



Dynamic region of interest generation for maritime horizon line detection using time series analysis

Ahmet Agaoglu¹ · Nezih Topaloglu¹

Accepted: 13 December 2024 / Published online: 8 January 2025
© The Author(s) 2024

Abstract

Maritime horizon line detection is critical in applications such as marine navigation and ocean surveillance. This study introduces a novel system that leverages dynamic region of interest (ROI) generation and time series analysis to improve both accuracy and computational efficiency. By employing parallelogram-shaped ROIs with dynamically controlled heights, the system minimizes processing time while maintaining high detection accuracy. It incorporates a control system to correct inaccurate ROIs and mechanisms to detect the presence or absence of the horizon line. Evaluated on the Buoy Dataset (BD) and Singapore Maritime Dataset (SMD), the system demonstrates superior performance over state-of-the-art algorithms, indicating its suitability for real-time applications in dynamic maritime environments. The proposed system achieves median positional and angular errors of 0.83 pixels and 0.08 degrees, respectively, on the SMD, and 0.91 pixels and 0.12 degrees, respectively, on the BD. Additionally, the mean ROI size corresponds to 3% of the video frame size, enabling the system to achieve 20 frames per second for SMD and 33.3 frames per second for BD. The source code and associated datasets utilized in this study are publicly available and can be accessed through <https://github.com/Ahmet-Agaoglu/HL-Detection-using-TSA>, ensuring reproducibility and facilitating further research. Additionally, the result videos, which showcase the outcomes of this research, are available at <https://doi.org/10.5281/zenodo.13768637>.

Keywords Time series analysis · Object tracking · Horizon line detection · Region of interest

1 Introduction

In maritime images, horizon line (HL) detection is the process of identifying the line that separates the sky from the sea in images or video frames. This task is essential in various applications such as marine navigation, aerial imaging, and autonomous vehicle systems, where it aids in camera calibration, digital video stabilization, target detection and tracking, and distance estimation of detected targets [1–5]. Accurate HL detection improves situational awareness and operational efficiency, particularly in challenging environmental conditions.

Horizon line detection systems have evolved significantly over the years [6, 7]. The performance of these systems is

defined by two key components: speed and accuracy, both of which directly impact the success of the application. Accurate detection ensures reliable target tracking, while high-speed processing is essential for real-time applications such as autonomous navigation.

One effective strategy to enhance both speed and accuracy is the use of the region of interest (ROI). By focusing the processing power on a smaller portion of the image, the computational time can be significantly reduced. By narrowing the area of the image to be analyzed, the algorithm can focus on the most relevant part of the frame, which is where the HL is likely to be found. This reduces the chance of misidentifying other edges as the HL, thereby improving accuracy. However, if the ROI does not include the actual HL, the accuracy may suffer. Therefore, while the use of ROI undoubtedly enhances processing speed, its impact on accuracy can vary.

ROI types can be broadly classified into two primary geometric categories. The first type is a rectangular ROI that spans the entire horizontal length of the image and is formed by vertically cropping the image [8, 9]. The second type is

✉ Ahmet Agaoglu
ahmet.agaoglu@yeditepe.edu.tr

Nezih Topaloglu
nezih.topaloglu@yeditepe.edu.tr

¹ Department of Mechanical Engineering, Yeditepe University, Inonu Mah. Kayisdagi Cad., 34755 Atasehir, Istanbul, Turkey

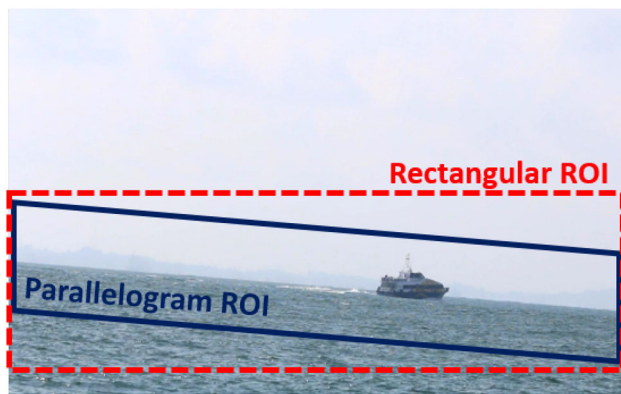


Fig. 1 ROI types

a parallelogram, with its longer edge expected to align with the HL [10–12]. These two types can be seen in Fig. 1.

The parallelogram-shaped ROI has an additional degree of freedom—the orientation that offers a distinct advantage. As the orientation of the HL becomes steeper, the height of a rectangular ROI must increase to ensure the HL is fully contained within it. In this configuration, the HL can be visualized as the diagonal of the rectangular ROI. Thus, for steep HL orientations, a parallelogram can effectively encapsulate the HL within a smaller area, thereby reducing the computational load.

On the other hand, implementing a parallelogram-shaped ROI entails certain challenges due to the need for precise control of an additional variable: the orientation of the ROI. If the orientation is not accurately adjusted simultaneously with the position of the ROI, a significant portion of the HL may be excluded from the ROI, thereby adversely impacting accuracy.

Another significant design aspect in ROI determination is the height of the ROI. Many current methods use a fixed ROI height, which is not ideal. When the HL is stable, a smaller ROI can be employed. This reduces the computational load and improves the accuracy within the confined area. Conversely, in dynamic environments, the ROI height should be increased to accommodate the rapid changes in the HL.

In addition to a well-defined ROI, an HL detection system's performance is highly affected by the efficiency of its second component, the horizon line detection algorithm (HLDA). As the final decision-maker on the input ROI, the accuracy of the HLDA is critical. While a well-defined ROI can reduce computational load and improve accuracy, a sophisticated HLDA can further enhance accuracy. However, the computational complexity of a sophisticated HLDA may negate the speed benefits gained from using an ROI. The combined impact of ROI generation method and HLDA on the overall system performance has not been extensively investigated.

Our approach centers on a novel parallelogram-shaped ROI generation method that dynamically adjusts the ROI size to minimize computational load while accurately predicting the position and orientation of the HL, ensuring it remains within the ROI. This continuous adjustment enables the system to adapt to various conditions and maintain high performance.

Furthermore, a key innovation in our methodology is the employment of time series analysis for determining the ROI. This technique, which analyzes time-ordered data points to extract meaningful statistics and predict future trends, is a new concept in the context of ROI determination. Widely used in fields such as econometrics and finance for tasks like predicting stock prices based on historical data [13, 14], time series analysis offers a robust framework for forecasting the position and orientation of the HL based on previous video frames.

Additionally, our system also incorporates two novel features: a control system that corrects the ROI when an inaccurate forecast is encountered and a set of mechanisms that detect the absence or presence of the HL. Both components are essential since accurate forecasts rely on precise previous estimations.

The outline of the sequel is as follows. In Sect. 2, previous work on various HLDAs as well as ROI determination methods is introduced. Subsequently, in Sect. 3, the overview of the proposed system is detailed. In Sect. 4, the overall performance of the system is first examined and then compared to other methods using different datasets. Finally, conclusions are drawn in Sect. 5.

2 Related work

Horizon line detection in maritime videos has been extensively studied, with numerous methods proposed over the years. Each of these approaches leverages different techniques to accurately identify the HL amidst challenging maritime environments. These approaches can broadly be categorized into edge-based, region-based, and learning-based methods.

Edge-based methods rely on edge detection techniques to identify the HL by pinpointing the transition between the sky and sea. Traditional methods such as the Canny edge detector combined with the Hough Transform are typical examples of this approach [1, 15–21]. While these methods are computationally efficient, they often struggle in cluttered environments where edges from waves, ships, and other objects can obscure the true horizon.

Region-based methods, on the other hand, exploit the color and texture differences between the sky and the sea to segment the image and identify the HL. Techniques such as color clustering, texture analysis, and region growing

are commonly employed in this category [8, 11, 22–24]. These methods tend to perform better in varying lighting and weather conditions compared to edge-based methods but can be computationally intensive due to the segmentation process.

Learning-based methods have gained prominence with the rise of machine learning, particularly deep learning. These approaches involve training classifiers or neural networks to distinguish between pixels or segments of the image that belong to the sky or sea, thus identifying the HL [10, 25–30]. Convolutional neural networks (CNNs) have been especially effective, offering high accuracy by learning complex features from large datasets. However, these methods require significant computational resources and extensive annotated training data. Although not directly applied to horizon detection, neural network architectures used in other image processing tasks can be utilized to detect the horizon line. Zhao et al. proposed a novel network architecture that integrates deformable convolution networks (DCN) with sparse-to-dense matching strategies [31]. This method significantly enhances the accuracy of feature correspondence across multi-view images, suitable for robust horizon line detection where dynamic environmental changes are prevalent. Lin et al. utilized an attention mechanism to focus progressively on important features at multiple scales [32]. This method can effectively improve the detection of horizon line by focusing computational resources on critical areas within the field of view.

As stated in the introduction, focusing on an ROI further enhances the efficiency of horizon line detection and tracking. By identifying and isolating the region where the horizon line is likely to be found, the processing can be confined to a smaller portion of the video frame. This reduction in the area to be processed substantially increases the system's speed, making real-time operation more attainable. The remainder of this section provides an overview of various ROI generation methods.

Jeong et al. [8] and Umair et al. [9] employ an approach that utilizes overlapping horizontal strips to define the ROI, which is typically rectangular in shape. These methods focus on dividing the image into multiple horizontal strips and then analyzing the color or intensity variations within each strip to identify the region most likely to contain the horizon line. Jeong et al. calculate the color distribution within these strips and selects the strip with the highest color difference as the ROI, effectively narrowing down the search area for the horizon line. Similarly, Umair et al. use a comparable technique where overlapping strips are analyzed to pinpoint the horizon line, but they introduce additional refinement steps to enhance the accuracy of horizon detection in complex maritime environments.

Sandru et al. [10] introduce a method that employs a parallelogram-shaped ROI to improve the detection of the

horizon line in maritime images. This ROI is generated by defining two new lines, parallel to the previously detected horizon line but offset by a fixed value $\Delta\rho$. These lines, combined with the vertical boundaries of the image, form a parallelogram that encompasses the expected location of the horizon in the current frame. However, setting $\Delta\rho$ to a value that is too large may result in poor tracking performance, while too small a value might prevent the algorithm from accurately tracking the horizon in highly dynamic situations.

Mou et al. [11] introduce a parallelogram-shaped ROI generation technique for horizon line detection, specifically designed to handle the challenges posed by dynamic maritime environments. The method employs the RANSAC (random sample consensus) algorithm to predict the horizon line, which is then used to define the parallelogram ROI. By aligning the parallelogram with the predicted horizon, the method effectively isolates the region most likely to contain the horizon, thereby enhancing detection accuracy. This approach is particularly adept at dealing with scenarios where the horizon's orientation changes rapidly, ensuring that the ROI consistently captures the horizon throughout the video sequence.

Praczyk [12] employs a distinctive approach for ROI generation by utilizing a border-defined parallelogram, which is adjusted iteratively. In the first iteration, the ROI includes a wide range of possible horizon line locations. As the iterations progress, the ROI size decreases, focusing more narrowly on likely horizon line positions. The approach uses border pixels that represent the beginning and end of each potential line, refining the ROI in subsequent iterations based on the detected horizon line in the previous frame. This iterative reduction of the ROI allows for the efficient and precise detection of the horizon line, particularly in dynamic maritime environments.

Previous approaches either use a fixed-height ROI based on the last detected horizon line [8–10] or a dynamic ROI whose height adjusts according to the horizon line detection method [11, 12]. However, when the horizon line shifts rapidly and the video frame rate is low, relying solely on the previous horizon line can result in missing the correct horizon. Estimating the ROI based on the horizon line detection method can also be risky, particularly when the accuracy of the detection algorithm is low.

This study proposes a novel system that uses time series analysis for adaptive ROI generation. The main contributions of this study are as follows: The system introduces a parallelogram-shaped ROI whose size and orientation are dynamically adjusted based on time series analysis predictions, significantly improving computational efficiency and accuracy compared to fixed ROIs. The application of time series models (TSM) to predict the position and orientation of the horizon line enables real-time adaptability to changing environmental conditions in maritime videos. A control sys-

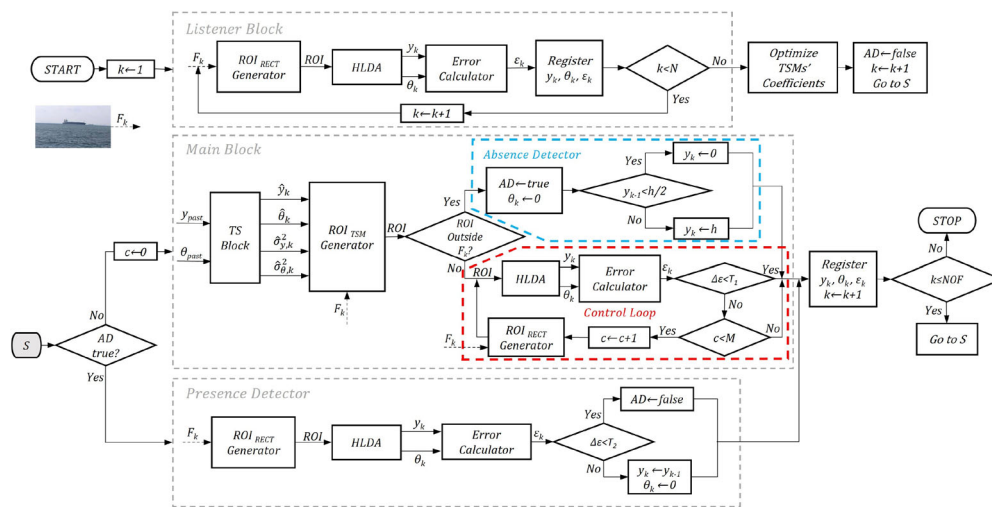


Fig. 2 System diagram illustrating the components

tem is developed to detect and correct forecast inaccuracies, preventing TSM divergence caused by error propagation. Additionally, novel mechanisms for detecting the presence or absence of the horizon line are integrated, allowing the system to adapt to scenarios where the horizon line temporarily disappears.

3 Overview of the proposed system

This section provides an overview of the system and introduces its key components, as illustrated in Fig. 2. The system is divided into three main parts: the Listener Block, the Main Block, and the Presence Detector. Before describing these blocks, the HLDA which is used in all these blocks is described.

3.1 The horizon line detection algorithm

The HLDA processes an input video frame F_k (k representing the frame index) to estimate the state of the HL. The state of the HL, denoted as (y_k, θ_k) , includes the vertical distance y_k (in pixels) from the midpoint of the frame's top edge to the HL and the orientation angle θ_k (in degrees) measured clockwise from the frame's top edge, as shown in Fig. 3.

To estimate the state, the HLDA employed in this study primarily relies on edge detection utilizing the Radon transform. Figure 4 illustrates the sequence of the operations performed by HLDA. The process begins by converting the input image (a) to grayscale and applying the Canny edge detection method to highlight edges (b). Weak edges are subsequently removed using a threshold value set at double the mean of the edge strengths (c). Connected components with fewer than 0.5% of the total number of active pixels in the

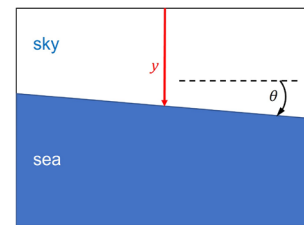


Fig. 3 Illustration of the HL state variables

binary image are then eliminated(d). Finally, the Radon transform is applied to detect the strongest line, resulting in the estimated state of the HL (e).

3.2 Listener Block

The Listener Block serves as the initial stage of the system and operates exclusively during the first N frames of the input video. For each frame, the ROI is initially determined as a rectangle centered on the HL detected in the previous frame, with its height extended equally in both directions by 7.5% of the total frame height. The ROI is then processed by the HLDA. The state (y_k, θ_k) estimated by the HLDA is subsequently passed to the error calculator function, the output of which is utilized by both the control system and the Presence Detector, which are novel features of the system. Detailed explanations of these components will be provided in their respective sections. The N states collected during these frames are then used to optimize the coefficients of the TSMs. After optimization, the system routes to the Main Block.

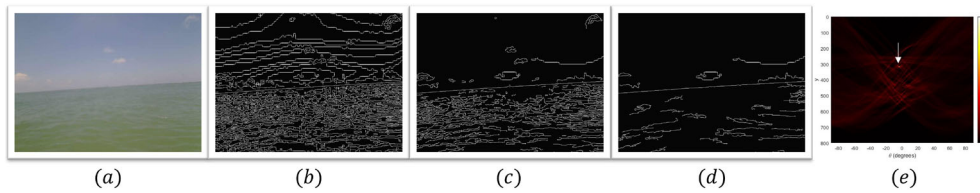


Fig. 4 Sequence of operations in the HLDA

3.3 Main Block

The Main Block, the central component of the system, begins its operation with the Time Series (TS) Block, the outputs of which are then passed to the ROI generator. Following this, the system may proceed along one of two paths: the absence detector or the control loop. The following sections provide a detailed explanation of these structures.

3.3.1 The Time Series Block

The TS Block is central to our study, as it enables the fast and accurate prediction of the HL state without relying on the HLDA. It uses past state estimations by HLDA and the coefficients optimized in the Listener Block to forecast the current frame's state variables ($\hat{y}_k, \hat{\theta}_k$) and their variances $\hat{\sigma}_{y,k}^2$ and $\hat{\sigma}_{\theta,k}^2$. The block employs two composite TSMs separately for y and θ . The models employed for y are ARIMA(2,2,0) and GARCH(1,1), where ARIMA (Autoregressive Integrated Moving Average) models the conditional mean, and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) models the conditional variance. These models are mathematically expressed as follows:

$$(1 - \phi_1 B - \phi_2 B^2)(1 - B)^2 y_k = \epsilon_{y,k} \quad (1)$$

$$\sigma_{y,k}^2 = \alpha_0 + \alpha_1 \epsilon_{y,k-1}^2 + \beta_1 \sigma_{y,k-1}^2 \quad (2)$$

In Eq. 1, B is the backshift operator ($B y_k = y_{k-1}$), ϵ_y is the error term, and ϕ_1 and ϕ_2 are the coefficients of the mean model. In Eq. 2, α_0 is the constant term, α_1 is the coefficient for the squared error term $\epsilon_{y,k-1}^2$, indicating the impact of past errors on variance, and β_1 is the coefficient for the previous variance term $\sigma_{y,k-1}^2$, reflecting variance persistence in the GARCH model.

The models for θ are ARIMA(2,2,0) and GARCH(1,1), mathematically expressed as:

$$(1 - \gamma_1 B - \gamma_2 B^2)\theta_k = \epsilon_{\theta,k} \quad (3)$$

$$\sigma_{\theta,k}^2 = \lambda_0 + \lambda_1 \epsilon_{\theta,k-1}^2 + \omega_1 \sigma_{\theta,k-1}^2 \quad (4)$$

Here, ϵ_{θ} is the error term, γ_1 and γ_2 are the mean model coefficients, λ_0 is the constant term, λ_1 is the coefficient for the squared error term, and ω_1 is the coefficient for the previous variance term in the GARCH model.

The selection process for these models and the number of lag terms involves statistical analysis of the ground truth data of a reference video from Buoy Dataset [19]. This process is detailed in “1”.

3.3.2 ROI generator

The ROI generator block constructs the ROI based on the outputs of the TS Block. \hat{y}_k and $\hat{\theta}_k$ defined the forecasted line around which the ROI will be centered. Their variances, $\hat{\sigma}_{y,k}^2$ and $\hat{\sigma}_{\theta,k}^2$, are used to create four lines that determine the height of the ROI as shown in Fig. 5. These four lines are derived as follows:

$$\begin{aligned} l_1 &: (\hat{y}_k - z\hat{\sigma}_{y,k}, \hat{\theta}_k - z\hat{\sigma}_{\theta,k}) \\ l_2 &: (\hat{y}_k + z\hat{\sigma}_{y,k}, \hat{\theta}_k - z\hat{\sigma}_{\theta,k}) \\ l_3 &: (\hat{y}_k - z\hat{\sigma}_{y,k}, \hat{\theta}_k + z\hat{\sigma}_{\theta,k}) \\ l_4 &: (\hat{y}_k + z\hat{\sigma}_{y,k}, \hat{\theta}_k + z\hat{\sigma}_{\theta,k}) \end{aligned} \quad (5)$$

Here, z denotes the standard (z) score set by the user. The ROI is then defined as a parallelogram aligned with the forecasted line, enclosing the candidate lines. In addition, if the ROI extends beyond the frame, the ROI is cropped to include only the portion that remains inside the frame.

3.3.3 The control system

An accurate forecast by the TSM relies heavily on the precision of previous estimates by HLDA. Any inaccuracy in these initial estimates can compromise the accuracy of subsequent forecasts, which in turn determines the ROI and, consequently, the following estimates. This cycle of errors can lead to a propagation of inaccuracies. Hence, there is a need for an intelligent system capable of detecting and correcting inaccurate estimates to prevent the TSMs from diverging.

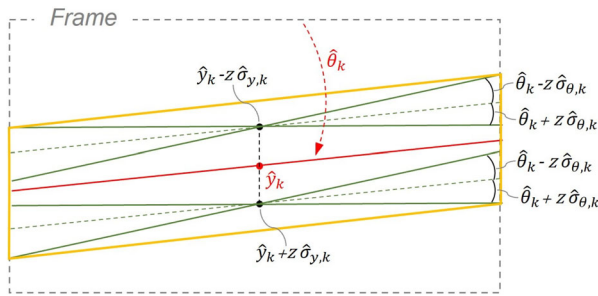


Fig. 5 ROI determination: The red line represents the forecasted HL, green lines are the candidate lines, and the ROI is shown in yellow

To address this, a control system is implemented. This control system possesses two main components: error detection and correction. In the former, the system continuously monitors the estimate and evaluates its accuracy by an error metric ε_k . If any significant deviation in the error metric is observed, then a correction mechanism that adjusts the input data is triggered.

In order to assess the accuracy of the estimates, one relies on the basic assumption that a correctly estimated HL divides the image into two regions. Let A_k^{sky} and A_k^{sea} represent the sky and sea regions that are above and below the HL at frame k , respectively. These regions exhibit similar visual attributes within themselves but different attributes between them.

When the HL is inaccurately estimated, some pixels from A_k^{sky} may switch places with some pixels from A_k^{sea} , causing A_k^{sky} and A_k^{sea} to become more similar to each other. This results in a resemblance in the histograms of the regions. Conversely, a dissimilarity between histograms suggests that the HL partitions the frame accurately. To quantify this, we define the error metric as the square root of the mean of the squared differences between the histograms of the two regions:

$$\varepsilon_k = -\sqrt{\frac{1}{256} \sum_{i=0}^{255} (\text{hist}(A_k^{sky})_i - \text{hist}(A_k^{sea})_i)^2} \quad (6)$$

where the histograms are normalized and i denotes the bin index. The negative sign ensures that the error metric increases when the sky and sea regions are similar, and vice versa. Note that although color (3D) histogram can show better performance, grayscale histogram is used for the sake of computational time.

Furthermore, as the HL shifts from one frame to another, the dimensions and characteristics of the adjacent sky and sea regions transform. For example, if the HL is higher in one frame and lower in the next, the sky region expands while the sea region contracts. This causes significant changes in pixel quantities and their distributions. The enlarged sky

region may include various features like clouds, whereas the reduced sea region may contain fewer elements such as boats or waves. These modifications influence the visual homogeneity within each region. Even with accurate HL estimates across different frames, pixel distributions can unpredictably become more or less similar, causing unjustified deviations in the error metric.

Thus, it is essential to accurately select pixels from both regions and consistently maintain this selection to preserve the reliability of the error metric. Our strategy targets the area surrounding the HL, where visual consistency is more pronounced. The sky region A_{sky} is identified as the area immediately above the HL, and the sea region A_{sea} as the area directly below it. Both regions can be defined by equal-sized parallelograms ($A_{sky} = A_{sea} = A$) with one edge collinear to the HL. The height of the parallelograms is a user-defined parameter, specified as a fraction of the total frame height.

After computing the error, the relative change in the error is calculated based on the previous five errors, as shown below:

$$\Delta\varepsilon_k = \left| \frac{\varepsilon_k - \frac{1}{5} \sum_{i=k-4}^k \varepsilon_i}{\frac{1}{5} \sum_{i=k-4}^k \varepsilon_i} \right| \quad (7)$$

If $\Delta\varepsilon_k$ is greater than T_1 percent, a new ROI is defined in a different manner: the frame is cropped vertically, creating a rectangular region, the diagonal of which aligns with (y_{k-1}, θ_{k-1}) of the previous frame. Along with this rectangular ROI, it is expected to better estimate the state of the HL, thereby having a smaller error. The control loop iterates by gradually enlarging the ROI height in both directions around the same diagonal by an amount defined by the user, until one of three criteria is met: the error deviation falls below T_1 , the maximum number of iterations M is reached, or the ROI size exceeds the frame dimensions.

3.3.4 Horizon line absence detector

In some video frames only the sky or sea is present, which means that a HL is not present. The ROI determined by the TSMs effectively tracks the HL until it is gradually lost frame by frame. Initially, a portion of the ROI extends outside the frame, and eventually, the entire ROI resides outside the frame. This condition is indicated by a Boolean variable, AD , the acronym of absence detection.

When AD becomes True, the system routes to the Absence Detector Block. Since the HL is absent, its state cannot be determined. In such cases, a flat HL aligned with either the top or bottom edge of the frame is registered, depending on the proximity of the previous position y_{k-1} to these edges.

Once AD becomes True, the system routes to the Presence Detector in the next frame.

3.4 Horizon line presence detector

The Presence Detector Block anticipates the reappearance of the disappeared HL. During this phase, a rectangular ROI is generated, with one edge adjacent to either the top or bottom edge, based on the previous position. The height of this rectangular ROI is a user-defined parameter. The HLDA processes the ROI, and then the error metric is calculated. If the percentage change of error relative to the last error recorded when the HL was present is less than a threshold T_2 , AD is set to False, indicating that the HL is successfully detected again, and the system returns to Main Block. Otherwise, the process continues by updating the current state to the last state.

4 Results and discussion

This section examines the performance of the proposed system in terms of accuracy and computational time through detailed analyses and experiments conducted on two datasets: The Singapore Maritime Dataset (SMD) (acquired by [33] and made available through [34]) and the Buoy Dataset (BD) [19]. The SMD is categorized into onboard and onshore segments. The onshore segment was excluded from this study because in these videos, the camera is stationary, and the HL remains constant. Once the correct ROI is determined, consistent results can be achieved using the same ROI throughout the video. Since our method is designed for applications where the HL dynamically changes, which is highly challenging, we focused solely on the onboard segment.

For ease of reference, video names have been shortened as follows: the video named *MVI_0788_VIS_OB* in SMD is referred to as *S0788*, and the video coded *buoyGT_2_5_3_0* in BD is represented as *B2530*.

The BD comprises 10 videos totaling 996 frames, each with a resolution of 800x600 pixels, captured from buoys floating in the sea. This dataset features highly dynamic environments with significant variations in sea state, lighting conditions. It is particularly challenging due to the aggressive oscillations of the HL by waves.

The SMD includes 11 videos totaling 2772 frames, recorded at a resolution of 1920x1080 pixels from cameras mounted on ships. These frames exhibit a wide range of environmental conditions, including varying weather, sea states, and lighting. The SMD is less dynamic than the BD but still presents considerable challenges due to the presence of ships and other maritime structures. Both datasets provide a comprehensive basis for evaluating the robustness and efficiency

of horizon detection algorithms under diverse maritime conditions.

In the experiments, accuracy is measured in terms of positional and angular errors. Positional error $|y - y_{GT}|$ represents the vertical distance in pixels between the estimated HL and the ground truth (GT) HL. Angular error $|\theta - \theta_{GT}|$ is the difference in degrees between the estimated and actual orientation angles of the HL. Mean processing time (MPT) in seconds is evaluated by measuring the mean time required to process each frame. The computational environment used in the experiments includes an Intel(R) Core(TM) i7-13700H 2.40 GHz processor, 32GB DDR4 3200MHz RAM, and MATLAB R2024a [35].

The parameter selection process was informed by theoretical considerations and validated experimentally using the SMD and BD, ensuring robust system performance across varied scenarios. Below, the rationale for each parameter is detailed.

- The number of listening frames N was set to 20 for BD and 60 for SMD, approximately 1/5 of the mean video lengths. This ratio provides enough data for accurate TSM optimization while avoiding delays.
- The standard score for ROI generation was set to $z = 1.96$, corresponding to a 95% confidence interval for a Gaussian distribution. Lower z values reduce ROI size but increase the likelihood of triggering the control loop, degrading efficiency.
- The parallelogram height for error metric calculation was set to 2.5% of the frame height, minimizing the influence of unrelated features (see Sect. 3.3.3).
- The maximum iterations in the control loop $M = 3$ and rectangular ROI enlargement by 15% per iteration allow sufficient error correction without excessive overhead. The final ROI size (45% of the frame height) is sufficient for dynamic conditions, with larger sizes providing no added benefit.
- The rectangular ROI height in presence detection mode was set to 15% of the frame height, consistent with the control loop enlargement step, simplifying parameter tuning.
- The threshold $T_1 = 15\%$ ensures a balance between accuracy and efficiency across both datasets. A higher threshold may delay necessary corrections, allowing inaccuracies to propagate, while a lower threshold could result in excessive adjustments, increasing computational overhead.
- The threshold $T_2 = 3\%$ reliably detects the horizon line's reappearance without prematurely exiting absence detection mode. This value successfully handled the two disappearance events in the datasets.



Fig. 6 Sample results from all videos

To provide a general overview of the algorithm performance, one sample frame from each video, along with the result of the proposed system, is shown in Fig. 6. In these images, the ROI is depicted by a yellow parallelogram, and the estimated HL is represented by a red line.

The subsequent sections provide a detailed analysis of the proposed system's performance. The first subsection focuses on assessing the system's accuracy and speed using various metrics and visualizations. The second subsection benchmarks the proposed system against leading algorithms.

4.1 Performance evaluation

The first analysis involves demonstrating the precision of TSM forecasts on an example video, S0788. In Fig. 7, the top image showcases the y predictions, while the bottom image displays that of θ . Remarkably, in both instances, GT values consistently lie within the prediction intervals, underscoring the high accuracy and reliability of our forecasting model. The mean interval width for y is 11.07 pixels, which corresponds to approximately 1% of the frame height. Such a narrow interval width achieving this level of accuracy is remarkable. For θ , the mean interval width is 0.58 degrees.

Figure 8 provides a broader view of the TSM's performance across all videos in the datasets. It represents the percentage of GT values that fall within the prediction interval (success rate) for each video, with separate evaluations

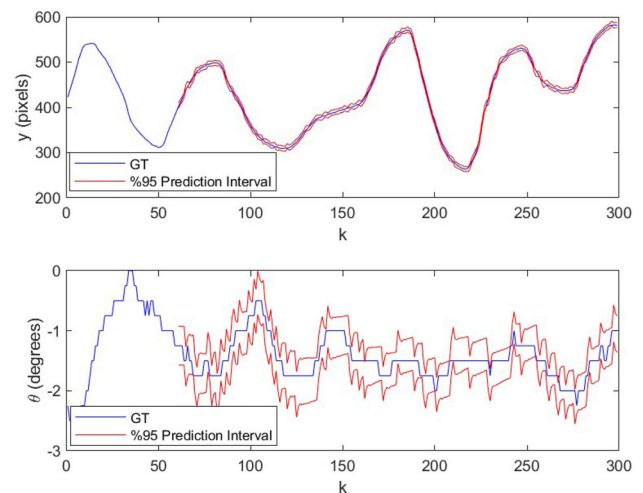
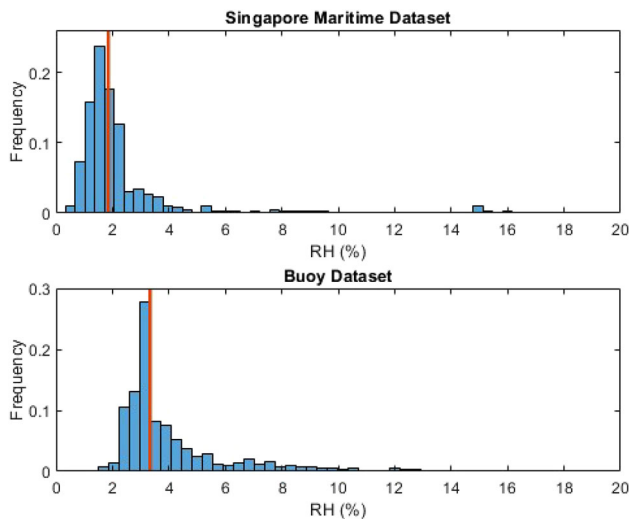
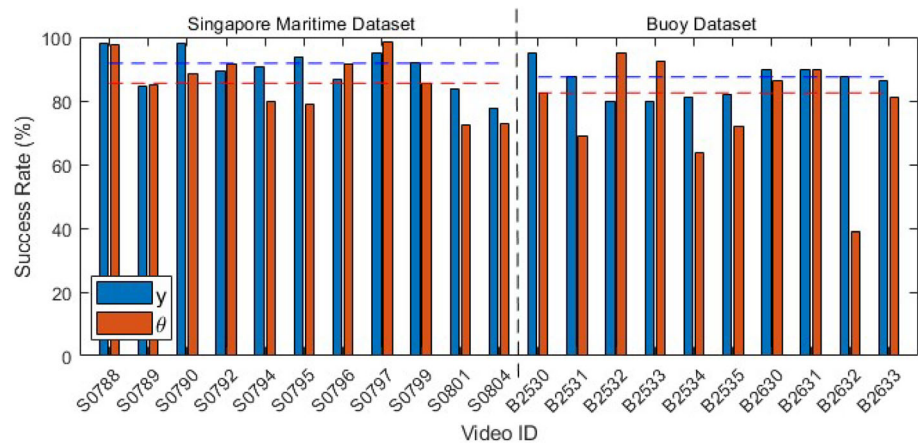
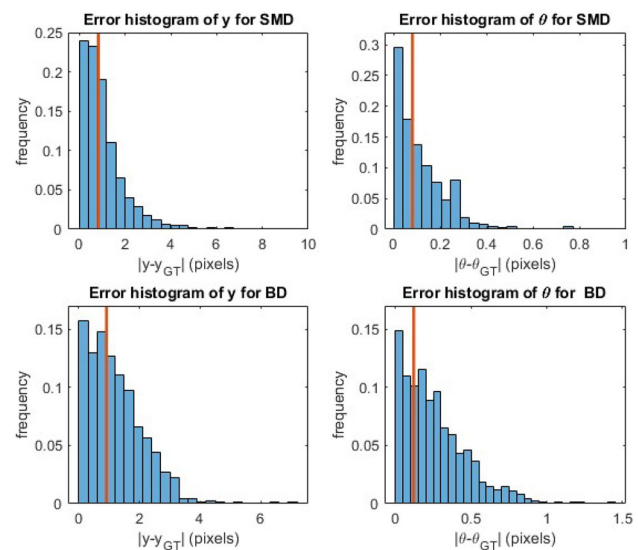


Fig. 7 Prediction intervals of S0788 video

for y (blue bars) and θ (red bars). The weighted mean success rates for each dataset, calculated based on the frame count, are indicated by dashed lines. For the SMD, the mean values are 91.8% for y and 85.5% for θ , while for the BD, the mean values are 87.5% for y and 82.5% for θ . These results highlight the robust predictive capabilities of the TSM across diverse video samples.

Figure 9 shows the normalized histogram of the ROI height as a percentage of the frame height (RH) for all frames

Fig. 8 Success rates in y and θ **Fig. 9** Distribution of RH for all frames in both datasets. Median values are indicated by red lines**Fig. 10** Error histograms. Median values are indicated by red lines

across the SMD (top) and the BD (bottom). This metric is significant as it directly correlates with the number of pixels processed per frame, indicating the load on the HLDA. Due to the presence of outliers, the histograms are truncated to provide better visualization.

For the SMD, the median RH value is 1.9%, while for the BD, it is 3.3%. These results are remarkably efficient since the system processes approximately only 3% of the frame, filtering out 97% of irrelevant data, which is an outstanding achievement in terms of computational efficiency.

The speed of the algorithm is also evaluated by measuring the MPT for each dataset. For the SMD, the proposed system exhibits an MPT of 0.05 s per frame, corresponding to a frame rate of 20 frames per second (FPS). Considering the large frame size of the SMD and the use of a standard CPU, 20 FPS is a satisfactory result. The MPT for the BD is 0.03 s per frame, corresponding to 33.3 FPS, which is sufficient for many real-time applications.

Next, the accuracy of the system is evaluated. Figure 10 presents the error distributions for all frames in the SMD (top) and BD (bottom). The median positional errors are 0.83 pixels for SMD and 0.91 pixels for BD. The median angular errors are 0.08 degrees for SMD and 0.12 degrees for BD. Although the BD results are higher due to its dynamic nature, the accuracy remains impressive, maintaining a positional error of less than one pixel and an angular error of approximately one-tenth of a degree even in highly dynamic conditions.

These analyses underscore the speed and accuracy of the proposed system across different videos, proving its adaptability and providing a robust foundation for subsequent performance comparisons.

Some videos in both datasets exhibit notably wider distributions in both RH and error, indicating challenging conditions. To gain a deeper understanding of the system's behavior under these conditions, we analyze one such video,

S0799 in detail, as seen in Fig. 11. The plot on the top right shows the RH across frames (k), whereas the images on the bottom left illustrate specific frames corresponding to different regions marked on the plot. After the listening phase (A), the system starts to generate ROI with an average height of 15 pixels which corresponds to 1.4% RH (B). Immediately after, the HL disappears and the system starts generating an ROI at the bottom with a fixed height of 0.15(h), which is 162 pixels (C). The HL then reappears and is immediately detected, after which the TSM starts to decrease the ROI height gradually (D). Around $k = 150$, the ROI height decreases to 15 pixels again. In the following frames, the clarity of images is distorted. The HLDA used in the system is not suited for this as it uses an edge filtering process. Incorrect estimations that increase the variance, and therefore the ROI height determined by the GARCH(1,1) model, follow one another (E). During this period, the RH reaches a maximum value of 38% with an average of 18%, and the control loop is triggered several times. Then, the clarity is regained, and the TSM starts to decrease the height gradually until it eventually stabilizes at 15 pixels again (F).

4.2 Performance comparison

In this section, the performance of the proposed system is compared to three state-of-the-art algorithms from the literature. These specific algorithms were selected due to their demonstrated superiority in tests conducted on the SMD and/or BD within their respective studies, consistently outperforming other competing methods.

The first algorithm is MuSCoWERT (Multi-Scale Consistence of Weighted Edge Radon Transform) [36]. This method detects long linear features over multiple scales using a combination of multi-scale median filtering and the Radon transform on a weighted edge map. It has shown excellent performance, achieving a median positional error of about 2 pixels and a median angular error of less than 0.4 degrees. MuSCoWERT outperformed seven other methods, particularly in dynamic and cluttered environments.

The second algorithm is a fast horizon detection method for maritime images by Jeong et al. [8], which will be referred to as JFast in the sequel. It utilizes an ROI approach combined with multi-scale edge detection to reduce the processing area. The Hough transform and least-squares method are then applied within the ROI to estimate the HL accurately. This method achieves a median positional error of less than 2 pixels and a median angular error of approximately 0.1 degrees in both the SMD and BD. It processes SMD images (1920x1080) at around 15 FPS, offering a robust balance of speed and accuracy in dynamic maritime environments. This method is compared to six other methods and shows the best performance in the tests, which were implemented in Python.

Table 1 Performances of the methods on videos from the SMD

	$ y - y_{GT} $ (pixels)			$ \theta - \theta_{GT} $ (°)		
	25th	50th	95th	25th	50th	95th
roiTSM	0.40	0.83	3.24	0.03	0.08	0.36
JFast	0.51	1.23	3.99	0.05	0.12	0.39
JCNN	0.53	1.21	3.79	0.04	0.1	0.38
MuSCoWERT	0.54	1.49	8.17	0.06	0.25	0.88

The best results for each metric are indicated in bold

Table 2 Performances of different methods on videos from the BD

	$ y - y_{GT} $ (pixels)			$ \theta - \theta_{GT} $ (°)		
	25th	50th	95th	25th	50th	95th
roiTSM	0.47	0.91	2.68	0.06	0.12	0.51
JFast	0.53	1.07	2.98	0.07	0.15	0.45
MuSCoWERT	0.72	1.44	4.61	0.11	0.27	1.00

The best results for each metric are indicated in bold

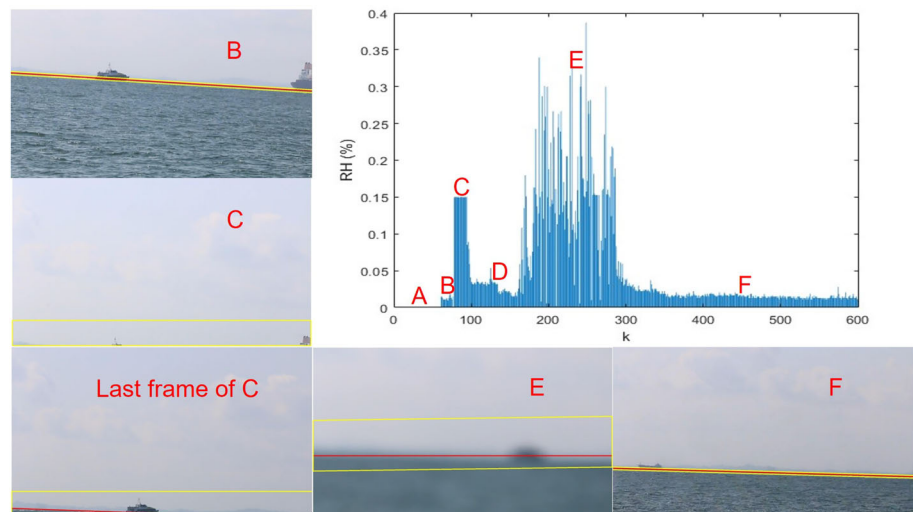
The last algorithm is a machine learning-based approach referred to as the best of its kind according to [6]. Jeong et al. [37] propose three variations: CNN, AdaBoost, and Baseline. Among these, the CNN variant, which will be referred to as JCNN in the sequel, has demonstrated superior performance. This method combines a multi-scale edge detection with a CNN to accurately classify edge pixels belonging to the horizon, effectively handling complex maritime scenes. The CNN variant achieves a median positional error of less than 1.7 pixels and a median angular error of approximately 0.1 degrees, outperforming other methods in accuracy. It was tested against six other methods and demonstrated superior performance.

The accuracies of these algorithms and the proposed algorithm, referred to as *roiTSM*, based on experiments conducted using the SMD and BD are presented in Tables 1 and 2, respectively. These tables show error statistics at the 25th, 50th, and 95th percentiles. The JCNN algorithm results are not presented in Table 2, as the original paper does not include an evaluation of the algorithm on the BD.

In terms of positional error, *roiTSM* demonstrates highest accuracy across all percentiles. At the 25th and 50th percentiles, *roiTSM* achieves the lowest errors of 0.40 pixels and 0.83 pixels, respectively, outperforming JFast, JCNN, and MuSCoWERT. At the 95th percentile, *roiTSM* maintains a relatively low error of 3.24 pixels, which is significantly better than MuSCoWERT's 8.17 pixels and also superior to the remaining competitor algorithms. Furthermore, for angular error, *roiTSM* again shows the best accuracy, achieving the lowest errors at all percentiles.

The accuracy trends observed in the SMD continue in the BD. *roiTSM* again shows strong accuracy, with the lowest positional errors at the all percentiles. For angular errors,

Fig. 11 Analysis of video S0799 from the SMD under challenging conditions. The graph on the top right depicts the RH across frames (k)



roiTSM's 95th percentile error (0.51 degrees) is higher than JFast's (0.45 degrees), suggesting a slight disadvantage in angular accuracy under extreme conditions. Despite this, the overall accuracy of roiTSM remains superior to JFast, particularly in less extreme scenarios.

While the proposed system demonstrates robust performance on both the SMD and BD datasets, several limitations must be acknowledged. The video durations in these datasets are relatively short, and the TSMs employed are optimized for such timeframes. In real-world scenarios with longer video durations, the dynamic characteristics of the horizon line may evolve, potentially causing the TSMs to diverge if the estimates become inaccurate. To address this, the system may require a mechanism to periodically update or re-optimize the model parameters. A simple solution could involve restarting the system to recalibrate the TSMs, ensuring sustained accuracy over extended video durations.

Another limitation arises from the dependency of the dynamic ROI generation algorithm's performance on the HLDA, particularly during the Listener Block. Observations from the HLDA at this stage are critical for optimizing the TSM coefficients, which are subsequently used for ROI generation throughout the video. Inaccurate HLDA estimates during the Listener Block can result in suboptimal model optimization, thereby affecting ROI accuracy over time. While the modularity of the ROI generation algorithm allows for flexibility in integrating different HLDA implementations, ensuring the accuracy and robustness of the HLDA is essential to minimize cascading errors and maintain system reliability.

5 Conclusion

This study presents a novel approach for horizon line detection using dynamic ROI generation based on time series analysis. The proposed system is designed to enhance both accuracy and computational efficiency in highly dynamic environments. The key contributions of this work can be summarized as follows:

1. *Dynamic ROI generation:* By employing a novel ROI generation method, the system continuously adjusts the ROI size based on the forecasted position and orientation of the HL. This dynamic adjustment allows the system to maintain a high level of accuracy while minimizing the computational load on the HLDA.
2. *Time series analysis:* The use of TSM to forecast the HL state based on previous frames is a novel approach. Time series analysis proved effective in predicting the HL's position and orientation, thereby enhancing the system's performance in real-time applications.
3. *Control and detection mechanisms:* The proposed system incorporates a control system to correct the ROI when an inaccurate forecast is encountered and detector mechanisms to sense the absence or presence of the HL. These features are important for maintaining accuracy and ensuring the system adapts to various conditions.

The system's performance was extensively evaluated using two challenging datasets. The results demonstrated that the proposed system consistently outperformed state-of-the-art algorithms in terms of both accuracy and computational efficiency. Specifically, the system achieved lower positional and angular errors while maintaining real-time processing speeds, making it suitable for applications in dynamic maritime environments. Furthermore, the comparison with other

methods highlighted the advantages of the dynamic ROI generation and time series analysis approach. The system's ability to adapt to changing conditions and maintain high accuracy underlines its potential for various practical applications.

In conclusion, the proposed system represents a significant advancement in the field, offering a robust and efficient solution for real-time HL detection in challenging environments. Future work could explore the application of TSMs to other object tracking tasks. For instance, TSM-based prediction of an object's state could be employed to define smaller, dynamic ROIs tailored to the object's movement, thereby significantly enhancing computational efficiency. The system could also be extended to detect horizons in infrared videos, which may pose unique challenges, such as the need to adapt the error estimation algorithm for different imaging modalities. Additionally, the method's applicability could be examined for horizon detection in aerial videos, such as those captured by unmanned aerial vehicles, as well as for non-maritime videos where horizon detection remains relevant.

Appendix A Selection of the time series model

Depending on the camera movement, the two horizon line parameters y and θ can either be either dependent or independent. For instance, when the camera rotates about an axis perpendicular to its projection plane and through the focal point, y and θ change dependently, making a bivariate model reasonable. However, when the camera shifts downward without altering its orientation, y decreases while θ remains constant, which necessitates modeling y and θ as two separate univariate time series. Since decomposing camera movement into translations and rotations with respect to each axis is a difficult task, the horizon line parameters are treated independently. Although univariate models may sacrifice some accuracy, they can still perform tasks typically handled by multivariate models.

The analyses for determining TSMs for y and θ can be conducted using the GT data of a reference video. Video B2631 is chosen due to its highly dynamic nature, ensuring that the TSMs can accurately capture rapid and significant changes in both variables. Although models determined by a single video might appear to compromise overall performance across all videos, it effectively demonstrates compatibility with various potential applications. Note that the models in this study can be updated, based on the subsequent analyses, for specific applications to achieve better results.

First, a model for y and then for θ is determined, drawing on principles from the Box-Jenkins methodology [38]. The y series fails the augmented Dickey–Fuller test for stationarity (Best Lag: 3, Test Statistic: -1.2685 , p -value: 0.1875 ,

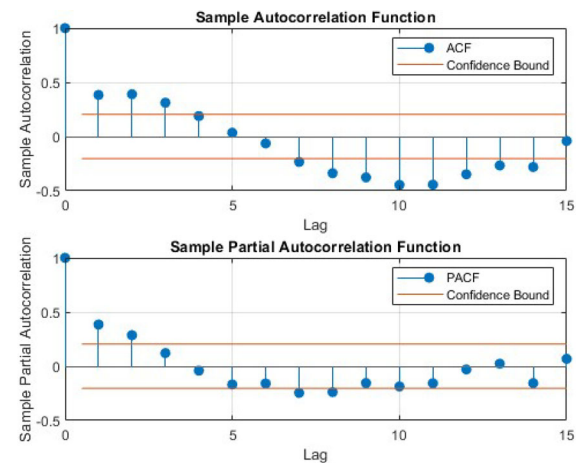


Fig. 12 ACF and PACF plots of differenced series

Critical Value: -1.9444). To achieve stationarity, successive differencing is applied. After two differencing operations, the series attains stationarity (Best Lag: 2, Test Statistic: -3.2085 , p -value: 0.0021 , Critical Value: -1.9445).

With stationarity achieved, the next step involves determining the order of the autoregressive and moving-average terms. This identification is facilitated by examining the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots, shown in Fig. 12. The ACF plot exhibits a damped sinusoidal decay, while the PACF plot cuts off after the second lag. This behavior suggests a second-degree autoregressive model [38], leading to the conditional mean model of y as ARIMA(2,2,0), where 2 is the order of the autoregressive model, 2 is the degree of differencing, and 0 is the order of the moving-average model, respectively.

Once the conditional mean model is determined, the next step is to examine the variance. Figure 13 illustrates the ACF and PACF plots of the squared residuals of the differenced series, both showing autocorrelation and suggesting a GARCH model with lagged variances and lagged squared innovations. A GARCH(1,1) model is particularly appropriate due to its parsimony, effectively balancing model complexity with a few number of parameters, allowing it to accurately capture the dynamics of conditional variance without overfitting.

Further, to validate the presence of conditional heteroscedasticity, Engle's ARCH test is applied to the residuals series, specifying a two-lag ARCH model as the alternative hypothesis. The test yields a test statistic of 12.345 , a p -value of 0.001 , and a critical value of -3.45 with 2 lags. These results reject the null hypothesis of no ARCH effects, indicating the suitability of an ARCH model with two lagged squared innovations. This model is locally equivalent to a GARCH(1,1) model, further supporting the selection of the GARCH(1,1) framework for modeling conditional heteroscedasticity.

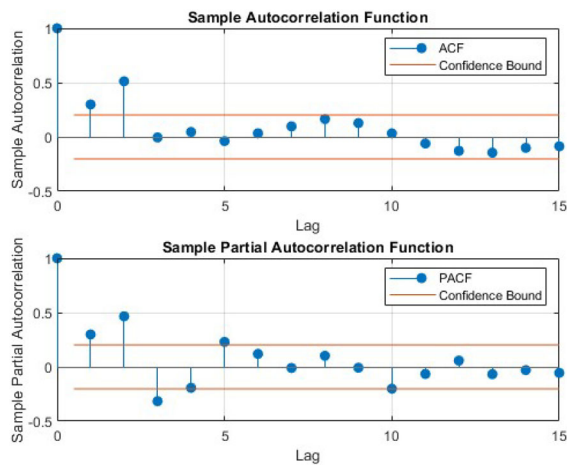


Fig. 13 ACF and PACF plots of squared residuals

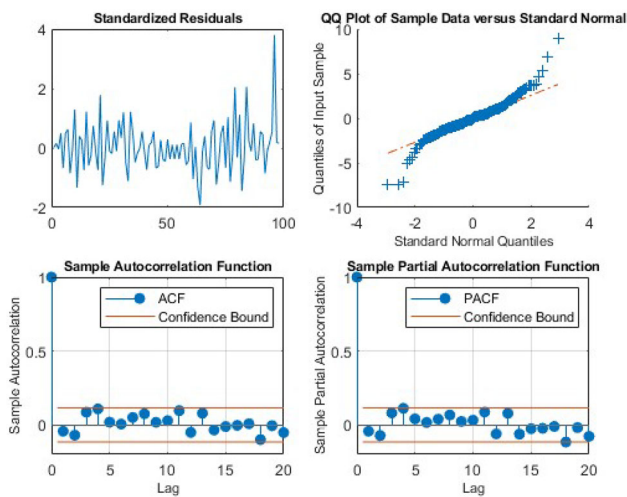


Fig. 14 Goodness-of-fit diagnostics for the ARIMA(2,2,0) GARCH(1,1) model

After determining the composite model ARIMA(2,2,0) GARCH(1,1) as the corresponding conditional mean and variance models, respectively, the residuals are inferred from the composite model to check the goodness of fit, see Fig. 14. The top-left plot shows the standardized residuals, indicating no significant autocorrelation. The top-right plot is a QQ plot of the sample data versus a standard normal distribution, showing that the residuals are reasonably normally distributed. The bottom-left and bottom-right plots display the ACF and PACF of the residuals, respectively, both indicating no significant autocorrelation, further confirming the adequacy of the model fit.

When the same procedure is applied to the (GT) θ series, a comparable model is identified, ARIMA(2,0,0) GARCH(1,1). The difference between the models is attributed to the stationarity of the θ series, which eliminates the need for differencing, thereby setting the integration parameter to 0.

Author Contributions Both authors contributed to the algorithm design, data collection, simulation, and analyses. The first draft of the manuscript was written by “Ahmet Agaoglu,” and both authors commented on previous versions of the manuscript.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Data Availability The Singapore Maritime Dataset and Buoy Dataset utilized in this study were obtained from <https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset> and https://drive.google.com/file/d/0B43_rYxEgelVVngtMVBpWGFqckE/, respectively.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Kong, X., Liu, L., Qian, Y., Cui, M.: Automatic detection of sea-sky horizon line and small targets in maritime infrared imagery. *Infrared Phys. Technol.* **76**, 185–199 (2016). <https://doi.org/10.1016/j.infrared.2016.01.016>
- Jian, L., Wen, G.: Maritime target detection and tracking. In: 2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), pp. 309–314 (2019). <https://doi.org/10.1109/AUTEEE48671.2019.9033200>
- Broek, S.P., Bouma, H., Degache, M.A.C.: Discriminating small extended targets at sea from clutter and other classes of boats in infrared and visual light imagery. In: Drummond, O.E. (ed.) *Signal and Data Processing of Small Targets 2008*, 6969, 69690. SPIE (2008). <https://doi.org/10.1117/12.777542>
- Gladstone, R., Moshe, Y., Barei, A., Shenhav, E.: Distance estimation for marine vehicles using a monocular video camera. In: 2016 24th European Signal Processing Conference (EUSIPCO), pp. 2405–2409 (2016). <https://doi.org/10.1109/EUSIPCO.2016.7760680>
- Voles, P., Smith, A.A.W., Teal, M.K.: Nautical scene segmentation using variable size image windows and feature space reclustering. In: Vernon, D. (ed.) *Computer Vision—ECCV 2000*, pp. 324–335. Springer, Berlin (2000)
- Zardoua, Y., Astito, A., Boulaala, M.: A survey on horizon detection algorithms for maritime video surveillance: advances and future techniques. *Vis. Comput.* **39**(1), 197–217 (2023)
- Zardoua, Y., Astito, A., Boulaala, M., Dokkali, Y.: A short overview of horizon detection methods applied to maritime video stabilization. In: Kacprzyk, J., Balas, V.E., Ezziyyani, M. (eds.) *Advanced*

- Intelligent Systems for Sustainable Development (AI2SD'2020), pp. 857–864. Springer, Cham (2022)
8. Jeong, C.Y., Yang, H.S., Moon, K.: Fast horizon detection in maritime images using region-of-interest. *Int. J. Distrib. Sens. Netw.* **14**(7), 1550147718790753 (2018). <https://doi.org/10.1177/1550147718790753>
 9. Umair, M., Hashmani, M.A., Keiichi, H.: Rough-sea-horizon-line detection using a novel color clustering and least squares regression method. *J. Hunan Univ. Natl. Sci.* **47**(12) (2020)
 10. Sandru, A., Kujala, P., Visala, A.: Horizon detection and tracking in sea-ice conditions using machine vision. *IFAC-PapersOnLine* **56**(2), 6724–6730 (2023). <https://doi.org/10.1016/j.ifacol.2023.10.377>
 11. Mou, X., Shin, B.-S., Wang, H.: Hierarchical RANSAC for accurate horizon detection. In: 2016 24th Mediterranean Conference on Control and Automation (MED), pp. 1158–1163 (2016). <https://doi.org/10.1109/MED.2016.7535933>
 12. Praczyk, T.: A quick algorithm for horizon line detection in marine images. *J. Mar. Sci. Technol.* **23**(1), 164–177 (2018). <https://doi.org/10.1007/s00773-017-0464-8>
 13. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: *Time Series Analysis: Forecasting and Control*. Wiley, Hoboken (2016)
 14. Idrees, S.M., Alam, M.A., Agarwal, P.: A prediction approach for stock market volatility based on time series data. *IEEE Access* **7**, 17287–17298 (2019). <https://doi.org/10.1109/ACCESS.2019.2895252>
 15. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **15**(1), 11–15 (1972). <https://doi.org/10.1145/361237.361242>
 16. Shan, X., Zhao, D., Pan, M., Wang, D., Zhao, L.: Sea-sky line and its nearby ships detection based on the motion attitude of visible light sensors. *Sensors* (2019). <https://doi.org/10.3390/s19184004>
 17. Liang, D., Liang, Y.: Horizon detection from electro-optical sensors under maritime environment. *IEEE Trans. Instrum. Meas.* **69**(1), 45–53 (2020). <https://doi.org/10.1109/TIM.2019.2893008>
 18. Kim, S.: Sea-based infrared scene interpretation by background type classification and coastal region detection for small target detection. *Sensors* **15**(9), 24487–24513 (2015). <https://doi.org/10.3390/s150924487>
 19. Fefilat'yev, S., Goldgof, D., Shreve, M., Lembke, C.: Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Eng.* **54**, 1–12 (2012)
 20. Prasad, D.K., Rajan, D., Rachmawati, L., Rajabally, E., Quek, C.: Muscower: multi-scale consistence of weighted edge radon transform for horizon detection in maritime images. *J. Opt. Soc. Am. A* **33**(12), 2491–2500 (2016). <https://doi.org/10.1364/JOSAA.33.002491>
 21. Zhang, Y., Hu, Q., Li, D., Luo, H., Li, W.: Texture feature-based local adaptive otsu segmentation and Hough transform for sea-sky line detection. *Multimed. Tools Appl.* **83**(12), 34477–34498 (2024). <https://doi.org/10.1007/s11042-023-17012-2>
 22. Gershikov, E., Libe, T., Kosolapov, S.: Horizon line detection in marine images: which method to choose? *Int. J. Adv. Intell. Syst.* **6**, 79–88 (2013)
 23. Sun, Y., Fu, L.: Coarse-fine-stitched: a robust maritime horizon line detection method for unmanned surface vehicle applications. *Sensors* (2018). <https://doi.org/10.3390/s18092825>
 24. Topaloglu, N.: An intensity-difference-based maritime horizon detection algorithm. *Signal Image Video Process.* (2024). <https://doi.org/10.1007/s11760-024-03219-9>
 25. Ahmad, T., Bebis, G., Nicolescu, M., Nefian, A., Fong, T.: An edge-less approach to horizon line detection. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 1095–1102 (2015). <https://doi.org/10.1109/ICMLA.2015.67>
 26. Carrio, A., Bavle, H., Campoy, P.: Attitude estimation using horizon detection in thermal images. *Int. J. Micro Air Veh.* **10**(4), 352–361 (2018). <https://doi.org/10.1177/1756829318804761>
 27. Mo, W., Pei, J.: Sea-sky line detection in the infrared image based on the vertical grayscale distribution feature. *Vis. Comput.* **39**(5), 1915–1927 (2022). <https://doi.org/10.1007/s00371-022-02455-9>
 28. Fu, J., Li, F., Zhao, J.: Real-time infrared horizon detection in maritime and land environments based on hyper-laplace filter and convolutional neural network. *IEEE Trans. Instrum. Meas.* **72**, 1–13 (2023). <https://doi.org/10.1109/TIM.2023.3282656>
 29. Abramov, N., Emelyanova, Y., Fralenko, V., Khachumov, V., Khachumov, M., Shustova, M., Talalaev, A.: Intelligent methods for forest fire detection using unmanned aerial vehicles. *Fire* (2024). <https://doi.org/10.3390/fire7030089>
 30. Li, C., Cai, C., Zhou, W., Wu, K.: A sea-sky-line detection method for long wave infrared image based on improved swin transformer. *Infrared Phys. Technol.* **138**, 105125 (2024). <https://doi.org/10.1016/j.infrared.2024.105125>
 31. Zhao, Y., Zhang, H., Lu, P., Li, P., Wu, E., Sheng, B.: Dsd-matchingnet: deformable sparse-to-dense feature matching for learning accurate correspondences. *Virtual Reality Intell. Hardware* **4**(5), 432–443 (2022). <https://doi.org/10.1016/j.vrih.2022.08.007>
 32. Lin, X., Sun, S., Huang, W., Sheng, B., Li, P., Feng, D.D.: Eapt: efficient attention pyramid transformer for image processing. *IEEE Trans. Multimed.* **25**, 50–61 (2023). <https://doi.org/10.1109/TMM.2021.3120873>
 33. Prasad, D.K., Rajan, D., Rachmawati, L., Rajabally, E., Quek, C.: Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey. *IEEE Trans. Intell. Transp. Syst.* **18**(8), 1993–2016 (2017). <https://doi.org/10.1109/TITS.2016.2634580>
 34. Prasad, D.K.: Singapore maritime dataset. <https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset>. Accessed 06 Aug 2024 (2024)
 35. Inc., T.M.: MATLAB Version: 24.1.0 (R2024a). <https://www.mathworks.com>
 36. Prasad, D.K., Rajan, D., Rachmawati, L., Rajabally, E., Quek, C.: Muscower: multi-scale consistence of weighted edge radon transform for horizon detection in maritime images. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.* **33**(12), 2491–2500 (2016). <https://doi.org/10.1364/JOSAA.33.002491>
 37. Jeong, C., Yang, H.S., Moon, K.: A novel approach for detecting the horizon using a convolutional neural network and multi-scale edge detection. *Multidimens. Syst. Signal Process.* **30**(3), 1187–1204 (2019). <https://doi.org/10.1007/s11045-018-0602-4>
 38. Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: *Time Series Analysis: Forecasting and Control*, 3rd edn. Prentice Hall, Englewood Cliffs (1994)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ahmet Agaoglu was born in Turkey in 1984. He received the BSc degree in Mechanical Engineering in 2007, the MSc degree in Mechanical Engineering in 2010, and the PhD degree in Mechanical Engineering in 2023, all from Yeditepe University, Turkey. His PhD research focused on novel vertex enumeration algorithms with mechanical engineering applications. He is currently a Senior Lecturer in the Department of Mechanical Engineering at Yeditepe University, Turkey. His

research interests include machine learning, convex optimization, computational geometry, signal processing, machinery and mechanism design, and solid modeling. He is also passionate about developing new algorithms and exploring their applications in engineering and computational sciences.



Nezih Topaloglu is a professor in the Faculty of Engineering at Yeditepe University, Turkey. He earned his BS degree in Electrical and Electronics Engineering from Bogazici University, Turkey, and his PhD in Mechanical and Mechatronics Engineering from the University of Waterloo, Canada, where he studied with a full scholarship. From 2009 to 2017, he served as an assistant professor in the Department of Mechanical Engineering at Yeditepe University. He was pro-

moted to associate professor in 2017 and became a full professor in 2024. His research focuses on mechatronic and smart systems, machine learning, and signal and image processing.