sensors

MDPI

Article

Using Conventional Cameras as Sensors for Estimating Confidence Intervals for the Speed of Vessels from Single Images

Jose L. Huillca *,† and **Leandro A. F. Fernandes** †

Instituto de Computação, Universidade Federal Fluminense (UFF), Avenida General Milton Tavares de Souza, s/n, Niterói 24210-346, RJ, Brazil; laffernandes@ic.uff.br

* Correspondence: josehuillca@id.uff.br

† These authors contributed equally to this work.

Abstract: In this paper, we describe an image-based approach for estimating the speed of a moving vessel using the wakes that remain on the surface of water after the vessel has passed. The proposed method calculates the speed of the vessel using only one RGB image. In this study, we used the vanishing line of the mean water plane, the camera height concerning the level of the tide, and the intrinsic parameters of the camera to perform geometric rectification on the surface plane of the water. We detected the location of troughs on one of the wake arms and computed the distance between them in the rectified image to estimate the speed of the vessel as a so-called inverse ship wake problem. We used a radar that was designed to monitor ships to validate the proposed method. We used statistical studies to determine the reliability and error propagation of the estimated values throughout the calculation process. The experiments showed that the proposed method produced precise and accurate results that agreed with the actual radar data when using a simple capture device, such as a conventional camera.

Keywords: vessel speed; Kelvin wake; ship wake; error propagation; object detection; planar homography; metrology

check for updates

Citation: Huillca, J.L.; Fernandes, L.A.F. Using Conventional Cameras as Sensors for Estimating Confidence Intervals for the Speed of Vessels from Single Images. *Sensors* **2022**, *22*, 4213. https://doi.org/10.3390/s22114213

Academic Editor: Gemine Vivone

Received: 3 March 2022

Accepted: 12 Apri 2022

Published: 1 June 2022

1. Introduction

Radar-based electronic navigation systems (ENSs) that keep the pilot informed about the location and speed of nearby vessels have been a significant technological advance for the maritime industry. Unfortunately, radars and other sensors may fail to detect stealth ships and non-metallic targets because they reflect a low amount of radiation. Additionally, many vessels are not equipped with an ENS. As a result, naked eye visibility still plays a key role in making decisions, especially at close range. Going further, projects have been carried out on the creation of autonomous vessels [1,2], in which autonomous navigation makes use of motion flow techniques, such as FeatFlow [3], as auxiliary sources of information.

Vessel detection and tracking using computer vision-based systems are convenient methods for measuring vessel speed [4]. A large number of algorithms estimate the vessel speed by analyzing the traces that are left by the vessel using synthetic-aperture radar (SAR) imagery [5,6]. Techniques, such as the CRSN [7], have been used to improve the quality of SAR images and CenterNet++ [8] performs vessel detection. In the work developed by Liu et al. [9], ship wakes were used to detect vessels and to identify the position and direction of the vessel using optical images. Although these techniques provide good results, the SAR and optical images that were considered in the above-mentioned studies were not captured at close range; they were generated by airborne sensors or satellites and could not be obtained from inside an autonomous vessel.

Other techniques perform tracking and estimations of speed using image sequences that were taken with a digital camera [10–12]. However, video-based techniques require fixed cameras to estimate speed from the relative motion. Therefore, this type of method is limited to applications in which the camera is onshore, since cameras that are placed on autonomous vessels and drones produce additional movement in the optical flow.

In a recent study, Huillca and Fernandes [13] presented a semi-automatic method to calculate the speed of a vessel directly from one image that was acquired by a conventional camera. The method is based on projective geometry and estimates vessel speed from an analysis of the Kelvin wake pattern. The key observation is that naval objects leave traces of their movements on the surface of the water and the appearance of these traces is related to speed. Actually, as Lord Kelvin demonstrated in 1887, the wakes that are left by vessels maintaining a constant course and speed can be modeled as a function of speed [14]. The inverse ship wake problem consists of estimating parameters, such as speed, from the observation of wake patterns [15]. The approach that was proposed by Huillca and Fernandes [13] detects features (crests or troughs) of the extracted wave arm and applies the inverse ship wake problem to estimate the speed of the corresponding vessel. However, the technique requires the horizon line to be visible in the image. The curves that are adjusted to the wave arms are also sensitive to noise and the analysis of the results of the study was limited to a few test cases using ground truth that was based on rough estimates of the maximum and minimum speeds that were reached by the observed vessels.

This paper presents an extended version of the approach that was proposed by Huillca and Fernandes [13]. The extension included the automatic estimation of the vanishing line, a different approach to identifying the wave arms, the use of the troughs in the closest wave arm to estimate the wavelength, the use of radar to validate the results, the estimation of confidence intervals for the measurements, and the analysis of error propagation throughout the computational chain. Unlike previous work, the use of a more sophisticated vanishing line detection method allowed for the application of the new method even when the line between the sky and the water was not visible. Our new clustering strategy for identifying the two wave arms made our approach less prone to misidentification. In the previous study, the validation was performed using only the speed information that was presented on the Rio de Janeiro Ferry Services website. In this work, we verified the accuracy and reliability of the measurements that were obtained through our method using ground truth data that were collected from a radar that was designed to monitor vessels, including ferries and other types of vessels. We also used a sampling-based method and first-order error propagation [16] to estimate the uncertainty of the estimations that were produced using our technique. We showed that the confidence intervals that were obtained with error propagation were equivalent to those that were computed by sampling, but also had the advantage of allowing for uncertainty estimation when using one image. Our results were consistent and agreed with the values that were obtained by the radar and did not present the limitations of video-based approaches, since all estimates were performed using a single image.

2. Computing Vessel Speed

For the proposed method, it is assumed that the camera is mounted at a given height h above sea level in such a way that the target ship and the traces that are left by the target ship can be observed (see Figure 1). The camera could be onshore, aboard a vessel or on a moving drone. The orientation of the camera in world space is estimated during the process. The speed is computed using the wavelength of the trace information. For each acquired image, the processing steps include: (i) the estimation of the vanishing line of the mean water plane; (ii) the definition of the corners of the region of interest (ROI), including the ship wake; (iii) the identification of the wave arms and troughs that are present in the ROI; and (iv) the estimation of the speed of the vessel using the wavelength.

The notation adopted in this article was inspired by Hartley and Zisserman [17] and is shown in Table 1.

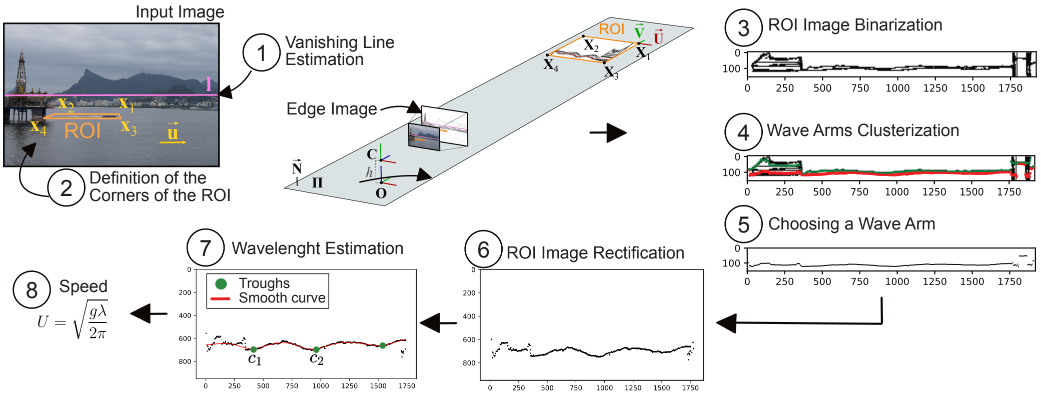


Figure 1. The pipeline followed to estimate the vessel speed. We intentionally flipped the rectified region of interest (ROI) image to make the vessel travel to the left.

Table 1. Notation convention that is used in this article.

Notation	Meaning
$\mathcal{I}, \mathcal{B}$	Input and edge image, respectively
$\mathbf{\Pi}$	Water plane
$\mathbf{x}_i$	The i-th point in image space under homogeneous coordinates
$\mathbf{X}_i$	The i-th point in world space under homogeneous coordinates
$x_{\mathbf{x}_i}, y_{\mathbf{x}_i}$	Coordinates of point $\mathbf{x}_i$ in image space
$X_{\mathbf{X}_i}, Y_{\mathbf{X}_i}, Z_{\mathbf{X}_i}$	Coordinates of point $\mathbf{X}_i$ in world space
$\mathbf{l}$	Vector encoding the vanishing line in image space
$\vec{\mathbf{v}}$	Vector in image space under homogeneous coordinates
$\vec{\mathbf{V}}$	Vector in world space under homogeneous coordinates
$\mathbf{M}$	An $m \times n$ matrix
$\mathbf{M}^{-1}$	Inverse of a matrix $\mathbf{M}$
$\mathbf{M}^T$	Transpose of a matrix $\mathbf{M}$

2.1. Vanishing Line Estimation

We used the Horizon Line in the Wild (HLW) algorithm [18] to automatically detect the vanishing line of the water body. The HLW algorithm estimated the left- and right-hand endpoints of the vanishing line of the most prominent plane that was observed in the input color image $\mathcal{I}$. In our equations, we represented the endpoints using homogeneous coordinates as vectors $\mathbf{p}_l = (0, y_{\mathbf{p}_l}, 1)^T$ and $\mathbf{p}_r = (W-1, y_{\mathbf{p}_r}, 1)^T$, where W is the width of $\mathcal{I}$ and $y_{\mathbf{p}_l}$ and $y_{\mathbf{p}_r}$ are the vertical coordinates of the points that were returned by the HLW algorithm. The vector $\mathbf{l} = (A, B, C)^T$ that encoded the vanishing line of the mean water plane in homogeneous coordinates could be computed as the cross product of vectors $\mathbf{p}_l$ and $\mathbf{p}_r$ [17]:

$$\mathbf{l} = (A, B, C)^T = \mathbf{p}_l \times \mathbf{p}_r = \left(y_{\mathbf{p}_l} - y_{\mathbf{p}_r}, W-1, (1-W)y_{\mathbf{p}_l}\right)^T, \tag{1}$$

where A, B, and C are the coefficients of the general equation of the line $Ax + By + Cw = 0$.

It is important to emphasize that the horizon line did not need to be visible for the HLW algorithm to estimate the endpoints of the vanishing line of the mean water plane. For instance, notice in Figure 1 that the horizon line would be behind the mountains. For the RANSAC-based technique that was used by Huillca and Fernandes [13], on the other hand, the horizon line between the sky and the water must be visible because it is detected as the most apparent straight line in the edge image.

2.2. Definition of the Corners of the ROI

The ROI had to include the ship wake. It was defined in $\mathcal{I}$ as the quadrilateral that resulted from the projection of a rectangular region onto the water surface (see Figure 1). We found the set of corners $\{\mathbf{x}_k\}_{k=1}^{4}$ of the ROI in $\mathcal{I}$ using the reference corner $\mathbf{x}_1 = (x_{\mathbf{x}_1}, y_{\mathbf{x}_1}, 1)^T$, the direction $\vec{\mathbf{u}} = (x_{\vec{\mathbf{u}}}, y_{\vec{\mathbf{u}}}, 0)^T = (\cos\phi_{\vec{\mathbf{u}}}, \sin\phi_{\vec{\mathbf{u}}}, 0)^T$ of the vessel in the image, the vanishing line $\mathbf{l}$ of the water (1), the camera calibration matrix $\mathbf{K}$, the ROI size $W_{ROI} \times H_{ROI}$ (in meters) in 3D space, and the camera height h above sea level.

Using techniques such as YOLO [19] and the training model that was used by Breitinger et al. [20] to detect the vessels is straightforward. However, as this was a separate problem that was subject to the quality of the detection technique without the loss of generality, we chose to manually set $\mathbf{x}_1$ and $\vec{\mathbf{u}}$ in image space in our experiments. The selection condition for $\mathbf{x}_1$ was to place this point in image space to the left of the vessel in 3D space. It could be placed either near the bow or near the stern. We set $\mathbf{x}_1$ close to the bow for all images in our experiments. The definition of direction $\vec{\mathbf{u}}$ in image space was straightforward. It could be defined by tracing a line segment from the stern to the bow or it could be calculated from the edge pixels of the wake as the eigenvector with the largest eigenvalue pointing to the bow. In this work, we extracted metadata from the image files to compute $\mathbf{K}$. W_{ROI} and H_{ROI} were constant values that were defined by the user.

The camera height h was given by the construction and was defined as the distance between the camera and the mean water plane in meters. We assumed that the origin $\mathbf{O}$ of the world coordinate system (see Figure 1) lay on the orthogonal projection of the camera center $\mathbf{C}$ to the mean water plane $\mathbf{\Pi}$, that the X and Y axes spanned $\mathbf{\Pi}$, and that the Z axis was parallel to the vector $\vec{\mathbf{N}}$ that was normal to $\mathbf{\Pi}$. We set $\mathbf{C} = (0, 0, h, 1)^T$ and computed its orientation with respect to the world's frame from the vanishing line $\mathbf{l}$. Therefore, even though the camera could move with six degrees of freedom, only one (the height) had to be known a priori.

When a camera is mounted on a vessel's mast, its height h can be calculated from the mast's height and its relative orientation to the normal vector $\vec{\mathbf{N}}$ (computed from the vanishing line). For drones, the vehicle needs to be able to determine its altitude. In our experiments, the camera was mounted on a tripod that was inside a building. In this case, the camera height above the sea level was calculated as the sum of the heights h_g, h_f, h_t, and h_s representing the ground height, floor height, tripod height, and tide height, respectively:

$$h = h_g + \delta h_f + h_t - h_s, \tag{2}$$

where δ is the floor number on which the camera was mounted.

We let $\{\mathbf{X}_k\}_{k=1}^{4}$ be the set of corners of the ROI lying on plane $\mathbf{\Pi}$. By tracing a ray from the camera to $\mathbf{\Pi}$ through point $\mathbf{x}_1$, we computed:

$$\mathbf{X}_1 = (X_{\mathbf{X}_1}, Y_{\mathbf{X}_1}, 0, 1)^T = (-hX_{\vec{\mathbf{D}}}/Z_{\vec{\mathbf{D}}}, -hY_{\vec{\mathbf{D}}}/Z_{\vec{\mathbf{D}}}, 0, 1)^T, \tag{3}$$

where $\vec{\mathbf{D}} = (X_{\vec{\mathbf{D}}}, Y_{\vec{\mathbf{D}}}, Z_{\vec{\mathbf{D}}})^T = \mathbf{M}^{-1}\mathbf{x}_1$ is the direction of the traced ray, $\mathbf{M} = \mathbf{KR}$, and $\mathbf{R}$ is a rotation matrix whose columns correspond to the X, Y, and Z axes of the camera in the world's frame of reference (the red, green, and blue segments leaving $\mathbf{C}$ in Figure 1, respectively). The columns of $\mathbf{R}$ were $\vec{\mathbf{R}}_1 = \vec{\mathbf{R}}_3 \times \vec{\mathbf{R}}_2$, $\vec{\mathbf{R}}_2 = \operatorname{unit}((1, 0, 0)^T \times \vec{\mathbf{R}}_3)$, and $\vec{\mathbf{R}}_3 = \operatorname{unit}(\operatorname{up}(\mathbf{K}^T\mathbf{l}))$. The unit function normalized the vector to unit length and the up function changed the orientation of $\vec{\mathbf{R}}_3$ when its $Y_{\vec{\mathbf{R}}_3}$ coordinate was negative. It was necessary to correct the hand of the camera coordinate system by forcing the $\vec{\mathbf{R}}_2$ vector to point upward, as with the normal vector $\vec{\mathbf{N}}$ of the mean water plane and the Z axis of the world's frame of reference (see Figure 1), while $\vec{\mathbf{R}}_1$ and $\vec{\mathbf{R}}_3$ pointed to the right and the front, respectively.

$\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$ were computed by translating $\mathbf{X}_1$ by W_{ROI} and H_{ROI} in directions $\vec{\mathbf{U}} = \text{unit}(\mathbf{A}\vec{\mathbf{E}})$ and $\vec{\mathbf{V}} = \mathbf{B}\vec{\mathbf{U}}$ on plane $\mathbf{\Pi}$:

$$\mathbf{X}_2 = \mathbf{X}_1 - W_{ROI}\vec{\mathbf{U}}, \tag{4a}$$

$$\mathbf{X}_3 = \mathbf{X}_1 - H_{ROI}\vec{\mathbf{V}}, \tag{4b}$$

$$\mathbf{X}_4 = \mathbf{X}_1 - W_{ROI}\vec{\mathbf{U}} - H_{ROI}\vec{\mathbf{V}}, \tag{4c}$$

where $\vec{\mathbf{U}}$ is the direction of the vessel in world space and $\vec{\mathbf{V}}$ is the perpendicular direction that was computed by rotating $\vec{\mathbf{U}}$ in $\mathbf{\Pi}$ through an angle of $\pi/2$. Thus, $\mathbf{B}$ was a constant rotation matrix. Here, $\vec{\mathbf{E}} = (X_{\vec{\mathbf{E}}}, Y_{\vec{\mathbf{E}}}, Z_{\vec{\mathbf{E}}})^T = \mathbf{M}^{-1}\vec{\mathbf{u}}$ was the back projection of $\vec{\mathbf{u}}$ (an improper point) onto the world coordinate system and $\mathbf{A}$ encoded the orthogonal projection onto $\mathbf{\Pi}$.

Finally, the corners of the ROI in $\mathcal{I}$ were computed as:

$$\mathbf{x}_k = \mathbf{P}\mathbf{X}_k, \tag{5}$$

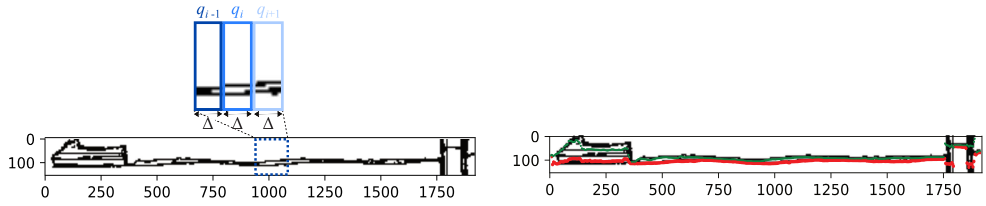
where $k \in \{2,3,4\}$, $\mathbf{P} = \mathbf{M}\big(\mathbf{I} \mid -\tilde{C}\big)$ is the 3×4 camera matrix, $\mathbf{I}$ is a 3×3 identity matrix, and $\tilde{C} = (0,0,h)^T$.

2.3. Finding the Wave Arms

We used the edge image $\mathcal{B}$ of the color image $\mathcal{I}$ to find the wave arms. To avoid processing the whole image or working with a non-rectangular ROI, the procedure for finding the wave arms considered a small portion of $\mathcal{B}$ that was defined as the axis-aligned bounding box of the ROI. Thus, in the remainder of this section, all described processing was restricted to that portion of $\mathcal{B}$.

In this work, we used the Richer Convolutional Features (RCF) algorithm [21] to compute the edge image $\mathcal{B}$ from $\mathcal{I}$. RCF helped to detect the wakes that were left by the vessels by making the waves more visible. Among the edge detection strategies that we tested, RCF was less sensitive to weather conditions and poor natural light.

The edge image $\mathcal{B}$ that was produced by RCF was an intensity image, in which 1 indicated that the pixel had a high chance of being an edge while 0 meant the opposite. We used the Otsu algorithm to find an automatic threshold t to separate the pixels into the two classes, i.e., edge and non-edge, and compute the binary image $\mathcal{B}^*$. We used the k-means algorithm [22] with $k = 2$ to differentiate the two wave arms that were present in the wakes that were left by the ships. The image coordinates of the edge pixels were taken as the inputs for the algorithm. As illustrated in Figure 2a, the k-means algorithm was not directly applied to the entire binary image. We divided the portion of $\mathcal{B}^*$ that was inside the bounding box of the ROI into vertical partitions q_i, each with a width of Δ pixels and height that was equal to the height of the bounding box. In each partition q_i, we applied the k-means algorithm to obtain the centroid of each wake arm. The k-means points in partition q_i were equal to the centroids that were calculated in q_{i-1}, except for the first partition ($i = 1$) or when no centroid was detected in q_{i-1}. For those cases, the two points were taken as random. The points that were obtained as the centroids defined the samples for the curves that described each wake arm. Figure 2a highlights three consecutive partitions, while Figure 2b shows the detected centroids using green and red for the most distant and closest wake arms, respectively. In our experiments, we set Δ to 5 pixels. According to our experience, the results of the curve fitting process that follows did not differ when a Δ value between 3 and 9 pixels was chosen.



(**a**) (**b**)

Figure 2. Finding the wave arms: (**a**) we applied the k-means algorithm to the edge pixels that were included in each partition q_i of the ROI's bounding box and the partitions each had a width of Δ pixels; (**b**) the detected wave arms.

2.4. Wavelength and Speed Estimation

To solve the inverse Kelvin wake problem, we needed to calculate the Euclidean distance between at least two consecutive troughs or crests in the world coordinate system. A key observation is that the image of the wave arm that was closest to the camera was the least affected by the turbulence of the wake. In addition, its troughs were less affected by errors when we performed the rectification of the ROI. The steps for performing wavelength and speed estimation are described below.

2.4.1. ROI Image Rectification

The objective of the ROI rectification was to eliminate the projective distortion that was introduced by the camera from the image of the mean water plane, thereby simulating an aerial view of the ROI that was similar to the sketch that is presented in Figure 3. We used the line at infinity $\mathbf{l}_\infty$ of the water plane to remove affine distortion and the camera height above sea level to eliminate projective ambiguity. The line at infinity $\mathbf{l}_\infty$ allowed for the recovery of the related properties of image elements, such as parallelism and the proportion of areas [17].

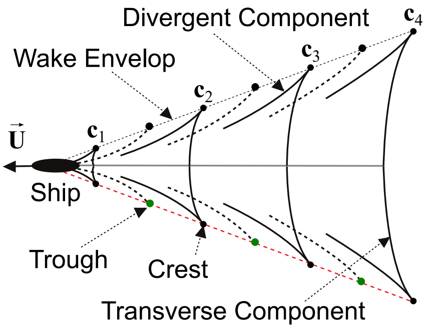


Figure 3. The Kelvin wake structure, indicating the transverse and divergent components as well as the crests and troughs of the wave arms.

For convenience, we used canonical coefficients to define the general equation of the line at infinity, i.e., $\mathbf{l}_\infty = (0,0,1)^T$ in homogeneous coordinates. By taking the vanishing line $\mathbf{l} = (A,B,C)^T$ from (1), the projective transformation $\mathbf{H}$ that mapped $\mathbf{l}$ onto $\mathbf{l}_\infty$ was given by:

$$\mathbf{H} = \mathbf{H}_A\mathbf{H}_\mathbf{l} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ A & B & C \end{pmatrix}. \tag{6}$$

Using $\mathbf{H}$, we were able to rectify each point on the mean water plane in the image. In (6), $\mathbf{H}_\mathbf{l}$ was used to offset the affine matrix $\mathbf{H}_A$, in which a_{11}, a_{12}, a_{21}, and a_{22} defined a 2×2 non-singular matrix and $(a_{13}, a_{23})^T$ was a translation vector.

A planar affine transformation $\mathbf{H}_A$ has six degrees of freedom, which can be computed using three-point correspondences. We used the correspondence between the corners of the rectified ROI using homogeneous coordinates $(0,0,1)^T$, $(W_{ROI}, 0, 1)^T$, and $(0, H_{ROI}, 1)^T$

and points $\mathbf{x}'_k = \mathbf{H}_1\mathbf{x}_k$ for $k \in \{1,2,3\}$, where $\mathbf{x}_k$ are the corners of the ROI in image space (5). From the correspondences, we defined a linear system of the equation in matrix form:

$$\mathcal{A}\,\vec{\mathbf{v}} = \begin{pmatrix} x_{\mathbf{x}'_1} & y_{\mathbf{x}'_1} & w_{\mathbf{x}'_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{\mathbf{x}'_1} & y_{\mathbf{x}'_1} & w_{\mathbf{x}'_1} & 0 \\ x_{\mathbf{x}'_2} & y_{\mathbf{x}'_2} & w_{\mathbf{x}'_2} & 0 & 0 & 0 & -W_{ROI} \\ 0 & 0 & 0 & x_{\mathbf{x}'_2} & y_{\mathbf{x}'_2} & w_{\mathbf{x}'_2} & 0 \\ x_{\mathbf{x}'_3} & y_{\mathbf{x}'_3} & w_{\mathbf{x}'_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{\mathbf{x}'_3} & y_{\mathbf{x}'_3} & w_{\mathbf{x}'_3} & -H_{ROI} \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where $\mathcal{A}$ is a matrix of known values and $\vec{\mathbf{v}}$ is the vector of variables. The coefficients of $\mathbf{H}_A$ were computed as $a_{ij} = v_{ij}/v_{33}$ after solving the system for $\vec{\mathbf{v}}$ as the right-hand null space of $\mathcal{A}$. We used the singular value decomposition $\mathcal{A} = \mathbf{USV}^T$ to solve the system by taking $\vec{\mathbf{v}}$ (i.e., the vector that was associated with the zero singular value) as the last column of matrix $\mathbf{V}$.

By construction, the wave arm that was closest to the camera was always as shown below in the ROI image. Therefore, the rectification only needed to be applied to the points that were obtained in Section 2.3 for that wave arm, since we were interested in the red line that is presented in Figure 3. The black dots in Figure 4 represent the rectified set of points that was used to fit the red curve in Figure 2.

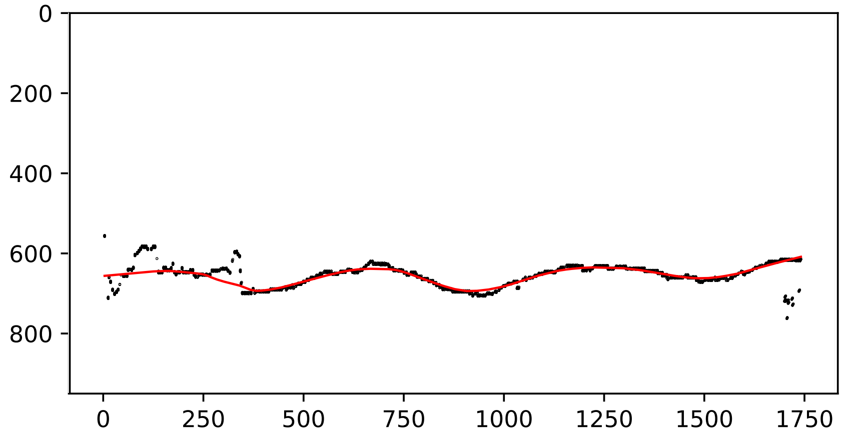


Figure 4. The black points are the curve samples and the red line is the smooth curve that was computed by the LOWESS algorithm [23]. Axes in centimeters.

2.4.2. Curve Fitting

As can be seen in the black points in Figure 4, the discrete set of points that was obtained in Section 2.3 could be corrupted by noise, which made it challenging to find the troughs of the wake. We used the locally weighted scatterplot smoothing (LOWESS) algorithm [23] to smooth the digital curve that was extracted from the rectified ROI image. The black points in Figure 4 correspond to the input curve samples in this example, while the red line is the resulting smooth curve that was used to find the troughs.

Huillca and Fernandes [13] took the point samples for one of the wave arms and followed a naive approach that used a Savitzky–Golay filter [24] for curve fitting. They performed both procedures in the rectified image of the ROI. According to our experience, the use of the k-means algorithm on the input image domain to obtain the point samples followed by the application of the LOWESS algorithm to the points that were mapped onto the domain of the rectified ROI was much less sensitive to noise.

2.4.3. Wavelength Estimation

The wavelength of the transverse components of the Kelvin wake pattern (see Figure 3) can be estimated using the distance between the successive crests or troughs. From those wavelengths, the speed of a vessel can then be estimated [5]. One finds a set of crests/troughs as the curve maximum or minimum, depending on the direction of the vessel with respect to the camera. Since we had a curve that corresponded to the arm that was closest in the V-shaped pattern, we could find the crests and troughs as follows:

- When the vessel went to the right in the input image $\mathcal{I}$, the curve maximum and minimum corresponded to the troughs and crests of the wave arm;
- When the vessel went to the left in the input image $\mathcal{I}$, the curve maximum and minimum corresponded to the crests and troughs of the wave arm.

We avoided the identification of noisy troughs by imposing a minimum horizontal distance of $\psi = 20$ m between valid consecutive troughs and by only extracting two minimums and maximums (depending on the case). The value of ψ was empirically defined based on the observation that the speed is close to 10 knots for a wavelength of approximately 20 m and, as discussed in Section 4, most vessels that leave visible tracks move at higher speeds. We computed the wavelength λ by replacing D in:

$$\lambda = \frac{\sqrt{3}}{2} D, \tag{7}$$

where $D = \mathrm{dist}_E(\mathbf{c}_n, \mathbf{c}_{n+1})$ is the Euclidean distance (in meters) between the location of the crests (or troughs) $\mathbf{c}_n$ and $\mathbf{c}_{n+1}$.

2.4.4. Vessel Speed Estimation

Finally, the speed of the vessel was:

$$U = 1.944 \sqrt{\frac{g}{2\pi} \lambda}, \tag{8}$$

were λ is given by (7) and $g \approx 9.80665\ \mathrm{m/s^2}$ is the acceleration of gravity. Since *knot* is the unit of measurement that is used for speed in maritime navigation, the m/s values had to be multiplied by 1.944 to be converted into *knots*.

3. First-Order Error Propagation

The theory of errors [16] provides the expressions that were needed to estimate the standard uncertainty of a measurement U using the standard uncertainties of the experimental values in the dataset $\vartheta = \{y_{\mathbf{p}_l}, y_{\mathbf{p}_r}, x_{\mathbf{x}_1}, y_{\mathbf{x}_1}, \phi_{\ddot{\mathbf{u}}}, h_g, h_f, h_t, h_s\}$. In matrix form, the first-order error propagation of such uncertainties was expressed by:

$$\sigma_U^2 = \nabla_U \Lambda_\vartheta \nabla_U^T, \tag{9}$$

where

$$\nabla_U = \left(\frac{\partial U}{\partial y_{\mathbf{p}_l}}, \frac{\partial U}{\partial y_{\mathbf{p}_r}}, \frac{\partial U}{\partial x_{\mathbf{x}_1}}, \frac{\partial U}{\partial y_{\mathbf{x}_1}}, \frac{\partial U}{\partial \phi_{\ddot{\mathbf{u}}}}, \frac{\partial U}{\partial h_g}, \frac{\partial U}{\partial h_f}, \frac{\partial U}{\partial h_t}, \frac{\partial U}{\partial h_s} \right) \tag{10}$$

is the Jacobian matrix of the function that calculated the speed U of a vessel using the method that was described in Section 2 and

$$\Lambda_\vartheta = \mathrm{diag}(\sigma_{y_{\mathbf{p}_l}}^2, \sigma_{y_{\mathbf{p}_r}}^2, \sigma_{x_{\mathbf{x}_1}}^2, \sigma_{y_{\mathbf{x}_1}}^2, \sigma_{\phi_{\ddot{\mathbf{u}}}}^2, \sigma_{h_g}^2, \sigma_{h_f}^2, \sigma_{h_t}^2, \sigma_{h_s}^2) \tag{11}$$

is the covariance matrix that encoded the uncertainty of the input variables that were used to compute U. In this paper, we assumed the independence of the input variables. Thus, their covariances were zero, with Λ_ϑ being a diagonal matrix and σ_θ being the standard deviation of the input variable $\theta \in \vartheta$. The partial derivatives in ∇_U (10) were taken with respect to the nine variables in ϑ. Appendix A presents the expressions that were used to compute those partial derivatives.

The computational flow of the U function is illustrated in Figure 5, in which the circles represent the input variables with uncertain values, pentagons represent the input variables that we assumed to have no uncertainty, rectangles represent the intermediary variables, and the rhombus is the estimated speed of the vessel. Altogether, the proposed method has 20 input variables:

$y_{\mathbf{p}_l}$, $y_{\mathbf{p}_r}$: The y axis coordinates of the endpoints $\mathbf{p}_l$ and $\mathbf{p}_r$ of the vanishing line that are estimated for the mean water plane, respectively (Section 2.1);

$x_{\mathbf{x}_1}$, $y_{\mathbf{x}_1}$: The coordinates of the corner $\mathbf{x}_1$ of the ROI in the input image $\mathcal{I}$;

$\phi_{\vec{\mathbf{u}}}$: The angle that defines the direction $\vec{\mathbf{u}}$ of the vessel in the input image $\mathcal{I}$;

h_g, h_f, h_t, h_s: The set of heights that is used in (2) to calculate the camera height h above the sea level in meters;

$x_{\mathbf{c}'_1}$, $y_{\mathbf{c}'_1}$, $w_{\mathbf{c}'_1}$, $x_{\mathbf{c}'_2}$, $y_{\mathbf{c}'_2}$, $w_{\mathbf{c}'_2}$: The homogeneous coordinates of two adjacent troughs of the wave arm that is closest to the camera, i.e., points $\mathbf{c}_1$ and $\mathbf{c}_2$ that were used in (7), but represented by pixel coordinates in $\mathcal{I}$. These variables are not taken as sources of uncertainty because their rectified counterparts ($\mathbf{c}_1 = (x_{\mathbf{c}_1}, y_{\mathbf{c}_1}, 1)^T$ and $\mathbf{c}_2 = (x_{\mathbf{c}_2}, y_{\mathbf{c}_2}, 1)^T$) naturally include the uncertainty that is propagated from other variables (see Appendix A for details);

α_x, α_y, γ, $x_\mathbf{o}$, $y_\mathbf{o}$: The intrinsic parameters that define the camera calibration matrix $\mathbf{K}$. They are the focal length in terms of pixel dimensions in the x and y directions, skew, and the coordinates of the principal point in terms of pixel dimensions, respectively [17]. Recall from Section 2.2 that we extracted metadata from the input image file to compute $\mathbf{K}$. In this work, we assumed that the intrinsic parameters of the camera were constant values since it was observed that they do not usually have much influence on the uncertainty of image-based measurements [25].

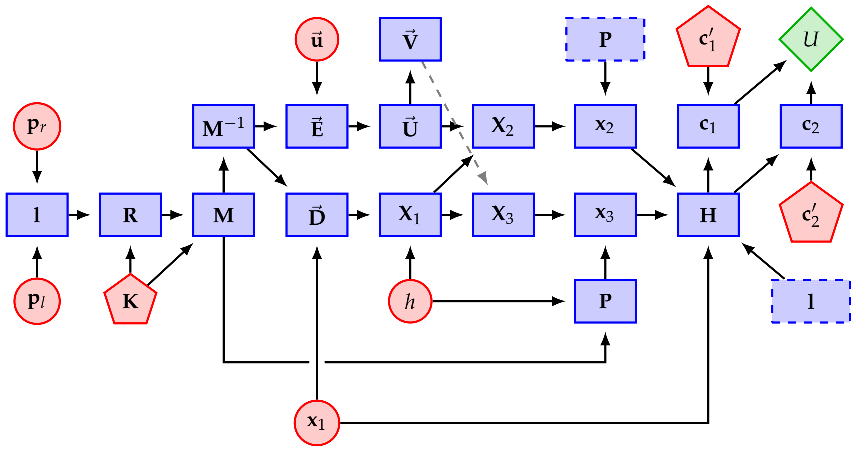


Figure 5. Computational chain for estimating vessel speed (rhombus) using experimental variables with (circles) and without (pentagons) uncertainty.

Section 4 describes how to estimate the uncertainty in $\theta \in \vartheta$.

4. Experiments and Results

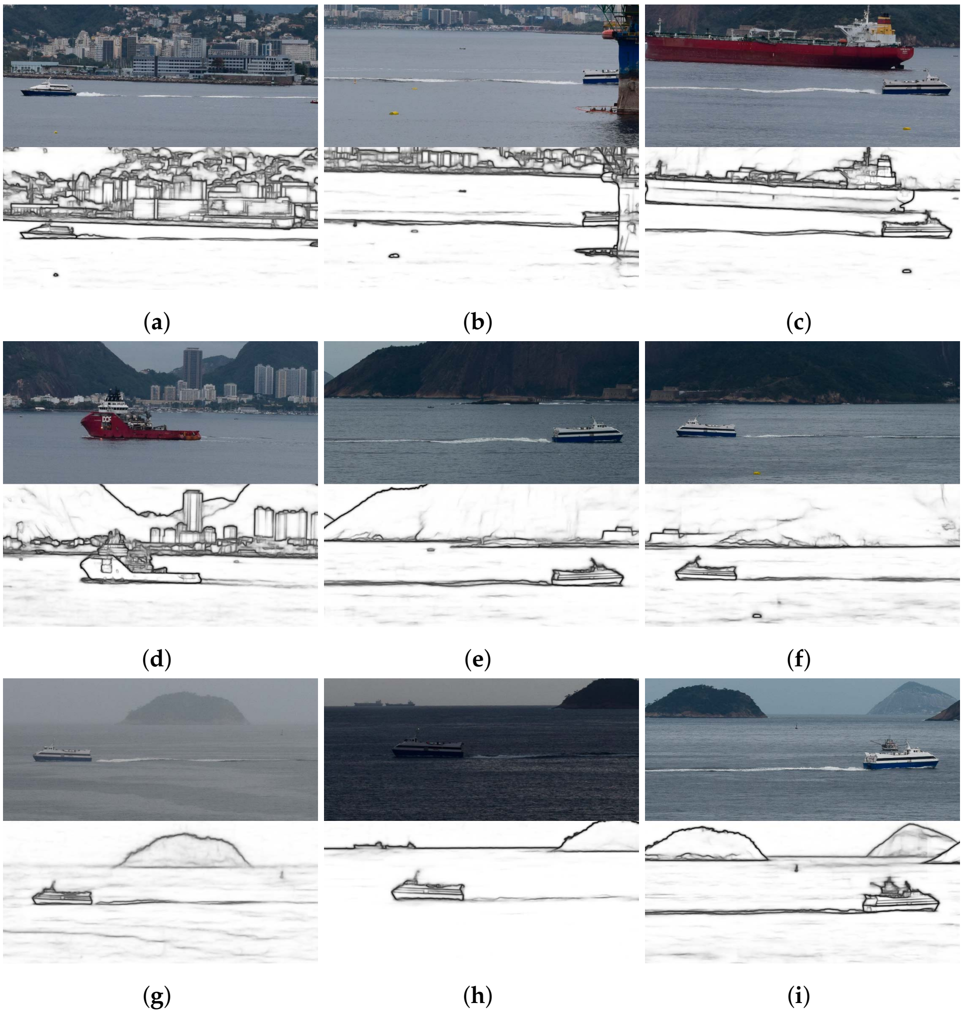
The procedures that were described in Section 2 were implemented in Python 2.7.0. Speeds were calculated on a computer that had an Intel Xeon CPU E-2698 v4 with 2.20 GHz and a Tesla P100-SXM2 video card with 16 GB of VRAM.

In the experiments, the images of moving vessels were acquired under natural lighting and different weather conditions. The images were taken using a Nikon D3300 camera with 24.2 megapixels and were then encoded in JPG format file (any other image format can be used without affecting the results). The lens model that we used was an AF-S DX NIKKOR, with an 18∼55 mm focal length and vibration reduction (VR II) [26]. The resolution of the captured images was 6000 × 4000 pixels. The ROI size lying on plane $\mathbf{\Pi}$ was intentionally set to $W_{ROI} \times H_{ROI} = 180 \times 90$ m in order to cover the wave arms of vessels that were traveling at more than 10 knots, as the speed of the vessels was rarely less than this threshold within the field of view that was used for the experiments. The camera was mounted in two places. The camera height was approximately $h = 27.79$ and $h = 23.88$ m for images i1–i17 and i18–i23, respectively. The camera height varied according to the tide height at the time the image was taken and the floor of the building.

A total of 40 images was obtained, of which 23 were used to analyze the results that are presented in this section. Table 2 describes the conditions under which the images that we used were acquired and Figure 6 shows some of the cropped versions. Special attention was paid to the noisy inputs that we included in our experiments, such as images i18 to i23 (see Figure 6g,h), because those cases included natural noise as they were acquired during a scattered storm or the partially cloudy and windy weather that followed the storm. The remaining 17 of the original 40 images were not considered because they were taken during unfavorable weather conditions (Figure 7a) and low natural lighting (Figure 7b), which prevented the edge detection approach from being successful in the detection of the troughs. Additionally, some of the vessels in those images were merchant ships (Figure 7c). As such, their speed had to be low because they were close to a port area. In all of those cases, at least two troughs of the traces that were left by the vessels could not be distinguished, even by human observers. In Table 2, the tide height was obtained from webpages that freely provide sea conditions [27–29]. The noise that was introduced by the wind speed was not considered in these experiments.

Table 2. Information regarding each captured image, where U denotes the speed that was measured by a radar (ground truth), $\hat{U}$ is the speed that was estimated by our approach using the troughs of the wave arms, and ε_a and ε_r are the absolute and relative errors of estimations, respectively. The subtable summarizes the results of Huillca and Fernandes [13].

Image	**Vessel**		**Time (hh:mm)**	**Weather**	**Tide (Meters)**	**Speed (*knots*)**		**Error**		**Results of [13]**	
	Model	**Name**				U	$\hat{U}$	ε_a	ε_r	$\hat{U}$	ε_r
i1	*HSC*	Fenix	10:09	Cloudy	0.30	18.2	18.339	0.139	0.008	17.234	0.053
i2	*MC25*	Apolo	10:12	Cloudy	0.30	20.6	21.533	0.933	0.045	25.467	0.236
i3	*MC25*	Apolo	10:13	Cloudy	0.30	20.5	20.609	0.109	0.005	26.629	0.299
i4	*MC25*	Neptuno	10:19	Cloudy	0.50	19.2	19.434	0.234	0.012	21.788	0.135
i5	*MC25*	Neptuno	10:21	Cloudy	0.50	17.3	16.683	0.617	0.036	20.346	0.176
i6	*MC25*	Neptuno	10:22	Cloudy	0.50	17.1	16.530	0.570	0.033	27.785	0.625
i7	*Other*	Escander Amazonas	10:28	Cloudy	0.60	9.2	19.262	10.062	1.094	17.326	0.883
i8	*MC25*	*Missing*	10:39	Cloudy	0.70	19.5	19.288	0.212	0.011	26.036	0.335
i9	*HSC*	Fenix	10:41	Cloudy	0.70	15.6	15.984	0.384	0.025	18.231	0.169
i10	*HSC*	Fenix	10:43	Cloudy	0.70	16.5	16.611	0.111	0.007	20.133	0.220
i11	*MC25*	*Missing*	11:08	Cloudy	0.50	20.4	20.771	0.371	0.018	26.436	0.296
i12	*MC25*	*Missing*	11:09	Cloudy	0.50	20.4	20.543	0.143	0.007	23.863	0.170
i13	*MC25*	*Missing*	11:12	Cloudy	0.50	17.3	16.846	0.454	0.026	23.259	0.344
i14	*MC25*	Zeus	11:42	Cloudy	0.70	19.6	20.533	0.933	0.048	26.982	0.377
i15	*MC25*	Neptuno	11:43	Cloudy	0.70	19.1	19.861	0.761	0.040	19.407	0.016
i16	*MC25*	Neptuno	12:10	Cloudy	0.90	20.2	22.120	1.920	0.095	25.951	0.285
i17	*MC25*	*Missing*	12:13	Cloudy	0.90	20.3	20.638	0.338	0.017	22.254	0.096
i18	*MC25*	Zeus	16:12	Scattered storm	1.10	18.9	19.617	0.717	0.038	22.086	0.169
i19	*MC25*	Zeus	16:50	Partly cloudy	0.70	17.5	17.342	0.158	0.009	22.249	0.271
i20	*MC25*	Zeus	16:51	Partly cloudy	0.70	17.6	17.808	0.208	0.012	19.947	0.133
i21	*MC25*	Zeus	16:51	Partly cloudy	0.70	17.8	18.947	1.147	0.064	25.107	0.410
i22	*MC25*	*Missing*	17:01	Partly cloudy	0.70	20.3	16.490	3.810	0.188	23.448	0.155
i23	*MC25*	Zeus	17:21	Partly cloudy	0.50	16.8	16.084	0.716	0.043	21.008	0.251



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 6. Cropped versions of some of the images that were used in the experiments: images i1, i9, and i10 show HSC passenger vessels; images i10, i13, i18, and i22 show MC25 passenger vessels; image i7 shows another vessel model. (**a**) Image i1; (**b**) Image i2; (**c**) Image i5; (**d**) Image i7; (**e**) Image i13; (**f**) Image i16; (**g**) Image i18; (**h**) Image i22; and (**i**) Image i17.

A radar that was designed to monitor vessels was used as a resource to validate the proposed method. The radar was a $FAR - 21 \times 7$ series of X- and S-bands with a 19-inch LCD screen [30]. The radar screen information was captured using a smartphone camera. The radar screen included the name and *knot* speed of the tracked vessel. The images that were taken of the radar screen and the moving vessel were acquired at approximately the same time. Table 2 shows the identification of the vessels (columns "Model" and "Name"), the time at which both pictures were taken (column "Time"), and the speed that was measured by the radar (column "U"). Of the 23 images that were used, 22 were of six different passenger vessels (models HSC and MC25) and one was a tugboat (image i7, Figure 6d). As we had limited access to the radar (upon authorization), it was necessary to restrict the image sections and only take 40 images. Additionally, later access was not allowed due to the COVID-19 pandemic. Even so, the set of images that was captured led to exciting results that prove the technique to be promising.

Considering that the data that were used as the input (e.g., the y coordinate of the endpoints of the vanishing line, the camera height, the reference corner of the ROI, and direction of the vessel in image space) were subject to errors, it was expected that the estimated speeds would also have uncertainties. By comparing the computed values to the speeds that were measured by the radar, it was possible to develop an idea of the accuracy and precision of the proposed technique. In Section 4.1, we analyze the relative error of our estimations and compare the quality of the estimates that were made using our technique to the approach that was presented by Huillca and Fernandes [13]. Sections 4.2 and 4.3 present an analysis of the confidence intervals that were computed using sampling and first-order error propagation. In Sections 4.4–4.6, we discuss the influence of each experimental variable on the uncertainty of the estimated speed, the resilience to changes in the resolution of the input image, and the variations in JPG compression rate, respectively.

For Sections 4.2–4.4, the input uncertainties were estimated as follows:

$\sigma_{y_{\mathbf{p}_l}}$, $\sigma_{y_{\mathbf{p}_r}}$: The standard deviations of $y_{\mathbf{p}_l}$ and $y_{\mathbf{p}_r}$ were estimated using:

$$\sigma_z = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(z_i - \mu_z)^2}, \tag{12}$$

where μ_z is the mean value of the differences z_i between the y coordinates that were observed on endpoints that were returned by HLW and the endpoints of vanishing lines that were manually identified by us in a dataset comprising $n = 30$ images, which led to $\sigma_{y_{\mathbf{p}_l}} = 22.72$ and $\sigma_{y_{\mathbf{p}_r}} = 23.62$ pixels;

$\sigma_{x_{\mathbf{x}_1}}$, $\sigma_{y_{\mathbf{x}_1}}$: We also used the $n = 30$ experimental samples to set $\sigma_{x_{\mathbf{x}_1}} = 5.51$ and $\sigma_{y_{\mathbf{x}_1}} = 2.41$ pixels. The samples were obtained by repeatedly selecting the first corner of the ROI in the chosen image to serve as a reference. The standard deviations for the $x_{\mathbf{x}_1}$ and $y_{\mathbf{x}_1}$ coordinates were calculated using (12) for the coordinates of the selected points;

$\sigma_{\phi_{\ddot{\mathbf{u}}}}$: The same reference image was used to indicate the orientation of a vessel, which produced $n = 30$ angular samples that were used to compute $\sigma_{\phi_{\ddot{\mathbf{u}}}} = 0.011$ radians;

σ_{h_g}, σ_{h_f}, σ_{h_t}, σ_{h_s}: The standard deviations of the tripod height and the floor height were empirically set to $\sigma_{h_t} = 0.006$ and $\sigma_{h_f} = 0.03$ m, respectively, by assuming a conservative uncertainty for the measurement of those input variables. We used Google Maps to measure the ground height and set its uncertainty to $\sigma_{h_g} = 1.15$ m based on the variations we observed in this tool. We estimated $\sigma_{h_s} = 0.05$ m by applying (12) to a set of $n = 27$ average tidal heights z_i that were computed from observations in [27–29].

We assumed that the locations $\mathbf{c}_1$ and $\mathbf{c}_2$ of the troughs that were detected in the rectified ROI images carried the uncertainties that were introduced by the input variables, but the detection process itself did not introduce any new uncertainties (see Figure 5 and Appendix A).

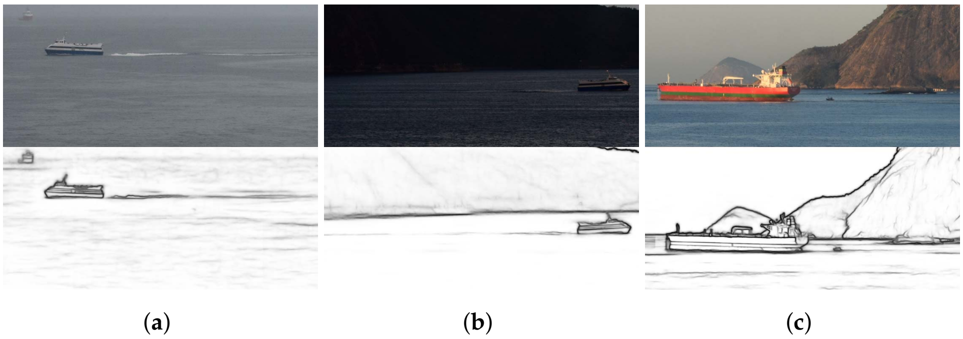


Figure 7. Cropped versions of some of the images in which our approach could not detect wave arms that had at least two well-defined troughs in the edge image.

4.1. Analysis of Relative Error

Relative error $\varepsilon_r = \varepsilon_a / U$ indicates the proportion of the absolute error $\varepsilon_a = |\hat{U} - U|$ of an estimated value $\hat{U}$ with regard to the true value U. We used ε_r to determine the accuracy of our approach. In Table 2, the absolute error is given in *knots*. $\hat{U}$ was calculated by applying the proposed method, while U was measured by the radar.

We used the troughs of the closest wave arms to estimate the $\hat{U}$ values that are presented in Table 2. Those troughs were the least affected by the noise that was introduced by the vessel's turbulence and the distortion of the rectified elements that were not in the actual mean water plane. According to Table 2, the relative error was below 2% for ten images, within the $[2\%, 4\%]$ interval in six cases, within the $(4\%, 5\%]$ range in three images, and between 5% and 10% in three cases. Only image i7 (Figure 6d) had a relative error that surpassed the true measure (109%). The explanation for this behavior is that the trace that was left by the tugboat was weak because it was traveling at 9.2 *knots* and we set the ψ parameter for the minimum horizontal distance between valid consecutive troughs as 20 m, which limited the estimated speeds to a minimum of 10 *knots*. The mean and median relative errors that are presented in Table 2 were 8.18% and 2.60%, respectively. The mean relative error was clearly affected by the result in image i7. Observing the robust statistics that were provided by the median, we could conclude that the proposed method was accurate.

To the best of our knowledge, the only work that has presented an approach for the estimation of the speed of moving vessels using single color images was developed by Huillca and Fernandes [13]. The subtable in Table 2 summarizes the results that were obtained by their approach after we replaced their RANSAC-based vanishing line detection scheme with the HLW algorithm. Otherwise, their method would not have been applicable to most of the images in Table 2. Unfortunately, it was not possible to perform a comparison between our approach and techniques that use image sequences, such as [10,11], because their implementation was not available and the descriptions that are presented in the articles proved to be insufficient for proper reproduction. In any case, such techniques cannot be used for video from cameras that are onboard vessels, which limits the scope of their application.

Observing ε_r in Table 2, our new technique outperformed the previous approach for 20 out of the 23 images. For two cases in which our relative errors were higher, the difference in the errors was only 2.4% on image i15 and 3.3% on image i22. The third case was image i7, which, as previously mentioned, was not correctly handled by our approach because of the ψ parameter that was set.

4.2. Analysis of Confidence Intervals Estimated Using the Samples

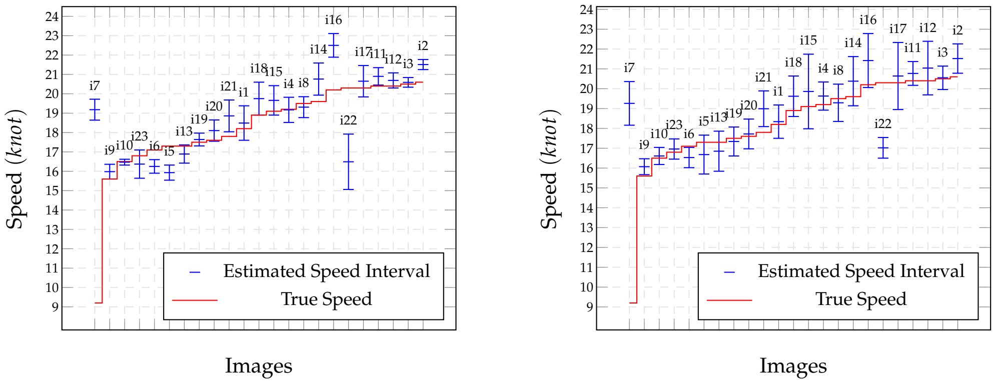
In practice, each image that is presented in Table 2 provided one sample for which we could estimate speed. One way to assess the accuracy of the technique was to analyze the variations in the speed estimates that were obtained for the same vessel when considering a set of images that were captured under similar conditions as the input. To simulate several image captures of the same ship, we introduced small variations in the input variables that were considered to be sources of uncertainty for each captured image. We used the standard deviations that were assumed in the error propagation model (11) to produce $n = 150$ Gaussian-distributed variations in the original set of input values for each image in Table 2, thereby generating samples from which we could compute speeds and the corresponding confidence intervals:

$$CI(\gamma) = \left[\bar{U} - t_\gamma \frac{s_U}{\sqrt{n}}, \bar{U} + t_\gamma \frac{s_U}{\sqrt{n}}\right], \tag{13}$$

where $\bar{U}$ is the mean speed of the sample, s_U is the standard deviation of the sample, t_γ is a t-Student variable with $n - 1$ degrees of freedom, and γ is the confidence level.

Figure 8a shows the confidence intervals with $\gamma = 99.8\%$ that were calculated for the vessels using sampling (Table 2). The narrowest confidence intervals were for images i10 and i3, which were 0.30 and 0.50 *knots* wide, respectively. Notice that the mean speed $\bar{U}$ was close to the true speed U in most cases and was included within the confidence interval in 16 out of the 23 images. Images i7, i16, and i22 were the cases with the most considerable distances between U and $\bar{U}$, whose confidence intervals did not include the true speed. The problem with image i7 was discussed in Section 4.1. For image i16 (Figure 6f), the location of the troughs was affected by the weather conditions. Notice the presence of more capillary wakes that were due to wind in this image than in the other images in Figure 6. For image i22 (Figure 6h), low natural lighting made the trail of the vessel very blurred.

For the remaining four cases in which the true speed was outside of the confidence interval (images i6, i11, i14, and i21), the distance to the limits of the interval was negligible and ranged from 0.05 to 0.50 *knots*.



(**a**) (**b**)

Figure 8. Confidence intervals ($\gamma = 99.8\%$) that were computed using (**a**) sampling and (**b**) first-order error propagation. Images sorted by vessel speed.

Including all cases that are presented in Figure 8a, the largest confidence intervals were 2.86 and 1.78 *knots* for images i1 and i22, respectively. The median interval was only 0.3 *knots*. The variations in the values that were reported by the radar for the three consecutive speed measurements of vessels Zeus (images i19 to i21), Neptune (images i4 to i6), and *Missing* (images i11 to i13) were 0.3, 2.1, and 3.1 *knots*, respectively. Thus, we could conclude that the proposed approach was accurate. However, we cannot make a strong statement in this regard because each interval was calculated using samples that were generated from one image.

4.3. *Analysis of Confidence Intervals Estimated Using Error Propagation*

In this section, we analyze the confidence intervals that were produced by the first-order error propagation approach, as discussed in Section 3, and compare them to the intervals that were produced using sampling.

First-order error propagation may provide the correct Gaussian uncertainty for the resulting estimations when the uncertainty of the input variables follows Gaussian distribution and the process for computing the resulting values is linear. Otherwise, it provides a first-order approximation of the error [16]. To verify which is the case for our approach, we used the Shapiro and Wilk [31] test to check whether the resulting samples that were produced in Section 4.2 fit Gaussian distribution. The null hypothesis of this test was: the data are normally distributed when $\rho > \alpha$. For $\alpha = 0.05$, images i7, i16, i19, i22, and i23 had ρ-values that suggested evidence of non-normality. Therefore, we could only expect an approximation from the first-order error propagation of these cases.

The ratio $r = s_U / \sigma_{\hat{U}}$ between the standard deviations that were computed using sampling and propagation showed that the first-order error propagation approach was equivalent to and slightly more conservative than the sampling-based approach. The only exceptions were $r > 1$ for images i1, i22, and i23. In 16 cases, $0.4 \leq r \leq 1$. This was reflected in the results that are presented in Figure 8b, in which only two confidence intervals clearly did not include the true speeds (images i7 and i22) and four almost included them (images i2, i6, i9, and i21). Notice that images i7, i22, and i23 did not pass the normality test.

The narrowest intervals in Figure 8b were 0.80 and 0.86 *knots* in width, while the widest were 3.38 *knots* (image i17) and 3.76 knots (image i15). Using first-order error propagation, the mean width of the confidence intervals was 0.942 *knots* and the median was 0.46 *knots*.

With error propagation, it was easy to detect cases that had more considerable uncertainty for the calculated speeds because this approach does not require several samples.

4.4. Impact of the Uncertainty of Each Input Variable

The impact of each input variable on the uncertainty of the estimated speed could be assessed using the error propagation model (9). The absolute contribution of any input variable $\theta \in \vartheta$ was obtained using a covariance matrix Λ_{ϑ} (11), for which the only non-zero elements were those related to θ in the main diagonal. The relative impact of θ was obtained by dividing its absolute impact by the sum of the absolute impacts of all input variables.

For this analysis, we grouped the input variables into four groups (see the columns of the heatmap tables in Figure 9): the first group $\mathbf{l}$ included $y_{\mathbf{p}_l}$ and $y_{\mathbf{p}_r}$; the second group represented $\mathbf{x}_1$ and included $x_{\mathbf{x}_1}$ and $y_{\mathbf{x}_1}$; the third group was $\vec{\mathbf{u}}$ and only included the angle $\phi_{\vec{\mathbf{u}}}$; and the last group included the variables h_g, h_f, h_t, and h_s, which were used to compute the camera height h.

Figure 9 (left) shows the relative impact of each group of input variables, assuming that they all had the same uncertainty ($\sigma_\theta = 1$ for all $\theta \in \vartheta$) and were independent. Taking this as a premise, the largest sources of uncertainty for the estimated speeds were the parameters of $\mathbf{x}_1$ and $\vec{\mathbf{u}}$. One possible explanation is that $\mathbf{x}_1$ and $\vec{\mathbf{u}}$ were the variables that were used to calculate the first point $\mathbf{X}_1$ in the world coordinate system and $\mathbf{X}_1$ was used to calculate the remaining corners. Furthermore, the ROI coordinates in image space came from the world coordinate system and they directly fed the uncertainties into the matrix $\mathbf{H}$.

Figure 9 (right) shows the scenario illustrating the results that were obtained by the estimated uncertainties of the input variables for the proposed method. It can be seen that the most significant source of uncertainty was the camera height h, which demonstrates the importance of this parameter in breaking projective ambiguity and, consequently, the correct scale of the estimates that are produced. In the particular case of our experiments, it was necessary to assume $\sigma_{h_g} = 1.15$ since we observed inconsistent height readings at close points on the map.

	$\mathbf{l}$	$\mathbf{x}_1$	$\vec{\mathbf{u}}$	h
i1	0.045	0.090	0.805	0.061
i2	0.014	0.026	0.947	0.013
i3	0.172	0.334	0.286	0.208
i4	0.205	0.402	0.209	0.184
i5	0.233	0.465	0.011	0.291
i6	0.193	0.384	0.000	0.423
i7	0.326	0.515	0.001	0.157
i8	0.280	0.529	0.001	0.190
i9	0.004	0.007	0.985	0.004
i10	0.170	0.332	0.112	0.385
i11	0.223	0.426	0.069	0.282
i12	0.249	0.475	0.123	0.153
i13	0.347	0.431	0.009	0.213
i14	0.100	0.196	0.642	0.062
i15	0.029	0.058	0.893	0.020
i16	0.186	0.345	0.330	0.138
i17	0.123	0.239	0.545	0.093
i18	0.008	0.016	0.973	0.004
i19	0.256	0.437	0.028	0.278
i20	0.094	0.162	0.665	0.078
i21	0.063	0.121	0.768	0.048
i22	0.093	0.516	0.018	0.372
i23	0.249	0.415	0.005	0.331

Input with the same uncertainty

	$\mathbf{l}$	$\mathbf{x}_1$	$\vec{\mathbf{u}}$	h
i1	0.003	0.017	0.000	0.980
i2	0.004	0.023	0.000	0.973
i3	0.003	0.019	0.000	0.978
i4	0.004	0.025	0.000	0.971
i5	0.003	0.019	0.000	0.978
i6	0.002	0.011	0.000	0.987
i7	0.009	0.038	0.000	0.953
i8	0.006	0.032	0.000	0.962
i9	0.004	0.018	0.000	0.979
i10	0.002	0.010	0.000	0.988
i11	0.003	0.018	0.000	0.979
i12	0.004	0.025	0.000	0.971
i13	0.005	0.016	0.000	0.979
i14	0.005	0.025	0.000	0.970
i15	0.004	0.024	0.000	0.972
i16	0.004	0.020	0.000	0.976
i17	0.004	0.020	0.000	0.976
i18	0.005	0.031	0.000	0.964
i19	0.003	0.013	0.000	0.984
i20	0.004	0.017	0.000	0.980
i21	0.004	0.020	0.000	0.976
i22	0.003	0.049	0.000	0.949
i23	0.002	0.010	0.000	0.988

Input with experimental uncertainty

Figure 9. The relative impact of the input parameters on the computed speeds, assuming that (**left**) all input variables were independent and had the same uncertainty or (**right**) had the uncertainty that was estimated for the experiments: $\mathbf{l}$ represents the vanishing line; $\mathbf{x}_1$ is the reference corner of the ROI; $\vec{\mathbf{u}}$ is the direction of the vessel; and h is the camera height. Stronger shades of blue indicate greater relative impact.

4.5. Resilience to Variations in Resolution

In this experiment, the tests were carried out with scaled versions of the original image. This experiment simulated the use of lower resolution cameras and the speed estimation of more distant vessels. Here, we discuss the results that assumed scaling factors of 0.5, 0.25, and 0.125. In each case, the camera calibration matrix $\mathbf{K}$ and the location of the reference corner $\mathbf{x}_1$ of the ROI were transformed accordingly. Furthermore, the corresponding edge image and the new y coordinates for the endpoints of the vanishing line $\mathbf{l}$ were obtained by applying the RCF and HLW algorithms to the scaled versions of the input images. The camera height h and the direction of the vessel $\vec{\mathbf{u}}$ did not change.

Table 3 summarizes the speed $\hat{U}_s$ and relative error ε_{r_s} that were computed for each input image, in which $s \in \{1.0, 0.50, 0.25\}$ denotes the scaling factor, $s = 1.0$ is the original scale, and U is the true speed value that was measured by the radar.

Table 3. Variation in the estimated speeds as a function of image resolution: U is the speed that was measured by the radar; $\hat{U}_s$ and ε_{r_s} are the speed that was estimated by our approach and the relative error that was obtained using an input image at a scale of s, respectively; and $s = 1.0$ for the original images.

Image	Speed $(knots)$				Relative Error		
	U	$\hat{U}_{1.00}$	$\hat{U}_{0.50}$	$\hat{U}_{0.25}$	$\varepsilon_{r_{1.00}}$	$\varepsilon_{r_{0.50}}$	$\varepsilon_{r_{0.25}}$
i1	18.2	18.34	18.75	–	0.01	0.03	–
i2	20.6	21.53	22.39	21.82	0.05	0.09	0.06
i3	20.5	20.61	23.53	13.64	0.01	0.15	0.33
i4	19.2	19.43	20.71	19.36	0.01	0.08	0.01
i5	17.3	16.68	19.10	13.96	0.04	0.10	0.19
i6	17.1	16.53	17.93	17.86	0.03	0.05	0.04
i7	9.2	19.26	–	–	1.09	–	–
i8	19.5	19.29	19.30	21.75	0.01	0.01	0.12
i9	15.6	15.98	19.38	–	0.02	0.24	–
i10	16.5	16.61	16.15	15.38	0.01	0.02	0.07
i11	20.4	20.77	22.24	6.40	0.02	0.09	0.69
i12	20.4	20.54	13.32	10.45	0.01	0.35	0.49
i13	17.3	16.85	18.78	–	0.03	0.09	–
i14	19.6	20.53	19.50	–	0.05	0.01	–
i15	19.1	19.86	11.08	–	0.04	0.42	–
i16	20.2	22.12	20.09	–	0.10	0.01	–
i17	20.3	20.64	18.95	–	0.02	0.07	–
i18	18.9	19.62	22.07	14.48	0.04	0.17	0.23
i19	17.5	17.34	21.46	18.58	0.01	0.23	0.06
i20	17.6	17.81	19.24	10.54	0.01	0.09	0.40
i21	17.8	18.95	–	–	0.06	–	–
i22	20.3	16.49	10.22	14.99	0.19	0.50	0.26
i23	16.8	16.08	17.09	–	0.04	0.02	–

Recall that the resolution of the original images ($s = 1.0$) was 6000 × 4000 pixels. However, it is important to notice that, except for the detection of the vanishing line, all visual information that was used by our approach was within the ROI in image space. The average resolution of the axis-aligned bounding box of the ROI at $s = 1.0$ was only 151 × 1647 pixels. For the 0.5, 0.25, and 0.125 scaled images, the average resolutions of the ROI were 73 × 807, 36 × 406, and 17 × 202 pixels, respectively.

According to Table 3, most of the speeds that were calculated with a scale of 0.5 were close to the true speed, the relative errors were below 10% for 13 images, and there were eight cases between 10% and 50%. Image i16 was the least affected by the change in resolution, which presented a degradation of only 10% of the relative error compared to the original image. It was only not possible to estimate speeds for images i7 and i21. Among those that resulted in measurements, the worst case was image i12 with a 3500%

degradation in the relative error. The mean and median degradations were 681.66% and 300%, respectively.

In total, it was not possible to estimate speeds using 43.5% of the images with a scale of 0.25. The relative errors for the remaining cases were below 13% for seven images and there were six cases between 18% and 70%. The mean relative error for images with a scale of 0.25 was 22.74%, while the mean and median degradations of the relative error concerning the original images were 1514.63% and 600%, respectively.

These results showed that the resolution of the ROI and, hence, the amount of information that was available to produce a quality edge image was critical to the performance of the proposed approach, since the edge image was used to provide the visual clues for the identification of the wave arms. In the practical use of this technique, automatic zooming could be used to increase the resolution of the ROI.

4.6. Resilience to Variations in JPG Compression Rate

In the last experiment, we analyzed the variation in estimated speeds as we changed the JPG image compression level. For each original image that was stored at 100% quality, we created copies at 90%, 75%, and 50% quality and applied the approach that was described in Section 2.

In Table 4 it is possible to observe that the estimated speed $\hat{U}_{100\%}$ did not change much compared to the lower resolution counterparts of the same image. Surprisingly, in images i1, i7, i14, i15, i16, i21, and i22, there was a decrease of 1 to 4% in the relative error of the estimates for images with 90%, 75%, or 50% quality. The biggest increase in relative error was 6% for image i18 at 75% quality. As mentioned before, image i18 (Figure 6g) was one of the noisy images that was captured under a scattered storm. In another 20 cases (images i3, i4, i5, i7, i13, i15, i17, i18, i20, and i22), the increase ranged from 1 to 5%. In 32 out of the 69 cases, the increase or decrease in the relative error was less than 1%.

Table 4. Variation in estimated speeds as a function of image compression: U is the speed that was measured by the radar; $\hat{U}_q$ and ε_{r_q} are the speed that was estimated by our approach and the relative error of using an image at quality compression q, respectively; and $q = 100\%$ for the original images.

Image	**Speed (*knots*)**					**Relative Error**			
	U	$\hat{U}_{100\%}$	$\hat{U}_{90\%}$	$\hat{U}_{75\%}$	$\hat{U}_{50\%}$	$\varepsilon_{q_{100\%}}$	$\varepsilon_{q_{90\%}}$	$\varepsilon_{q_{75\%}}$	$\varepsilon_{q_{50\%}}$
i1	18.2	18.34	18.21	18.21	18.21	0.0077	0.0005	0.0005	0.0005
i2	20.6	21.52	21.48	21.48	21.48	0.0447	0.0427	0.0427	0.0427
i3	20.5	20.55	20.52	20.42	20.00	0.0024	0.0010	0.0039	0.0244
i4	19.2	19.63	19.93	19.93	20.27	0.0224	0.0380	0.0380	0.0557
i5	17.3	16.68	16.34	16.34	16.34	0.0358	0.0555	0.0555	0.0555
i6	17.1	16.53	16.54	16.54	16.54	0.0333	0.0327	0.0327	0.0327
i7	9.2	19.26	19.32	19.32	19.12	1.0935	1.1000	1.1000	1.0783
i8	19.5	19.29	19.28	19.28	19.28	0.0108	0.0113	0.0113	0.0113
i9	15.6	16.07	16.09	16.09	16.09	0.0301	0.0314	0.0314	0.0314
i10	16.5	16.61	16.55	16.55	16.55	0.0067	0.0030	0.0030	0.0030
i11	20.4	20.77	20.83	20.83	20.83	0.0181	0.0211	0.0211	0.0211
i12	20.4	21.04	21.10	21.10	21.03	0.0314	0.0343	0.0343	0.0309
i13	17.3	16.85	16.82	16.00	15.99	0.0260	0.0277	0.0751	0.0757
i14	19.6	20.38	20.17	20.21	20.21	0.0398	0.0291	0.0311	0.0311
i15	19.1	19.86	19.70	19.73	19.70	0.0398	0.0314	0.0330	0.0314
i16	20.2	21.42	21.22	21.76	21.22	0.0604	0.0505	0.0772	0.0505
i17	20.3	20.64	20.62	20.54	20.82	0.0167	0.0158	0.0118	0.0256
i18	18.9	19.62	20.29	20.80	20.61	0.0381	0.0735	0.1005	0.0905
i19	17.5	17.34	17.40	17.40	17.40	0.0091	0.0057	0.0057	0.0057
i20	17.6	17.72	16.95	17.87	17.12	0.0068	0.0369	0.0153	0.0273
i21	17.8	18.99	18.36	18.36	18.36	0.0669	0.0315	0.0315	0.0315
i22	20.3	17.02	17.25	17.25	6.80	0.1616	0.1502	0.1502	0.6650
i23	16.8	16.96	16.99	16.99	16.99	0.0095	0.0113	0.0113	0.0113

The results that were presented in this section suggested that the proposed approach had a good resilience to the compression of the input image. We believe that the reason for this is the robustness of the edge detection technique that was used in our implementation.

5. Conclusions and Future Works

We presented a method for the estimation of vessel speed using single perspective projection images. The approach uses geometric constraints to remove perspective distortion from the images of traces that were left by a moving vessel and uses curve fitting and peak detection to identify troughs in the wave arms and natural constraints in components of Kelvin wakes in order to compute vessel speed.

We validated the measurements that were produced by our approach using the speeds that were obtained by a radar. The quality of the results was verified by the application of a statistical analysis on the estimated speeds. We also used error propagation along the computational chain to provide reliable confidence intervals that provided a notion of the quality of the speeds that were estimated from a single image and presented a study that could identify the set of input parameters that had more impact on the uncertainty of the estimated speeds. The statistical analysis revealed that the estimated speeds were accurate and precise. We believe that our algorithm could be used by autonomous vessels and for maritime surveillance using drones and smart lighthouses.

In order to consider the use of our technique in real situations, it is necessary to draw some recommendations:

1. Lighting conditions affect edge detection and the detection of wave arms. In our experiments, we had no problems in daylight, but it was not always possible to process images that were captured at dawn or dusk and our solution cannot be applied at night. The same applies to rain and fog;
2. Due to geographical restrictions in our experiments, we used images of the port and starboard of vessels that were traveling in the left and right directions in front of the camera and moving along a linear course at a (supposedly) constant speed. However, we believe that our approach is robust to variations in camera orientation since it was possible to see the troughs in the wake, even at the grazing angle;
3. As demonstrated in our experiments, well-defined capillary wakes due to wind and, possibly, those generated by nearby vessels may affect the Kelvin wake pattern. However, we believe that this is a problem that could be overcome by the detection of crossing wakes;
4. Since this method is to be applied to single images, the use of video could provide dozens of independent measurements per second, which could be combined to reduce error or eliminate spurious estimates;
5. Although we did not try this in our experiments, pre-processing the images to increase contrast could help in the detection of the wakes of slower vessels.

Unfortunately, the radar that we used could not automatically display information on the speed of small vessels, e.g., sailboats and fishing boats. This is because small vessels do not have to be equipped with an automatic identification system (AIS), which allows sensors to display the ship's speed information. In addition, the social isolation that was imposed by the COVID-19 pandemic prevented us from having access to the radar to expand data acquisition. Even so, we were able to obtain speed information from several medium-sized vessels. The results showed that our approach was robust. It was validated by considering the measurements from standard nautical equipment and we believe that it could be applied to any vessel that leaves distinguishable wake patterns. As a direction for future work, we point to the investigation of the influence of climatic conditions to better analyze the few cases in which the results were not consistent. For this analysis, images must be systematically captured at different times of day, under various lighting conditions, in different weather conditions, at different wind speeds, and in all four seasons of the year. So, ideally, systematic captures need to take place over at least one year in order to obtain a wide range of image conditions.

Another direction for future work is to extend the analyses that were presented in this paper through the application of the ISO Guide to the Expression of Uncertainty of Measurement (GUM) [32].

The testbed implementation of our approach took approximately 12 s to process each image. Most of that time was used to compute the edge image using the RCF algorithm. Faster methods for calculating the edge images proved ineffective in enhancing the ship wakes. So, we aim to optimize this stage to calculate the vessel speed in real time. We are also exploring ways to work with super-resolution images in order to improve the quality of the information being included in the ROI and ways to use a sequence of images to calculate the vessel speed in each image and analyze the resulting speeds, although the central idea of this work was to use only one image. Our implementation will be made available after the publication of this paper.

Author Contributions: Conceptualization, J.L.H. and L.A.F.F.; methodology, J.L.H. and L.A.F.F.; software, J.L.H.; validation, J.L.H.; formal analysis, J.L.H. and L.A.F.F.; investigation, J.L.H. and L.A.F.F.; resources, J.L.H. and L.A.F.F.; data curation, J.L.H. and L.A.F.F.; writing—original draft preparation, J.L.H. and L.A.F.F.; writing—review and editing, J.L.H. and L.A.F.F.; visualization, J.L.H. and L.A.F.F.; supervision, L.A.F.F.; funding acquisition, L.A.F.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the CNPq (311.037/2017-8) and FAPERJ (E-26/202.718/2018) agencies. Jose Luis Huillca was sponsored by a CAPES fellowship.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Arthur Ayres Neto (LAGEMAR-UFF), Fabio Hochleitner (LAMCE-COPPE-UFRJ), Artur Brandão, and Jacó Julio de Souza Costa for their assistance in building the dataset.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Partial Derivatives for Error Propagation

Section 3 shows that we used first-order error propagation to compute the standard uncertainty of the estimated vessel speed (σ_U) using the uncertainties of the experimental input variables in the dataset ϑ as the sandwich product of the Jacobian matrix ∇_U and the covariance matrix Λ_ϑ (9). This appendix includes the expressions that were used to compute the partial derivatives in ∇_U using the chain rule. Here, we present the partial derivatives for the variables of the computational chain that is illustrated in Figure 5.

Vanishing line $\mathbf{l}$. According to (1), the vanishing line of the water surface was computed from points $\mathbf{p}_l$ and $\mathbf{p}_r$. Thus:

$$\frac{\partial \mathbf{l}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial A}{\partial \vartheta} \\ \frac{\partial B}{\partial \vartheta} \\ \frac{\partial C}{\partial \vartheta} \end{pmatrix} = \begin{pmatrix} \frac{\partial y_{\mathbf{p}_l}}{\partial \vartheta} - \frac{\partial y_{\mathbf{p}_r}}{\partial \vartheta} \\ 0 \\ (1 - W)\frac{\partial y_{\mathbf{p}_l}}{\partial \vartheta} \end{pmatrix},$$

where $\frac{\partial y_{\mathbf{p}_l}}{\partial \vartheta}$ and $\frac{\partial y_{\mathbf{p}_r}}{\partial \vartheta}$ are 1 when the derivatives are taken with respect to $y_{\mathbf{p}_l}$ and $\partial y_{\mathbf{p}_r}$, respectively, and are otherwise 0.

Camera height h. In this work, the camera height was given by (2). The partial derivative of h was:

$$\frac{\partial h}{\partial \vartheta} = \frac{\partial h_g}{\partial \vartheta} + \delta \frac{\partial h_f}{\partial \vartheta} + \frac{\partial h_t}{\partial \vartheta} - \frac{\partial h_s}{\partial \vartheta}.$$

where h_g, h_f, h_t, and h_s are used as input; thus, their partial derivatives are 1 when taken with respect to themselves and are otherwise 0.

Rotation matrix **R**. It was easier to obtain the derivatives of **R** by considering its columns $\vec{\mathbf{R}}_1$, $\vec{\mathbf{R}}_2$, and $\vec{\mathbf{R}}_3$ separately. In this case:

$$\frac{\partial \mathbf{R}}{\partial \vartheta} = \left(\frac{\partial \vec{\mathbf{R}}_1}{\partial \vartheta}, \frac{\partial \vec{\mathbf{R}}_2}{\partial \vartheta}, \frac{\partial \vec{\mathbf{R}}_3}{\partial \vartheta} \right),$$

where

$$\begin{aligned}
\frac{\partial \vec{\mathbf{R}}_1}{\partial \vartheta} &= \vec{\mathbf{R}}_3 \times \frac{\partial \vec{\mathbf{R}}_2}{\partial \vartheta} + \frac{\partial \vec{\mathbf{R}}_3}{\partial \vartheta} \times \vec{\mathbf{R}}_2, \\
\frac{\partial \vec{\mathbf{R}}_2}{\partial \vartheta} &= \frac{Z_{\vec{\mathbf{R}}_3} \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} - \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} Y_{\vec{\mathbf{R}}_3}}{(Y_{\vec{\mathbf{R}}_3}^2 + Z_{\vec{\mathbf{R}}_3}^2)^{\frac{3}{2}}} \left(0, Y_{\vec{\mathbf{R}}_3}, Z_{\vec{\mathbf{R}}_3}\right)^T, \text{ and} \\
\frac{\partial \vec{\mathbf{R}}_3}{\partial \vartheta} &= \pm \frac{1}{(X_{\vec{\mathbf{L}}}^2 + Y_{\vec{\mathbf{L}}}^2 + Z_{\vec{\mathbf{L}}}^2)^{\frac{3}{2}}} \begin{pmatrix} (Y_{\vec{\mathbf{L}}}^2 + Z_{\vec{\mathbf{L}}}^2) \frac{\partial X_{\vec{\mathbf{L}}}}{\partial \vartheta} - X_{\vec{\mathbf{L}}} (Y_{\vec{\mathbf{L}}} \frac{\partial Y_{\vec{\mathbf{L}}}}{\partial \vartheta} + Z_{\vec{\mathbf{L}}} \frac{\partial Z_{\vec{\mathbf{L}}}}{\partial \vartheta}) \\ (X_{\vec{\mathbf{L}}}^2 + Z_{\vec{\mathbf{L}}}^2) \frac{\partial Y_{\vec{\mathbf{L}}}}{\partial \vartheta} - Y_{\vec{\mathbf{L}}} (X_{\vec{\mathbf{L}}} \frac{\partial X_{\vec{\mathbf{L}}}}{\partial \vartheta} + Z_{\vec{\mathbf{L}}} \frac{\partial Z_{\vec{\mathbf{L}}}}{\partial \vartheta}) \\ (X_{\vec{\mathbf{L}}}^2 + Y_{\vec{\mathbf{L}}}^2) \frac{\partial Z_{\vec{\mathbf{L}}}}{\partial \vartheta} - Z_{\vec{\mathbf{L}}} (X_{\vec{\mathbf{L}}} \frac{\partial X_{\vec{\mathbf{L}}}}{\partial \vartheta} + Y_{\vec{\mathbf{L}}} \frac{\partial Y_{\vec{\mathbf{L}}}}{\partial \vartheta}) \end{pmatrix}.
\end{aligned}$$

The orientation of $\frac{\partial \vec{\mathbf{R}}_3}{\partial \vartheta}$ was set according to the up function. Here, $\vec{\mathbf{L}} = (X_{\vec{\mathbf{L}}}, Y_{\vec{\mathbf{L}}}, Z_{\vec{\mathbf{L}}})^T = \mathbf{K}^T \mathbf{l}$ was an auxiliary variable with:

$$\frac{\partial \vec{\mathbf{L}}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial X_{\vec{\mathbf{L}}}}{\partial \vartheta} \\ \frac{\partial Y_{\vec{\mathbf{L}}}}{\partial \vartheta} \\ \frac{\partial Z_{\vec{\mathbf{L}}}}{\partial \vartheta} \end{pmatrix} = \begin{pmatrix} \alpha_x \frac{\partial A}{\partial \vartheta} \\ \gamma \frac{\partial A}{\partial \vartheta} \\ \frac{\partial C}{\partial \vartheta} + x_{\mathbf{o}} \frac{\partial A}{\partial \vartheta} \end{pmatrix}.$$

Matrices **M** *and* $\mathbf{M}^{-1}$. We assumed that the intrinsic parameters of the camera had no uncertainty. So, the derivatives of **M** were given by the multiplication of a constant matrix **K** and $\frac{\partial \mathbf{R}}{\partial \vartheta}$:

$$\frac{\partial \mathbf{M}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial m_{11}}{\partial \vartheta} & \frac{\partial m_{12}}{\partial \vartheta} & \frac{\partial m_{13}}{\partial \vartheta} \\ \frac{\partial m_{21}}{\partial \vartheta} & \frac{\partial m_{22}}{\partial \vartheta} & \frac{\partial m_{23}}{\partial \vartheta} \\ \frac{\partial m_{31}}{\partial \vartheta} & \frac{\partial m_{32}}{\partial \vartheta} & \frac{\partial m_{33}}{\partial \vartheta} \end{pmatrix} = \mathbf{K} \frac{\partial \mathbf{R}}{\partial \vartheta}.$$

The derivatives of the inverse of **M** were:

$$\frac{\partial \mathbf{M}^{-1}}{\partial \vartheta} = -\mathbf{M}^{-1} \frac{\partial \mathbf{M}}{\partial \vartheta} \mathbf{M}^{-1}.$$

Direction $\vec{\mathbf{D}}$. The ray from the camera center to the reference point $\mathbf{X}_1$ required the input of point $\mathbf{x}_1$ and the computation of $\mathbf{M}^{-1}$ (see (3)), from which it followed:

$$\frac{\partial \vec{\mathbf{D}}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial X_{\vec{\mathbf{D}}}}{\partial \vartheta} \\ \frac{\partial Y_{\vec{\mathbf{D}}}}{\partial \vartheta} \\ \frac{\partial Z_{\vec{\mathbf{D}}}}{\partial \vartheta} \end{pmatrix} = \mathbf{M}^{-1} \frac{\partial \mathbf{x}_1}{\partial \vartheta} + \frac{\partial \mathbf{M}^{-1}}{\partial \vartheta} \mathbf{x}_1, \text{ were } \frac{\partial \mathbf{x}_1}{\partial \vartheta} = \left(\frac{\partial x_{\mathbf{x}_1}}{\partial \vartheta}, \frac{\partial y_{\mathbf{x}_1}}{\partial \vartheta}, 0 \right)^T$$

where $\frac{\partial x_{\mathbf{x}_1}}{\partial \vartheta}$ and $\frac{\partial y_{\mathbf{x}_1}}{\partial \vartheta}$ are 1 when taken with respect to themselves and are otherwise 0.

ROI corner $\mathbf{X}_1$ *in world space*. After computing the direction $\vec{\mathbf{D}}$, the computation of the derivatives of $\mathbf{X}_1$ was straightforward:

$$\frac{\partial \mathbf{X}_1}{\partial \vartheta} = \begin{pmatrix} \frac{\partial X_{\mathbf{X}_1}}{\partial \vartheta} \\ \frac{\partial Y_{\mathbf{X}_1}}{\partial \vartheta} \\ \frac{\partial Z_{\mathbf{X}_1}}{\partial \vartheta} \\ \frac{\partial W_{\mathbf{X}_1}}{\partial \vartheta} \end{pmatrix} = \frac{1}{Z_{\vec{\mathbf{D}}}^2} \begin{pmatrix} h X_{\vec{\mathbf{D}}} \frac{\partial Z_{\vec{\mathbf{D}}}}{\partial \vartheta} - Z_{\vec{\mathbf{D}}} \left(X_{\vec{\mathbf{D}}} \frac{\partial h}{\partial \vartheta} + h \frac{\partial X_{\vec{\mathbf{D}}}}{\partial \vartheta} \right) \\ h Y_{\vec{\mathbf{D}}} \frac{\partial Z_{\vec{\mathbf{D}}}}{\partial \vartheta} - Z_{\vec{\mathbf{D}}} \left(Y_{\vec{\mathbf{D}}} \frac{\partial h}{\partial \vartheta} + h \frac{\partial Y_{\vec{\mathbf{D}}}}{\partial \vartheta} \right) \\ 0 \\ 0 \end{pmatrix}.$$

Direction $\vec{\mathbf{E}}$. The back projection of the improper point that was encoding the direction of the vessel in $\mathcal{I}$ was $\vec{\mathbf{E}} = \mathbf{M}^{-1}\vec{\mathbf{u}}$, which led to:

$$\frac{\partial \vec{\mathbf{E}}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial X_{\vec{\mathbf{E}}}}{\partial \vartheta} \\ \frac{\partial Y_{\vec{\mathbf{E}}}}{\partial \vartheta} \\ \frac{\partial Z_{\vec{\mathbf{E}}}}{\partial \vartheta} \end{pmatrix} = \mathbf{M}^{-1}\frac{\partial \vec{\mathbf{u}}}{\partial \vartheta} + \frac{\partial \mathbf{M}^{-1}}{\partial \vartheta}\vec{\mathbf{u}}, \text{ where } \frac{\partial \vec{\mathbf{u}}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial x_{\vec{\mathbf{u}}}}{\partial \vartheta} \\ \frac{\partial v_{\vec{\mathbf{u}}}}{\partial \vartheta} \\ \frac{\partial w_{\vec{\mathbf{u}}}}{\partial \vartheta} \end{pmatrix} = \frac{\partial \phi_{\vec{\mathbf{u}}}}{\partial \vartheta} \begin{pmatrix} -\sin \phi_{\vec{\mathbf{u}}} \\ \cos \phi_{\vec{\mathbf{u}}} \\ 0 \end{pmatrix}.$$

where, as an input variable, the derivative of $\phi_{\vec{\mathbf{u}}}$ is 1 with respect to itself and 0 for any other case.

Vessel direction $\vec{\mathbf{U}}$ *in world space*. We let $\mathbf{A}$ be a constant matrix that encoded the orthogonal projection onto $\mathbf{\Pi}$. We computed $\vec{\mathbf{U}}$ by applying $\mathbf{A}$ to $\vec{\mathbf{E}}$ and normalizing the result to unit length:

$$\vec{\mathbf{U}} = \left(X_{\vec{\mathbf{U}}}, Y_{\vec{\mathbf{U}}}, Z_{\vec{\mathbf{U}}}, W_{\vec{\mathbf{U}}}\right)^T = \operatorname{unit}\left(\mathbf{A}\vec{\mathbf{E}}\right) = \frac{1}{\sqrt{X_{\vec{\mathbf{E}}}^2 + Y_{\vec{\mathbf{E}}}^2}} \left(X_{\vec{\mathbf{E}}}, Y_{\vec{\mathbf{E}}}, 0, 0\right)^T.$$

The partial derivatives of $\vec{\mathbf{U}}$ with respect to ϑ were:

$$\frac{\partial \vec{\mathbf{U}}}{\partial \vartheta} = \left(\frac{\partial X_{\vec{\mathbf{U}}}}{\partial \vartheta}, \frac{\partial Y_{\vec{\mathbf{U}}}}{\partial \vartheta}, \frac{\partial Z_{\vec{\mathbf{U}}}}{\partial \vartheta}, \frac{\partial W_{\vec{\mathbf{U}}}}{\partial \vartheta}\right)^T = \frac{X_{\vec{\mathbf{E}}}\frac{\partial X_{\vec{\mathbf{E}}}}{\partial \vartheta} + Y_{\vec{\mathbf{E}}}\frac{\partial Y_{\vec{\mathbf{E}}}}{\partial \vartheta}}{(X_{\vec{\mathbf{E}}}^2 + Y_{\vec{\mathbf{E}}}^2)^{\frac{3}{2}}} \left(-X_{\vec{\mathbf{E}}}, -Y_{\vec{\mathbf{E}}}, 0, 0\right)^T.$$

Direction $\vec{\mathbf{V}}$. Vector $\vec{\mathbf{V}}$ was computed by rotating $\vec{\mathbf{U}}$ on the mean water plane by $\pi/2$ radians:

$$\vec{\mathbf{V}} = \left(X_{\vec{\mathbf{V}}}, Y_{\vec{\mathbf{V}}}, Z_{\vec{\mathbf{V}}}, W_{\vec{\mathbf{V}}}\right)^T = \mathbf{B}\vec{\mathbf{U}} = \left(-Y_{\vec{\mathbf{U}}}, X_{\vec{\mathbf{U}}}, 0, 0\right)^T,$$

where $\mathbf{B}$ is a constant rotation matrix. The derivatives of $\vec{\mathbf{V}}$ came from $\vec{\mathbf{U}}$:

$$\frac{\partial \vec{\mathbf{V}}}{\partial \vartheta} = \left(\frac{\partial X_{\vec{\mathbf{V}}}}{\partial \vartheta}, \frac{\partial Y_{\vec{\mathbf{V}}}}{\partial \vartheta}, \frac{\partial Z_{\vec{\mathbf{V}}}}{\partial \vartheta}, \frac{\partial W_{\vec{\mathbf{V}}}}{\partial \vartheta}\right)^T = \left(-\frac{\partial Y_{\vec{\mathbf{U}}}}{\partial \vartheta}, \frac{\partial X_{\vec{\mathbf{U}}}}{\partial \vartheta}, 0, 0\right)^T.$$

ROI corners $\vec{\mathbf{X}}_2$ *and* $\vec{\mathbf{X}}_3$ *in world space*. Equation (4) showed how to compute corners $\mathbf{X}_k$, for $k = \{2, 3, 4\}$, from the intermediate variables $\mathbf{X}_1$, $\vec{\mathbf{U}}$, and $\vec{\mathbf{V}}$. However, only $\mathbf{X}_2$ and $\mathbf{X}_3$ were needed in the remainder of the computational flow. Their derivatives were:

$$\frac{\partial \mathbf{X}_2}{\partial \vartheta} = \left(\frac{\partial X_{\mathbf{X}_2}}{\partial \vartheta}, \frac{\partial Y_{\mathbf{X}_2}}{\partial \vartheta}, \frac{\partial Z_{\mathbf{X}_2}}{\partial \vartheta}, \frac{\partial W_{\mathbf{X}_2}}{\partial \vartheta}\right)^T = \frac{\partial \mathbf{X}_1}{\partial \vartheta} - W_{ROI}\frac{\partial \vec{\mathbf{U}}}{\partial \vartheta}, \text{ and}$$

$$\frac{\partial \mathbf{X}_3}{\partial \vartheta} = \left(\frac{\partial X_{\mathbf{X}_3}}{\partial \vartheta}, \frac{\partial Y_{\mathbf{X}_3}}{\partial \vartheta}, \frac{\partial Z_{\mathbf{X}_3}}{\partial \vartheta}, \frac{\partial W_{\mathbf{X}_3}}{\partial \vartheta}\right)^T = \frac{\partial \mathbf{X}_1}{\partial \vartheta} - H_{ROI}\frac{\partial \vec{\mathbf{V}}}{\partial \vartheta}.$$

Camera matrix $\mathbf{P}$. By rewriting the camera matrix as:

$$\mathbf{P} = \left(\mathbf{M} \mid -\mathbf{M}\tilde{C}\right) = \begin{pmatrix} m_{11} & m_{12} & m_{13} & -h\, m_{13} \\ m_{21} & m_{22} & m_{23} & -h\, m_{23} \\ m_{31} & m_{32} & m_{33} & -h\, m_{33} \end{pmatrix},$$

it was easy to see that the partial derivatives of its components were:

$$\frac{\partial \mathbf{P}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial p_{11}}{\partial \vartheta} & \frac{\partial p_{12}}{\partial \vartheta} & \frac{\partial p_{13}}{\partial \vartheta} & \frac{\partial p_{14}}{\partial \vartheta} \\ \frac{\partial p_{21}}{\partial \vartheta} & \frac{\partial p_{22}}{\partial \vartheta} & \frac{\partial p_{23}}{\partial \vartheta} & \frac{\partial p_{24}}{\partial \vartheta} \\ \frac{\partial p_{31}}{\partial \vartheta} & \frac{\partial p_{32}}{\partial \vartheta} & \frac{\partial p_{33}}{\partial \vartheta} & \frac{\partial p_{34}}{\partial \vartheta} \end{pmatrix} = \left(\begin{array}{cc} & -\left(h\frac{\partial m_{13}}{\partial \vartheta} + m_{13}\frac{\partial h}{\partial \vartheta}\right) \\ \frac{\partial \mathbf{M}}{\partial \vartheta} & -\left(h\frac{\partial m_{23}}{\partial \vartheta} + m_{23}\frac{\partial h}{\partial \vartheta}\right) \\ & -\left(h\frac{\partial m_{33}}{\partial \vartheta} + m_{33}\frac{\partial h}{\partial \vartheta}\right) \end{array} \right).$$

ROI corners $\mathbf{x}_2$ *and* $\mathbf{x}_3$ *in image space*. Recall from (5) that, except for the reference corner $\mathbf{x}_1$, the corners $\mathbf{x}_k$ were computed by projecting $\mathbf{X}_k$ from world space onto image space. The partial derivatives of those corners were:

$$\frac{\partial \mathbf{x}_k}{\partial \vartheta} = \begin{pmatrix} \frac{\partial x_{\mathbf{x}_k}}{\partial \vartheta} \\ \frac{\partial y_{\mathbf{x}_k}}{\partial \vartheta} \\ \frac{\partial w_{\mathbf{x}_k}}{\partial \vartheta} \end{pmatrix} = \begin{pmatrix} X_{\mathbf{X}_k} \frac{\partial m_{11}}{\partial \vartheta} + m_{11} \frac{\partial X_{\mathbf{X}_k}}{\partial \vartheta} + Y_{\mathbf{X}_k} \frac{\partial m_{12}}{\partial \vartheta} + m_{12} \frac{\partial Y_{\mathbf{X}_k}}{\partial \vartheta} + \frac{\partial p_{14}}{\partial \vartheta} \\ X_{\mathbf{X}_k} \frac{\partial m_{21}}{\partial \vartheta} + m_{21} \frac{\partial X_{\mathbf{X}_k}}{\partial \vartheta} + Y_{\mathbf{X}_k} \frac{\partial m_{22}}{\partial \vartheta} + m_{22} \frac{\partial Y_{\mathbf{X}_k}}{\partial \vartheta} + \frac{\partial p_{24}}{\partial \vartheta} \\ X_{\mathbf{X}_k} \frac{\partial m_{31}}{\partial \vartheta} + m_{31} \frac{\partial X_{\mathbf{X}_k}}{\partial \vartheta} + Y_{\mathbf{X}_k} \frac{\partial m_{32}}{\partial \vartheta} + m_{32} \frac{\partial Y_{\mathbf{X}_k}}{\partial \vartheta} + \frac{\partial p_{34}}{\partial \vartheta} \end{pmatrix}.$$

Homographic matrix $\mathbf{H}$. In (6), $\mathbf{H}$ was computed by multiplying the matrices $\mathbf{H}_A$ and $\mathbf{H}_\mathbf{l}$, where the former was defined by the coefficients of the right-hand null space vector $\vec{\mathbf{v}}$ and the later was related to the vanishing line $\mathbf{l}$. In its final form:

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{33} & h_{33} \end{pmatrix} = \mathbf{H}_A \mathbf{H}_\mathbf{l}.$$

The partial derivatives of $\mathbf{H}$ with respect to ϑ were:

$$\frac{\partial \mathbf{H}}{\partial \vartheta} = \begin{pmatrix} \frac{\partial h_{11}}{\partial \vartheta} & \frac{\partial h_{12}}{\partial \vartheta} & \frac{\partial h_{13}}{\partial \vartheta} \\ \frac{\partial h_{21}}{\partial \vartheta} & \frac{\partial h_{22}}{\partial \vartheta} & \frac{\partial h_{23}}{\partial \vartheta} \\ \frac{\partial h_{31}}{\partial \vartheta} & \frac{\partial h_{32}}{\partial \vartheta} & \frac{\partial h_{33}}{\partial \vartheta} \end{pmatrix} = \mathbf{H}_A \frac{\partial \mathbf{H}_\mathbf{l}}{\partial \vartheta} + \frac{\partial \mathbf{H}_A}{\partial \vartheta} \mathbf{H}_\mathbf{l},$$

where

$$\frac{\partial \mathbf{H}_A}{\partial \vartheta} = \begin{pmatrix} \frac{v_{33} \frac{\partial v_{11}}{\partial \vartheta} - v_{11} \frac{\partial v_{33}}{\partial \vartheta}}{v_{33}^2} & \frac{v_{33} \frac{\partial v_{12}}{\partial \vartheta} - v_{12} \frac{\partial v_{33}}{\partial \vartheta}}{v_{33}^2} & \frac{v_{33} \frac{\partial v_{13}}{\partial \vartheta} - v_{13} \frac{\partial v_{33}}{\partial \vartheta}}{v_{33}^2} \\ \frac{v_{33} \frac{\partial v_{21}}{\partial \vartheta} - v_{21} \frac{\partial v_{33}}{\partial \vartheta}}{v_{33}^2} & \frac{v_{33} \frac{\partial v_{22}}{\partial \vartheta} - v_{22} \frac{\partial v_{33}}{\partial \vartheta}}{v_{33}^2} & \frac{v_{33} \frac{\partial v_{23}}{\partial \vartheta} - v_{23} \frac{\partial v_{33}}{\partial \vartheta}}{v_{33}^2} \\ 0 & 0 & 1 \end{pmatrix}, \text{and}$$

$$\frac{\partial \mathbf{H}_\mathbf{l}}{\partial \vartheta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{\partial A}{\partial \vartheta} & 0 & \frac{\partial C}{\partial \vartheta} \end{pmatrix}.$$

The procedure to obtain the derivatives of $\vec{\mathbf{v}}$ from the singular value decomposition can be found in the literature [33].

Location $\mathbf{c}_1$ *and* $\mathbf{c}_2$ *of troughs in the rectified ROI*. We were not able to estimate the uncertainty of the location of the maximum and minimum of the smooth curve that was computed by the LOWESS algorithm. Our approach to not interrupting the uncertainty propagation of the computational chain while it was estimating $\mathbf{c}_1$ and $\mathbf{c}_2$ in the rectified ROI was:

1. Take the coordinates of the troughs in the rectified ROI;
2. Map them onto $\mathcal{I}$ using the inverse homography $\mathbf{H}^{-1}$, thereby obtaining $\mathbf{c}'_1$ and $\mathbf{c}'_2$:

$$\mathbf{c}'_j = \left(x_{\mathbf{c}'_j}, y_{\mathbf{c}'_j}, w_{\mathbf{c}'_j}\right)^T = \mathbf{H}^{-1}\left(x_{\mathbf{c}_j}, y_{\mathbf{c}_j}, 1\right)^T;$$

3. Consider that $\mathbf{c}'_1$ and $\mathbf{c}'_2$ are variables without uncertainty (see the pentagons in Figure 5);
4. Transport the uncertainty that are propagated on $\mathbf{H}$ onto $\mathbf{c}_1$ and $\mathbf{c}_2$ by mapping $\mathbf{c}'_1$ and $\mathbf{c}'_2$ back to the rectified ROI using:

$$\mathbf{c}_j = \left(x_{\mathbf{c}_j}, y_{\mathbf{c}_j}, 1\right)^T = \mathbf{H}\mathbf{c}'_j = \mathbf{H}\left(x_{\mathbf{c}'_j}, y_{\mathbf{c}'_j}, w_{\mathbf{c}'_j}\right)^T.$$

The derivatives of $\mathbf{c}_j$, for $j \in \{1,2\}$, were:

$$\frac{\partial \mathbf{c}_j}{\partial \vartheta} = \begin{pmatrix} \frac{\partial x_{\mathbf{c}_j}}{\partial \vartheta} \\ \frac{\partial y_{\mathbf{c}_j}}{\partial \vartheta} \\ \frac{\partial w_{\mathbf{c}_j}}{\partial \vartheta} \end{pmatrix} = \begin{pmatrix} \frac{\partial h_{11}}{\partial \vartheta} x_{\mathbf{c}'_j} + \frac{\partial h_{12}}{\partial \vartheta} y_{\mathbf{c}'_j} + \frac{\partial h_{13}}{\partial \vartheta} w_{\mathbf{c}'_j} \\ \frac{\partial h_{21}}{\partial \vartheta} x_{\mathbf{c}'_j} + \frac{\partial h_{22}}{\partial \vartheta} y_{\mathbf{c}'_j} + \frac{\partial h_{23}}{\partial \vartheta} w_{\mathbf{c}'_j} \\ \frac{\partial h_{31}}{\partial \vartheta} x_{\mathbf{c}'_j} + \frac{\partial h_{32}}{\partial \vartheta} y_{\mathbf{c}'_j} + \frac{\partial h_{33}}{\partial \vartheta} w_{\mathbf{c}'_j} \end{pmatrix}.$$

Vessel speed U. By replacing D and λ (7) in (8), U was written in terms of a constant term r that multiplied the square root of the Euclidean distance between points $\mathbf{c}_1$ and $\mathbf{c}_2$:

$$U = r\sqrt[4]{(x_{\mathbf{c}_2} - x_{\mathbf{c}_1})^2 + (y_{\mathbf{c}_2} - y_{\mathbf{c}_1})^2}, \text{ where } r = 1.944\sqrt{\frac{g\sqrt{3}}{4\pi}}.$$

The components of ∇_U (10) were computed as:

$$\frac{\partial U}{\partial \vartheta} = \frac{r\left((x_{\mathbf{c}_1} - x_{\mathbf{c}_2})\left(\frac{\partial x_{\mathbf{c}_1}}{\partial \vartheta} - \frac{\partial x_{\mathbf{c}_2}}{\partial \vartheta}\right) + (y_{\mathbf{c}_1} - y_{\mathbf{c}_2})\left(\frac{\partial y_{\mathbf{c}_1}}{\partial \vartheta} - \frac{\partial y_{\mathbf{c}_2}}{\partial \vartheta}\right)\right)}{2\left((x_{\mathbf{c}_1} - x_{\mathbf{c}_2})^2 + (y_{\mathbf{c}_1} - y_{\mathbf{c}_2})^2\right)^{\frac{3}{4}}}.$$

References

1. Hand, M. Autonomous Shipping: Are We Being Blinded by Technology? 2019. Available online: https://www.seatrade-maritime.com/asia/autonomous-shipping-are-we-being-blinded-technology (accessed on 12 december 2020).
2. Hellenic Shipping News Worldwide. Autonomous Shipping: Trends and Innovators in a Growing Industry. 2020. Available online: https://www.nasdaq.com/articles/autonomous-shipping%3A-trends-and-innovators-in-a-growing-industry-2020-02-18 (accessed on 12 december 2020).
3. Li, Q.; Wang, C.; Li, X.; Wen C. FeatFlow: Learning geometric features for 3D motion estimation. *Pattern Recognit.* **2021**, *111*, 107574. [CrossRef]
4. Wawrzyniak, N.; Hyla, T.; Popik, A. Vessel detection and tracking method based on video surveillance. *Sensors* **2019**, *19*, 5230. [CrossRef] [PubMed]
5. Panico, A.; Graziano, M. D.; Renga, A. SAR-based vessel velocity estimation from partially imaged Kelvin Pattern. *IEEE Geosci. Remote. Sens. Lett.* **2017**, *14*, 2067–2071. [CrossRef]
6. Wang, C.; Shen, P.; Li, X.; Zhu, J.; Li, Z. A novel vessel velocity estimation method using dual-platform TerraSAR-X and TanDEM-X full polarimetric SAR data in pursuit monostatic mode. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6130–6144. [CrossRef]
7. Huang, S.; Liu, D.; Gao, G.; Guo, X. A novel method for speckle noise reduction and ship target detection in SAR images. *Pattern Recognit.* **2009**, *42*, 1533–1542. [CrossRef]
8. Guo, H.; Yang, X.; Wang, N.; Gao, X. A CenterNet++ model for ship detection in SAR images. *Pattern Recognit.* **2021**, *112*, 107787. [CrossRef]
9. Liu, Y.; Zhao, J.; Qin, Y. A novel technique for ship wake detection from optical images. *Remote Sens. Environ.* **2021**, *258*, 112375. [CrossRef]
10. Broggi, A.; Cerri, P.; Grisleri, P.; Paterlini, M. Boat speed monitoring using artificial vision. In *Image Analysis and Processing—ICIAP 2009, Proceedings of the 15th International Conference, Vietri sul Mare, Italy, 8–11 September 2009*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 327–336.
11. Tan, S.; Piepmeier, J. A.; Kriebel, D. L. A computer vision system for monitoring vessel motion in conjunction with vessel wake measurements. In Proceedings of the 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, USA, 4–7 Novemebr 2012; pp. 1830–1834.
12. Peng, J.; Wang, T.; Lin, W.; Wang, J.; See, J.; Wen, S.; Ding, E. TPM: Multiple object tracking with tracklet-plane matching. *Pattern Recognit.* **2020**, *102*, 107480. [CrossRef]
13. Huillca, J.L.; Fernandes, L.A.F. Computing vessel velocity from single perspective projection images. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 325–329.
14. Thomson, W. On ship waves. *Proc. Inst. Mech. Eng.* **1887**, *38*, 409–434. [CrossRef]
15. Newman, J.N. The inverse ship-wave problem. In Proceedings of the Sixth International Workshop on Water Waves and Floating Bodies, Falmouth, MA, USA, 14–17 April 1991; pp. 193–197.
16. Taylor, J.R. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, 2nd ed.; University Science Books: Sausalito, CA, USA, 1997.
17. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
18. Workman, S.; Zhai, M.; Jacobs, N. Horizon lines in the wild. *arXiv* **2016**, arXiv:1604.02129.

19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Breitinger, A.; Clua, E.; Fernandes, L.A.F. An Augmented Reality Periscope for Submarines with Extended Visual Classification. *Sensors* **2021**, *21*, 7624. [CrossRef] [PubMed]
21. Liu, Y.; Cheng, M. M.; Hu, X.; Bian, J.; Zhang, L.; Bai, X.; Tang, J. Richer convolutional features for edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1939–1946. [CrossRef] [PubMed]
22. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; pp 281–297.
23. Wilcox, R. The regression smoother LOWESS: A confidence band that allows heteroscedasticity and has some specified simultaneous probability coverage. *J. Mod. Appl. Stat. Methods* **2017**, *16*, 29–38. [CrossRef]
24. Savitzky, A.; Golay, M.J.E. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* **1964**, *36*, 1627–1639. [CrossRef]
25. Fernandes, L.A.F.; Oliveira, M.; Silva, R. Uncertainty propagation: Avoiding the expensive sampling process for real-time image-based measurements. *Comput. Stat. Data Anl.* **2008**, *52*, 3852–3876. [CrossRef]
26. Nikon. User's Manual Nikon D3300. Available online: https://downloadcenter.nikonimglib.com/pt/products/21/D3300.html (accessed on 12 December 2020).
27. Tábuas de Maré. Available online: https://www.marinha.mil.br/chm/tabuas-de-mare (accessed on 18 November 2019).
28. Tábua de Marés e SOLUNARES de Pescaria. Available online: https://tabuademares.com (accessed on 18 November 2019).
29. TidesChart: Check the Tide Anywhere in World. Available online: https://pt.tideschart.com (accessed on 18 November 2019).
30. FURUNO S.A. FAR-21x7 Series Brochure. Available online: https://www.furuno.com/files/Brochure/236/upload/far-21x7.pdf (accessed on 12 December 2020).
31. Shapiro, S. S.; Wilk, M. B. An analysis of variance test for normality. *Biometrika* **1965**, *52*, 591–611. [CrossRef]
32. Joint Committee for Guides in Metrology. *Evaluation of Measurement Data—Guide to the Expression of Uncertainty in Measurement*; BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML, JCGM 100:2008, GUM 1995 with Minor Corrections; International Bureau of Weights and Measures (BIPM): Sèvres, France, 2008.
33. Papadopoulo, T.; Lourakis, M. I. A. Estimating the Jacobian of the Singular Value Decomposition. In *Computer Vision—ECCV 2000, Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland, 26 June–1 July 2000*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 554–570.