

UNIVERSIDADE FEDERAL FLUMINENSE

**JOSE LUIS HUILLCA MANGO**

COMPUTING VESSEL VELOCITY FROM SINGLE  
PERSPECTIVE PROJECTION IMAGES

NITERÓI

2020

UNIVERSIDADE FEDERAL FLUMINENSE

**JOSE LUIS HUILLCA MANGO**

# COMPUTING VESSEL VELOCITY FROM SINGLE PERSPECTIVE PROJECTION IMAGES

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Computação Visual.

Orientador:

Prof. Dr. Leandro Augusto Frata Fernandes

NITERÓI

2020

Ficha catalográfica automática - SDC/BEE  
Gerada com informações fornecidas pelo autor

H899c Huillca mango, Jose Luis  
COMPUTING VESSEL VELOCITY FROM SINGLE PERSPECTIVE PROJECTION  
IMAGES / Jose Luis Huillca mango ; Leandro Augusto Frata  
Fernandes, orientador. Niterói, 2020.  
119 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense,  
Niterói, 2020.

DOI: <http://dx.doi.org/10.22409/PGC.2020.m.06479440730>

1. Visão computacional. 2. Processamento de imagem. 3.  
Produção intelectual. I. Frata Fernandes, Leandro Augusto,  
orientador. II. Universidade Federal Fluminense. Instituto de  
Computação. III. Título.

CDD -

JOSE LUIS HUILLCA MANGO

COMPUTING VESSEL VELOCITY FROM SINGLE PERSPECTIVE PROJECTION IMAGES

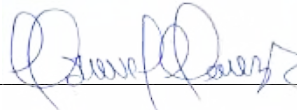
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Computação Visual.

Aprovada em abril de 2020.

BANCA EXAMINADORA



Prof<sup>a</sup>. Leandro Augusto Frata Fernandes, Dr. – Orientador, UFF



Prof. Manuel Menezes de Oliveira Neto, Dr. – UFRGS



Prof. Cesar Beltran Castañon, Dr. – PUCP-Lima



Prof. Anselmo Antunes Montenegro, Dr. – UFF

Niterói

2020

*To my parents Carmela Mango Condori and Pedro Pablo Huillca and  
my brother Juan Felix Huillca Mango.*

# Acknowledgments

First and foremost, I would like to thank my family, mainly my mother, for all the support she always gives me. I would also like to thank my advisor, Prof. Leandro Augusto Frata Fernandes, for teachings, academic orientation, and for the patience he had with me throughout the master's degree.

I would like to give a special thanks to all the people who made it possible to lend us the radar to validate the proposed technique, especially Arthur Ayres Neto, from *Laboratório de Geologia Marinha* (LAGEMAR) - UFF, Fabio Hochleitner, from *Laboratório de Métodos Computacionais em Engenharia* (LAMCE) - COPPE/UFRJ, and Artur Brandão, from *Ello Projetos e Consultoria Ltda.* Also, thank my friend Jacó Julio for helping me take the vessel images.

I am very grateful with the UFF computer postgraduate program, for the knowledge provided, and CAPES for the award of a research scholarship.

# Resumo

Neste trabalho, descrevemos uma abordagem baseada em imagem para estimar a velocidade da embarcação em movimento a partir dos rastros que permanecem na superfície da água. Apenas uma imagem obtida com projeção em perspectiva foi usada para estimar a velocidade. As embarcações que se movem em velocidade e direção constantes mostram um padrão familiar em forma de V, diferenciando apenas um do outro pelo comprimento de onda. Esse comprimento de onda está relacionado à velocidade da embarcação. Na técnica proposta, conceitos de geometria projetiva, homografia plana e restrições naturais foram utilizados para encontrar as cristas/vales dos vasos e, consequentemente, encontrar o comprimento de onda. Nós utilizamos imagens ópticas adquiridas por câmeras convencionais.

Definimos uma região de interesse que continha o rastro deixada pela embarcação para calcular a velocidade. Para localizar a região de interesse, escolhemos manualmente uma posição na imagem de entrada que representa um canto da região de interesse. A imagem de entrada também foi convertida em uma imagem de borda para identificar melhor os rastros das embarcações. O algoritmo  $k$ -means foi usado para agrupar o rastro em dois braços de onda (o padrão em forma de V da onda). Usamos apenas o braço ondulado mais próximo da câmera e menos afetado pela turbulência para encontrar o comprimento de onda. Usando o problema inverso das ondas de Kelvin, conseguimos estimar a velocidade da embarcação.

Um modelo algorítmico foi desenvolvido para calcular a velocidade da embarcação usando apenas uma imagem. Para validar o método proposto, usamos um radar projetado para monitorar embarcações. Capturamos imagens da tela do radar com informações sobre a velocidade da embarcação e comparamos com o resultado do nosso método. As fotos dos navios e a tela do radar foram tiradas ao mesmo tempo.

Neste trabalho, estudos estatísticos foram apresentados para determinar a confiabilidade da velocidade obtida. Também foi apresentado um estudo sobre a propagação de erros ao longo do processo de cálculo da velocidade. Experimentos mostram que o método proposto produz resultados convincentes que concordam com os dados reais obtidos pelo radar.

**Palavras-chave:** Velocidade da embarcação, imagem óptica, geometria computacional, detecção de objetos, homografia plana, geometria de cristas de navios.

# Abstract

In this work, we describe an image-based approach to estimate the vessel's speed in motion from the wakes that remain on the water surface. Only one image obtained with perspective projection was used to estimate speed. Vessels moving at constant speed and direction show a familiar V-shaped pattern, differing only from each other by wavelength. This wavelength is related to vessel speed. In the proposed technique, projective geometry concepts, planar homography, and natural constraints were used to find the crests/troughs of the vessels and, consequently, find the wavelength. We use optical images acquired by conventional cameras.

We defined a region of interest that contained the wake left by the ship to calculate the vessel speed. To find the region of interest, we manually choose a position in the input image representing a corner of the region of interest. The input image was also converted to an edge image to identify the wakes of the vessels better. The  $k$ -means algorithm was used to group the wake into two wave arms (the V-shaped pattern of the wave). We only use the wave arm closest to the camera and least affected by turbulence to find the wavelength. Using the inverse problem of Kelvin waves, we were able to estimate the vessel speed.

An algorithmic model was developed to calculate the vessel speed using only one image. To validate the proposed method, we used a radar designed to monitor vessels. We captured images from the radar screen with information on the vessel speed and compare it with the result of our method. The pictures of the vessels and the radar screen were taken at the same time.

In this work, statistical studies were presented to determine the reliability of the obtained speed. A study on the propagation of errors throughout the speed calculation process was also presented. Experiments show that the proposed method produces convincing results that agree with the actual data obtained by the radar.

**Keywords:** Vessel velocity, ship wakes, optical image, computational geometry, object detection, planar homography, ship wake crests geometry.



# List of Figures

1.1	Our approach takes the cusp wave arms, performs geometric rectification of the mean water plane, and uses the distance between crests/troughs to estimate vessel speed. . . . .	17
1.2	Flowchart used in this manuscript to estimate the vessel speed. Each box represents a chapter in this manuscript. The input image for the proposed method is an image containing one or more vessels, where the trace left by it is visible, and the output is the vessel speed. . . . .	19
2.1	Two basic ways to represent $f_{x,y}$ . (a) is a plot of the function, with two axes determining spatial location and the third axis being the values of $f$ (intensities) as a function of the two spatial variables $x$ and $y$ . (b) It shows $f_{x,y}$ as it would appear on a monitor or photograph. Image extracted from [Gonzalez 2002] . . . . .	22
2.2	Some examples of Richer Convolutional Features (RCF). The images were taken from the <i>Instituto de Computação</i> of the <i>Universidade Federal Fluminense</i> , facing the entrance of the Guanabara Bay. The four images were taken at a resolution of $6000 \times 4000$ pixels. The two images in the first column are cropped images, having a size of $800 \times 600$ pixels each. . . . .	24
2.3	Pinhole camera model: $\mathbf{O}$ is the camera's projection center; $\mathbf{X}'$ is a point in 3D space, projected at point $\mathbf{x}'$ on the image plane; and $\mathbf{o}$ is the main point and origin of the image coordinate system. The focal length $f$ is calculated between the distance from $\mathbf{O}$ to $\mathbf{o}$ . . . . .	25
2.4	The two sets of parallel lines on the scene plane $\pi$ converge to the vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$ in the image. The line $\mathbf{l}$ through $\mathbf{v}_1$ and $\mathbf{v}_2$ is the vanishing line of the plane. . . . .	28
2.5	We need to identify four points on the plane such that the points are used to provide constraints on the affine transformation. . . . .	29
2.6	The wake crests diagram includes transverse and divergent crests. The envelop connects the cusp waves. . . . .	31

2.7	Image (a) represents image (b) in 3D space. Image (b) represents the Kelvin wave structure viewed from a processed Synthetic-Aperture Radar imagery. Images extracted from [Maver Ksenija 2004] . . . . .	31
3.1	Kelvin waves pattern seen from a SAR imagery. Image extracted from [Panico, Graziano e Renga 2017]	
4.1	Vanishing lines found using the Horizon Lines in the Wild (HLW) algorithm, with the <i>so_imagenet</i> method. The red dots in each image (left and right points) are the result of the HLW algorithm. The vanishing line is the line that passes through these two points, a red line. The images were taken in different settings, different weather conditions, and different ambient lighting. The top three images are <i>i1</i> , <i>i2</i> , and <i>i15</i> , respectively, shown in Chapter 9, Figures 9.2, and 9.4. The results are very close to the actual vanishing line. . . . .	41
5.1	The vanishing line $\mathbf{l}$ of a plane $\pi$ is obtained by intersecting the image plane with a plane through the camera center $\mathbf{C}_h$ and parallel to $\pi$ . In the 3-dimensional space, the Region of Interest (ROI) lies in $\pi$ , and $\vec{\mathbf{n}}$ is a normal vector. The initial coordinates of the ROI in the image, i.e., the image point $\mathbf{x}_1$ , is obtained manually. . . . .	44
5.2	The image shows the positions of the four corners $\{\mathbf{x}_k\}_{k=1}^4$ of the ROI on the plane $\pi$ . Where $\mathbf{l}$ is the vanishing line, and $\mathbf{v}_p$ is a vanishing point. . . . .	46
6.1	Representing the ROI's bounding box. Image (a) is a cropped image to visualize the ROI and the ROI's bounding box, in blue line and green dotted line, respectively . Image (b) shows the four corners $\{\mathbf{x}_k^*\}_{k=1}^4$ of the ROI's bounding box in detail, where $a \geq 0$ is a constant value to expand the ROI's bounding box. . . . .	49
6.2	Image binarization of the ROI's bounding box using a threshold $t$ calculated by the Otsu algorithm [Otsu 1979]. . . . .	50
6.3	The image shows the windows used where the k-means algorithm was applied. Each $q_i$ window has a width of $d$ pixels and a height equal to that of the new ROI. Each window slides by $\alpha$ pixels, for $\alpha \geq 1$ . . . . .	51
6.4	Image shows the waves separated into two groups, using k-means, an upper wave and a bottom wave, depicted with green and red colors, respectively. . . . .	51

7.1	Image (a) shows the Kelvin waves' structure, indicating the transverse and divergent components, along with the crests and troughs of the wave arm. Image (b) represents the Kelvin wave structure viewed from a processed Synthetic-Aperture Radar imagery. Image (a) represents image (b) in 3D space. The dotted red line represents the chosen wave arm. Green stars are the troughs used to find the wavelength. Images (a) and (b) were extracted from [Maver Ksenija 2004] . . . . .	54
7.2	Image (a) shows one of the waves (the closest wave) found in Chapter 6. Image (b) shows the rectified ROI image showing only the closest wave, where the ROI size is in centimeters. . . . .	55
7.3	The black points correspond to the digital curve in this example, while the red line is the resulting smooth curve using LOWESS algorithm [Wilcox 2017]. Axes in centimeters. . . . .	57
7.4	The red line represents the fitting curve of the right wave arm. The green stars and yellow dots represent the troughs found on the wave. Only the green stars are used to calculate the wavelength, the left and right green stars correspond to $\mathbf{c}_n$ , and $\mathbf{c}_{n+1}$ , respectively. The wave arm shown is from a ship going in the right direction. . . .	59
7.5	The wave arm of the ship going in the left direction, image (a), is similar to rotating $180^\circ$ the wave arm of the ship going in the right direction, image (b). When making the rotation, the point-views of the crests and troughs change according to the ship's course. Taking image (b) as a reference, the yellow stars in image (b) are the troughs and white stars are crests, while in image (a) the white stars are the troughs and yellow stars are crests. . . . .	59
8.1	Error propagation along the computational chain. It was not considered an uncertainty for the parameters represented by pentagons. In contrast, the input parameters represented as circles contain errors that are propagated by the intermediate stages (squares) until the velocity (rhombus) is calculated. . . . .	63
9.1	Examples of screenshots of the $FAR - 21 \times 7$ radar monitor. The pictures were taken with a smartphone camera. . . . .	79
9.2	Cropping the images used in the experiments. Image $i1$ shows the a HSC vessel. Images $i2 - i6$ show a MC25 vessel. . . . .	81
9.3	Cropping the images used in the experiments. Images $i8$ , and $i9$ show a HSC vessel. Images $i7, i10, i11$ , and $(i12)$ show a MC25 vessel. . . . .	82
9.4	Cropping the images used in the experiments. All images sinclude a MC25 vessel. .	83

9.5	Cropping the images used in the experiments. All images sinclude a MC25 vessel. .	84
9.6	The first column shows the confidence intervals (CI), presented in Table 9.7, using samples randomly created. The second column shows the result of the confidence intervals using error propagation. The wave arm troughs were used to create the confidence intervals. . . . .	91
9.7	The first column shows the confidence intervals (CI), presented in Table 9.9, using samples randomly created. The second column shows the result of the confidence intervals using error propagation. The wave arm crests were used to create the confidence intervals. . . . .	92
9.8	Images with smaller standard deviation (a-e), so the confidence interval is narrower. The images with higher standard deviation are presented in (f, g, h). Here, $U$ is the estimated speed and $s$ is the standard deviation calculated using First-Order Error Propagation. . . . .	96
A.1	Four of the 30 images were chosen to visualize the results of the models presented by the HLW algorithm. The black line was found manually. It serves as a reference to identify the best method for detecting the vanishin line. . . . .	107
A.2	Cumulative distribution of errors for the HWL algorithm models. 30 images were used. <i>so_imagenet</i> presents good results, obtaining a very low error. This model is used to find future vanishing lines. . . . .	108
B.1	Cropping the images used in the experiments. Images <i>i24</i> – <i>i29</i> shown in Table B.1.	113
B.2	Cropping the images used in the experiments. Images <i>i30</i> – <i>i35</i> shown in Table B.1.	114
B.3	Cropping the images used in the experiments. Images <i>i36</i> – <i>i40</i> shown in Table B.1.	115
B.4	Each image shows the region of interest rectified along with the image name and the speed calculated by the proposed method, where the red dotted line is the chosen wave arm to find the wavelength. The images where it was not possible to identify the traces of the vessel and estimate the speed were shown as unknown speed. . . .	116

# List of Tables

9.1	Ground-truth information of the vessel in each captured image. . . . .	80
9.3	Table showing absolute error( $\varepsilon_a$ ) and relative error( $\varepsilon_r$ ) using the wave arms troughs.	86
9.5	Table showing absolute error( $\varepsilon_a$ ) and relative error( $\varepsilon_r$ ) using the wave arms crests.	87
9.7	Confidence intervals for estimated speeds in <i>knots</i> , using the wave arms troughs. .	89
9.9	Confidence intervals for estimated speeds in <i>knots</i> , using the wave arms crests. . .	90
9.11	Normality Tests using Shapiro-Wilk. If $\rho - value > \alpha$ then the null hypothesis cannot be rejected, where $\alpha = 0.05$ . It was used the wave arm troughs for the Normality Test. . . . .	94
9.14	Normality Tests using Shapiro-Wilk. If $\rho - value > \alpha$ then the null hypothesis cannot be rejected, where $\alpha = 0.05$ . It was used the wave arm crests for the Normality Test. . . . .	95
A.1	Result of the cosine distance between each method for detecting the vanishing line and the reference line on the images presented in Figure A.1. The smaller distances are highlighted in red. . . . .	107
A.3	Tides Table of Rio de Janeiro - Niteroi. The values in this table are used to estimate the average error of the tide between three sources: Nautide [Nautide], Tide Chart [Tides Chart], and <i>Marinha do Brasil</i> [Marinha do Brasil], respectively. . . . .	110
B.1	Ground-truth information of the vessel in each captured image. This images were not considered in the main experiments. . . . .	112

# List of Abbreviations and Acronyms

<b>UFF</b>	Universidade Federal Fluminense . . . . .	20
<b>ROI</b>	Region of Interest . . . . .	vii
<b>GPS</b>	Global Positioning System . . . . .	16
<b>GNSS</b>	Global Navigation Satellite Systems . . . . .	34
<b>CI</b>	Confidence Interval . . . . .	88
<b>HLW</b>	Horizon Lines in the Wild . . . . .	vii
<b>SAR</b>	Synthetic-Aperture Radar . . . . .	18
<b>HED</b>	Holistically Nested Edge Detection . . . . .	23
<b>RCF</b>	Richer Convolutional Features . . . . .	vi
<b>CNN</b>	Convolutional Neural Network . . . . .	21
<b>UTC</b>	Coordinated Universal Time . . . . .	34
<b>SVD</b>	Singular Value Decomposition . . . . .	57
<b>LOWESS</b>	LOcally WEighted Scatter-plot Smoother . . . . .	58
<b>LIDAR</b>	Light Detection and Ranging . . . . .	33
<b>DGPS</b>	Differential GPS . . . . .	34
<b>NCC</b>	Normalized Cross Correlation . . . . .	37
<b>AIS</b>	Automatic Identification System . . . . .	85

# Contents

1	Introduction	16
1.1	Motivation . . . . .	18
1.2	Research Objective . . . . .	18
1.2.1	Secondary Objectives . . . . .	18
1.2.2	Validation Approach . . . . .	20
1.3	Contributions . . . . .	20
2	Background	21
2.1	Digital Image . . . . .	21
2.2	Edge Detection . . . . .	22
2.2.1	Deep Learning Based Algorithms . . . . .	23
2.3	The Pinhole Camera Model . . . . .	24
2.4	Camera Parameters . . . . .	25
2.5	Projective Geometry . . . . .	26
2.5.1	Vanishing Points and Vanishing Lines . . . . .	27
2.5.2	Projective Transformations . . . . .	28
2.6	Kelvin Waves . . . . .	30
2.7	Discussion . . . . .	32
3	Related Work	33
3.1	Devices Used to Estimate Speed . . . . .	33
3.1.1	Radar Detectors . . . . .	33

3.1.2	Lidar Detector . . . . .	34
3.1.3	Satellite Navigation Device . . . . .	34
3.1.4	Vision Based System . . . . .	34
3.2	Velocity Estimation Using Images and Videos . . . . .	35
3.3	Velocity Estimation Using Radar Images . . . . .	36
3.4	Discussion . . . . .	38
4	Vanishing Line Estimation	39
4.1	Vanishing Line . . . . .	39
4.2	Finding the Vanishing Line . . . . .	39
4.3	Discussion . . . . .	41
5	Definition of the Corners of the Region of Interest	43
5.1	Region of Interest . . . . .	43
5.2	Finding the Remaining Corners of the ROI . . . . .	44
5.2.1	Camera Height . . . . .	44
5.2.2	Remainig Corners of the ROI . . . . .	45
5.3	Discussion . . . . .	46
6	Finding the Wave Arms	48
6.1	The ROI's Bounding Box . . . . .	48
6.2	ROI Image Binarization . . . . .	49
6.3	Wave Arms Clusterization . . . . .	50
6.4	Discussion . . . . .	52
7	Vessel Velocity Estimation	53
7.1	ROI Image Rectification . . . . .	53
7.1.1	Findig the Homographic Matrix . . . . .	54
7.2	Curve Fitting . . . . .	57



7.3	Wavelength Estimation . . . . .	58
7.4	Vessel Velocity Estimation . . . . .	60
7.5	Discussion . . . . .	60
8	First-Order Error Propagation	61
8.1	Error Propagation . . . . .	62
8.2	A Line Through Two Points . . . . .	65
8.3	Camera Height . . . . .	65
8.4	Rotation Matrix . . . . .	66
8.5	Sub-matrix of the Projection Matrix . . . . .	68
8.6	Projection Matrix . . . . .	69
8.7	Compute the Location of the Four Corners of the ROI . . . . .	70
8.8	Matrix of the Projective Transformation . . . . .	73
8.9	Find Peaks . . . . .	75
8.10	Vessel Velocity Estimation . . . . .	76
8.11	Discussion . . . . .	76
9	Analysis of the Results	78
9.1	Materials . . . . .	79
9.2	Analysis of Relative Error . . . . .	85
9.3	Analysis of Confidence Intervals . . . . .	87
9.4	Analysis of the Error Propagation . . . . .	90
9.4.1	Errors in Input Data . . . . .	93
9.4.2	Verifying the Use of Error Propagation . . . . .	94
9.4.3	Analyzing the Results . . . . .	95
9.5	Discussion . . . . .	96
10	Conclusion and Future Work	98

---

10.1 Future Works . . . . .	100
References	101
Appendix A – Standard Deviation for Input Variables	106
A.1 Finding the Standard Deviation for the Vanishing Line . . . . .	106
A.2 Finding the Standard Deviation of the Camera Height . . . . .	109
Appendix B – Vessel Images not Considered in the Experiments	111
B.1 Analysis of Results . . . . .	111

# Chapter 1

## Introduction

One of the most significant technological advances in the maritime industry is electronic navigation, where Satellite images and Global Positioning System (GPS) have been widely used [Tetley e Calcutt 2001]. These systems, using a monitoring system, keep the pilot informed about the location and speed of nearby vessels. Unfortunately, there is the possibility of difficulties in both transmission and reception. Also, many vessels are not equipped with such systems. As a result, eye contact still plays an essential role in decision making, especially at close range.

The importance of visual contact must be taken into account while commercial, military, and recreational navigation move toward the self-sailing ships era. According to specialists, driverless technology will soon enter such major global industry. The problem is that radars and other sensors may fail to detect stealth ships and non-metallic targets, like inflatable boats, because they reflect a very low amount of radiation. As a result, computer vision techniques for estimating ships' relative location and velocity from visual clues will probably be an important component of self-sailing systems.

In this manuscript, it is presented a method to calculate the speed of vessels directly from perspective projection images acquired by conventional cameras (Figure 1.1). The method is based on projective geometry and estimates the vessel speed using Kelvin's ship-wave pattern [Maver Ksenija 2004]. The key observation is that sailing ships (even stealthy ones) leave traces of their movement in the form of wake patterns on the surface of the water. In 1887, Lord Kelvin demonstrated that speed of vessels could be estimated from the wakes left by vessels moving at a constant course and speed [Thomson 1887]. These wakes have universal characteristics, e.g., the wakes left by a boat are similar to the wakes left by a duck. The proposed method should attend the next restrictions to estimate the vessel speed:

- Be able to estimate the vessel speed from a single image taken by a conventional camera;

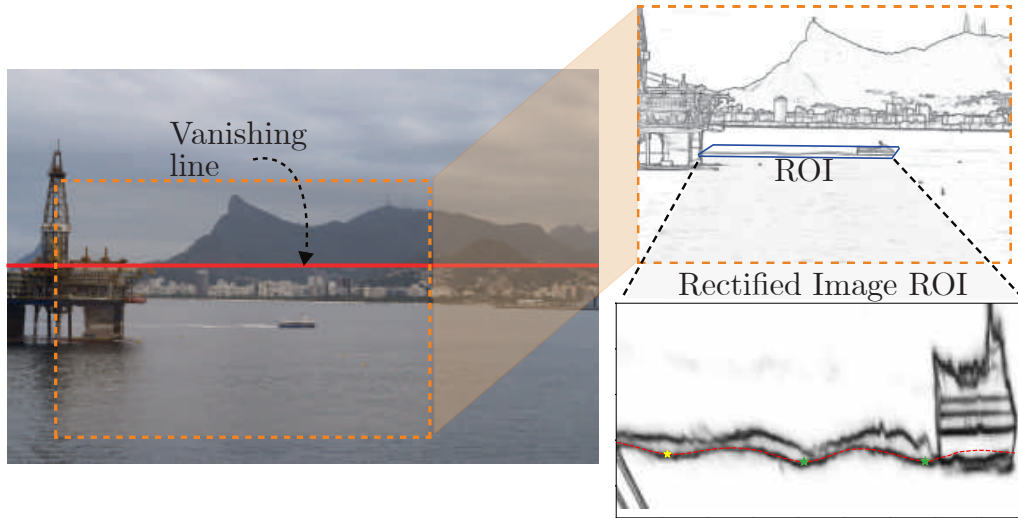


Figure 1.1: Our approach takes the cusp wave arms, performs geometric rectification of the mean water plane, and uses the distance between crests/troughs to estimate vessel speed.

- Be robust in the presence of different weather conditions and different ambient lighting;
- Be accurate and precise; and
- Do not require user intervention throughout the process.

Weather conditions and natural light are factors that hinder the process from detecting wakes and consequently estimating the vessel speed. The proposed method uses Kelvin's ship-wave pattern to estimate vessel speed. If the wakes are weak or poorly visible, it will not be easy to estimate the speed. In this work, the RCF algorithm was used to detect the edges image. RCF helps in detecting the wakes left by the vessel, making the waves more visible. Besides, we manually choose one first corner of the region of interest that contains the wake of the vessel, making the process not wholly automatic.

Concepts of projective geometry and computer vision are introduced in Chapter 2. Related works are shown in Chapter 3. The vanishing line estimation, which serves as the basis for removing perspective distortion, is introduced in Chapter 4. Manual marking was used to indicate one of the corners of the region of interest. The calculation of the remaining corners of the region of interest was described in Chapter 5. In Chapter 7, we found the wave arms present in the region of interest. In Chapter 6, we choose the wave closest to the observer and least affected by turbulence. Also, the problem of projective ambiguity is discussed. The velocity is calculated by an inverse Kelvin wave problem, once the two adjacent wave crests(or troughs) and their distance are known. Chapter 8 shows the propagation of errors along the computational chain. Chapter 9 presents the results and statistical analysis. Finally, Chapter 10 presents conclusions and future work.

## 1.1 Motivation

The advantages of the proposed approach are twofold. First, the use of single images allows the estimation of the speed of nearby vessels from each frame of a video sequence independently. Estimate velocity from motion in video sequences, on the other hand, would be a challenging task due to the natural movement of both reference and target ships as a result of wave action. Second, ship wakes are more distinct than the hulls and can be observed in optical images [Liu e Deng 2018].

Most of the works that address the problem of estimating the vessel speed use Synthetic-Aperture Radar (SAR) images [Pelich et al. 2016, Panico, Graziano e Renga 2017, Reggiannini e Bedini 2018, Carona e Marques 2014] or use some navigation systems, e.g., GPS [Yang et al. 2019], and different types of radars [Ao et al. 2018, Wang et al. 2019, Li et al. 2019]. No works were found that used a single optical image to estimate the vessel speed. An analysis of the techniques that reflect state of the art in this segment of computer vision is shown in Chapter 3.

Another motivation that was taken into account is the technological advancement regarding autonomous navigation. In recent years, projects have been seen on the creation of autonomous vessels [Hand, Bajpai], both for passenger and merchant ships. The proposed method is intended to support the operation of autonomous vessels. As our method requires only an image taken from a considerable height, there is the possibility of placing the camera in a lighthouse, and the estimated velocity serves as relevant information for the autonomous vessel.

## 1.2 Research Objective

The main objective of this research is to estimate the vessel speed from a single perspective projection image, based on the wakes left by the vessel and information on Kelvin's ship-wave pattern. The sequence of process followed to achieve this goal is shown in Figure 1.2.

### 1.2.1 Secondary Objectives

In order for the speed to be estimated in the way it is being proposed, it is necessary to solve the following problems:

- **Identify the region of interest:** The region of interest must contain the wake left by the vessel;
- **Identify the least two crests or troughs of the wave:** These crests or troughs must be present in the region of interest;

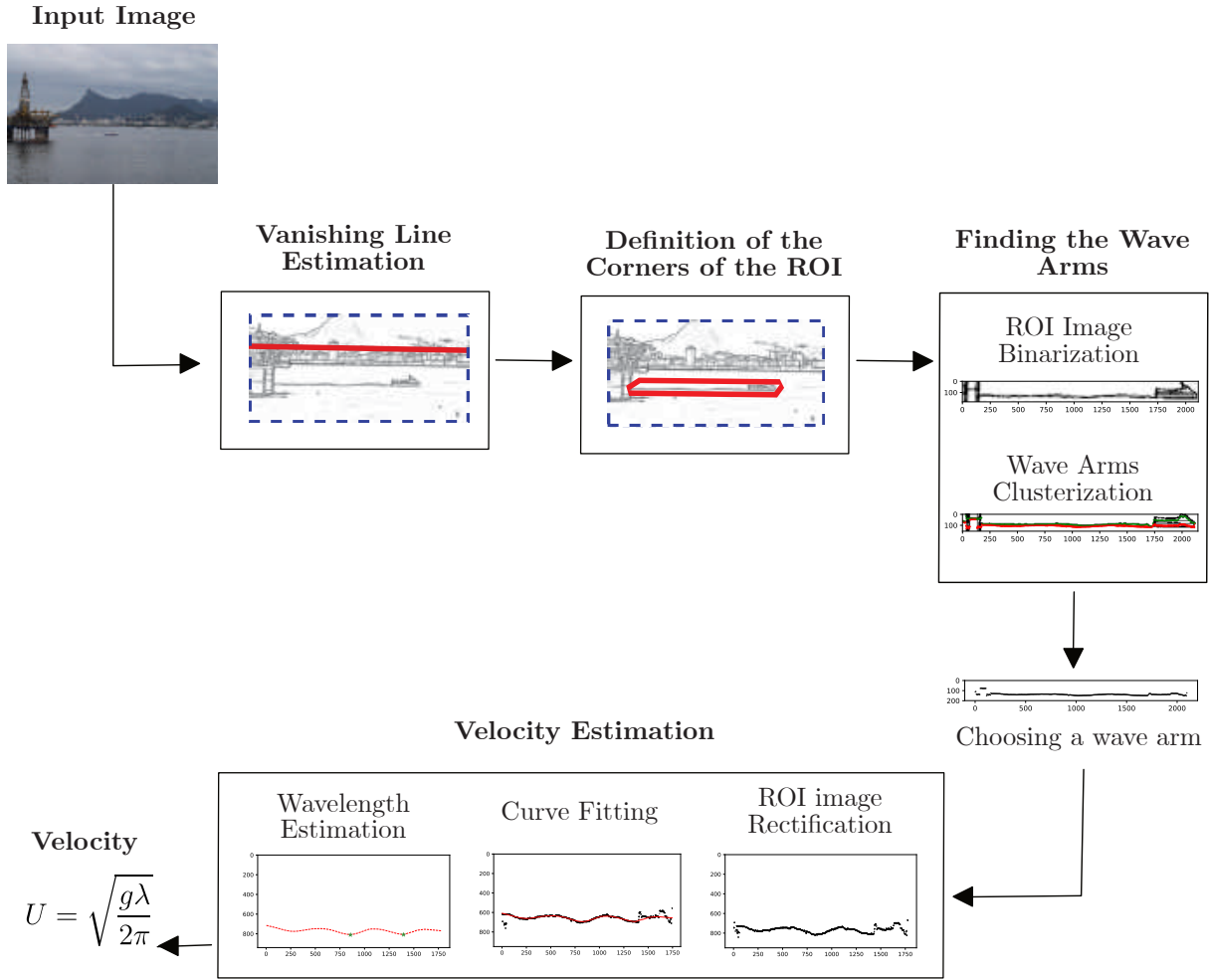


Figure 1.2: Flowchart used in this manuscript to estimate the vessel speed. Each box represents a chapter in this manuscript. The input image for the proposed method is an image containing one or more vessels, where the trace left by it is visible, and the output is the vessel speed.

- **Remove perspective distortion from images:** Only the region of interest image will be used to remove the projective distortion; and
- **Estimate the vessel speed:** Estimate speed from a single optical image.

Another challenge in this research is to identify the wavelength when another object occludes it. In order for the wave and wavelength to be found, there must be no relatively large objects on the wave. Also, finding the wave arms in lousy weather conditions and natural low light, i.e., when it is night, is still a challenge in this research.

### 1.2.2 Validation Approach

The proposed method validation was performed using a radar installed in the *Instituto de Geociências* at Universidade Federal Fluminense (UFF) to measure the vessel speed. The radar had to be manually maneuvered. The vessel images were taken manually and synchronized with the radar. For each vessel image, one note of the radar content was also taken.

Statistical experiments to estimate the reliability of the calculated speeds were described in Chapter 9. A study of the sources of error and the propagation of errors in the computational chain was also presented. In this way, it is possible to estimate the error associated with a given speed, which varies dynamically according to the capture conditions of the input images.

## 1.3 Contributions

The main contribution of this manuscript is a technique for computing vessel velocity as an inverse Kelvin wave problem using data extracted from single optical images. We have performed experiments on images of different moving vessels. The results comparing the speeds estimated by the proposed method and the speeds obtained by the radar show the precision of the proposed method. The vessels used for the experiments are passenger vessels that followed a service route through Guanabara Bay, Niteroi, Rio de Janeiro. We believe that our approach can be applied not only by self-sailing systems but also by surveillance drones and by smart lighthouses for monitoring of ship traffic.

# Chapter 2

## Background

This chapter discusses the main concepts used throughout this dissertation. Section 2.1 presents basic concepts of a digital image and a simple way to represent it mathematically. Section 2.2 shows traditional methods and the use of Convolutional Neural Networks (CNNs) to detect edge-images. Section 2.3 and 2.4 introduce concepts about the pinhole camera model and the intrinsic and extrinsic parameters of the camera. Section 2.5 introduces the main concepts of projective geometry used in this dissertation, i.e., concepts about vanishing points, vanishing lines, and homography. Finally, Section 2.6 describes some of the main concepts of the Kelvin wave structure. The Kelvin's waves model is used to estimate the vessel speed.

### 2.1 Digital Image

A digital image is a two-dimensional representation of an image from a numerical matrix (Equation 2.1). Depending on whether the image resolution is fixed, it can be vector or raster. Raster images have a finite set of digital values, called image elements or pixels [Gonzalez 2002].

Let  $f_{x,y}$  represent a continuous image function of two continuous variables,  $x$  and  $y$ . We convert this function into a digital image by sampling and quantization. Where, digitizing the coordinate values is called sampling, and digitizing the amplitude values is called quantization [Gonzalez 2002]. Suppose that we sample the continuous image into a 2D array,  $f_{x,y}$ , containing  $m$  rows and  $n$  columns, where  $x, y$  are discrete coordinates ( $x = 0, 1, 2, \dots, m - 1$  and  $y = 0, 1, 2, \dots, n - 1$ ). In



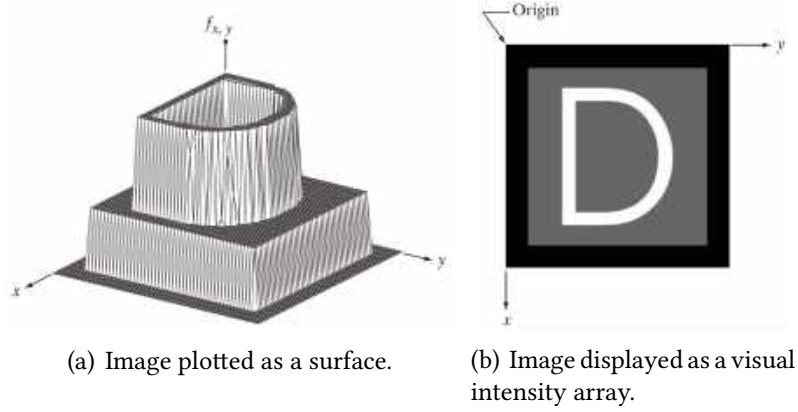


Figure 2.1: Two basic ways to represent  $f_{x,y}$ . (a) is a plot of the function, with two axes determining spatial location and the third axis being the values of  $f$  (intensities) as a function of the two spatial variables  $x$  and  $y$ . (b) It shows  $f_{x,y}$  as it would appear on a monitor or photograph. Image extracted from [Gonzalez 2002]

equation form, we write the representation of an  $m \times n$  numerical array as:

$$f_{x,y} = \begin{pmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,n-1} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,n-1} \\ \vdots & \vdots & & \vdots \\ f_{m-1,0} & f_{m-1,1} & \cdots & f_{m-1,n-1} \end{pmatrix}. \quad (2.1)$$

A digital image can also be represented as [Schalkoff 1989]:

- A grayscale image: it is a monochromatic digital image with an intensity value per pixel.
- A multispectral image: it is an image that has a vector of values in each pixel. If the image is a color image, then the vector has three elements.
- A binary image: it is a digital image where all pixel values are 0 or 1.

For the acquisition of digital images, the light source is an important parameter. The light reflects the objects and sometimes passes through the objects to create an image, this information can be captured by the human eye or by some sensor [Gonzalez 2002].

## 2.2 Edge Detection

Object boundaries often generate abrupt changes in brightness: a light object may be on a dark background, or a dark object may be on a light background [Forsyth 2013]. These sudden changes

in brightness can be quite distinctive. Points in the image where brightness changes particularly sharply are often called edges or edge points [Forsyth 2013].

Edge detection can be viewed as a method for extracting visually salient edges. It is possible to differentiate two types of edges, edges with ideal contour and edges with noisy contour [Gonzalez 2002]. An ideal contour border has two regions with different intensities on its borders, where there is an abrupt or smooth transition, but well defined. A border with a noisy contour is due to the lack of uniformity of the material. Edge detection is also used for object detection applications [Ullman e Basri 1991, Ferrari et al. 2008], and image segmentation [Arbelaez et al. 2014, Cheng et al. 2016].

Traditional methods extract local signals of brightness, color, gradient, and texture [Forsyth 2013]. Detecting changes in intensity for the purpose of finding edges can be accomplished using first- or second-order derivatives [Gonzalez 2002]. Differential operators most used for edge detection are [Forsyth 2013]:

- Square gradient: Uses first-order derivatives, it is represented as a crest of height proportional to the square of the difference in intensity at the edge, linked in Sobel operator; and
- Laplacian: Uses second-order derivatives, it is represented as two parallel crests of opposite polarities on each side of the edge, like in the Zero-crossing [Mazeika e Draudviliene 2010] method.

The variation of the intensities, using the gradient, is a reasonable approach used to locate the edges. An advantage of using the image gradient is its easy calculation, and the problem is that it increases noise.

Among the pioneer algorithms that mainly focus on the use of intensity and color gradients, we have Canny [Canny 1986]. The Canny edge detector is a multi-stage algorithm to detect a wide range of edges in images, possessing good detection, good location, and minimal response.

Although these methods are shown to be promising in some cases, these early methods are usually not accurate enough for real-life applications [Liu et al. 2019].

### 2.2.1 Deep Learning Based Algorithms

Algorithms based on deep learning have made great advances in many computer vision tasks. Recently, Xie et al. [Xie e Tu 2017] developed an efficient and accurate edge detector, Holistically Nested Edge Detection (HED), which performs training and prediction of an image to image. Xu et al. [Xu et al. 2018] introduced a deep hierarchical model to merge representations of learned edges



Figure 2.2: Some examples of RCF. The images were taken from the *Instituto de Computação* of the *Universidade Federal Fluminense*, facing the entrance of the Guanabara Bay. The four images were taken at a resolution of  $6000 \times 4000$  pixels. The two images in the first column are cropped images, having a size of  $800 \times 600$  pixels each.

at different scales solidly. These methods generally use the last convolutional layer characteristics, ignoring the intermediate convolutional layers, remembering that the intermediate convolutional layers have finer details.

Currently, Richer Convolutional Features (RCF) [Liu et al. 2019] is state of the art in edge detection. RCF proposes a solution using a CNN, inspired by the VGG-16 network [Zhang et al. 2016], with the HED architecture. RCF explores multiscale, taking the average of the results in pyramids of images. It also explores multilevel information, considering intermediate results of VGG-16 stages in data fusion. It introduces a new loss function, where they consider the edge pixels that were labeled by annotators as positive samples and the pixels that no annotators labeled as negative. They also ignore border pixels with confusing attributes labeled by annotators.

In this work, we use RCF to detect the edges of the input image. The edge image helps us in the process of wave arms detection present in the wake left by the vessel. Figure 2.2 shows some examples of RCF.

## 2.3 The Pinhole Camera Model

The pinhole camera model captures the relationship between an  $\mathbf{x}'$  point in the image with the position in the world  $\mathbf{X}'$ . The pinhole camera model aims to create a mathematical model of the image creation process [Forsyth 2013], where the image (2D plane) is created by the rays from the object to the camera's center of projection. The rays of the object pass through the hole of the pinhole camera. Figure 2.3 illustrates the pinhole camera model, where  $\mathbf{O}$  is the center of projection

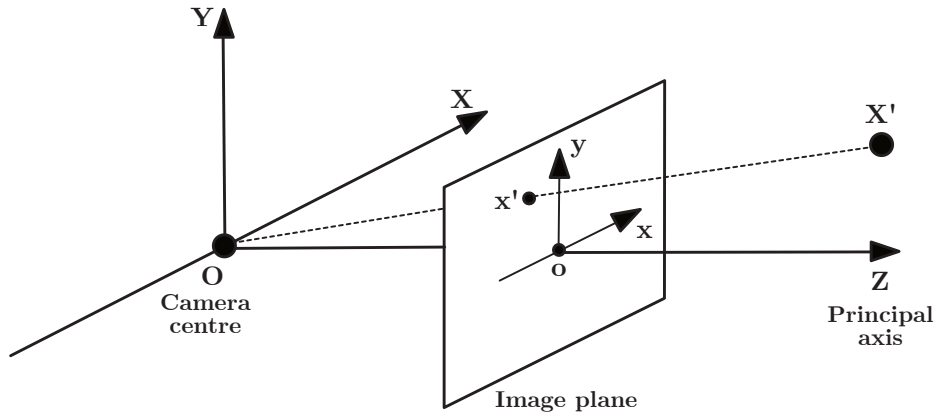


Figure 2.3: Pinhole camera model:  $\mathbf{O}$  is the camera's projection center;  $\mathbf{X}'$  is a point in 3D space, projected at point  $\mathbf{x}'$  on the image plane; and  $\mathbf{o}$  is the main point and origin of the image coordinate system. The focal length  $f$  is calculated between the distance from  $\mathbf{O}$  to  $\mathbf{o}$ .

of the camera and also the coordinate origin of the 3D space. The camera does not need to be positioned at the origin of the World Coordinate System. In the model, point  $\mathbf{X}' = (X_{X'}, Y_{X'}, Z_{X'})^T$  in 3D space is projected to point  $\mathbf{x}' = (x_{x'}, y_{x'}, z_{x'})^T$  on the image plane. Where  $\mathbf{x}'$  is calculated by the intersection of the image plane and the straight line that passes through  $\mathbf{O}$  and  $\mathbf{X}'$ . The principal point  $\mathbf{o}$  is the origin of the image coordinate system, also known as the principal point, calculated by the intersection between the  $Z$ -axis and the image plane.

Using similar triangles,  $\mathbf{x}' = (\frac{fX_{X'}}{Z_{X'}}, \frac{fY_{X'}}{Z_{X'}}, f)^T$  can be calculated, where  $f$  is the focal distance calculated by the difference of the  $Z$  coordinates between  $\mathbf{O}$  and  $\mathbf{o}$ . When the distance between the image plane and the projection center are at a finite distance, the mapping of the scene is done by perspective projection. In the perspective projection, objects farther from the distance from  $\mathbf{O}$  are seen as smaller than objects closer to  $\mathbf{O}$ , these objects being of equal size.

## 2.4 Camera Parameters

A camera is the mapping between the 3D world and a 2D image [Hartley e Zisserman 2004]. A camera has two sets of parameters, the intrinsic parameters, and the extrinsic parameters.

Intrinsic parameters are those that define the internal geometry and optics of the camera. These determine how the camera projects the points of the 3D world to the 2D image plane, being constant as long as the characteristics and relative positions between the optics and the image sensor do not

vary. These parameters make up the matrix  $\mathbf{K}$ :

$$\mathbf{K} = \begin{pmatrix} f_x & s & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.2)$$

where  $f_x = fm_x$ ,  $f_y = fm_y$ ,  $f$  is the focal length,  $m_x$  and  $m_y$  are the ratios between the sensor and the image measurements,  $s$  is the coefficient that defines the angle between the  $x$  and  $y$ -axis of the pixels,  $o_x$  and  $o_y$  are the coordinates of the principal point  $\mathbf{o}$ .

The external parameters of the camera are those that define the position and orientation of the camera reference frame with respect to the real world, that is, they give the external orientation of the camera, and such parameters are:

- Translation Vector  $\vec{\mathbf{t}}^T$ : Vector that determines the location of the optical center of the camera ( $O$ ) with respect to the axes of the real world ; and
- Rotation Matrix  $\mathbf{R}$ :  $3 \times 3$  Matrix that relates the rotation of the camera position with respect to the axis of the real world.

The following is the relationship between a 3D point  $\mathbf{X}'$  and its projection  $\mathbf{x}'$  on the image:

$$\mathbf{x}' = \mathbf{P}\mathbf{X}', \quad (2.3)$$

where  $\mathbf{P} = \mathbf{K}[\mathbf{R}|\vec{\mathbf{t}}^T]$  is the projection matrix.

In the case of real cameras approaching the pinhole model (see Figure 2.3), the intrinsic parameters can be obtained by consulting the documentation provided by the manufacturers of the camera and the lens used. If this information is not available, it is possible to use a camera calibration process.

## 2.5 Projective Geometry

Projective geometry 2D is the study of the properties of the  $\mathbb{P}^2$  projective plane that are invariable under a group of transformations known as projectivities. Projective Geometry enables the design of approaches to manipulate 2D projections of 3D objects. The projective space has more points than the Euclidean space, for a given dimension, allowing geometric transformations that take extra points, called points at infinity, into Euclidean points, and vice versa. In this mode, objects at infinity can be represented and manipulated with projective geometry [Hartley e Zisserman 2004].

In standard Euclidean space, using column vector notation, we represent a 2D point  $\mathbf{x}$  as:

$$\mathbf{x} = \begin{pmatrix} x_{\mathbf{x}} \\ y_{\mathbf{x}} \end{pmatrix}, \quad (2.4)$$

However, for homogeneous coordinates, we add an extra dimension to the vector, with an entry of 1. Also, we can multiply the entire new vector by an arbitrary scaling factor non-zero  $k$ :

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \doteq \begin{pmatrix} kx \\ ky \\ k \end{pmatrix}. \quad (2.5)$$

A point  $\mathbf{x} = (x, y, 1)^T$  lies on a line  $\mathbf{l} = (a, b, c)^T$  if and only if  $ax + by + c = 0$ . One way to represent this is by writing as an inner product of vectors  $(x, y, 1)(a, b, c)^T = (x, y, 1)\mathbf{l} = 0$ , where the points can be represented in homogeneous coordinates. We can say that the line is also in homogeneous coordinates.

Hartley and Zisserman [Hartley e Zisserman 2004] show basic properties of projective geometry on points and lines:

- The intersection of two lines  $\mathbf{l} = (a, b, c)^T$  and  $\mathbf{l}' = (a', b', c')^T$  at a point  $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$  is given by the cross-product of  $\mathbf{l}$  and  $\mathbf{l}'$ .
- The line  $\mathbf{l}$  through two points  $\mathbf{x}$  and  $\mathbf{x}'$  is given by the cross-product  $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$ .
- The basic incidence equation  $\mathbf{l}^T \mathbf{x} = 0$  for line and point is symmetric, since  $\mathbf{l}^T \mathbf{x} = 0$  implies  $\mathbf{x}\mathbf{l}^T = 0$ , in which the positions of line and point are swapped. This is known as the Duality principle.

### 2.5.1 Vanishing Points and Vanishing Lines

In perspective projection images, all parallel lines meet at a single point. This point is called the vanishing point. Vanishing points are often outside the image boundary. Given parallel lines, the vanishing points can fall to infinity. The points, in homogeneous coordinates,  $\mathbf{x}' = (x_{\mathbf{x}'}, y_{\mathbf{x}'}, w_{\mathbf{x}'})^T$  with last coordinate  $w_{\mathbf{x}'} = 0$  are known as ideal points, or points at infinity [Hartley e Zisserman 2004].

Parallel lines that are not orthogonal to the optical axis meet at a vanishing point. Two sets of parallel lines on the ground plane will give two vanishing points. The vanishing line is formed by joining these two points (Figure 2.4). The vanishing line is also represented as the intersection of two parallel planes in a plane at infinity ( $\pi_{\infty}$ ).  $\pi_{\infty}$  is a fixed plane under the projective transformation  $\mathbf{H}$ , where  $\mathbf{H}$  is considered an affinity (see Section 2.5.2).

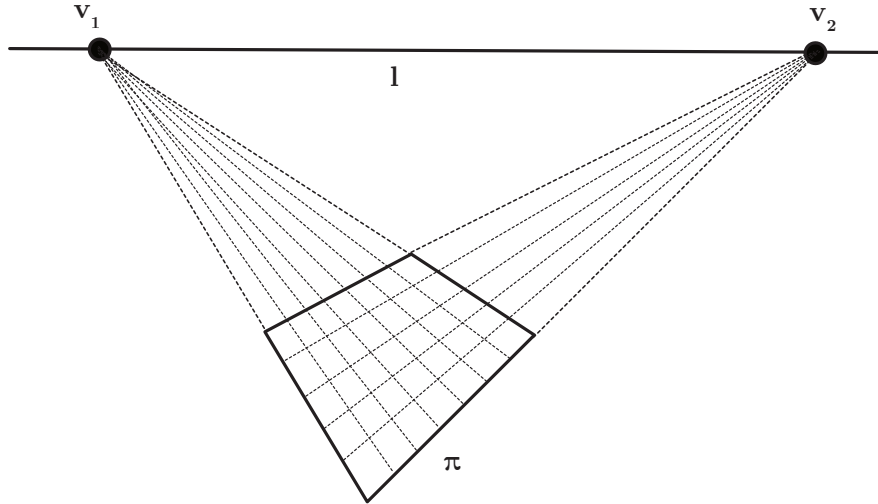


Figure 2.4: The two sets of parallel lines on the scene plane  $\pi$  converge to the vanishing points  $\mathbf{v}_1$  and  $\mathbf{v}_2$  in the image. The line  $\mathbf{l}$  through  $\mathbf{v}_1$  and  $\mathbf{v}_2$  is the vanishing line of the plane.

A scene plane's vanishing line can be used to determine information about the plane  $\pi$  only if the  $\mathbf{K}$  matrix is known [Hartley e Zisserman 2004]. Using only the vanishing line, we can know the plane's orientation to the camera and rectify the plane metrically.

One of the common problems in computer vision is identifying the vanishing line from a single image [Workman, Zhai e Jacobs 2016]. The vanishing line can be identified by knowing repeated textures of the image or determining vanishing points from parallel lines found in the image. In this work, we find the vanishing line using the Horizon Lines in the Wild (HLW) algorithm [Workman, Zhai e Jacobs 2016]. The HLW algorithm is a method that uses CNNs to estimate the location of the vanishing line from raw pixel intensities. Chapter 4 shows the vanishing line calculation using HLW.

### 2.5.2 Projective Transformations

Two images of the same planar surface are related by a homography. Homography is used for image rectification (making parallel lines parallel in images). It is possible to remove perspective distortion of a plane in a scene if we can find the vanishing line of the plane. An example of image rectification can be seen in Figure 2.5. In general, we need four points for rectifying a plane that has been distorted by a perspective projection.

Recall, a coordinate  $\mathbf{X}'$  is projected to  $\mathbf{x}'$  through the matrix  $\mathbf{P}$  (Equation 2.3), where  $\mathbf{x}'$  is a image

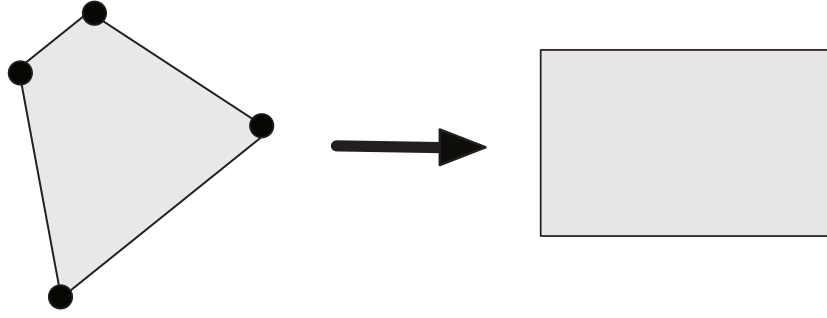


Figure 2.5: We need to identify four points on the plane such that the points are used to provide constraints on the affine transformation.

coordinate and  $\mathbf{X}'$  is a world coordinate:

$$\begin{pmatrix} x_{\mathbf{x}'} \\ y_{\mathbf{x}'} \\ w_{\mathbf{x}'} \end{pmatrix} = \mathbf{P}\mathbf{X}' = \mathbf{K}[r_1 \ r_2 \ r_3 \ \vec{\mathbf{t}}^T] \begin{pmatrix} X_{\mathbf{X}'} \\ Y_{\mathbf{X}'} \\ Z_{\mathbf{X}'} \\ 1 \end{pmatrix}, \quad (2.6)$$

where  $r_1, r_2, r_3$ , and  $r_4$  are columns of matrix  $\mathbf{R}$ . We are free to choose the coordinates of the world [Hartley e Zisserman 2004]. For an  $xy$ -plane  $Z = 0$ , therefore:

$$\mathbf{x}' = \mathbf{K}[r_1 \ r_2 \ r_3 \ \vec{\mathbf{t}}^T] \begin{pmatrix} X_{\mathbf{X}'} \\ Y_{\mathbf{X}'} \\ 0 \\ 1 \end{pmatrix}, \quad (2.7)$$

This simplifies to:

$$\mathbf{x}' = \mathbf{K}[r_1 \ r_2 \ \vec{\mathbf{t}}^T] \begin{pmatrix} X_{\mathbf{X}'} \\ Y_{\mathbf{X}'} \\ 1 \end{pmatrix}, \quad (2.8)$$

where  $\mathbf{H} = \mathbf{K}[r_1 \ r_2 \ \vec{\mathbf{t}}^T]$  is a  $3 \times 3$  matrix, and it is called the Homography Matrix.

The homography matrix has nine unknowns and is defined up to an unknown scale:

$$\begin{pmatrix} x_{\mathbf{x}'_i} \\ y_{\mathbf{x}'_i} \\ w_{\mathbf{x}'_i} \end{pmatrix} = \mathbf{H}\mathbf{X}' = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X_{\mathbf{X}'_i} \\ Y_{\mathbf{X}'_i} \\ 1 \end{pmatrix}. \quad (2.9)$$



We get:

$$x_{\mathbf{x}'_i} = \frac{h_{11}X_{\mathbf{x}'_i} + h_{12}Y_{\mathbf{x}'_i} + h_{13}}{h_{31}X_{\mathbf{x}'_i} + h_{32}Y_{\mathbf{x}'_i} + h_{33}} \quad (2.10a)$$

$$y_{\mathbf{x}'_i} = \frac{h_{21}X_{\mathbf{x}'_i} + h_{22}Y_{\mathbf{x}'_i} + h_{23}}{h_{31}X_{\mathbf{x}'_i} + h_{32}Y_{\mathbf{x}'_i} + h_{33}}, \quad (2.10b)$$

where  $x_{\mathbf{x}'_i}$ ,  $y_{\mathbf{x}'_i}$  and  $w_{\mathbf{x}'_i}$  have to be normalized by  $w_{\mathbf{x}'_i}$ .

We can write it as a system of linear equations:

$$\mathcal{A}h = \begin{pmatrix} X_{\mathbf{x}'_i} & Y_{\mathbf{x}'_i} & 1 & 0 & 0 & 0 & -x_{\mathbf{x}'_i}X_{\mathbf{x}'_i} & -x_{\mathbf{x}'_i}Y_{\mathbf{x}'_i} & -x_{\mathbf{x}'_i} \\ 0 & 0 & 0 & X_{\mathbf{x}'_i} & Y_{\mathbf{x}'_i} & 1 & -y_{\mathbf{x}'_i}X_{\mathbf{x}'_i} & -y_{\mathbf{x}'_i}Y_{\mathbf{x}'_i} & -y_{\mathbf{x}'_i} \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (2.11)$$

This has for form  $\mathcal{A}h = 0$  and the solution is the right singular vector corresponding to the smallest singular value of  $\mathcal{A}$ , i.e.  $\mathcal{A} = USV^T$ , the last column of  $V$  is equal to  $h$ .

## 2.6 Kelvin Waves

The waves pattern left by a ship in the sea consists of a wave system that wraps the hull along and it is intertwined with a cross-wave system. The two systems move forward with the vessel to remain stationary concerning it. The pattern is more impressive when viewed from a high point, but the same phenomenon on a more modest scale develops behind a duck that swims in a pond [Maver Ksenija 2004].

When a disturbance (i.e., a ship) travels on a surface of the water, it carries with it a familiar pattern of the bow and stern waves that Lord Kelvin [Ursell 1959] first explained mathematically. Instead of the ship's waves, Lord Kelvin considered the waves generated by a prescribed pressure distribution that moves with a constant velocity  $U$  and acts on the surface of the water.

The Kelvin waves structure [Thomson 1887] consists of two components: the *transverse component* develops in the same direction of vessel heading, while the *divergent component* is shown as diagonal crest moving outward from the ship's location (see Figure 2.6 and see Figure 2.7 for a 3D representation). At constant heading and speed, the wake patterns generated by vessels is confined to a V-shaped envelope whose aperture angle  $\beta_{lim} = \arcsin(1/3)$ . The aperture angle is independent of both the typology and velocity of the ship. The cusp waves, represented by  $c_n$ , can be seen as the intersection of the transverse and divergent components. More specifically, the cusp waves can be

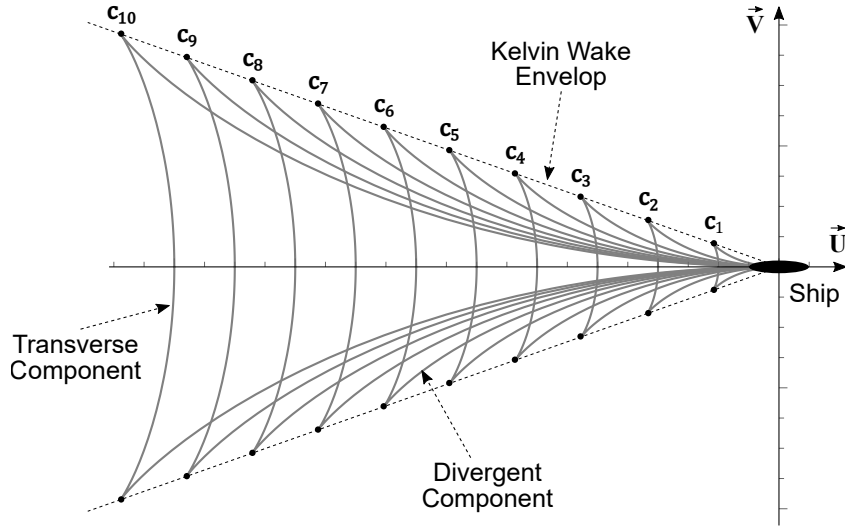


Figure 2.6: The wake crests diagram includes transverse and divergent crests. The envelop connects the cusp waves.

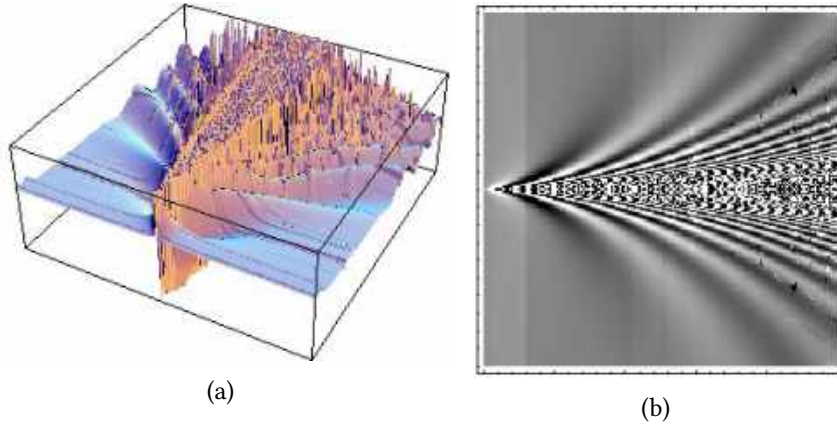


Figure 2.7: Image (a) represents image (b) in 3D space. Image (b) represents the Kelvin wave structure viewed from a processed Synthetic-Aperture Radar imagery. Images extracted from [Maver Ksenija 2004]

modeled on the mean water plane by the parametric functions:

$$u_n(p) = -\frac{1}{4} (5 \cos p - \cos 3p) n \lambda, \quad (2.12a)$$

$$v_n(p) = -\frac{1}{4} (\sin p + \sin 3p) n \lambda, \quad (2.12b)$$

where  $u_n$  and  $v_n$  are parametric functions for the coordinates plotting the transverse and divergent components of the  $n$ -th cusp wave behind the moving ship, where parameter  $p \in [-\pi/2, \pi/2]$  and the wavelength  $\lambda$  is given in meters, for  $n \in \{1, 2, \dots\}$ .

The location  $c_n = (\dot{u}_n, \dot{v}_n)$  of the cusps of the  $n$ -th wave can be computed by deriving Equation 2.12a and Equation 2.12b with respect to  $p$  and setting the derivatives equal to zero, leading

to:

$$\dot{u}_n = -\frac{4\sqrt{6}}{9}n\lambda, \quad \dot{v}_n = \pm\frac{2\sqrt{3}}{9}n\lambda. \quad (2.13)$$

The distance between successive crests or successive troughs can be used to estimate  $\lambda$ . For instance, let  $D$  be the Euclidean distance measured in meters between  $\mathbf{c}_n$  and  $\mathbf{c}_{n+1}$ :

$$D = \text{dist}_E(\mathbf{c}_n, \mathbf{c}_{n+1}). \quad (2.14)$$

After replacing the right side of Equation 2.14 with the coordinates of  $\mathbf{c}_n$  and  $\mathbf{c}_{n+1}$ , and solving for  $\lambda$ , the result is:

$$\lambda = \frac{\sqrt{3}}{2}D. \quad (2.15)$$

Wavelength  $\lambda$  and ship's speed  $U$  (in m/s) are related through:

$$U = \sqrt{\frac{g\lambda}{2\pi}}, \quad (2.16)$$

where  $g \approx 9.80665 \text{ m/s}^2$  is the acceleration of gravity. The problem can be formulated as follows: the instantaneous location of the boat is the origin of a circular wave system; this origin is in uniform rectilinear motion, its speed being the vessel speed  $U$ .

## 2.7 Discussion

The main concepts of projective geometry used in this dissertation were introduced in this chapter. A small introduction of representation and acquisition of a digital image was made. Methods for edge detection in a digital image were described. Concepts were presented on the vanishing line, the homography matrix, in the part of productive transformations. Concepts of the the Kelvin wave structure were also presented.

The definitions given in this chapter are fundamental for the calculation of the vessel speed, proposed in this work, and will be used mainly in Chapters 3, 7 and 9, where the bibliographic revision of the techniques used for the calculation of moving object speeds, where the proposed method of velocity calculation is presented, and where the analysis of the results, respectively, are shown.

For this work, we used a real camera, which is modeled as an pinhole camera. The matrix of the intrinsic parameters  $\mathbf{K}$  was filled using the specifications of the camera and the camera lens provided by the manufacturer. In this work, we did not calibrate the camera because we used the information stored in the image (e.g., focal length) to build matrix  $\mathbf{K}$ . The image information is provided by the camera information, too. We used the HLW algorithm to find the vanishing line in the input image. In addition, we used the RCF algorithm to detect edges in the input image.

# Chapter 3

## Related Work

In this chapter, an analysis of techniques used to estimate the vessel speed is carried out. The speed of an object, a vessel or a vehicle, can be calculated with the help of sensors, e.g. radars and Global Positioning System (GPS). Also, the speed can be estimated from images or videos using Computer Vision techniques.

Some devices to estimate the speed of an object are listed in Section 3.1. Section 3.2 shows examples of techniques that use images and videos to estimate the speed of objects, where the main tasks are detecting and tracking the object. Section 3.3 shows works that use radar images to estimate speed.

### 3.1 Devices Used to Estimate Speed

Speed measuring devices vary with the job. Whether measuring travel speed, wind speed, acceleration, there is a suitable instrument to calculate that form of speed. Radar and Light Detection and Ranging (LIDAR) are used to measure the speed of automobile traffic. GPS is mostly used to monitor vessels. Digital cameras, with the help of concepts of computer vision, are capable of estimating the speed of an object.

#### 3.1.1 Radar Detectors

Radar is one of the object detection systems used in aeronautics, navigation, astronomy, which serves to indicate the presence of an object and determine the distance at which it is located, by issuing unusual waves of very high frequency reflected on it. Radar can be handheld, mounted on a vehicle, or mounted on a fixed object, such as a traffic signal [Wikipedia 2020]. Its disadvantages are its high cost, lower accuracy, and radio interference.

### 3.1.2 Lidar Detector

LIDAR is a device that allows you to determine the distance from a laser emitter to an object or surface using a pulsed laser beam. The distance to the object is determined by measuring the delay time between the pulse emission and its detection through the reflected signal [Wikipedia 2019]. One problem with LIDAR sensors is that they are expensive and inaccessible to an ordinary person. Besides, the sensor must be close to the target object to estimate the speed. It also requires an environment free of precipitation.

### 3.1.3 Satellite Navigation Device

Global Positioning System (GPS) is a device that is capable of receiving information from Global Navigation Satellite Systems (GNSS) and then to calculate the device's geographical position [Wikipedia 2020]. Yang et al. [Yang et al. 2019] developed a ship speed and maneuverability measurement and analysis system (SSMMAS). The developed system SSMMAS can be installed and used directly on a computer through a USB connection together with a Differential GPS (DGPS) receiver. Where Coordinated Universal Time (UTC) time, position (latitude and longitude), heading, speed, and quality of satellites were obtained by serial communication between the DGPS and the computer with the system SSMMAS installed.

The measurement and analysis of the vessel speed are crucial for the owner of the ship, the designer, and the construction company. The distance measured by radar and fixed marked positions are the main traditional methods of speed measurement, which are primarily restricted by time and sea area. As a high-precision satellite navigation system, the GPS device has been widely applied in speed measurement [Yang et al. 2019].

### 3.1.4 Vision Based System

Measuring the speed of an object on a computer vision-based system is one of the most convenient methods available in speed detection [Wawrzyniak N. 2019]. A large number of algorithms to estimate speed use videos or image sequences, taken with a digital camera, some of the works are described in Section 3.2. Other techniques calculate the vessel speed using satellite imagery, detailed in Section 3.3.

## 3.2 Velocity Estimation Using Images and Videos

Ship detection and tracking is a primary task in any vessel traffic monitored area, whether marine or inland. It has a significant impact on navigational safety, and thus, different systems and technologies are used to determine the best possible methods of detecting and identifying sailing units [Wawrzyniak N. 2019].

Broggi et al. [Broggi et al. 2009] describes a method of detecting, measuring speed, and extracting statistics for vessels moving over a wide water surface using images stream taken from a grayscale camera. Their technique is based on the extraction of the background (sea) with the classification and vessel tracking. They demonstrated stable vessel detection even with sea waves and strong light reflections. The speed was estimated by the *pixel/frame* measurement, where this is the main difference between our proposed method. The images were taken with a high-resolution digital camera fixed to a pole at 10 meters of altitude (Burano-Venice island) and processed in real-time. The field of view distance was previously known. They emphasize that the tide-height cannot be ignored. In the experiments carried out without considering the tide-height, they obtained an error of 24%. Their system was tested with motorboat equipped with a DGPS.

Based on physical arguments, the wavelength of a moving ship is scalar at speed, given by  $\lambda = 2\pi U^2/g$ , where  $U$  is the speed, and  $g$  is the gravitational constant (this formula is used in our proposed technique). Kriebel and Seelig [Kriebel e N.S. 2001] suggest that the wave can be normalized as  $gH/U^2$ , where  $H$  is a measured wave height.  $H$  is measured using a wave sensor. Tan et al. [Tan, Piepmeier e Kriebel 2012] proposed to develop an updated equation model capable of predicting the wave heights generated by the ship, given a set of basic ship parameters. The parameters were obtained through video observations, such as the vessel length, speed, relative distance, and correlate these measurements with the records of wave height. They used the parameters obtained from the ship to validate the speed based on the wave height. They used video observations to estimate vessel speed. They mounted a digital camera at a height of 19.51 meters above sea level, and the field of view distance was calculated using Google Maps. They use image processing techniques to detect the ship and morphological operations to eliminate the traces of the vessels. They have difficulty detecting small vessels because the vessels are mistaken for the traces. For validation, they used two known ships, where the data was manually paired. Finally, they highlight the feasibility of conducting future wake studies in an efficient and low-cost manner, which can be used to better support and analyze the environmental impacts of shipping and ship traffic.

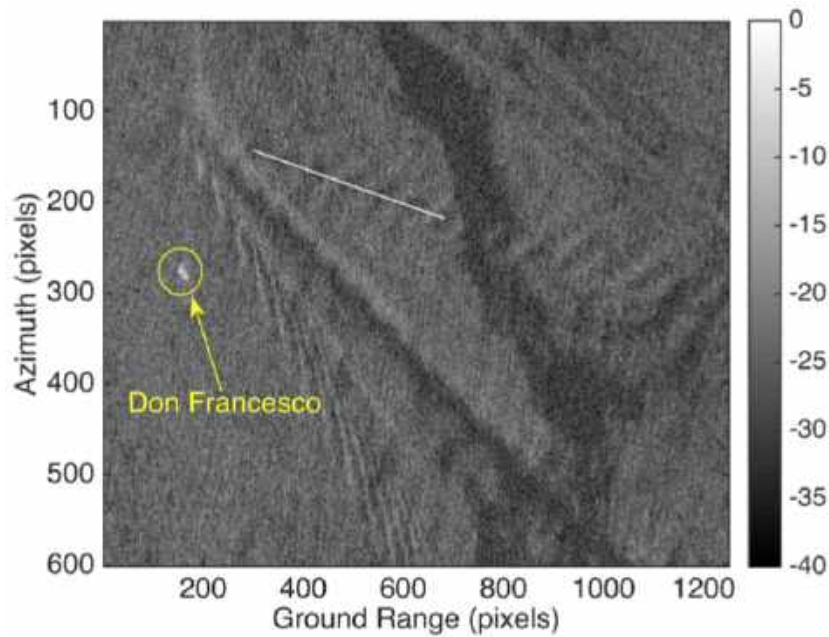


Figure 3.1: Kelvin waves pattern seen from a SAR imagery. Image extracted from [Panico, Graziano e Renga 2017]

### 3.3 Velocity Estimation Using Radar Images

Synthetic-Aperture Radar (SAR) imagery is different from how optical imaging works. While optical cameras rely on light created by other sources, like the Sun, a radar instrument actively sends its own radio waves towards its target and then measures what is reflected back. The direct radar return from a ship is the most common ship signature in SAR imagery. Pichel et al. [Pichel et al. 2004] describes five limitations of detecting vessels with SAR systems:

- **Ship characteristics:** Such as ship size and material used in the building of the ship. Identifying ships made of iron is easier than a ship made of wood;
- **Environmental conditions:** sea state, wind speed, presence of ice affect the detection ships using SAR images.
- **Radar characteristics (instrument);**
- **Image quality:** Image resolution is affected by image processing error and speckle noise; and
- **Image resolution:** Image resolution depends on the height of the radar and the characteristics of the radar.

Kirscht et al. [Kirscht 1996] presented a method for detecting and estimating the azimuth [Wikipedia 2020] velocity component of moving objects, which works on a sequence of SAR images.

They used images that show the ground at different viewing angles and moments. They estimated the velocities of the objects by evaluating the temporal correlation of successive images, where it was applied to real and simulated raw data. One of the mentioned advantages was that they could detect objects, whose range velocity component was zero. They achieved high estimation precision, even if actual clutter data overlay the objects.

Carona et al. [Carona e Marques 2014] presented a technique for detecting ships and estimating their speeds. They use Radon Transformation to detect the ship wake and evaluate the range velocity component. For detection, they used the Normalized Cross Correlation (NCC) of the split aspect. To calculate the azimuth velocity component, they used a sequence of SAR images processed with center frequency filters. They detected, in the pictures, candidates for moving objects, and a displacement vector was made for each candidate, which made it possible to estimate the speed of the ship using SAR images.

Wang et al. [Wang et al. 2019] used two satellites, TerraSAR-X (TSX) and TanDEM-X (TDX), dual-platform to obtain SAR images with a time delay to monitor maritime traffic. Wind conditions were a cause of obtaining unstable speeds. To solve this problem, they proposed an object-oriented Polarimetric Probability Ratio (PolLRT) test method based on the complex Wishart [Conradsen et al. 2003] distribution. The proposed method makes PolLRT statistics of the detected target pixels to eliminate the effect of the varied marine disorder. Two pairs of complete SAR data sets covering the Strait of Gibraltar acquired by dual-platform TSX/TDX in monostatic search mode with a time delay of approximately 10 seconds. They used the angular velocity azimuth and their range of displacement as the basis for estimating vessel speeds. Experimental results showed that the proposed PolLRT method performs better than the classical NCC method.

The use of waves left by moving objects is exploited by Kelvin wave structure, which it is a method for vessel speed estimation [Ursell 1959]. The visibility of the Kelvin wave full pattern in SAR imagery is an infrequent event. This is because of visibility results from the combination of different phenomena and observation parameters. The appearance of the Kelvin wave structure depends mostly on the relative geometry between the ship's heading and the direction of flight of the SAR sensor [Pichel et al. 2004].

Tunaley [Tunaley 2003] used high-resolution SAR images and the information of the wakes left by the ships to monitor and track ships from a satellite. He used the information from the wakes to obtain the ship's direction and orientation. Also, he used Radarsat as a source for image extraction. To estimate the vessel speed, he identified the center of the ship manually by clicking on a zoom box. The cross-range separation between the vessel location in the image and the wake provides an estimate of speed. He also mention two other techniques for estimating the vessel speed,



highlighting the use of the Kelvin wave structure.

Panico et al. [Panico, Graziano e Renga 2017] contributed in the task of recovering the vessel speed through wake analysis. They developed a method for estimating calm sea speed and applied it to seven X-band SAR images, collected by the COSMO-SkyMed mission over the Gulf of Naples, Italy. In their work, they exploit the well-known relationship between the wavelength of the waves that make up the Kelvin pattern and the vessel speed. Also, they extend the applicability of existing wake-based techniques, as they provide for wavelength evaluation along a general direction at the Kelvin angle. They achieved promising results. Furthermore, they highlight that ship/wake displacement can be used when both the ship's main feature and the turbulent wake are clearly represented, while due to the relative radar-ship geometry, cusp wave imaging is a relatively rare event. They propose to apply the Kelvin pattern in a calm sea.

### 3.4 Discussion

In this chapter, techniques to estimate the vessel speed were presented, using raster images and SAR images. Object detection and tracking techniques were used to estimate the speed in an image stream (video). The speed is calculated using the displacement distance and the time within each frame of the video. Techniques that use SAR images and radars to obtain the vessel speed were also cited. These techniques use the traces left behind by vessels as support, where concepts of azimuthal distance and Kelvin pattern are used.

In this work, we will not use image stream, SAR images, or other sensors, with the exception of a digital camera to estimate the vessel speed. In our method, we take the tide height as a source of uncertainty, as explained in Broggi et al. [Broggi et al. 2009]. The wakes left by the vessels are a source of information to efficiently estimate the vessel speed at low cost, as mentioned in Tan et al. [Tan, Piepmeier e Kriebel 2012]. Besides, in our work, we use the Kelvin wave patterns mentioned in techniques that use SAR images. In this work, only static digital images taken by a conventional camera were used.

# Chapter 4

## Vanishing Line Estimation

In this chapter, we find the vanishing line present in the input image  $\mathcal{I}$ . Basic concepts about the vanishing line and vanishing points were described in Chapter 2, Section 2.5.

Section 4.1 presents a general definition of the vanishing line. Section 4.2 shows the vanishing line detection in the input images.

### 4.1 Vanishing Line

The image location of the vanishing line is defined as the projection of the line at the infinity for any plane that shares the normal or any parallel plane, which is orthogonal to the local gravity vector. The gravity vector often coincides with the local ground plane surface normal, but not always [Workman, Zhai e Jacobs 2016].

Geometrically, the vanishing line is constructed by intersecting the image with a plane parallel to the scene plane through the center of the camera [Criminisi e Zisserman 2000], where the vanishing line depends only on the orientation of the plane, and not its position.

### 4.2 Finding the Vanishing Line

Criminisi and Zisserman [Criminisi e Zisserman 2000] show two ways to determine a vanishing line from a scene plane. The first way is to determine the vanishing points for two sets of lines parallel to the plane, and then construct the line through the two vanishing points. Another way is to determine it directly, without using vanishing points as an intermediate step, where the vanishing line can be calculated given a set of equally spaced coplanar parallel line images. The later method can be applied in images where structures, such as stairs, windows on the wall of a building, fences,

and zebra crossings, are visible.

The input images used in this work do not present the structures mentioned above, but in most of the images, the horizon can be visible. We can define the horizon as the line in which the Earth's surface and the sky seem to meet. But the problem of detecting the vanishing line is different than detecting the horizon [Workman, Zhai e Jacobs 2016]. In this work, we have tried three ways to find the vanishing line in the input image  $\mathcal{I}$ :

- Manual detection, where the user marks two points on the image to form the vanishing line. This requires that the horizon to be visible by the user.
- Semi-automatic detection, using RANSAC [Fischler e Bolles 1981], where the vanishing line is found between the intersection of the sky and the sea [Huillca e Fernandes 2019]. This method also requires that the horizon to be visible.
- Automatic detection, where the horizon does not need to be visible to find the vanishing line.

In this work, we used the third way, automatic detection, to not require the user's intervention when defining the vanishing line. The algorithm used to find the vanishing line was Horizon Line in the Wild (HLW) [Workman, Zhai e Jacobs 2016].

The HLW algorithm finds two points that form the vanishing line using Convolutional Neural Networks (CNNs), without the need to make explicit geometric assumptions about the content of the scene. The HLW algorithm calculates the endpoints of the vanishing line on the image  $\mathcal{I}$ . Each endpoints,  $\mathbf{p}_{vl}$  and  $\mathbf{p}_{vr}$ , represent the left and right points, respectively. These points are converted to homogeneous coordinates for future use:

$$\begin{aligned} \mathbf{p}_{vl} &= \begin{pmatrix} x_{\mathbf{p}_{vl}} \\ y_{\mathbf{p}_{vl}} \\ w_{\mathbf{p}_{vl}} \end{pmatrix} = \begin{pmatrix} 0 \\ v_l \\ 1 \end{pmatrix} \\ \mathbf{p}_{vr} &= \begin{pmatrix} x_{\mathbf{p}_{vr}} \\ y_{\mathbf{p}_{vr}} \\ w_{\mathbf{p}_{vr}} \end{pmatrix} = \begin{pmatrix} W - 1 \\ v_r \\ 1 \end{pmatrix} \end{aligned} \tag{4.1}$$

Equation 4.1 shows the points of the vanishing line, where the coordinates  $x_{\mathbf{p}_{vl}} = 0$  and  $x_{\mathbf{p}_{vr}} = W - 1$  are constant values, and  $W$  is the width of the input image  $\mathcal{I}$ . Thus, the  $x$ -coordinate of points  $\mathbf{p}_{vl}$  and  $\mathbf{p}_{vr}$  do not vary in this work.

HLW includes five methods to estimate the vanishing line: *so\_imagenet*, *so\_places*, *so\_posenet*, *so\_salient* and *so\_scratch*. The *so\_imagenet* method was chosen to be used in our work because this

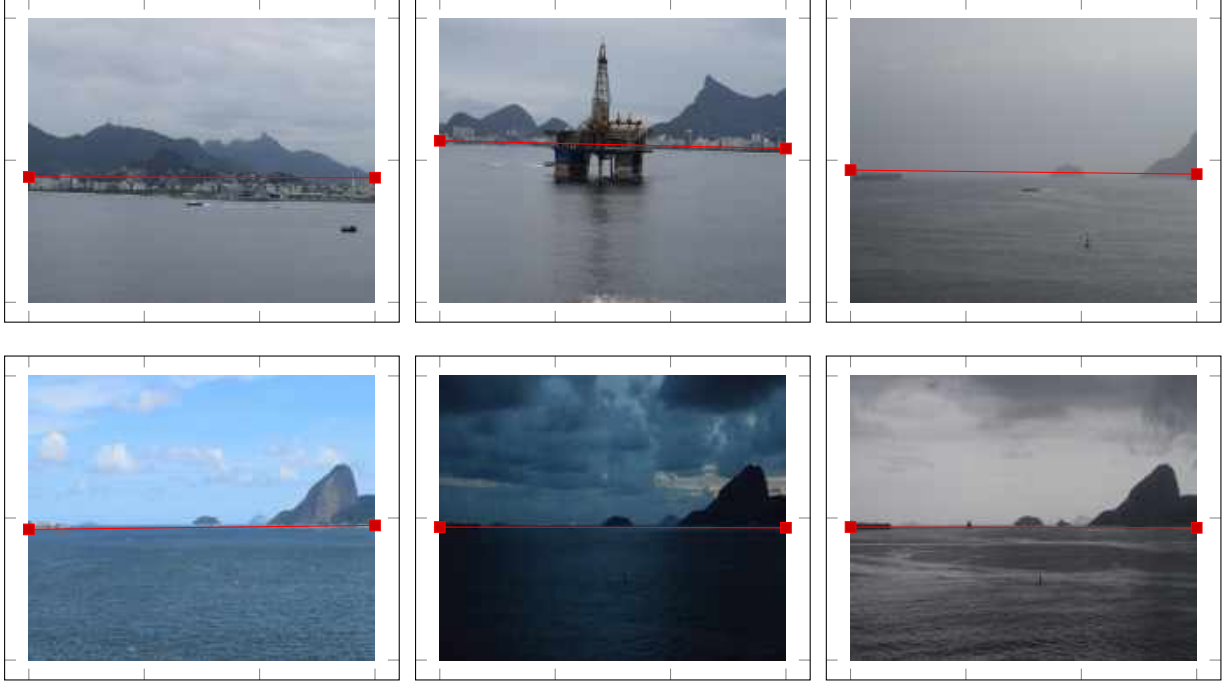


Figure 4.1: Vanishing lines found using the HLW algorithm, with the *so\_imagenet* method. The red dots in each image (left and right points) are the result of the HLW algorithm. The vanishing line is the line that passes through these two points, a red line. The images were taken in different settings, different weather conditions, and different ambient lighting. The top three images are *i1*, *i2*, and *i15*, respectively, shown in Chapter 9, Figures 9.2, and 9.4. The results are very close to the actual vanishing line.

method provides a good approximation to the actual vanishing line, as shown in Appendix A.

The vanishing line  $\mathbf{l}$  can be computed as the cross product between points  $\mathbf{p}_{vl}$  and  $\mathbf{p}_{vr}$  represented in homogeneous coordinates [Hartley e Zisserman 2004]:

$$\mathbf{l} = \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \mathbf{p}_{vl} \times \mathbf{p}_{vr} = \begin{pmatrix} v_l - v_r \\ W - 1 \\ (W - 1)v_l \end{pmatrix} \quad (4.2)$$

### 4.3 Discussion

The estimation of the vanishing line on the input image  $\mathcal{I}$  was presented in this chapter. The HLW algorithm was used to find the vanishing line in our input images. HLW has includes methods: *so\_imagenet*, *so\_places*, *so\_posenet*, *so\_salient* and *so\_scratch*. Each method was trained with a different database. For each method, the output is two coordinates that form the vanishing line. The method chosen in this work was *so\_imagenet*. Appendix A describes the methodology applied to

choose the appropriate method for the input images proposed in this work.

Recall that the HLW algorithm returns two extreme points of the image that form the vanishing line. The  $x$ -coordinates of each endpoint remains constant, where it is zero for the left, and the width of the input image  $\mathcal{I}$  for the right point. In this way, we say that the  $y$ -axis has greater participation when it comes to creating the vanishing line.

An important detail regarding the result of the two coordinates of the HLW algorithm is that the coordinate system origin was in the center of the image. Some computer vision libraries are used to work with the coordinate system origin on the upper left corner of the image. For this reason, it is important to move the coordinate system origin according to the library with which you are working.

As shown in Figure 4.1, the HLW algorithm with the *so\_imagenet* method shows acceptable results, where the found vanishing line is very close to the true vanishing line. The method proved to be robust for images with different weather conditions, different natural light, and where the horizon does not need to be visible, as shown in the images of Figure 4.1.

# Chapter 5

## Definition of the Corners of the Region of Interest

In this work, the input image  $\mathcal{I}$  was converted to an edge image  $\mathcal{B}$ . Image  $\mathcal{B}$  showed the wake of the ship in greater detail. In order to estimate the vessel speed, it was necessary to solve the following problems:

- Find the Region of Interest (ROI) that contains the cusp wave arms of the moving vessel; and
- Estimate the wavelength, which is the distance between two adjacent cusp of the wave.

This chapter describes the basic concepts of ROI. Section 5.1 defines ROI roughly. Section 5.2 calculates the four points that make up the ROI, which will help in rectifying it. The estimation of the peaks and the wavelength will be described in Chapter 7.

### 5.1 Region of Interest

In this work, ROI consists of a quadrilateral region in the input image  $\mathcal{I}$  (see Figure 1.1) corresponding to the projection of a rectangular region on the surface of the water in the 3-dimensional space. The ROI must include the cusp wave arms of the target moving vessel. We have set the ROI by manually selecting a reference corner  $\mathbf{x}_1$  and indicating the direction  $\vec{\mathbf{u}}$  of the vessel in image space. The location of the remaining ROI corners in the image is computed by the procedure described in Section 5.2.

An automatic procedure for detecting vessels in images and determining vessels' heading is out of the scope of this work. This task consists of a different research problem and can be accomplished by existing techniques, *e.g.*, [Bloisi et al. 2012, Moreira et al. 2014, Bloisi et al. 2017].

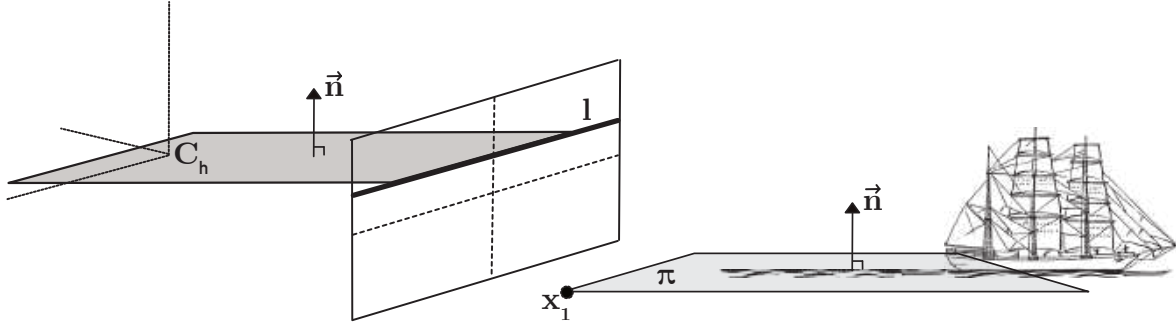


Figure 5.1: The vanishing line  $l$  of a plane  $\pi$  is obtained by intersecting the image plane with a plane through the camera center  $C_h$  and parallel to  $\pi$ . In the 3-dimensional space, the ROI lies in  $\pi$ , and  $\vec{n}$  is a normal vector. The initial coordinates of the ROI in the image, i.e., the image point  $x_1$ , is obtained manually.

## 5.2 Finding the Remaining Corners of the ROI

The variables used to find the corners of the ROI are:

- The vanishing line of the mean water plane,
- The camera height,
- The intrinsic parameters of the camera; and
- The ROI size in the 3-dimensional space.

Matrix  $\mathbf{K}$ , defined in Chapter 2, Section 2.4, is defined using the intrinsic parameters of the camera. The ROI size,  $\pi_W$  and  $\pi_H$  are defined in 3-dimensional space on the mean water plane  $\pi$ , in centimeters. Chapter 4 describes how to find the vanishing line. Section 5.2.1 describes the calculation of the camera height. Finally, Section 5.2.2 describes the calculation of the corners of the ROI in image space.

### 5.2.1 Camera Height

In this work, the camera height above the sea level is calculated as the sum of the heights  $h_g$ ,  $h_f$ ,  $h_t$ , and  $h_s$  representing, respectively, the ground height, the floor height, the tripod height, and the tide height. Therefore, the camera height is written as:

$$h = h_g + F h_f + h_t + h_s, \quad (5.1)$$

where  $F$  is the floor number where the camera was mounted.

In perfect conditions,  $h_s = 0$ . However, there are four tides per day. Therefore, one has to consider the capture and compensate for the movement of the tide to correct estimate the camera height. Since the tide height modifies the camera height above the sea level. The sources used to estimate the tide height are Nautide [Nautide], Tides Chart [Tides Chart], and Marinha do Brasil [Marinha do Brasil].

With the calculated camera height, it is possible to find the location of the camera center in the world coordinate system (see Image 5.1), defined as:

$$\mathbf{C}_h = (0, 0, h)^T \quad (5.2)$$

### 5.2.2 Remaining Corners of the ROI

Let  $\{\mathbf{x}_k\}_{k=1}^4$  be the set of corners of the ROI in the input image  $\mathcal{I}$ ,  $\mathcal{I}'$  the image of the frontal orthogonal view of the ROI on the mean water plane  $\pi$ , and  $\{\mathbf{x}'_k\}_{k=1}^4$  the set of corners of  $\mathcal{I}'$ . The ROI image rectification process is explained in Chapter 7, section 7.1.

Our algorithm makes  $\mathcal{I}'$  isotropic to the ROI lying in  $\pi$ . We ensure isotropy by computing  $\mathbf{x}_2$ ,  $\mathbf{x}_3$ , and  $\mathbf{x}_4$  from the reference corner  $\mathbf{x}_1 = (x_1, y_1, 1)^T$ , the direction  $\vec{\mathbf{u}} = (x_{\vec{\mathbf{u}}}, y_{\vec{\mathbf{u}}}, 0)^T$  of the vessel, the expected size of the ROI in 3-D space ( $\pi_W \times \pi_H$  centimeters), the camera height  $h$ , and the coefficients of the general equation of the vanishing line  $\mathbf{l} = (A, B, C)^T$  (Equation 4.2) of the water body. We set the size of  $\mathcal{I}'$  to  $\pi'_w \times \pi'_h$  pixels, where  $\pi'_w = \lfloor \alpha \pi_W \rfloor$  and  $\pi'_h = \lfloor \alpha \pi_H \rfloor$ , for  $\alpha$  being a scaling factor for converting centimeters to pixel units.

Let the origin of the 3-dimensional space be defined on the mean water plane  $\pi$  as the orthogonal projection of the camera center  $\mathbf{C}_h = (0, 0, h)^T$  (Equation 5.2) on  $\pi$ , where  $\mathbf{C}_h$  is part of the external camera parameters. The vector normal to  $\pi$  is aligned to the  $\vec{\mathbf{Z}}$ -axis of the world, which extends vertically upward. Also, let  $\{\mathbf{X}_k\}_{k=1}^4$  be the set of corners of the ROI lying on plane  $\pi$ . The corner  $\mathbf{X}_1 = (X_1, Y_1, 0, 1)^T$  is computed as the intersection between  $\pi$  and the ray back-projected from point  $\mathbf{x}_1$ :

$$\mathbf{X}_1 = \begin{pmatrix} -hx_{\vec{\mathbf{D}}}/z_{\vec{\mathbf{D}}} \\ -hy_{\vec{\mathbf{D}}}/z_{\vec{\mathbf{D}}} \\ 0 \\ 1 \end{pmatrix}, \quad (5.3)$$

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 ray to the world origin from the camera centre,  $\mathbf{M} = \mathbf{K}\mathbf{R}$ ,  $\mathbf{K}$  is the matrix of intrinsic parameters, and  $\mathbf{R}$  is a rotation matrix.  $\mathbf{R}$  gives us the camera orientation, which is part of the external camera parameters whose columns are  $\vec{\mathbf{R}}_1 = \vec{\mathbf{R}}_3 \times \vec{\mathbf{R}}_2$ ,



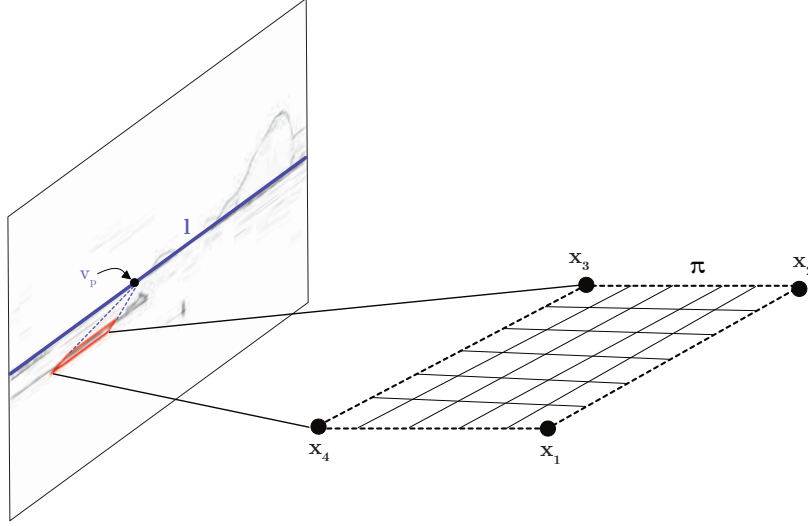


Figure 5.2: The image shows the positions of the four corners  $\{\mathbf{x}_k\}_{k=1}^4$  of the ROI on the plane  $\pi$ . Where  $\mathbf{l}$  is the vanishing line, and  $\mathbf{v}_p$  is a vanishing point.

$\vec{\mathbf{R}}_2 = \text{unit}((1, 0, 0)^T \times \vec{\mathbf{R}}_3)$ , and  $\vec{\mathbf{R}}_3 = \text{unit}(\text{up}(\mathbf{K}^T \mathbf{l}))$ , respectively. The unit function normalizes the vector to have unit length and up function forces the vector to point upward. This causes the initial  $\mathbf{x}_1$  coordinate to be positioned at the bottom of the  $\mathbf{x}_4$  coordinate, in image coordinates, according to Figure 5.2.

$\mathbf{X}_2$ ,  $\mathbf{X}_3$ , and  $\mathbf{X}_4$  were computed by translating  $\mathbf{X}_1$  by  $\pi_W$  and  $\pi_H$ . These points are located in the world coordinate system, and defined as:

$$\begin{aligned}\mathbf{X}_2 &= (x_{\mathbf{X}_1} + \pi_W, y_{\mathbf{X}_1}, z_{\mathbf{X}_1}, 1)^T \\ \mathbf{X}_3 &= (x_{\mathbf{X}_1} + \pi_W, y_{\mathbf{X}_1} + \pi_H, z_{\mathbf{X}_1}, 1)^T \\ \mathbf{X}_4 &= (x_{\mathbf{X}_1}, y_{\mathbf{X}_1} + \pi_H, z_{\mathbf{X}_1}, 1)^T\end{aligned}\tag{5.4}$$

Finally, the corners of the ROI in  $\mathcal{I}$  are computed as (see Figure 5.2):

$$\mathbf{x}_k = \mathbf{P}\mathbf{X}_k,\tag{5.5}$$

where  $\mathbf{P} = \mathbf{M}[\mathbf{I} | -C_h]$  is the affine camera matrix (see Equation 2.3).

### 5.3 Discussion

The calculation of the points that make up the ROI was presented in this chapter. Finding the ROI is an essential step in estimating vessel speed. The ROI contains the wake of the vessel. In Chapter 6, a wave inside the ROI will be chosen to determine the wavelength. ROI rectification is described in Chapter 7.

The calculation of the points that form the ROI was on the input image  $\mathcal{I}$ . In our experiments, initial coordinate  $\mathbf{x}_1$  is obtained manually. The position of  $\mathbf{x}_1$ , along with the other coordinates, must cover a large part of the trails left by the vessels. Usually,  $\mathbf{x}_1$  is located close to the stern of the boat, because the traces of the wake are more reliable. The vanishing line and the camera height are used as input variables for calculating the corners of the ROI.

A tape measure was used to calculate the floor height, the tripod height, and to measure the ground height we use Google maps. The tide height used three sources, taking *Marinha do Brasil* [Marinha do Brasil] as the primary source. In an ideal scenario, the height of the tide would be equal to zero, and the traces of the ships would not be modified by the wind or weather conditions.

# Chapter 6

## Finding the Wave Arms

In this chapter, the non-rectified Region of Interest (ROI) image is used to find the wave arms. We utilize the edge image  $\mathcal{B}$  for this purpose. We create a new image for the ROI, based on the set of corners  $\{\mathbf{x}_k\}_{k=1}^4$ . The new image of the ROI will have a rectangular shape that facilitates the detection of the wave arms.

In section 6.1, we define the new set of corners of the ROI's bounding box. In Section 6.2, we binarize the image present in the ROI's bounding box of the edge-image  $\mathcal{B}$ . Finally, in Section 6.3 we find the wave arms.

### 6.1 The ROI's Bounding Box

Let  $\{\mathbf{x}_k^*\}_{k=1}^4$  be the set of corners of the ROI's bounding box, they are calculated based on the set of corners  $\{\mathbf{x}_k\}_{k=1}^4$  of the ROI. The corners of the ROI are calculated in Chapter 5. The corners of the ROI form a quadrilateral, which will be rectified in Chapter 7. The new image, created to facilitate the detection of the wave arms, will be inside of the ROI's bounding box.

It is only necessary to calculate two diagonal points,  $\mathbf{x}_{lower}^*$  and  $\mathbf{x}_{upper}^*$ , of the quadrilateral to find the four corners of the ROI's bounding box. These points are represented as two non-contiguous vertices of the quadrilateral:

$$\begin{aligned}\mathbf{x}_{lower}^* &= (x_{lower}, y_{lower})^T \\ \mathbf{x}_{upper}^* &= (x_{upper}, y_{upper})^T,\end{aligned}\tag{6.1}$$

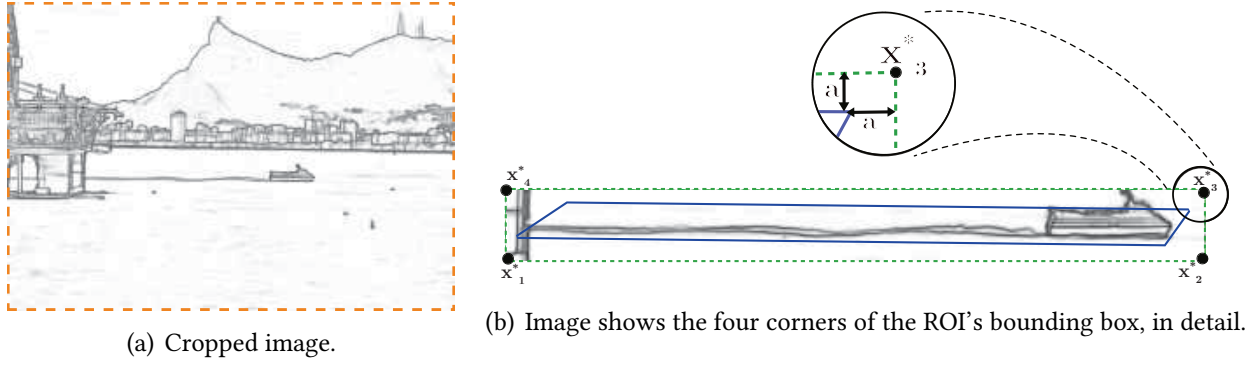


Figure 6.1: Representing the ROI's bounding box. Image (a) is a cropped image to visualize the ROI and the ROI's bounding box, in blue line and green dotted line, respectively. Image (b) shows the four corners  $\{\mathbf{x}_k^*\}_{k=1}^4$  of the ROI's bounding box in detail, where  $a \geq 0$  is a constant value to expand the ROI's bounding box.

where  $x_{lower}$ ,  $y_{lower}$ ,  $x_{upper}$ , and  $y_{upper}$  are shown in the next equation:

$$\begin{aligned}
 x_{lower} &= \min(x_{\mathbf{x}_1}, x_{\mathbf{x}_2}, x_{\mathbf{x}_3}, x_{\mathbf{x}_4}) - a \\
 y_{lower} &= \min(y_{\mathbf{x}_1}, y_{\mathbf{x}_2}, y_{\mathbf{x}_3}, y_{\mathbf{x}_4}) - a \\
 x_{upper} &= \max(x_{\mathbf{x}_1}, x_{\mathbf{x}_2}, x_{\mathbf{x}_3}, x_{\mathbf{x}_4}) + a \\
 y_{upper} &= \max(y_{\mathbf{x}_1}, y_{\mathbf{x}_2}, y_{\mathbf{x}_3}, y_{\mathbf{x}_4}) + a,
 \end{aligned} \tag{6.2}$$

and  $a \geq 0$  is a constant value used to extend the ROI's bounding box.

The right image in Figure 6.1 shows the corners of the ROI's bounding box in greater detail. The ROI's bounding box covers ROI, having more information about the wave. It should be noted that the corner set of the ROI's bounding box does not replace the corner set of the ROI at the time of rectification. The ROI's bounding box is used in this manuscript only to work in a friendlier way when finding the wave arms.

## 6.2 ROI Image Binarization

The first step to find the wave arms is to remove the noise that is next to the waves and is presented as non-edge. The image into of the ROI's bounding box is a part of the edge image  $\mathcal{B}$ , where the edge image  $\mathcal{B}$  is a grayscale image. The waves are subject to noise. This noise is seen as weak pixels and, in some cases, almost indistinguishable, i.e., pixels with values practically close to 255, where 255 represents the white intensity. In order to find the correct wave arm properly, it is necessary to eliminate noise.

In this work, the Otsu algorithm [Otsu 1979] was used to find an automatic threshold  $t$  to se-

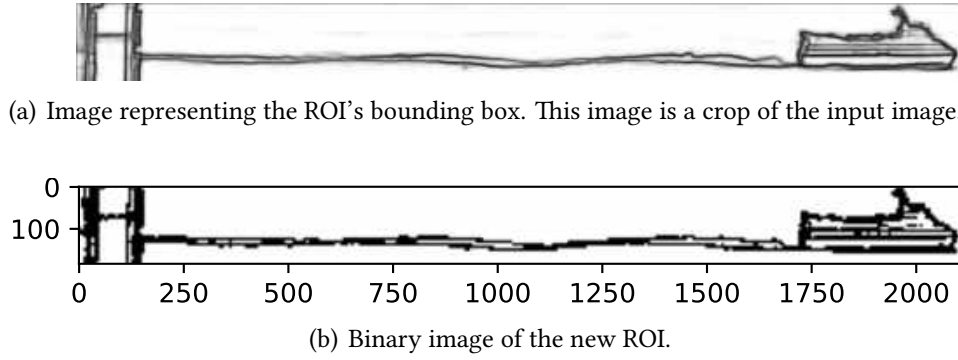


Figure 6.2: Image binarization of the ROI's bounding box using a threshold  $t$  calculated by the Otsu algorithm [Otsu 1979].

parates the pixels into two classes, edge and non-edge. The threshold  $t$  is calculated only for the image that is inside of the ROI's bounding box. With the calculated threshold, we eliminate the pixels whose intensities are greater than the threshold  $t$  to obtain the binarized image  $\mathcal{B}^*$ , as shown in the Equation 6.3:

$$\mathcal{B}_{i,j}^* = \begin{cases} 0 & , \mathcal{B}_{[i+x_{lower}], [j+y_{lower}]} < t \\ 255 & \text{otherwise} \end{cases}, \quad (6.3)$$

where image  $\mathcal{B}_{i,j}^*$  is treated as a matrix, and  $\mathcal{B}_{[i+x_{lower}], [j+y_{lower}]}$  represents the value of a pixel in image  $\mathcal{B}$ . The  $i$  and  $j$  indices are used to loop through the matrix. Figure 6.2 shows the effect of image binarization within ROI's bounding box.

### 6.3 Wave Arms Clusterization

The binary image  $\mathcal{B}^*$  within of the ROI's bounding box was calculated in the previous step. The image  $\mathcal{B}^*$  is used to find the wave arms. In this work, we assume that there are two waves in the binary image, one upper and one bottom wave.

We use the k-means algorithm [Jin e Han 2010] to differentiate the two wave arms present in the wake left by the ship. The k-means algorithm is a grouping method that iteratively assigns data points to k-groups. This algorithm begins with k-seeds that are adjusted to the minimums of a cost function. In our case, the cost function is going to be the default cost function of k-means, where a Euclidean distance will be used between the coordinates of the pixels, regardless of their intensity. We use k-means to group the coordinates of the binary image into two groups. The image coordinates where the pixel value was equal to zero were taken as input to the algorithm. The k-means algorithm is not applied to the entire image directly. We divided the  $\mathcal{B}^*$  image into  $q$  windows vertically, each window having width of  $d$  pixels, and this window slides every  $\alpha$  pixel, as

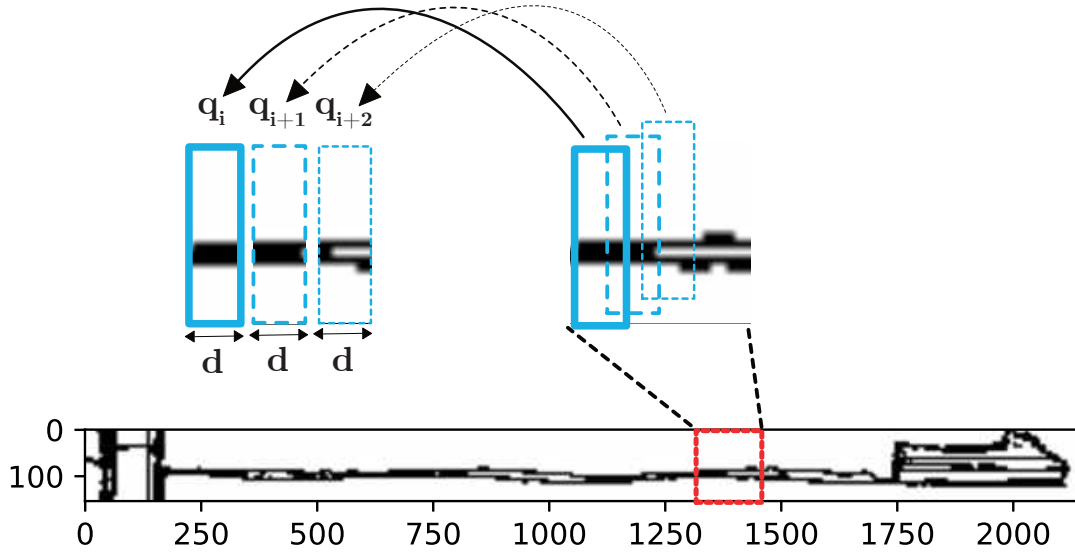


Figure 6.3: The image shows the windows used where the k-means algorithm was applied. Each  $q_i$  window has a width of  $d$  pixels and a height equal to that of the new ROI. Each window slides by  $\alpha$  pixels, for  $\alpha \geq 1$ .

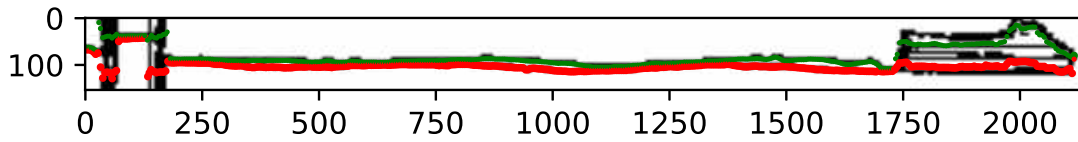


Figure 6.4: Image shows the waves separated into two groups, using k-means, an upper wave and a bottom wave, depicted with green and red colors, respectively.

illustrated in Figure 6.3. In this work,  $d$  and  $\alpha$  are equal to 5 pixels. In each  $q_i$  window, we applied the k-means algorithm, and we obtained their centroids. The two seeds in window  $q_i$  are initially taken as random to initialize the k-means algorithm. Subsequently, the seeds in window  $q_{i+1}$  are equal to the two centroids calculated in window  $q_i$ . If no centroids were found in window  $q_i$ , the two seeds are obtained randomly.

In this work, we use the centroids of each group to obtain the wave curves. In this way, we will have two wave curves. Each wave is formed by a pixel obtained from the centroid. These centroids are differentiated by color in Figure 6.4, where the upper centroids are green and the bottom centroids are red. In this work, we use only one wave arm to estimate the wavelength and consequently, to estimate the vessel speed. Chapter 7 describes the criteria for choosing the appropriate wave.

## 6.4 Discussion

In this chapter, we found the two wave arms. The waves will be transformed according to ROI rectification in Chapter 7.

The threshold  $t$ , computed by Otsu algorithm, was assumed as a correct threshold to binarize the image. When the image was binarized, it was observed that the waves have a thickness. Centroids were used to find the waves correctly and try to solve the wave thickness problem, and in this way obtain the skeleton of the wave. The k-means algorithm was used to differentiate the two wave arms. In this work, noise related to objects that occlude the wave traces was not analyzed. The existence of higher noise occurs when the traces of the vessels are not very visible.

We divide the binarized image within of the ROI's bounding box into several small pieces to calculate the centroids of the wave arms within each window. Each window was independent, which facilitates parallelization. Every window slides  $\alpha > 0$  pixels, following a direction from left to right. The assumed  $\alpha$  value must be a small value and less than the window size to obtain the most significant number of wave centroids. In this work,  $\alpha = 5$  was empirically chosen. The more centroids are found, better the wave is represented.

# Chapter 7

## Vessel Velocity Estimation

In this chapter, the wavelength is used as the basis for estimating the vessel speed. The wavelength is extracted from the rectified Region of Interest (ROI) image. Here, it is assumed that the ROI contains only the wave arms. The wave arms are found using the procedure presented in Chapter 6, where the wave arms image replaced the ROI content. In this way, only the wave arms were rectified, which facilitates the estimation of the wavelength. Vessel speed is estimated using the Kelvin wave structure equations, described in Chapter 2, Section 2.6.

We chose only one wave arm present in the region of interest, represented as the dotted red line in Figure 7.1, to image rectify. Section 7.1 describes the image rectification. The rectified image in Figure 7.2 (b) is relatively similar to the red dotted lines in Figure 7.1 (a), (b), and (c). After rectifying the chosen wave arm, it was necessary to smooth and adjust the curve to more easily find the crests or troughs of the wave. Section 7.2 describes the curve fitting. The crests or troughs can be used to calculate the wavelength. In this work, we only use the troughs (shown as green stars in Figure 7.1) as they were the least affected by the turbulence generated by the ship. The calculation of the wavelength is described in section 7.3. The wavelength was used to estimate the vessel speed, described in Section 7.4. The vessel speed is found using the inverse problem of Kelvin waves.

### 7.1 ROI Image Rectification

Recall that the process to find the wave arms were performed on the ROI edge image. The edge image  $\mathcal{B}$  was found with the Richer Convolutional Features (RCF) [Liu et al. 2019] algorithm. RCF was used on the input image  $\mathcal{I}$ . In this way, the set of corners of the ROI, calculated in Chapter 5, can be used in both the edge image  $\mathcal{B}$  and the input image  $\mathcal{I}$ .

Let  $\{\mathbf{x}_k\}_{k=1}^4$  be the set of corners of the ROI in the edge image  $\mathcal{B}$ ,  $\mathcal{B}'$  the image of the frontal



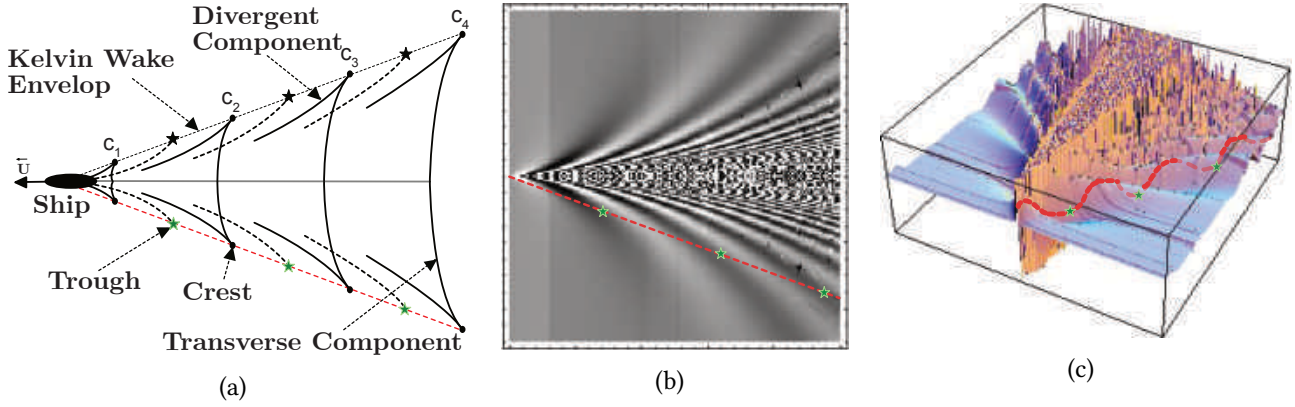


Figure 7.1: Image (a) shows the Kelvin waves' structure, indicating the transverse and divergent components, along with the crests and troughs of the wave arm. Image (b) represents the Kelvin wave structure viewed from a processed Synthetic-Aperture Radar imagery. Image (a) represents image (b) in 3D space. The dotted red line represents the chosen wave arm. Green stars are the troughs used to find the wavelength. Images (a) and (b) were extracted from [Maver Ksenija 2004]

orthogonal view of the ROI on the mean water plane  $\pi$ , and  $\{\mathbf{x}'_k\}_{k=1}^4$  the set of corners of  $\mathcal{B}'$ . The rectification consists on mapping the ROI from  $\mathcal{B}$  to  $\mathcal{B}'$  in order to simulate the aerial view having camera's optical axis perpendicular to the mean water plane. We use the point-to-point correspondences  $\{\mathbf{x}_k \leftrightarrow \mathbf{x}'_k\}$  and the Direct Linear Transformation (DLT) [Hartley e Zisserman 2004] to compute the 2-D homography that performs such mapping.

A projective transformation matrix  $\mathcal{M}$  is used to rectify the ROI image. The matrix  $\mathcal{M}$  is described in the next section. Figure 7.2 (a) shows the chosen wave arm. We chose the wave arm closest to the observer because it was the least affected by the turbulence. Figure 7.2 (b) shows the rectified image of Figure 7.2 (a) using matrix  $\mathcal{M}$ .

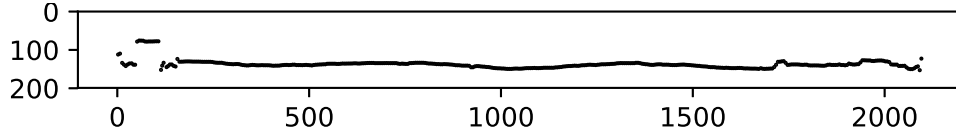
### 7.1.1 Findig the Homographic Matrix

In projective geometry, a homography is an isomorphism of projective spaces, induced by an isomorphism of the vector spaces from which the projective spaces derive [Artin 1957]. In this work, homography is used to transfer the points from image  $\mathcal{B}^*$  to image  $\mathcal{B}'$  through the plane  $\pi$ . The homography  $\mathbf{H}$  of  $\mathbf{x}^*$  to  $\mathbf{x}'$  is written as:

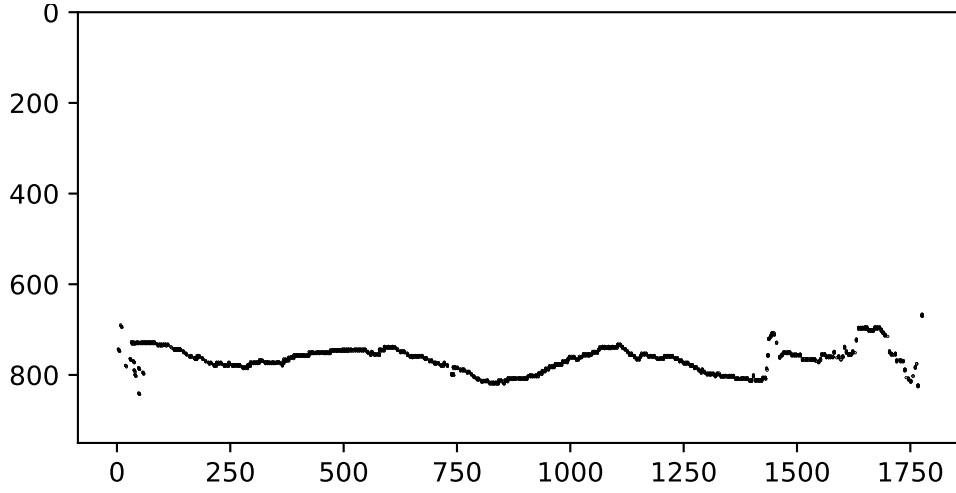
$$\mathbf{x}^* = \mathbf{H}\mathbf{x}', \quad (7.1)$$

where  $\mathbf{x}^*$  is an image  $\mathcal{B}^*$  point, and  $\mathbf{x}'$  is an image  $\mathcal{B}'$  point.

The (projective) matrix  $\mathbf{H}$  which achieves this transformation can be applied to every point in the image in order to affinely rectify the image, i.e. after the transformation, affine measurements



(a) The ROI image showing the closest wave.



(b) The rectified ROI image showing only the closest wave. Axes in centimeters.

Figure 7.2: Image (a) shows one of the waves (the closest wave) found in Chapter 6. Image (b) shows the rectified ROI image showing only the closest wave, where the ROI size is in centimeters.

can be made directly from the rectified image [Hartley e Zisserman 2004].

The objective of image rectifying is to eliminate the projective distortion present in the plane image. Projective distortion can be removed by specifying four reference points (8 degrees of freedom) on the plane and calculating the transformation that assigns the reference points to its images. In a projective transformation, it is only necessary to specify 4 degrees of freedom to determine metric properties. These degrees of freedom (dof) are associated with the line at infinity  $\mathbf{l}_\infty$  (2 dof), and two circular points on  $\mathbf{l}_\infty$  (2 dof), where  $\mathbf{l}_\infty$  removes projective distortion, and the circular points on  $\mathbf{l}_\infty$  remove affine distortion. The line at infinity  $\mathbf{l}_\infty$  allows recovering related properties of the images, i.e., parallelism, the proportion of areas [Hartley e Zisserman 2004]. In this work, we used affine transformation. Also, it was only necessary to specify three reference points on the plane (6 degrees of freedom).

Recall, that when a point  $\mathbf{x}$  lies on a line  $\mathbf{l}$ , the transformed point  $\mathbf{x}' = \mathbf{H}\mathbf{x}$  under a projective transformation lies on  $\mathbf{l}' = \mathbf{H}^{-T}\mathbf{l}$  [Hartley e Zisserman 2004], where  $\mathbf{H}$  is an homography. Following the same analogy, we can represent  $\mathbf{l}'_\infty = \mathbf{H}_A^{-T}\mathbf{l}_\infty$ , where  $\mathbf{H}_A$  is an affinity. To facilitate the process, the canonical equation of  $\mathbf{l}_\infty = (0, 0, 1)^T$  is used. Taking  $\mathbf{l} = (A, B, C)^T$ , calculated in Chapter 4, as an imagined line at infinity ( $\mathbf{l}'_\infty$ ), for  $C \neq 0$  will provide a projective transformation

that maps  $\mathbf{l}$  back to  $\mathbf{l}_\infty$ :

$$\mathbf{H} = \mathbf{H}_A \mathbf{H}_1 = \mathbf{H}_A \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ A & B & C \end{pmatrix}, \quad (7.2)$$

where the multiplication of  $\mathbf{H}_A$  and  $\mathbf{H}_1$  will create an affine transformation matrix with the last row equal to  $\mathbf{l}^T$ . The  $\mathbf{H}_1$  matrix is used to offset the affine matrix  $\mathbf{H}_A$ . Using the matrix  $\mathbf{H}$ , we find a (projective) matrix that accurately rectifies each point on the mean water plane. With the rectified image, we will be able to perform affine measurements.

The  $\mathbf{H}_A$  matrix is an affine transformation matrix, written as:

$$\mathbf{H}_A = \begin{pmatrix} \mathbf{A} & t \\ 0^T & 1 \end{pmatrix} = \begin{pmatrix} v_{h1} & v_{h2} & v_{h3} \\ v_{h4} & v_{h5} & v_{h6} \\ 0 & 0 & v_{h7} \end{pmatrix}, \quad (7.3)$$

where  $\mathbf{A}$  is a  $2 \times 2$  non-singular matrix, and  $t = (t_x, t_y)^T$  is a translation vector. A planar affine transformation has six degrees of freedom corresponding to the six matrix elements. Such transformation can be computed from three point correspondences defined as:

$$\mathbf{x}'_k = \begin{pmatrix} x_{\mathbf{x}'_k} \\ y_{\mathbf{x}'_k} \\ w_{\mathbf{x}'_k} \end{pmatrix} = \mathbf{H}_1 \mathbf{x}_k, \quad (7.4)$$

where  $k \in 1, 2, 3$ .

The new coordinates have to be transformed to a homogeneous coordinate system. Equation 7.5 shows its calculation:

$$\mathbf{x}'_k \doteq \begin{pmatrix} x_{\mathbf{x}'_k} / w_{\mathbf{x}'_k} \\ y_{\mathbf{x}'_k} / w_{\mathbf{x}'_k} \\ 1 \end{pmatrix}, \quad (7.5)$$

where  $k \in 1, 2, 3$ . The new coordinates  $\mathbf{x}'_k$  are the corners of the ROI on the water plane that will help us get affinity. We create a linear system of equation  $\mathcal{A} v_h = 0$  to obtain the affine transformation matrix, where  $\mathcal{A}$  is an  $m \times n$  matrix of known values. The system of linear equations (Equation 7.6) is used to calculate the affine transformation matrix  $\mathbf{H}_A$ .

Equation 7.6 shows the matrix  $\mathcal{A}$  using the coordinates  $(\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3)$ , where  $\pi_W$  and  $\pi_H$  are the width and height of the ROI, respectively. The ROI size is in centimeters.

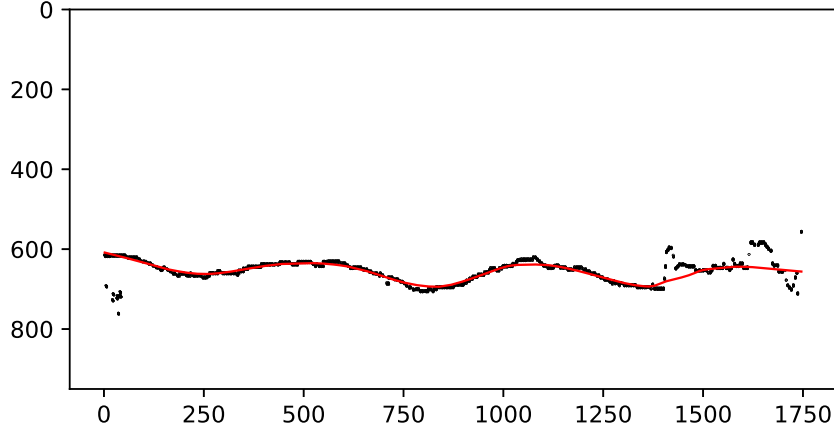


Figure 7.3: The black points correspond to the digital curve in this example, while the red line is the resulting smooth curve using LOWESS algorithm [Wilcox 2017]. Axes in centimeters.

$$\mathcal{A} v_h = \begin{pmatrix} x_{\mathbf{x}'_1} & y_{\mathbf{x}'_1} & w_{\mathbf{x}'_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{\mathbf{x}'_1} & y_{\mathbf{x}'_1} & w_{\mathbf{x}'_1} & 0 \\ x_{\mathbf{x}'_2} & y_{\mathbf{x}'_2} & w_{\mathbf{x}'_2} & 0 & 0 & 0 & -(\pi_W - 1) \\ 0 & 0 & 0 & x_{\mathbf{x}'_2} & y_{\mathbf{x}'_2} & w_{\mathbf{x}'_2} & 0 \\ x_{\mathbf{x}'_3} & y_{\mathbf{x}'_3} & w_{\mathbf{x}'_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{\mathbf{x}'_3} & y_{\mathbf{x}'_3} & w_{\mathbf{x}'_3} & -(\pi_H - 1) \end{pmatrix} \begin{pmatrix} v_{h1} \\ v_{h2} \\ v_{h3} \\ v_{h4} \\ v_{h5} \\ v_{h6} \\ v_{h7} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (7.6)$$

Any solver of linear system of equations serves to solve the DLT. In this dissertation, we choose Singular Value Decomposition (SVD) [Baker 2013]. SVD presents the decomposed matrix  $\mathcal{A}$  as:

$$\mathcal{A}_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T. \quad (7.7)$$

The necessary values for the  $\mathbf{H}_A$  matrix are found in  $V_{n \times n}^T$ . Only the last column of  $V_{n \times n}$  (i.e., the vector  $v_h$ ) is necessary to create the  $\mathbf{H}_A$  matrix.

The homographic matrix  $\mathbf{H}$  is found by calculating Equation 7.2. In Chapters 8 and 9, the homographic matrix  $\mathbf{H}$  is known as the projective transformation matrix  $\mathcal{M}$ .

## 7.2 Curve Fitting

Curve fitting is the process of constructing a curve, or mathematical function, that best fits a series of data points, possibly subject to constraints [Arlinghaus 1994]. Curve fitting can involve interpolation [Kiusalaas 2013], where exact fitting to the data is required, or smoothing [Savitzky e Golay 1964, Wilcox 2017], in which a smooth function is constructed that roughly fits the data.

The rectified ROI image is a binary image where the wave arms are present as edge pixels. We will only use one wave arm for the curve fitting process. We choose the wave closest to the observer (shown in Figure 7.2) because it is the wave least affected by the turbulence, i. e., in Figure 6.4, shown in Chapter 6, the chosen wave is the red wave. This wave has the highest value in the  $y$ -coordinate. The binary image of the chosen wave present noise, which makes it challenging to find the cusp of the wave. We use the LOcally WEighted Scatter-plot Smoother (LOWESS) algorithm [Wilcox 2017] to smooth the digital curve extracted from the rectified ROI image. The black points in Figure 7.3 corresponds to the digital curve in this example, while the red line is the resulting smooth curve used to estimate wavelength and velocity in the next steps of our approach.

### 7.3 Wavelength Estimation

The wavelength of the transverse waves (see Figure 2.6) can be estimated through Equation 2.15 by using the distance between successive crests/troughs of cusp waves. Thus, by finding the peaks of the curve resulting from the previous step of our approach, one finds a set of crests or a set of troughs, depending on the direction of the vessel with respect to the camera. Since we have the curve that corresponds to the arm closest to the V-pattern, we can find the crests and troughs depending on the direction of the vessel as follows (see Figure 7.5):

- When the vessel goes to the right in the original image, the curve maxima (yellow stars) corresponds to the wave arm's troughs;
- When the vessel goes to the right in the original image, the minimums of the curve (white stars) correspond to the wave arm's crests;
- When the vessel goes to the left in the original image, the curve maxima (yellow stars) correspond to the wave arm crests; and
- When the vessel goes to the left in the original image, the minimums of the curve (white stars) correspond to the wave arm's troughs.

It is important to note that the graph shown in Figure 7.5 presents the  $y$ -axis in a decreasing order, where the  $y = 0$  coordinate is at the top of the graph. We avoid the identification of noisy crests/troughs by imposing a minimum horizontal distance  $\psi$  between valid neighbor crests/troughs, and by extracting only the two minimums/maximums ones (the green stars in Figure 7.4 are the maximums).

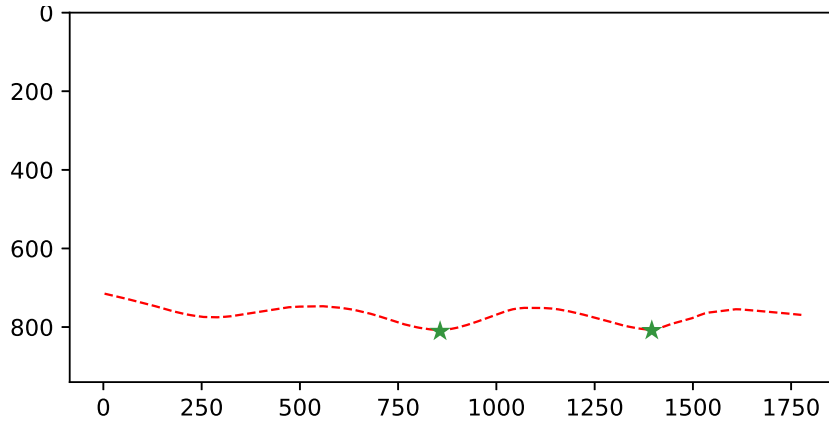


Figure 7.4: The red line represents the fitting curve of the right wave arm. The green stars and yellow dots represent the troughs found on the wave. Only the green stars are used to calculate the wavelength, the left and right green stars correspond to  $\mathbf{c}_n$ , and  $\mathbf{c}_{n+1}$ , respectively. The wave arm shown is from a ship going in the right direction.

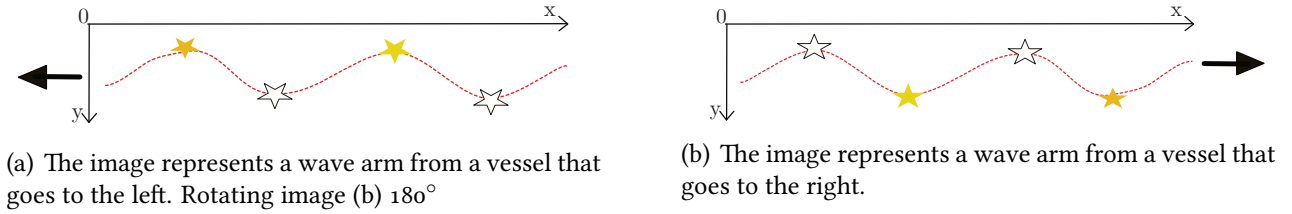


Figure 7.5: The wave arm of the ship going in the left direction, image (a), is similar to rotating  $180^\circ$  the wave arm of the ship going in the right direction, image (b). When making the rotation, the point-views of the crests and troughs change according to the ship's course. Taking image (b) as a reference, the yellow stars in image (b) are the troughs and white stars are crests, while in image (a) the white stars are the troughs and yellow stars are crests.

We compute the wavelength  $\lambda$  by replacing  $D$  in Equation 2.15, rewritten as:

$$\lambda = \frac{\sqrt{3}}{2}D, \quad (7.8)$$

where  $D$  is the Euclidean distance:

$$D = \text{dist}_E(\mathbf{c}_n, \mathbf{c}_{n+1}), \quad (7.9)$$

where the result of the Euclidean distance between the location of the extracted cusp (or troughs),  $\mathbf{c}_n$  and  $\mathbf{c}_{n+1}$ , is represented in centimeters. We calculate the wavelength using Equation 7.8. We convert the wavelength to meters to use Equation 7.10, recall that  $g$  is the gravitational acceleration ( $m/s^2$ ).

## 7.4 Vessel Velocity Estimation

Finally, we compute the velocity  $U$  of the vessel by evaluating (2.16), rewritten as:

$$U = \sqrt{\frac{g\lambda}{2\pi}}, \quad (7.10)$$

replacing  $\lambda$  found in Equation 7.8.

Equation 7.10 returns the vessel speed at  $m/s$ . Equation 7.11 is used to convert the speed from  $m/s$  to *knot*, where *knot* is a unit of measurement of speed, used for maritime navigation.

$$1 \text{ knot} = 0.5144 \text{ m/s} \quad (7.11)$$

## 7.5 Discussion

The ROI image rectification, curve fitting, and wavelength estimation to estimate the vessel speed was described in this chapter.

In this chapter, we assume that there are two wave arms present in the rectified ROI image and we chosen the closest to the camera. In the chosen wave of the ROI image must have at least two crests/troughs to find the wavelength. In this work, we used the troughs to calculate the wavelength because it was the least affected by the turbulence and the foam left by the ship. The existence of noise makes it difficult to find the correct wavelength. One way to mitigate this problem was to identify the noise presents in troughs by imposing a minimum horizontal distance  $\psi$  between the troughs of the valid neighbors.

If the wavelength is not calculated, it will be impossible to estimate the vessel speed with the proposed method. The vessel speed is strongly linked to the wavelength - the higher the wavelength, the greater the speed.

The results obtained from the application of the techniques described in this chapter represent one of the main objectives of this work, and the analysis of these results is presented in Chapter 9.

# Chapter 8

## First-Order Error Propagation

First-Order Error Propagation is a method that determines the uncertainty in a function of independent variables, each with an uncertainty. In this chapter, we apply First-Order Error Propagation to determine the uncertainty in the function to estimate vessel speed. The vessel speed  $U$  was computed by the method described in Chapter 7.

The measurement process in some contexts is not guaranteed if a confidence interval did not accompany it, because, if the confidence interval were too high, it is concluded that the measurement is not accurate enough to be used. For this reason, it is interesting to have a confidence interval in the measurements. Confidence intervals may be estimated from several samples. In our case, each sample would come from a different image of the same vessel, captured virtually at the same time. An alternative to not creating confidence intervals from a large number of samples is using First-Order Error Propagation.

In order to use the First-Order Error Propagation model, it is necessary to verify the premises that allow its application. First-order error propagation can be applied by knowing the result of the error propagation, and this can be done if and only if the uncertainty input follows a Gaussian distribution, the output follows a Gaussian distribution, and the process is linear. Now, if the process is not linear or the inputs are not Gaussian, we can work with First-Order Error Propagation and have an approximation.

The study of the propagation of errors along the computational chain (Figure 8.1) is presented in Section 8.1. In the following sections, the calculation of the Jacobian matrix in each of the intermediate variables is explained.



## 8.1 Error Propagation

In Theory of Errors [Vuolo 1996], it is presented the expressions to calculate the standard uncertainty of a measurements  $w$ , which is calculated from standard uncertainties of other experimental great-nesses  $(x, y, z, \dots)$ . The standard uncertainty of  $w$  is given by the Equation 8.1 in the case that the variables are not independent (or correlated). The covariance  $\sigma_{ij}$  indicates the degree of correlation between two variables:

$$\begin{aligned} \sigma_w^2 = & \left( \frac{\partial w}{\partial x} \right)^2 \sigma_x^2 + \left( \frac{\partial w}{\partial y} \right)^2 \sigma_y^2 + \left( \frac{\partial w}{\partial z} \right)^2 \sigma_z^2 + \dots \\ & + 2 \left( \frac{\partial w}{\partial x} \right) \left( \frac{\partial w}{\partial y} \right) \sigma_{xy} + 2 \left( \frac{\partial w}{\partial x} \right) \left( \frac{\partial w}{\partial z} \right) \sigma_{xz} + 2 \left( \frac{\partial w}{\partial y} \right) \left( \frac{\partial w}{\partial z} \right) \sigma_{yz} + \dots \end{aligned} \quad (8.1)$$

The computational flow of the velocity calculation method described in Chapter 7 is illustrated in Figure 8.1. In that Figure, the input variables that have an uncertainty value are represented by circles, and the input variables without uncertainty are represented by a pentagon, squares represent the intermediary variables, and a rhombus represents the result of the vessel speed. Equation 8.1 is adopted in this work. The Equation 8.1 is rewritten in matrix form, to simplify its notation:

$$\Lambda_w = \nabla \mathcal{J} \Lambda_\theta \nabla \mathcal{J}^T, \quad (8.2)$$

where  $\Lambda_w$  is a  $m \times m$  matrix of variances and covariances,  $\nabla \mathcal{J}$  is an  $m \times n$  Jacobian matrix that will tell us how much of the input uncertainty will influence the output result, and  $\Lambda_\theta$  is an  $n \times n$  covariance matrix that correlates the input variables. For the vessel's speed estimation problem, only one resulting variable is calculated, the velocity. Therefore,  $m = 1$ .

In total, there are 18 input parameters in the proposed:

- $v_i$ , for  $i = (l, r)$ , are the  $y$ -axis coordinates of, respectively, the left and right endpoints of the detected vanishing line. The points are taken as homogeneous coordinates  $\mathbf{p}_{vi} = (x_i, v_i, 1)^T$ . The values of  $x_i$  are constant since they are the same for all samples, leading to only two input parameters affected by uncertainty;
- $h_j$ , where  $j \in \{g, f, t, s\}$ , is the list of four input variables related to the four parameters used to calculate the camera height, these four variables are affected by uncertainties;
- **K**: matrix of the intrinsic parameters of the camera.  $\alpha_x = fm_x$ ,  $\alpha_y = fm_y$ ,  $\gamma$ ,  $o_x$ ,  $o_y$ , where  $f$  is the focal length and  $m_x$  and  $m_y$  are the ratios between the sensor and the image measurements,  $s$  is the coefficient that defines the angle between the  $x$ - and  $y$ -axis of the pixels, and  $o_x$  and  $o_y$  are the coordinates of the image center. These parameters, taken as constant

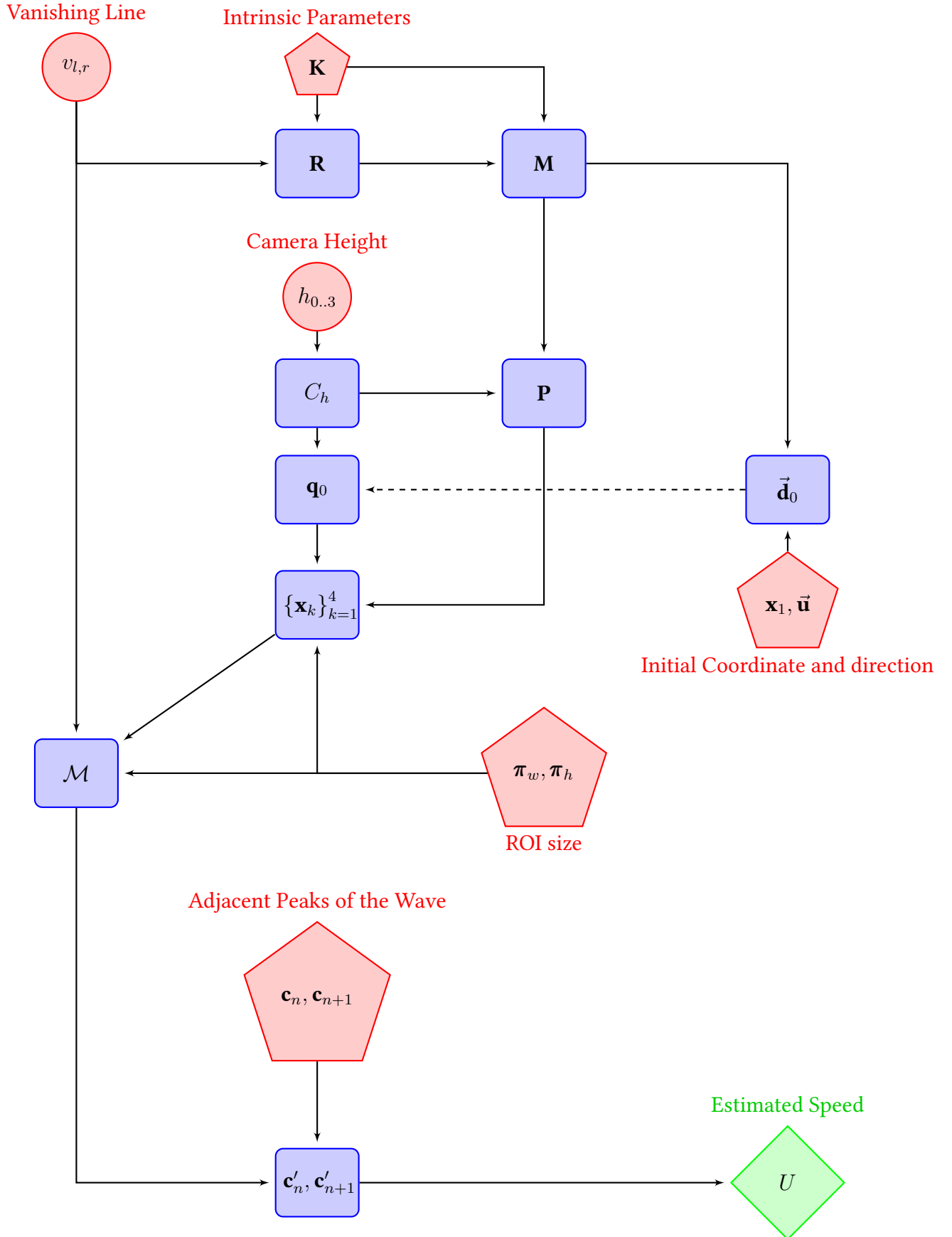


Figure 8.1: Error propagation along the computational chain. It was not considered an uncertainty for the parameters represented by pentagons. In contrast, the input parameters represented as circles contain errors that are propagated by the intermediate stages (squares) until the velocity (rhombus) is calculated.

values, used to compute the matrix of intrinsic parameters  $\mathbf{K}$ . The intrinsic parameters of the camera are obtained by extracting metadata from the image and are assumed as constant values since they usually do not have much influence on the uncertainty of the measurements [Fernandes, Oliveira e Silva 2008];

- $\pi_w$  and  $\pi_h$  define the size of the ROI in world coordinate system, represented in meters. These two parameters are taken as constant values;
- $\mathbf{x}_1 = (x_1, y_1, 1)^T$  and  $\vec{\mathbf{u}} = (x_{\vec{\mathbf{u}}}, y_{\vec{\mathbf{u}}}, 0)^T$  are, respectively, the coordinates of the first vertex of the and the direction of the boat in image space. These coordinates are manually defined and they are not taken as a source of uncertainty; and
- $\mathbf{c}_n, \mathbf{c}_{n+1}$ , the two adjacent crests/troughs of the wave, extracted from non-rectified ROI image, these two peaks are variables not taken as a source of uncertainty, because their rectified versions will naturally include the uncertainty propagated from other input variables.

As mentioned earlier in this section, only the coordinates of the vanishing line and the camera height contain uncertainties. The input variables have covariance equal to zero because we are assuming independence on the variables and independence implies covariance equal to zero. These variables are used as input parameters for matrix  $\Lambda_\vartheta$ . Then, matrix  $\Lambda_\vartheta$  of variances and covariances is a  $6 \times 6$  diagonal matrix:

$$\Lambda_\vartheta = \text{diag}(\Lambda_{v_l}, \Lambda_{v_r}, \Lambda_{h_g}, \Lambda_{h_f}, \Lambda_{h_t}, \Lambda_{h_s}) \quad (8.3)$$

The following sections show the calculation of the partial derivatives of the Jacobian matrix used in the propagation of errors from the input variables to the estimated velocity.

To estimate the uncertainty on the computed velocity, it is necessary to find the Jacobian matrix of the function that calculates the velocity (Equation 8.4) and solve Equation 8.2 using the Jacobian ( $(\nabla_U)$ ) and the matrix  $\Lambda_\vartheta$  of Equation 8.3, where

$$\nabla_U = \left( \frac{\partial U}{\partial v_l} \quad \frac{\partial U}{\partial v_r} \quad \frac{\partial U}{\partial h_g} \quad \frac{\partial U}{\partial h_f} \quad \frac{\partial U}{\partial h_t} \quad \frac{\partial U}{\partial h_s} \right) \quad (8.4)$$

The confidence intervals calculated by the error propagation define the expected error in each of the measurements made by the system. Such propagation is expressed by Equation 8.2, and rewritten as:

$$\sigma_U^2 = \Lambda_U = \nabla_U \Lambda_\vartheta \nabla_U^T, \quad (8.5)$$

where  $\sigma_U^2$  is the variance of the vessel speed,  $\Lambda_\vartheta$  is a  $6 \times 6$  covariance matrix that encodes the uncertainty of the input variables, and  $\nabla_U$  is the Jacobian matrix of the function that calculates the

speed based on the six variables included in the set  $\vartheta$  (see Chapter 5):

$$\vartheta = \{v_l, v_r, h_g, h_f, h_t, h_s\}. \quad (8.6)$$

Writing the equation that calculates the vessel speed based on the input variables and calculating its Jacobian matrix is unavailable because the equation would be very extensive and with many terms. One solution found for this problem is to solve the partial derivatives contained in the  $\nabla_U$  matrix using the chain rule and apply First Order Error Propagation [Vuolo 1996]. Figure 8.1 shows the flow of operations used to estimate speed. This figure helps us to monitor the operations chain and, accordingly, to apply the chain rule.

## 8.2 A Line Through Two Points

Following Figure 8.1, the first stage of the processing consists in obtaining the vanishing line, which is a line defined by a pair of points. The input variables  $v_l, v_r$  are used to calculate the vanishing line  $\mathbf{l}$ :

$$\mathbf{l} = \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \mathbf{p}_{vl} \times \mathbf{p}_{vr} = \begin{pmatrix} v_l - v_r \\ W - 1 \\ (W - 1)v_l \end{pmatrix}, \quad (8.7)$$

where  $v_l, v_r$  are the  $y$ -axis of each point, 0 and  $W$  are the  $x$ -axis of each point, respectively, where  $W$  is the input image width. Given that the pair of points is in homogeneous coordinates.

Partial derivatives of the vanishing line  $\mathbf{l}$  are given by:

$$\frac{\partial A}{\partial \vartheta} = \frac{\partial v_l}{\partial \vartheta} + \frac{\partial v_r}{\partial \vartheta} \quad (8.8a)$$

$$\frac{\partial B}{\partial \vartheta} = 0 \quad (8.8b)$$

$$\frac{\partial C}{\partial \vartheta} = (W - 1) \frac{\partial v_l}{\partial \vartheta} \quad (8.8c)$$

## 8.3 Camera Height

Camera height  $h$  is calculated by adding the ground height, floor height, tripod height, and sea tide height, i.e.,  $h_g, h_f, h_t$ , and  $h_s$ , respectively:

$$h = h_g + F h_f + h_t + h_s, \quad (8.9)$$

where  $F$  is a constant that represents the floor number on which the camera is mounted. The partial derivative of the camera height  $\frac{\partial h}{\partial \vartheta}$  is given by:

$$\frac{\partial h}{\partial \vartheta} = \frac{\partial h_g}{\partial \vartheta} + F \frac{\partial h_f}{\partial \vartheta} + \frac{\partial h_t}{\partial \vartheta} + \frac{\partial h_s}{\partial \vartheta} \quad (8.10)$$

## 8.4 Rotation Matrix

The rotation matrix  $\mathbf{R}$  aligns the world coordinate system to the camera coordinate system. For the creation of the rotation matrix, it is used the intrinsic parameters matrix of the camera  $\mathbf{K}$  (Equation 2.2) and the vanishing line coefficients  $\mathbf{l}$ , in image coordinates. The rotation matrix is a  $3 \times 3$  matrix, defined by:

$$\mathbf{R} = \begin{pmatrix} \vec{\mathbf{R}}_1^T \\ \vec{\mathbf{R}}_2^T \\ \vec{\mathbf{R}}_3^T \end{pmatrix}, \quad (8.11)$$

where  $\vec{\mathbf{R}}_1$ ,  $\vec{\mathbf{R}}_2$  and  $\vec{\mathbf{R}}_3$  are  $3 \times 1$  vectors.

To find  $\vec{\mathbf{R}}_1$  and  $\vec{\mathbf{R}}_2$  is necessary to calculate  $\vec{\mathbf{R}}_3$  first (as described in Section 5.2.2):

$$\vec{\mathbf{R}}_3 = \text{unit}(\text{up}(\mathbf{K}^T \mathbf{l})). \quad (8.12)$$

But to find  $\vec{\mathbf{R}}_3$ , it is necessary to calculate  $\mathbf{K}^T \mathbf{l}$

$$\Phi = \begin{pmatrix} X_\Phi \\ Y_\Phi \\ Z_\Phi \end{pmatrix} = \mathbf{K}^T \mathbf{l} = \begin{pmatrix} \alpha_x A \\ \gamma A - \alpha_y B \\ o_x A + o_y B + C \end{pmatrix}, \quad (8.13)$$

along with its partial derivatives:

$$\frac{\partial X_\Phi}{\partial \vartheta} = \alpha_x \frac{\partial A}{\partial \vartheta} \quad (8.14a)$$

$$\frac{\partial Y_\Phi}{\partial \vartheta} = \gamma \frac{\partial A}{\partial \vartheta} + \alpha_y \frac{\partial B}{\partial \vartheta} \quad (8.14b)$$

$$\frac{\partial Z_\Phi}{\partial \vartheta} = o_x \frac{\partial A}{\partial \vartheta} + o_y \frac{\partial B}{\partial \vartheta} + \frac{\partial C}{\partial \vartheta} \quad (8.14c)$$

The unit function in Equation 8.12 normalizes the vector to have unit length. Finding the normal for  $\Phi$ :

$$\varkappa_1 = \sqrt{X_\Phi^2 + Y_\Phi^2 + Z_\Phi^2}, \quad (8.15)$$

and the partial derivative  $\frac{\partial \kappa_1}{\partial \vartheta}$  is given by:

$$\frac{\partial \kappa_1}{\partial \vartheta} = \frac{X_\Phi \left( \frac{\partial X_\Phi}{\partial \vartheta} \right) + Y_\Phi \left( \frac{\partial Y_\Phi}{\partial \vartheta} \right) + Z_\Phi \left( \frac{\partial Z_\Phi}{\partial \vartheta} \right)}{\sqrt{X_\Phi^2 + Y_\Phi^2 + Z_\Phi^2}} \quad (8.16)$$

The up function in Equation 8.12 forces the vector to point upward. If  $Y_{\vec{\mathbf{R}}_3}$  is less than zero then  $\vec{\mathbf{R}}_3$  is equal to  $-\vec{\mathbf{R}}_3$ , where  $\vec{\mathbf{R}}_3$  is defined by:

$$\vec{\mathbf{R}}_3 = \begin{pmatrix} X_{\vec{\mathbf{R}}_3} \\ Y_{\vec{\mathbf{R}}_3} \\ Z_{\vec{\mathbf{R}}_3} \end{pmatrix} = \begin{pmatrix} X_\Phi / \kappa_1 \\ Y_\Phi / \kappa_1 \\ Z_\Phi / \kappa_1 \end{pmatrix}, \quad (8.17)$$

and the partial derivatives of the components of  $\vec{\mathbf{R}}_3$  are given by:

$$\frac{\partial X_{\vec{\mathbf{R}}_3}}{\partial \vartheta} = \frac{\kappa_1 \left( \frac{\partial X_\Phi}{\partial \vartheta} \right) - X_\Phi \left( \frac{\partial \kappa_1}{\partial \vartheta} \right)}{\kappa_1^2} \quad (8.18a)$$

$$\frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} = \frac{\kappa_1 \left( \frac{\partial Y_\Phi}{\partial \vartheta} \right) - Y_\Phi \left( \frac{\partial \kappa_1}{\partial \vartheta} \right)}{\kappa_1^2} \quad (8.18b)$$

$$\frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} = \frac{\kappa_1 \left( \frac{\partial Z_\Phi}{\partial \vartheta} \right) - Z_\Phi \left( \frac{\partial \kappa_1}{\partial \vartheta} \right)}{\kappa_1^2} \quad (8.18c)$$

After calculating  $\vec{\mathbf{R}}_3$ , we can calculate  $\vec{\mathbf{R}}_2$ , where  $\vec{\mathbf{R}}_2$  is defined by  $\vec{\mathbf{R}}_2 = \text{unit}((1, 0, 0)^T \times \vec{\mathbf{R}}_3)$ :

$$\vec{\mathbf{R}}_2 = \begin{pmatrix} X_{\vec{\mathbf{R}}_2} \\ Y_{\vec{\mathbf{R}}_2} \\ Z_{\vec{\mathbf{R}}_2} \end{pmatrix} = \begin{pmatrix} 0 \\ -Z_{\vec{\mathbf{R}}_3} / \kappa_2 \\ Y_{\vec{\mathbf{R}}_3} / \kappa_2 \end{pmatrix}, \quad (8.19)$$

where  $\kappa_2$  is the norm, and is defined by Equation 8.20, together with its partial derivative:

$$\kappa_2 = \sqrt{(-Z_{\vec{\mathbf{R}}_3})^2 + Y_{\vec{\mathbf{R}}_3}^2} \quad (8.20a)$$

$$\frac{\partial \kappa_2}{\partial \vartheta} = \frac{Z_{\vec{\mathbf{R}}_3} \left( \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \right) + Y_{\vec{\mathbf{R}}_3} \left( \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \right)}{\sqrt{(-Z_{\vec{\mathbf{R}}_3})^2 + Y_{\vec{\mathbf{R}}_3}^2}} \quad (8.20b)$$

, and the derivatives of the  $\vec{\mathbf{R}}_2$  coefficients (Equation 8.19) are equal to

$$\frac{\partial X_{\vec{\mathbf{R}}_2}}{\partial \vartheta} = 0 \quad (8.21a)$$

$$\frac{\partial Y_{\vec{\mathbf{R}}_2}}{\partial \vartheta} = \frac{-\kappa_2 \left( \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \right) + Z_{\vec{\mathbf{R}}_3} \left( \frac{\partial \kappa_2}{\partial \vartheta} \right)}{\kappa_2^2} \quad (8.21b)$$

$$\frac{\partial Z_{\vec{\mathbf{R}}_2}}{\partial \vartheta} = \frac{\kappa_2 \left( \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \right) - Y_{\vec{\mathbf{R}}_3} \left( \frac{\partial \kappa_2}{\partial \vartheta} \right)}{\kappa_2^2} \quad (8.21c)$$

Finally, to find the rotation matrix  $\mathbf{R}$  we calculate  $\vec{\mathbf{R}}_1$ , which is defined by  $\vec{\mathbf{R}}_1 = \vec{\mathbf{R}}_3 \times \vec{\mathbf{R}}_2$ .

$$\vec{\mathbf{R}}_1 = \begin{pmatrix} X_{\vec{\mathbf{R}}_1} \\ Y_{\vec{\mathbf{R}}_1} \\ Z_{\vec{\mathbf{R}}_1} \end{pmatrix} = \begin{pmatrix} Y_{\vec{\mathbf{R}}_3} Z_{\vec{\mathbf{R}}_2} - Z_{\vec{\mathbf{R}}_3} Y_{\vec{\mathbf{R}}_2} \\ -X_{\vec{\mathbf{R}}_3} Z_{\vec{\mathbf{R}}_2} \\ X_{\vec{\mathbf{R}}_3} Y_{\vec{\mathbf{R}}_2} \end{pmatrix}, \quad (8.22)$$

and the derivatives of the  $\vec{\mathbf{R}}_1$  coefficients (Equation 8.22) are equal to

$$\frac{\partial X_{\vec{\mathbf{R}}_1}}{\partial \vartheta} = \left( Z_{\vec{\mathbf{R}}_2} \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} + Y_{\vec{\mathbf{R}}_3} \frac{\partial Z_{\vec{\mathbf{R}}_2}}{\partial \vartheta} \right) - \left( Y_{\vec{\mathbf{R}}_2} \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} + Z_{\vec{\mathbf{R}}_3} \frac{\partial Y_{\vec{\mathbf{R}}_2}}{\partial \vartheta} \right) \quad (8.23a)$$

$$\frac{\partial Y_{\vec{\mathbf{R}}_1}}{\partial \vartheta} = - \left( Z_{\vec{\mathbf{R}}_2} \frac{\partial X_{\vec{\mathbf{R}}_3}}{\partial \vartheta} + X_{\vec{\mathbf{R}}_3} \frac{\partial Z_{\vec{\mathbf{R}}_2}}{\partial \vartheta} \right) \quad (8.23b)$$

$$\frac{\partial Z_{\vec{\mathbf{R}}_1}}{\partial \vartheta} = Y_{\vec{\mathbf{R}}_2} \frac{\partial X_{\vec{\mathbf{R}}_3}}{\partial \vartheta} + X_{\vec{\mathbf{R}}_3} \frac{\partial Y_{\vec{\mathbf{R}}_2}}{\partial \vartheta} \quad (8.23c)$$

## 8.5 Sub-matrix of the Projection Matrix

Once calculated the rotation matrix  $\mathbf{R}$ , and having as input the matrix  $\mathbf{K}$ , it is possible to calculate  $\mathbf{M}$ , where  $\mathbf{M}$  is defined by :

$$\mathbf{M} = \begin{pmatrix} \vec{\mathbf{M}}_1^T \\ \vec{\mathbf{M}}_2^T \\ \vec{\mathbf{M}}_3^T \end{pmatrix} = \mathbf{K} \mathbf{R} \quad (8.24)$$

Each component of the matrix  $\mathbf{M}$  is described in  $\vec{\mathbf{M}}_1$ ,  $\vec{\mathbf{M}}_2$ , and  $\vec{\mathbf{M}}_3$ , together with their corresponding partial derivatives:

$$\vec{\mathbf{M}}_1 = \begin{pmatrix} X_{\vec{\mathbf{M}}_1} \\ Y_{\vec{\mathbf{M}}_1} \\ Z_{\vec{\mathbf{M}}_1} \end{pmatrix} = \begin{pmatrix} \alpha_x X_{\vec{\mathbf{R}}_1} + \gamma X_{\vec{\mathbf{R}}_2} + o_x X_{\vec{\mathbf{R}}_3} \\ \alpha_x Y_{\vec{\mathbf{R}}_1} + \gamma Y_{\vec{\mathbf{R}}_2} + o_x Y_{\vec{\mathbf{R}}_3} \\ \alpha_x Z_{\vec{\mathbf{R}}_1} + \gamma Z_{\vec{\mathbf{R}}_2} + o_x Z_{\vec{\mathbf{R}}_3} \end{pmatrix} \quad (8.25a)$$

$$\frac{\partial X_{\vec{\mathbf{M}}_1}}{\partial \vartheta} = \alpha_x \frac{\partial X_{\vec{\mathbf{R}}_1}}{\partial \vartheta} + \gamma \frac{\partial X_{\vec{\mathbf{R}}_2}}{\partial \vartheta} + o_x \frac{\partial X_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.25b)$$

$$\frac{\partial Y_{\vec{\mathbf{M}}_1}}{\partial \vartheta} = \alpha_x \frac{\partial Y_{\vec{\mathbf{R}}_1}}{\partial \vartheta} + \gamma \frac{\partial Y_{\vec{\mathbf{R}}_2}}{\partial \vartheta} + o_x \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.25c)$$

$$\frac{\partial Z_{\vec{\mathbf{M}}_1}}{\partial \vartheta} = \alpha_x \frac{\partial Z_{\vec{\mathbf{R}}_1}}{\partial \vartheta} + \gamma \frac{\partial Z_{\vec{\mathbf{R}}_2}}{\partial \vartheta} + o_x \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.25d)$$

$$\vec{\mathbf{M}}_2 = \begin{pmatrix} X_{\vec{\mathbf{M}}_2} \\ Y_{\vec{\mathbf{M}}_2} \\ Z_{\vec{\mathbf{M}}_2} \end{pmatrix} = \begin{pmatrix} \alpha_y X_{\vec{\mathbf{R}}_2} + o_y X_{\vec{\mathbf{R}}_3} \\ \alpha_y Y_{\vec{\mathbf{R}}_2} + o_y Y_{\vec{\mathbf{R}}_3} \\ \alpha_y Z_{\vec{\mathbf{R}}_2} + o_y Z_{\vec{\mathbf{R}}_3} \end{pmatrix} \quad (8.26a)$$

$$\frac{\partial X_{\vec{\mathbf{M}}_2}}{\partial \vartheta} = \alpha_y \frac{\partial X_{\vec{\mathbf{R}}_2}}{\partial \vartheta} + o_y \frac{\partial X_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.26b)$$

$$\frac{\partial Y_{\vec{\mathbf{M}}_2}}{\partial \vartheta} = \alpha_y \frac{\partial Y_{\vec{\mathbf{R}}_2}}{\partial \vartheta} + o_y \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.26c)$$

$$\frac{\partial Z_{\vec{\mathbf{M}}_2}}{\partial \vartheta} = \alpha_y \frac{\partial Z_{\vec{\mathbf{R}}_2}}{\partial \vartheta} + o_y \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.26d)$$

$$\vec{\mathbf{M}}_3 = \begin{pmatrix} X_{\vec{\mathbf{M}}_3} \\ Y_{\vec{\mathbf{M}}_3} \\ Z_{\vec{\mathbf{M}}_3} \end{pmatrix} = \begin{pmatrix} X_{\vec{\mathbf{R}}_3} \\ Y_{\vec{\mathbf{R}}_3} \\ Z_{\vec{\mathbf{R}}_3} \end{pmatrix} \quad (8.27a)$$

$$\frac{\partial X_{\vec{\mathbf{M}}_3}}{\partial \vartheta} = \frac{\partial X_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.27b)$$

$$\frac{\partial Y_{\vec{\mathbf{M}}_3}}{\partial \vartheta} = \frac{\partial Y_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.27c)$$

$$\frac{\partial Z_{\vec{\mathbf{M}}_3}}{\partial \vartheta} = \frac{\partial Z_{\vec{\mathbf{R}}_3}}{\partial \vartheta} \quad (8.27d)$$

## 8.6 Projection Matrix

The  $3 \times 4$  projection matrix  $\mathbf{P}$  maps points in 3D world coordinates to points in 2D image coordinates.  $\mathbf{P}$  can be decomposed as  $\mathbf{P} = \mathbf{M}[\mathbf{I} | -\mathbf{C}_h]$  [Hartley e Zisserman 2004]. Indeed, letting  $\mathbf{M}$  be the left  $3 \times 3$  submatrix of  $\mathbf{P}$ , one decomposes  $\mathbf{M}$  as a product  $\mathbf{M} = \mathbf{K}\mathbf{R}$ .  $\mathbf{I}$  is a  $3 \times 3$  identity matrix and  $\mathbf{C}_h$  a point  $\mathbf{C}_h = (0, 0, h)^T$ , where  $h$  is the camera height defined in Equation 8.9. We can rewrite the projection matrix  $\mathbf{P}$  as:

$$\mathbf{P} = \mathbf{M}[\mathbf{I} | -\mathbf{C}_h] = \begin{pmatrix} \vec{\mathbf{P}}_1^T \\ \vec{\mathbf{P}}_2^T \\ \vec{\mathbf{P}}_3^T \end{pmatrix} \quad (8.28)$$

Each component of the matrix  $\mathbf{P}$  is described in  $\vec{\mathbf{P}}_1$ ,  $\vec{\mathbf{P}}_2$ , and  $\vec{\mathbf{P}}_3$ , together with their corresponding



partial derivatives:

$$\vec{\mathbf{P}}_i = \begin{pmatrix} X_{\vec{\mathbf{P}}_i} \\ Y_{\vec{\mathbf{P}}_i} \\ Z_{\vec{\mathbf{P}}_i} \\ W_{\vec{\mathbf{P}}_i} \end{pmatrix} = \begin{pmatrix} X_{\vec{\mathbf{M}}_i} \\ Y_{\vec{\mathbf{M}}_i} \\ Z_{\vec{\mathbf{M}}_i} \\ -hZ_{\vec{\mathbf{M}}_i} \end{pmatrix} \quad (8.29a)$$

$$\frac{\partial X_{\vec{\mathbf{P}}_i}}{\partial \vartheta} = \frac{\partial X_{\vec{\mathbf{M}}_i}}{\partial \vartheta} \quad (8.29b)$$

$$\frac{\partial Y_{\vec{\mathbf{P}}_i}}{\partial \vartheta} = \frac{\partial Y_{\vec{\mathbf{M}}_i}}{\partial \vartheta} \quad (8.29c)$$

$$\frac{\partial Z_{\vec{\mathbf{P}}_i}}{\partial \vartheta} = \frac{\partial Z_{\vec{\mathbf{M}}_i}}{\partial \vartheta} \quad (8.29d)$$

$$\frac{\partial W_{\vec{\mathbf{P}}_i}}{\partial \vartheta} = -Z_{\vec{\mathbf{M}}_i} \frac{\partial h}{\partial \vartheta} - h \frac{\partial Z_{\vec{\mathbf{M}}_i}}{\partial \vartheta}, \quad (8.29e)$$

where  $i \in \{1, 2, 3\}$ . The projection matrix  $\mathbf{P}$  is used to calculate the coordinates  $\mathbf{x}_1$ ,  $\mathbf{x}_2$ ,  $\mathbf{x}_3$ , and  $\mathbf{x}_4$  in the image coordinate system.

## 8.7 Compute the Location of the Four Corners of the ROI

One of the four coordinates of the ROI is taken as an input variable. The other ones are computed from it and using existing information. The  $\mathbf{x}_1$  coordinate is the initial corner of the ROI. This input variable is represented as a homogeneous coordinate  $\mathbf{x}_1 = (x_{\mathbf{x}_1}, y_{\mathbf{x}_1}, 1)^T$ . To find the positions of the four corners in image coordinates, we will first determine the direction of the ray  $\vec{\mathbf{d}}_0$  to the reference point  $\mathbf{q}_0$  (Equation 5.3) in world coordinates. The direction of the ray is defined by Equation 8.30, and their respective partial derivative by Equation 8.31:

$$\vec{\mathbf{d}}_0 = \begin{pmatrix} X_{\vec{\mathbf{d}}_0} \\ Y_{\vec{\mathbf{d}}_0} \\ Z_{\vec{\mathbf{d}}_0} \end{pmatrix} = \mathbf{M}^{-1} \mathbf{x}_1 = \begin{pmatrix} X_{\mathbf{M}_1^{-1}} x_{\mathbf{x}_1} + Y_{\mathbf{M}_1^{-1}} y_{\mathbf{x}_1} + Z_{\mathbf{M}_1^{-1}} \\ X_{\mathbf{M}_2^{-1}} x_{\mathbf{x}_1} + Y_{\mathbf{M}_2^{-1}} y_{\mathbf{x}_1} + Z_{\mathbf{M}_2^{-1}} \\ X_{\mathbf{M}_3^{-1}} x_{\mathbf{x}_1} + Y_{\mathbf{M}_3^{-1}} y_{\mathbf{x}_1} + Z_{\mathbf{M}_3^{-1}} \end{pmatrix}, \quad (8.30)$$

where  $\mathbf{M}^{-1}$  is the inverse of  $\mathbf{M}$ . The derivative of  $\vec{\mathbf{d}}_0$  with respect to  $\vartheta$  is:

$$\frac{\partial X_{\vec{\mathbf{d}}_0}}{\partial \vartheta} = x_{\mathbf{x}_1} \frac{\partial X_{\mathbf{M}_1^{-1}}}{\partial \vartheta} + y_{\mathbf{x}_1} \frac{\partial Y_{\mathbf{M}_1^{-1}}}{\partial \vartheta} + \frac{\partial Z_{\mathbf{M}_1^{-1}}}{\partial \vartheta} \quad (8.31a)$$

$$\frac{\partial Y_{\vec{\mathbf{d}}_0}}{\partial \vartheta} = x_{\mathbf{x}_1} \frac{\partial X_{\mathbf{M}_2^{-1}}}{\partial \vartheta} + y_{\mathbf{x}_1} \frac{\partial Y_{\mathbf{M}_2^{-1}}}{\partial \vartheta} + \frac{\partial Z_{\mathbf{M}_2^{-1}}}{\partial \vartheta} \quad (8.31b)$$

$$\frac{\partial Z_{\vec{\mathbf{d}}_0}}{\partial \vartheta} = x_{\mathbf{x}_1} \frac{\partial X_{\mathbf{M}_3^{-1}}}{\partial \vartheta} + y_{\mathbf{x}_1} \frac{\partial Y_{\mathbf{M}_3^{-1}}}{\partial \vartheta} + \frac{\partial Z_{\mathbf{M}_3^{-1}}}{\partial \vartheta} \quad (8.31c)$$

The location of the reference point  $\mathbf{q}_0$  of the ROI in world coordinates is defined by Equation 5.3,

converted to homogeneous coordinate system, and rewrite as:

$$\mathbf{q}_0 = \begin{pmatrix} X_{\mathbf{q}_0} \\ Y_{\mathbf{q}_0} \\ Z_{\mathbf{q}_0} \\ W_{\mathbf{q}_0} \end{pmatrix} = \begin{pmatrix} -h(X_{\vec{\mathbf{d}}_0}/Z_{\vec{\mathbf{d}}_0}) \\ -h(Y_{\vec{\mathbf{d}}_0}/Z_{\vec{\mathbf{d}}_0}) \\ 0 \\ 1 \end{pmatrix}^T, \quad (8.32)$$

where the partial derivative of each component is:

$$\frac{\partial X_{\mathbf{q}_0}}{\partial \vartheta} = -X_{\vec{\mathbf{d}}_0} \frac{-Z_{\vec{\mathbf{d}}_0} \frac{\partial h}{\partial \vartheta} + h \frac{\partial Z_{\vec{\mathbf{d}}_0}}{\partial \vartheta}}{Z_{\vec{\mathbf{d}}_0}^2} - \frac{h}{Z_{\vec{\mathbf{d}}_0}} \frac{\partial X_{\vec{\mathbf{d}}_0}}{\partial \vartheta} \quad (8.33a)$$

$$\frac{\partial Y_{\mathbf{q}_0}}{\partial \vartheta} = -Y_{\vec{\mathbf{d}}_0} \frac{-Z_{\vec{\mathbf{d}}_0} \frac{\partial h}{\partial \vartheta} + h \frac{\partial Z_{\vec{\mathbf{d}}_0}}{\partial \vartheta}}{Z_{\vec{\mathbf{d}}_0}^2} - \frac{h}{Z_{\vec{\mathbf{d}}_0}} \frac{\partial Y_{\vec{\mathbf{d}}_0}}{\partial \vartheta} \quad (8.33b)$$

$$\frac{\partial Z_{\mathbf{q}_0}}{\partial \vartheta} = \frac{\partial W_{\mathbf{q}_0}}{\partial \vartheta} = 0 \quad (8.33c)$$

$$(8.33d)$$

With the calculated reference point  $\mathbf{q}_0$  and direction of the ray  $\vec{\mathbf{d}}_0$ , we find the remaining points in the world coordinate system.  $\mathbf{X}_2$ ,  $\mathbf{X}_3$ , and  $\mathbf{X}_4$  are calculated by moving  $\mathbf{X}_1$  by  $\pi_W$  and  $\pi_H$ , where  $\pi_W$  and  $\pi_H$  are the width and height of the ROI image in meters, respectively. The points in the world coordinate system are defined by:

$$\mathbf{X}_1 = \mathbf{q}_0 \quad (8.34a)$$

$$\mathbf{X}_2 = (X_{\mathbf{q}_0} + \pi_W, Y_{\mathbf{q}_0}, 0, 1) \quad (8.34b)$$

$$\mathbf{X}_3 = (X_{\mathbf{q}_0} + \pi_W, Y_{\mathbf{q}_0} + \pi_H, 0, 1) \quad (8.34c)$$

$$\mathbf{X}_4 = (X_{\mathbf{q}_0}, Y_{\mathbf{q}_0} + \pi_H, 0, 1) \quad (8.34d)$$

The next step is to convert the world coordinate system points for the image coordinate system. For this, points  $\mathbf{X}_1$ ,  $\mathbf{X}_2$ ,  $\mathbf{X}_3$ , and  $\mathbf{X}_4$  are multiplied by the projection matrix  $\mathbf{P}$ :

$$\mathbf{x}_k = \mathbf{P}\mathbf{X}_k^T, \text{ where } k \in \{1, 2, 3, 4\}. \quad (8.35)$$

$\mathbf{x}_1$  in the image coordinate system is calculated by Equation 8.36

$$\mathbf{x}_1 = \begin{pmatrix} x_{\mathbf{x}_1} \\ y_{\mathbf{x}_1} \\ w_{\mathbf{x}_1} \end{pmatrix} = \begin{pmatrix} X_{\vec{\mathbf{p}}_1} X_{\mathbf{q}_0} + Y_{\vec{\mathbf{p}}_1} Y_{\mathbf{q}_0} + W_{\vec{\mathbf{p}}_1} \\ X_{\vec{\mathbf{p}}_2} X_{\mathbf{q}_0} + Y_{\vec{\mathbf{p}}_2} Y_{\mathbf{q}_0} + W_{\vec{\mathbf{p}}_2} \\ X_{\vec{\mathbf{p}}_3} X_{\mathbf{q}_0} + Y_{\vec{\mathbf{p}}_3} Y_{\mathbf{q}_0} + W_{\vec{\mathbf{p}}_3} \end{pmatrix}, \quad (8.36)$$

along with the partial derivatives of its components:

$$\frac{\partial x_{\mathbf{x}_1}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_1} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_1} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} \quad (8.37a)$$

$$\frac{\partial y_{\mathbf{x}_1}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_2} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_2} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} \quad (8.37b)$$

$$\frac{\partial w_{\mathbf{x}_1}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_3} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_3} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} \quad (8.37c)$$

$\mathbf{X}_2$  in the world coordinate system is similar to  $\mathbf{X}_1$ , with the difference that the  $x$ -axis of  $\mathbf{X}_2$  is translated by  $\pi_W$ . The result of  $\mathbf{x}_2$  in image coordinates is:

$$\mathbf{x}_2 = \begin{pmatrix} x_{\mathbf{x}_2} \\ y_{\mathbf{x}_2} \\ w_{\mathbf{x}_2} \end{pmatrix} = \begin{pmatrix} X_{\tilde{\mathbf{p}}_1}(X_{\mathbf{q}_0} + \pi_W) + Y_{\tilde{\mathbf{p}}_1}Y_{\mathbf{q}_0} + W_{\tilde{\mathbf{p}}_1} \\ X_{\tilde{\mathbf{p}}_2}(X_{\mathbf{q}_0} + \pi_W) + Y_{\tilde{\mathbf{p}}_2}Y_{\mathbf{q}_0} + W_{\tilde{\mathbf{p}}_2} \\ X_{\tilde{\mathbf{p}}_3}(X_{\mathbf{q}_0} + \pi_W) + Y_{\tilde{\mathbf{p}}_3}Y_{\mathbf{q}_0} + W_{\tilde{\mathbf{p}}_3} \end{pmatrix} \quad (8.38a)$$

$$\frac{\partial x_{\mathbf{x}_2}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_1} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_W \frac{\partial X_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_1} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} \quad (8.38b)$$

$$\frac{\partial y_{\mathbf{x}_2}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_2} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_W \frac{\partial X_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_2} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} \quad (8.38c)$$

$$\frac{\partial w_{\mathbf{x}_2}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_3} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_W \frac{\partial X_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_3} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} \quad (8.38d)$$

$$(8.38e)$$

In a similar way  $\mathbf{x}_3$  is calculated:

$$\mathbf{x}_3 = \begin{pmatrix} x_{\mathbf{x}_3} \\ y_{\mathbf{x}_3} \\ w_{\mathbf{x}_3} \end{pmatrix} = \begin{pmatrix} X_{\tilde{\mathbf{p}}_1}(X_{\mathbf{q}_0} + \pi_W) + Y_{\tilde{\mathbf{p}}_1}(Y_{\mathbf{q}_0} + \pi_H) + W_{\tilde{\mathbf{p}}_1} \\ X_{\tilde{\mathbf{p}}_2}(X_{\mathbf{q}_0} + \pi_W) + Y_{\tilde{\mathbf{p}}_2}(Y_{\mathbf{q}_0} + \pi_H) + W_{\tilde{\mathbf{p}}_2} \\ X_{\tilde{\mathbf{p}}_3}(X_{\mathbf{q}_0} + \pi_W) + Y_{\tilde{\mathbf{p}}_3}(Y_{\mathbf{q}_0} + \pi_H) + W_{\tilde{\mathbf{p}}_3} \end{pmatrix} \quad (8.39a)$$

$$\frac{\partial x_{\mathbf{x}_3}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_1} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_W \frac{\partial X_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_1} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_H \frac{\partial Y_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_1}}{\partial \vartheta} \quad (8.39b)$$

$$\frac{\partial y_{\mathbf{x}_3}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_2} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_W \frac{\partial X_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_2} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_H \frac{\partial Y_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_2}}{\partial \vartheta} \quad (8.39c)$$

$$\frac{\partial w_{\mathbf{x}_3}}{\partial \vartheta} = \left( X_{\mathbf{q}_0} \frac{\partial X_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} + X_{\tilde{\mathbf{p}}_3} \frac{\partial X_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_W \frac{\partial X_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} \right) + \left( Y_{\mathbf{q}_0} \frac{\partial Y_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} + Y_{\tilde{\mathbf{p}}_3} \frac{\partial Y_{\tilde{\mathbf{q}}_0}}{\partial \vartheta} + \pi_H \frac{\partial Y_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} \right) + \frac{\partial W_{\tilde{\mathbf{p}}_3}}{\partial \vartheta} \quad (8.39d)$$

It is not necessary to calculate the derivatives of  $\mathbf{x}_4$ , i.e., the fourth corner of the ROI, because

only points  $\mathbf{x}_1$ ,  $\mathbf{x}_2$ , and  $\mathbf{x}_3$  will be used to create the system of linear equations  $\mathcal{A} v_h = 0$  (Equation 7.6), used to find the matrix of projective transformation  $\mathcal{M}$  (Equation 7.2).

To finish with the calculation of the ROI corners, we divide all the components of each point by  $Z_{\mathbf{x}_k}$ . This way, we ensure that  $Z_{\mathbf{x}_k}$  is equal to one, and use the points in the image coordinate system:

$$\mathbf{x}_k = \begin{pmatrix} x_{\mathbf{x}_k}/w_{\mathbf{x}_k} \\ y_{\mathbf{x}_k}/w_{\mathbf{x}_k} \\ 1 \end{pmatrix}. \quad (8.40)$$

The partial derivatives of each component of  $\mathbf{x}_k$  (Equation 8.40), for  $k \in \{1, 2, 3\}$ , are:

$$\frac{\partial x_{\mathbf{x}_k}}{\partial \vartheta} = \frac{w_{\mathbf{x}_k} \frac{\partial x_{\mathbf{x}_k}}{\partial \vartheta} - x_{\mathbf{x}_k} \frac{\partial w_{\mathbf{x}_k}}{\partial \vartheta}}{w_{\mathbf{x}_k}^2} \quad (8.41a)$$

$$\frac{\partial y_{\mathbf{x}_k}}{\partial \vartheta} = \frac{w_{\mathbf{x}_k} \frac{\partial y_{\mathbf{x}_k}}{\partial \vartheta} - y_{\mathbf{x}_k} \frac{\partial w_{\mathbf{x}_k}}{\partial \vartheta}}{w_{\mathbf{x}_k}^2} \quad (8.41b)$$

$$\frac{\partial w_{\mathbf{x}_k}}{\partial \vartheta} = 0. \quad (8.41c)$$

## 8.8 Matrix of the Projective Transformation

Let  $\mathbf{H}$  be the homographic matrix, where  $A$ ,  $B$ , and  $C$  are the components of the vanishing line, and  $\mathbf{x}_{H_k}$  are the projected points on a homographic plane, where  $k \in \{1, 2, 3\}$ :

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{A}{C} & \frac{B}{C} & 1 \end{pmatrix} \quad (8.42)$$

Let  $\mathbf{x}_{H_1}$ ,  $\mathbf{x}_{H_2}$ , and  $\mathbf{x}_{H_3}$  be input parameters for the system of linear equations  $\mathcal{A} v_h = 0$ , shown in Equation 7.6, are defined as:

$$\mathbf{x}_{H_k} = \begin{pmatrix} x_{\mathbf{x}_{H_k}} \\ y_{\mathbf{x}_{H_k}} \\ w_{\mathbf{x}_{H_k}} \end{pmatrix} = \mathbf{H}_1 \mathbf{x}_k = \begin{pmatrix} x_{\mathbf{x}_k} \\ y_{\mathbf{x}_k} \\ x_{\mathbf{x}_k} \frac{A}{C} + y_{\mathbf{x}_k} \frac{B}{C} + 1 \end{pmatrix}, \quad (8.43)$$

where  $k \in \{1, 2, 3\}$ . The partial derivative of  $\frac{\partial x_{\mathbf{x}_{H_k}}}{\partial \vartheta}$ ,  $\frac{\partial y_{\mathbf{x}_{H_k}}}{\partial \vartheta}$  and  $\frac{\partial w_{\mathbf{x}_{H_k}}}{\partial \vartheta}$  is defined as:

$$\frac{\partial x_{\mathbf{x}_{H_k}}}{\partial \vartheta} = \frac{\partial x_{\mathbf{x}_k}}{\partial \vartheta} \quad (8.44a)$$

$$\frac{\partial y_{\mathbf{x}_{H_k}}}{\partial \vartheta} = \frac{\partial y_{\mathbf{x}_k}}{\partial \vartheta} \quad (8.44b)$$

$$\frac{\partial w_{\mathbf{x}_{H_k}}}{\partial \vartheta} = x_{\mathbf{x}_k} \left( \frac{C \frac{\partial A}{\partial \vartheta} - A \frac{\partial C}{\partial \vartheta}}{C^2} \right) + \frac{A}{C} \frac{\partial x_{\mathbf{x}_k}}{\partial \vartheta} + y_{\mathbf{x}_k} \left( \frac{C \frac{\partial B}{\partial \vartheta} - B \frac{\partial C}{\partial \vartheta}}{C^2} \right) + \frac{B}{C} \frac{\partial y_{\mathbf{x}_k}}{\partial \vartheta} \quad (8.44c)$$

, where the partial derivatives of  $\frac{\partial A}{\partial \vartheta}$ ,  $\frac{\partial B}{\partial \vartheta}$ , and  $\frac{\partial C}{\partial \vartheta}$  are given by the Equation 8.8.

The projective transformation matrix  $\mathcal{M}$  is the result of the product between the matrix  $\mathbf{H}_A$  and the matrix  $\mathbf{H}_I$ , where  $\mathbf{H}_A$  is defined by:

$$\mathbf{H}_A = \begin{pmatrix} v_{h_1} & v_{h_2} & v_{h_3} \\ v_{h_4} & v_{h_5} & v_{h_6} \\ 0 & 0 & v_{h_7} \end{pmatrix}, \quad (8.45)$$

and  $v_{h_i}$  is the result of the system of linear equations  $\mathcal{A} v_h = 0$  (Equation 7.6), which is used to create the matrix  $\mathbf{H}_A$  (Equation 8.45), where  $i = \{1, 2, \dots, 7\}$ . The partial derivatives of  $v_{h_i}$  are given by:

$$\frac{\partial \vec{v}_{h_i}}{\partial \vartheta} = \left( \frac{\partial \vec{v}_{h_1}}{\partial \vartheta} \dots \frac{\partial \vec{v}_{h_7}}{\partial \vartheta} \right)^T \quad (8.46)$$

The projective transformation matrix  $\mathcal{M}$  is defined by  $\mathcal{M} = \mathbf{T}_{v_h} \mathbf{H}$ , and rewritten as:

$$\mathcal{M} = \begin{pmatrix} \vec{\mathcal{M}}_1^T \\ \vec{\mathcal{M}}_2^T \\ \vec{\mathcal{M}}_3^T \end{pmatrix} = \begin{pmatrix} X_{\vec{\mathcal{M}}_1} & Y_{\vec{\mathcal{M}}_1} & Z_{\vec{\mathcal{M}}_1} \\ X_{\vec{\mathcal{M}}_2} & Y_{\vec{\mathcal{M}}_2} & Z_{\vec{\mathcal{M}}_2} \\ X_{\vec{\mathcal{M}}_3} & Y_{\vec{\mathcal{M}}_3} & Z_{\vec{\mathcal{M}}_3} \end{pmatrix} = \begin{pmatrix} \vec{v}_{h_1} + \vec{v}_{h_3} \frac{A}{C} & \vec{v}_{h_2} + \vec{v}_{h_3} \frac{B}{C} & \vec{v}_{h_3} \\ \vec{v}_{h_4} + \vec{v}_{h_6} \frac{A}{C} & \vec{v}_{h_5} + \vec{v}_{h_6} \frac{B}{C} & \vec{v}_{h_6} \\ \vec{v}_{h_7} \frac{A}{C} & \vec{v}_{h_7} \frac{B}{C} & \vec{v}_{h_7} \end{pmatrix} \quad (8.47)$$

The partial derivatives of each component of  $\vec{\mathcal{M}}_1$  are given by:

$$\frac{\partial X_{\vec{\mathcal{M}}_1}}{\partial \vartheta} = \frac{\partial \vec{v}_{h_1}}{\partial \vartheta} + \frac{\partial \vec{v}_{h_3}}{\partial \vartheta} \frac{A}{C} + \vec{v}_{h_3} \left( \frac{C \frac{\partial A}{\partial \vartheta} - A \frac{\partial C}{\partial \vartheta}}{C^2} \right) \quad (8.48a)$$

$$\frac{\partial Y_{\vec{\mathcal{M}}_1}}{\partial \vartheta} = \frac{\partial \vec{v}_{h_2}}{\partial \vartheta} + \frac{\partial \vec{v}_{h_3}}{\partial \vartheta} \frac{B}{C} + \vec{v}_{h_3} \left( \frac{C \frac{\partial B}{\partial \vartheta} - B \frac{\partial C}{\partial \vartheta}}{C^2} \right) \quad (8.48b)$$

$$\frac{\partial Z_{\vec{\mathcal{M}}_1}}{\partial \vartheta} = \frac{\partial \vec{v}_{h_3}}{\partial \vartheta} \quad (8.48c)$$

The partial derivatives of each component of  $\vec{\mathcal{M}}_2$  are given by:

$$\frac{\partial X_{\vec{\mathcal{M}}_2}}{\partial \vartheta} = \frac{\partial \vec{\mathbf{v}}_{h_4}}{\partial \vartheta} + \frac{\partial \vec{\mathbf{v}}_{h_6}}{\partial \vartheta} \frac{A}{C} + \vec{\mathbf{v}}_{h_6} \left( \frac{C \frac{\partial A}{\partial \vartheta} - A \frac{\partial C}{\partial \vartheta}}{C^2} \right) \quad (8.49a)$$

$$\frac{\partial Y_{\vec{\mathcal{M}}_2}}{\partial \vartheta} = \frac{\partial \vec{\mathbf{v}}_{h_5}}{\partial \vartheta} + \frac{\partial \vec{\mathbf{v}}_{h_6}}{\partial \vartheta} \frac{B}{C} + \vec{\mathbf{v}}_{h_6} \left( \frac{C \frac{\partial B}{\partial \vartheta} - B \frac{\partial C}{\partial \vartheta}}{C^2} \right) \quad (8.49b)$$

$$\frac{\partial Z_{\vec{\mathcal{M}}_2}}{\partial \vartheta} = \frac{\partial \vec{\mathbf{v}}_{h_6}}{\partial \vartheta} \quad (8.49c)$$

The partial derivatives of each component of  $\vec{\mathcal{M}}_3$  are given by:

$$\frac{\partial X_{\vec{\mathcal{M}}_3}}{\partial \vartheta} = \frac{\partial \vec{\mathbf{v}}_{h_7}}{\partial \vartheta} \frac{A}{C} + \vec{\mathbf{v}}_{h_7} \left( \frac{C \frac{\partial A}{\partial \vartheta} - A \frac{\partial C}{\partial \vartheta}}{C^2} \right) \quad (8.50a)$$

$$\frac{\partial Y_{\vec{\mathcal{M}}_3}}{\partial \vartheta} = \frac{\partial \vec{\mathbf{v}}_{h_7}}{\partial \vartheta} \frac{B}{C} + \vec{\mathbf{v}}_{h_7} \left( \frac{C \frac{\partial B}{\partial \vartheta} - B \frac{\partial C}{\partial \vartheta}}{C^2} \right) \quad (8.50b)$$

$$\frac{\partial Z_{\vec{\mathcal{M}}_3}}{\partial \vartheta} = \frac{\partial \vec{\mathbf{v}}_{h_7}}{\partial \vartheta} \quad (8.50c)$$

## 8.9 Find Peaks

Let  $\mathbf{c}_n = (x_{\mathbf{c}_n}, y_{\mathbf{c}_n}, w_{\mathbf{c}_n})^T$  and  $\mathbf{c}_{n+1} = (x_{\mathbf{c}_{n+1}}, y_{\mathbf{c}_{n+1}}, w_{\mathbf{c}_{n+1}})^T$  be two adjacent wave coordinates, extracted from the ROI image. These coordinates are taken as input variables without uncertainty, calculated in Chapter 7. The transformation matrix  $\mathcal{M}$  is used to rectify the two coordinates to  $\mathbf{c}'_n$  and  $\mathbf{c}'_{n+1}$ , respectively. The rectified coordinates are defined as:

$$\mathbf{c}'_p = \begin{pmatrix} x_{\mathbf{c}'_p} \\ y_{\mathbf{c}'_p} \\ w_{\mathbf{c}'_p} \end{pmatrix} = \mathcal{M} \mathbf{c}_p = \begin{pmatrix} x_{\mathbf{c}_p} X_{\vec{\mathcal{M}}_1} + y_{\mathbf{c}_p} Y_{\vec{\mathcal{M}}_1} + Z_{\vec{\mathcal{M}}_1} \\ x_{\mathbf{c}_p} X_{\vec{\mathcal{M}}_2} + y_{\mathbf{c}_p} Y_{\vec{\mathcal{M}}_2} + Z_{\vec{\mathcal{M}}_2} \\ x_{\mathbf{c}_p} X_{\vec{\mathcal{M}}_3} + y_{\mathbf{c}_p} Y_{\vec{\mathcal{M}}_3} + Z_{\vec{\mathcal{M}}_3} \end{pmatrix}, \quad (8.51)$$

where  $p \in \{n, n+1\}$ , and its partial derivatives are given by:

$$\frac{\partial x_{\mathbf{c}'_p}}{\partial \vartheta} = x_{\mathbf{c}_p} \frac{\partial X_{\vec{\mathcal{M}}_1}}{\partial \vartheta} + y_{\mathbf{c}_p} \frac{\partial Y_{\vec{\mathcal{M}}_1}}{\partial \vartheta} + \frac{\partial Z_{\vec{\mathcal{M}}_1}}{\partial \vartheta} \quad (8.52a)$$

$$\frac{\partial y_{\mathbf{c}'_p}}{\partial \vartheta} = x_{\mathbf{c}_p} \frac{\partial X_{\vec{\mathcal{M}}_2}}{\partial \vartheta} + y_{\mathbf{c}_p} \frac{\partial Y_{\vec{\mathcal{M}}_2}}{\partial \vartheta} + \frac{\partial Z_{\vec{\mathcal{M}}_2}}{\partial \vartheta} \quad (8.52b)$$

$$\frac{\partial w_{\mathbf{c}'_p}}{\partial \vartheta} = x_{\mathbf{c}_p} \frac{\partial X_{\vec{\mathcal{M}}_3}}{\partial \vartheta} + y_{\mathbf{c}_p} \frac{\partial Y_{\vec{\mathcal{M}}_3}}{\partial \vartheta} + \frac{\partial Z_{\vec{\mathcal{M}}_3}}{\partial \vartheta} \quad (8.52c)$$

## 8.10 Vessel Velocity Estimation

Once the two wave peaks are found, the vessel speed can be estimated by the distance between these points. Equation 2.16 estimates the speed of the vessel and is rewritten by Equation 8.53, where  $D$  is the Euclidean distance between the two points that represent the crests(or troughs) of the wave.

$$U = \sqrt{\frac{g\sqrt{3}}{4\pi}} \sqrt{D} \quad (8.53)$$

Knowing that  $g$  is the gravitational acceleration,  $r = \sqrt{\frac{g\sqrt{3}}{4\pi}}$  is taken as a constant. Replacing  $D$  (Equation 2.14) in Equation 8.53, we obtain the following equation:

$$U = r \sqrt[4]{(x_{\mathbf{c}'_{n+1}} - x_{\mathbf{c}'_n})^2 + (y_{\mathbf{c}'_{n+1}} - y_{\mathbf{c}'_n})^2 + (w_{\mathbf{c}'_{n+1}} - w_{\mathbf{c}'_n})^2} \quad (8.54)$$

By knowing the function  $U$  and the partial derivatives of the coordinates  $\mathbf{c}'_n$  and  $\mathbf{c}'_{n+1}$ , the derivative of the Jacobian matrix  $\nabla_U$  (Equation 8.5) is calculated by:

$$\begin{aligned} \frac{\partial U}{\partial \vartheta} = r & \frac{(x_{\mathbf{c}'_{n+1}} - x_{\mathbf{c}'_n}) \left( \frac{\partial x_{\mathbf{c}'_{n+1}}}{\partial \vartheta} - \frac{\partial x_{\mathbf{c}'_n}}{\partial \vartheta} \right) + (y_{\mathbf{c}'_{n+1}} - y_{\mathbf{c}'_n}) \left( \frac{\partial y_{\mathbf{c}'_{n+1}}}{\partial \vartheta} - \frac{\partial y_{\mathbf{c}'_n}}{\partial \vartheta} \right)}{2 \sqrt[4]{(x_{\mathbf{c}'_{n+1}} - x_{\mathbf{c}'_n})^2 + (y_{\mathbf{c}'_{n+1}} - y_{\mathbf{c}'_n})^2 + (w_{\mathbf{c}'_{n+1}} - w_{\mathbf{c}'_n})^2}} \\ & + \frac{(w_{\mathbf{c}'_{n+1}} - w_{\mathbf{c}'_n}) \left( \frac{\partial w_{\mathbf{c}'_{n+1}}}{\partial \vartheta} - \frac{\partial w_{\mathbf{c}'_n}}{\partial \vartheta} \right)}{2 \sqrt[4]{(x_{\mathbf{c}'_{n+1}} - x_{\mathbf{c}'_n})^2 + (y_{\mathbf{c}'_{n+1}} - y_{\mathbf{c}'_n})^2 + (w_{\mathbf{c}'_{n+1}} - w_{\mathbf{c}'_n})^2}} \quad (8.55) \end{aligned}$$

Once the matrix of variances and covariances  $\Lambda_\vartheta$  (Equation 8.3) is known, and the matrix  $\nabla_U$  is calculated, the error in the calculation of the vessel speed can be estimated from Equation 8.5.

## 8.11 Discussion

The computational chain to estimate the standard deviation in the calculation of the vessel speed was presented in this chapter. Also, the standard deviations of the intermediate variables shown in Figure 8.1 were defined.

The wavelength, described in Chapter 7, is calculated from a distance between two points. These points,  $\mathbf{c}'_n$  and  $\mathbf{c}'_{n+1}$ , are two adjacent peaks of the wave present in the rectified ROI. Points  $\mathbf{c}_n$  and  $\mathbf{c}_{n+1}$  represent points  $\mathbf{c}'_n$  and  $\mathbf{c}'_{n+1}$ , respectively, without rectification. We use matrix  $\mathcal{M}$  to rectify  $\mathbf{c}'_n$

and  $\mathbf{c}'_{n+1}$ , knowing that matrix  $\mathcal{M}$  has sources of calculated uncertainties along the computational chain.

All input data measured from the real world is a source of uncertainty. However, we did not consider the location of the initial ROI coordinate  $\mathbf{x}_1$ , ship direction  $\vec{\mathbf{u}}$ ,  $\mathbf{c}_n$ , and  $\mathbf{c}_{n+1}$  (points in the non-rectified ROI image) as sources of uncertainty in the proposed model. The reason was that we did not have how to estimate the uncertainties in these input variables. Camera height, vanishing line components,  $\mathbf{c}'_n$  and  $\mathbf{c}'_{n+1}$  points in the rectified ROI image were considered with uncertainty. The uncertainties, of  $\mathbf{c}'_n$  and  $\mathbf{c}'_{n+1}$ , came from the rectification that was incorporated into the homographic matrix  $\mathcal{M}$ . The matrix  $\mathcal{M}$  and the points  $\mathbf{c}_n$  and  $\mathbf{c}_{n+1}$  are important variables to determine the uncertainty of the vessel speed function.



# Chapter 9

## Analysis of the Results

To validate the techniques proposed in this work, it was developed a system that calculates the vessel speed from a single image. The project involved the use of a professional camera and the implementation of software, mentioned in Section 9.1.

Captured vessel images were taken with a digital camera. A radar estimates the captured vessel speed. Considering that the data used as input (e.g., coordinates of the vanishing line, camera height) are subject to an amount of error, it is expected that the estimated speeds are not accurate. By comparing the calculated values and the actual speeds, it is possible to have an idea of the accuracy and precision of the developed system. Accuracy can be defined as the proximity between the experimentally obtained value and the actual value. In other words, it is the measure of how much the obtained value approximates the correct value. Whereas the precision is the proximity between the values obtained by the repetition of the measurement process. In other words, the precision indicates the reproducibility of the result. Section 9.3 presents the analysis of the accuracy and precision of the system through statistical studies.

In this Chapter, we perform the analysis of:

- Accuracy of the proposed method,
- Precision of the proposed method; and
- The ability to applying the First-Order Error Propagation.

The materials used are described in Section 9.1. Section 9.2 shows a method to verify the accuracy. Section 9.3 shows a method to verify the precision. Results using error propagation are presented in Section 9.4.



(a) HSC vessel model, with an 19.20 *knot* speed.



(b) Another vessel type, with an 10.4 *knot* speed.

Figure 9.1: Examples of screenshots of the  $FAR - 21 \times 7$  radar monitor. The pictures were taken with a smartphone camera.

## 9.1 Materials

The images were taken using a Nikon D3300 camera, with 24.2 megapixels and encoded in JPG format file. The lens model used was an AF-S DX NIKKOR, with  $18 \sim 55$  mm focal length and with vibration reduction (VR II) [Nikon Corporation 2010]. The camera was mounted on a tripod. This tripod was placed on three different locations: the fourth floor, the fifth floor, and the sixth floor of the *Instituto de Computação* at UFF, facing the entrance of the Guanabara Bay (Figure 1.1).

In the experiments, the images of acquired moving vessels were under natural light conditions and different climatic situations. A total of 40 images were acquired, where 23 images were used to analyze the results presented in this chapter. The remaining images, 17 images, are shown in Appendix B. These images were not considered because they showed not very visible traces of the ship, and it was not possible to identify the wave arms. The resolution of the captured image  $\mathcal{I}$  was set to  $6000 \times 4000$  pixels. The Region of Interest (ROI) size lying on the plane  $\pi$  was set to  $18000 \times 9000$  cm. Given the scale factor  $\alpha = 10$ , the dimensions of the binary images  $\mathcal{B}'$  of the rectified ROIs was set to  $1800 \times 900$  pixels. Under this configuration, the side of, 1-pixel unit in the rectified image corresponds to 10 cm.

In Figures 9.2, 9.3, 9.4, and 9.5, the ship's actual speed is informed next to each image. Each image shows the real cropped picture, next to its respective edge image. The resolution of the real image and the edge-image are the same. The images were cropped to highlight the vessels and their trails.

In Figure 9.2, image  $i1$  shows the *HSC* vessel model, while images  $i2$  to  $i6$  show the *MC25* vessel model. In Figure 9.3, images  $i9$  and  $i10$  show the *HSC* vessel model. Images  $i8$ ,  $i11$ , and  $i12$

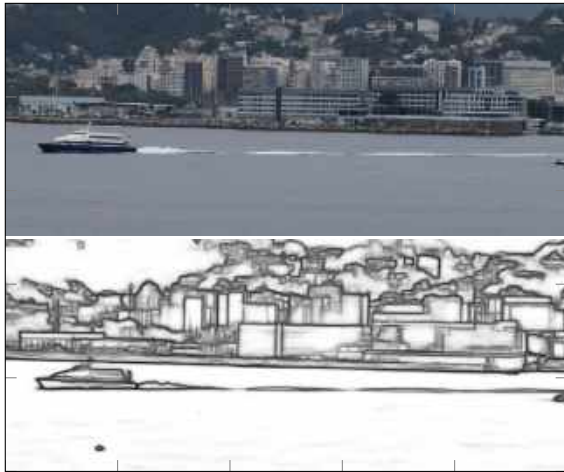
Table 9.1: Ground-truth information of the vessel in each captured image.

Image	Vessel Model	Vessel Name	Speed ( <i>knots</i> )	Time (hh:mm)	Weather	Tide Height (meters)
i1	<i>HSC</i>	Fenix	18.20	10:09	Cloudy	0.30
i2	<i>MC25</i>	Apolo	20.60	10:12	Cloudy	0.30
i3	<i>MC25</i>	Apolo	20.50	10:13	Cloudy	0.30
i4	<i>MC25</i>	Neptuno	19.20	10:19	Cloudy	0.50
i5	<i>MC25</i>	Neptuno	17.30	10:21	Cloudy	0.50
i6	<i>MC25</i>	Neptuno	17.10	10:22	Cloudy	0.50
i7	<i>Other</i>	Escander Amazonas	9.20	10:28	Cloudy	0.60
i8	<i>MC25</i>	Missing	19.50	10:39	Cloudy	0.70
i9	<i>HSC</i>	Fenix	15.60	10:41	Cloudy	0.70
i10	<i>HSC</i>	Fenix	16.50	10:43	Cloudy	0.70
i11	<i>MC25</i>	Missing	20.40	11:08	Cloudy	0.50
i12	<i>MC25</i>	Missing	20.40	11:09	Cloudy	0.50
i13	<i>MC25</i>	Missing	17.30	11:12	Cloudy	0.50
i14	<i>MC25</i>	Zeus	19.60	11:42	Cloudy	0.70
i15	<i>MC25</i>	Neptuno	19.10	11:43	Cloudy	0.70
i16	<i>MC25</i>	Neptuno	20.20	12:10	Cloudy	0.90
i17	<i>MC25</i>	Missing	20.30	12:13	Cloudy	0.90
i18	<i>MC25</i>	Zeus	18.90	16:12	Scattered storms	1.10
i19	<i>MC25</i>	Zeus	17.50	16:50	Partly cloudy	0.70
i20	<i>MC25</i>	Zeus	17.60	16:51	Partly cloudy	0.70
i21	<i>MC25</i>	Zeus	17.80	16:51	Partly cloudy	0.70
i22	<i>MC25</i>	Missing	20.30	17:01	Partly cloudy	0.70
i23	<i>MC25</i>	Zeus	16.80	17:21	Partly Cloudy	0.50

show the *MC25* vessel model. Only image *i7* is an unknown vessel model. In Figure 9.4, all images are of the *MC25* vessel model. All images were taken on the same day and at different times.

Table 9.1 shows, respectively, the vessel model, the vessel name, the true vessel speed ( $U$ , in knots), the time, the weather condition, and the tide height at which the image was taken. The name of the ship and its actual speed were extracted from a radar. Time was retrieved manually.

The radar installed in the *Instituto de Geociências* at UFF was used as a resource to validate the proposed method. The radar is a FAR – 21  $\times$  7 series of  $X$  and  $S$ -band, with a 19-inch LCD screen [Furuno Electric Co.]. The radar screen information was captured with a cell phone camera (see Figure 9.1). The radar screen contains *knots* speed information of the selected vessel in motion. Images taken from the radar screen and the moving vessel were acquired at approximately the same time. Table 9.1 shows the time at which both pictures were taken. The system returns the vessel speed in  $m/s$ . In order to compare with the resulting radar values, we used Equation 7.11 to



(i1)  $U \approx 18.20$  knots



(i2)  $U \approx 20.60$  knots



(i3)  $U \approx 20.50$  knots



(i4)  $U \approx 19.20$  knots



(i5)  $U \approx 17.30$  knots

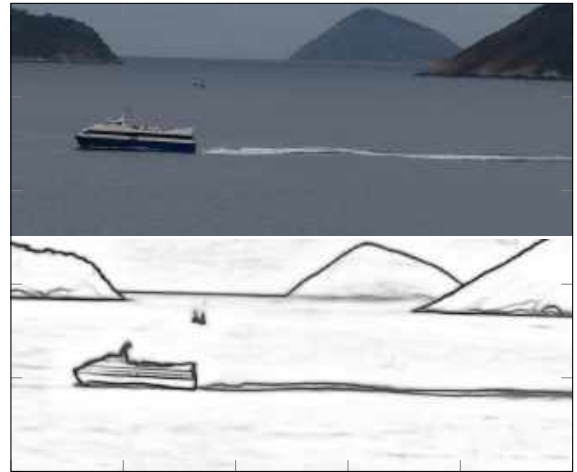


(i6)  $U \approx 17.10$  knots

Figure 9.2: Cropping the images used in the experiments. Image  $i1$  shows the a HSC vessel. Images  $i2 - i6$  show a MC25 vessel.



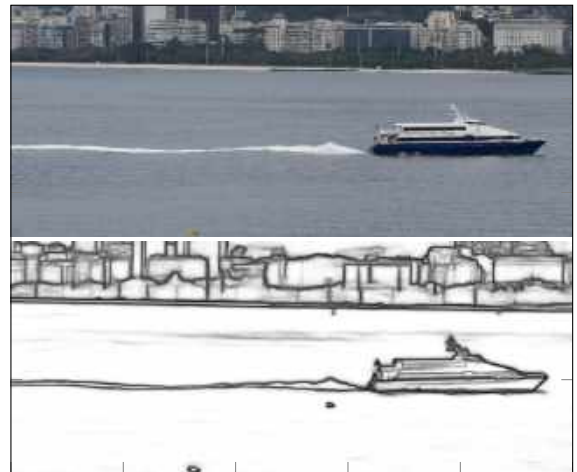
(i7)  $U \approx 9.20 \text{ knots}$



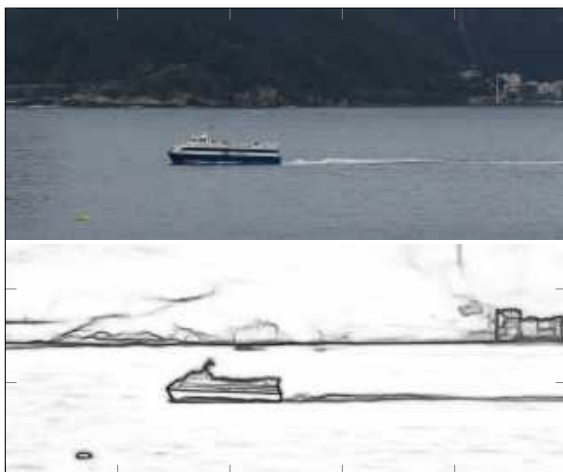
(i8)  $U \approx 19.50 \text{ knots}$



(i9)  $U \approx 15.60 \text{ knots}$



(i10)  $U \approx 16.50 \text{ knots}$

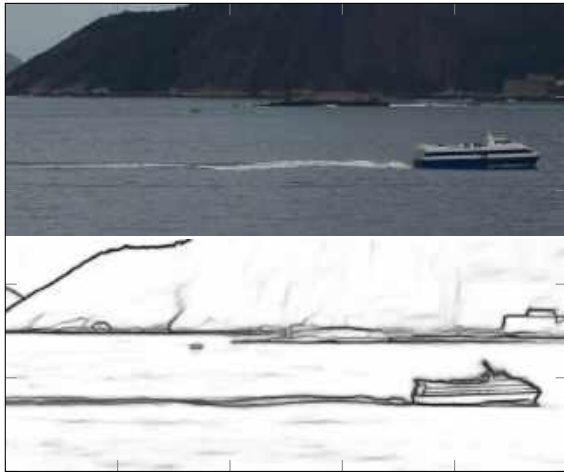


(i11)  $U \approx 20.40 \text{ knots}$



(i12)  $U \approx 20.40 \text{ knots}$

Figure 9.3: Cropping the images used in the experiments. Images  $i8$ , and  $i9$  show a HSC vessel. Images  $i7$ ,  $i10$ ,  $i11$ , and  $i12$  show a MC25 vessel.



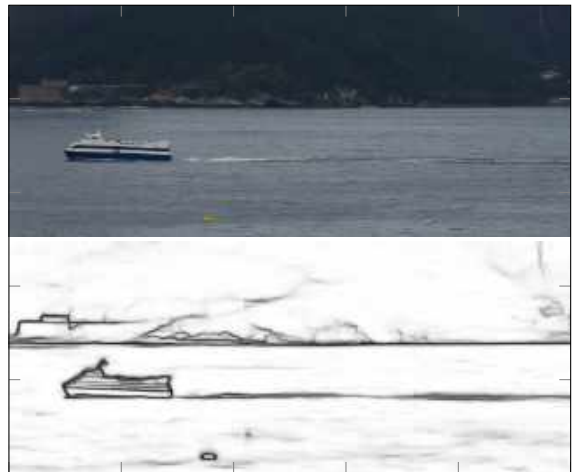
(i13)  $U \approx 17.30 \text{ knots}$



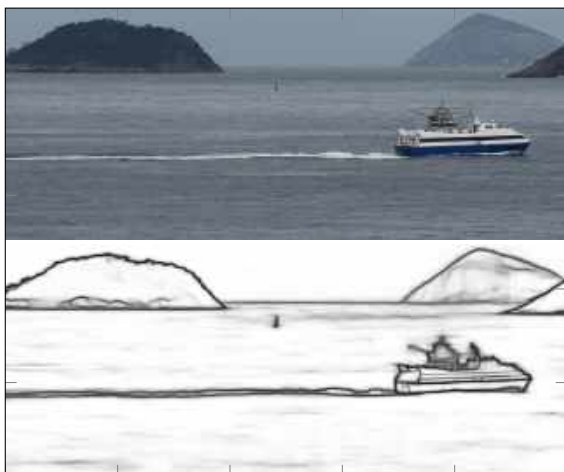
(i14)  $U \approx 19.60 \text{ knots}$



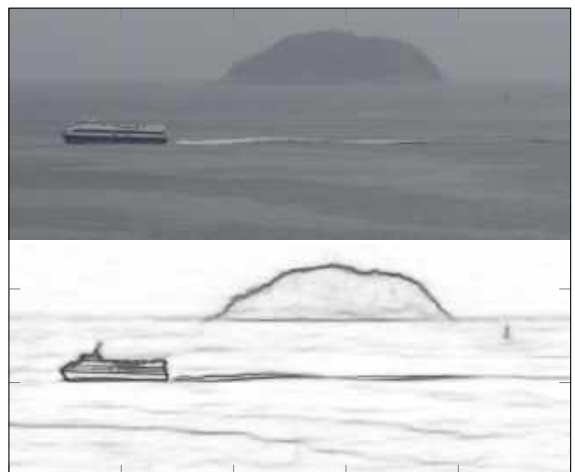
(i15)  $U \approx 19.10 \text{ knots}$



(i16)  $U \approx 20.20 \text{ knots}$



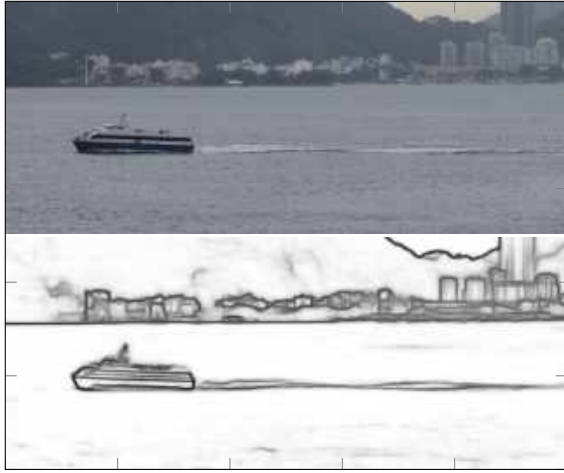
(i17)  $U \approx 20.30 \text{ knots}$



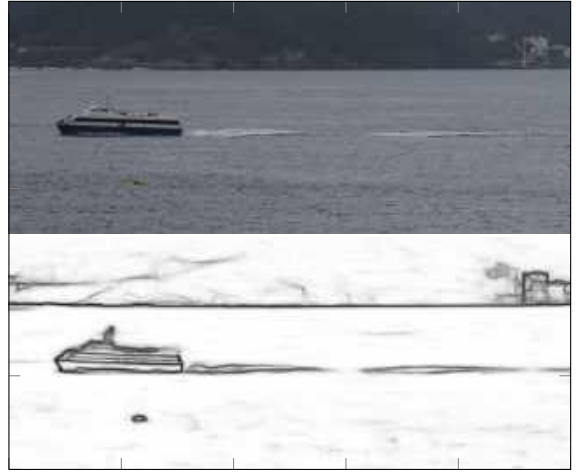
(i18)  $U \approx 18.90 \text{ knots}$

Figure 9.4: Cropping the images used in the experiments. All images sinclude a MC25 vessel.

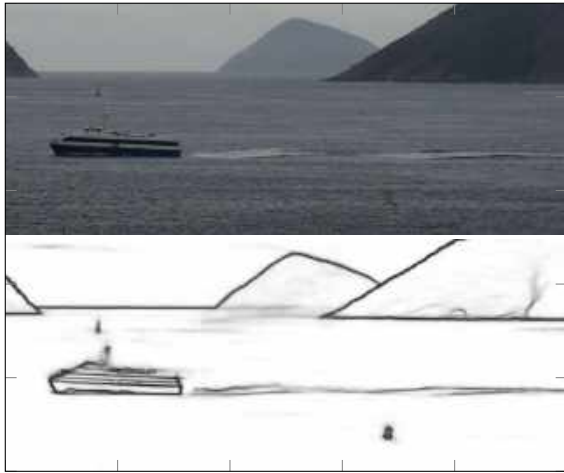




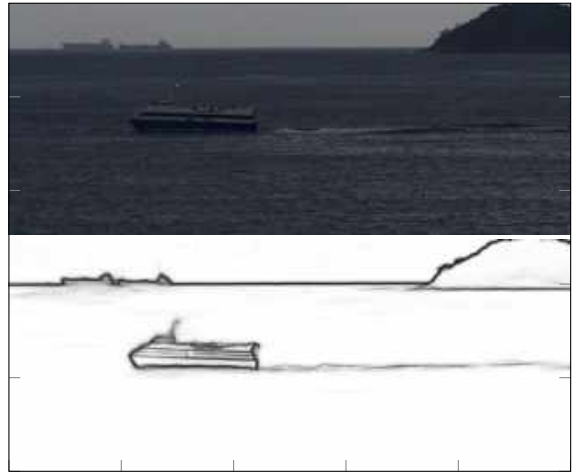
(i19)  $U \approx 17.50 \text{ knots}$



(i20)  $U \approx 17.60 \text{ knots}$



(i21)  $U \approx 17.80 \text{ knots}$



(i22)  $U \approx 20.30 \text{ knots}$



(i23)  $U \approx 16.80 \text{ knots}$

Figure 9.5: Cropping the images used in the experiments. All images sinclude a MC25 vessel.

transform from  $m/s$  to *knots*.

The images shown in Figures 9.2, 9.3, 9.4, and 9.5 belong to passenger's vessels, with the exception of image *i7*. These vessels follow the route that connects the ferry terminals, Praça XV, in Rio de Janeiro, and Charitas, in Niteroi, Brazil. It was not possible to use another type of ship because the radar was not able to display information on the speeds of small vessels, e.g., sailboats and yachts. Small vessels did not have Automatic Identification System (AIS), which helps display the ship's speed information on the radar screen. AIS is an automatic tracking system that uses transponders on vessels and is used by ship traffic services, where passenger and cargo ships are obliged to have them [Omar Systems Ltd. 2020]. We did not take cargo ships into account because the tracks left were very weak, and the proposed method did not correctly identify the wave arms. Also, the cargo ships had a speed fewer than 10 *knots*. This is because the cargo ship was close to a port area, and its speed had to be limited. The two models of passenger vessels operating on the observed route are MC25 and HSC.

Speeds were calculated on a computer with an Intel (R) Xeon (R) CPU E5-2698 v4 with 2.20 GHz, and a Tesla P100 – SXM2 video card with 16 GB of VRAM. The language used to implement the software was Python<sup>21</sup>. We used the Exif<sup>2</sup> module to extract the metadata from the images. The extracted metadata was used to construct the matrix of intrinsic parameters of the camera. Also, we used OpenCV<sup>3</sup> to work with different image processing functions, Scipy<sup>4</sup> library as a mathematical algorithm tool, and the Numpy<sup>5</sup> package to work with vectors and matrices. To work with error propagation, we used the First-Order Error Propagation Template Library (EPTL<sup>6</sup>) developed by ourselves.

## 9.2 Analysis of Relative Error

Relative error  $\varepsilon_r$  was used to determine the accuracy of the measurement,  $\varepsilon_r$  indicates the proportion of the error concerning the exact value of the measurement. It is necessary to know the expected value to calculate the relative error. Equation 9.1 is used to calculate the relative error, where  $U_m$  is the measured value, and  $U_e$  is the expected value.

$$\varepsilon_r = \frac{|U_m - U_e|}{U_e} \quad (9.1)$$

---

<sup>1</sup><https://github.com/Prograf-UFF/lighthouse>

<sup>2</sup><https://pypi.org/project/ExifRead/>

<sup>3</sup><https://pypi.org/project/opencv-python/>

<sup>4</sup><https://www.scipy.org/>

<sup>5</sup><https://numpy.org/>

<sup>6</sup><https://github.com/laffernandes/eptl>



Table 9.3: Table showing absolute error( $\varepsilon_a$ ) and relative error( $\varepsilon_r$ ) using the wave arms troughs.

Image	Expected value ( $U_e$ )	Measured value ( $U_m$ )	Absolute error ( $\varepsilon_a$ )	Relative error ( $\varepsilon_r$ )
i1	18.200	18.180	0.020	0.001
i2	20.600	22.404	1.804	0.088
i3	20.500	20.330	0.170	0.008
i4	19.200	19.582	0.382	0.020
i5	17.300	16.980	0.320	0.018
i6	17.100	16.453	0.647	0.038
i7	9.200	15.827	6.627	0.720
i8	19.500	19.158	0.342	0.018
i9	15.600	16.022	0.422	0.027
i10	16.500	16.401	0.099	0.006
i11	20.400	20.617	0.217	0.011
i12	20.400	20.664	0.264	0.013
i13	17.300	16.857	0.443	0.026
i14	19.600	20.974	1.374	0.070
i15	19.100	20.143	1.043	0.055
i16	20.200	17.957	2.243	0.111
i17	20.300	20.127	0.173	0.009
i18	18.900	19.485	0.585	0.031
i19	17.500	17.595	0.095	0.005
i20	17.600	18.023	0.423	0.024
i21	17.800	18.151	0.351	0.020
i22	20.300	16.802	3.498	0.172
i23	16.800	16.784	0.016	0.001

It is worth mentioning that in Equation 9.1, the numerator is also known as absolute error. Absolute error, defined as  $\varepsilon_a = |U_m - U_e|$ , is a measure of how 'far' a measurement is from an expected value.

In order to verify the accuracy of the proposed method, we calculated the relative error. We used 23 captured images for this, shown in Figures 9.2, 9.3, 9.4, and 9.5. Tables 9.3 and 9.5 show the image name, the expected value, the measured value, the absolute error, and relative error. The relative error has no units because they are canceled during the calculation. The relative error is a proportion, so we can express it as a percentage by multiplying the relative error by 100. The measured value is calculated on the image applying the proposed method, and the expected value is the measurement made by the radar.

In Table 9.3, we use the wave arms troughs. The troughs of the closest wave arm are the least affected by the noise introduced by the vessel's turbulence, and be the distortion of rectifying points

Table 9.5: Table showing absolute error( $\varepsilon_a$ ) and relative error( $\varepsilon_r$ ) using the wave arms crests.

Image	Expected value ( $U_e$ )	Measured value ( $U_m$ )	Absolute error ( $\varepsilon_a$ )	Relative error ( $\varepsilon_r$ )
i1	18.200	19.976	1.776	0.098
i2	20.600	21.461	0.861	0.042
i3	20.500	20.638	0.138	0.007
i4	19.200	20.860	1.660	0.086
i5	17.300	18.599	1.299	0.075
i6	17.100	16.927	0.173	0.010
i7	9.200	15.381	6.181	0.672
i9	15.600	19.582	3.982	0.255
i10	16.500	17.925	1.425	0.086
i13	17.300	17.633	0.333	0.019
i14	19.600	19.469	0.131	0.007
i17	20.300	21.694	1.394	0.069
i18	18.900	17.355	1.545	0.082
i19	17.500	15.944	1.556	0.089

that are not in the actual mean water plane. The results of the relative error were below 6%. Some results are between 6% and 17%, but they have, at most, one occurrence each. Only one image, *i7*, had a relative error equal to 72%. One possible explanation for this error is that the trace left behind was weak due to the 9.2 *knots* speed. It should be noted that the ship was close to a port area where the regulation limits the approach speed of larger vessels. The average of all relative errors obtained is equal to 6.48%, which shows that the proposed calculation method is accurate.

In Table 9.5, we work with the wave arm crests. The relative error was less than 10%, and only one case with a relative error equal to 25.5% was observed. We can also note that in images *i8*, *i11*, *i12*, *i15*, *i16*, *i20*, *i21*, *i22*, and *i23* it was not possible to obtain the vessel speed by our proposed method. The average relative error obtained was 11.407%, without taking into account the images where it was not possible to get the speed. The turbulence noise is a possible explanation for the significant relative error, and the vessel speed not estimated in some images.

### 9.3 Analysis of Confidence Intervals

In the experiments, it was not possible to make several captures of the vessels' images at the same time. To simulate the capture of various vessels' images, we made small variations for each image in the input parameters that were taken as a source of uncertainty, points that form the vanishing line and the camera height. These minor variations in each image were taken as samples.

Along with the image capture, annotations were made of tripod height, floor number where the camera was mounted, floor height above sea level, tide level height, and the time when the picture was taken for each picture. The height to each floor was obtained with the help of the architectural plans of the *Instituto de Computação*. The ground height above sea level was obtained from the Google Maps information. Nautide [Nautide], Tides Chart [Tides Chart], and *Marinha do Brasil* [Marinha do Brasil] were used to estimate sea level height. Annotations were also made of the time when the images of the radar screen were captured (Figure 9.1). It is essential to mention that the radar was in the *Instituto de Geociências* building, next to the *Instituto de Computação*. For that reason, it was necessary to write down the time vessels' images and radar screen images were taken. The annotated time format is *hh:mm* (hours:minutes). For each picture,  $n = 150$  samples were made, obtaining  $n$  speeds estimations using the method described in Chapter 7.

In this analysis, samples were used to calculate the mean, standard deviation, and confidence interval of each speed. Equation 9.2 gives the Confidence Interval (CI) for velocity,

$$CI(\gamma) = \left[ \bar{x} - t_{\gamma} \frac{s}{\sqrt{n}}, \bar{x} + t_{\gamma} \frac{s}{\sqrt{n}} \right], \quad (9.2)$$

where  $\bar{x}$  is the mean velocity and  $s$  is the standard deviation. In Equation 9.2,  $t_{\gamma}$  is a  $t$ -Student variable with  $n-1$  degrees of freedom so that the probability that velocity  $x$  belongs to the confidence interval is  $\gamma$ .

Table 9.7 and Table 9.9 show the confidence intervals calculated for each vessel image, using the wave arm troughs and the wave arm crests, respectively. The first column of the tables represent the name of the image. The second column shows the true speed in *knots* taken by the radar. The average velocity  $\bar{x}$  and standard deviation  $s$ , are shown in the third and fourth columns, respectively, followed by the confidence intervals (CI), where  $\gamma$  is equal to 95.0% ( $t_{\gamma} = 1.96$ ), 98.0% ( $t_{\gamma} = 2.33$ ), and 99.8% ( $t_{\gamma} = 3.08$ ), respectively.

The standard deviation  $s$  measures the dispersion of the values around the mean  $\bar{x}$ . The greater the standard deviation is, the less reproducible the estimated values and the less accurate the results. The standard deviation is expressed in the same unit of the data (*knots*), which facilitates its comparison. Observing Table 9.7, it is possible to notice that  $s$  varies between 0.19 and 1.50 *knots*. It is observed that for images with better natural lighting, the edges of the traces left by vessels are better (*i3*, *i6*, *i10*), and they have smaller standard deviations. In contrast, the standard deviations are higher for images with little natural lighting and taken very close at dusk (e.g., *i16*). The same applies to images taken in bad weather, especially *i18*. This image was captured when it was raining, leaving the trail of waves less visible. Images *i1* and *i9* have longer intervals because the wave crests are wide, making it difficult to find the troughs.

Table 9.7: Confidence intervals for estimated speeds in *knots*, using the wave arms troughs.

Image	True speed	$\bar{x}$	$s$	$CI(95, 0\%)$	$CI(98, 0\%)$	$CI(99, 8\%)$
i1	18.20	18.32	1.50	[18.08, 18.56]	[18.03, 18.61]	[17.94, 18.70]
i2	20.60	21.91	0.40	[21.85, 21.97]	[21.83, 21.99]	[21.81, 22.01]
i3	20.50	20.74	0.35	[20.68, 20.80]	[20.67, 20.81]	[20.65, 20.83]
i4	19.20	19.32	0.54	[19.23, 19.41]	[19.22, 19.42]	[19.18, 19.46]
i5	17.30	17.43	0.55	[17.34, 17.52]	[17.33, 17.53]	[17.29, 17.57]
i6	17.10	17.37	0.28	[17.33, 17.41]	[17.32, 17.42]	[17.30, 17.44]
i7	9.20	17.36	0.77	[17.24, 17.49]	[17.22, 17.51]	[17.17, 17.56]
i8	19.50	19.76	0.59	[19.67, 19.85]	[19.65, 19.87]	[19.61, 19.91]
i9	15.60	15.79	1.12	[15.61, 15.97]	[15.58, 16.00]	[15.51, 16.07]
i10	16.50	16.63	0.19	[16.60, 16.66]	[16.59, 16.67]	[16.58, 16.68]
i11	20.40	20.81	0.54	[20.72, 20.90]	[20.71, 20.91]	[20.67, 20.95]
i12	20.40	20.00	0.57	[19.91, 20.09]	[19.89, 20.11]	[19.86, 20.14]
i13	17.30	17.19	0.38	[17.13, 17.25]	[17.12, 17.26]	[17.09, 17.29]
i14	19.60	20.39	0.60	[20.29, 20.49]	[20.28, 20.50]	[20.24, 20.54]
i15	19.10	19.50	0.60	[19.40, 19.59]	[19.38, 19.61]	[19.35, 19.65]
i16	20.20	20.39	0.83	[20.26, 20.52]	[20.23, 20.55]	[20.18, 20.60]
i17	20.30	19.20	0.45	[19.13, 19.27]	[19.11, 19.29]	[19.09, 19.31]
i18	18.90	18.91	0.91	[18.77, 19.05]	[18.74, 19.08]	[18.68, 19.14]
i19	17.50	17.92	0.54	[17.83, 18.01]	[17.82, 18.02]	[17.78, 18.06]
i20	17.80	18.50	0.54	[18.41, 18.59]	[18.40, 18.60]	[18.36, 18.64]
i21	17.20	17.18	0.41	[17.11, 17.25]	[17.10, 17.26]	[17.08, 17.28]
i22	20.30	17.69	0.42	[17.62, 17.75]	[17.61, 17.77]	[17.58, 17.79]
i23	16.80	17.87	0.52	[17.79, 17.96]	[17.78, 17.97]	[17.74, 18.00]

Table 9.6 shows the confidence intervals for the wave arm crests. We can see that in only two images (*i16* and *i18*), the average value calculated by the proposed method is within the confidence interval. The average speeds calculated for images *i5*, *i9*, *i15*, and *i19* are the most distant from the true value, approximately 4.23 *knots*. For images *i7*, *i8*, *i11*, *i12*, *i20*, *i21*, *i22*, and *i23*, it was not possible to estimate their speeds, a possible explanation was that the crests were affected by the turbulence, mentioned in Section 9.2.

The standard deviation result is related to the confidence interval. The larger the standard deviation is, the higher the confidence interval will also be. Narrow confidence intervals indicate greater precision in the results. The first column of Figure 9.6 and Figure 9.7 help to illustrate the confidence intervals shown in Table 9.7 and Table 9.9, respectively. In these Figures, the red line represents the real vessel speed in increasing order. Blue lines indicate the average of the speeds calculated by the system, and the uncertainty bars the confidence intervals.

In our analysis, we have observed that the intervals (Table 9.7 and the first column of Figure 9.6)

Table 9.9: Confidence intervals for estimated speeds in *knots*, using the wave arms crests.

Images	True speed	$\bar{x}$	$s$	$CI(95, 0\%)$	$CI(98, 0\%)$	$CI(99, 8\%)$
i1	18.20	19.31	1.72	[19.04, 19.59]	[18.99, 19.64]	[18.88, 19.74]
i2	20.60	21.71	0.45	[21.63, 21.78]	[21.62, 21.79]	[21.59, 21.82]
i3	20.50	21.38	0.96	[21.23, 21.53]	[21.20, 21.56]	[21.14, 21.62]
i4	19.20	20.23	0.50	[20.15, 20.31]	[20.14, 20.33]	[20.11, 20.36]
i5	17.30	19.58	1.20	[19.39, 19.77]	[19.35, 19.81]	[19.28, 19.88]
i6	17.10	16.76	0.96	[16.61, 16.91]	[16.58, 16.94]	[16.52, 17.00]
i9	15.60	19.83	0.64	[19.73, 19.93]	[19.71, 19.95]	[19.67, 19.99]
i10	16.50	18.06	1.15	[17.88, 18.24]	[17.84, 18.28]	[17.77, 18.35]
i13	17.30	17.78	1.01	[17.62, 17.94]	[17.59, 17.97]	[17.53, 18.04]
i14	19.60	19.51	0.43	[19.44, 19.58]	[19.43, 19.59]	[19.40, 19.62]
i15	19.10	17.89	2.00	[17.57, 18.21]	[17.51, 18.27]	[17.39, 18.40]
i17	20.30	21.68	0.47	[21.60, 21.75]	[21.59, 21.77]	[21.56, 21.80]
i18	18.90	18.44	2.73	[18.00, 18.87]	[17.92, 18.96]	[17.75, 19.12]
i19	17.50	15.62	0.23	[15.59, 15.66]	[15.58, 15.67]	[15.56, 15.68]

are not so narrow, and most part of the confidence intervals does not incorporate true speeds. The images whose speeds are within the confidence interval are *i1*, *i4*, *i5*, *i9*, *i13*, *i15*, and *i18*. Images *i2*, *i22* and *i7* show a greater distance between actual speed and average speed. The difference between the actual speed and the average speed are approximately 1.6, 3.5, and 8.4 *knots*, to *i2*, *i22* and *i7* respectively. A possible explanation for the result of *i2* is that the wake and the vessel passed close to an oil platform, altering the shape of the waves since there were other waves coming from the platform disturbing the water surface. In the edge image *i22*, the traces left by the ship were weak. This is because it was taken in poor natural light. In image *i7*, the vessel's traces were also weak, but this was due to the speed the ship was going. For the remaining images, although the actual speed was not within the confidence interval, the average speeds were close, approximately 0.36 *knots* on average.

## 9.4 Analysis of the Error Propagation

In this section it is developed an analytical study on the propagation of errors along the computational chain presented in Figure 8.1.

The coordinates that form the vanishing line and the camera's height are determined from a set of experimental data. The error  $\eta$  is given by the difference between the experimental value  $x$  and the true value  $x_v$  [Vuolo 1996]. Where the true value  $x_v$  is unknown, and the error  $\eta$  is also

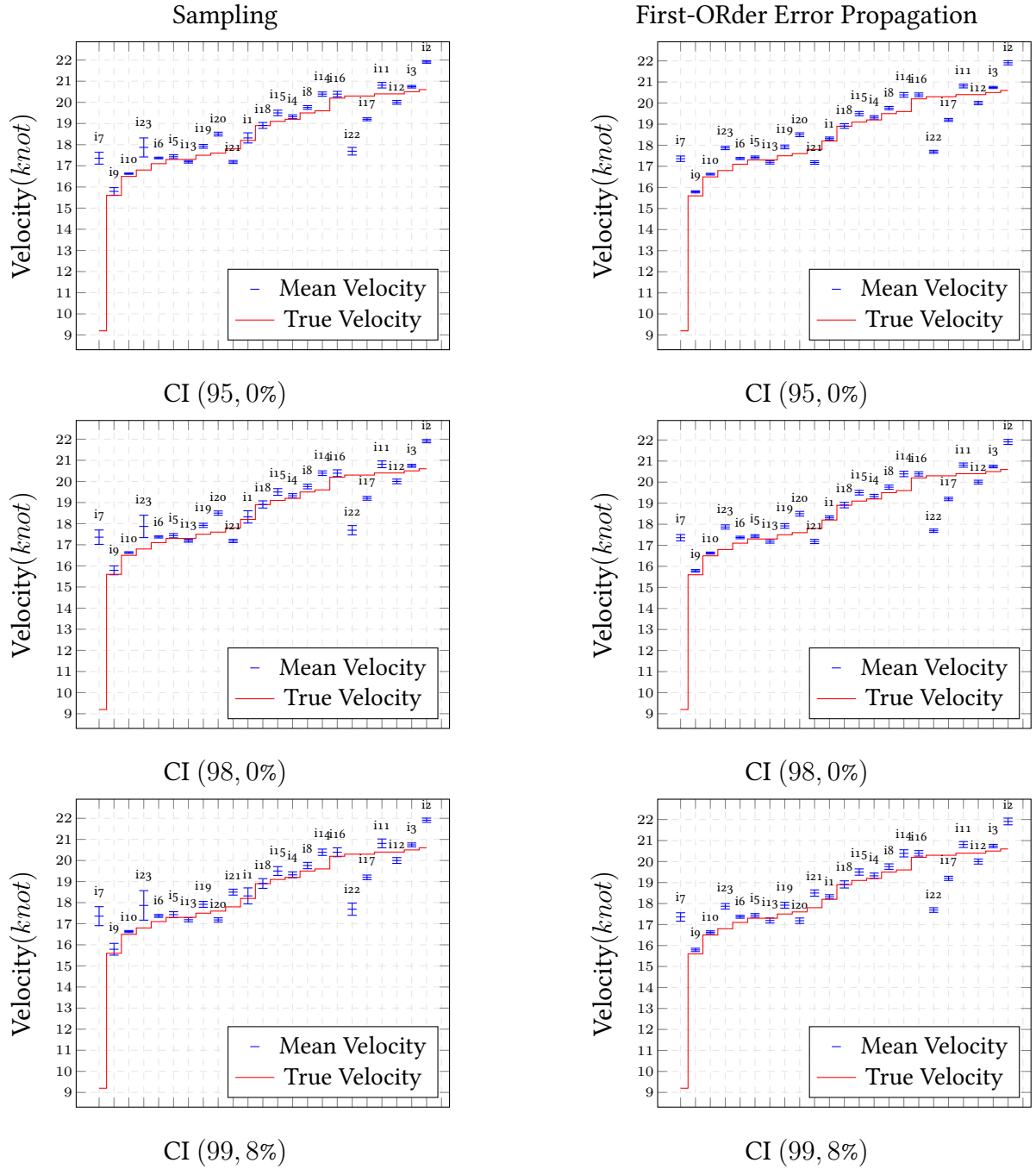


Figure 9.6: The first column shows the confidence intervals (CI), presented in Table 9.7, using samples randomly created. The second column shows the result of the confidence intervals using error propagation. The wave arm troughs were used to create the confidence intervals.

unknown. Therefore,  $\eta$  can only be estimated in terms of probabilities:

$$\eta = x - x_v \quad (9.3)$$

The confidence interval for a true value  $x_v$ , around an experimental value, is defined as:

$$x - \sigma < x_v < x + \sigma \quad (9.4)$$

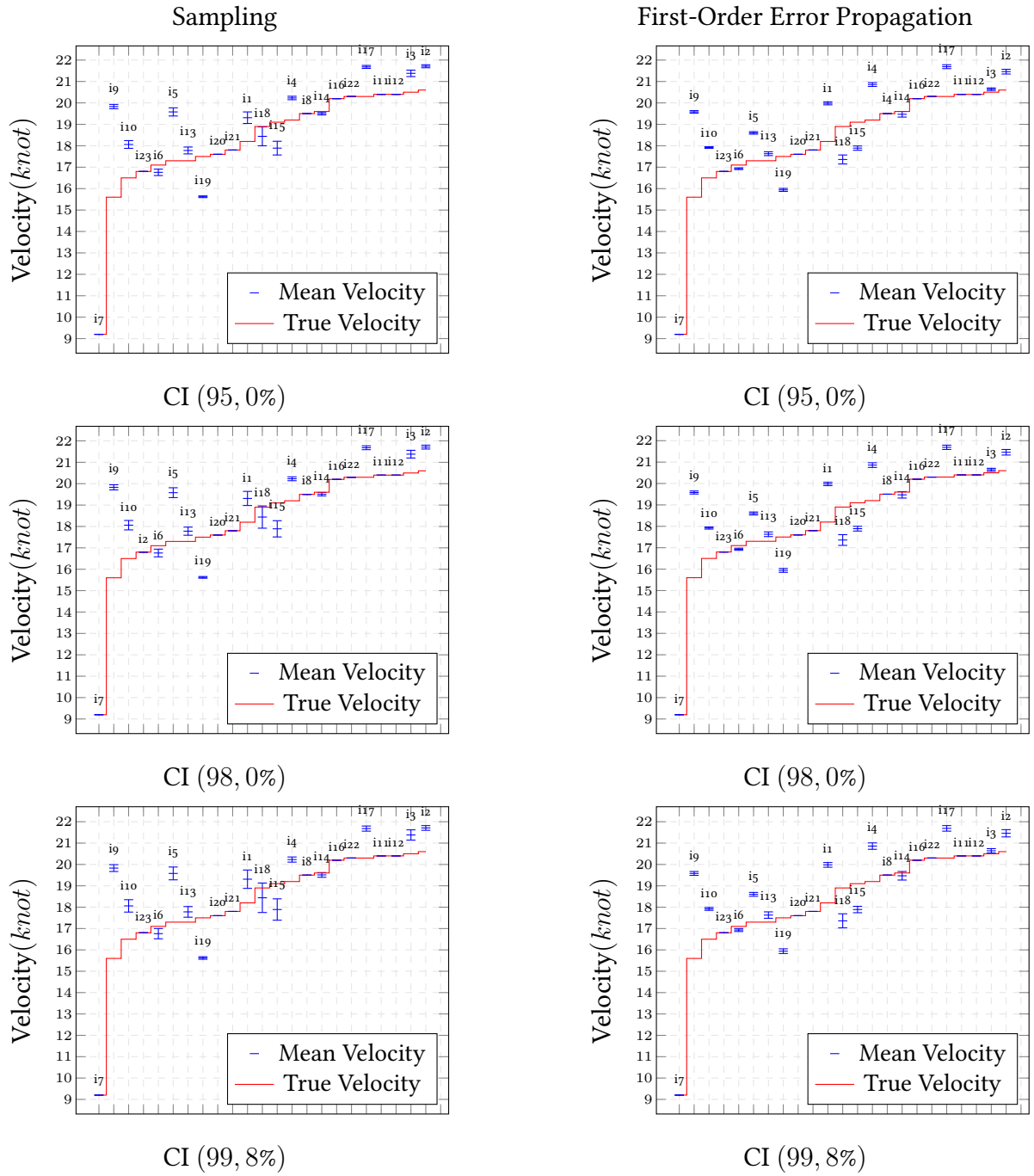


Figure 9.7: The first column shows the confidence intervals (CI), presented in Table 9.9, using samples randomly created. The second column shows the result of the confidence intervals using error propagation. The wave arm crests were used to create the confidence intervals.

Interpreting Equation 9.4, we can say that there is a probability  $P$  of  $x_v$  being within the range, for  $\sigma \geq 0$ . Considering that the distribution of errors is Gaussian and that standard uncertainty is obtained experimentally.

### 9.4.1 Errors in Input Data

The input data of the system are the vessel image, coordinates that form the vanishing line, camera height, intrinsic parameters of the camera, an initial ROI coordinate, the direction of the vessel, ROI size (in meters), and two adjacent crests/troughs of the wave. In this work, we assume that the last five input parameters are values unaffected by uncertainties. Two coordinates of the vanishing line, which represents the horizon, are extracted from the image. These coordinates are calculated using the Horizon Line of the Wild (HLW) algorithm, explained in Chapter 5. Due to the discretization, lighting, weather conditions, and image resolution, the coordinates are estimated with small variations in position. The camera height is subject to variations due to external influences, such as tide height, instrument defects, and in the measurement process. All these errors in the input data will influence the estimated vessel speed.

The coordinates to obtain the vanishing line, together with the camera height, are associated with an error. Both values used as input parameters have different ways of estimating their error. These errors are used in the composition of matrix  $\Lambda_\theta$  (Equation 8.3).

The vanishing line is created from two coordinates, where the  $y$ -axis is taken as the input variable (recall Section 4.2). The left and right axis  $v_l$  and  $v_r$ , respectively, are obtained from the HLW algorithm. Due to discretization, lighting, weather conditions, and image resolution, the true values of  $v_l$  and  $v_r$  are in the range of  $v_i \pm s_{v_i}$ , for  $i$  in  $(l, r)$ . Thus, the matrix of variances and covariances of the coordinates of the vanishing line is given by Equation 9.5:

$$\Lambda_{v_i} = \begin{pmatrix} s_{v_l}^2 & 0 \\ 0 & s_{v_r}^2 \end{pmatrix}. \quad (9.5)$$

The camera height is given by the sum of the floor height, ground height, tripod height, and the subtraction of the tide height. The standard deviation calculation for each input parameter is described in Appendix A. The matrix of variances, and covariances for the camera height is given by:

$$\Lambda_{h_j} = \begin{pmatrix} s_{h_g}^2 & 0 & 0 & 0 \\ 0 & s_{h_f}^2 & 0 & 0 \\ 0 & 0 & s_{h_t}^2 & 0 \\ 0 & 0 & 0 & s_{h_s}^2 \end{pmatrix} \quad (9.6)$$

The image size of the rectified ROI  $\pi_w, \pi_h$  and the initial coordinate of the ROI,  $\mathbf{X}_1$ , are not included as a source of uncertainty.  $\pi_w$  and  $\pi_h$  are a constant value defined manually. Also, Fernandes et al. [Fernandes, Oliveira e Silva 2008] showed that in an experiments where the impact of the uncertainty of each input variable was calculated for the estimation of the size of boxes from images,



Table 9.11: Normality Tests using Shapiro-Wilk. If  $\rho - value > \alpha$  then the null hypothesis cannot be rejected, where  $\alpha = 0.05$ . It was used the wave arm troughs for the Normality Test.

Images	Shapiro-Wilk W	$\rho$ -value	Images	Shapiro-Wilk W	$\rho$ -value
i1	0.899	$9 \times 10^{-9}$	i13	0.987	0.151
i2	0.987	0.175	i14	0.987	0.151
i3	0.982	0.062	i15	0.980	0.051
i4	0.990	0.400	i16	0.907	$3 \times 10^{-8}$
i5	0.983	0.051	i17	0.990	0.440
i6	0.998	0.999	i18	0.993	0.648
i7	0.865	$5 \times 10^{-7}$	i19	0.969	<b>0.002</b>
i8	0.988	0.252	i20	0.985	0.106
i9	0.862	$1 \times 10^{-10}$	i21	0.985	0.084
i10	0.990	0.426	i22	0.913	$7 \times 10^{-8}$
i11	0.983	0.051	i23	0.641	$2 \times 10^{-16}$
i12	0.987	0.180			

the intrinsic parameters of camera **K** carry insignificant uncertainty, so we assume that observation in this work and did not include the intrinsic parameters of the camera as a source of uncertainty.

#### 9.4.2 Verifying the Use of Error Propagation

The input variables used in the sample procedure described in Section 9.3 were created randomly following a Gaussian distribution. In this way, we guarantee that the entry of the uncertainties is Gaussian. To verify if the output is also Gaussian, we use the Shapiro-Wilk [SHAPIRO e WILK 1965] test.

The Gaussian distribution is widely used to describe experimental errors. It is also called the normal error function [Vuolo 1996]. The Central Limit Theorem justifies the Gaussian function as an error distribution. Where, if the total error  $\eta$  is the sum of several  $\eta_i$  errors that have any distribution with finite variances, the probability distribution tends to be Gaussian. In order to use first-order error propagation, it is necessary to verify if the sample's result of each image is approximately normally distributed. The normal distribution peaks in the middle and is symmetric about the mean. The data does not need to be correctly distributed normally for the tests to be reliable.

The normality test chosen is the most powerful in most situations [Shapiro e Wilk 2007]. The null hypothesis of this test is: the data is normally distributed, if  $\rho > \alpha$ . Therefore, if the  $\rho$ -value is less than the chosen  $\alpha$  level, then the null hypothesis is rejected, and there is evidence that the tested data is not normally distributed. On the other hand, if the  $\rho$ -value is higher than the chosen  $\alpha$  level, then the null hypothesis that the data comes from a normally distributed population cannot

Table 9.14: Normality Tests using Shapiro-Wilk. If  $\rho - value > \alpha$  then the null hypothesis cannot be rejected, where  $\alpha = 0.05$ . It was used the wave arm crests for the Normality Test.

Images	Shapiro-Wilk		Images	Shapiro-Wilk	
	W	$\rho$ -value		W	$\rho$ -value
i1	0.916	$1 \times 10^{-7}$	i10	0.365	$1 \times 10^{-16}$
i2	0.958	<b>0.007</b>	i13	0.979	0.051
i3	0.958	$4 \times 10^{-5}$	i14	0.981	0.051
i4	0.985	0.091	i15	0.94649	$1 \times 10^{-5}$
i5	0.859	$1 \times 10^{-10}$	i17	0.984	0.053
i6	0.983	0.087	i18	0.932	$1 \times 10^{-6}$
i9	0.634	$1 \times 10^{-16}$	i19	0.983	0.082

be rejected based on the data available.

Table 9.11 shows the results of the samples of each image applying the Shapiro-Wilk test, where the wave arm troughs were used. The chosen level of significance is  $\alpha = 0.05$ . Recalling the null hypothesis, if the  $\rho$ -value is higher than  $\alpha$ , normality can be assumed. In Table 9.11,  $\rho$ -values less than  $\alpha$  is highlighted in red. Images *i1*, *i8*, *i13*, and *i16* have  $\rho$ -values equal to  $9 \times 10^{-9}$ ,  $1 \times 10^{-10}$ ,  $3 \times 10^{-8}$ , and 0.002, respectively, which suggest strong evidence of non-normality in those cases. A possible reason for images *i1*, *i8*, and *i13* that do not follow a normal distribution is the size of the confidence interval, seen in Section 9.3, showing that the results in those images have low precision.

Table 9.14 shows the results of the normality test applied to the wave arm crests. Only images *i4*, *i6*, *i13*, *i14*, *i17*, and *i19* present a Gaussian distribution. In images *i7*, *i8*, *i11*, *i12*, *i16*, and *i20* – *i23*, it was impossible to apply the normality test because the vessel speeds were not estimated. The wave arm crests were not a good source for calculating the vessel speed.

Recall from Chapter 8 that, by knowing that the samples of each image follow a Gaussian distribution, it is possible to use first-order error propagation for the wave arm troughs.

### 9.4.3 Analyzing the Results

Applying the error propagation on various vessel images shown in the Figures 9.2, 9.3, 9.4, and 9.5, it was observed that the use of the standard deviation in the calculation of the Equation 9.4 is sufficient to create confidence intervals that include or are very close to the actual speed, where the actual value is far away from the interval, as seen in the second column of Figure 9.6. Other confidence intervals are approximately 0.32 knots from the actual speed. In total, 72.2% of confidence intervals do not incorporate actual speed. This result is similar to that obtained in the statistical analysis presented in Section 9.3. The confidence intervals where the standard deviations are most considerable

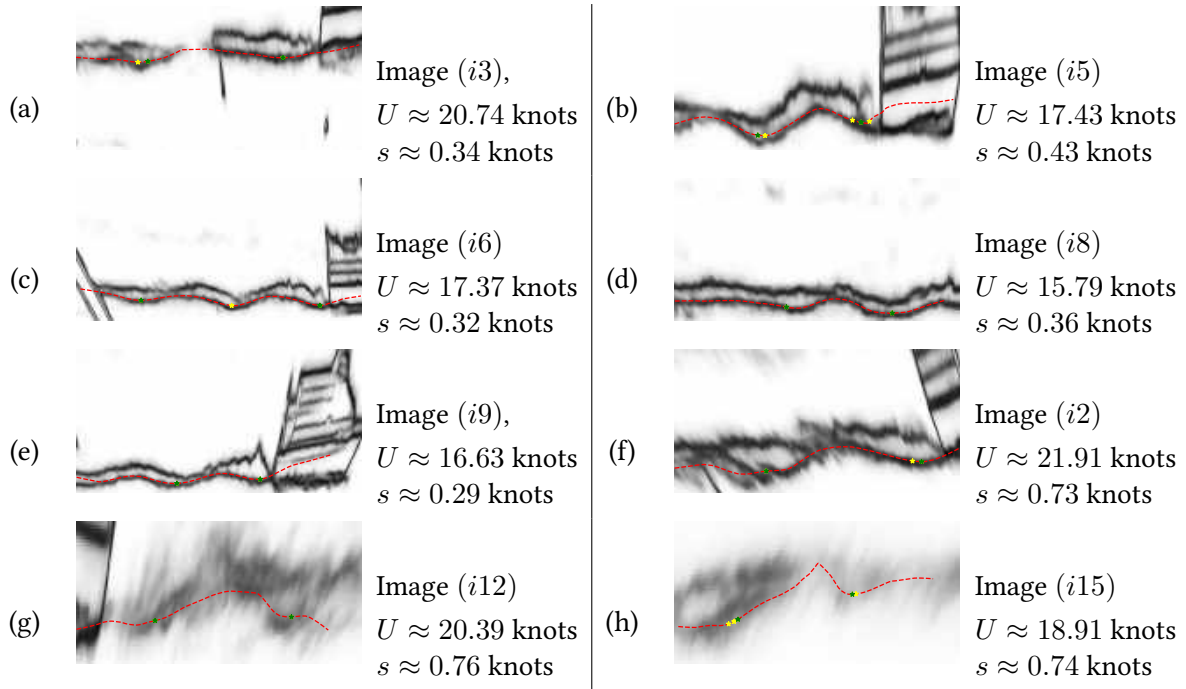


Figure 9.8: Images with smaller standard deviation (a-e), so the confidence interval is narrower. The images with higher standard deviation are presented in (f, g, h). Here,  $U$  is the estimated speed and  $s$  is the standard deviation calculated using First-Order Error Propagation.

are in the  $i7$ ,  $i22$ , and  $i23$  images. We believe that this is because these images did not pass the normality test. We can deduce that the result of the confidence intervals is an approximation.

With the help of error propagation, it is easy to detect cases where greater uncertainty is generated in the calculated speeds. Using a single image to calculate confidence intervals makes error analysis easier.

## 9.5 Discussion

The results of the method applied to calculate the speeds proposed in this work were presented in this chapter. The hardware and software used to estimate and validate the method were presented. Statistical studies and an analytical study of the error propagation in the input data for the calculated speeds were also presented. We used 23 images, of the 40 acquired, for the analysis of the results. The remaining 17 images are shown in Appendix B.

The initial ROI coordinate is an important parameter to estimate the vessel speed. This coordinate is responsible for placing the ROI in the position where the traces of the ship. The ROI has to cover a large part of the path left by the ship. Tidal altitude also plays an essential role in calculating speed. This variable is responsible for modifying the camera height. Speed is estimated from the edge image, where the trail left by the ships is visible. Image edge detection is the heaviest process

in the system. In the edge image, it is observed that for smaller or slower vessels, the tracks are fragile and not visible, making it difficult to estimate their speed.

Lighting conditions play an important role in estimating the vessel speed. If the ambient lighting is good, there are more possibilities to capture the traces of the ship, and consequently, to better distinguish its waves. To estimate the vessel speed, it is necessary that the trail is visible. Problems with overexposure or underexposure with any object are not covered in this manuscript.

As can be seen in the Figures 9.2, 9.3, 9.4, and 9.5 used in the experiments, the vessels are taken laterally, following a straight line. If the vessel does not follow a straight line and its trace is visible, the  $\vec{u}$  direction (showed in Seccion 5.2.2) must be modified, as well as the ROI. In this manuscript, we used vessels that followed a straight line for the experiments. Also, we assumed that the vessel had a constant speed.

The method proved to have moderate accuracy and good precision. The accuracy and precision of the system were verified by applying statistical analysis on a set of vessel images. The images were taken with different sources of natural light, different climatic conditions (with rain and wind, the latter being responsible for affecting the tide height). The average relative error for the vessel, using the wave arm troughs was 6.480%.

Applying the Error Theory, an analytical method was presented for the study of how the errors of the input parameters are propagated to calculate the speed. Confidence intervals were calculated from a single image. The application of error propagation showed similar results to the statistical analysis. For that reason, there was an advantage of not requiring to capture multiple images under the same conditions. With the help of error propagation, it was possible to identify situations with more significant uncertainty. One problem found was the resolution of the waves present in the ROI image. It is proposed to include the ROI image resolution as one of the sources of error in the samples, thus improving the confidence intervals to contain the actual speeds.

# Chapter 10

## Conclusion and Future Work

This dissertation presented a method for estimating vessel velocity from single perspective projection images. The approach uses planar homography to remove perspective distortion from images of traces of a moving vessel, curve fitting and peak detection to identify crests/troughs in the cusp wave arms, and natural constraints on components of Kelvin wakes to estimate wavelength and compute vessel velocity. Experimental results comparing estimated velocities to truth data have demonstrated the accuracy of our approach.

In this last chapter, the main points presented throughout this dissertation are listed. Also, a discussion is presented on the advantages and limitations of the proposed approach. Finally, suggestions for future work are listed.

Four problems must be solved to estimate the vessel speed from a single image: (1) identify the region of interest that contains the trace of the vessel; (2) Identify at least two adjacent crests or troughs of the wave included in the region of interest; (3) Remove perspective distortion from images of traces of moving vessels; and (4) Estimate the vessel speed from a single optical image.

To solve the first problem, we manually select an initial corner of the region of interest as a reference and indicate the direction of the vessel in the image space, making sure that the region of interest will contain the trace of the ship. The region of interest must be very close to the ship, because the trails are stronger, which will make it easier to find the correct wave.

Once the region of interest has been identified, the wave arms will be classified, and a wave arm closer to the observer will be chosen. The closest wave arm was chosen because it was observed that the wave profile is less affected by turbulence, making the crests/troughs more visible. Knowing the chosen wave arm, two adjacent wave crests/troughs were found, thus solving the second problem.

The problem of perspective distortion, the third problem, was solved with planar homography. The distance between two crests/troughs is calculated using a Euclidean distance. We used the

wavelength, along with the Kelvin wave structure to estimate the vessel speed, solving the fourth problem. A scalar factor  $\alpha = 10$  was used to convert the wavelength distance from centimeters to meters. The result was converted to knots to validate the proposed method with the radar.

The quality of the results was verified by the application of statistical analysis on the estimated speeds. Statistical analysis reveals that speeds are accurate and precise. It was not possible to make several captures of the same vessel at the same time, so we created small variations in the input variables as a sample to define the confidence intervals. The expected average relative error was 6.480%. For this, 23 images were used. The error propagation along the computational chain was also relevant to provide the user with reliable intervals that give a notion of quality in the estimated speeds made from an image.

The vessels used are shown in Figures 9.2, 9.3, 9.4, and 9.5. We assume that those vessels follow a linear path at constant speed. Taking the image of the vessel laterally helps to visualize the traces left behind. The vessel speed coming from different directions can also be calculated using the proposed technique, where one restriction is that the trails of the vessels have to be visible. In the experiments, vessels that followed another direction were not used.

A limitation of the technique is when the traces of the vessels are over-exposed or under-exposed. If any object occludes the traces of the vessel, our method will not be able to detect the wave crests/troughs correctly.

Another limitation that exists is the distance between the camera and the traces of the vessels. If the trace of the vessel is not visible due to the camera distance, it will be difficult to estimate its speed. This is directly affected by the resolution image and the camera lens. Besides, vessels with a speed of fewer than 10 knots have very weak and difficult traces to detect. Weather conditions and natural light play an important role in detecting the traces of the vessels correctly.

We assume that the vessel speed is constant in the presented image, but in reality, the vessel speed is not always constant. The acceleration of the vessel can be seen if several images of the same vessel are taken at different times, the acceleration will be observed if the speeds estimated by our method change from one image to another.

The proposed method is able to estimate the vessel speed from a single image. The proposed method is not entirely robust in the presence of different weather conditions and different ambient lighting, i. e., at night or on a stormy day, where the waves are not visible. The proposed method proved to be accurate and precise, seen in the Results Analysis Chapter. The proposed method requires user intervention to choose the initial coordinate of the ROI, even though it is a challenge to calculate it automatically.

## 10.1 Future Works

Throughout the development of this work, issues were identified that would serve as a direction for the extension or improvement of the techniques discussed here:

- **The direction of the vessel:** In this dissertation, experiments were performed with vessels taken laterally that followed a straight line. A possible extension for the technique is to see the behavior when the vessel does not follow a straight line.
- **Identify the corners of the region of interest:** In this dissertation, a corner of the region of interest is manually selected. A possible extension for the technique is to automate this step.
- **Improve image quality:** We are exploring the use of image super-resolution to improve the estimation of the crests/troughs present in the region of interest.
- **Real-time technique:** The time it takes to execute the proposed method is not made to work in real-time. Processing only one image, once the user has specified the initial corner of the Region of Interest (ROI) and vessel direction, takes approximately twelve seconds. The part of the code that takes the longest to process is dedicated to compute the edge image. A possible improvement and extension of the technique is to detect edges only in the portion where the ROI is located.

# References

- [Ao et al. 2018]AO, D. et al. Moving ship velocity estimation using TanDEM-X Data based on subaperture decomposition. *IEEE Geoscience and Remote Sensing Letters*, IEEE, v. 15, n. 10, p. 1560–1564, 2018. ISSN 15580571.
- [Arbelaez et al. 2014]ARBELAEZ, P. et al. Multiscale combinatorial grouping. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 328–335. ISSN 1063-6919.
- [Arlinghaus 1994]ARLINGHAUS, S. L. *Practical handbook of curve fitting*. [S.l.]: CRC Press, 1994.
- [Artin 1957]ARTIN, E. *Geometric Algebra*. [S.l.]: Interscience Publishers Inc., 1957.
- [Bajpai]BAJPAI, P. *Autonomous Shipping: Trends And Innovators In A Growing Industry*. Disponível em: <<https://www.nasdaq.com/articles/autonomous-shipping%3A-trends-and-innovators-in-a-growing-industry-2020-02-18>>. Acesso em: 2020-02-18.
- [Baker 2013]BAKER, K. Singular value decomposition tutorial. v. 2005, 01 2013.
- [Bloisi et al. 2012]BLOISI, D. et al. Camera based target recognition for maritime awareness. In: *Proc. Int. Conf. Inf. Fusion*. [S.l.: s.n.], 2012. p. 1982–1987.
- [Bloisi et al. 2017]BLOISI, D. D. et al. Enhancing automatic maritime surveillance systems with visual information. *IEEE Trans. Intell. Transp. Syst.*, v. 18, p. 824–833, 2017.
- [Broggi et al. 2009]BROGGI, A. et al. Boat speed monitoring using artificial vision. In: FOGGIA, P.; SANSONE, C.; VENTO, M. (Ed.). *Image Analysis and Processing – ICIAP 2009*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 327–336. ISBN 978-3-642-04146-4.
- [Canny 1986]CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, Nov 1986. ISSN 1939-3539.
- [Carona e Marques 2014]CARONA, D.; MARQUES, P. Vessel Detection and velocity estimation using SAR amplitude images. n. October, 2014.
- [Cheng et al. 2016]CHENG, M. M. et al. HFS: Hierarchical feature selection for efficient image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 9907 LNCS, p. 867–882, 2016. ISSN 16113349.
- [Conradsen et al. 2003]CONRADSEN, K. et al. A test statistic in the complex wishart distribution and its application to change detection in polarimetric sar data. *IEEE Transactions on Geoscience and Remote Sensing*, v. 41, n. 1, p. 4–19, 2003.
- [Criminisi e Zisserman 2000]CRIMINISI, A.; ZISSERMAN, A. Shape from texture: homogeneity revisited. In: *Proc. British Machine Vision Conference (BMVC)*. [s.n.], 2000. p. 82–91. Disponível em: <<https://www.microsoft.com/en-us/research/publication/shape-from-texture-homogeneity-revisited/>>.



- [Fernandes, Oliveira e Silva 2008]FERNANDES, L. A.; OLIVEIRA, M. M.; SILVA, R. da. Uncertainty propagation: Avoiding the expensive sampling process for real-time image-based measurements. *Computational Statistics & Data Analysis*, v. 52, n. 7, p. 3852 – 3876, 2008. ISSN 0167-9473. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167947307004550>>.
- [Ferrari et al. 2008]FERRARI, V. et al. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 1, p. 36–51, Jan 2008. ISSN 1939-3539.
- [Fischler e Bolles 1981]FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/358669.358692>>.
- [Forsyth 2013]FORSYTH, P. *Computer Vision: A modern approach*. [S.l.: s.n.], 2013. 229–252 p. ISSN 1098-6596. ISBN 9788578110796.
- [Furuno Electric Co.]Furuno Electric Co. *FAR-21x7 series Brochure*. 1–8 p. Disponível em: <<https://www.furuno.com/files/Brochure/236/upload/far-21x7.pdf>>. Acesso em: 2020-03-11.
- [Gonzalez 2002]GONZALEZ, R. C. *Digital Image Processing*. 2nd. ed. [s.n.], 2002. 1–793 p. Disponível em: <[www.prenhall.com/gonzalezwoods](http://www.prenhall.com/gonzalezwoods)>.
- [Hand]HAND, M. *Autonomous shipping - are we being blinded by technology?* Disponível em: <<https://www.seatrade-maritime.com/asia/autonomous-shipping-are-we-being-blinded-technology>>.
- [Hartley e Zisserman 2004]HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. [S.l.]: Cambridge University Press, 2004.
- [Huillca e Fernandes 2019]HUILICA, J. L.; FERNANDES, L. A. F. Computing vessel velocity from single perspective projection images. In: *2019 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2019. p. 325–329.
- [Jin e Han 2010]JIN, X.; HAN, J. K-means clustering. In: \_\_\_\_\_. *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 563–564. ISBN 978-0-387-30164-8.
- [Kirscht 1996]KIRSCHT, M. Detection and velocity estimation of moving objects in a sequence of single-look sar images. In: *IGARSS '96. 1996 International Geoscience and Remote Sensing Symposium*. [S.l.: s.n.], 1996. v. 1, p. 333–335 vol.1. ISSN null.
- [Kiusalaas 2013]KIUSALAAS, J. *Numerical Methods in Engineering with Python 3*. [S.l.]: Cambridge University Press, 2013.
- [Kriebel e N.S. 2001]KRIEBEL; N.S., S. An empirical model for ship generated waves. 2001.
- [Li et al. 2019]LI, H. et al. Ship detection and tracking method for satellite video based on multiscale saliency and surrounding contrast analysis. *Journal of Applied Remote Sensing*, v. 13, n. 02, p. 1, 2019. ISSN 1931-3195.
- [Liu et al. 2019]LIU, Y. et al. Richer Convolutional Features for Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 41, n. 8, p. 1939–1946, 2019. ISSN 19393539.

- [Liu e Deng 2018]LIU, Y.; DENG, R. Ship wakes in optical images. *J. Atmospheric Ocean. Technol.*, v. 35, p. 1633–1648, 2018.
- [Marinha do Brasil]Marinha do Brasil. *Tide schedules and tables for Niteroi*. Disponível em: <<https://www.marinha.mil.br>>. Acesso em: 2020-02-16.
- [Maver Ksenija 2004]MAVER KSENIJA, R. P. Kelvin Ship Waves. p. 16, 2004.
- [Mazeika e Draudviliene 2010]MAZEIKA, L.; DRAUDVILIENE, L. Analysis of the zero-crossing technique in relation to measurements of phase velocities of the so mode of the lamb waves. *Ultrargarsas (Ultrasound)*, v. 65, 01 2010.
- [Moreira et al. 2014]MOREIRA, R. S. et al. A survey on video detection and tracking of maritime vessels. *Int. J. Res. Rev. Appl. Sci.*, v. 20, p. 37–50, 2014.
- [Nautide]Nautide. *Tide and Solunar Table - Niteroi*. Disponível em: <<https://tabuademaes.com/br/rio-de-janeiro/niteroi>>. Acesso em: 2020-03-16.
- [Nikon Corporation 2010]Nikon Corporation. *User's Manual Nikon D3300*. 2010. 1–38 p. Disponível em: <<https://download.nikonimglib.com>>. Acesso em: 2020-03-11.
- [Omar Systems Ltd. 2020]Omar Systems Ltd. *Abaout Automatic Radar System - What is AIS?* 2020. Disponível em: <<https://comarsystems.com/what-is-ais/>>. Acesso em: 2020-05-20.
- [Otsu 1979]Otsu, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979.
- [Panico, Graziano e Renga 2017]PANICO, A.; GRAZIANO, M. D.; RENGA, A. SAR-Based vessel velocity estimation from partially imaged kelvin pattern. *IEEE Geoscience and Remote Sensing Letters*, v. 14, n. 11, p. 2067–2071, 2017. ISSN 1545598X.
- [Pelich et al. 2016]PELICH, R. et al. Vessel refocusing and velocity estimation on SAR imagery using the fractional fourier transform. *IEEE Transactions on Geoscience and Remote Sensing*, v. 54, n. 3, p. 1670–1684, 2016. ISSN 01962892.
- [Pichel et al. 2004]PICHEL, W. G. et al. Ship and wake detection. *SAR Marine Users Manual*, p. 277–303, 2004.
- [Reggiannini e Bedini 2018]REGGIANNINI, M.; BEDINI, L. Synthetic Aperture Radar Processing for Vessel Kinematics Estimation. *Proceedings*, v. 2, n. 2, p. 91, 2018. ISSN 2504-3900.
- [Savitzky e Golay 1964]SAVITZKY, A.; GOLAY, M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, v. 36, p. 1627–1639, 1964.
- [Schalkoff 1989]SCHALKOFF, R. *Digital image processing and computer vision*. John Wiley & Sons Australia, Limited, 1989. ISBN 9780471857181. Disponível em: <<https://books.google.com.br/books?id=-vtRAAAAMAAJ>>.
- [SHAPIRO e WILK 1965]SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, v. 52, n. 3-4, p. 591–611, 1965. ISSN 0006-3444.
- [Shapiro e Wilk 2007]SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality ( complete samp 1 es ) t. In: . [S.l.: s.n.], 2007.

- [Tan, Piepmeier e Kriebel 2012] TAN, S.; PIEPMEIER, J. A.; KRIEBEL, D. L. A computer vision system for monitoring vessel motion in conjunction with vessel wake measurements. In: *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. [S.l.: s.n.], 2012. p. 1830–1834.
- [Tetley e Calcutt 2001] TETLEY, L.; CALCUTT, D. *Electronic Navigation Systems*. 3rd. ed. [S.l.: Butterworth-Heinemann, 2001.
- [Thomson 1887] THOMSON, W. On ship waves. *Proc. Inst. Mech. Eng.*, v. 38, p. 641–649, 1887.
- [Tides Chart] Tides Chart. *Tide schedules and tables for Niteroi*. Disponível em: <<https://pt.tideschart.com>>. Acesso em: 2020-03-16.
- [Tunaley 2003] Tunaley, J. K. E. The estimation of ship velocity from sar imagery. In: *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*. [S.l.: s.n.], 2003. v. 1, p. 191–193.
- [Ullman e Basri 1991] ULLMAN, S.; BASRI, R. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 13, n. 10, p. 992–1006, Oct 1991. ISSN 1939-3539.
- [Ursell 1959] URSELL, F. On Kelvin’s ship-wave pattern. n. October, p. 418–431, 1959.
- [Vuolo 1996] VUOLO, J. H. *Fundamentos da teoria dos erros*. [S.l.]: E. Blucher, 1996.
- [Wang et al. 2019] WANG, C. et al. A novel vessel velocity estimation method using Dual-Platform TerraSAR-X and TanDEM-X Full Polarimetric SAR Data in Pursuit Monostatic Mode. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 57, n. 8, p. 6130–6144, 2019. ISSN 15580644.
- [Wawrzyniak N. 2019] WAWRZYNIAK N., H. T. . P. A. Vessel detection and tracking method based on video surveillance. *Sensors (Basel, Switzerland)*, v. 19, n. 23, 2019.
- [Wikipedia 2019] WIKIPEDIA. *Lidar*. 2019. Disponível em: <<https://en.wikipedia.org/wiki/Lidar>>. Acesso em: 2020-03-16.
- [Wikipedia 2020] WIKIPEDIA. *Azimuthal Equidistant Projection*. 2020. Disponível em: <[https://en.wikipedia.org/wiki/Azimuthal\\_equidistant\\_projection](https://en.wikipedia.org/wiki/Azimuthal_equidistant_projection)>. Acesso em: 2020-04-14.
- [Wikipedia 2020] WIKIPEDIA. *Radar*. 2020. Disponível em: <<https://en.wikipedia.org/wiki/Radar>>. Acesso em: 2020-02-13.
- [Wikipedia 2020] WIKIPEDIA. *Satellite Navigation Device*. 2020. Disponível em: <[https://en.wikipedia.org/wiki/Satellite\\_navigation\\_device](https://en.wikipedia.org/wiki/Satellite_navigation_device)>. Acesso em: 2020-04-13.
- [Wilcox 2017] WILCOX, R. The regression smoother lowess: A confidence band that allows heteroscedasticity and has some specified simultaneous probability coverage. *Journal of Modern Applied Statistical Methods*, v. 16, p. 29–38, 12 2017.
- [Workman, Zhai e Jacobs 2016] WORKMAN, S.; ZHAI, M.; JACOBS, N. Horizon lines in the wild. *British Machine Vision Conference 2016, BMVC 2016*, v. 2016-Sept, p. 20.1–20.12, 2016.
- [Xie e Tu 2017] XIE, S.; TU, Z. Holistically-Nested Edge Detection. *International Journal of Computer Vision*, v. 125, n. 1-3, p. 3–18, 2017. ISSN 15731405.

- [Xu et al. 2018]XU, D. et al. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. 01 2018.
- [Yang et al. 2019]YANG, S. et al. Ship speed and maneuverability measurement & analysis system based on GPS device. *Proceedings - 2019 4th International Conference on Electromechanical Control Technology and Transportation, ICECTT 2019*, IEEE, p. 18–22, 2019.
- [Zhang et al. 2016]ZHANG, X. et al. Accelerating Very Deep Convolutional Networks for Classification and Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 10, p. 1943–1955, 2016. ISSN 01628828.

## APPENDIX A – Standard Deviation for Input Variables

This appendix describes how the standard deviations of the random input variables used in Chapter 9 were calculated. The input variables used are the  $y$ -coordinates ( $v_l, v_r$ ) of each point that forms the vanishing line, and the components of the camera height.

### A.1 Finding the Standard Deviation for the Vanishing Line

The horizon vanishing line computed as the cross product of the vectors encoding two points in homogeneous coordinates, and these vectors can be defined by:

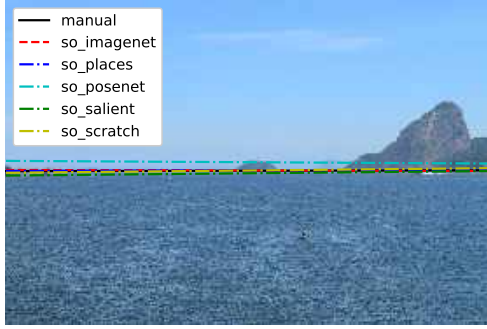
$$\begin{aligned}\mathbf{p}_{v_l} &= (0, v_l, 1)^T \\ \mathbf{p}_{v_r} &= (W, v_r, 1)^T,\end{aligned}\tag{A.1}$$

where  $W$  is the width of the input image. Only the  $y$ -axes of the points are taken as input variables because the  $x$ -axis is constant. These variables,  $v_l$  and  $v_r$ , are found using the Horizon Line of the Wild (HLW) algorithm [Workman, Zhai e Jacobs 2016], discussed in Section 4.2. HLW presents five models to find the vanishing line in an image. An example of the results of the possible models to be used is shown in Figure A.1.

Figure A.1 shows four of the  $n = 30$  images used to choose the model of the HLW algorithm which best approximates the actual vanishing line. To choose the best model, the points that define the vanishing line  $l_t$  were manually marked, and these points are taken as true points. Table A.1 shows the cosine distances between the actual vanishing line and the vanishing line of each model of the images presented in Figure A.1, compute as:

$$d_{\cosine} = 1 - \frac{l_i l_t}{\|l_i\|_2 \|l_t\|_2}\tag{A.2}$$

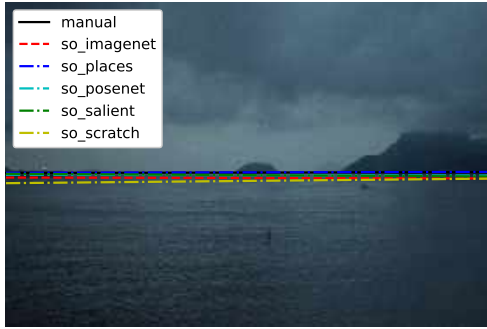
The cosine distance (Equation A.2) was used to measure how close each model is to the true line. The  $n = 30$  images were taken under different weather conditions and day times. The cosine



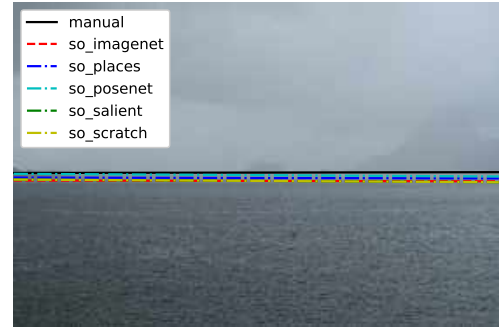
(a) Photograph taken on a sunny day with good weather.



(b) Photograph taken in the afternoon, a few hours after the sun goes down, the weather is also good.



(c) Photograph taken at dusk, with little natural lighting, on a cloudy day.



(d) Photograph taken at noon, with an unfavorable climate, with rain and fog.

Figure A.1: Four of the 30 images were chosen to visualize the results of the models presented by the HLW algorithm. The black line was found manually. It serves as a reference to identify the best method for detecting the vanishing line.

Table A.1: Result of the cosine distance between each method for detecting the vanishing line and the reference line on the images presented in Figure A.1. The smaller distances are highlighted in red.

Method	Distances			
	(a)	(b)	(c)	(d)
so_imagenet	$6.02 \times 10^{12}$	<b><math>6.02 \times 10^{12}</math></b>	$7.82 \times 10^{11}$	$2.01 \times 10^{10}$
so_places	<b><math>1.91 \times 10^{13}</math></b>	$4.20 \times 10^{11}$	<b><math>1.91 \times 10^{13}</math></b>	$7.82 \times 10^{11}$
so_posenet	$4.68 \times 10^{10}$	$4.69 \times 10^{10}$	$1.25 \times 10^{11}$	<b><math>8.29 \times 10^{12}</math></b>
so_salient	$7.86 \times 10^{11}$	$9.90 \times 10^{11}$	$8.29 \times 10^{12}$	$1.61 \times 10^{10}$
so_scratch	$1.66 \times 10^{11}$	$3.66 \times 10^{10}$	$3.66 \times 10^{10}$	$1.61 \times 10^{10}$

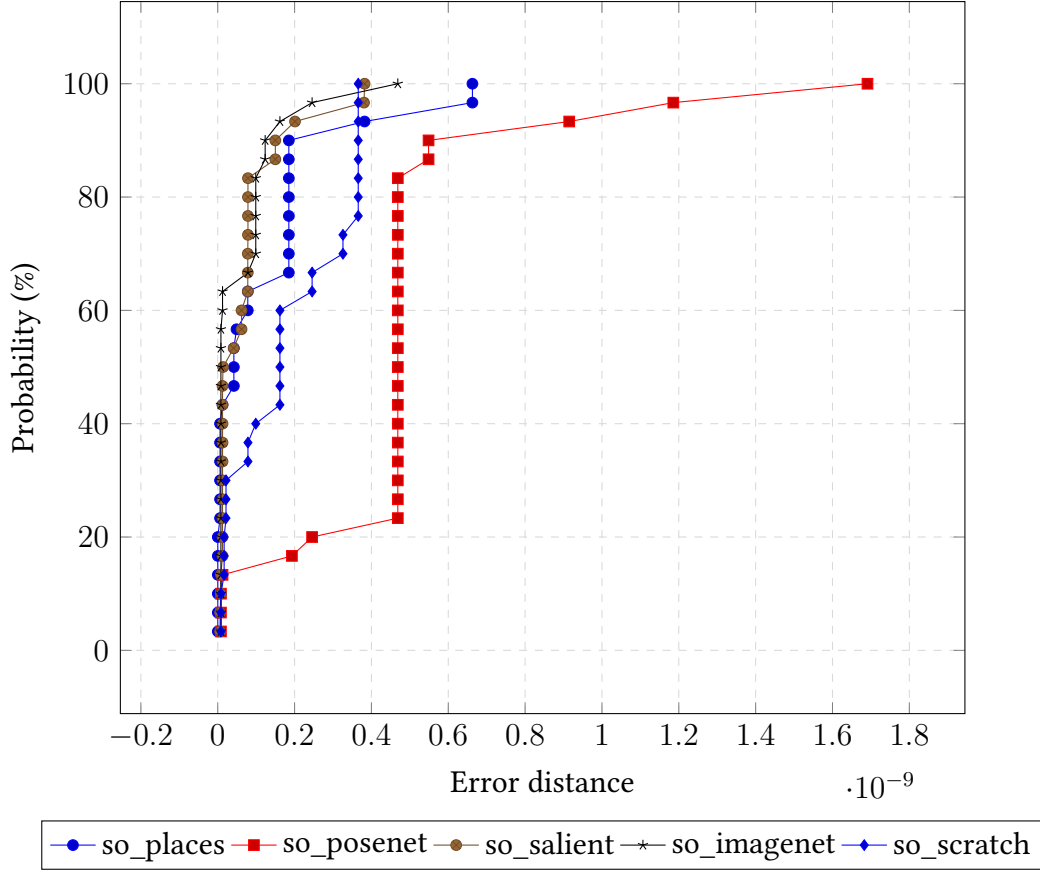


Figure A.2: Cumulative distribution of errors for the HWL algorithm models. 30 images were used. *so\_imagenet* presents good results, obtaining a very low error. This model is used to find future vanishing lines.

distance between the true line  $l_t$  and each line  $l_i$  was calculated. Figure A.2 shows the cumulative error distribution for each model. Results show that the model with the lowest error is *so\_imagenet*. Therefore, we have chosen this method to calculate the points that defines the vanishing line.

A standard deviation for each  $y$ -axis is calculated using Equation A.3, where  $\bar{y}$  is the average of all  $y_i$ , and  $y_i$  is the error on each point of the  $y$ -coordinate. The error is calculated by measuring the Euclidean distance between the point calculated by the HLW algorithm and the manually marked point.  $\sigma$  represents the standard deviation of the point on the  $y$ -axis. The standard deviations calculated are 22.72 and 23.62 for  $v_l$  and  $v_r$ , respectively. Where  $\sigma$  is used to generate random points, following a Gaussian distribution, which will in turn be used for producing samples in Section 9.3:

$$\sigma = \sqrt{\frac{\sum_i (y_i - \bar{y})^2}{n - 1}} \quad (\text{A.3})$$

where  $n = 30$  is the number of images used.

## A.2 Finding the Standard Deviation of the Camera Height

The camera height is computed from four input variables: the ground height, the floor height, the tripod height, and the sea tide height, i.e.,  $h_g, h_f, h_t, h_s$ , respectively. A tape measure was used to calculate the floor height, and the tripod height. We use Google Maps to measure the ground height. The deviation used for the components of the camera height was 0.03  $m$ , except for the tide height. To estimate the deviation of the tide height, three sources were used Nautide [Nautide], Tides Chart [Tides Chart], and Marinha do Brasil [Marinha do Brasil], shown in the Table A.3.

Table A.3 shows the date, time, and tide height for each source. The last column of the table shows the average to each tide height ( $\bar{\tau}_k$ ), defined by:

$$\bar{\tau}_k = \frac{1}{m} \sum_{i=1}^m \tau_i, \quad (\text{A.4})$$

where  $m = 3$  is the number of sources used, and  $\tau_i$  is the value of each row in Tides column.

We calculate the standard deviation using Equation A.3 , assuming  $n = 27$  tidal heights. The result of the standard deviation obtained is 0.049  $m$ .



Table A.3: Tides Table of Rio de Janeiro - Niteroi. The values in this table are used to estimate the average error of the tide between three sources: Nautide [Nautide], Tide Chart [Tides Chart], and *Marinha do Brasil* [Marinha do Brasil], respectively.

Date	Time			Tides (cm)			Average
	Source 1	Source 2	Source 3	Source 1	Source 2	<b>Source 3</b>	
Nov 8, 2019	07:01	06:35	06:28	10	15	20	15.00
	12:02	12:07	12:34	110	100	110	106.67
	19:21	18:55	19:04	30	26	30	28.67
Nov 9, 2019t	00:01	00:06	00:38	110	106	110	108.67
	07:39	07:13	07:08	10	10	20	13.33
	12:31	12:33	13:02	110	106	120	112.00
	19:59	19:33	19:38	20	19	30	23.00
Nov 10, 2019	00:38	00:43	01:09	120	108	120	116.00
	08:23	07:57	07:47	10	8	20	12.67
	13:06	13:11	13:30	120	112	120	117.33
	20:40	21:51	20:08	10	10	20	13.33
Nov 11, 2019	01:22	01:27	01:47	120	117	130	122.33
	09:05	08:39	08:23	10	8	20	12.67
	13:47	12:52	14:00	120	112	120	117.33
	21:17	20:51	20:39	10	10	20	13.33
Nov 12, 2019	02:10	02:15	02:17	130	121	130	127.00
	09:40	09:14	09:00	10	10	20	13.33
	14:28	14:33	14:30	120	113	130	121.00
	21:46	21:20	21:09	10	9	10	9.67
Nov 13, 2019	02:53	02:58	02:54	130	123	130	127.67
	10:11	09:45	09:39	20	17	20	19.00
	15:04	15:09	15:00	120	113	120	117.67
	22:12	21:46	21:43	10	11	10	10.33
Nov 14, 2019	03:31	03:36	03:28	130	124	130	128.00
	10:40	10:14	10:21	30	28	30	29.33
	15:36	15:41	15:30	120	110	120	116.67
	22:33	22:07	22:23	20	16	20	18.67

## APPENDIX B – Vessel Images not Considered in the Experiments

In this appendix, images not considered for the experiments and analysis of the results, Chapter 9, are shown. The images were not considered due to being in adverse weather conditions, little natural light, and not being able to estimate the vessel speed with the proposed method.

A total of 40 images were captured for the experiments, where 23 images were used for the main experiments (see Figures 9.2, 9.3, 9.4, and 9.5, Chapter 9). In the remaining images, 17 images (see Figures B.1, B.2, and B.3), the wake left by the vessel was very weak or not visible, and they were not considered. Table B.1 shows information about the 17 vessel images, where the first column presents the image name. The second column displays the vessel model. The third column shows the vessel name. The fourth, fifth, and sixth columns present the vessel speed give by the radar, the time was taken the image, and the weather condition, respectively. The last column shows the tide height. Two types of vessels are shown, passenger ship (model *MC25*) and merchant ships (*i24*, *i25*, *i27*, and *i32*) are shown in Table B.1 as ship models equal to others because it was not possible to obtain the exact ship model.

We used the proposed method to estimate the speed of the 17 images. Section B.1 presents an analysis of the relative error of the calculated speeds.

### B.1 Analysis of Results

From the 17 images shown (Figures B.1, B.2, B.3, and ??), we grouped them into two groups according to the trace left by the vessel:

- First group:** images where the vessel's trace is not completely visible. These images are *i27*, *i29*, and *i37*. They can also be seen in Figure B.4 (b), (d) and (f), respectively;
- Second group:** images where the vessel's trace is very weak, and it is not possible to distinguish the crests/troughs. These images are *i24* – *i26*, *i28*, *i30* – *i36*, and *i38* – *i40*. They can

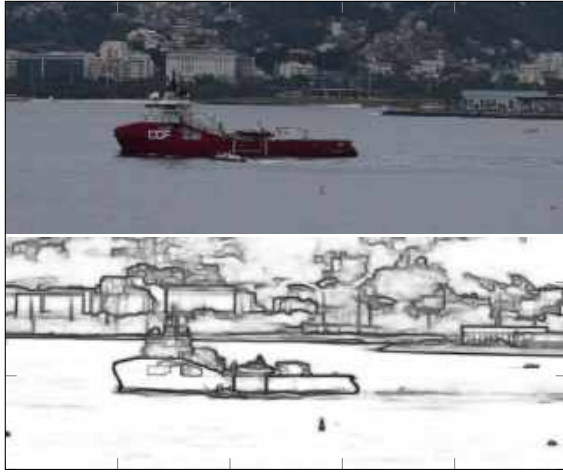
Table B.1: Ground-truth information of the vessel in each captured image. This images were not considered in the main experiments.

Image	Vessel Model	Vessel Name	Speed ( <i>knots</i> )	Time (hh:mm)	Weather	Tide Height (meters)
i24	<i>Other</i>	Skandi Peregrino	10.10	10:05	Cloudy	0.30
i25	<i>Other</i>	Escander Amazonas	9.10	10:26	Cloudy	0.50
i26	<i>MC25</i>	Missing	18.90	10:38	Cloudy	0.50
i27	<i>Other</i>	Bahia San Blas	8.20	10:46	Cloudy	0.70
i28	<i>MC25</i>	Zeus	19.60	11:41	Cloudy	0.70
i29	<i>MC25</i>	Neptuno	14.30	11:43	Cloudy	0.70
i30	<i>MC25</i>	Neptuno	20.20	12:09	Cloudy	0.90
i31	<i>MC25</i>	Apolo	18.90	16:12	Scattered storms	1.10
i32	<i>Other</i>	Far Scotsman	8.00	16:25	Scattered storms	1.10
i33	<i>MC25</i>	Apolo	21.10	16:40	Partly Cloudy	0.70
i34	<i>MC25</i>	Missing	20.30	17:01	Partly cloudy	0.70
i35	<i>MC25</i>	Missing	20.30	17:02	Partly cloudy	0.70
i36	<i>MC25</i>	Zeus	17.90	17:21	Partly cloudy	0.50
i37	<i>MC25</i>	Zeus	17.70	17:22	Partly Cloudy	0.50
i38	<i>MC25</i>	Missing	15.70	17:29	Cloudy	0.50
i39	<i>MC25</i>	Apolo	22.80	17:38	Cloudy	0.50
i40	<i>MC25</i>	Apolo	22.20	17:39	Cloudy	0.50

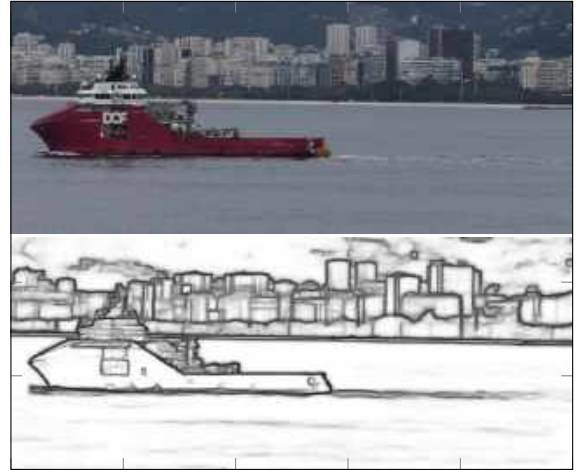
also be seen in Figure B.4 (a), (c), (e), (g), (h), (i) and (j).

In the first and second groups, the vessels' speed cannot be estimated because the wake and crests/troughs were not identified. The common problems found in the two groups are:

- Weather conditions:** images taken on a cloudy day, or a day with a storm or rain, also, the presence of strong wind affects the shape of the wave; and
- Natural light:** depending on the time the vessel image was taken (day, afternoon, night).



(i24)  $U \approx 10.10 \text{ knots}$



(i25)  $U \approx 9.10 \text{ knots}$



(i26)  $U \approx 18.90 \text{ knots}$



(i27)  $U \approx 8.20 \text{ knots}$



(i28)  $U \approx 19.60 \text{ knots}$



(i29)  $U \approx 14.30 \text{ knots}$

Figure B.1: Cropping the images used in the experiments. Images  $i24 - i29$  shown in Table B.1.



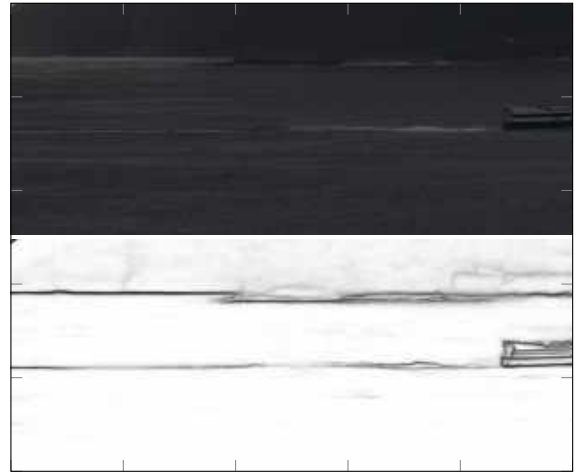
(i30)  $U \approx 20.20 \text{ knots}$



(i31)  $U \approx 18.90 \text{ knots}$



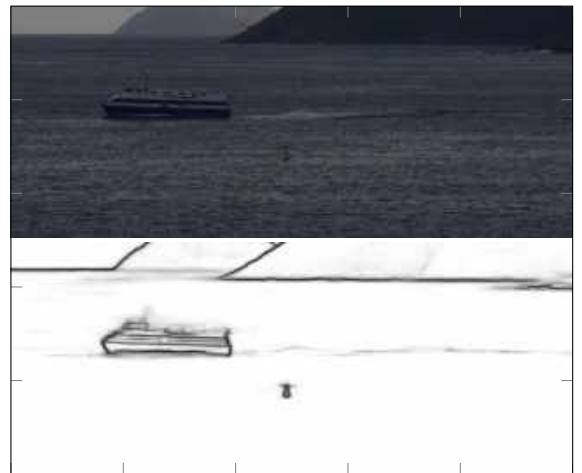
(i32)  $U \approx 8.00 \text{ knots}$



(i33)  $U \approx 21.10 \text{ knots}$

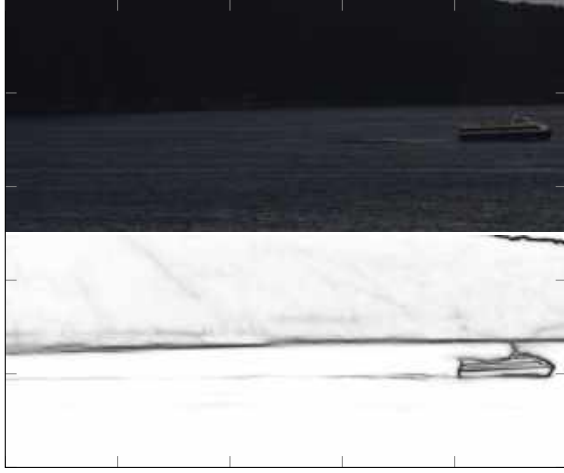


(i34)  $U \approx 20.30 \text{ knots}$



(i35)  $U \approx 20.30 \text{ knots}$

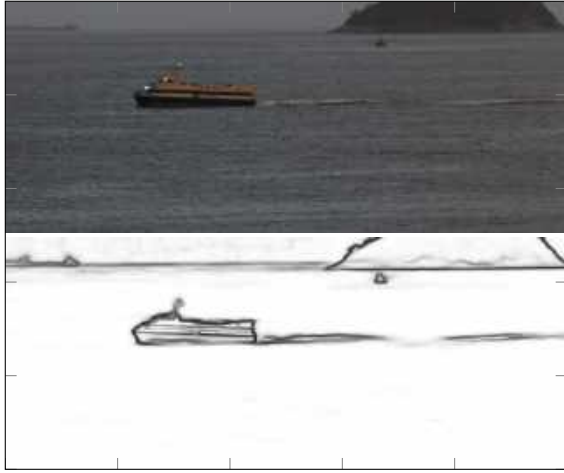
Figure B.2: Cropping the images used in the experiments. Images  $i30 - i35$  shown in Table B.1.



(i36)  $U \approx 17.90 \text{ knots}$



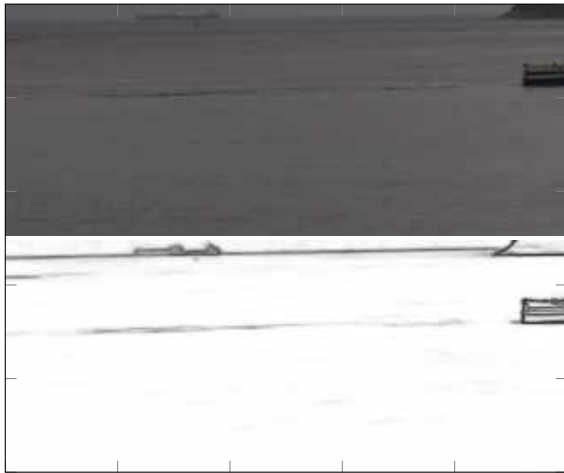
(i37)  $U \approx 17.70 \text{ knots}$



(i38)  $U \approx 15.70 \text{ knots}$



(i39)  $U \approx 22.80 \text{ knots}$



(i40)  $U \approx 22.20 \text{ knots}$

Figure B.3: Cropping the images used in the experiments. Images  $i36 - i40$  shown in Table B.1.

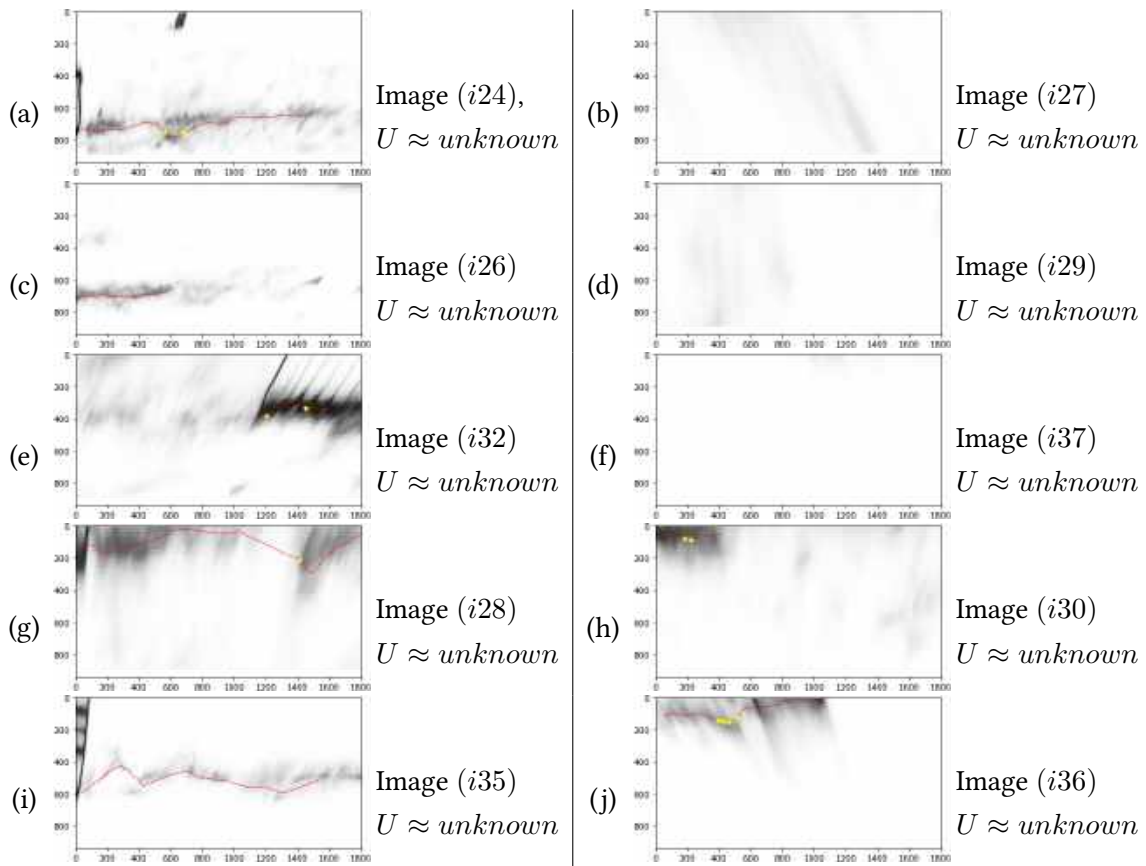


Figure B.4: Each image shows the region of interest rectified along with the image name and the speed calculated by the proposed method, where the red dotted line is the chosen wave arm to find the wavelength. The images where it was not possible to identify the traces of the vessel and estimate the speed were shown as unknown speed.