2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Real-Time Horizon Locking on Unmanned Surface Vehicles

Benjamin Kiefer[1] and Andreas Zell[1]

***Abstract*— The expanding use of automated vision, assistance systems, and augmented reality applications in marine settings calls for reliable and accurate horizon detection and locking. Traditional methods utilizing Inertial Measurement Units (IMU) or feature-based computer vision techniques often yield inconsistent results, particularly when unmanned surface vehicles or boats are subject to high-speed movement or choppy waters. Addressing this, our work introduces a computer vision (CV)-based solution for real-time horizon locking. Employing real-time semantic segmentation, we accurately differentiate between sky, land or water in the frame, enabling computational locking of the horizon's position. This stable visual reference significantly improves the performance and reliability of onboard systems for autonomous navigation, augmented reality overlays, and multi-object tracking. Supported by a dataset collected under various marine conditions, our method has proven to achieve high accuracy with low computational latency, making it a promising avenue for wide-scale implementation on automated and semi-automated systems.**

I. Introduction

The marine industry is undergoing a significant transformation due to advances in automated vision systems, assistance applications, and augmented reality technologies [1]–[4] (see Figures 4 and 5). These innovations aim to bolster navigational safety, efficiency, and user experience onboard on unmanned surface vehicles (USVs) and boats or ships. However, the consistent capturing and real-time locking of a stable and accurate horizon line, a foundational visual reference for these applications, present challenges, especially in dynamically unstable marine environments [5]–[8].

Traditional methods for horizon detection either rely on Inertial Measurement Units (IMU) or use basic computer vision techniques like feature and edge detection [7], [9]. These methods frequently underperform in marine settings, particularly when boats are in motion, leading to inconsistent and unreliable horizon data as shown in Figures 2 and 3. To address this gap and add empirical weight to our research, we have collected a comprehensive dataset featuring real-time video recordings and corresponding IMU data under various marine conditions.

Against this backdrop, this paper introduces a robust computer vision (CV)-based method for real-time horizon locking. Leveraging semantic segmentation, our approach identifies and differentiates the sky from land or water, enabling us to lock the horizon computationally. This technique dramatically improves the reliability and effectiveness of various onboard applications, particularly those that require a stable visual frame of reference for real-time operation.

[1]Both authors are with the Faculty of Computer Science, University of Tuebingen, Germany. prename.surname@uni-tuebingen.de

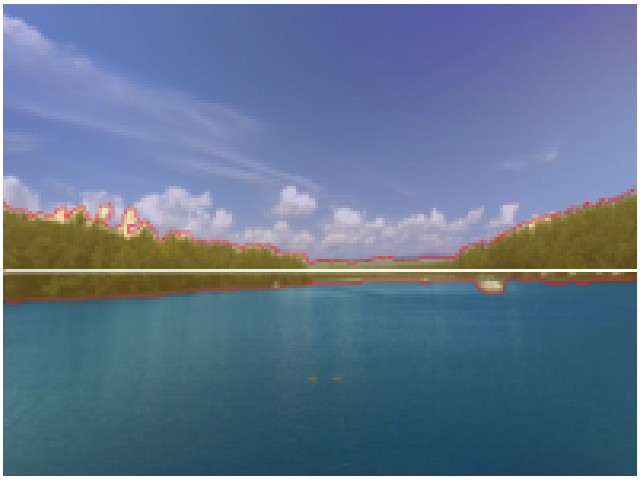

Fig. 1. Our goal is to robustly find the horizon line, which is necessary for several downstream applications. We employ a semantic segmentation-based approach to finding the horizon which is more stable than conventional CV- or IMU-based approaches.

Because there is no public benchmark for this problem, we captured and manually labeled videos for object detection and multi-object tracking labels with accompanying IMU data for each frame.

Our main contributions are:

- We present a novel, CV-based method for real-time horizon detection and locking, clearly outperforming traditional IMU-based and feature-based computer vision methods.
- We have assembled a unique dataset that includes real-time video along with corresponding IMU data, captured under diverse marine conditions. We manually annotate the horizon location for each frame and Multi-Object Tracking (MOT) labels (bounding boxes for boats & buoys with temporally consistent IDs) for downstream MOT evaluation. This dataset serves as a robust testing ground for validating our approach and we make it publicly available[1].
- We demonstrate the efficacy of our approach through experiments focused on augmented reality overlays and multi-object tracking, leveraging our dataset for empirical validation. Our results indicate significant improvements in both accuracy and computational efficiency when compared to traditional methods.

The remainder of the paper is organized as follows: Section 2 offers a comprehensive literature review on existing

[1] https://cloud.cs.uni-tuebingen.de/index.php/s/28zdd4YrsrrAN9T

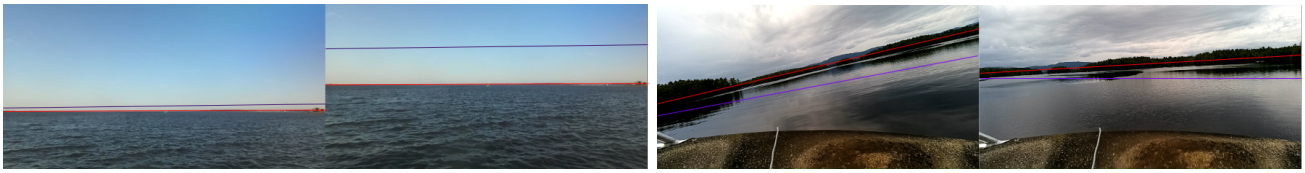

Fig. 2. Left: Limitation of IMU-based approach. A sudden wave hitting the boat cause IMU-based methods (purple) to fail whereas our segmentation-based method is robust (red). Right: Another limitation of IMU-based approach. Fast acceleration and/or curve movements cause IMU-based methods (purple) to fail whereas our segmentation-based method is robust (red).

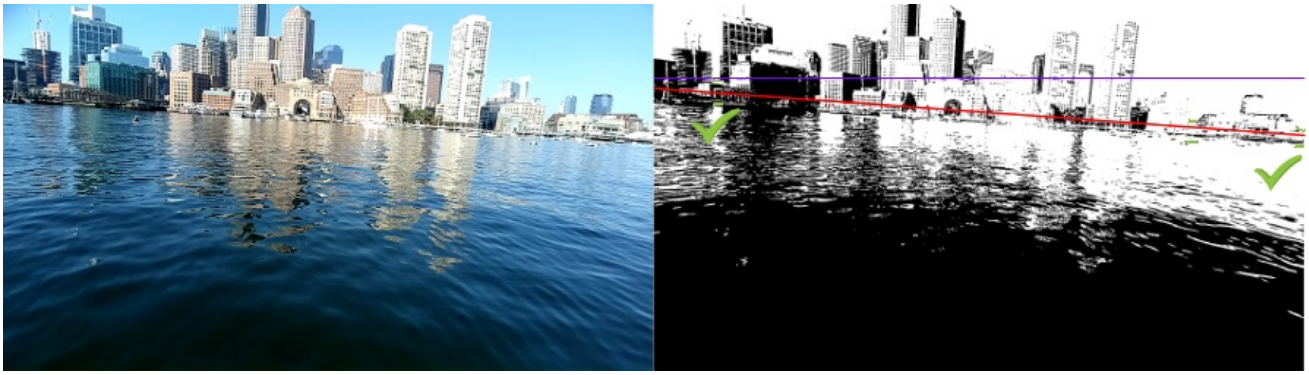

Fig. 3. Limitation of traditional CV-based approach using Otsu segmentation. Our approach yields a regressed line by taking the points where the waterline and horizon line are close (green ticks and green line parts).

methods for horizon detection, including IMU-based and traditional computer vision techniques. Section 3 details the methodology behind our semantic segmentation-based approach. Section 4 discusses the experimental results, especially those derived from our dataset, and validates the efficacy of our method in downstream applications like augmented reality overlays and multi-object tracking. Section 5 concludes the paper and suggests future directions for this research.

II. Related Work

Horizon detection methods have evolved significantly over the years, with traditional approaches relying on either Inertial Measurement Units (IMU) or basic computer vision techniques. This section reviews these prevalent methodologies, particularly in dynamically unstable marine conditions.

A. IMU-based Methods

IMU-based horizon detection relies on using inertial sensors like accelerometers and gyroscopes to estimate the tilt and orientation of a platform. By combining these measurements over time, the angle of the horizon relative to the device can be calculated through sensor fusion algorithms [10]. However, IMU measurements drift over time, leading to inaccurate horizon estimates, especially in dynamic environments like boats at sea. The accumulated drift is proportional to the duration of operation, so these methods perform poorly for continuous real-time horizon locking. Low-cost IMU sensors are also sensitive to vibrations and shocks frequently experienced on marine vessels, further degrading their reliability [11].

B. Computer Vision Techniques

[12] employ an autocalibration method that estimates the rotation matrices of stereo cameras using the sea horizon and a point at infinity, primarily utilizing a Haar-like feature-based approach for horizon detection. Their technique is designed for static environments and focuses on rectifying images so that the horizon appears horizontally. While their method provides a framework for autocalibration in marine settings, it does not offer a solution for real-time horizon locking, especially under the dynamically challenging conditions encountered in high-speed movement or choppy waters. Our approach, by contrast, leverages semantic segmentation to robustly differentiate between the sky and land or water, enabling real-time and reliable horizon locking.

Computer vision methods like edge detection, Hough transforms, and vanishing point estimation have been applied for horizon finding. However, these techniques rely on hand-crafted features and often fail in environments without clearly defined edges or lines. On open seas, factors like weather, lighting changes, and wave patterns create noisy imagery that lacks stable features. Learning-based methods like HoG classifiers and CNNs have shown more promise by learning robust image features. But they are still susceptible to challenging marine environments and struggle with generalizability across different sea conditions.

Several works propose water boundary line detection methods either based on Gray Level Co-occurrence Matrix (GLCM) texture entropy, which effectively segments water, land, and air for water line regression, especially in images with high-brightness areas [13], by combining the strengths of local binary patterns (LBPs) and GLCM for

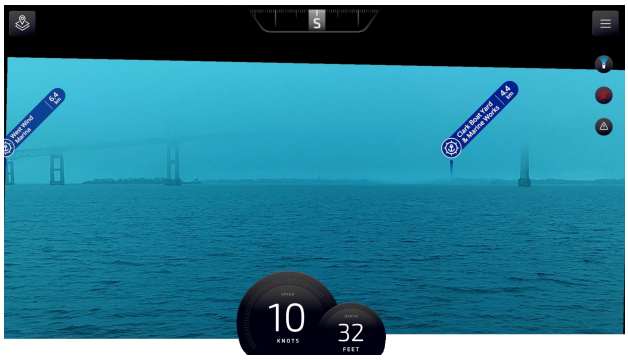

Fig. 4. Application example augmented reality. Points of interested can successfully be overlaid on the image frame pointing to a far away point on the horizon line which would otherwise point into the water or sky.

Fig. 5. Application example augmented reality. A grid can successfully be overlaid on the image frame, as we know the water surface orientation via the horizon line. The black boundary on top comes from locking the horizon in the middle of the screen for simplicity.

texture analysis and structure extraction to preprocess river images, thereby mitigating the effects of USV motion, wind, and illumination [14], or via using binary masks [15]. It's essential to distinguish between the waterline and the horizon line: while the former delineates the boundary between water and land or objects, the latter marks the division between the sky and the earth or water, serving as a pivotal visual reference in many marine applications. Essentially, we obtain the waterline for free, given the semantic segmentation. However, the actual horizon line is not trivially given by the waterline yet.

A recent study fused IMU data with computer vision techniques, employing a Kalman filter for visual horizon tracking in marine environments [16]. Unlike our method, their approach does not harness the full potential of semantic segmentation for horizon detection.

Overall, traditional CV techniques are inconsistent for real-time horizon locking, especially amidst the dynamic motions and appearance changes commonly experienced onboard marine vessels. Our approach addresses these limitations through end-to-end representation learning tailored to maritime imagery.

C. Marine Datasets

In the environmental monitoring domain, there are various datasets for species detection and segmentation [17]–[21].

There are multiple datasets from the viewpoint of USVs and ships for object detection [22]–[25], tracking [3], and segmentation [2].

For IMU-related data in the maritime domain, there are various platforms or datasets [25], [26], [26]–[31], but they don't have a focus on horizon detection (no labels available) and partly are not publicly available.

III. Methodology

Our approach employs semantic segmentation to enable robust, real-time horizon locking in marine settings. This section delves into the architecture of our model, our unique dataset, and the computational techniques employed.

A. Semantic Segmentation Model

Our work is based on the eWaSR network [32], which is an optimized variant of the original WaSR [33], tailored for embedded devices. It replaces WaSR's computationally intensive ResNet-101 backbone with a lightweight ResNet-18, while maintaining semantic richness by concatenating features from various layers. The feature mixer in eWaSR, termed Light-weight Scale-Aware Semantic Extractor (LSSE), efficiently refines these features using two metaformer refinement modules: CRM and SRM. The decoder is streamlined, leveraging the TopFormer semantic-enrichment routines and fusing features using the Semantic Injection Modules (SIM). Overall, eWaSR offers a balance between computational efficiency and detection accuracy, ideal for real-time maritime applications.

We downscale the frames to run the semantic segmentation on an image resolution of 512×384. Table I shows the runtime of the network on different hardware, showing its real-time capability.

TABLE I

Runtime performance of our method with ResNet-18 on OAK-D, NVIDIA RTX 3070Ti, and NVIDIA RTX 4090 Mobile.

Device	Runtime (ms)	FPS
OAK-D Camera [32]	182	5.5
NVIDIA Orin AGX	28	36.7
NVIDIA RTX 3070Ti [32]	8.7	115
NVIDIA RTX 4090 Mobile	20	50

B. Horizon Detection from Segmentation Map

The output of our semantic segmentation model provides a detailed map, classifying each pixel into one of three categories: obstacles, water, and sky. This segmentation map serves as the foundation for our horizon detection process. The method is described in the following and an illustration of the pipeline is shown in Figure 6.

1) Semantic Segmentation: Initially, an image I is passed through a semantic segmentation model to obtain a segmentation map S:

$$S = f(I; \theta), \tag{1}$$

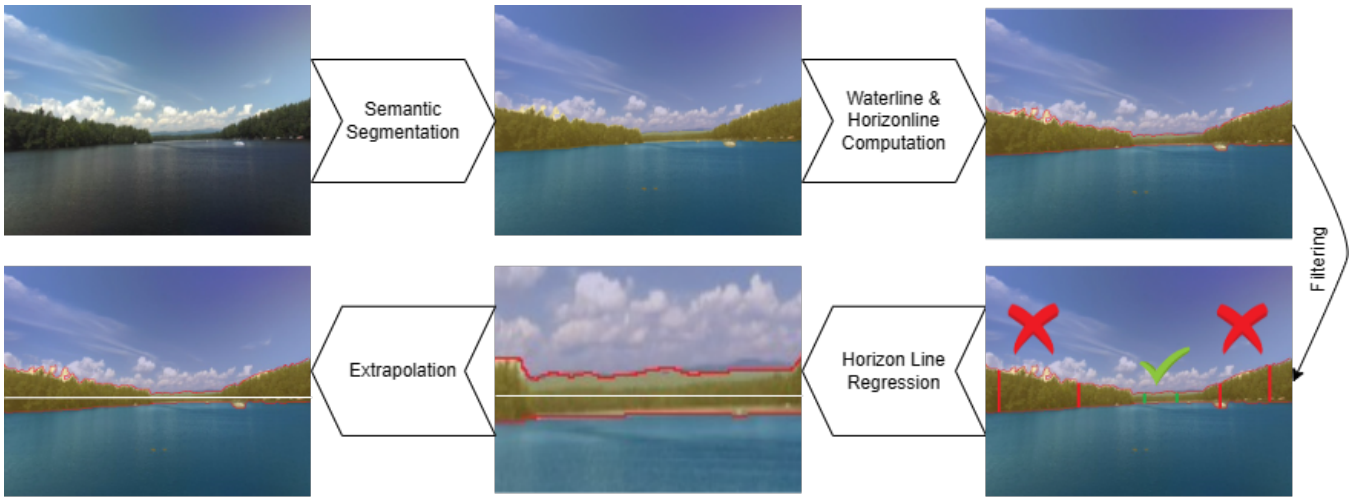

Fig. 6. Pipeline of our approach. First, we apply a semantic segmentation on the image frame, then we compute the waterline and horizon line, respectively. We filter out columns whose waterline and horizon line values are too far apart. Afterwards, we apply a linear regression on the remaining points. The resulting line is extrapolated to yield the final horizon line.

where S_{ij} denotes the segmentation class of the pixel at position (i, j), and θ encapsulates the model's parameters. The segmentation classes include sky, water, and obstacles.

2) Column-wise Analysis: For each column $S_{:,j}$ in the segmented image, we identify the highest water pixel (W_j) and the lowest sky pixel (K_j). The mean of these points is taken as the horizon point H_j for that column:

$$H_j = \frac{1}{2}\left(\min\left(i|S_{ij} = \text{sky}\right) + \max\left(i|S_{ij} = \text{water}\right)\right). \quad (2)$$

3) Linear Regression: A linear regression is then applied to the set of points (j, H_j) to fit the horizon line, minimizing the squared error between the predicted and actual horizon points:

$$m, b = \arg\min_{m,b} \sum_j (H_j - (mj + b))^2, \quad (3)$$

where the equation $y = mx + b$ represents the horizon line, with m being the slope and b the y-intercept.

4) Refinement Stage: The vanilla estimation of the horizon line suffers from outliers caused by a potentially ill-structured semantic segmentation map (water pixels above sky pixels), and, more generally, noisy pixel predictions. Hence, a refinement stage is employed to enhance the accuracy of the horizon detection. This process involves filtering and outlier removal based on the segmentation map and z-scores of the remaining data points.

Column Filtering The first step in the refinement process is to filter out columns in the segmented image that do not meet the criterion for a plausible horizon transition. Specifically, we exclude columns where the highest water pixel (W_j) is located above the lowest sky pixel (K_j), as such configurations contradict the expected natural order of sky above water or land:

$$\text{Valid columns: } W_j > K_j. \quad (4)$$

This filtering step ensures that only columns with potential horizon points are considered in subsequent analyses, thereby improving the robustness of the horizon detection.

Z-Score Calculation For the remaining data points, we calculate the z-score to identify outliers. The z-score for a data point is a measure of its deviation from the mean, expressed in terms of the standard deviation of the set of points. For each horizon point H_j in the valid columns, the z-score z_j is computed as follows:

$$z_j = \frac{H_j - \mu}{\sigma}, \quad (5)$$

where μ is the mean of the horizon points and σ is their standard deviation. This calculation standardizes the deviation of each point, facilitating the identification of outliers.

Outlier Removal Points with a z-score exceeding a predefined threshold are considered outliers and are removed from the dataset. The threshold is chosen based on the distribution of z-scores and the desired level of sensitivity in outlier detection. Typically, a threshold of $|z_j| > 2$ or 3 is used, corresponding to points that are more than two or three standard deviations away from the mean, respectively:

$$\text{Valid points for regression: } |z_j| \leq \tau, \quad (6)$$

where τ is the z-score threshold.

Final Horizon Line Fitting With outliers removed, a final linear regression is performed on the subset of valid, filtered points to determine the refined parameters of the horizon line. This produces a more accurate representation of the horizon, minimizing the influence of anomalous data points.

This refinement stage significantly enhances the precision of the horizon detection as shown in Section IV.

C. Horizon Locking based on Detected Horizon

Once the horizon line is accurately detected using the method outlined in the previous sections, the next step

is to apply appropriate image transformations to lock the horizon in a stable orientation across frames. This process significantly enhances the usability of marine vision systems by providing a consistent reference frame for navigation, object tracking, and augmented reality applications.

1) Image Transformation for Horizon Locking: Given the equation of the detected horizon line

$$y = mx + b,$$

where m is the slope and b is the y-intercept, we aim to transform the image such that the horizon line becomes parallel to the x-axis of the image. This transformation involves two primary steps: rotation and vertical shift.

a) Rotation: The first step is to rotate the image to correct any tilt in the horizon line. The angle θ required for rotation can be derived from the slope m of the horizon line:

$$\theta = \tan^{-1}(m), \tag{7}$$

where θ is the angle in radians. The image is then rotated by $-\theta$ to align the horizon line horizontally.

b) Vertical Shift: After rotation, the horizon line may not be positioned at the desired vertical location in the image. A vertical shift is applied to center the horizon line vertically within the frame. If h is the desired vertical position of the horizon line (e.g., the middle of the frame), the required shift Δy is calculated as:

$$\Delta y = h - b', \tag{8}$$

where b' is the new y-intercept of the horizon line after rotation. The entire image is then shifted vertically by Δy.

These image transformations ensure that the horizon line is consistently positioned and oriented across consecutive frames, providing a stable visual reference for further processing. This stabilization is crucial for enhancing the performance of marine vision systems in dynamic conditions. Notably, we don't correct for the tilting of the camera to the front or back as this is negligible in the whole process.

D. Dataset

We curated a dataset to validate our approach and provide a benchmark for the community. The dataset's salient features are as follows:

- **Rich Video Content**: The dataset comprises multiple RGB Full-HD videos captured in realistic boating scenarios using a Oak-D Wide Pro and a ZED 2. These videos provide a diverse range of maritime conditions, ensuring that our model is tested under various challenges.
- **Integrated IMU Data**: Alongside the video content, we have recorded IMU data, capturing roll, pitch, and yaw (absolute heading) measurements. This data provides a valuable reference for validating the accuracy of our horizon detection method. The IMU data comes from the cameras' onboard IMU sensors, which provide AHRS measurements. While for the ZED2 camera, the onboard IMU is not known, a comparison for Oak's IMU is shown in Table III.
- **Diverse Locations**: The videos were recorded in a variety of scenes, including Boston, Squam Lake, Lake Constance, Newport, and Portsmouth. This geographical diversity ensures that our dataset captures a wide range of marine environments and conditions.
- **Manually Labeled Horizon**: To provide a ground truth for horizon detection, each video frame in our dataset has been manually annotated to mark the horizon, ensuring that our evaluations are based on accurate and reliable ground truth data.
- **Object Detection and Tracking Annotations**: Beyond horizon detection, our dataset also includes annotations for object detection and tracking. This makes our dataset versatile, catering to a broader range of maritime computer vision tasks.
- **Public Availability**: We have made it publicly available (see Footnote 1). We believe that this will serve as a robust benchmark for horizon prediction and related tasks, fostering further advancements in the field.

The dataset's comprehensive nature, combined with our semantic segmentation-based approach, provides a holistic solution to the challenges of real-time horizon locking in marine settings. By making our dataset publicly available, we aim to catalyze further research and innovation in this domain. See an overview of the dataset in Table II.

TABLE II

DIFFERENT DATASETS CAPTURED. EACH CLIP HAS A LENGTH OF AT LEAST TEN SECONDS. COLLECTED IN VARIOUS CONDITIONS, SUCH AS CHOPPY WATERS (CHW), CITY HORIZON (CIH), EMPTY HORIZON (EMH), HARBOR SCENES (HARS), BUSY WATERWAYS (BUW), CALM SEA (CAS), LANDMASS-OBSTRUCTED HORIZON VIEW (LOH), FAST DRIVES (FAD). #V DENOTES THE NUMBER OF VIDEOS.

Location	Cam	#V	Properties
Boston	Oak	5	ChW, CiH, EmH
Newport	Oak	8	HarS, BuW
Squam Lake	Oak	4	CaS, LoH
Lake Constance	ZED2	3	CaS, EmH
Portsmouth	ZED2	3	ChW, FaD

TABLE III

ACCURACY PERFORMANCE OF OAK-D PRO IMU (BNO08X) [34]. NOTABLY, THE MANUFACTURER REPORTS A TYPICAL ACCURACY OF 5° IN PRACTISE.

Performance Metric	**Dynamic Accuracy**
Rotation Vector	3.5°
Accelerometer	0.3 m/s^2
Gyroscope	3.1°/s
Magnetometer	1.4uT

TABLE IV

HORIZON PREDICTION ACCURACY AS MEASURED IN ROLL ANGLE ERROR AND BORDER PIXEL OFFSET ON SEVERAL DATASETS IN DIFFERENT CONDITIONS. OUR SEGMENTATION-BASED APPROACH SIGNIFICANTLY OUTPERFORMS THE IMU-ONLY AND TRADITIONAL CV-BASED APPROACHES.

	Model	Roll error (°)	Border offset (pixels)
BOSTO	IMU-only	3.5	25
	CV	2.4	18
	Ours	**1.2**	**8**
NEWPO	IMU-only	3.8	28
	CV	2.8	20
	Ours	**1.5**	**10**
SQUAM	IMU-only	4.0	30
	CV	5.2	35
	Ours	**1.8**	**12**
CONST	IMU-only	3.6	26
	CV	**1.4**	10
	Ours	**1.4**	**9**
PORTS	IMU-only	3.9	29
	CV	3.0	21
	Ours	**1.6**	**11**

TABLE V

MULTI-OBJECT TRACKING ACCURACY ON DIFFERENT DATASETS. COMPARISON BETWEEN STANDARD AND HORIZON-LOCKED MECHANISMS ON SEVERAL SEA CONDITIONS.

	Model	HOTA↑	MOTA↑	IDs↓	Frag↓
BOS	ByteTracker	79.9	89.8	23	678
	+ Hor. Lock	**81.5**	**91.5**	**16**	**635**
NEW	Standard	65.6	67.0	69	876
	+ Hor. Lock	**67.4**	**69.8**	**59**	**825**
SQU	Standard	66.0	67.5	68	870
	+ Hor. Lock	**68.2**	**70.2**	**58**	**810**
CON	Standard	66.2	67.7	67	865
	+ Hor. Lock	**68.5**	**70.5**	**57**	**805**
POR	Standard	66.4	68.0	66	860
	+ Hor. Lock	**68.8**	**71.0**	**56**	**800**

IV. EXPERIMENTAL RESULTS

To validate the efficacy of our proposed method, we conducted a series of experiments comparing our approach against several baseline methods. This section provides a detailed overview of our experimental setup, the methodologies employed, and the results obtained.

A. Experimental Setup

- **Comparison Baselines**:
  1) **IMU-only**: For this baseline, we solely rely on the IMU data to estimate the horizon. The horizon's angle is computed based on the roll and pitch measurements from the IMU and the installed height [10]. This method serves as a representative of traditional hardware-based horizon detection techniques.
  2) **CV/Feature-based**: This method employs traditional computer vision techniques for horizon detection. We utilize feature detection and extraction methods, such as SIFT or SURF, followed by a Hough transform to identify the horizon line. This approach represents the conventional software-based techniques for horizon detection [35].
- **Evaluation Metrics**:
  1) **Roll Angle Error**: This metric quantifies the difference in the roll angle of the detected horizon from the ground truth. A lower roll angle error indicates a more accurate horizon detection.
  2) **Border Offset Error**: We compute the offset errors at both the left and right image borders. This metric measures the vertical distance between the detected horizon and the ground truth at the image's edges. A smaller offset error indicates that the detected horizon closely aligns with the actual horizon, especially at the image boundaries.

With the setup defined, we proceeded to evaluate our semantic segmentation-based method against the aforementioned baselines using our comprehensive dataset, as described in Table II. We manually annotate each frame with a horizon location on the left and right image borders, respectively.

B. Results

Table IV shows how our segmentation-based approach outperforms the traditional methods significantly. This is reflected by both, the lower roll angle error and lower border pixel offset. Figure 8 shows the case when an object partly occludes the horizon. Figure 9 shows the case when the horizon is only partly visible. Figure 2 showcases some of the problems of IMU-based methods and Figure 3 shows the limitations of traditional CV-based methods.

Furthermore, we test what efffect the horizon locking has on the Muli-Object Tracking (MOT) performance has. For that, we test a MOT model on the individual subsets with and without horizon locking.

Table V shows that we significantly improve the performance of ByteTracker by locking the horizon. Using the horizon location, we lock the horizon to the center of the image across time. The improvement is caused by the more stable positioning of objects across time, as Figure 7 illustrates.

Lastly, we test the horizon locking on an AR application where we lay a grid on the water line. Figure 5 illustrates that the grid is correctly laid on the water surface.

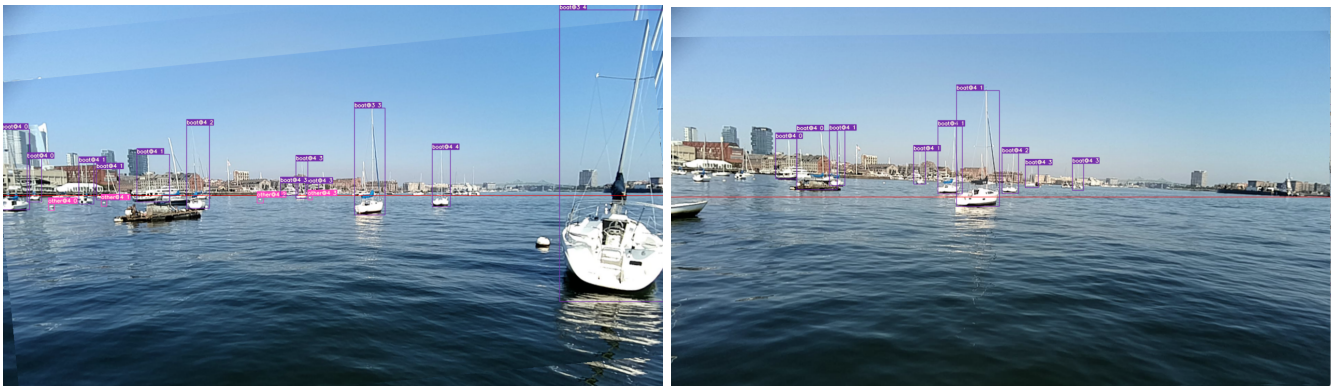

Fig. 7. Application example multi-object tracking improvement. Objects are tracked successfully from frame 0 to frame 60 after locking the horizon to the center of the screen (see the rotation of the video via the boundaries on the top edges).

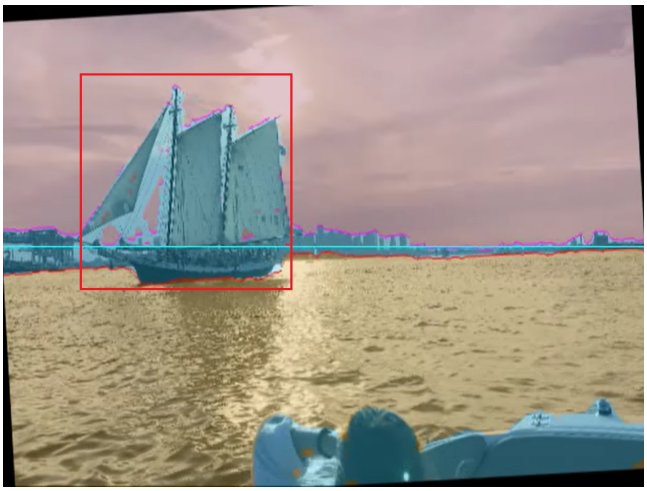

Fig. 8. Robust results when an object occludes the scene.

Fig. 10. Limitation of our approach in situations where the horizon is nowhere visible as the upmost waterline and downmost skyline are too far part, and we must revert to IMU-based horizon prediction.

Fig. 9. Robust results when the horizon is only partly visible. The white line indicates the regression line, the purple is the extrapolated line. The blue color overlay on the dock is caused by errors of the semantic segmentation method.

V. Conclusion, Limitations and Future Work

This paper presented a novel, CV-based approach for robust, real-time horizon locking in marine settings. Our method outperforms traditional techniques, offering a reliable solution for an industry increasingly relying on automated systems. One limitation of our approach is that it does not work when the horizon is not visible, as shown in Figure 10, or when the lighting conditions are poor. However, as we know that it is not visible, we can actively decide to fall back to IMU-based horizon prediction, making the system interpretable.

Future work aims to fuse IMU-based and segmentation-based techniques for more robust performance in bad weather and lighting conditions, for which more data needs to be collected. Furthermore, it would be interesting to see how the improved horizon detection affects a geometric-based distance prediction on the water surface.

REFERENCES

[1] B. Kiefer, M. Kristan, J. Perš, L. Žust, F. Poiesi, F. Andrade, A. Bernardino, M. Dawkins, J. Raitoharju, Y. Quan, A. Atmaca, T. Höfer, Q. Zhang, Y. Xu, J. Zhang, D. Tao, L. Sommer, R. Spraul, H. Zhao, H. Zhang, Y. Zhao, J. L. Augustin, E.-i. Jeon, I. Lee, L. Zedda, A. Loddo, C. Di Ruberto, S. Verma, S. Gupta, S. Muralidhara, N. Hegde, D. Xing, N. Evangeliou, A. Tzes, V. Bartl, J. Špaňhel, A. Herout, N. Bhowmik, T. P. Breckon, S. Kundargi, T. Anvekar, R. A. Tabib, U. Mudenagudi, A. Vats, Y. Song, D. Liu, Y. Li, S. Li, C. Tan, L. Lan, V. Somers, C. De Vleeschouwer, A. Alahi, H.-W. Huang, C.-Y. Yang, J.-N. Hwang, P.-K. Kim, K. Kim, K. Lee, S. Jiang, H. Li, Z. Ziqiang, T.-A. Vu, H. Nguyen-Truong, S.-K. Yeung, Z. Jia, S. Yang, C.-C. Hsu, X.-Y. Hou, Y.-A. Jhang, S. Yang, and M.-T. Yang, "1st workshop on maritime computer vision (macvi) 2023: Challenge results," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, January 2023, pp. 265–302.

[2] L. Žust, J. Perš, and M. Kristan, "Lars: A diverse panoptic maritime obstacle detection dataset and benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 304–20 314.

[3] B. Kiefer, L. Žust, M. Kristan, J. Perš, M. Teršek, A. Wiliem, M. Messmer, C.-Y. Yang, H.-W. Huang, Z. Jiang, H.-C. Kuo, J. Mei, J.-N. Hwang, D. Stadler, L. Sommer, K. Huang, A. Zheng, W. Chong, K. Lertniphonphan, J. Xie, F. Chen, J. Li, Z. Wang, L. Zedda, A. Loddo, C. Di Ruberto, T.-A. Vu, H. Nguyen-Truong, T.-S. Ha, Q.-D. Pham, S.-K. Yeung, Y. Feng, N. T. Thien, L. Tian, A. Michel, W. Gross, M. Weinmann, B. Carrillo-Perez, A. Klein, A. Alex, E. Solano-Carrillo, Y. Steiniger, A. B. Rodriguez, S.-Y. Kuan, Y.-H. Ho, F. Sattler, M. Fabijanić, M. Šimunec, and N. Kapetanović, "2nd workshop on maritime computer vision (macvi) 2024: Challenge results," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, January 2024, pp. 869–891.

[4] B. Kiefer, T. Höfer, and A. Zell, "Stable yaw estimation of boats from the viewpoint of uavs and usvs," in *2023 European Conference on Mobile Robots (ECMR)*, 2023, pp. 1–6.

[5] E. Gershikov, T. Libe, and S. Kosolapov, "Horizon line detection in marine images: which method to choose?" *International Journal on Advances in Intelligent Systems*, vol. 6, no. 1, 2013.

[6] T. Praczyk, "A quick algorithm for horizon line detection in marine images," *Journal of Marine Science and Technology*, vol. 23, pp. 164–177, 2018.

[7] M. A. Hashmani, M. Umair, S. S. H. Rizvi, and A. R. Gilal, "A survey on edge detection based recent marine horizon line detection methods and their applications," in *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 2020, pp. 1–5.

[8] T. Libe, E. Gershikov, and S. Kosolapov, "Comparison of methods for horizon line detection in sea images," *Proceedings of the Content*, pp. 79–85, 2012.

[9] S. Singhal, Y. Ao, D. Maas, B. Arsenali, and S. Maranò, "Marine vessel attitude estimation from coastline and horizon," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 6149–6154.

[10] B. Kiefer and A. Zell, "Fast region of interest proposals on maritime uavs," *arXiv preprint arXiv:2301.11650*, 2023.

[11] D. Capriglione, M. Carratu, M. Catelani, L. Ciani, G. Patrizi, R. Singuaroli, and P. Sommella, "Experimental analysis of imu under vibration," in *16th IMEKO TC10 Conference:"Testing, Diagnostics & Inspection as a comprehensive value chain for Quality & Safety*, 2019, pp. 26–31.

[12] A. F. Fadhil, R. Kanneganti, and L. Gupta, "Runway and horizon detection through fusion of enhanced vision system and synthetic vision system images," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2014, p. 1.

[13] Y. Liu, L. Ma, W. Xie, X. Zhang, and Y. Zhang, "Water boundary line detection for unmanned surface vehicles," *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, vol. 13, no. 8, pp. 1145–1152, 2020.

[14] Y. Wei and Y. Zhang, "Effective waterline detection of unmanned surface vehicles based on optical images," *Sensors*, vol. 16, no. 10, p. 1590, 2016.

[15] L. Steccanella, D. D. Bloisi, A. Castellini, and A. Farinelli, "Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring," *Robotics and Autonomous Systems*, vol. 124, p. 103346, 2020.

[16] H. Guo, Y.-M. Zhang, J. Zhou, and Y.-Q. Zhang, "A fast and robust vision-based horizon tracking method," in *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2015, pp. 71–74.

[17] H. Lu, Y. Li, and S. Serikawa, "Computer vision for ocean observing," *Artificial intelligence and computer vision*, pp. 1–16, 2017.

[18] D. Levy, Y. Belfer, E. Osherov, E. Bigal, A. P. Scheinin, H. Nativ, D. Tchernov, and T. Treibitz, "Automated analysis of marine video with limited data," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1385–1393.

[19] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 18–26.

[20] L. Xu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Deep learning for marine species recognition," *Handbook of deep learning applications*, pp. 129–145, 2019.

[21] M. Waszak, A. Cardaillac, B. Elvesæter, F. Rødølen, and M. Ludvigsen, "Semantic segmentation in underwater ship inspections: Benchmark and data set," *IEEE Journal of Oceanic Engineering*, vol. 48, no. 2, pp. 462–473, 2022.

[22] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE transactions on multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[23] B. Iancu, V. Soloviev, L. Zelioli, and J. Lilius, "Aboships—an inshore and offshore maritime vessel detection dataset with precise annotations," *Remote Sensing*, vol. 13, no. 5, p. 988, 2021.

[24] L. Su, Y. Chen, H. Song, and W. Li, "A survey of maritime vision datasets," *Multimedia Tools and Applications*, vol. 82, no. 19, pp. 28 873–28 893, 2023.

[25] B. Bovcon, J. Muhovič, D. Vranac, D. Mozetič, J. Perš, and M. Kristan, "Mods—a usv-oriented object detection and obstacle segmentation benchmark," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13 403–13 418, 2021.

[26] A. Shurin, A. Saraev, M. Yona, Y. Gutnik, S. Faber, A. Etzion, and I. Klein, "The autonomous platforms inertial dataset," *IEEE Access*, vol. 10, pp. 10 191–10 201, 2022.

[27] Chalmers, "Reeds Dataset," https://reeds.opendata.chalmers.se/, accessed: 2024-01-01.

[28] K. Chen, M. Wu, J. Liu, and C. Zhang, "Fgsd: A dataset for fine-grained ship detection in high resolution satellite images," *arXiv preprint arXiv:2003.06832*, 2020.

[29] J. Sjöblom, "Estimating a boat's vertical velocity with unpositioned 6dof imu: s: How sensor fusion and knowledge of the system dynamics can be used to estimate the imu positions and produce fused estimates," 2023.

[30] D. Chung, J. Kim, C. Lee, and J. Kim, "Pohang canal dataset: A multimodal maritime dataset for autonomous navigation in restricted waters," *The International Journal of Robotics Research*, vol. 42, no. 12, pp. 1104–1114, 2023.

[31] S. Yao, R. Guan, Z. Wu, Y. Ni, Z. Zhang, Z. Huang, X. Zhu, Y. Yue, Y. Yue, H. Seo, *et al.*, "Waterscenes: A multi-task 4d radar-camera fusion dataset and benchmark for autonomous driving on water surfaces," *arXiv preprint arXiv:2307.06505*, 2023.

[32] M. Teršek, L. Žust, and M. Kristan, "ewasr—an embedded-compute-ready maritime obstacle detection network," *Sensors*, vol. 23, no. 12, p. 5386, 2023.

[33] B. Bovcon and M. Kristan, "Wasr—a water segmentation and refinement maritime obstacle detection network," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 12 661–12 674, 2021.

[34] CEVA, Inc., "BNO085/BNO086 9-AXIS SYSTEM IN PACKAGE (SIP) IMU Datasheet," https://www.ceva-dsp.com/wp-content/uploads/2019/10/BNO080_085-Datasheet.pdf, accessed: 2024-02-25.

[35] A. M. Neto, A. C. Victorino, I. Fantoni, and D. E. Zampieri, "Robust horizon finding algorithm for real-time autonomous navigation based on monocular vision," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 532–537.