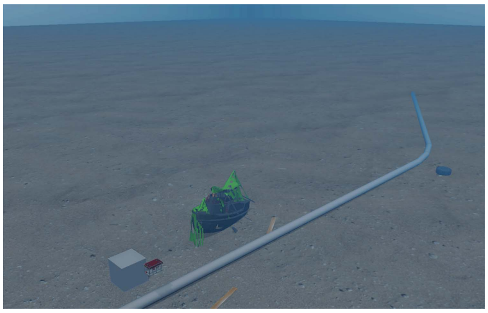
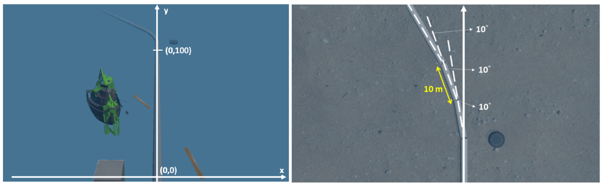


Fig. 12. Underwater scene designed to verify the proposed algorithm.



(a)

(b)

Fig. 13. (a) Full view of the pipeline used to verify the proposed algorithm. (b) Detailed dimension of the gradual turn in the pipeline.

improvement, navigation system robustness, and optimization of the AUV navigation will be shown in the following parts.

A. Simulation Environment

The experiment was done in a virtual machine operating under a 64-bit Ubuntu 16.04 system with 12 GB RAM. The closed-loop framework of the algorithm consisted of ROS (Robot Operating System), Gazebo, Python, and Matlab. A simulated underwater scene and an Autonomous Underwater Vehicle were created in Gazebo [34]. The data measured by the simulated sensors on the vehicle were subscribed using ROS and processed by Python code. Retrieving the processed result from the ROS core, the Simulink-based function made maneuvering decisions by the fuzzy logic package in Matlab software. Finally, the decisions were published to ROS core in order to control the vehicle. The whole framework and the communication between each module are shown in Fig. 11.

The simulated underwater scene is shown in Fig. 12. A pipeline with a diameter of 2 meters was placed at the center of the world coordinate. Several random objects were also placed in the test scene to raise the complexity, including the ones with straight edges (the cube and boards) and the ones with irregular edges (the shipwreck and the tire). The underwater pipeline was designed with a left turn in its path. To precisely locate every point on it, the gradual turn was created by combining three segments of cylinders. The geometrical relation between the segments is shown in Fig. 13, where (a) and (b) illustrate the global and local view respectively. The segments contributing to the gradual turn were 10 meters long. And the angle between adjacent pieces was 10°.

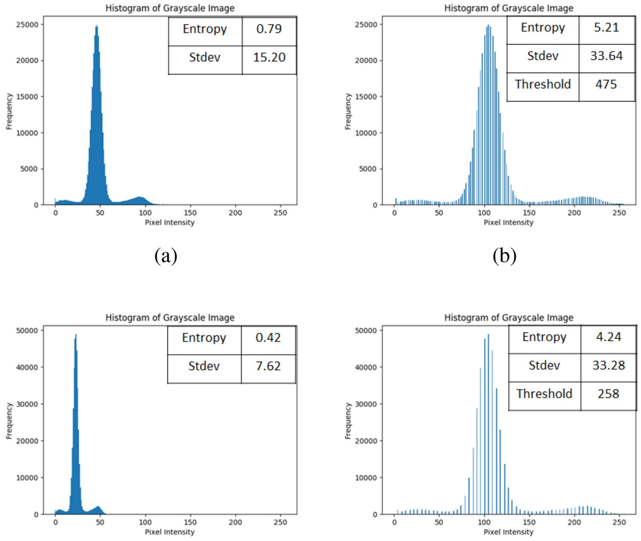
According to [34], the difficulties often encountered in complicated underwater environments were also presented in the designed scene. The environmental noise and the velocity of the underwater current could be tuned to a higher value to simulate highly unstable environments. In addition, the attenuation of light in different wavelengths was also included in this simulation model, making it more realistic. However, the limitation of this model still exists. For example, the damping light source caused by the periodic waves could not be simulated yet.

According to [39], a reasonable visibility range used in underwater vision systems is around 10 m. As a result, this work also assumed that the AUV was launched away from the pipeline at a distance closer than the visibility range. Within this visibility range, the proposed system optimized the image processing parameters thereby improving the accuracy of navigation.

In addition to visibility levels, sea currents were also available in the simulator. Experiments done in [35] showed that the AUV was able to maintain a navigation error of 2.5 m under 1.2 m/s current speed, proving that the visibility assumption was reasonable.

B. The Simulated AUV

The vehicle used in the simulation experiment was designed by [34] as well. With the size of 1.5 (W) × 2.6 (L) × 1.6 (H) m^3, the vehicle was equipped with eight actuators and seven types of sensors. The sensor used in this study is the color camera with a resolution of 768 × 492 and a 124.2 × 100.8 degrees field of view. The camera was mounted at the coordinate of (1.15,0,0.4) relative to the origin of the AUV, with the line of sight 0.6 rd



(a) (b)

(c) (d)

Fig. 14. Graylevel probability density function (pdf) of images under (a), (b) normal illumination level, and (c) (d) dark scenario. (a) and (c) Shows the pdf of the raw input image while Figure (b) and (d) shows the pdf after being processed by the contrast enhancement module.

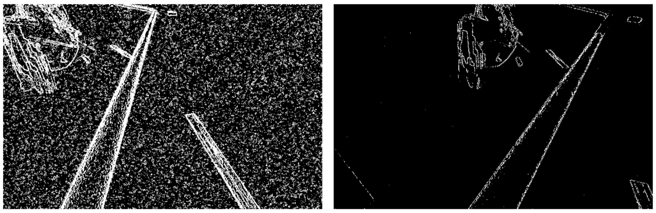
pointing downward relative to the x-axis (front) of the vehicle. The localization and dynamical model developed by [38] was adopted in this work.

C. Contrast Improvement

The capabilities of the image improvement module were demonstrated independently by examining the input and output images with their gray-level probability density functions. Experiments were conducted to demonstrate the fuzzy thresholding algorithm's robustness over various brightness conditions.

Under normal (default) system illumination levels, the contrast adjustment algorithm was performed. The experimental result is shown in Fig. 14. The gray-level probability density function (pdf) of the input image and the output image under normal conditions are shown in Fig. 14(a) and (b). Starting at an initial value of 5, the contrast enhancement threshold was tuned and stabilized at the level of 450. In the example figure shown in Fig. 14(b), the threshold value was 475, successfully increasing the range of the pdf, entropy, as well as the standard deviation value of the image. As common metrics in describing the information included in images, the increase in the entropy and the standard deviation of the image indicated a better image information density.

The experiment was repeated in the "dark scenario", where all input pixel values were halved. The experimental result is shown in Fig. 14(c) and (d). Different from the threshold value in the normal illumination experiment, the fuzzy-based system stabilized at the level of 200. In the example shown in Fig. 14(d), the contrast of the image was significantly improved by the module applying a threshold of 258. A notable increase in the entropy and the standard deviation of the image was observed, indicating a better image contrast.



(a) (b)

Fig. 15. Edge pixels that were filtered at the (a) 1^{st}. (b) 30^{th} frame. The tuned threshold values were 7.9 and 100.1, respectively.

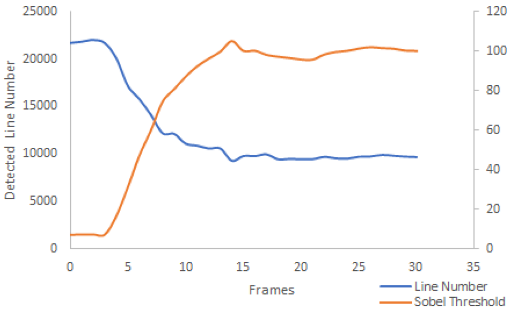


Fig. 16. Plot illustrates the relationship between the detected line numbers, the tuned threshold values, and the respective frame number.

D. Processing Time Improvement

As mentioned above, the threshold value in the Sobel edge detection function filters the most reliable edge pixels, thereby reducing the processing time of the Hough Transform. The image before and after the filtering process is shown in Fig. 15(a) and (b). It was shown that the threshold value of 100 successfully filters most of the noise pixels, leaving a clear contour of the pipeline and other objects.

The variation of the detected line number and the tuned threshold value is shown in Fig. 16. It is clear that the threshold was raised according to the high number of detected lines. The system was stabilized within 20 frames after the fuzzy tuner was activated.

In addition to the comparison of the image quality, the processing time was further examined to demonstrate the improvement in efficiency. Without the fuzzy-tuned Sobel threshold, the Hough Transform took an average of 53 ms to process the image. The fuzzy tuner successfully decreased the processing time to 8.30 ms per frame. This improvement effectively tackled the preexisting efficiency problem of the Hough Transform and broadened its field of application.

An additional experiment was performed to evaluate the processing time of the fuzzy tuner. In order to focus on the efficiency of the proposed algorithm, the measurement was conducted excluding the simulator and ROS environment. With fixed threshold values applied to the contrast enhancement and edge detection modules, the overall processing time was 26.24 ms/frame. After incorporating the fuzzy parameter tuner, the processing time only increased by 2.09%, to 26.80 ms/frame.

TABLE IV
RMS NAVIGATION ERROR OF THE AUV UNDER VARIOUS CAMERA NOISE LEVELS

Noise level	With fuzzy tuner	Without fuzzy tuner
0.05	2.52	Diverged
0.04	2.44	2.74
0.03	2.27	2.31
0.02	2.28	2.26

Though more time was consumed by the fuzzy tuner, the adaptability contributed by the module and the time it saved in the Hough Transform was more important.

E. System's Robustness Over Noise Levels

The whole fuzzy navigation framework could steer the AUV toward the pipeline with high reliability. A repetitive experiment was done to demonstrate the stability of the navigation system. The root mean square (rms) error of the navigation error was 2.31 ± 0.04 m.

To further demonstrate the system's reliability in complex environments, higher noise levels were applied to the underwater scene. While the pixel readout of the grayscale image was normalized, Gaussian noise was applied with the average set to zero. A comparison between the proposed algorithm and the benchmark (standard Plateau Equalization and Sobel Edge detector) was included in this experiment. The standard deviation of the noise was varied in order to simulate different visibility levels. The corresponding rms navigation error of the AUV is shown in Table IV. Following the previous experiment, the AUV also started from 10 m away from the pipeline with a linear velocity of 0.15 m/s. As expected, the navigation error increased with the noise level. When the standard deviation of the noise was raised to 0.05, the system that did not incorporate the fuzzy tuner failed to navigate along the pipeline. On the other hand, the system with the fuzzy tuner could still navigate the AUV toward the pipeline successfully, demonstrating the robustness of the system.

IV. CONCLUSION

In this study, it was shown that the combination of image processing modules with fuzzy inference systems successfully solved the problem of parameter tuning in existing algorithms. The proposed framework demonstrated its adaptability across diverse noise levels and illumination conditions, resulting in enhanced performance in AUV navigation. Additionally, through experimentation with a highly realistic UUV simulator, the algorithm showcased its potential applicability in real-world scenarios.

Though demonstrated adaptability in complicated environments, the proposed work faces two potential limitations: visibility range and processing efficiency. In this study, a visibility range of 10 meters was assumed. Addressing lower visibility scenarios may require further adaptation and potentially additional sensors. Additionally, the inclusion of the fuzzy parameter tuner resulted in a slight processing time increase (2%). However, this impact was deemed insignificant as the AUV operates at low speeds. The fuzzy tuner effectively optimized the computational effort of existing modules, such as the Hough Transform line detector.

There are numerous avenues for future research based on the proposed algorithm. For example, further exploration into identifying 'observable parameters' holds promise. While this study tested multiple parameters before selecting standard deviation, dynamic range, and detected line number, this process can be refined systematically.

As mentioned in the introduction, many computer vision systems require parameter tuning in different application scenarios. This finding has significant potential for application to more image processing algorithms. Similar to the work done in this paper, the thresholding process of other contrast enhancement methods or gradient detection approaches is expected to be greatly improved using fuzzy inference systems. This can be applied to on-road vehicles and significantly improve the detection results [40]. Moreover, the application to other computer vision approaches is promising. For instance, the kernel sizes used in convolutional processes could be adjusted with adaptive control systems.

Other than the wider applications of the fuzzy parameter tuner, the adaptive optimization of the fuzzy inference systems is being developed in an ongoing research project.

REFERENCES

[1] B. Yu, R. Tibbetts, T. Barna, A. Morales, I. Rekleitis, and M. J. Islam, "Weakly supervised caveline detection for AUV navigation inside underwater caves," in *Proc. 2023 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2023, pp. 9933–9940.

[2] A. A. Al-Habob, O. A. Dobre, and H. V. Poor, "Age-optimal information gathering in linear underwater networks: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13129–13138, Dec. 2021.

[3] X. Hou, J. Wang, T. Bai, Y. Deng, Y. Ren, and L. Hanzo, "Environment-aware AUV trajectory design and resource management for multi-tier underwater computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 474–490, Feb. 2023.

[4] J. Agbakwuru, "Oil/gas pipeline leak inspection and repair in underwater poor visibility conditions: Challenges and perspectives," *J. Environ. Protection*, vol. 03, pp. 394–399, 2012.

[5] M. A. Adegboye, W.-K. Fung, and A. Karnik, "Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches," *Sensors*, vol. 19, no. 11, 2019, Art. no. 2548. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2548

[6] M. Jacobi and D. Karimanzira, "Underwater pipeline and cable inspection using autonomous underwater vehicles," in *Proc. MTS/IEEE OCEANS - Bergen*, 2013, pp. 1–6.

[7] M. Jacobi and D. Karimanzira, "Multi sensor underwater pipeline tracking with AUVs," in *Proc. IEEE Oceans*, 2014, pp. 1–6.

[8] N. Mohamed, I. Jawhar, J. Al-Jaroodi, and L. Zhang, "Sensor network architectures for monitoring underwater pipelines," *Sensors*, vol. 11, no. 11, pp. 10738–10764, 2011. [Online]. Available: https://www.mdpi.com/1424-8220/11/11/10738

[9] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "An architecture for using autonomous underwater vehicles in wireless sensor networks for underwater pipeline monitoring," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1329–1340, Mar. 2019.

[10] B. Zhang, D. Ji, S. Liu, X. Zhu, and W. Xu, "Autonomous underwater vehicle navigation: A review," *Ocean Eng.*, vol. 273, 2023, Art. no. 113861.

[11] G. Foresti, "Visual inspection of sea bottom structures by an autonomous underwater vehicle," *IEEE Trans. Syst., Man, Cybern., Part B. (Cybern.)*, vol. 31, no. 5, pp. 691–705, Oct. 2001.

[12] C.-C. Cheng and B.-T. Jiang, "A robust visual servo scheme for underwater pipeline following," in *Proc. IEEE 19th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, 2012, pp. 456–459.

[13] G. Allibert, M.-D. Hua, S. Krupínski, and T. Hamel, "Pipeline following by visual servoing for autonomous underwater vehicles," *Control Eng. Pract.*, vol. 82, pp. 151–160, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066118306312

[14] A. Ortiz, M. Simó, and G. Oliver, "A vision system for an underwater cable tracker," *Mach. Vis. Appl.*, vol. 13, no. 3, pp. 129–140, 2002.

[15] B. M. Nordfeldt-Fiol, F. Bonin-Font, and G. Oliver, "Evolving real-time stereo odometry for AUV navigation in challenging marine environments," *J. Intell. Robotic Syst.*, vol. 108, no. 4, p. 83, 2023.

[16] Y. Liu, F. Wang, Z. Lv, K. Cao, and Y. Lin, "Pixel-to-action policy for underwater pipeline following via deep reinforcement learning," in *Proc. IEEE Int. Conf. Intell. Robotic Control Eng. (IRCE)*, 2018, pp. 135–139.

[17] S. A. Fjerdingen, E. Kyrkjeboe, and A. A. Transeth, "AUV pipeline following using reinforcement learning," in *Proc. IEEE ISR 2010 (41st Int. Symp. Robotics) ROBOTIK 2010 (6th German Conf. Robotics)*, 2010, pp. 1–8.

[18] W. Wei, J. Wang, J. Du, Z. Fang, Y. Ren, and C. L. P. Chen, "Differential game-based deep reinforcement learning in underwater target hunting task," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 27, 2023, doi: 10.1109/TNNLS.2023.3325580.

[19] V. E. Vickers, "Plateau equalization algorithm for real-time display of high-quality infrared imagery," *Opt. Eng.*, vol. 35, no. 7, pp. 1921–1926, 1996.

[20] S. Li, W. Jin, L. Li, and Y. Li, "An improved contrast enhancement algorithm for infrared images based on adaptive double plateaus histogram equalization," *Infrared Phys. Technol.*, vol. 90, pp. 164–174, 2018.

[21] Y. Jin, L. M. Fayad, and A. F. Laine, "Contrast enhancement by multiscale adaptive histogram equalization," in *Proc. Wavelets, Appl. Signal Image Process. IX*, 2001, vol. 4478, pp. 206–213.

[22] Y. Zhu and C. Huang, "An adaptive histogram equalization algorithm on the image gray level mapping," *Phys. Procedia*, vol. 25, pp. 601–608, 2012.

[23] M. S. Hitam, E. A. Awalludin, W. N. J. H. W. Yussof, and Z. Bachok, "Mixture contrast limited adaptive histogram equalization for underwater image enhancement," in *Proc. IEEE Int. Conf. Comput. Appl. Technol. (ICCAT)*, 2013, pp. 1–5.

[24] H.-D. Cheng and H. Xu, "A novel fuzzy logic approach to contrast enhancement," *Pattern Recognit.*, vol. 33, no. 5, pp. 809–819, 2000.

[25] S. Di Zenzo, L. Cinque, and S. Levialdi, "Image thresholding using fuzzy entropies," *IEEE Trans. Systems, Man, Cybern., Part B (Cybern.)*, vol. 28, no. 1, pp. 15–23, Feb. 1998.

[26] Y. Yong, Z. Chongxun, and L. Pan, "A novel fuzzy c-means clustering algorithm for image thresholding," *Meas. Sci. Rev.*, vol. 4, no. 1, pp. 11–19, 2004.

[27] H. R. Tizhoosh, "Image thresholding using type II fuzzy sets," *Pattern Recognit.*, vol. 38, no. 12, pp. 2363–2372, 2005.

[28] W. Wu, X. Yang, H. Li, K. Liu, L. Jian, and Z. Zhou, "A novel scheme for infrared image enhancement by using weighted least squares filter and fuzzy plateau histogram equalization," *Multimedia Tools Appl.*, vol. 76, pp. 24789–24817, 2017.

[29] F. Zhou, Z. Jia, J. Yang, and N. Kasabov, "Method of improved fuzzy contrast combined adaptive threshold in NSCT for medical image enhancement," *BioMed Res. Int.*, vol. 2017, 2017, Art. no. 3969152.

[30] C.-C. Kang and W.-J. Wang, "Fuzzy reasoning-based directional median filter design," *Signal Process.*, vol. 89, no. 3, pp. 344–351, 2009.

[31] M. Habib, A. Hussain, and T.-S. Choi, "Adaptive threshold based fuzzy directional filter design using background information," *Appl. soft Comput.*, vol. 29, pp. 471–478, 2015.

[32] N. Mathur, S. Mathur, and D. Mathur, "A novel approach to improve sobel edge detector," *Procedia Comput. Sci.*, vol. 93, pp. 431–438, 2016.

[33] G. Reina and A. Milella, "FLane: An adaptive fuzzy logic lane tracking system for driver assistance," *J. Dyn. Syst., Meas., Control*, vol. 133, no. 2, 2011, Art. no. 021002.

[34] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *Proc. MTS/IEEE OCEANS Monterey*, 2016, pp. 1–8.

[35] I.-C. Sang and W. R. Norris, "An autonomous underwater vehicle simulation with fuzzy sensor fusion for pipeline inspection," *IEEE Sensors J.*, vol. 23, no. 8, pp. 8941–8951, Apr. 2023.

[36] W. R. Norris, "A design framework for qualitative human-in-the-loop system development," Ph.D. dissertation, Univ. Illinois Urbana-Champaign, 2001.

[37] R. Jain et al., *Mach. Vis.*, vol. 5, New York, NY, USA: McGraw-Hill, 1995, pp. 140–185.

[38] A. Rajput, "Steering control and kalman filter position estimation comparison for an autonomous underwater vehicle," Master's thesis, Dept. Systems. Eng., Univ. Illinois at Urbana-Champaign, Champaign, IL, USA, 2021.

[39] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *Proc. IFAC Conf. Manoeuvering Control Mar. Craft*, 2006, vol. 88, pp. 1–12.

[40] I.-C. Sang and W. R. Norris, "A robust lane detection algorithm adaptable to challenging weather conditions," *IEEE Access*, vol. 12, pp. 11185–11195, 2024.

I-Chen Sang received the B.S. and M.S. degrees in physics from National Tsing-Hua University, Hsinchu, Taiwan, in 2014 and 2016, respectively. Having five-years of experience in industry as a systems engineer, she is currently a Ph.D. student in the Systems Engineering program with the University of Illinois Urbana-Champaign, Champaign, IL, USA. She is also a Research Assistant with the Autonomous and Unmanned Vehicle Systems Laboratory (AUVSL). Her research interests include the navigation of autonomous vehicles, sensor fusion, and image processing.

William R. Norris (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in systems engineering from the University of Illinois Urbana-Champaign, Champaign, IL, USA, in 1996, 1997, and 2001, respectively, and the MBA degree from The Fuqua School of Business, Duke University, Durham, NC, USA, in 2007. He has more than 23 years of industry experience with autonomous systems. He is currently a Research Professor with the Industrial and Enterprise Systems Engineering Department, University of Illinois Urbana-Champaign, Director of the Autonomous and Unmanned Vehicle System Lab (AUVSL), and the Founder and Director of the Center of Excellence for Autonomous Construction and Manufacturing at Scale (CEACaMS).