BRUNO GIORDANO LEITE

MARITIME NAVIGATIONAL ASSISTANCE BY VISUAL AUGMENTATION

Revised Version

São Paulo
2022

BRUNO GIORDANO LEITE

MARITIME NAVIGATIONAL ASSISTANCE BY VISUAL AUGMENTATION

Revised Version

Dissertation presented to the **Escola Politécnica da Universidade de São Paulo** to obtain the degree of **Master of Science**

São Paulo
2022

BRUNO GIORDANO LEITE

MARITIME NAVIGATIONAL ASSISTANCE BY VISUAL AUGMENTATION

Revised Version

Dissertation presented to the **Escola Politécnica da Universidade de São Paulo** to obtain the degree of **Master of Science**

Concentration Area:

Control and Automation

Advisor:

Prof. Dr. Eduardo Aoun Tannuri

Co-Advisor:

Prof. Dr. Newton Maruyama

São Paulo
2022

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 05 de abril de 2022

Assinatura do autor: Bruno Giordano Leite

Assinatura do orientador:

Catalogação-na-publicação

Leite, Bruno Giordano
Maritime navigational assistance by visual augmentation / B. G. Leite -- versão corr. -- São Paulo, 2022.
193 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Maritime navigational assistance 2.Augmented reality systems I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

AGRADECIMENTOS (ACKNOWLEDGMENTS)

Aos professores Dr. Eduardo Aoun Tannuri e Dr. Newton Maruyama pela oportunidade, confiança e orientação.

Aos professores da Escola Politécnica pelos anos de aprendizado.

À toda equipe do *Tanque de Provas Numérico da Universidade de São Paulo* (**TPN-USP**) pela infraestrutura disponibilizada.

Às instituições públicas de fomento à pesquisa, em especial à *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (**CAPES**), ao *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (**CNPq**) e à *Fundação de Apoio à Universidade de São Paulo* (**FUSP**) pelo suporte financeiro.

À *Petrobras* e à *Agência Nacional de Petróleo* (**ANP**) pelo suporte ao projeto de pesquisa.

Ao MSc. Helio Takahiro Sinohara pela ajuda prática e pela disponibilização de dados experimentais.

À família e aos amigos que tornaram suportáveis momentos dificeis.

ABSTRACT

The incorporation of arising technology in maritime processes and equipment have been historically very important for the development of commercial navigation. Several types of equipment have been developed over the years to assist ship operators with their tasks. Generally, the incorporation of new technologies on maritime equipment are associated with an increase on the amount of information and alerts displayed to the corresponding users. Note that information from new equipment needs to be efficiently presented to avoid the introduction of unnecessary cognitive load. A potentially optimal option for the presentation of combined navigational information is the deployment of *augmented-reality systems* (AR). In such a system, virtual elements are rendered on top of the navigation scene to assist operators in their perception. In this context, the present work is a preliminary investigation towards the development of an AR system for navigational assistance. Particularly, foundations regarding AR systems are presented and a simple conceptual design for navigational assistance is proposed. Additionally, important components from the proposed conceptual design are exemplified with numerical implementations. Proposed methodologies are validated with experiments from videos of a ship navigating through a channel delimited by nautical buoys. Enhanced scenes combining pertinent virtual elements are shown on top of the navigation scene exemplifying potential assistance applications for this scenario. As there are still many challenges for the development of operational maritime tools with AR methods, the author also expects to contribute with the popularization of AR systems in maritime environments.

Keywords – Maritime Navigational Assistance, Augmented Reality Systems, Camera Models

RESUMO

A incorporação de tecnologias emergentes em processos e equipamentos marítimos foi historicamente muito importante para o desenvolvimento da navegação comercial. Diversos tipos de equipamentos foram desenvolvidos durante os anos para ajudar os operadores com suas tarefas. De uma maneira geral, a incorporação de novas tecnologias em equipamentos marítimos está associada com um aumento na quantidade de informação e alertas apresentados aos usuários. Note que informações associadas a equipamentos novos precisam ser apresentadas eficientemente para evitar a introdução de cargas cognitivas desnecessárias. Uma opção potencialmente ótima para a apresentação combinada de informação de navegação é a utilização de sistemas de *realidade aumentada*. Em um sistema desse tipo, elementos virtuais são desenhados em cima da cena de navegação para auxiliar operadores na percepção da cena de navegação. Nesse contexto, o trabalho atual é uma investigação preliminar com o intuito de desenvolver um sistema de realidade aumentada para assistência à navegação. Particularmente, fundamentos relacionados a sistemas de *realidade aumentada* são apresentados e um simples design conceitual para assistência à navegação é proposto. Adicionalmente, componentes importantes do design conceitual proposto são exemplificados com implementações numéricas. As metodologias propostas são validadas com experimentos de videos onde um navio navega em um canal marítimo delimitado por bóias náuticas. Possíveis cenas virtuais que combinam elementos virtuais pertinentes são apresentados para exemplificar possíveis aplicações de assistência a esse cenario de navegação. Como ainda existem diversos desafios para o desenvolvimento de uma ferramenta marítima operacional com métodos de realidade aumentada, o autor também espera contribuir com a popularização de sistemas de *realidade aumentada* em ambientes marítimos.

.

Palavras-Chave – Assistente de Navegação Marítima, Sistemas de Realidade Aumentada, Modelos de Câmera

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Tanque de Provas Numerico – **(TPN)**

Universidade de Sao Paulo – **(USP)**

Integrated Bridge Systems – **(IBS)**

Computer Processing Unit – **(CPU)**

Graphical Processing Unit – **(GPU)**

Augmented-Reality – **(AR)**

Ship Maneuvering Simulator – **(SMS)**

Information Technology – **(IT)**

Information Systems – **(IS)**

Electronic Data Interchange – **(EDI)**

Port Community Systems – **(PCS)**

Terminal Operating Systems – **(TOS)**

Value-Added Logistics – **(VAL)**

Vessel Traffic Services – **(VTS)**

Automatic Identification Systems – **(AIS)**

Truck Appointment System – **(TAS)**

Radio Frequency Identification – **(RFID)**

Maritime Navigation Safety and Efficiency System – **(MNSES)**

Global Maritime Distress and Safety System – **(GMDSS)**

Maritime Autonomous Surface Ships – **(MASS)**

Technology Readiness Levels – **(TRL)**

International Maritime Organization – **(IMO)**

United Nation Organization – **(UN)**

International Convention on Safety Of Life at Sea – **(SOLAS)**

International Convention for Prevention of Pollution from Ships – **(MARPOL)**

International Convention on International Regulations for Preventing Collision at Sea – **(COLREG)**

International Convention on Standards of Training, Certification and Watchkeeping for Seafarers – **(STCW)**

International Convention on Maritime Search and Rescue – **(SAR)**

Maritime Safety Information – **(MSI)**

Head-mounted device – **(HMD)**

Mixed Reality – **(MR)**

Augmented Virtuality – **(AV)**

Institute of Electrical and Electronic Engineers – **(IEEE)**

Association for Computing Machinery – **(ACM)**

International Workshop on Augmented Reality – **(IWAR)**

International Symposium on Mixed Reality – **(ISMR)**

International Symposium on Augmented Reality – **(ISAR)**

Three-dimensional – **(3D)**

Maritime Augmented Reality Navigational Assistance – **(MARNAA)**

Automatic Identification System – **(AIS)**

Electronic Chart Display and Information Systems – **(ECDIS)**

Pan/Tilt/Zoom – **(PTZ)**

Attitude and Heading Reference System – **(AHRS)**

Monitor Augmented Reality – **(MAR)**

National Maritime Electronics Association – **(NMEA)**

Global Positioning System – **(GPS)**

Extended Kalman Filter – **(EKF)**

Power Spectral Density – **(PSD)**

Inertial Measurement Unit – **(IMU)**

Inertial Navigation System – **(INS)**

Global Navigation Satellite Systems – **(GNSS)**

World Coordinate System – **(WCS)**

Ship Coordinate System – **(SCS)**

Camera Coordinate System – **(CCS)**

World Geodetic System – **(WGS)**

Portable Pilot Unit – **(PPU)**

LIST OF SYMBOLS

$S_w \Rightarrow$ World coordinate system

$O_w \Rightarrow$ Origin of the world coordinate system

$[\vec{x}_w, \vec{y}_w, \vec{z}_w] \Rightarrow$ Axes of the world coordinate system

$S_s \Rightarrow$ Ship coordinate system

$O_s \Rightarrow$ Origin of the ship coordinate system

$[\vec{i}_s, \vec{j}_s, \vec{k}_s] \Rightarrow$ Axes of the ship coordinate system

$[\vec{i}_s^w, \vec{j}_s^w, \vec{k}_s^w] \Rightarrow$ Axes of the ship coordinate system described in the world coordinate system

$\vec{i}_s^w = [i_s^{w,x}, i_s^{w,y}, i_s^{w,z}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{i}_s^w$ with respect to the world coordinate system

$\vec{j}_s^w = [j_s^{w,x}, j_s^{w,y}, j_s^{w,z}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{j}_s^w$ with respect to the world coordinate system

$\vec{k}_s^w = [k_s^{w,x}, k_s^{w,y}, k_s^{w,z}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{k}_s^w$ with respect to the world coordinate system

$R_s^w \Rightarrow$ Rotation matrix from the ship coordinate system to the world coordinate system

$R_w^s \Rightarrow$ Rotation matrix from the world coordinate system to the ship coordinate system

$[\vec{x}_w^s, \vec{y}_w^s, \vec{z}_w^s] \Rightarrow$ Axes of the world coordinate system described in the ship coordinate system

$\vec{a} \cdot \vec{b} \Rightarrow$ Dot-product between vectors $\vec{a}$ and $\vec{b}$

$I_{n \times n} \Rightarrow$ Identity matrix with size $n \times n$

$c(.) \Rightarrow$ Trigonometric function cos(.)

$s(.) \Rightarrow$ Trigonometric function sin(.)

$R_\alpha^\beta(\phi, \theta, \psi)^{XYZ} \Rightarrow$ Rotation matrix with three independent rotations starting with the $z-$axis

$R_\alpha^\beta(\phi, \theta, \psi)^{ZYX} \Rightarrow$ Rotation matrix with three independent rotations starting with the $x-$axis

$p_{wQ}^w = [p_{wQ}^{w,x}, p_{wQ}^{w,y}, p_{wQ}^{w,z}]^T \Rightarrow$ Position of an arbitrary point Q with respect to the world coordinate system

$p_{sQ}^s = [p_{sQ}^{s,i}, p_{sQ}^{s,j}, p_{sQ}^{s,k}]^T \Rightarrow$ Position of an arbitrary point Q with respect to the ship coordinate system

$p_{ws}^w = [p_{ws}^{w,x}, p_{ws}^{w,y}, p_{ws}^{w,z}]^T \Rightarrow$ Position of the ship with respect to the world coordinate

system

$T_s^w \Rightarrow$ Transformation matrix from the ship coordinate system to the world coordinate system

$T_w^s \Rightarrow$ Transformation matrix from the world coordinate system to the ship coordinate system

$S_i \Rightarrow$ Inertial coordinate system

$O_i \Rightarrow$ Origin of the inertial coordinate system

$[\vec{x}_i, \vec{y}_i, \vec{z}_i] \Rightarrow$ Axes of the inertial coordinate system

$S_e \Rightarrow$ Earth coordinate system

$O_e \Rightarrow$ Origin of the earth coordinate system

$[\vec{x}_e, \vec{y}_e, \vec{z}_e] \Rightarrow$ Axes of the earth coordinate system

$R_i^e \Rightarrow$ Rotation matrix from the inertial coordinate system to the earth coordinate system

$R_e^i \Rightarrow$ Rotation matrix from the earth coordinate system to the inertial coordinate system

$\omega_{ie} \Rightarrow$ Angular rate of the earth with respect to the inertial coordinate system

$R_0^{WGS} \Rightarrow$ Equatorial radius from the earth accordingly to the WGS 84 geoid model

$R_0^{WGS} \Rightarrow$ Polar radius from the earth accordingly to the WGS 84 geoid model

$f^{WGS} \Rightarrow$ Flattening from the earth accordingly to the WGS 84 geoid model

$e^{WGS} \Rightarrow$ Eccentricity from the earth accordingly to the WGS 84

$p_{eQ}^e \Rightarrow$ Cartesian position of an arbitrary point Q with respect to the earth frame

$[L_q, \lambda_q, h_q] \Rightarrow$ Curvilinear position of an arbitrary point Q with respect to the earth frame

$L_q \Rightarrow$ Latitude of an arbitrary point Q with respect to the earth frame

$\lambda_q \Rightarrow$ Longitude of an arbitrary point Q with respect to the earth frame

$h_q \Rightarrow$ Height of an arbitrary point Q with respect to the earth frame

$S_n \Rightarrow$ Navigation coordinate system

$O_n \Rightarrow$ Origin of the navigation coordinate system

$[\vec{x}_n, \vec{y}_n, \vec{z}_n] \Rightarrow$ Axes of the navigation coordinate system

$R_n^e(L_q, \lambda_q, h_q) \Rightarrow$ Rotation matrix from the navigation coordinate system to the earth coordinate system at point Q as a function of the curvilinear position of Q

$R_e^n(L_q, \lambda_q, h_q) \Rightarrow$ Rotation matrix from the earth coordinate system to the navigation coordinate system at point Q as a function of the curvilinear position of Q

$g_0(L_q) \Rightarrow$ Acceleration due to gravity at point Q as a function of the latitude of Q accordingly to the WGS 84

$\vec{g}^n_Q(L_q) \Rightarrow$ Gravity vector at point Q expressed in the navigation frame as a function of the latitude of Q

$\vec{g}^e_Q(L_q, \lambda_q, h_q) \Rightarrow$ Gravity vector at point Q expressed in the earth frame as a function of the curvilinear position of Q

$[L_w, \lambda_w, h_w] \Rightarrow$ Curvilinear position of the origin O_w from the world coordinate system described in the earth frame

$p^e_{ew} \Rightarrow$ Cartesian position of the origin O_w from the world coordinate system described in the earth frame

$[\vec{x}^n_w, \vec{y}^n_w, \vec{z}^n_w] \Rightarrow$ Axes of the world coordinate system described in the navigation coordinate system

$\vec{x}^n_w = [x^{n,x}_w, x^{n,y}_w, x^{n,z}_w]^T \Rightarrow$ Three-dimensional representation for axis $\vec{x}^n_w$ with respect to the navigation coordinate system

$\vec{y}^n_w = [y^{n,x}_w, y^{n,y}_w, y^{n,z}_w]^T \Rightarrow$ Three-dimensional representation for axis $\vec{y}^n_w$ with respect to the navigation coordinate system

$\vec{z}^n_w = [z^{n,x}_w, z^{n,y}_w, z^{n,z}_w]^T \Rightarrow$ Three-dimensional representation for axis $\vec{z}^n_w$ with respect to the navigation coordinate system

$R^n_w \Rightarrow$ Rotation matrix from the world coordinate system to the navigation coordinate system

$R^e_w \Rightarrow$ Rotation matrix from the world coordinate system to the earth coordinate system

$p^e_{eQ} = [p^{e,x}_{eQ}, p^{e,y}_{eQ}, p^{e,z}_{eQ}]^T \Rightarrow$ Position of an arbitrary point Q with respect to the earth coordinate system

$p^\gamma_{\beta\alpha} \Rightarrow$ Position of an arbitrary object frame α with respect to a reference frame β expressed in a resolving frame γ

$v^\gamma_{\beta\alpha} \Rightarrow$ Velocity of an arbitrary object frame α with respect to a reference frame β expressed in a resolving frame γ

$a^\gamma_{\beta\alpha} \Rightarrow$ Acceleration of object frame α with respect to a reference frame β expressed in a resolving frame γ

$\omega^\gamma_{\beta\alpha} \Rightarrow$ Angular rate of an arbitrary object frame α with respect to a reference frame β expressed in a resolving frame γ

$\left[\omega^\gamma_{\beta\alpha}\hat{\times}\right] = \Omega^\gamma_{\beta\alpha} \Rightarrow$ Skew-symmetric representation for the angular rate of an arbitrary object frame α with respect to a reference frame β expressed in a resolving frame γ

$p^e_{es} \Rightarrow$ Position of the ship with respect to the earth frame expressed in the earth frame

$v^e_{es} \Rightarrow$ Velocity of the ship with respect to the earth frame expressed in the earth frame

$a^e_{is} \Rightarrow$ Acceleration of the ship with respect to the inertial frame expressed in the earth frame

$R_s^e \Rightarrow$ Rotation matrix from the ship coordinate system to the earth coordinate system

$\omega_{is}^s \Rightarrow$ Angular rate of the ship frame with respect to the inertial frame expressed in the ship frame

$\omega_{es}^s \Rightarrow$ Angular rate of the ship frame with respect to the earth frame expressed in the ship frame

$\omega_{ie}^s \Rightarrow$ Angular rate of the earth frame with respect to the inertial frame expressed in the ship frame

$\vec{\eta}_X \Rightarrow$ Normally distributed $n-$dimensional random vector

$\vec{\mu}_X \Rightarrow$ Mean of the normal random variable $\vec{\eta}_X$

$P_X \Rightarrow$ Covariance of the normal random variable $\vec{\eta}_X$

$pd_X(\vec{\varepsilon}) \Rightarrow$ Probability density function of random variable $\vec{\eta}_X$

$\mathbb{E}[.] \Rightarrow$ Expected value operator

$\vec{\beta}(t) \Rightarrow$ Vectorial Brownian motion

$Q_\beta(t) \Rightarrow$ Covariance of the vectorial Brownian motion $\vec{\beta}(t)$

$\vec{s}(t) \Rightarrow$ Kalman filter state

$F_S(t) \Rightarrow$ Kalman filter dynamics matrix

$B_S(t) \Rightarrow$ Kalman filter input matrix

$G_S(t) \Rightarrow$ Kalman filter noise input matrix

$\vec{\eta}_S(t) \Rightarrow$ Gaussian random variable representing the Kalman filter estimation

$\vec{\mu}_S(t) \Rightarrow$ Mean of the Kalman filter estimation at time t

$P_S(t) \Rightarrow$ Covariance of the Kalman filter estimation at time t

$\vec{\mu}_{S_0} \Rightarrow$ Initial mean of the Kalman filter estimation

$P_{S_0} \Rightarrow$ Initial covariance of the Kalman filter estimation

$\vec{z}(t) \Rightarrow$ Kalman filter measurements

$H_Z(t) \Rightarrow$ Kalman filter measurement matrix

$\vec{\eta}_Z(t) \Rightarrow$ Kalman filter measurement noise

$R_Z(t) \Rightarrow$ Kalman filter measurement covariance

$\vec{\mu}_S(t_k^-), P_S(t_k^-) \Rightarrow$ Mean and covariance of the estimator before an update at time t_k

$\vec{\mu}_S(t_k^+), P_S(t_k^+) \Rightarrow$ Mean and covariance of the estimator after an update at time t_k

$K(t_k) \Rightarrow$ Kalman gain matrix at time t_k

$f(\vec{s}(t), \vec{u}(t), t) \Rightarrow$ Nonlinear state dynamics function at time t

$F_S(\vec{\mu}_S(t), t) = \left.\frac{\partial f}{\partial s}\right|_{\vec{s}(t)=\vec{\mu}_S(t)} \Rightarrow$ Extended Kalman filter dynamics matrix

$h(\vec{s}(t), t) \Rightarrow$ Nonlinear measurement function at time t

$H(\vec{\mu}_S(t_i^-), t) = \left.\frac{\partial h}{\partial s}\right|_{\vec{s}(t)=\vec{\mu}_S(t_i^-)} \Rightarrow$ Extended Kalman filter measurement matrix

$f_{is}^s(t) \Rightarrow$ Accelerometer truth specific force with respect to the ship frame at time t

$b_a \Rightarrow$ Accelerometer bias

$M_a \Rightarrow$ Accelerometer scale-factor and cross-coupling matrix

$\eta_{fa} \Rightarrow$ Accelerometer random noise

$w_{fa} \Rightarrow$ Covariance of the accelerometer random noise η_{fa}

$w_{ba} \Rightarrow$ Covariance of the accelerometer bias b_a variation

$\hat{f}_{is}^s(t) \Rightarrow$ Accelerometer measurement of specific force with respect to the ship frame at time t

$\omega_{is}^s(t) \Rightarrow$ Gyrometer truth angular rate with respect to the ship frame at time t

$b_g \Rightarrow$ Gyrometer bias

$M_g \Rightarrow$ Gyrometer scale-factor and cross-coupling matrix

$G_g \Rightarrow$ Gyrometer gravity-cross-coupling matrix

$\eta_{\omega g} \Rightarrow$ Gyrometer random noise

$w_{\omega g} \Rightarrow$ Covariance of the gyrometer random noise $\eta_{\omega g}$

$w_{bg} \Rightarrow$ Covariance of the gyrometer bias b_g variation

$\hat{\omega}_{is}^s(t) \Rightarrow$ Gyrometer measurement of angular rate with respect to the ship frame at time t

$\hat{p}_{es}^e(t) \Rightarrow$ INS estimation at time t for the position of the ship with respect to the earth frame expressed in the earth frame

$\hat{v}_{es}^e(t) \Rightarrow$ INS estimation at time t for the velocity of the ship with respect to the earth frame expressed in the earth frame

$\hat{R}_s^e(t) \Rightarrow$ INS estimation at time t for the rotation matrix from the ship coordinate system to the earth coordinate system

$\alpha_{is}^s(t) \Rightarrow$ Angle increment measured by the gyrometer during the INS sampling interval

$\bar{R}_s^e(t) \Rightarrow$ INS estimation at time t for the average rotation matrix from the ship coordinate system to the earth coordinate system during the INS sampling interval

$\hat{f}_{is}^e(t) \Rightarrow$ INS estimation at time t for the accelerometer measurement of specific force with respect to the earth frame

$p_{sa}^s \Rightarrow$ Position of the user antenna from a GNSS with respect to the ship frame

$\hat{p}_{sa}^s \Rightarrow$ Estimation for the position of the user antenna from a GNSS with respect to the ship frame

$p_{ea}^e \Rightarrow$ Position of the user antenna from a GNSS with respect to the earth frame

$\hat{p}^e_{ea} \Rightarrow$ Estimation for the position of the user antenna from a GNSS with respect to the earth frame

$p^e_{es,j} \Rightarrow$ Position of the $j-$th satellite from a GNSS with respect to the earth frame

$\hat{p}^e_{es,j} \Rightarrow$ Estimation for the position of the $j-$th satellite from a GNSS with respect to the earth frame

$\vec{\rho} \Rightarrow$ Range measurements in the user antenna from a GNSS

$\rho_j \Rightarrow$ Range measurement between the $j-$th satellite and the user antenna from a GNSS

$\eta_{\rho p} \Rightarrow$ Random noise in the range measurement from the user antenna

$w_{\rho p} \Rightarrow$ Covariance of the range random noise $\eta_{\rho p}$

$\hat{\rho}_j \Rightarrow$ Estimation for range measurement between the $j-$th satellite and the user antenna from a GNSS

$\delta\rho_{rc} \Rightarrow$ Range error from the user antenna due to the receiver clock offset

$\delta\hat{\rho}_{rc} \Rightarrow$ Estimation for the range error from the user antenna due to the receiver clock offset

$v^e_{ea} \Rightarrow$ Velocity of the user antenna from a GNSS with respect to the earth frame

$\hat{v}^e_{ea} \Rightarrow$ Estimation for the velocity of the user antenna from a GNSS with respect to the earth frame

$v^e_{es,j} \Rightarrow$ Velocity of the $j-$th satellite from a GNSS with respect to the earth frame

$\hat{v}^e_{es,j} \Rightarrow$ Estimation for the velocity of the $j-$th satellite from a GNSS with respect to the earth frame

$\dot{\rho}_j \Rightarrow$ Range-rate measurement between the $j-$th satellite and the user antenna from a GNSS

$\eta_{\rho v} \Rightarrow$ Random noise in the range-rate measurement from the user antenna

$w_{\rho v} \Rightarrow$ Covariance of the range-rate random noise $\eta_{\rho v}$

$\hat{\dot{\rho}}_j \Rightarrow$ Estimation for the range-rate measurement between the $j-$th satellite and the user antenna from a GNSS

$\delta\dot{\rho}_{rc} \Rightarrow$ Range-rate error from the user antenna due to the receiver clock drift

$\delta\hat{\dot{\rho}}_{rc} \Rightarrow$ Estimation for the range-rate error from the user antenna due to the receiver clock drift

$S_\psi \Rightarrow$ Compass coordinate system

$p^s_{s\psi} \Rightarrow$ Position of the compass with respect to the ship frame

$p^e_{e\psi} \Rightarrow$ Position of the compass with respect to the earth frame

$R^s_\psi \Rightarrow$ Rotation matrix from the compass coordinate system to the ship coordinate system

$R^e_\psi \Rightarrow$ Rotation matrix from the compass coordinate system to the earth coordinate

system

$R^n_\psi \Rightarrow$ Rotation matrix from the compass coordinate system to the navigation coordinate system

$\psi_{n\psi} \Rightarrow$ Angle between the compass and the north direction from the navigation frame $(\vec{x}_n)$

$\eta_\psi \Rightarrow$ Compass random noise

$w_\psi \Rightarrow$ Covariance of the compass random noise η_ψ

$\hat{\psi}_{n\psi} \Rightarrow$ Compass measurement of the angle between the compass and the north direction from the navigation frame $(\vec{x}_n)$

$\hat{p}^s_{sa} \Rightarrow$ Estimation for the position of the user antenna from a GNSS with respect to the ship frame

$\hat{p}^s_{s\psi} \Rightarrow$ Estimation for the position of the compass with respect to the ship frame

$\hat{R}^s_\psi \Rightarrow$ Estimation for the rotation matrix from the compass coordinate system to the ship coordinate system

$\hat{w}_{fa} \Rightarrow$ Estimation for the covariance of the accelerometer random noise η_{fa}

$\hat{w}_{\omega g} \Rightarrow$ Estimation for the covariance of the gyrometer random noise $\eta_{\omega g}$

$\hat{w}_{ba} \Rightarrow$ Estimation for the covariance of the accelerometer bias b_a variation

$\hat{w}_{bg} \Rightarrow$ Estimation for the covariance of the gyrometer bias b_g variation

$\hat{w}_{\rho p} \Rightarrow$ Estimation for the covariance of the range random noise $\eta_{\rho p}$

$\hat{w}_{\rho v} \Rightarrow$ Estimation for the covariance of the range-rate random noise $\eta_{\rho v}$

$\hat{w}_\psi \Rightarrow$ Estimation for the covariance of the compass random noise η_ψ

$\acute{p}^e_{es} \Rightarrow$ Integrated estimation for the position of the ship with respect to the earth frame expressed in the earth frame

$\acute{v}^e_{es} \Rightarrow$ Integrated estimation for the velocity of the ship with respect to the earth frame expressed in the earth frame

$\acute{R}^e_s \Rightarrow$ Integrated estimation for the rotation matrix from the ship coordinate system to the earth coordinate system

$S_c \Rightarrow$ Camera coordinate system

$O_c \Rightarrow$ Origin of the camera coordinate system

$p^s_{sc} = [p^{s,i}_{sc}, p^{s,j}_{sc}, p^{s,k}_{sc}]^T \Rightarrow$ Position of the camera with respect to the ship coordinate system

$[\vec{\chi}_c, \vec{\gamma}_c, \vec{\kappa}_c] \Rightarrow$ Axes of the camera coordinate system

$[\vec{\chi}^s_c, \vec{\gamma}^s_c, \vec{\kappa}^s_c] \Rightarrow$ Axes of the camera coordinate system described in the ship coordinate system

$\vec{\chi}^s_c = [\chi^{s,i}_c, \chi^{s,j}_c, \chi^{s,k}_c]^T \Rightarrow$ Three-dimensional representation for axis $\vec{\chi}^s_c$ with respect to the ship coordinate system

$\vec{\gamma}_c^s = [\gamma_c^{s,i}, \gamma_c^{s,j}, \gamma_c^{s,k}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{\gamma}_c^s$ with respect to the ship coordinate system

$\vec{\kappa}_c^s = [\kappa_c^{s,i}, \kappa_c^{s,j}, \kappa_c^{s,k}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{\kappa}_c^s$ with respect to the ship coordinate system

$R_c^s \Rightarrow$ Rotation matrix from the camera coordinate system to the ship coordinate system

$T_c^s \Rightarrow$ Transformation matrix from the camera coordinate system to the ship coordinate system

$R_s^c \Rightarrow$ Rotation matrix from the ship coordinate system to the camera coordinate system

$T_s^c \Rightarrow$ Transformation matrix from the ship coordinate system to the camera coordinate system

$\vec{p}_{cQ}^c = [p_{cQ}^{c,\chi}, p_{cQ}^{c,\gamma}, p_{cQ}^{c,\kappa}]^T \Rightarrow$ Position of an arbitrary point Q with respect to the camera coordinate system

$f \Rightarrow$ Focal distance of ideal pinhole camera

$[u^{ph}, v^{ph}] \Rightarrow$ Measurements from ideal pinhole camera model

$[\Delta_u, \Delta_v] \Rightarrow$ Size of each pixel unit from the sensor array in the image plane

$[u_0, v_0] \Rightarrow$ Principal point of the image in pixel units

$K_{in} = [f\Delta_u^{-1}, f\Delta_v^{-1}, u_0, v_0] \Rightarrow$ Intrinsic camera parameters

$[u_Q, v_Q] \Rightarrow$ Image coordinates of an arbitrary point Q described by its position p_{cQ}^c in the camera frame

$\hat{K}_{in} = [\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0] \Rightarrow$ Estimation for the intrinsic camera parameters

$\hat{p}_{sc}^s = [p_{sc}^{s,i}, p_{sc}^{s,j}, p_{sc}^{s,k}]^T \Rightarrow$ Estimation for the position of the camera with respect to the ship coordinate system

$\hat{R}_c^s \Rightarrow$ Estimation for the rotation matrix from the camera coordinate system to the ship coordinate system

$[\hat{u}_Q, \hat{v}_Q] \Rightarrow$ Estimation for the image coordinates of an arbitrary point Q described by its position p_{cQ}^c in the camera frame

$p_{wc}^w = [p_{wc}^{w,x}, p_{wc}^{w,y}, p_{wc}^{w,z}]^T \Rightarrow$ Position of the camera with respect to the world coordinate system

$\hat{p}_{wc}^w = [\hat{p}_{wc}^{w,x}, \hat{p}_{wc}^{w,y}, \hat{p}_{wc}^{w,z}]^T \Rightarrow$ Estimation for the position of the camera with respect to the world coordinate system

$R_c^w \Rightarrow$ Rotation matrix from the camera coordinate system to the world coordinate system

$\hat{R}_c^w \Rightarrow$ Estimation for the rotation matrix from the camera coordinate system to the world coordinate system

$d_s \Rightarrow$ Draft of the ship at the section of the camera

$h_w \Rightarrow$ Vertical distance between the camera and the water

$\Pi_{INS} \Rightarrow$ Set of parameters representing a INS in simulation

$\Pi_{GNSS} \Rightarrow$ Set of parameters representing a GNSS in simulation

$\Pi_{CMPS} \Rightarrow$ Set of parameters representing a compass in simulation

$\Pi_{LCI} \Rightarrow$ Set of parameters representing a loosely coupled integration tracker in simulation

$S_K \Rightarrow$ Relative state estimator coordinate system

$R_K \Rightarrow$ Rotation matrix from the relative state estimator coordinate system S_K to the camera coordinate system

$R_K^{ri} \Rightarrow$ Vector representing the $i-$th row of the rotation matrix R_K

$r_{kj}^{ri} \Rightarrow j-$th component from the $i-$th row of the rotation matrix R_K

$\vec{c}_K = [c_x^K, c_y^K, c_z^K]^T \Rightarrow$ Camera center in the relative state estimator coordinate system S_K

$\vec{p}_K = [p_x^K, p_y^K, p_z^K]^T \Rightarrow$ Point p described with coordinates in the relative state estimator coordinate system

$S_\Pi \Rightarrow$ Plane coordinate system

$O_\Pi \Rightarrow$ Origin of the plane coordinate system

$p_{c\Pi}^c = [p_{c\Pi}^{c,\chi}, p_{c\Pi}^{c,\gamma}, p_{c\Pi}^{c,\kappa}]^T \Rightarrow$ Position of the plane origin O_Π described in the camera coordinate system

$[\vec{x}_\Pi, \vec{y}_\Pi, \vec{z}_\Pi] \Rightarrow$ Axes of the plane coordinate system

$[\vec{x}_\Pi^c, \vec{y}_\Pi^c, \vec{z}_\Pi^c] \Rightarrow$ Axes of the plane coordinate system described in the camera coordinate system

$\vec{x}_\Pi^c = [x_\Pi^{c,\chi}, x_\Pi^{c,\gamma}, x_\Pi^{c,\kappa}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{x}_\Pi^c$ with respect to the camera coordinate system

$\vec{y}_\Pi^c = [y_\Pi^{c,\chi}, y_\Pi^{c,\gamma}, y_\Pi^{c,\kappa}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{y}_\Pi^c$ with respect to the camera coordinate system

$\vec{z}_\Pi^c = [z_\Pi^{c,\chi}, z_\Pi^{c,\gamma}, z_\Pi^{c,\kappa}]^T \Rightarrow$ Three-dimensional representation for axis $\vec{z}_\Pi^c$ with respect to the camera coordinate system

$R_\Pi^c \Rightarrow$ Rotation matrix from the plane coordinate system to the camera coordinate system

$T_\Pi^c \Rightarrow$ Transformation matrix from the plane coordinate system to the camera coordinate system

$p_{cP_\pi}^c \Rightarrow$ Position of an arbitrary point P_Π from the plane described in the camera coordinate system

$n_x \Rightarrow$ Number of rectangles in the $x-$direction of a planar rectangular highlight

$n_y \Rightarrow$ Number of rectangles in the $y-$direction of a planar rectangular highlight

$[\Delta L_x \times \Delta L_y] \Rightarrow$ Size of each rectangle of a planar rectangular highlight

$r_c \Rightarrow$ List of different circle radius to be highlighted with a planar circular highlight

$N_\theta \Rightarrow$ Number of sections for each circle from a planar circular highlight

CONTENTS

1 INTRODUCTION

One of the most important challenges associated with enabling world-wide commercial navigation relates to safety. Institutions, regulations and procedures have been developed over time to ensure safety at sea. Over the years, equipment have been designed to assist ship operations as new technologies arises. Particularly, fields such as electronics, radio, computer science, automatic control engineering, data presentation and space technologies led to the development of standardized integrated equipment [KOPACZ; MORGAŚ; URBAŃSKI, 2004]. These integrated systems became indispensable to the safety and economic feasibility of commercial maritime navigation.

A particularly important type of system for maintaining safety at sea is classified as *Integrated Bridge Systems* (**IBS**) [INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2007]. These systems integrate and display useful information to support decisions of an operator. As such navigational equipment became standardized and popularized [SAROLIC, 2004], companies started to seek for competitive advantages by investigating innovative trends and its applicability to their equipment. A notable ongoing trend aims at the digitization and automation of maritime processes [HEILIG; LALLA-RUIZ; VOSS, 2017], [HEILIG; SCHWARZE; VOSS, 2017]. Nowadays, with remarkable improvements in general purpose computing resources, most of such navigational equipment uses a **c**omputer **p**rocessing **u**nit (**CPU**) that periodically collects outputs from onboard sensors. These outputs are rendered in real-time on a monitor display with data presentation techniques to facilitate the visualization of an operator.

Alongside this development in computing devices which enabled the popularization of digital equipment, one can observe similar improvements in camera devices over the last years. These advancements enabled the development of computer vision systems for many purposes. An interesting type of computer vision technique that this work focuses is commonly referred as *Augmented-Reality*. The technique enhances an original scene image with synthetic elements [AZUMA, 1997]. In this context, tools with *augmented-reality* (**AR**) methods have been proposed for presenting multiple navigational data during

maritime operations [LAERA et al., 2021]. Such tools may facilitate the interpretation of navigational information by operators.

The current work is a preliminary investigation towards the development of an AR system for navigational assistance. Simple methodologies for the generation of augmented virtual elements in videos of maritime navigation are proposed. Augmented elements proposed in the present work may be used for research and validation of conceptual navigational assistance AR system designs. The methods are validated with videos from the TPN-USP ship maneuvering simulator and real experiments. The TPN-USP Simulation Center is the brazilian largest ship maneuvering simulation center, equipped with 3 full-mission simulators and 3 tug stations, as well as 1 part-task simulator. The mathematical model adopted in the simulator is described in [TANNURI et al., 2014], and the real-time visualization framework for the generation of realistic images of maritime scenario is presented in [MAKIYAMA et al., 2020].

Proposed enhanced scenes exemplifies potential assistance applications for a ship navigating in restricted waters. Note that different operators performing a same task might find different sets of information better suited than others. Thus, it is not the goal of this work to find optimal representations for particular ship operations, although it is an interesting topic to be further investigated.

1.1 Motivation

As the incorporation of arising technology in maritime processes and equipment have been historically very important for the development of commercial navigation, it is interesting to be watchful for current technologies that might impact the future of commercial navigation. Two technological trends motivates the present work: the *digital transformation of maritime operations* and *autonomous maritime navigation*.

In [HEILIG; SCHWARZE; VOSS, 2017], a historical review of digital transformation in seaports is discussed. The authors identify three major stages of transformations in seaports which are going to be briefly presented. Each one of these stages are further detailed under the framework presented in [VENKATRAMAN, 1994]. This framework proposes five possible levels of business transformations with information systems and technology: (1) *Localized Exploitation*; (2) *Internal Integration*; (3) *Business Process Redesign*; (4) *Business Network Redesign*; and (5) *Business Scope Redefinition*.

The first trend dates from the beginning of the 1960s until the late 1980s and is pre-

sented as the *Transformation to Paperless Procedures*. In the need of improvements in overall planning, management and coordination of operational activities, electronic devices have been employed to ensure efficient information flows. Such equipment improved the availability and management of information enabling better planning and execution of internal processes.

The next transformation trend can be summarized as the *Transformation to Automated Procedures*. In the beginning of the 1990s, typical port operations started to be automated with the assistance of laser technologies. By the end of the 2000s, these automated procedures became an important competitive advantage as the global economic crisis of 2008-2009 led to a more strict evaluation and selection of ports.

Finally, the most recent generation of digital transformations is presented in the work as the *Transformation to Smart Procedures*. Fueled by recent advancements in promising fields such as the *internet of things*, *big data*, *mobile computing* and *cloud computing*; the application of such technologies within the maritime industry became to be investigated. The main idea of this transformation phase is to provide relevant information based on real-time data for actors actively involved in transport activities. Raw data from multiple sensors is processed and then transferred to a central information system that distributes it to the many involved actors and decision makers.

Alongside these efforts for automating port operations, there is an ongoing research on the automation of ship navigation as well. In June 2017, the *Maritime Safety Committee* (**MSC**) of the *International Maritime Organization* (**IMO**) carried out a scoping exercise to discuss how the operation of *Maritime Autonomous Surface Ships* (**MASS**) may be introduced in IMO instruments [INTERNATIONAL MARITIME ORGANIZATION, 2017]. In the next year, the MSC endorsed a framework to analyze the most appropriate way of addressing such operations [INTERNATIONAL MARITIME ORGANIZATION, 2018a], [INTERNATIONAL MARITIME ORGANIZATION, 2018b]. Ships have been categorized according to four degrees of autonomy: (1) *Ship with automated processes and decision support*, (2) *Remotely controlled ship with seafarers onboard* (3) *Remotely controlled ship without seafarers on board* and (4) *Fully autonomous ship*. Guidelines for MASS trials were also discussed for ensuring safety in the development of this technology. In 2019, the MSC formally approved guidelines for MASS trials [INTERNATIONAL MARITIME ORGANIZATION, 2019]. The main recommendation is that trials should be conducted ensuring at least the same degree of safety, security and protection of the environment as current instruments. Any personnel involved in these trials should be appropriately qualified and experienced regarding the technologies to be validated. Possible

risks associated with trials should be identified along with measures to reduce these risks.

In the context of ships with the first degree of autonomy, a decision support equipment that is being investigated proposes the utilization of augmented reality techniques [AZUMA, 1997] for an optimal presentation of navigational data to ship operators. A systematic review regarding the use of augmented reality technology for data-visualization in the field of maritime navigation is presented in [LAERA et al., 2021]. The work analyze each study in view of its maturation regarding the incorporation of this technology into maritime operations. An interesting set of definitions for such analysis is referred as *Technology Readiness Levels* (TRL), which are shown in Table 1.

Table 1: Definitions of technology readiness levels (TRL)

TRL 1	Basic principles observed
TRL 2	Technology concept formulated
TRL 3	Experimental proof of concept
TRL 4	Technology validated in laboratory
TRL 5	Technology validated in relevant environment (industrially relevant environment in the case of key enabling technologies)
TRL 6	Technology demonstrated in relevant environment (industrially relevant environment in the case of key enabling technologies)
TRL 7	System prototype demonstration in operational environment
TRL 8	System complete and qualified
TRL 9	Actual system proven in operational environment

Source: [COMMISSION DECISION C(2014)4995, 2014]

Most solutions reviewed by the work presented in [LAERA et al., 2021] have been classified into initial *technology readiness levels*, which indicates that the application of AR in the nautical field has not been sufficiently investigated yet. In such context, the current work formulates technology concepts for a preliminary investigation of AR in the nautical field (**TRL 2**). As *augmented reality systems* are generally focused on the enhancement of video streams from cameras, simple methodologies for generating metric virtual elements in videos of maritime applications are proposed towards navigational assistance in a scenario of a ship navigating in restricted waters. These methods are pursued as a means for popularization of *augmented reality systems* within maritime contexts.

Considering computer vision techniques as a new technology to be incorporated in maritime operations, equipment based on augmented-reality techniques could be useful to investigate the performance of general computer vision methods in the field. As the challenging conditions regarding maritime navigation are usually not present in other

types of applications, typical computer vision techniques may not perform well enough in view of all safety requirements. Thus, such equipment may also assist in the adaptation of general computer vision results into specific solutions for the automation of maritime operations.

1.2 Objectives

The main objective of the present work is the proposal of preliminary methods towards the development of an *augmented reality* system for navigational assistance in restricted waters. Proposed methodologies have been validated with videos from the TPN-USP ship maneuvering simulator and from experiments around the *Paranagua Port* at the state of Paraná. Both videos are recorded from experiments of a ship navigating in restricted waters with an onboard fixed camera.

As complementary objectives, the author expects to contribute with the overall safety of maritime operations by assisting in the popularization of *Augmented-Reality* methods in *Maritime Decision Support Systems*. Guidelines for simple features and prototypes are presented for a particular scenario of a ship traversing a channel delimited by nautical buoys. Potential assistance applications are exemplified for this scenario assuming knowledge of the navigation scene from additional onboard equipment. Further research with computer vision technologies are indicated for similar scenarios without external knowledge from additional equipment.

1.3 Structure of the Text

On chapter 2 is presented an overview of legal instruments developed over time to ensure *safety at sea*. Technological trends are discussed with a particular emphasis on the development of navigation equipment. The end of the chapter overviews *augmented reality* applications within the maritime industry.

Chapter 3, 4 presents foundations for the development of *Augmented Reality Systems* within the context of maritime navigation. Useful navigational definitions are proposed on chapter 3 in order to adequately describe the ship navigation and its surroundings. Camera projection models are subsequently presented on chapter 4 as foundations for proposed virtual visual elements. Chapter 5 proposes a *monitor augmented reality* system architecture and chapter 6 discusses the experimental setup used throughout the current

work.

Chapter 7 discusses methods for estimating mechanical parameters from the camera scene. An *extended Kalman filter* is proposed for estimating the relative state of surrounding obstacles from camera measurements. The estimator is validated with measurements computed from camera models and with measurements determined by image processing algorithms. The integration of camera measurements in the estimation of the ship state is also briefly discussed.

Chapter 8 presents methods for the augmentation of maritime scenes. Proposed methods are developed assuming that aforementioned installation parameters are known. The content of the section focuses on the synthesis of pertinent enhanced images. The goal is to assist operators with the scene perception. External information about the scene is assumed to be available.

Chapter 9 discusses potential assistance applications combining aforementioned features. The development of a MAR prototype is also briefly discussed. Finally, chapter 10 presents conclusions from the current work indicating further research.

2 BIBLIOGRAPHIC REVIEW

The division of this chapter is in two sections. The first section overviews the evolution of safety systems for maritime navigation as presented on review articles from *The Journal of Navigation*. Its main components and elements are presented according to [KOPACZ; MORGAŚ; URBAŃSKI, 2001]. Fundamental changes in maritime navigation driven by technological trends are presented in view of [KOPACZ; MORGAŚ; URBAŃSKI, 2004]. Additional technological aspects regarding the digitization of maritime operations are presented based on another review article [SANCHEZ-GONZALEZ et al., 2019]. The last section from this chapter presents interesting research regarding augmented reality applications within maritime environments [VASILIJEVIC; BOROVIĆ; VUKIĆ, 2011], [LAERA et al., 2021].

2.1 Safety Systems for Maritime Navigation

In [KOPACZ; MORGAŚ; URBAŃSKI, 2001] there is an overall review of the institutions, regulations and procedures concerned with *safety at sea*. *Safety at sea* is defined as *such desirable conditions of human activity at sea that do not endanger human life and property, and are not harmful to the maritime environment*. Such conditions of activity have been specified in many legal documents. The proposed term is further expressed by means of four constituent parts: (1) *Technological and operational ships' safety*, (2) *safety of navigation*, (3) *safety of persons in distress* and (4) *prevention of pollution of the maritime environment from ships*. The set of institutions, standards and procedures to guarantee *safety at sea* is defined as the *maritime safety system*.

The main organization responsible for *safety at sea* is the *International Maritime Organization* (**IMO**). It is an agency of the *United Nation Organization* (**UN**). There are different organs within the IMO regulating the legal instruments of the *maritime safety system*. One of the most important document regarding *safety at sea* is the International Convention on 'Safety Of Life At Sea' (**SOLAS**) [INTERNATIONAL MARITIME OR-

GANIZATION, 1974]. This convention have been adopted at a London Conference in 1914 with posterior amendments over the years. The main objective of the SOLAS convention is to specify minimum standards for the construction, equipment and operation of ships, compatible with their safety. Another very important convention that directly specifies *safety at sea* is the International Convention on 'Present of Pollution from Ships' (**MARPOL**) [INTERNATIONAL MARITIME ORGANIZATION, 1973]. The Convention includes regulations aimed at preventing and minimizing pollution from ships.

Note that foundations for ensuring *safety at sea* are distributed within different research fields. Among these domains of safety, *safety of navigation* is a major constituent of safety at sea. It can be considered as *such conditions of conducting the ships at sea which ensure that ships are not endangered by collisions, stranding or storm damage.* There are resolutions from the IMO regarding navigational performance standards and operational procedures further specifying *safety of navigation.* Similarly as with the aforementioned concept of *safety at sea*, there are several domains that contributes for ensuring *safety of navigation.*

Over the years, new equipment described within *International Maritime Conventions* incorporated new technologies that enabled improvements in the overall efficiency of maritime navigation. In [KOPACZ; MORGAŚ; URBAŃSKI, 2004], the authors discusses how emerging technologies shifted main components of the *maritime safety system* towards a *maritime safety and efficiency system.* The most important technological progresses for maritime navigation are related with fields of science such as electronics, computer science, automatic control engineering and electronic data processing/transmission/presentation. The progress in these fields enabled the development of equipment towards the integration of all ship processes into one centralized control system. Further research for the development of such integrated systems have been proposed.

As more onboard equipment became available for operators, an increase in the workload to analyze the corresponding information have been perceived. The need for interaction with such digital equipment became indispensable, thus the scope of desired competences for operators within the maritime industry required further discussion. Proper training for analyzing information from different equipment became an important topic. Further developments to electronic data processing, transmission, display and recording technologies have been suggested as an important factor for further improvements in safety and efficiency of maritime operations.

Note that there are many challenges for the deployment of new equipment in the

maritime industry. The marine environment is characterized by visibility issues such as atmospheric phenomena. Particularly, in the case of offshore navigation, the lack of landmarks adds significantly complexity as seafarers have no visual reference for self-localization and immediate planning. Therefore, onboard ship equipment needs to be very reliable with regard to its accurateness, robustness and user-friendliness. Such requisites may considerably increase overall costs of navigational equipment.

In [SANCHEZ-GONZALEZ et al., 2019], a systematic literature review verifies the application of eight innovative digital domains for maritime operations: *autonomous vehicles and robotics*; *artificial intelligence*; *big data*; *virtual, augmented and mixed reality*; *internet of things*; *cloud and edge computing*; *digital security*; and *3D printing and additive engineering*. The objective of the review is to present the status of digitization among the maritime transport chain to discover possible opportunities for improving the performance of current processes.

More than 2900 works related to *the digitization of maritime transport* were examined in view of: (1) their contribution in innovation; (2) use of scientific methods; (3) and relevance of the study. Important works were selected resulting in 99 papers that were further analyzed. The maritime transport industry have been divided in three sectors for the purpose of the analysis: *ship design and shipbuilding*; *shipping*; and *ports*. The authors were able to show that the digitization of maritime transport is advancing at different speeds for each different sector by analyzing the relative number of published works over the years. Table 2 presents the number of published manuscripts per domain of study.

Table 2: References per domain

Domain	**Number of Manuscripts**
Robotics	36
Artificial Intelligence	19
Big Data	16
Virtual Reality	10
Internet of Things	9
Cloud	5
Security	3
3D Printing	1
TOTAL	99

Source: [SANCHEZ-GONZALEZ et al., 2019]

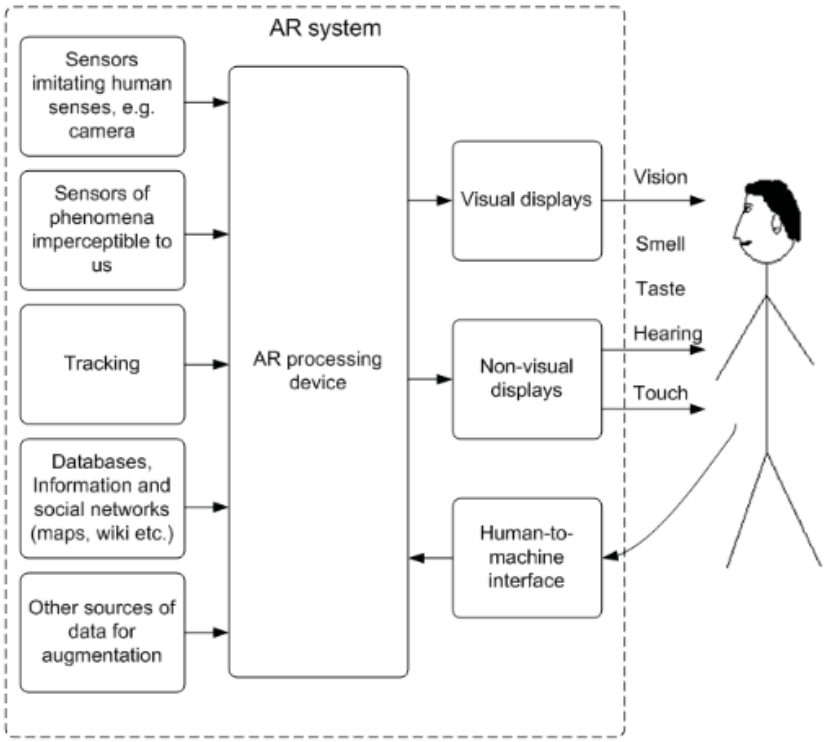
As for future research, the authors suggests that developments in all aforementioned

domains and sectors should be pursued as them could enable promising applications to the maritime industry. In this context, the current work investigates the application of a *virtual reality* technique usually referred as *augmented reality*. The development of *augmented reality* as research field is described in Appendix A.

2.2 Augmented Reality Applications within Maritime Environments

This section describes representative examples of augmented reality applications within maritime environments. A review of potential *augmented reality* applications for the maritime sector is presented in [VASILIJEVIC; BOROVIĆ; VUKIĆ, 2011]. The authors proposed a definition for *Augmented-Reality* (**AR**) systems with three main components. Figure 1 shows a diagram for the proposed operation of AR systems.

Figure 1: AR system inputs and outputs



Source: [VASILIJEVIC; BOROVIĆ; VUKIĆ, 2011]

The first component have been defined as the *inputs*. These inputs have been further divided in three different classes: sensors, tracking devices and database links. Sensors are typically digital cameras. Tracking devices provides accurate measurements for the camera's position and attitude (orientation) in the real world. External database links

provide auxiliary relevant information for the augmentation of the AR scene. The second component have been defined as the *AR processing device*. These devices are CPUs with graphical capabilities that processes the inputs. The last component have been defined as *System-to-human interface*. This interface handles all communication between human and the AR. Two categories have been proposed for these interfaces: displays where data flows from the AR system towards operators; interfaces where operators provide inputs to the AR system. Challenges and limitations with the application of this technology in maritime navigation were also discussed by the authors. Accurate tracking methods have been presented as one of the major technical challenges for implementing these technologies in operational processes.

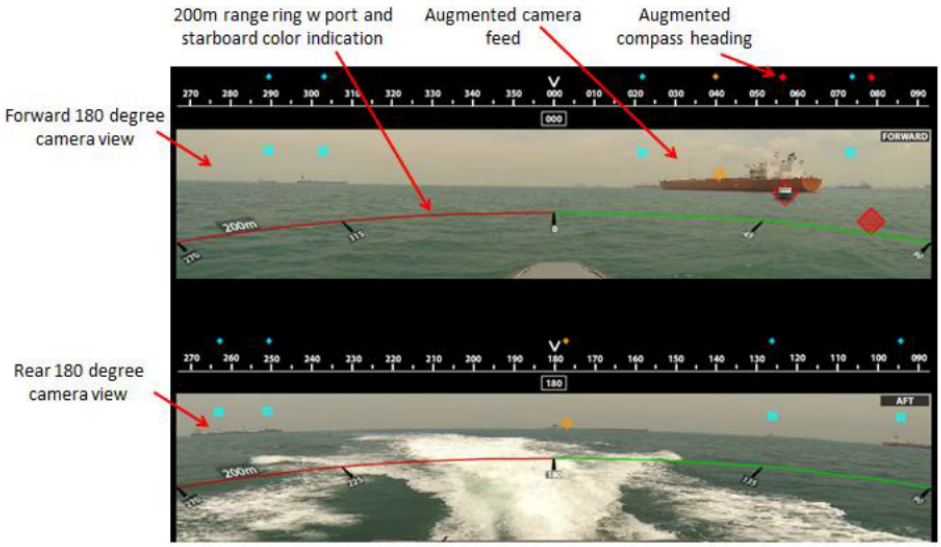
Further, a real-life outdoor Maritime Augmented Reality Navigational Assistance Application (**MARNAA**) designed to alleviate cognitive loads issues for vessels and recreational boats have been proposed in [MORGÈRE; DIGUET; LAURENT, 2014]. The authors proposes a solution for the generation of an *Augmented Electronic Navigational Chart*. Three types of application data have been presented in such charts. The first one is the *state data*, similar to a system status with information such as the course, speed, position and the current time. The second type of information have been defined as the *alert data*. Two types of such alerts have been described within the work. Collision detection alerts, in case of a collision detection perceived from data of an external Automatic Identification System (**AIS**); or speed alerts, in case of navigation within restricted areas such as channels or harbors. The third type of information is related with navigational assistance. Useful navigational data from online databases such as seamarks, landmarks and water depth are presented to assist operators.

The authors also briefly discusses hardware requirements for the implementation. The system was implemented with a *microelectromechanical* system for tracking the camera's orientation, along with a Global Positioning System (**GPS**) for determining the geolocation of the system. It has been proposed that the GPS error can vary between 1 and 5 meters while the orientation error should be inferior to a half degree. Requirements for the computing resources should be defined in view of three main tasks. The first one relates to the orientation computation implemented by *Kalman-based filters*. The second task refers to parsing the signal from the external GPS device. The third task relates to all graphical computations. The authors suggests an additional Graphical Processing Unit (**GPU**) co-processor for efficiently computing these graphical elements.

In [HONG; ANDREW; KENNY, 2015], a set of experiments were conducted to investigate the applicability of a *Maritime Augmented Reality System* (**MARS**). Participants

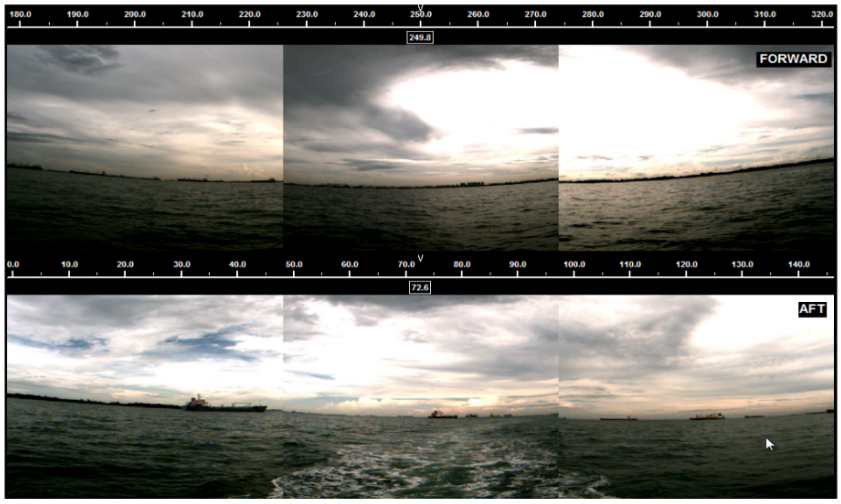
of the experiment reviewed a video footage of a prototype boat navigating in the straits of Singapore with six onboard cameras. Two scenarios were presented to the operators: one with AIS information displayed as AR annotations on the camera feeds; another with AIS information displayed in a separate screen. Participants felt that the AR provided them information which was important to aid them with respect to their situational awareness. Results for the interface with AR were generally more positive compared to the results without the AR annotations. Figure 2 shows an example of the interface with the annotations and Figure 3 shows an example of the interface without annotations.

Figure 2: MARS GUI



Source: [HONG; ANDREW; KENNY, 2015]

Figure 3: Non-AR GUI

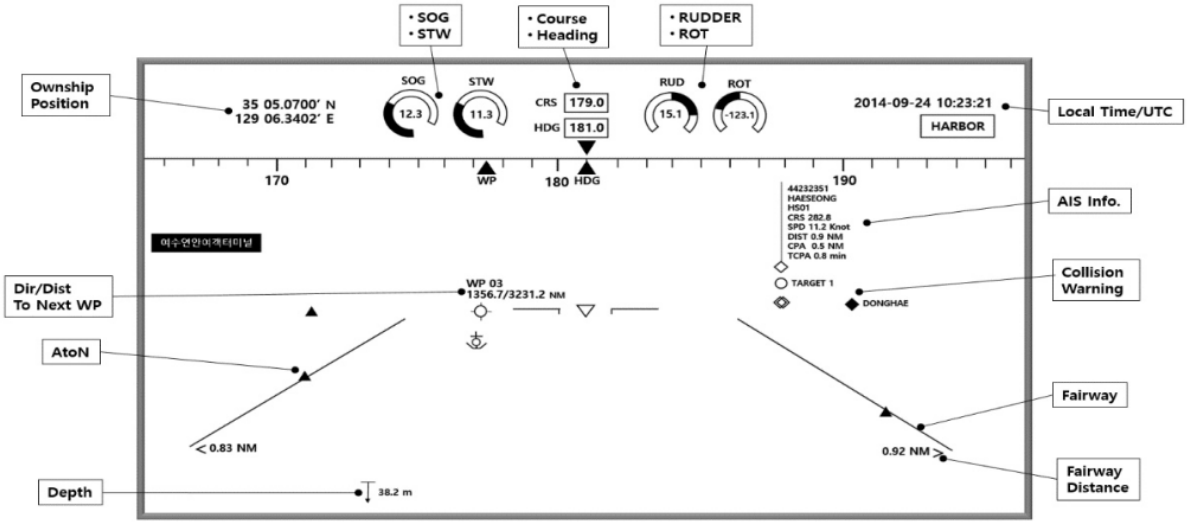


Source: [HONG; ANDREW; KENNY, 2015]

In [OH; PARK; KWON, 2016], a new concept have been proposed for a navigational system based on AR technology. The work discusses that one of the largest problem with typical bridge equipment is the provision of excessive and unnecessary information. It has been proposed that future equipment in the bridge needs to be designed taking in consideration effective information provision methods rather than simply increasing the amount of information delivered.

A survey was initially carried out to determine user requirements. Three different categories have been defined for the display information of the navigational assistant system: (1) data regarding the own-ship operation; (2) data regarding traffic ships; and (3) data from Electronic Chart Display and Information System (**ECDIS**). Own-ship data are displayed on the upper portion of the display. Data of traffic ships are overlaid on their corresponding positions in the environment. Additional information from AIS data regarding surrounding traffic ships can be displayed upon selection of the operators. The ECDIS data is displayed according to the geolocation of the ship. Figure 4 illustrates the proposed interface of the navigation system.

Figure 4: User Interface of AR Navigation System

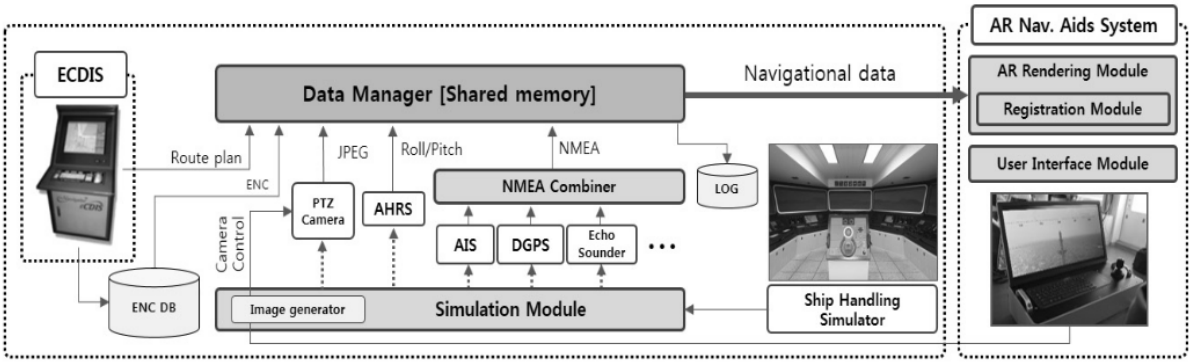


Source: [OH; PARK; KWON, 2016]

The hardware components of the system are a *Pan/Tilt/Zoom* (**PTZ**) digital camera for measuring the scene; a user console with an additional joystick for the camera control; an *Attitude and Heading Reference System* (**AHRS**) for tracking the orientation of the camera; and a NMEA combiner for the integration of navigational data such as AIS and DGPS. These components have been integrated in four software modules: *data manager module*, *user interface module*, *registration module* and *augmented image rendering*

module. Figure 5 shows a diagram representing the data flow between each module.

Figure 5: Configuration of Navigation Aids System based on AR



Source: [OH; PARK; KWON, 2016]

In [MIHOC; CATER, 2017], an assistive maritime navigation application have been developed. The work presents the usage of augmented reality as a means to present digital information from typical bridge equipment. An initial user research was conducted on board of a variety of different ships. The study concentrated on the interaction between navigating officers and the current devices which they use for navigation. Main functions have been identified for the development of a prototype that was built on a Nexus 9 tablet. The tablet's sensors include GPS, gyroscope, accelerometers, compass and camera. These sensors have been integrated in an application that calculates the GPS position of the ship, the bearing and distance to navigational aids.

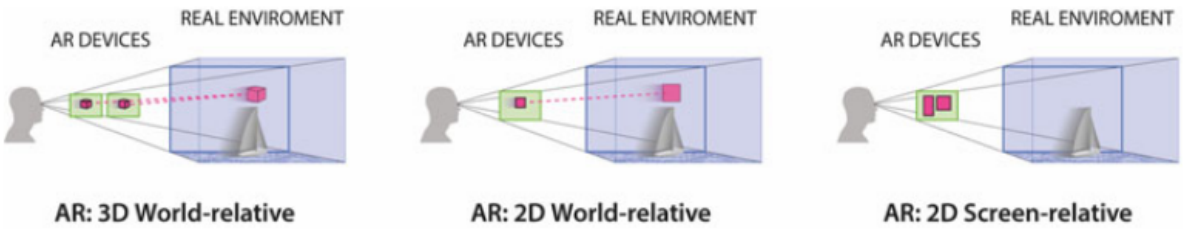
It has been reported that the initialization of the application at sea in the AR mode took over 1 minute on the Nexus 9 tablet. The application was able to correctly display proposed virtual elements. Two additional considerations about the sensors have been made. A magnetic interference with the bridge of the ship generated a continuous compass error that varied between 5 and 12 degrees. Besides, the GPS signal was maintained accurate only if the Wi-Fi was deactivated on the tablet. Despite such observations, the officers were very interested and positive about the possibility of such application be available to support navigation. The authors propose further research for the calibration of the compass of the device. Moreover, it was proposed future developments to address the issues related to the presentation of high volume of information on limited sized screens.

Moreover, a systematic review regarding the use of augmented reality (AR) technology for data-presentation in the field of maritime navigation is presented in [LAERA et al., 2021]. Three types of virtual visual elements have been defined in the work: *3D World-*

relative, *2D World-relative* and *2D Screen-relative*.

Planar areas, lines or points that are rendered on top of the camera scene as virtual visual elements were considered as *3D World-relative*. Conversely, assets that are inserted around a region from the navigation scene without being perceived as three-dimensional element were considered as *2D World-relative*. Finally, virtual elements that are not inserted in any particular part of the scene were classified as *2D Screen-relative*. Note that *2D Screen-relative* assets are not considered *augmented reality*. These different classes of virtual assets are illustrated by Figure 6.

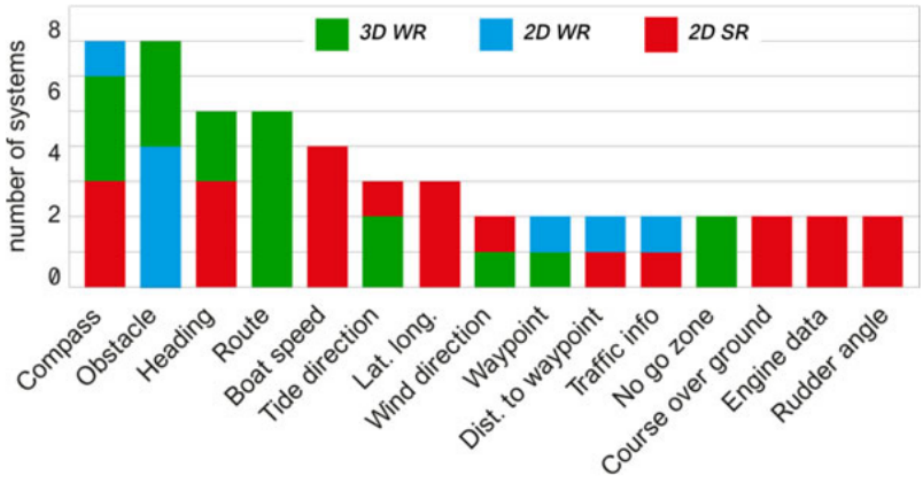
Figure 6: The spatiality of AR assets: 3D World-relative (WR), 2D World-relative (WR), 2D Screen-relative (SR)



Source: [LAERA et al., 2021]

Primitive information such as course, compass degree, boat speed and geographic coordinates were highlighted as fundamental information to be represented in AR solutions. Figure 7 presents the frequency of different types of information elements found in the studies reviewed.

Figure 7: Frequency of information elements found in the solutions analyzed by the systematic review

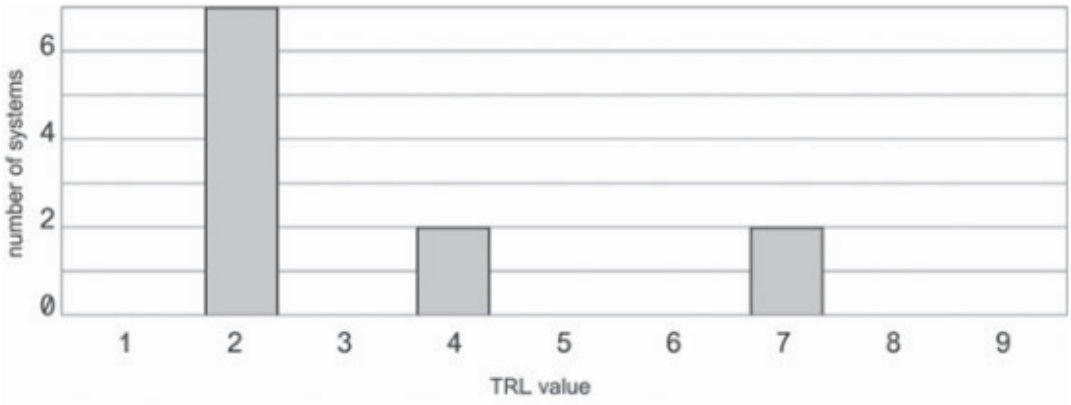


Source: [LAERA et al., 2021]

The authors discusses that there is no representative standard for displaying each information element. Further research on interface proposals is suggested as it would potentially benefit the development of AR solutions within the maritime field. Note that the usefulness of each information element depends on the particular operation of the user.

The majority of analyzed works by the review uses an architecture referred as *monitor augmented-reality* (MAR), which is cheaper and simpler than head-mounted architectures. Eleven AR maritime solutions are described within the work and further classified according to their *technology readiness level* (Table 1). Figure 8 presents a histogram illustrating the distribution of solutions according to their TRLs.

Figure 8: Classification of AR solutions regarding their *technological readiness level* (TRL)



Source: [LAERA et al., 2021]

Most solutions have been rated as *Technology concept formulated - TRL 2* (7), while few solutions have been rated as *Technology validated in laboratory - TRL 4* (2) and as *System prototype demonstration in operational environment - TRL 7* (2). Thus, the authors indicate that the application of AR in the maritime field is not yet ready for the market as most solutions have been classified under initial *technology readiness levels*.

In this context, the current work formulates technology concepts (TRL 2) towards the development of a *monitor augmented reality* system for navigational assistance. The following chapters addresses foundations for proper modeling such scenarios.

3 NAVIGATION

Consider a reference coordinate system S_w (index w) fixed in the earth described by axes $[\vec{x}_w, \vec{y}_w, \vec{z}_w]$ with origin at a point O_w. Further, consider a coordinate system S_s (index s) associated with a ship navigating. This ship coordinate system is described by axes $[\vec{i}_s, \vec{j}_s, \vec{k}_s]$ with origin at a point O_s. Each axis of the ship coordinate system can be described in the world coordinate system as a vector with three-dimensional coordinates:

$$\vec{i}_s^{\,w} = i_s^{w,x}\vec{x}_w + i_s^{w,y}\vec{y}_w + i_s^{w,z}\vec{z}_w = \begin{bmatrix} i_s^{w,x} & i_s^{w,y} & i_s^{w,z} \end{bmatrix}^T \tag{3.1}$$

$$\vec{j}_s^{\,w} = j_s^{w,x}\vec{x}_w + j_s^{w,y}\vec{y}_w + j_s^{w,z}\vec{z}_w = \begin{bmatrix} j_s^{w,x} & j_s^{w,y} & j_s^{w,z} \end{bmatrix}^T \tag{3.2}$$

$$\vec{k}_s^{\,w} = k_s^{w,x}\vec{x}_w + k_s^{w,y}\vec{y}_w + k_s^{w,z}\vec{z}_w = \begin{bmatrix} k_s^{w,x} & k_s^{w,y} & k_s^{w,z} \end{bmatrix}^T \tag{3.3}$$

Consider an arbitrary vector $\vec{v}$ and let $\vec{v}^s$ be its description in the ship coordinate system with three-dimensional coordinates:

$$\vec{v}^s = v^{s,i}\vec{i}_s + v^{s,j}\vec{j}_s + v^{s,k}\vec{k}_s = \begin{bmatrix} v^{s,i} & v^{s,j} & v^{s,k} \end{bmatrix}^T \tag{3.4}$$

It is possible to compute the corresponding three-dimensional coordinates $\vec{v}^w$ in the world coordinate system:

$$\begin{aligned} \vec{v}^s = v^{s,i}\vec{i}_s + v^{s,j}\vec{j}_s + v^{s,k}\vec{k}_s =& v^{s,i}\left(i_s^{w,x}\vec{x}_w + i_s^{w,y}\vec{y}_w + i_s^{w,z}\vec{z}_w\right) && (3.5)\\ &+ v^{s,j}\left(j_s^{w,x}\vec{x}_w + j_s^{w,y}\vec{y}_w + j_s^{w,z}\vec{z}_w\right) && (3.6)\\ &+ v^{s,k}\left(k_s^{w,x}\vec{x}_w + k_s^{w,y}\vec{y}_w + k_s^{w,z}\vec{z}_w\right) \\ \vec{v}^w = v^{w,x}\vec{x}_w + v^{w,y}\vec{y}_w + v^{w,z}\vec{z}_w =& \left(v^{s,i}i_s^{w,x} + v^{s,j}j_s^{w,x} + v^{s,k}k_s^{w,x}\right)\vec{x}_w && (3.7)\\ &+ \left(v^{s,i}i_s^{w,y} + v^{s,j}j_s^{w,y} + v^{s,k}k_s^{w,y}\right)\vec{y}_w && (3.8)\\ &+ \left(v^{s,i}i_s^{w,z} + v^{s,j}j_s^{w,z} + v^{s,k}k_s^{w,z}\right)\vec{z}_w && (3.9) \end{aligned}$$

Note that Equation 3.9 can be expressed in a matricial form:

$$\vec{v}^w = \begin{bmatrix} v^{w,x} \\ v^{w,y} \\ v^{w,z} \end{bmatrix} = \begin{bmatrix} v^{s,i} i_s^{w,x} + v^{s,j} j_s^{w,x} + v^{s,k} k_s^{w,x} \\ v^{s,i} i_s^{w,y} + v^{s,j} j_s^{w,y} + v^{s,k} k_s^{w,y} \\ v^{s,i} i_s^{w,z} + v^{s,j} j_s^{w,z} + v^{s,k} k_s^{w,z} \end{bmatrix} = \begin{bmatrix} i_s^{w,x} & j_s^{w,x} & k_s^{w,x} \\ i_s^{w,y} & j_s^{w,y} & k_s^{w,y} \\ i_s^{w,z} & j_s^{w,z} & k_s^{w,z} \end{bmatrix} \begin{bmatrix} v^{s,i} \\ v^{s,j} \\ v^{s,k} \end{bmatrix} \Rightarrow$$

$$\vec{v}^w = R_s^w \vec{v}^s \tag{3.10}$$

The matrix R_s^w is usually referred as the *rotation matrix* (or coordinate transformation matrix) between both coordinate systems. Note that R_s^w represents the transformation *from* the ship coordinate system *to* the world coordinate system. The inverse transformation may be computed by the transposition of the rotation matrix R_s^w:

$$R_w^s = (R_s^w)^{-1} = (R_s^w)^T \tag{3.11}$$

where R_w^s is the rotation matrix from the world coordinate system to the ship coordinate system. Thus:

$$R_w^s = (R_s^w)^T = \begin{bmatrix} i_s^{w,x} & j_s^{w,x} & k_s^{w,x} \\ i_s^{w,y} & j_s^{w,y} & k_s^{w,y} \\ i_s^{w,z} & j_s^{w,z} & k_s^{w,z} \end{bmatrix}^T = \begin{bmatrix} i_s^{w,x} & i_s^{w,y} & i_s^{w,z} \\ j_s^{w,x} & j_s^{w,y} & j_s^{w,z} \\ k_s^{w,x} & k_s^{w,y} & k_s^{w,z} \end{bmatrix} \Rightarrow$$

$$\vec{x}_w^s = i_s^{w,x}\vec{i}_s + j_s^{w,x}\vec{j}_s + k_s^{w,x}\vec{k}_s \tag{3.12}$$

$$\vec{y}_w^s = i_s^{w,y}\vec{i}_s + j_s^{w,y}\vec{j}_s + k_s^{w,y}\vec{k}_s \tag{3.13}$$

$$\vec{z}_w^s = i_s^{w,z}\vec{i}_s + j_s^{w,z}\vec{j}_s + k_s^{w,z}\vec{k} \tag{3.14}$$

Note that:

$$(R_s^w)^T(R_s^w) = \tag{3.15}$$

$$\begin{bmatrix} i_s^{w,x} & i_s^{w,y} & i_s^{w,z} \\ j_s^{w,x} & j_s^{w,y} & j_s^{w,z} \\ k_s^{w,x} & k_s^{w,y} & k_s^{w,z} \end{bmatrix} \begin{bmatrix} i_s^{w,x} & j_s^{w,x} & k_s^{w,x} \\ i_s^{w,y} & j_s^{w,y} & k_s^{w,y} \\ i_s^{w,z} & j_s^{w,z} & k_s^{w,z} \end{bmatrix} = \begin{bmatrix} \vec{i}_s \cdot \vec{i}_s & \vec{i}_s \cdot \vec{j}_s & \vec{i}_s \cdot \vec{k}_s \\ \vec{j}_s \cdot \vec{i}_s & \vec{j}_s \cdot \vec{j}_s & \vec{j}_s \cdot \vec{k}_s \\ \vec{k}_s \cdot \vec{i}_s & \vec{k}_s \cdot \vec{j}_s & \vec{k}_s \cdot \vec{k}_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

$$(R_s^w)(R_s^w)^T = \tag{3.17}$$

$$\begin{bmatrix} i_s^{w,x} & j_s^{w,x} & k_s^{w,x} \\ i_s^{w,y} & j_s^{w,y} & k_s^{w,y} \\ i_s^{w,z} & j_s^{w,z} & k_s^{w,z} \end{bmatrix} \begin{bmatrix} i_s^{w,x} & i_s^{w,y} & i_s^{w,z} \\ j_s^{w,x} & j_s^{w,y} & j_s^{w,z} \\ k_s^{w,x} & k_s^{w,y} & k_s^{w,z} \end{bmatrix} = \begin{bmatrix} \vec{x}_w \cdot \vec{x}_w & \vec{x}_w \cdot \vec{y}_w & \vec{x}_w \cdot \vec{z}_w \\ \vec{y}_w \cdot \vec{x}_w & \vec{y}_w \cdot \vec{y}_w & \vec{y}_w \cdot \vec{z}_w \\ \vec{z}_w \cdot \vec{x}_w & \vec{z}_w \cdot \vec{y}_w & \vec{z}_w \cdot \vec{z}_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

Similarly, for a time-varying rotation matrix, the following equality must hold for all

times:

$$R_s^w(t)^T R_s^w(t) = R_w^s(t) R_s^w(t) = R_s^w(t) R_s^w(t)^T = R_s^w(t) R_w^s(t) = I_{3\times 3} \qquad \forall t$$

where $I_{3\times 3}$ denotes an identity matrix with size 3×3. There are useful forms for R_s^w as a function of pertinent angles. A common expression defines three independent rotations around each axis. Let $c(.)$ and $s(.)$ denotes the trigonometric functions cos(.) and sin(.):

$$\begin{aligned} R_s^w(\phi,\theta,\psi)^{XYZ} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c(\psi)c(\theta) & -c(\theta)s(\psi) & s(\theta) \\ c(\psi)s(\theta)s(\phi) + c(\phi)s(\psi) & c(\psi)c(\phi) - s(\psi)s(\theta)s(\phi) & -c(\theta)s(\phi) \\ -c(\psi)c(\phi)s(\theta) + s(\psi)s(\phi) & c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) & c(\theta)c(\phi) \end{bmatrix} \end{aligned} \tag{3.19}$$

Alternatively:

$$\begin{aligned} R_s^w(\phi,\theta,\psi)^{ZYX} &= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \\ &= \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - c(\phi)s(\psi) & c(\psi)c(\phi)s(\theta) + s(\psi)s(\phi) \\ c(\theta)s(\psi) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \end{aligned} \tag{3.20}$$

Another form for the rotation matrix is referred as *Rodrigues formula*. It is related with the rotation of an angle ϕ^R around an unitary axis $\vec{u} = [u_x, u_y, u_z]^T$:

$$R_s^w(\phi,\vec{u})^R = c(\phi) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + s(\phi) \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{3.21}$$

$$+ \Big(1 - c(\phi)\Big) \begin{bmatrix} u_x u_x & u_x u_y & u_x u_z \\ u_y u_x & u_y u_y & u_y u_z \\ u_z u_x & u_z u_y & u_z u_z \end{bmatrix} \tag{3.22}$$

Moreover, it is possible to use *unitary quaternions* for the representation of rotations. A quaternion may be interpreted as an extension of the imaginary numbers. In addition to

the traditional imaginary axis, there are two more axes. Thus, a quaternion is represented with 4 parameters:

$$\vec{q} = q_w\vec{1}^Q + q_x\vec{i}^Q + q_y\vec{j}^Q + q_z\vec{k}^Q \tag{3.23}$$

where $\vec{1}^Q, \vec{i}^Q, \vec{j}^Q, \vec{k}^Q$ represents the corresponding axes of the quaternion representation. The rotation matrix associated with an unitary quaternion $\vec{q} = [q_w, q_x, q_y, q_z]^T$ can be computed as:

$$R_s^w(\vec{q})^{QT} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_xq_y - q_wq_z) & 2(q_xq_z + q_wq_y) \\ 2(q_xq_y + q_wq_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_yq_z - q_wq_x) \\ 2(q_xq_z - q_wq_y) & 2(q_yq_z + q_wq_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{3.24}$$

A rotation of an angle ϕ around an unitary axis $\vec{u} = [u_x, u_y, u_z]^T$ can be expressed with an unitary quaternion as:

$$\vec{q} = \cos\left(\frac{\phi}{2}\right)\vec{1}^Q + u_x\sin\left(\frac{\phi}{2}\right)\vec{i}^Q + u_y\sin\left(\frac{\phi}{2}\right)\vec{j}^Q + u_z\sin\left(\frac{\phi}{2}\right)\vec{k}^Q = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right) \\ u_x\sin\left(\frac{\phi}{2}\right) \\ u_y\sin\left(\frac{\phi}{2}\right) \\ u_z\sin\left(\frac{\phi}{2}\right) \end{bmatrix} \tag{3.25}$$

Note that such quaternion is indeed unitary:

$$\begin{aligned} \left|\left|\ \vec{q}\ \right|\right|^2 &= \cos^2\left(\frac{\phi}{2}\right) + u_x^2\sin^2\left(\frac{\phi}{2}\right) + u_y^2\sin^2\left(\frac{\phi}{2}\right) + u_z^2\sin^2\left(\frac{\phi}{2}\right) \\ &= \cos^2\left(\frac{\phi}{2}\right) + \left(u_x^2 + u_y^2 + u_z^2\right)\sin^2\left(\frac{\phi}{2}\right) = \cos^2\left(\frac{\phi}{2}\right) + \sin^2\left(\frac{\phi}{2}\right) = 1 \end{aligned} \tag{3.26}$$

Consider an arbitrary point Q and let p_Q be its corresponding position. It is possible to define a three-dimensional vectorial representation for the point with a vector $Q - O_w$ from the origin O_w of the world coordinate system to the respective point Q. Let p_{wQ}^w be the position of Q in the world coordinate system:

$$p_{wQ}^w = (Q - O_w) = p_{wQ}^{w,x}\vec{x}_w + p_{wQ}^{w,y}\vec{y}_w + p_{wQ}^{w,z}\vec{z}_w = \begin{bmatrix} p_{wQ}^{w,x} & p_{wQ}^{w,y} & p_{wQ}^{w,z} \end{bmatrix}^T \tag{3.27}$$

Similarly, let p_{ws}^w be the position of origin O_s from the ship coordinate system described in the world coordinate system:

$$p_{ws}^w = (O_s - O_w) = p_{ws}^{w,x}\vec{x}_w + p_{ws}^{w,y}\vec{y}_w + p_{ws}^{w,z}\vec{z}_w = \begin{bmatrix} p_{ws}^{w,x} & p_{ws}^{w,y} & p_{ws}^{w,z} \end{bmatrix}^T \tag{3.28}$$

Note that the position p_Q can be represented with coordinates in the ship coordinate system:

$$
\begin{aligned}
p^w_{wQ} - p^w_{ws} &= (Q - O_w) - (O_s - O_w) = (Q - O_s) \Rightarrow \\
(Q - O_s) &= \left(p^{w,x}_{wQ} - p^{w,x}_{ws}\right)\vec{x}_w + \left(p^{w,y}_{wQ} - p^{w,y}_{ws}\right)\vec{y}_w + \left(p^{w,z}_{wQ} - p^{w,z}_{ws}\right)\vec{z}_w \Rightarrow \\
(Q - O_s) &= \left(p^{w,x}_{wQ} - p^{w,x}_{ws}\right)\left(i^{w,x}_s\vec{i}_s + j^{w,x}_s\vec{j}_s + k^{w,x}_s\vec{k}_s\right) + \\
&+ \left(p^{w,y}_{wQ} - p^{w,y}_{ws}\right)\left(i^{w,y}_s\vec{i}_s + j^{w,y}_s\vec{j}_s + k^{w,y}_s\vec{k}_s\right) + \\
&+ \left(p^{w,z}_{wQ} - p^{w,z}_{ws}\right)\left(i^{w,z}_s\vec{i}_s + j^{w,z}_s\vec{j}_s + k^{w,z}_s\vec{k}_s\right)
\end{aligned}
$$

Let p^s_{sQ} be the position of point Q in the ship coordinate system:

$$
p^s_{sQ} = p^{s,i}_{sQ}\vec{i}_s + p^{s,j}_{sQ}\vec{j}_s + p^{s,k}_{sQ}\vec{k}_s \tag{3.29}
$$

$$
\begin{aligned}
p^s_{sQ} =& \left((p^{w,x}_{wQ} - p^{w,x}_{ws})i^{w,x}_s + (p^{w,y}_{wQ} - p^{w,y}_{ws})i^{w,y}_s + (p^{w,z}_{wQ} - p^{w,z}_{ws})i^{w,z}_s\right)\vec{i}_s + \\
&+ \left((p^{w,x}_{wQ} - p^{w,x}_{ws})j^{w,x}_s + (p^{w,y}_{wQ} - p^{w,y}_{ws})j^{w,y}_s + (p^{w,z}_{wQ} - p^{w,z}_{ws})j^{w,z}_s\right)\vec{j}_s + \\
&+ \left((p^{w,x}_{wQ} - p^{w,x}_{ws})k^{w,x}_s + (p^{w,y}_{wQ} - p^{w,y}_{ws})k^{w,y}_s + (p^{w,z}_{wQ} - p^{w,z}_{ws})k^{w,z}_s\right)\vec{k}_s
\end{aligned} \tag{3.30}
$$

Note that it is possible to describe Equation 3.30 in a matricial form as:

$$
\begin{bmatrix} p^{s,i}_{sQ} \\ p^{s,j}_{sQ} \\ p^{s,k}_{sQ} \end{bmatrix} = \begin{bmatrix} (p^{w,x}_{wQ} - p^{w,x}_{ws})i^{w,x}_s + (p^{w,y}_{wQ} - p^{w,y}_{ws})i^{w,y}_s + (p^{w,z}_{wQ} - p^{w,z}_{ws})i^{w,z}_s \\ (p^{w,x}_{wQ} - p^{w,x}_{ws})j^{w,x}_s + (p^{w,y}_{wQ} - p^{w,y}_{ws})j^{w,y}_s + (p^{w,z}_{wQ} - p^{w,z}_{ws})j^{w,z}_s \\ (p^{w,x}_{wQ} - p^{w,x}_{ws})k^{w,x}_s + (p^{w,y}_{wQ} - p^{w,y}_{ws})k^{w,y}_s + (p^{w,z}_{wQ} - p^{w,z}_{ws})k^{w,z}_s \end{bmatrix}
$$

$$
\begin{bmatrix} p^{s,i}_{sQ} \\ p^{s,j}_{sQ} \\ p^{s,k}_{sQ} \end{bmatrix} = \begin{bmatrix} i^{w,x}_s & i^{w,y}_s & i^{w,z}_s \\ j^{w,x}_s & j^{w,y}_s & j^{w,z}_s \\ k^{w,x}_s & k^{w,y}_s & k^{w,z}_s \end{bmatrix} \begin{bmatrix} p^{w,x}_{wQ} - p^{w,x}_{ws} \\ p^{w,y}_{wQ} - p^{w,y}_{ws} \\ p^{w,z}_{wQ} - p^{w,z}_{ws} \end{bmatrix}
$$

Alternatively:

$$
\begin{bmatrix} p^{s,i}_{sQ} \\ p^{s,j}_{sQ} \\ p^{s,k}_{sQ} \end{bmatrix} = \begin{bmatrix} i^{w,x}_s & i^{w,y}_s & i^{w,z}_s & -i^{w,x}_s p^{w,x}_{ws} - i^{w,y}_s p^{w,y}_{ws} - i^{w,z}_s p^{w,z}_{ws} \\ j^{w,x}_s & j^{w,y}_s & j^{w,z}_s & -j^{w,x}_s p^{w,x}_{ws} - j^{w,y}_s p^{w,y}_{ws} - j^{w,z}_s p^{w,z}_{ws} \\ k^{w,x}_s & k^{w,y}_s & k^{w,z}_s & -k^{w,x}_s p^{w,x}_{ws} - k^{w,y}_s p^{w,y}_{ws} - k^{w,z}_s p^{w,z}_{ws} \end{bmatrix} \begin{bmatrix} p^{w,x}_{wQ} \\ p^{w,y}_{wQ} \\ p^{w,z}_{wQ} \\ 1 \end{bmatrix} \Rightarrow
$$

$$
p^s_{sQ} = \begin{bmatrix} R^s_w & -R^s_w p^w_{ws} \end{bmatrix} \begin{bmatrix} p^w_{wQ} \\ 1 \end{bmatrix} \tag{3.31}
$$

The inverse transformation from the ship coordinate system to the world coordinate

system may be also described in a matricial form:

$$p_{wQ}^{w} = \begin{bmatrix} R_{s}^{w} & p_{ws}^{w} \end{bmatrix} \begin{bmatrix} p_{sQ}^{s} \\ 1 \end{bmatrix} \tag{3.32}$$

Note that it is possible to express relations from Equation 3.31 and 3.32 in a 4×4 matricial form:

$$\begin{bmatrix} p_{wQ}^{w} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{s}^{w} & p_{ws}^{w} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{sQ}^{s} \\ 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} p_{sQ}^{s} \\ 1 \end{bmatrix} = \begin{bmatrix} R_{w}^{s} & -R_{w}^{s} p_{ws}^{w} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{wQ}^{w} \\ 1 \end{bmatrix}$$
$$\begin{bmatrix} p_{wQ}^{w} \\ 1 \end{bmatrix} = T_{s}^{w} \begin{bmatrix} p_{sQ}^{s} \\ 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} p_{sQ}^{s} \\ 1 \end{bmatrix} = T_{w}^{s} \begin{bmatrix} p_{wQ}^{w} \\ 1 \end{bmatrix} \tag{3.33}$$

Matrices T_{w}^{s} and T_{s}^{w} from Equation 3.33 are commonly referred as *transformation matrices* between coordinate systems. Similarly to the notation used for describing rotation matrix, T_{w}^{s} is the transformation from the world coordinate system to the ship coordinate system and T_{s}^{w} is the transformation from the ship coordinate system to the world coordinate system.

$$T_{s}^{w} = \begin{bmatrix} R_{s}^{w} & p_{ws}^{w} \\ 0 & 1 \end{bmatrix} \Leftrightarrow T_{w}^{s} = \begin{bmatrix} R_{w}^{s} & -R_{w}^{s} p_{ws}^{w} \\ 0 & 1 \end{bmatrix} \tag{3.34}$$

Note that:

$$T_{w}^{s} T_{s}^{w} = \begin{bmatrix} R_{w}^{s} R_{s}^{w} + 0 & R_{w}^{s} p_{ws}^{w} - R_{w}^{s} p_{ws}^{w} \\ 0 + 0 & 0 + 1 \end{bmatrix} = \begin{bmatrix} I_{3\times3} & 0 \\ 0 & 1 \end{bmatrix} = I_{4\times4} \tag{3.35}$$

$$T_{s}^{w} T_{w}^{s} = \begin{bmatrix} R_{s}^{w} R_{w}^{s} + 0 & -R_{s}^{w} R_{w}^{s} p_{ws}^{w} + p_{ws}^{w} \\ 0 + 0 & 0 + 1 \end{bmatrix} = \begin{bmatrix} I_{3\times3} & 0 \\ 0 & 1 \end{bmatrix} = I_{4\times4} \tag{3.36}$$

The following sections from this chapter presents foundations for a systematic research on navigation as presented in [TITTERTON; WESTON, 2004] and [GROVES, 2008].

3.1 Earth Modeling

There are three customary reference frames in the context of navigation in the vicinity of the earth [TITTERTON; WESTON, 2004]. The *inertial* (index i) and *earth* (index e) frames have the same origin $O_i = O_e$ at the center of the earth. Axes $\vec{x}_e$ and $\vec{y}_e$ from the *earth* frame are fixed with respect to the earth. These axes rotates with respect to the

inertial frame at a rate ω_{ie} about the axis $\vec{z}_i$ and $\vec{z}_e$. Axes $\vec{z}_i$ and $\vec{z}_e$ from *inertial* and *earth* frames are both parallel to the earth's polar axis (which is assumed to be constant).

Let t_0 be the time instant when both frames are aligned. The rotation matrix relating the earth and the inertial frame may be expressed as a function of time and ω_{ie}:

$$R_i^e(t) = \begin{bmatrix} \cos(\omega_{ie}(t-t_0)) & \sin(\omega_{ie}(t-t_0)) & 0 \\ -\sin(\omega_{ie}(t-t_0)) & \cos(\omega_{ie}(t-t_0)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.37}$$

$$R_e^i(t) = \begin{bmatrix} \cos(\omega_{ie}(t-t_0)) & -\sin(\omega_{ie}(t-t_0)) & 0 \\ \sin(\omega_{ie}(t-t_0)) & \cos(\omega_{ie}(t-t_0)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.38}$$

$$\omega_{ie} \approx 7.292115 \cdot 10^{-5}\ rad/s \tag{3.39}$$

The *navigation* (index n) frame has its origin in an arbitrary point Q. Axis $\vec{x}_n$ of the *navigation* frame is parallel to the north, axis $\vec{y}_n$ is parallel to the east, and axis $\vec{z}_n$ is parallel to the down direction. At a first approximation, the gravitational acceleration is aligned with the down direction of the *navigation* frame.

The surface of the earth is customary described with a global ellipsoid model. A widely adopted standard for navigation systems is referred as the *World Geodetic System* 1984 (*WGS 84*). This standard defines a global ellipsoid in terms of the equatorial radius R_0^{WGS} and the flattening f^{WGS} of the earth [GROVES, 2008]:

$$R_0^{WGS} = 6378137.0m \tag{3.40}$$

$$f^{WGS} = \frac{1}{298.257223563} \tag{3.41}$$

The polar radius R_p^{WGS} and the eccentricity e^{WGS} may be computed as a function of R_0^{WGS} and f^{WGS} [TITTERTON; WESTON, 2004]:

$$R_p^{WGS} = R_0^{WGS}(1 - f^{WGS}) = 6356752.3142m \tag{3.42}$$

$$e^{WGS} = \sqrt{f^{WGS}(2 - f^{WGS})} = 0.0818191908426 \tag{3.43}$$

The position p_Q of a point Q is defined with respect to the reference ellipsoid in terms of its *latitude* L_q, *longitude* λ_q and *height* h_q. This representation is usually referred as the *Curvilinear* position of point p. Let $[p_{eQ}^{e,x}, p_{eQ}^{e,y}, p_{eQ}^{e,z}]$ be the *Cartesian* position of point Q with respect to the *earth* frame. An expression relating the *curvilinear* position with

the *cartesian* position may be expressed as [GROVES, 2008]:

$$R_E(L_q) = \frac{R_0^{WGS}}{\sqrt{1-(e^{WGS})^2\sin^2(L_q)}} \tag{3.44}$$

$$\begin{bmatrix} p_{eQ}^{e,x} \\ p_{eQ}^{e,y} \\ p_{eQ}^{e,z} \end{bmatrix} = \begin{bmatrix} (R_E(L_q)+h_q)\cos(L_q)\cos(\lambda_q) \\ (R_E(L_q)+h_q)\cos(L_q)\sin(\lambda_q) \\ [(1-(e^{WGS})^2)R_E(L_q)+h_q]\sin(L_q) \end{bmatrix} \tag{3.45}$$

Alternatively, the *curvilinear* coordinates $[L_q, \lambda_q, h_q]$ may be computed from *cartesian* coordinates $[p_{eQ}^{e,x}, p_{eQ}^{e,y}, p_{eQ}^{e,z}]$:

$$L_q = \arctan\left[\frac{p_{eQ}^{e,z}(R_E(L_q)+h_q)}{\sqrt{(p_{eQ}^{e,x})^2+(p_{eQ}^{e,y})^2}[(1-(e^{WGS})^2)R_E(L_q)+h_q]}\right] \tag{3.46}$$

$$\lambda_q = \arctan\left[\frac{p_{eQ}^{e,y}}{p_{eQ}^{e,x}}\right] \tag{3.47}$$

$$h_q = \frac{\sqrt{(p_{eQ}^{e,x})^2+(p_{eQ}^{e,y})^2}}{\cos(L_q)} - R_E(L_q) \tag{3.48}$$

Note that Equations 3.46 and 3.48 must be solved iteratively in order to determine the latitude L_q and the height h_q. The latitude L_q may be initialized with the following simplification:

$$L_q \approx \arctan\left[\frac{p_{eQ}^{e,z}}{\sqrt{(p_{eQ}^{e,x})^2+(p_{eQ}^{e,y})^2}}\right] \tag{3.49}$$

The rotation matrix relating the earth and the navigation frame may be expressed as a function of the *curvilinear* coordinates:

$$R_e^n(L_q,\lambda_q,h_q) = \begin{bmatrix} -\sin(L_q)\cos(\lambda_q) & -\sin(L_q)\sin(\lambda_q) & \cos(L_q) \\ -\sin(\lambda_q) & \cos(\lambda_q) & 0 \\ -\cos(L_q)\cos(\lambda_q) & -\cos(L_q)\sin(\lambda_q) & -\sin(L_q) \end{bmatrix} \tag{3.50}$$

$$R_n^e(L_q,\lambda_q,h_q) = \begin{bmatrix} -\sin(L_q)\cos(\lambda_q) & -\sin(\lambda_q) & -\cos(L_q)\cos(\lambda_q) \\ -\sin(L_q)\sin(\lambda_q) & \cos(\lambda_q) & -\cos(L_q)\sin(\lambda_q) \\ \cos(L_q) & 0 & -\sin(L_q) \end{bmatrix} \tag{3.51}$$

The WGS-84 datum provides a simple model of gravity as function of the *curvilinear*

position. The acceleration due to gravity is expressed as a function of latitude:

$$g_0(L_q) \approx 9.7803253359 \frac{1 + 0.001931853 \sin^2(L_q)}{\sqrt{1 - (e^{WGS})^2 \sin^2(L_q)}} \quad m/s^2 \tag{3.52}$$

The gravity direction is defined with the down unit vector of the local navigation frame (index n). Let $\vec{g}^n_Q$ be the gravity vector at point Q expressed in the navigation frame and let $\vec{g}^e_Q$ be the same gravity expressed in the earth frame:

$$\vec{g}^n_Q(L_q) = \begin{bmatrix} 0 \\ 0 \\ g_0(L_q) \end{bmatrix} \Rightarrow \tag{3.53}$$

$$\vec{g}^e_Q(L_q, \lambda_q, h_q) = R^e_n(L_q, \lambda_q, h_q)\vec{g}^n_Q(L_q) = \begin{bmatrix} -\cos(L_q)\cos(\lambda_q)g_0(L_q) \\ -\cos(L_q)\sin(\lambda_q)g_0(L_q) \\ -\sin(L_q)g_0(L_q) \end{bmatrix} \tag{3.54}$$

Consider a world reference coordinate system S_w. Assume that the origin O_w is described by its curvilinear position $[L_w, \lambda_w, h_w]$ with respect to the earth frame. The cartesian position p^e_{ew} of the origin O_w with respect to the earth frame is possible to be computed from Equation 3.45.

$$p^e_{ew} = \begin{bmatrix} p^{e,x}_{ew} \\ p^{e,y}_{ew} \\ p^{e,z}_{ew} \end{bmatrix} = \begin{bmatrix} (R_E(L_w) + h_w)\cos(L_w)\cos(\lambda_w) \\ (R_E(L_w) + h_w)\cos(L_w)\sin(\lambda_w) \\ [(1 - (e^{WGS})^2)R_E(L_w) + h_w]\sin(L_w) \end{bmatrix} \tag{3.55}$$

Further, assume that axes $[\vec{x}_w, \vec{y}_w, \vec{z}_w]$ from S_w are described by a rotation matrix R^n_w between world and navigation frame:

$$R^n_w = \begin{bmatrix} \vec{x}^n_w & \vec{y}^n_w & \vec{z}^n_w \end{bmatrix} = \begin{bmatrix} x^{n,x}_w & y^{n,x}_w & z^{n,x}_w \\ x^{n,y}_w & y^{n,y}_w & z^{n,y}_w \\ x^{n,z}_w & y^{n,z}_w & z^{n,z}_w \end{bmatrix} \tag{3.56}$$

Note that it is possible to compute the rotation matrix R^e_w between the world reference frame and the earth frame with Equation 3.51:

$$R^e_w = R^e_n(L_w, \lambda_w, h_w)R^n_w \tag{3.57}$$

$$R^e_w = \begin{bmatrix} -\sin(L_w)\cos(\lambda_w) & -\sin(\lambda_w) & -\cos(L_w)\cos(\lambda_w) \\ -\sin(L_w)\sin(\lambda_w) & \cos(\lambda_w) & -\cos(L_w)\sin(\lambda_w) \\ \cos(L_w) & 0 & -\sin(L_w) \end{bmatrix} \begin{bmatrix} x^{n,x}_w & y^{n,x}_w & z^{n,x}_w \\ x^{n,y}_w & y^{n,y}_w & z^{n,y}_w \\ x^{n,z}_w & y^{n,z}_w & z^{n,z}_w \end{bmatrix} \tag{3.58}$$

Therefore, let p^w_{wQ} be the position of an arbitrary point Q in the world reference frame. The corresponding position p^e_{eQ} of point Q in the earth frame may be expressed by:

$$p^e_{eQ} = p^e_{ew} + R^e_w p^w_{wQ} \tag{3.59}$$

Conversely, the position p^w_{wQ} of point Q in the world reference frame may be expressed as a function of its corresponding position p^e_{eQ} in the earth frame:

$$\begin{aligned} R^w_e \left[p^e_{eQ} \right] &= R^w_e \left[p^e_{ew} + R^e_w p^w_{wQ} \right] \Rightarrow \\ R^w_e p^e_{eQ} &= R^w_e p^e_{ew} + p^w_{wQ} \Rightarrow \\ R^w_e p^e_{eQ} - R^w_e p^e_{ew} &= p^w_{wQ} \Rightarrow \\ p^w_{wQ} &= R^w_e (p^e_{eQ} - p^e_{ew}) = (R^e_w)^T (p^e_{eQ} - p^e_{ew}) \end{aligned} \tag{3.60}$$

3.2 Kinematics

In navigation, it is customary to express linear and angular motion of one coordinate frame with respect to another. Generally, most kinematic quantities involve three coordinate frames: the *object*-frame (α); the *reference*-frame (β) and the *resolving*-frame (γ). Henceforth, a kinematic property K from frame α with respect to a frame β is expressed in frame γ as $K^\gamma_{\beta\alpha}$. For example, the cartesian position of frame α with respect to β expressed in frame γ is given by $p^\gamma_{\beta\alpha}$. The kinematic *velocity* of a frame α with respect to a frame β is defined as the time derivative of its position resolved about the same *resolving* frame β:

$$v^\beta_{\beta\alpha} = \dot{p}^\beta_{\beta\alpha} \tag{3.61}$$

Similarly, the kinematic *acceleration* from frame α with respect to frame β is defined as the time derivative of its velocity resolved about the same *resolving* frame β:

$$a^\beta_{\beta\alpha} = \dot{v}^\beta_{\beta\alpha} = \ddot{p}^\beta_{\beta\alpha} \tag{3.62}$$

The rotation matrix from frame α to frame β is expressed by R^β_α (or C^β_α). The inverse rotation matrix from frame β to frame α is expressed by R^α_β (or C^α_β). The *angular rate* vector is the rate of rotation of the frame α with respect to the frame β, resolved about a third frame γ. It is commonly represented as a three-dimensional vector $\omega^\gamma_{\beta\alpha}$ or by a

skew-symmetric matrix representation $\Omega^{\gamma}_{\beta\alpha}$:

$$\omega^{\gamma}_{\beta\alpha} = \begin{bmatrix} \omega^{\gamma,x}_{\beta\alpha} \\ \omega^{\gamma,y}_{\beta\alpha} \\ \omega^{\gamma,z}_{\beta\alpha} \end{bmatrix} \Leftrightarrow \left[\omega^{\gamma}_{\beta\alpha}\hat{\times}\right] = \Omega^{\gamma}_{\beta\alpha} = \begin{bmatrix} 0 & -\omega^{\gamma,z}_{\beta\alpha} & \omega^{\gamma,y}_{\beta\alpha} \\ \omega^{\gamma,z}_{\beta\alpha} & 0 & -\omega^{\gamma,x}_{\beta\alpha} \\ -\omega^{\gamma,y}_{\beta\alpha} & \omega^{\gamma,x}_{\beta\alpha} & 0 \end{bmatrix} \tag{3.63}$$

The skew-symmetric matrix representation of angular rate $\Omega^{\gamma}_{\beta\alpha}$ may be transformed to another frame δ with the following expression:

$$\Omega^{\delta}_{\beta\alpha} = R^{\delta}_{\gamma}\Omega^{\gamma}_{\beta\alpha}R^{\gamma}_{\delta} \tag{3.64}$$

The time derivatives of a rotation matrix R^{α}_{β} may be expressed as a function of the angular rate between both frames:

$$\dot{R}^{\alpha}_{\beta} = -R^{\alpha}_{\beta}\Omega^{\beta}_{\beta\alpha} = -\Omega^{\alpha}_{\beta\alpha}R^{\alpha}_{\beta} = R^{\alpha}_{\beta}\Omega^{\beta}_{\alpha\beta} = \Omega^{\alpha}_{\alpha\beta}R^{\alpha}_{\beta} \tag{3.65}$$

The kinematic *velocity* of a frame α with respect to a frame β may be expressed in another frame γ as $v^{\gamma}_{\beta\alpha}$.

$$v^{\gamma}_{\beta\alpha} = R^{\gamma}_{\beta}\dot{p}^{\beta}_{\beta\alpha} \tag{3.66}$$

Note that $v^{\gamma}_{\beta\alpha}$ is not equal to the time derivative of $p^{\gamma}_{\beta\alpha}$ unless there is no angular motion between γ and β.

$$\dot{p}^{\gamma}_{\beta\alpha} = \frac{d}{dt}\left(R^{\gamma}_{\beta}p^{\beta}_{\beta\alpha}\right) = \dot{R}^{\gamma}_{\beta}p^{\beta}_{\beta\alpha} + R^{\gamma}_{\beta}\dot{p}^{\beta}_{\beta\alpha} = \dot{R}^{\gamma}_{\beta}p^{\beta}_{\beta\alpha} + v^{\gamma}_{\beta\alpha} \tag{3.67}$$

The kinematic *acceleration* may be expressed in another frame γ as $a^{\gamma}_{\beta\alpha}$ with the appropriate coordinate transformation matrix:

$$a^{\gamma}_{\beta\alpha} = R^{\gamma}_{\beta}a^{\beta}_{\beta\alpha} = R^{\gamma}_{\beta}\dot{v}^{\beta}_{\beta\alpha} = R^{\gamma}_{\beta}\ddot{p}^{\beta}_{\beta\alpha} \tag{3.68}$$

Note that the acceleration $a^{\gamma}_{\beta\alpha}$ is not the same as the time derivative of $v^{\gamma}_{\beta\alpha}$ or the second time derivative of $p^{\gamma}_{\beta\alpha}$ unless there is no rotation between β and γ.

$$\frac{d}{dt}\left(v^{\gamma}_{\beta\alpha}\right) = \frac{d}{dt}\left(R^{\gamma}_{\beta}\dot{p}^{\beta}_{\beta\alpha}\right) \Rightarrow \dot{v}^{\gamma}_{\beta\alpha} = \dot{R}^{\gamma}_{\beta}\dot{p}^{\beta}_{\beta\alpha} + R^{\gamma}_{\beta}\ddot{p}^{\beta}_{\beta\alpha} \Rightarrow \dot{v}^{\gamma}_{\beta\alpha} = \dot{R}^{\gamma}_{\beta}v^{\beta}_{\beta\alpha} + a^{\gamma}_{\beta\alpha} \tag{3.69}$$

$$\frac{d}{dt}\left(\dot{p}^{\gamma}_{\beta\alpha}\right) = \frac{d}{dt}\left(\dot{R}^{\gamma}_{\beta}p^{\beta}_{\beta\alpha} + v^{\gamma}_{\beta\alpha}\right) \Rightarrow \ddot{p}^{\gamma}_{\beta\alpha} = \ddot{R}^{\gamma}_{\beta}p^{\beta}_{\beta\alpha} + \dot{R}^{\gamma}_{\beta}\dot{p}^{\beta}_{\beta\alpha} + \dot{v}^{\gamma}_{\beta\alpha} \Rightarrow$$

$$\Rightarrow \ddot{p}^{\gamma}_{\beta\alpha} = \ddot{R}^{\gamma}_{\beta}p^{\beta}_{\beta\alpha} + \dot{R}^{\gamma}_{\beta}v^{\beta}_{\beta\alpha} + (\dot{R}^{\gamma}_{\beta}v^{\beta}_{\beta\alpha} + a^{\gamma}_{\beta\alpha}) = \ddot{R}^{\gamma}_{\beta}p^{\beta}_{\beta\alpha} + 2\dot{R}^{\gamma}_{\beta}\dot{p}^{\beta}_{\beta\alpha} + a^{\gamma}_{\beta\alpha} \tag{3.70}$$

Note that:

$$\dot{p}^{\gamma}_{\beta\alpha} = \dot{R}^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} + R^{\gamma}_{\beta} \dot{p}^{\beta}_{\beta\alpha} \Rightarrow R^{\gamma}_{\beta} \dot{p}^{\beta}_{\beta\alpha} = \dot{p}^{\gamma}_{\beta\alpha} - \dot{R}^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} \tag{3.71}$$

From Equation 3.65:

$$\dot{R}^{\gamma}_{\beta} \dot{p}^{\beta}_{\beta\alpha} = -\Omega^{\gamma}_{\beta\gamma} R^{\gamma}_{\beta} \dot{p}^{\beta}_{\beta\alpha} = -\Omega^{\gamma}_{\beta\gamma} \Big(\dot{p}^{\gamma}_{\beta\alpha} - \dot{R}^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} \Big) = -\Omega^{\gamma}_{\beta\gamma} \Big(\dot{p}^{\gamma}_{\beta\alpha} + \Omega^{\gamma}_{\beta\gamma} R^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} \Big) \tag{3.72}$$

$$\ddot{R}^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} = -\dot{\Omega}^{\gamma}_{\beta\gamma} R^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} - \Omega^{\gamma}_{\beta\gamma} \dot{R}^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} = -\dot{\Omega}^{\gamma}_{\beta\gamma} R^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} + \Omega^{\gamma}_{\beta\gamma} \Omega^{\gamma}_{\beta\gamma} R^{\gamma}_{\beta} p^{\beta}_{\beta\alpha} \tag{3.73}$$

Therefore:

$$\begin{aligned} \ddot{p}^{\gamma}_{\beta\alpha} =& \Big(-\dot{\Omega}^{\gamma}_{\beta\gamma} p^{\gamma}_{\beta\alpha} + \Omega^{\gamma}_{\beta\gamma} \Omega^{\gamma}_{\beta\gamma} p^{\gamma}_{\beta\alpha} \Big) + 2 \Big(-\Omega^{\gamma}_{\beta\gamma} \Big(\dot{p}^{\gamma}_{\beta\alpha} + \Omega^{\gamma}_{\beta\gamma} p^{\gamma}_{\beta\alpha} \Big) \Big) + a^{\gamma}_{\beta\alpha} \\ \ddot{p}^{\gamma}_{\beta\alpha} =& -\Big(\Omega^{\gamma}_{\beta\gamma} \Omega^{\gamma}_{\beta\gamma} + \dot{\Omega}^{\gamma}_{\beta\gamma} \Big) p^{\gamma}_{\beta\alpha} - 2\Omega^{\gamma}_{\beta\gamma} \dot{p}^{\gamma}_{\beta\alpha} + a^{\gamma}_{\beta\alpha} \end{aligned} \tag{3.74}$$

The set of kinematic *acceleration* a^{e}_{is}, *velocity* v^{e}_{es} and *position* p^{e}_{es} determines the ship position in the earth frame throughout its navigation. Note that as the center of the earth frame is coincident with the center of the inertial frame, the position of a point with respect to the earth frame is the same as the position with respect to the inertial frame:

$$p^{i}_{i\alpha} = p^{i}_{e\alpha} \quad \Leftrightarrow \quad p^{e}_{i\alpha} = p^{e}_{e\alpha} \tag{3.75}$$

As the inertial frame does not rotate, the velocity of a point with respect to the earth frame is different from the velocity of a point with respect to the inertial frame:

$$v^{i}_{i\alpha} = R^{i}_{e}(v^{e}_{e\alpha} + \Omega^{e}_{ie} p^{e}_{e\alpha}) \Rightarrow R^{e}_{i} v^{i}_{i\alpha} = R^{e}_{i} R^{i}_{e}(v^{e}_{e\alpha} + \Omega^{e}_{ie} p^{e}_{e\alpha}) \Rightarrow v^{e}_{i\alpha} = v^{e}_{e\alpha} + \Omega^{e}_{ie} p^{e}_{e\alpha} \tag{3.76}$$

Considering Equation 3.74 assuming that the rotation of the earth ω^{e}_{ie} is constant:

$$\ddot{p}^{e}_{is} = -\Big(\Omega^{e}_{ie} \Omega^{e}_{ie} \Big) p^{e}_{is} - 2\Omega^{e}_{ie} \dot{p}^{e}_{is} + a^{e}_{is} \tag{3.77}$$

Expanding the term $\dot{p}^{e}_{is}$ from Equation 3.77 in terms of the ship state $[p^{e}_{es}, v^{e}_{es}, a^{e}_{is}]$:

$$\dot{p}^{e}_{is} = \frac{d}{dt} \Big[R^{e}_{i} p^{i}_{is} \Big] = \dot{R}^{e}_{i} p^{i}_{is} + R^{e}_{i} v^{i}_{is} = -\Omega^{e}_{ie} R^{e}_{i} p^{i}_{is} + v^{e}_{is} \Rightarrow \tag{3.78}$$

$$\dot{p}^{e}_{is} = -\Omega^{e}_{ie} p^{e}_{is} + v^{e}_{is} = -\Omega^{e}_{ie} p^{e}_{is} + v^{e}_{es} + \Omega^{e}_{ie} p^{e}_{es} = v^{e}_{es} \tag{3.79}$$

Therefore, the ship position in the earth frame may be determined with the integration

of the following expressions:

$$\dot{p}^e_{es} = v^e_{es} \tag{3.80}$$

$$\dot{v}^e_{es} = -\Omega^e_{ie}\Omega^e_{ie}p^e_{es} - 2\Omega^e_{ie}v^e_{es} + a^e_{is} \tag{3.81}$$

Equations 3.80 and 3.81 are commonly referred as *navigation equations*. Furthermore, consider R^e_s as the ship orientation with respect to the earth frame and let ω^s_{es} be the angular rate between the ship frame and the earth frame:

$$\dot{R}^e_s = R^e_s\Omega^s_{es} \tag{3.82}$$

Note that the angular rate between ship and inertial frame is composed by the angular rate between the ship and the earth with the angular rate between the earth and the inertial frame:

$$\omega^s_{is} = \omega^s_{ie} + \omega^s_{es} \Rightarrow \omega^s_{es} = \omega^s_{is} - \omega^s_{ie} \tag{3.83}$$

Therefore:

$$\dot{R}^e_s = R^e_s\Omega^s_{es} = R^e_s\Omega^s_{is} - R^e_s\Omega^s_{ie} = R^e_s\Omega^s_{is} - R^e_s\Big(R^s_e\Omega^e_{ie}R^e_s\Big) \Rightarrow$$

$$\dot{R}^e_s = R^e_s\Omega^s_{is} - \Omega^e_{ie}R^e_s \tag{3.84}$$

where the angular rate Ω^s_{is} may be measured by inertial sensors and Ω^e_{ie} is assumed to be known and constant.

3.3 Maritime Navigational Assistance

The marine environment is known to be unpredictably harsh and characterized by visibility issues related with different atmospheric phenomena. Further, in some cases of maritime navigation such as in offshore scenarios, the lack of landmarks contributes to increase complexity. Due to such complex factors, marine navigation remained an enormous challenge for many centuries. Nowadays, there are typical digital tools for assisting maritime navigation. An example of such equipment is a compass, which outputs the angle between the instrument case and the direction of the true north with respect to the earth. These measurements are extremely important for the maintenance of a desired ship route towards a destination.

A very important type of equipment for navigational assistance is usually referred as

Inertial Navigation Systems (**INS**). These systems are constituted by a navigation processor that integrates inertial measurements of acceleration and angular velocity providing the instantaneous state of the instrument. The integration is typically assisted by filtering algorithms such as the *Kalman filter*, which is presented in Appendix B. There are different INS manufacturers with corresponding different implementation technologies. Note that the precision of the instantaneous state provided by the INS will ultimately depend on the precision of its inertial measurements.

Another important equipment that assists ship operators during navigation is known as *Global Navigation Satellite Systems* (**GNSS**). GNSS are systems that provides estimates for the user position and velocity with respect to the earth. There are three main components in a GNSS architecture: the *space segment*, the *control segment* and the *user segment*. The *space segment* is constituted by satellites, which are collectively known as *constellation*. The *control segment* monitor the *space segment* to track each satellite and ensure their orbits. These satellites broadcasts signals to the *user segment* that incorporates ranging codes and navigation data messages. The ranging codes enable the user equipment to determine the time of broadcast by each satellite, while the navigation data messages includes timing parameters and information about the satellite orbits. Typically, GNSS user equipment are implemented with a *navigator processor* that calculates the instrument position and velocity providing such solutions as outputs.

When multiple equipment are available in a ship, it is possible to integrate all measurements into a integrated solution. An *Integrated Navigation System* consists of an equipment that integrates multiple real-time sensors in order to determine an estimation for the ship state described in a reference coordinate system. There are multiple algorithms for such estimation. Typically, a reference solution is computed from an INS. As there are multiple sources of errors that affects the accuracy of the inertial solution, the predicted reference state is combined with measurements from other auxiliary sources in order to determine an improved estimation for the ship state. Simple models for navigational assistance equipment that are used throughout the current work are presented in Appendix C.

4 CAMERAS

Consider a camera coordinate system S_c (index c) described by axes $[\vec{\chi}_c, \vec{\gamma}_c, \vec{\kappa}_c]$ with origin at a point O_c. Assume that three-dimensional coordinates for the position of O_c are known in the ship coordinate system S_S.

$$p^s_{sc} = O_c - O_s = p^{s,i}_{sc}\vec{i}_s + p^{s,j}_{sc}\vec{j}_s + p^{s,k}_{sc}\vec{k}_s = \begin{bmatrix} p^{s,i}_{sc} & p^{s,j}_{sc} & p^{s,k}_{sc} \end{bmatrix}^T \tag{4.1}$$

Further, consider that each axis of this additional coordinate system is known with respect to the ship coordinate system:

$$\vec{\chi}^s_c = \chi^{s,i}_c\vec{i}_s + \chi^{s,j}_c\vec{j}_s + \chi^{s,k}_c\vec{k}_s = \begin{bmatrix} \chi^{s,i}_c & \chi^{s,j}_c & \chi^{s,k}_c \end{bmatrix}^T \tag{4.2}$$

$$\vec{\gamma}^s_c = \gamma^{s,i}_c\vec{i}_s + \gamma^{s,j}_c\vec{j}_s + \gamma^{s,k}_c\vec{k}_s = \begin{bmatrix} \gamma^{s,i}_c & \gamma^{s,j}_c & \gamma^{s,k}_c \end{bmatrix}^T \tag{4.3}$$

$$\vec{\kappa}^s_c = \kappa^{s,i}_c\vec{i}_s + \kappa^{s,j}_c\vec{j}_s + \kappa^{s,k}_c\vec{k}_s = \begin{bmatrix} \kappa^{s,i}_c & \kappa^{s,j}_c & \kappa^{s,k}_c \end{bmatrix}^T \tag{4.4}$$

Parameters from Equations 4.1, 4.2, 4.3 and 4.4 yields the following transformations T^c_s and T^s_c between the camera frame S_c and the ship frame S_S:

$$R^s_c = \begin{bmatrix} \chi^{s,i}_c & \gamma^{s,i}_c & \kappa^{s,i}_c \\ \chi^{s,j}_c & \gamma^{s,j}_c & \kappa^{s,j}_c \\ \chi^{s,k}_c & \gamma^{s,k}_c & \kappa^{s,k}_c \end{bmatrix} \Rightarrow T^s_c = \begin{bmatrix} R^s_c & p^s_{sc} \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

$$R^c_s = \begin{bmatrix} \chi^{s,i}_c & \chi^{s,j}_c & \chi^{s,k}_c \\ \gamma^{s,i}_c & \gamma^{s,j}_c & \gamma^{s,k}_c \\ \kappa^{s,i}_c & \kappa^{s,j}_c & \kappa^{s,k}_c \end{bmatrix} \Rightarrow T^c_s = \begin{bmatrix} R^c_s & -R^c_s p^s_{sc} \\ 0 & 1 \end{bmatrix} \tag{4.6}$$

Therefore, for an arbitrary point Q with position p^c_{cQ} in the camera frame, the corre-

sponding position p^s_{sQ} in the ship frame can be computed as:

$$p^s_{sQ} = \begin{bmatrix} p^{s,i}_{sQ} \\ p^{s,j}_{sQ} \\ p^{s,k}_{sQ} \end{bmatrix} = \begin{bmatrix} \chi^{s,i}_c & \gamma^{s,i}_c & \kappa^{s,i}_c \\ \chi^{s,j}_c & \gamma^{s,j}_c & \kappa^{s,j}_c \\ \chi^{s,k}_c & \gamma^{s,k}_c & \kappa^{s,k}_c \end{bmatrix} \begin{bmatrix} p^{c,\chi}_{cQ} \\ p^{c,\gamma}_{cQ} \\ p^{c,\kappa}_{cQ} \end{bmatrix} + \begin{bmatrix} p^{s,i}_{sc} \\ p^{s,j}_{sc} \\ p^{s,k}_{sc} \end{bmatrix} \tag{4.7}$$

$$\begin{bmatrix} p^s_{sQ} \\ 1 \end{bmatrix} = \begin{bmatrix} R^s_c & p^s_{sc} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^c_{cQ} \\ 1 \end{bmatrix} \tag{4.8}$$

Conversely:

$$p^c_{cQ} = \begin{bmatrix} p^{c,\chi}_{cQ} \\ p^{c,\gamma}_{cQ} \\ p^{c,\kappa}_{cQ} \end{bmatrix} = \begin{bmatrix} \chi^{s,i}_c & \chi^{s,j}_c & \chi^{s,k}_c \\ \gamma^{s,i}_c & \gamma^{s,j}_c & \gamma^{s,k}_c \\ \kappa^{s,i}_c & \kappa^{s,j}_c & \kappa^{s,k}_c \end{bmatrix} \begin{bmatrix} p^{s,i}_{sQ} - p^{s,i}_{sc} \\ p^{s,j}_{sQ} - p^{s,j}_{sc} \\ p^{s,k}_{sQ} - p^{s,k}_{sc} \end{bmatrix} \tag{4.9}$$

$$\begin{bmatrix} p^c_{cQ} \\ 1 \end{bmatrix} = \begin{bmatrix} R^c_s & -R^c_s p^s_{sc} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p^s_{sQ} \\ 1 \end{bmatrix} \tag{4.10}$$

In addition to the projection of an arbitrary point Q, it is interesting to investigate expressions for the projection of points belonging to particular planes in space. As a plane can be defined by a point and a unitary normal direction vector, it is possible to describe an arbitrary plane Π with a coordinate system transformation.

Consider a *plane* coordinate system S_Π expressed by a set of axes $[\vec{x}_\Pi, \vec{y}_\Pi, \vec{z}_\Pi]$ and an origin O_Π. Assume that the origin O_Π belongs to the plane Π and that axes $[\vec{x}_\Pi, \vec{y}_\Pi]$ from S_Π are aligned with the plane Π. The position $p^c_{c\Pi}$ of O_Π is assumed to be known with respect to the camera frame:

$$p^c_{c\Pi} = O_\Pi - O_c = p^{c,\chi}_{c\Pi}\vec{\chi}_c + p^{c,\gamma}_{c\Pi}\vec{\gamma}_c + p^{c,\kappa}_{c\Pi}\vec{\kappa}_c = \begin{bmatrix} p^{c,\chi}_{c\Pi} & p^{c,\gamma}_{c\Pi} & p^{c,\kappa}_{c\Pi} \end{bmatrix}^T \tag{4.11}$$

Similarly, axes $[\vec{x}_\Pi, \vec{y}_\Pi, \vec{z}_\Pi]$ are assumed to be known with respect to the camera coordinate system:

$$\vec{x}^c_\Pi = x^{c,\chi}_\Pi \vec{\chi}_c + x^{c,\gamma}_\Pi \vec{\gamma}_c + x^{c,\kappa}_\Pi \vec{\kappa}_c \tag{4.12}$$

$$\vec{y}^c_\Pi = y^{c,\chi}_\Pi \vec{\chi}_c + y^{c,\gamma}_\Pi \vec{\gamma}_c + y^{c,\kappa}_\Pi \vec{\kappa}_c \tag{4.13}$$

$$\vec{z}^c_\Pi = z^{c,\chi}_\Pi \vec{\chi}_c + z^{c,\gamma}_\Pi \vec{\gamma}_c + z^{c,\kappa}_\Pi \vec{\kappa}_c \tag{4.14}$$

Thus, the position $p^c_{cP_\pi}$ of an arbitrary point P_π from the plane may be expressed in

the camera frame from a linear combination of $\vec{x}^c_\Pi$ and $\vec{y}^c_\Pi$:

$$p^c_{cP_\pi} = p^c_{c\Pi} + \mu_1 \vec{x}^c_\Pi + \mu_2 \vec{y}^c_\Pi \tag{4.15}$$

where μ_1 and μ_2 are arbitrary real numbers. Note that it is possible to define a transformation matrix T^c_Π that transform points from the plane frame to the camera frame:

$$R^c_\Pi = \begin{bmatrix} x^{c,\chi}_\Pi & y^{c,\chi}_\Pi & z^{c,\chi}_\Pi \\ x^{c,\gamma}_\Pi & y^{c,\gamma}_\Pi & z^{c,\gamma}_\Pi \\ x^{c,\kappa}_\Pi & y^{c,\kappa}_\Pi & z^{c,\kappa}_\Pi \end{bmatrix} \Rightarrow T^c_\Pi = \begin{bmatrix} R^c_\Pi & p^c_{c\Pi} \\ 0 & 1 \end{bmatrix} \Rightarrow \tag{4.16}$$

$$\begin{bmatrix} p^c_{cP_\pi} \\ 1 \end{bmatrix} = \begin{bmatrix} p^{c,\chi}_{cP_\pi} \\ p^{c,\gamma}_{cP_\pi} \\ p^{c,\kappa}_{cP_\pi} \\ 1 \end{bmatrix} = \begin{bmatrix} x^{c,\chi}_\Pi & y^{c,\chi}_\Pi & z^{c,\chi}_\Pi & p^{c,\chi}_{c\Pi} \\ x^{c,\gamma}_\Pi & y^{c,\gamma}_\Pi & z^{c,\gamma}_\Pi & p^{c,\gamma}_{c\Pi} \\ x^{c,\kappa}_\Pi & y^{c,\kappa}_\Pi & z^{c,\kappa}_\Pi & p^{c,\kappa}_{c\Pi} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 0 \\ 1 \end{bmatrix} \tag{4.17}$$

Assuming that other equipment of the ship are able to provide estimates for the state of the ship and surrounding points, it is possible to transform these points to the camera coordinate system with Equations 4.10 and 4.17. Once a point has its position described with respect to the camera frame, it is possible to compute its corresponding projection in the image scene with proper modeling.

The following sections presents camera models for the projection of arbitrary points with known position in a camera coordinate system. The following content is based on the book [SZELISKI, 2010] about computer vision methods and applications; a classical book of geometrical projection [HARTLEY; ZISSERMAN, 2003]; and the documentation of the open-source library OpenCV [BRADSKI; KAEHLER, 2008].

4.1 Pinhole Model

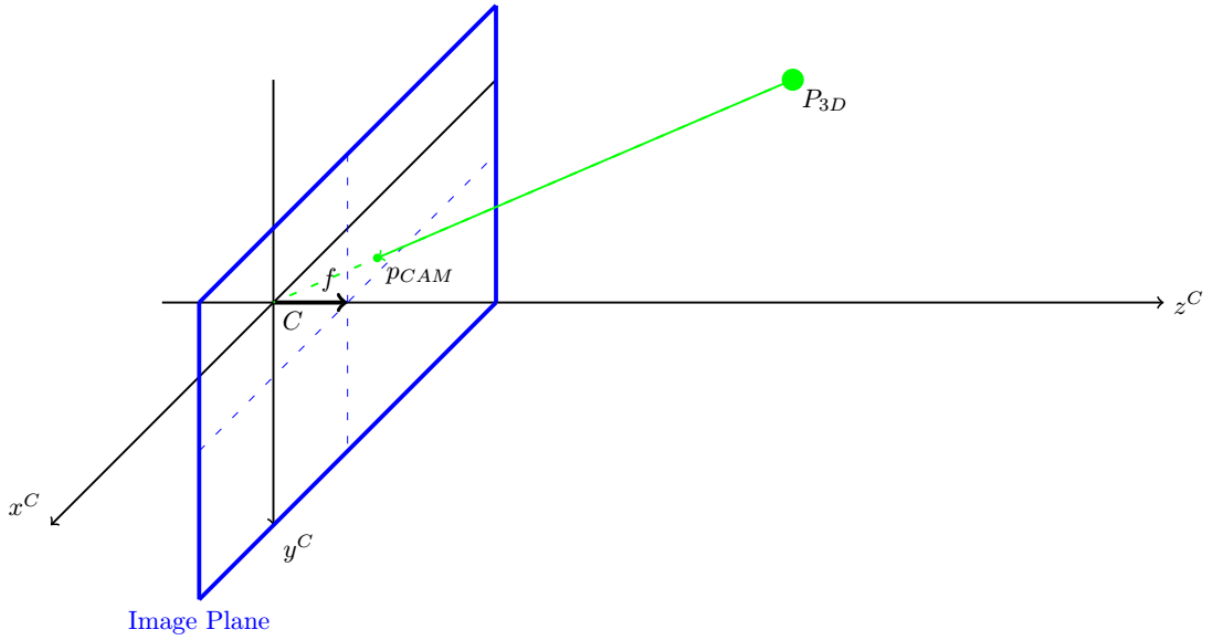
One of the simplest camera models considered by researchers is usually referred as the *pinhole* camera [HARTLEY; ZISSERMAN, 2003]. In this model, the projection of points in the scene are modeled assuming a process of *central projection*. A ray from the point is projected through the lens of a camera in a single point C defined as the *centre* of the lens. This ray intersects a particular plane in space defined as the *image plane*. Note that such modeling does not specifically model geometry and composition of the lens. Let f be the distance between C and the *image plane*. In the *ideal pinhole model*, an arbitrary point Q described by a three dimensional vector p^c_{cQ} projects to image coordinates $[u^{ph}, v^{ph}]^T$

accordingly to the following expression:

$$p_{cQ}^{c} = \begin{bmatrix} p_{cQ}^{c,\chi} \\ p_{cQ}^{c,\gamma} \\ p_{cQ}^{c,\kappa} \end{bmatrix} \Rightarrow \begin{bmatrix} u^{ph} \\ v^{ph} \end{bmatrix} = \begin{bmatrix} f\left((p_{cQ}^{c,\chi})(p_{cQ}^{c,\kappa})^{-1}\right) \\ f\left((p_{cQ}^{c,\gamma})(p_{cQ}^{c,\kappa})^{-1}\right) \end{bmatrix} \tag{4.18}$$

The process is illustrated in Figure 9.

Figure 9: Central projection of a point P_{3D} into image coordinates p_{CAM}



Source: Author (adapted from [HARTLEY; ZISSERMAN, 2003])

Note that image coordinates $[u^{ph}, v^{ph}]^T$ from Equation 4.18 are given in the same unit as the focal distance f and are defined with respect to a coordinate system with origin at the camera center. Let Δ_u and Δ_v be the size of each pixel unit in the image plane, and let $[u_0, v_0]^T$ be the coordinates of the center of the camera C in the scene image in pixels. A model for measurements $[u^{ccd}, v^{ccd}]^T$ of a digital camera may be expressed as:

$$\begin{bmatrix} u^{ccd} \\ v^{ccd} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\left((p_{cQ}^{c,\chi})(p_{cQ}^{c,\kappa})^{-1}\right) + u_0 \\ f\Delta_v^{-1}\left((p_{cQ}^{c,\gamma})(p_{cQ}^{c,\kappa})^{-1}\right) + v_0 \end{bmatrix} = \begin{bmatrix} \Delta_u^{-1}u^{ph} + u_0 \\ \Delta_v^{-1}v^{ph} + v_0 \end{bmatrix} \tag{4.19}$$

The set of parameters $K_{in} = [f\Delta_u^{-1}, f\Delta_v^{-1}, u_0, v_0]$ are referred as the *intrinsic parameters* of the camera. These intrinsic parameters defines the projection of points from the camera coordinate system to their corresponding image coordinates. If a point is described with respect to another coordinate system, it is necessary to transform it to the camera frame before its projection into the camera scene.

Consider the ship frame S_S and let p^s_{sc} be the position and R^c_s be the orientation of the camera with respect to the ship frame. An arbitrary point Q described with position p^s_{sQ} in the ship frame projects into the camera scene as:

$$p^c_{cQ} = R^c_s(p^s_{sQ} - p^s_{sc}) \Rightarrow \begin{bmatrix} p^{c,\chi}_{cQ} \\ p^{c,\gamma}_{cQ} \\ p^{c,\kappa}_{cQ} \end{bmatrix} = \begin{bmatrix} \chi^{s,i}_c & \chi^{s,j}_c & \chi^{s,k}_c \\ \gamma^{s,i}_c & \gamma^{s,j}_c & \gamma^{s,k}_c \\ \kappa^{s,i}_c & \kappa^{s,j}_c & \kappa^{s,k}_c \end{bmatrix} \begin{bmatrix} p^{s,i}_{sQ} - p^{s,i}_{sc} \\ p^{s,j}_{sQ} - p^{s,j}_{sc} \\ p^{s,k}_{sQ} - p^{s,k}_{sc} \end{bmatrix} \Rightarrow \tag{4.20}$$

$$\begin{bmatrix} p^{c,\chi}_{cQ} \\ p^{c,\gamma}_{cQ} \\ p^{c,\kappa}_{cQ} \end{bmatrix} = \begin{bmatrix} \chi^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \chi^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \chi^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc}) \\ \gamma^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \gamma^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \gamma^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc}) \\ \kappa^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \kappa^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \kappa^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc}) \end{bmatrix} \tag{4.21}$$

$$\begin{bmatrix} u^{ccd} \\ \\ v^{ccd} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\left(\frac{\chi^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \chi^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \chi^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc})}{\kappa^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \kappa^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \kappa^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc})}\right) + u_0 \\ \\ f\Delta_v^{-1}\left(\frac{\gamma^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \gamma^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \gamma^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc})}{\kappa^{s,i}_c(p^{s,i}_{sQ} - p^{s,i}_{sc}) + \kappa^{s,j}_c(p^{s,j}_{sQ} - p^{s,j}_{sc}) + \kappa^{s,k}_c(p^{s,k}_{sQ} - p^{s,k}_{sc})}\right) + v_0 \end{bmatrix} \tag{4.22}$$

The parameters $[p^s_{sc}, R^c_s]$ that specifies the camera coordinate system are usually referred as the *extrinsic parameters* of the camera. It is possible to estimate the camera parameters of a given camera device with calibration procedures. Consider a set of N_c points correspondences composed of detections $[u_k, v_k]^T$ and respective three-dimensional coordinates. For each three-dimensional coordinate, an estimated detection $[\hat{u}_k, \hat{v}_k]^T$ is computed as a function of estimated camera parameters and each point position. The error between observations $[u_k, v_k]^T$ and estimations $[\hat{u}_k, \hat{v}_k]^T$ can be defined with a single vector concatenating the error between each correspondence:

$$\vec{E}_{cc} = \begin{bmatrix} u_1 - \hat{u}_1 & v_1 - \hat{v}_1 & \dots & u_k - \hat{u}_k & v_k - \hat{v}_k & \dots & u_{N_c} - \hat{u}_{N_c} & v_{N_c} - \hat{v}_{N_c} \end{bmatrix} \tag{4.23}$$

Neglecting errors in the three-dimensional position of each calibration point, the error vector $\vec{E}_{cc}$ from Equation 4.23 tends to zero when estimated camera parameters are equal to real camera parameters. There are different optimization methods that can be used for the determination of calibration parameters from the minimization of $\vec{E}_{cc}$.

On a computational perspective, it is possible to use a general purpose optimizer for this calibration with open source numerical libraries. Typically, these optimization libraries expects expressions for the error as function of the parameters to be optimized. Considering the previous models which express each projection as a function of camera parameters and points coordinates, it is possible to solve for the calibration parameters by feeding these numerical libraries with the aforementioned functions. One example of

such optimizer is available in the *SciPy* library for optimization [VIRTANEN et al., 2020].

The implementation that is used throughout this work is available in the function *least_squares* of the library. The implementation search for an optimal $\vec{\lambda}_{optm}$ that minimizes a scalar cost function $E_{optm}(\vec{\lambda}_{optm})$. The scalar cost is computed from a loss function $\rho(.)$ and a set of M residual measurements $\vec{e}_i(\vec{\lambda}_{optm})$ accordingly to the following form:

$$E_{optm}(\vec{\lambda}_{optm}) = \frac{1}{2}\sum_{i=1}^{M}\left(\rho(||\vec{e}_i(\vec{\lambda}_{optm})||^2)\right) \tag{4.24}$$

where $\vec{e}_i(\vec{\lambda}_{optm})$ is the $i-$th residual as a function of the optimization parameters $\vec{\lambda}_{optm}$. The default implementation of the *least_squares* function uses a trust region reflective algorithm [BYRD; SCHNABEL; SHULTZ, 1988], [BRANCH; COLEMAN; LI, 1999] that allow bounds for the optimization parameters $\vec{\lambda}_{optm}$. Another implementation of the *least_squares* function available in the *SciPy* library is commonly referred as the Levenberg-Marquardt algorithm [LEVENBERG, 1944], [MARQUARDT, 1963], [MORÉ, 1978]. Furthermore, there are alternative optimization implementations in the *SciPy* library that may be further investigated.

4.2 Distortion Models

The projection of points into the image scene of real cameras deviates from the previous ideal *pinhole* camera model due to many factors. Different models have been proposed to characterize such *distortion*. Generally, the most important type of distortion is referred as radial distortion [HARTLEY; ZISSERMAN, 2003]. Radial distortion can be attributed to an imperfect radial curvature of the camera lens. Such distortions are more significant as the focal length of the lens decreases. Let $[\tilde{u}, \tilde{v}]^T$ be image coordinates from an ideal pinhole projection model with unitary focal length:

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\end{bmatrix} = \frac{1}{f}\begin{bmatrix}\Delta_u(u^{ccd}-u_0)\\ \Delta_v(v^{ccd}-v_0)\end{bmatrix} = \frac{1}{f}\begin{bmatrix}u^{ph}\\ v^{ph}\end{bmatrix} = \begin{bmatrix}(p_{cQ}^{c,\chi})(p_{cQ}^{c,\kappa})^{-1}\\ (p_{cQ}^{c,\gamma})(p_{cQ}^{c,\kappa})^{-1}\end{bmatrix} \tag{4.25}$$

Let $[u_d^{ccd}, v_d^{ccd}]^T$ be distorted coordinates after radial distortion. Such distortion can be modeled as a function of an ideal radius $\tilde{r}$ computed from the above ideal pinhole

projections alongside a set of *radial distortion* parameters:

$$\tilde{r} = \sqrt{\left(\tilde{u}\right)^2 + \left(\tilde{v}\right)^2} \tag{4.26}$$

$$L(\tilde{r}) = 1 + k_{r1}\tilde{r} + k_{r2}\tilde{r}^2 + ... + k_{rn}\tilde{r}^n \tag{4.27}$$

$$\begin{bmatrix} \tilde{u}_d \\ \tilde{v}_d \end{bmatrix} = L(\tilde{r}) \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} \Rightarrow \tag{4.28}$$

$$\begin{bmatrix} u_d^{ccd} \\ v_d^{ccd} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\tilde{u}_d + u_0 \\ f\Delta_v^{-1}\tilde{v}_d + v_0 \end{bmatrix} \tag{4.29}$$

Physically, the distortion function $L(\tilde{r})$ needs to be monotonic. In order words, $L(\tilde{r})$ needs to continuously increase as $\tilde{r}$ increases, or continuously decrease as $\tilde{r}$ increases.

Another component of lens distortion is classically referred as *decentering distortion*. Such distortion is mathematically complex to be expressed analytically. In [BROWN, 1966], *decentering distortion* is physically discussed and a distortion model is proposed based on the earlier work of [CONRADY, 1919]. Nowadays, this model is usually referred as the *Brown-Conrady* distortion model. There are implementations based on this model in the open-source library *OpenCV* [BRADSKI; KAEHLER, 2008]. Let $[\tilde{u}, \tilde{v}]^T$ be the coordinates from the ideal pinhole projection without distortion similarly as before, and let $[u_d^{ccd}, v_d^{ccd}]^T$ be the distorted coordinates after distortion.

$$\tilde{r} = \sqrt{\left(\tilde{u}\right)^2 + \left(\tilde{v}\right)^2} \tag{4.30}$$

$$L(\tilde{r}) = \frac{1 + k_{r1}\tilde{r}^2 + k_{r2}\tilde{r}^4 + k_{r3}\tilde{r}^6}{1 + k_{r4}\tilde{r}^2 + k_{r5}\tilde{r}^4 + k_{r6}\tilde{r}^6} \tag{4.31}$$

$$\begin{bmatrix} \tilde{u}_d \\ \tilde{v}_d \end{bmatrix} = \begin{bmatrix} L(\tilde{r})\tilde{u} + 2p_1\tilde{u}\tilde{v} + p_2(\tilde{r}^2 + 2\tilde{u}^2) + s_1\tilde{r}^2 + s_2\tilde{r}^4 \\ L(\tilde{r})\tilde{v} + 2p_2\tilde{u}\tilde{v} + p_1(\tilde{r}^2 + 2\tilde{v}^2) + s_3\tilde{r}^2 + s_4\tilde{r}^4 \end{bmatrix} \Rightarrow \tag{4.32}$$

$$\begin{bmatrix} u_d^{ccd} \\ v_d^{ccd} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\tilde{u}_d + u_0 \\ f\Delta_v^{-1}\tilde{v}_d + v_0 \end{bmatrix} \tag{4.33}$$

The function *calibrateCamera* of the *OpenCV* library implements an optimization for estimating the intrinsic and extrinsic parameters of the camera from points correspondences. The implementation is based on the Levenberg-Marquardt algorithm.

In [WANG et al., 2008] an interesting model for camera lens distortion is presented. The distortion is modeled with radial distortion coefficients and a transformation from the ideal image plane to the real sensor array. This yields a model with fewer parameters

to be calibrated and more explicit physical meaning. Four stages constitute the proposed image projection process. Let $[\tilde{u}, \tilde{v}]^T$ be the coordinates from the ideal pinhole projection without distortion:

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \frac{1}{f} \begin{bmatrix} \Delta_u(u^{ccd} - u_0) \\ \Delta_v(v^{ccd} - v_0) \end{bmatrix} = \begin{bmatrix} (p_{cQ}^{c,\chi})(p_{cQ}^{c,\kappa})^{-1} \\ (p_{cQ}^{c,\gamma})(p_{cQ}^{c,\kappa})^{-1} \end{bmatrix} \tag{4.34}$$

Coordinates $[\tilde{u}, \tilde{v}]$ from Equation 4.34 radially shifts to distorted coordinates $[u_{rd}, v_{rd}]$:

$$\tilde{r} = \sqrt{\tilde{u}^2 + \tilde{v}^2} \tag{4.35}$$

$$L(\tilde{r}) = 1 + k_{r1}\tilde{r}^2 + k_{r2}\tilde{r}^4 \tag{4.36}$$

$$\begin{bmatrix} u_{rd} \\ v_{rd} \end{bmatrix} = L(\tilde{r}) \begin{bmatrix} (p_{cQ}^{c,\chi})(p_{cQ}^{c,\kappa})^{-1} \\ (p_{cQ}^{c,\gamma})(p_{cQ}^{c,\kappa})^{-1} \end{bmatrix} \tag{4.37}$$

The radially distorted coordinates $[u_{rd}, v_{rd}]$ are further transformed to the real sensor array plane yielding sensor coordinates $[x_{srd}, y_{srd}, z_{srd}]$. This transformation between the ideal plane and the real sensor array is computed with a rotation matrix R_{DIS}:

$$\begin{bmatrix} x_{srd} \\ y_{srd} \\ z_{srd} \end{bmatrix} = R_{DIS} \begin{bmatrix} u_{rd} \\ v_{rd} \\ 0 \end{bmatrix} \tag{4.38}$$

The rotation matrix R_{DIS} is defined as a composition of two rotations around axes x and y. Suppose that the rotation around axis x is given by θ and the rotation around axis y is given by ψ.

$$R_{DIS} = R_\psi R_\theta = \begin{bmatrix} c(\psi) & 0 & s(\psi) \\ 0 & 1 & 0 \\ -s(\psi) & 0 & c(\psi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) \\ 0 & s(\theta) & c(\theta) \end{bmatrix} \tag{4.39}$$

The following simplifications can be used for θ and ψ small:

$$R_\psi \approx \begin{bmatrix} 1 & 0 & \psi \\ 0 & 1 & 0 \\ -\psi & 0 & 1 \end{bmatrix} \quad , \quad R_\theta \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\theta \\ 0 & \theta & 1 \end{bmatrix} \quad \Rightarrow R_{DIS} = R_\psi R_\theta \approx \begin{bmatrix} 1 & 0 & \psi \\ 0 & 1 & -\theta \\ -\psi & \theta & 1 \end{bmatrix} \tag{4.40}$$

Let $[u_{sd}, v_{sd}]^T$ denotes the image coordinates in the real sensor array plane. With the simplification from Equation 4.40 the corresponding coordinates in the real sensor array

plane can be computed as:

$$\begin{bmatrix} u_{sd} \\ v_{sd} \end{bmatrix} = \begin{bmatrix} u_{rd} f \left(\theta v_{rd} - \psi u_{rd} + f \right)^{-1} \\ v_{rd} f \left(\theta v_{rd} - \psi u_{rd} + f \right)^{-1} \end{bmatrix} \tag{4.41}$$

The corresponding final pixel coordinates $[u_d^{ccd}, v_d^{ccd}]^T$ in the measured digital image can be computed considering the sensor unit cell sizes $[\Delta_u, \Delta_v]$ and principal points coordinates $[u_0, v_0]^T$:

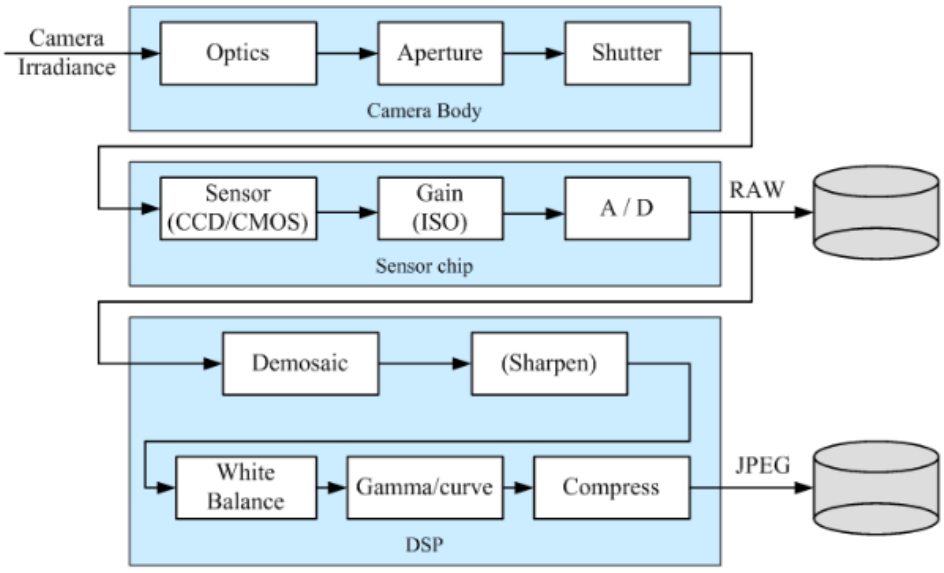
$$\begin{bmatrix} u_d^{ccd} \\ v_d^{ccd} \end{bmatrix} = \begin{bmatrix} f \Delta_u^{-1} u_{sd} + u_0 \\ f \Delta_v^{-1} v_{sd} + v_0 \end{bmatrix} \tag{4.42}$$

Note that more sophisticated models beyond the scope of the present work can be developed by taking in consideration mechanical properties of lens such as geometry and composition. For most applications it is possible to simply choose a camera device that does not present relevant distortion and use a camera model without distortion as from the previous section.

4.3 Hardware

A typical image sensing pipeline is shown in Figure 10.

Figure 10: Typical digital camera imaging pipeline



Source: [SZELISKI, 2010]

Light from the scene is measured by an *active sensing area* during a time interval referred as the *exposure time*. The measurement is further amplified and feed to analog to digital converters (**ADC**). Additional digital signal processing operations are performed before compressing and storing the final pixel intensity values. In colored cameras, these intensity values are expressed as three separate intensity values with regard to three basic colors: *blue*, *red*, and *green*.

Consider the representation for a digital colored image by concatenating three matrices I_B, I_R, and I_G. Each matrix represents the intensity values for each color.

$$I[u,v] = [I_B[u,v], I_R[u,v], I_G[u,v]] \tag{4.43}$$

For a sensor chip with n_u pixels in the u direction and n_v pixels in the v direction, the corresponding matrices I_B, I_R, and I_G have dimensions $n_u \times n_v$. Let $L_{u,v}(\lambda)$ be the spectrum of light reaching pixel $[u,v]^T$ of the sensor plane. $L_{u,v}(\lambda)$ describes the incoming light as a function of the wavelength λ. Assume that each sensor of the sensor plane has a *spectral sensitivity* function describing the response of the sensor for different incident light $L_{u,v}(\lambda)$. Let $S_B^{u,v}(\lambda)$, $S_G^{u,v}(\lambda)$, $S_R^{u,v}(\lambda)$ be the *spectral sensitivity* of the corresponding blue, green and red sensors of the pixel area. The pixel intensity for each color channel may be modeled as being proportional to each corresponding *spectral response function*, which is defined as:

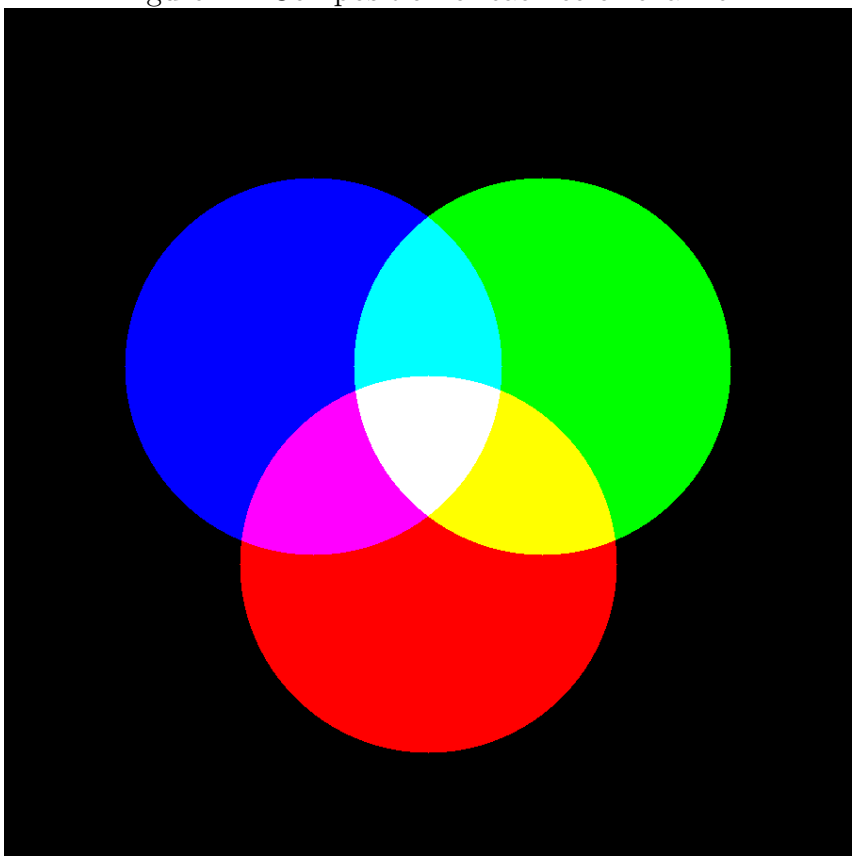
$$I_B[u,v] \sim B_{u,v} = \int L_{u,v}(\lambda) S_B^{u,v}(\lambda) d\lambda \tag{4.44}$$

$$I_G[u,v] \sim G_{u,v} = \int L_{u,v}(\lambda) S_G^{u,v}(\lambda) d\lambda \tag{4.45}$$

$$I_R[u,v] \sim R_{u,v} = \int L_{u,v}(\lambda) S_R^{u,v}(\lambda) d\lambda \tag{4.46}$$

Pixels are usually sampled at a 8-bit resolution, which yields values ranging from 0 to 255. Note that if the spectrum of incident light $L_{u,v}(\lambda)$ in a pixel area is null $\forall\lambda$, all pixel measurements are zero. Thus, the corresponding color for a pixel $I[u,v] = [0,0,0]$ is black. On the other hand, the pixel measurement relative to the white color is given by $I[u,v] = [255,255,255]$. Figure 11 shows a digital image generated with the *OpenCV* library with three circles for each color channel illustrating the composition of different colors.

Figure 11: Composition of each color channel



Source: Author (adapted from [SZELISKI, 2010])

4.4 Image Processing

This work uses the open-source library OpenCV [BRADSKI; KAEHLER, 2008], [KAEHLER; BRADSKI, 2016] for most of the digital processing implementations. Tables 3,4, 5 presents methods that are used in the following chapters. Table 3 summarizes methods related with the generation of the graphical elements. Methods for the implementation of calibration and projection procedures are summarized in Table 4. General computer vision algorithms are summarized in Table 5.

Template matching is a technique that is based on the comparison of an *template image* in every location of a *scene image*. An error is computed between the *template image* pixels and the scene pixels, yielding a map of errors. The pixel position with the minimum error value is the resulting detection of the algorithm. The following error

Table 3: **OpenCV** Methods for Image Synthesis

Method Name	Description
circle	Draws a circle in an image object
line	Draws a line between two points p_1 and p_2 described in pixel coordinates
arrowedLine	Draws an arrowed line between two points p_1 and p_2 described in pixel coordinates. The arrow points from p_1 towards p_2
rectangle	Draws a rectangle in an image object
putText	Draws a *string* in an image object
resize	Resize an image object for a different image size
addWeighted	Performs a weighted sum of two image objects

Source: Author (adapted from **OpenCV** documentation)

Table 4: **OpenCV** Methods for Calibration and Projection

Method Name	Description
calibrateCamera	Finds the camera intrinsic and extrinsic parameters from several views of a calibration pattern
solvePnP	Finds an object pose from 3D-2D points correspondences
projectPoints	Projects 3D points to an image plane

Source: Author (adapted from **OpenCV** documentation)

Table 5: **OpenCV** Methods for Computer Vision

Method Name	Description
matchTemplate minMaxLoc	Implementation for *template matching* between an image patch (*template*) and an input image
grabCut	Implementation for *image segmentation* of an input image from an initial mask

Source: Author (adapted from **OpenCV** documentation)

expressions are available in the *OpenCV* library:

$$R^{SQD}(u,v) = \sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} \Big(T(u_t,v_t) - I(u+u_t, v+v_t) \Big)^2 \tag{4.47}$$

$$R^{SQDN}(u,v) = \frac{\sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} \Big(T(u_t,v_t) - I(u+u_t, v+v_t) \Big)^2}{\sqrt{\Big(\sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} T(u_t,v_t)^2 \Big) \Big(\sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} I(u+u_t, v+v_t)^2 \Big)}} \tag{4.48}$$

Alternatively, it is possible to define a similarity expression instead of the errors from Equations 4.47 and 4.48. The similarity measurement computed in each pixel of the image yields a similarity map, and the pixel position with the maximum similarity value

is the resulting detection. The following similarity expressions are available in the *OpenCV* library:

$$R^{CCORR}(u,v) = \sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} \Big(T(u_t,v_t) I(u+u_t, v+v_t) \Big) \tag{4.49}$$

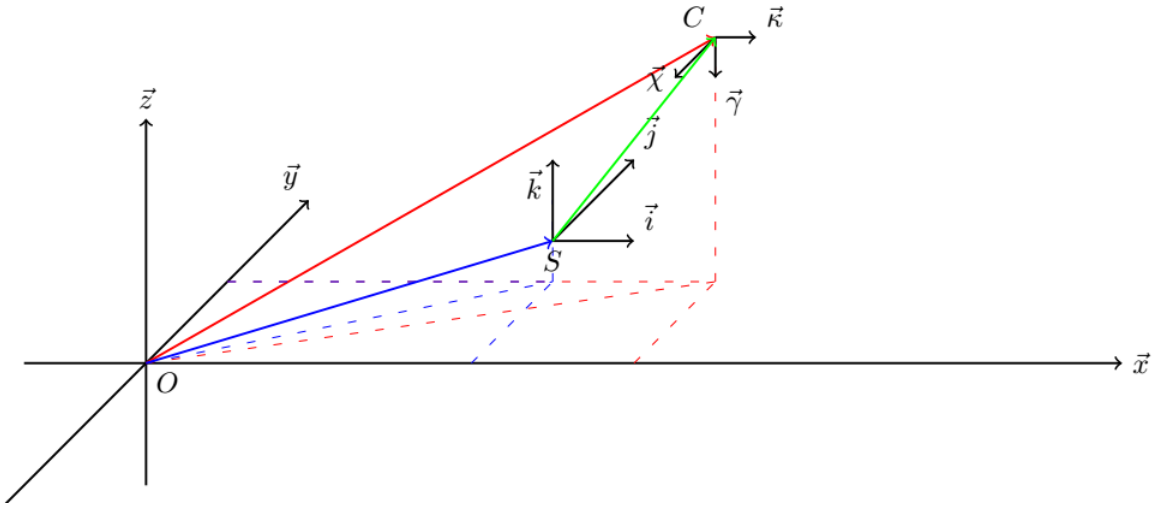
$$R^{CCORRN}(u,v) = \frac{\sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} \Big(T(u_t,v_t) I(u+u_t, v+v_t) \Big)}{\sqrt{\Big(\sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} T(u_t,v_t)^2 \Big) \Big(\sum_{u_t=0}^{N_{t,u}} \sum_{v_t=0}^{N_{t,v}} I(u+u_t, v+v_t)^2 \Big)}} \tag{4.50}$$

The image segmentation algorithm is initialized with a mask M of the same size as the input image. Each coordinate $[u,v]^T$ of the mask is related to the corresponding region of the input image. The values of the mask are constrained to four possible values accordingly to the corresponding types of segmentation areas: (0) *background*; (1) *foreground*; (2) *probable background*; and (3) *probable foreground*. The algorithm refines the initial input mask with nonlinear optimizations yielding an optimized mask.

5 MONITOR AUGMENTED REALITY

Consider the navigation of a ship equipped with an onboard camera. The video stream of the camera measures the scene as the ship moves in the sea. The video can be enhanced with visual elements to assist operators with the perception of navigation scene. In order to display such visual elements, it is interesting to define three different coordinate systems: the *world coordinate system* (**WCS** or S_W), the *ship coordinate system* (**SCS** or S_S) and the *camera coordinate system* (**CCS** or S_C). The transformation between S_S and S_C is fixed assuming that the camera is rigidly installed in the ship. On the other hand, the transformation between S_W and S_S changes as the ship moves in the sea. An overview of the defined coordinate systems is illustrated by Figure 12.

Figure 12: Overview of pertinent coordinates systems: S_W, S_S and S_C

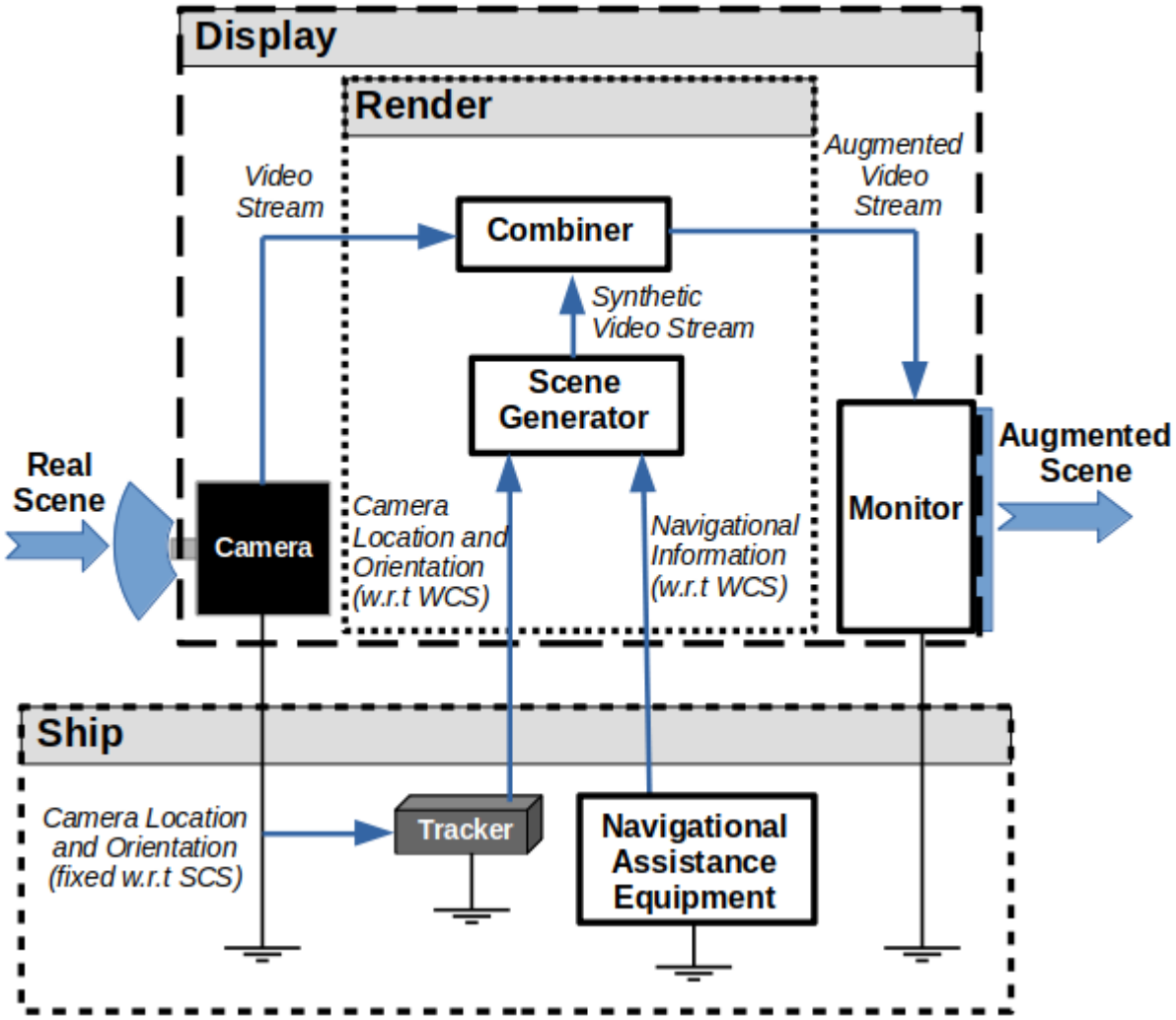


Source: Author

This chapter discusses a potential *monitor augmented reality* architecture for assisting in maritime navigation. Figure 13 presents a diagram of the proposed *monitor augmented reality* architecture. The system may be defined as a composition of four main blocks: (1) *display*; (2) *tracker*; (3) *navigational information*; (4) *render*.

The *display* block comprises all equipment that are related with recording and pre-

Figure 13: Overview of the proposed architecture



Source: Author

senting the navigation scene for operators: a calibrated camera and a display monitor. The camera is rigidly fixed to the ship and the monitor is installed in the ship control room (bridge). The video stream from the camera is available in the monitor. The camera state with respect to the SCS is assumed to be known.

The *tracker* block is related to the estimation of the ship state with respect to the WCS. There are different technologies for this implementation. A typical method for *tracking* defines a preliminary reference solution by the integration of inertial sensors. The reference solution is improved with information from other sensors. Note that as the camera is rigidly attached to the ship, it is possible to determine the camera state with respect to the WCS from the *tracker* solution for the ship state.

The *navigational information* block of the system comprises navigational assistance

equipment which are able to provide information regarding the ship navigation in a digital form. Examples of such *maritime navigational assistance equipment* are *radars* and *electronic charts*. The information provided by such equipment may be automatically projected into the camera scene to assist in its spatial perception. As it will become clearer in the following sections, it is interesting to differentiate between two types of navigational information: (1) *ship-related*; and (2) *scene-related* information.

Finally, the *render* block generates synthetic views to represent the information from other blocks of the system. Further, the *render* block overlay these synthetic views on top of the real camera scene. This block of the system is further detailed with experiments from recorded videos in the following chapters.

The present work is a preliminary investigation towards the development of such a *monitor augmented reality* system. The following sections further detail each block of the system defining sets of corresponding installation parameters, which are highlighted throughout the text. Note that the full development of an operational *monitor augmented reality* prototype for navigational assistance is beyond the scope of the current work. Instead, foundations for developing this equipment are presented and exemplified with recorded videos from navigation experiments. These videos are post-processed in order to generate potential synthetic views for navigational assistance. Proposed methods may be incorporated into the *render* block of an operational *monitor augmented reality* prototype.

5.1 Display

There are two basic instruments regarding the *display* block of the system: the camera and the monitor. Generally, the initial step for developing a *monitor augmented reality* system should be the determination of the camera device for the system. A preliminary recommendation is to choose a camera device that does not present significant distortion. Neglecting distortion effects, let K_{in} be the set of *intrinsic parameters* that describes the projection of points with respect to the CCS (S_C):

$$\boxed{K_{in} = [f\Delta_u^{-1}, f\Delta_v^{-1}, u_0, v_0] = [f_u, f_v, u_0, v_0]} \tag{5.1}$$

Let $[u_Q, v_Q]^T$ be the projection of an arbitrary point Q described by its position p_{cQ}^c

with respect to the camera frame as in Equation 4.19:

$$\begin{bmatrix} u_Q \\ v_Q \end{bmatrix} = \begin{bmatrix} f_u p_{cQ}^{c,\chi} (p_{cQ}^{c,\kappa})^{-1} + u_0 \\ f_v p_{cQ}^{c,\gamma} (p_{cQ}^{c,\kappa})^{-1} + v_0 \end{bmatrix} \tag{5.2}$$

Furthermore, let p_{sc}^s and R_c^s be the camera position and orientation with respect to the SCS (S_S).

$$\boxed{p_{sc}^s = \begin{bmatrix} p_{sc}^{s,i} \\ p_{sc}^{s,j} \\ p_{sc}^{s,k} \end{bmatrix} \quad , \quad R_c^s = \begin{bmatrix} \chi_c^{s,i} & \gamma_c^{s,i} & \kappa_c^{s,i} \\ \chi_c^{s,j} & \gamma_c^{s,j} & \kappa_c^{s,j} \\ \chi_c^{s,k} & \gamma_c^{s,k} & \kappa_c^{s,k} \end{bmatrix}} \tag{5.3}$$

For an arbitrary point Q described by its position p_{sQ}^s with respect to the ship frame, its projection $[u_Q, v_Q]^T$ may be computed with Equation 4.22:

$$\begin{bmatrix} u_Q \\ \\ v_Q \end{bmatrix} = \begin{bmatrix} f_u \left(\frac{\chi_c^{s,i}(p_{sQ}^{s,i} - p_{sc}^{s,i}) + \chi_c^{s,j}(p_{sQ}^{s,j} - p_{sc}^{s,j}) + \chi_c^{s,k}(p_{sQ}^{s,k} - p_{sc}^{s,k})}{\kappa_c^{s,i}(p_{sQ}^{s,i} - p_{sc}^{s,i}) + \kappa_c^{s,j}(p_{sQ}^{s,j} - p_{sc}^{s,j}) + \kappa_c^{s,k}(p_{sQ}^{s,k} - p_{sc}^{s,k})} \right) + u_0 \\ \\ f_v \left(\frac{\gamma_c^{s,i}(p_{sQ}^{s,i} - p_{sc}^{s,i}) + \gamma_c^{s,j}(p_{sQ}^{s,j} - p_{sc}^{s,j}) + \gamma_c^{s,k}(p_{sQ}^{s,k} - p_{sc}^{s,k})}{\kappa_c^{s,i}(p_{sQ}^{s,i} - p_{sc}^{s,i}) + \kappa_c^{s,j}(p_{sQ}^{s,j} - p_{sc}^{s,j}) + \kappa_c^{s,k}(p_{sQ}^{s,k} - p_{sc}^{s,k})} \right) + v_0 \end{bmatrix} \tag{5.4}$$

Consider a set of estimated camera parameters $\hat{K}_{in}$, $\hat{p}_{sc}^s$ and $\hat{R}_c^s$:

$$\boxed{\hat{K}_{in} = [\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0] \quad , \quad \hat{p}_{sc}^s = \begin{bmatrix} \hat{p}_{sc}^{s,i} \\ \hat{p}_{sc}^{s,j} \\ \hat{p}_{sc}^{s,k} \end{bmatrix} \quad , \quad \hat{R}_c^s = \begin{bmatrix} \hat{\chi}_c^{s,i} & \hat{\gamma}_c^{s,i} & \hat{\kappa}_c^{s,i} \\ \hat{\chi}_c^{s,j} & \hat{\gamma}_c^{s,j} & \hat{\kappa}_c^{s,j} \\ \hat{\chi}_c^{s,k} & \hat{\gamma}_c^{s,k} & \hat{\kappa}_c^{s,k} \end{bmatrix}} \tag{5.5}$$

Similarly, the corresponding estimation $[\hat{u}_Q, \hat{v}_Q]$ may be expressed with Equation 4.22:

$$\begin{bmatrix} \hat{u}_Q \\ \\ \hat{v}_Q \end{bmatrix} = \begin{bmatrix} \hat{f}_u \left(\frac{\hat{\chi}_c^{s,i}(p_{sQ}^{s,i} - \hat{p}_{sc}^{s,i}) + \hat{\chi}_c^{s,j}(p_{sQ}^{s,j} - \hat{p}_{sc}^{s,j}) + \hat{\chi}_c^{s,k}(p_{sQ}^{s,k} - \hat{p}_{sc}^{s,k})}{\hat{\kappa}_c^{s,i}(p_{sQ}^{s,i} - \hat{p}_{sc}^{s,i}) + \hat{\kappa}_c^{s,j}(p_{sQ}^{s,j} - \hat{p}_{sc}^{s,j}) + \hat{\kappa}_c^{s,k}(p_{sQ}^{s,k} - \hat{p}_{sc}^{s,k})} \right) + \hat{u}_0 \\ \\ \hat{f}_v \left(\frac{\hat{\gamma}_c^{s,i}(p_{sQ}^{s,i} - \hat{p}_{sc}^{s,i}) + \hat{\gamma}_c^{s,j}(p_{sQ}^{s,j} - \hat{p}_{sc}^{s,j}) + \hat{\gamma}_c^{s,k}(p_{sQ}^{s,k} - \hat{p}_{sc}^{s,k})}{\hat{\kappa}_c^{s,i}(p_{sQ}^{s,i} - \hat{p}_{sc}^{s,i}) + \hat{\kappa}_c^{s,j}(p_{sQ}^{s,j} - \hat{p}_{sc}^{s,j}) + \hat{\kappa}_c^{s,k}(p_{sQ}^{s,k} - \hat{p}_{sc}^{s,k})} \right) + \hat{v}_0 \end{bmatrix} \tag{5.6}$$

5.2 Tracker

One of the most important blocks of the system is the *tracker* block. This block is responsible for determining the ship position and orientation with respect to a world

coordinate system S_W. Let p^e_{es} be the ship cartesian position with respect to the earth frame and let R^e_s be the rotation matrix relating the ship frame with the earth frame:

$$\boxed{p^e_{es} = \begin{bmatrix} p^{e,x}_{es} \\ p^{e,y}_{es} \\ p^{e,z}_{es} \end{bmatrix} \quad , \quad R^e_s = \begin{bmatrix} i^{e,x}_s & j^{e,x}_s & k^{e,x}_s \\ i^{e,y}_s & j^{e,y}_s & k^{e,y}_s \\ i^{e,z}_s & j^{e,z}_s & k^{e,z}_s \end{bmatrix}} \tag{5.7}$$

Similarly, let $\acute{p}^e_{es}$ and $\acute{R}^e_s$ be the tracker outputs for position and orientation of the ship with respect to the earth, respectively.

$$\boxed{\acute{p}^e_{es} = \begin{bmatrix} \acute{p}^{e,i}_{es} \\ \acute{p}^{e,j}_{es} \\ \acute{p}^{e,k}_{es} \end{bmatrix} \quad , \quad \acute{R}^e_s = \begin{bmatrix} \acute{i}^{e,x}_s & \acute{j}^{e,x}_s & \acute{k}^{e,x}_s \\ \acute{i}^{e,y}_s & \acute{j}^{e,y}_s & \acute{k}^{e,y}_s \\ \acute{i}^{e,z}_s & \acute{j}^{e,z}_s & \acute{k}^{e,z}_s \end{bmatrix}} \tag{5.8}$$

Considering that the camera position p^s_{sc} and orientation R^s_c with respect to the SCS are time-invariant, the camera position p^e_{ec} and orientation R^e_c with respect to the earth may be computed as:

$$p^e_{ec} = p^e_{es} + R^e_s p^s_{sc} \tag{5.9}$$

$$R^e_c = R^e_s R^s_c \tag{5.10}$$

Thus, estimations $\hat{p}^e_{ec}$ and $\hat{R}^e_c$ for the camera position and orientation with respect to the earth frame may be computed from the tracker outputs $\acute{p}^e_{es}$ and $\acute{R}^e_s$:

$$\hat{p}^e_{ec} = \acute{p}^e_{es} + \acute{R}^e_s \hat{p}^s_{sc} \Rightarrow \begin{bmatrix} \hat{p}^{e,x}_{ec} \\ \hat{p}^{e,y}_{ec} \\ \hat{p}^{e,z}_{ec} \end{bmatrix} = \begin{bmatrix} \acute{p}^{e,i}_{es} \\ \acute{p}^{e,j}_{es} \\ \acute{p}^{e,k}_{es} \end{bmatrix} + \begin{bmatrix} \acute{i}^{e,x}_s & \acute{j}^{e,x}_s & \acute{k}^{e,x}_s \\ \acute{i}^{e,y}_s & \acute{j}^{e,y}_s & \acute{k}^{e,y}_s \\ \acute{i}^{e,z}_s & \acute{j}^{e,z}_s & \acute{k}^{e,z}_s \end{bmatrix} \begin{bmatrix} \hat{p}^{s,i}_{sc} \\ \hat{p}^{s,j}_{sc} \\ \hat{p}^{s,k}_{sc} \end{bmatrix} \tag{5.11}$$

$$\hat{R}^e_c = \acute{R}^e_s \hat{R}^s_c = \begin{bmatrix} \acute{i}^{e,x}_s & \acute{j}^{e,x}_s & \acute{k}^{e,x}_s \\ \acute{i}^{e,y}_s & \acute{j}^{e,y}_s & \acute{k}^{e,y}_s \\ \acute{i}^{e,z}_s & \acute{j}^{e,z}_s & \acute{k}^{e,z}_s \end{bmatrix} \begin{bmatrix} \hat{\chi}^{s,i}_c & \hat{\gamma}^{s,i}_c & \hat{\kappa}^{s,i}_c \\ \hat{\chi}^{s,j}_c & \hat{\gamma}^{s,j}_c & \hat{\kappa}^{s,j}_c \\ \hat{\chi}^{s,k}_c & \hat{\gamma}^{s,k}_c & \hat{\kappa}^{s,k}_c \end{bmatrix} \tag{5.12}$$

Let O_w be the origin of S_W described by its curvilinear position $[L_w, \lambda_w, h_w]$ with respect to the earth. The corresponding cartesian position p^e_{ew} of S_W with respect to the

earth frame may be computed with Equations 3.40, 3.43, 3.44 and 3.45:

$$\boxed{[L_w, \lambda_w, h_w] \Rightarrow p^e_{ew} = \begin{bmatrix} (R_E(L_w) + h_w)\cos(L_w)\cos(\lambda_w) \\ (R_E(L_w) + h_w)\cos(L_w)\sin(\lambda_w) \\ [(1 - (e^{WGS})^2)R_E(L_w) + h_w]\sin(L_w) \end{bmatrix}} \tag{5.13}$$

Furthermore, let $[\vec{x}_w, \vec{y}_w, \vec{z}_w]$ be the axes of S_w. It is possible to describe the axes of S_w with a rotation matrix R^e_w between S_w and the earth frame:

$$\boxed{R^e_w = \begin{bmatrix} \vec{x}^e_w & \vec{y}^e_w & \vec{z}^e_w \end{bmatrix} = \begin{bmatrix} x^{e,x}_w & y^{e,x}_w & z^{e,x}_w \\ x^{e,y}_w & y^{e,y}_w & z^{e,y}_w \\ x^{e,z}_w & y^{e,z}_w & z^{e,z}_w \end{bmatrix}} \tag{5.14}$$

Note that it is typically simpler to the define the axes of S_w in terms of the local navigation frame. Alternatively, assume that the axes of S_w are described by the rotation matrix R^n_w between S_w and the navigation frame:

$$\boxed{R^n_w = \begin{bmatrix} \vec{x}^n_w & \vec{y}^n_w & \vec{z}^n_w \end{bmatrix} = \begin{bmatrix} x^{n,x}_w & y^{n,x}_w & z^{n,x}_w \\ x^{n,y}_w & y^{n,y}_w & z^{n,y}_w \\ x^{n,z}_w & y^{n,z}_w & z^{n,z}_w \end{bmatrix}} \tag{5.15}$$

The corresponding rotation matrix R^e_w between the world reference frame and the earth frame may be computed from R^n_w with Equation 3.51:

$$R^e_w = R^e_n(L_w, \lambda_w, h_w)R^n_w \tag{5.16}$$

$$R^e_w = \begin{bmatrix} -\sin(L_w)\cos(\lambda_w) & -\sin(\lambda_w) & -\cos(L_w)\cos(\lambda_w) \\ -\sin(L_w)\sin(\lambda_w) & \cos(\lambda_w) & -\cos(L_w)\sin(\lambda_w) \\ \cos(L_w) & 0 & -\sin(L_w) \end{bmatrix} \begin{bmatrix} x^{n,x}_w & y^{n,x}_w & z^{n,x}_w \\ x^{n,y}_w & y^{n,y}_w & z^{n,y}_w \\ x^{n,z}_w & y^{n,z}_w & z^{n,z}_w \end{bmatrix} \tag{5.17}$$

Therefore, estimations $\hat{p}^w_{wc}$ and R^w_c for the camera position and orientation with respect to the world frame S_w may be computed from Equation 3.60:

$$\hat{p}^w_{wc} = R^w_e(\hat{p}^e_{ec} - p^e_{ew}) = (R^e_w)^T(\hat{p}^e_{ec} - p^e_{ew}) \tag{5.18}$$

$$\hat{R}^w_c = R^w_e\hat{R}^e_c = (R^e_w)^T\hat{R}^e_c \tag{5.19}$$

5.3 Navigational Information

Another important block of the system is the *navigational information* block. Note that most of the information related with navigation are provided by onboard navigational assistance equipment. If such information is digitally available, it is possible to automatically display navigational information on top of the camera scene. Examples of such *maritime navigational assistance equipment* are *radars* and *electronic charts*, which provides extremely important information regarding the navigation scene.

Consider a set of N_Q points $\mathbf{Q}$ corresponding to different features of the navigation scene:

$$\boxed{\mathbf{Q} = \begin{bmatrix} Q_1, ..., Q_k, ..., Q_{N_Q} \end{bmatrix}} \tag{5.20}$$

Assuming that the position of each feature Q_k is known with respect to the earth frame from onboard navigational assistance equipment, it is possible to determine image coordinates $[u_{Q_k}, v_{Q_k}]^T$ for each point with outputs $\hat{p}^e_{ec}$ and $\hat{R}^e_c$ from Equations 5.11 and 5.12:

$$p^e_{eQ_k} = \begin{bmatrix} p^{e,x}_{eQ_k} \\ p^{e,y}_{eQ_k} \\ p^{e,z}_{eQ_k} \end{bmatrix} \Rightarrow \hat{p}^c_{cQ_k} = (\hat{R}^e_c)^T(p^e_{eQ_k} - \hat{p}^e_{ec}) = \hat{R}^c_e(p^e_{eQ_k} - \hat{p}^e_{ec}) = \begin{bmatrix} \hat{p}^{c,\chi}_{cQ_k} \\ \hat{p}^{c,\gamma}_{cQ_k} \\ \hat{p}^{c,\kappa}_{cQ_k} \end{bmatrix} \Rightarrow \tag{5.21}$$

$$\begin{bmatrix} \hat{u}_{Q_k} \\ \hat{v}_{Q_k} \end{bmatrix} = \begin{bmatrix} \hat{f}_u \hat{p}^{c,\chi}_{cQ_k} (\hat{p}^{c,\kappa}_{cQ_k})^{-1} + \hat{u}_0 \\ \hat{f}_v \hat{p}^{c,\gamma}_{cQ_k} (\hat{p}^{c,\kappa}_{cQ_k})^{-1} + \hat{v}_0 \end{bmatrix} \tag{5.22}$$

Alternatively, if the position of each feature Q_k is known with respect to the world frame, estimations for image coordinates $[u_{Q_k}, v_{Q_k}]^T$ may be computed with outputs $\hat{p}^w_{wc}$ and $\hat{R}^w_c$ from Equations 5.18 and 5.19:

$$p^w_{wQ_k} = \begin{bmatrix} p^{w,x}_{wQ_k} \\ p^{w,y}_{wQ_k} \\ p^{w,z}_{wQ_k} \end{bmatrix} \Rightarrow \hat{p}^c_{cQ_k} = (\hat{R}^w_c)^T(p^e_{eQ_k} - \hat{p}^w_{wc}) = \hat{R}^c_w(p^w_{wQ_k} - \hat{p}^w_{wc}) = \begin{bmatrix} \hat{p}^{c,\chi}_{cQ_k} \\ \hat{p}^{c,\gamma}_{cQ_k} \\ \hat{p}^{c,\kappa}_{cQ_k} \end{bmatrix} \Rightarrow \tag{5.23}$$

$$\begin{bmatrix} \hat{u}_{Q_k} \\ \hat{v}_{Q_k} \end{bmatrix} = \begin{bmatrix} \hat{f}_u \hat{p}^{c,\chi}_{cQ_k} (\hat{p}^{c,\kappa}_{cQ_k})^{-1} + \hat{u}_0 \\ \hat{f}_v \hat{p}^{c,\gamma}_{cQ_k} (\hat{p}^{c,\kappa}_{cQ_k})^{-1} + \hat{v}_0 \end{bmatrix} \tag{5.24}$$

Similar expressions may be defined for points Q_k described with respect to the ship frame or another arbitrary frame. Note that there are multiple navigational assistance equipment from different manufacturers. Thus, different implementations may need to

be developed for each set of navigational assistance equipment. An interesting strategy for developing the *navigational information* block of the system is the investigation of standardized communication protocols between marine equipment. Once general implementations regarding standardized protocols is achieved, the integration with typical navigational assistance equipment is facilitated.

5.4 Render

The *render* block of the system effectively displays navigational information on top of the camera scene. It generates virtual visual elements accordingly to the operator task and the pertinent navigational information. Methods for the generation of such virtual visual elements are proposed in Chapter 8. For any particular ship operation, there are multiple potential combinations of virtual elements that may be helpful. Moreover, different operators performing a same task might find a different set of virtual elements better suited for their task than others. Therefore, it is interesting to analyze and compare potential virtual elements accordingly to standardized navigation operations. It is interesting to note that such usability research may be carried out without the actual development of a fully operational *monitor augmented reality* prototype.

6 EXPERIMENTAL SETUP

This chapter presents the experimental setup used throughout the work. A particular objective of the present work is the development of methods for generating virtual visual elements on navigation scenes. The combination of such virtual assets may assist operators in the perception of the navigation scenario. Note that the usefulness of each augmented visualization depends on the organization of the different virtual assets given a particular navigation context. Thus, it is interesting to further investigate potential arrangements that may be optimal with respect to particular ship operations.

It is possible to perform preliminary investigations about optimal arrangements for virtual elements with videos from navigation experiments. Particularly, the navigation experiments considered in the current work are presented in the following sections. For a given video of a navigation experiment, different augmented visualizations may be generated for further comparison and validation without the necessity of implementing an operational prototype. Methods for generating virtual visual elements in navigation videos are proposed in the following chapters considering a *monitor augmented reality* setup, which are pursued as a means for enabling preliminary research of *augmented reality* in maritime environments.

6.1 Experiment Description

Experiments from the current work consists of a ship navigating through a channel delimited by nautical buoys. In order to keep the ship inside the channel during its travel, it is extremely important that ship operators acknowledge the channel region during the entire path. Moreover, the ship trajectory needs to be executed exactly as planned. Note that both tasks are facilitated when the nautical buoys from the channel are easily perceived by the ship crew. Therefore, an onboard equipment that assist operators in their identification might be helpful for such scenarios. Generally, whenever a ship is navigating through *restricted waters*, it is interesting that the ship crew acknowledge

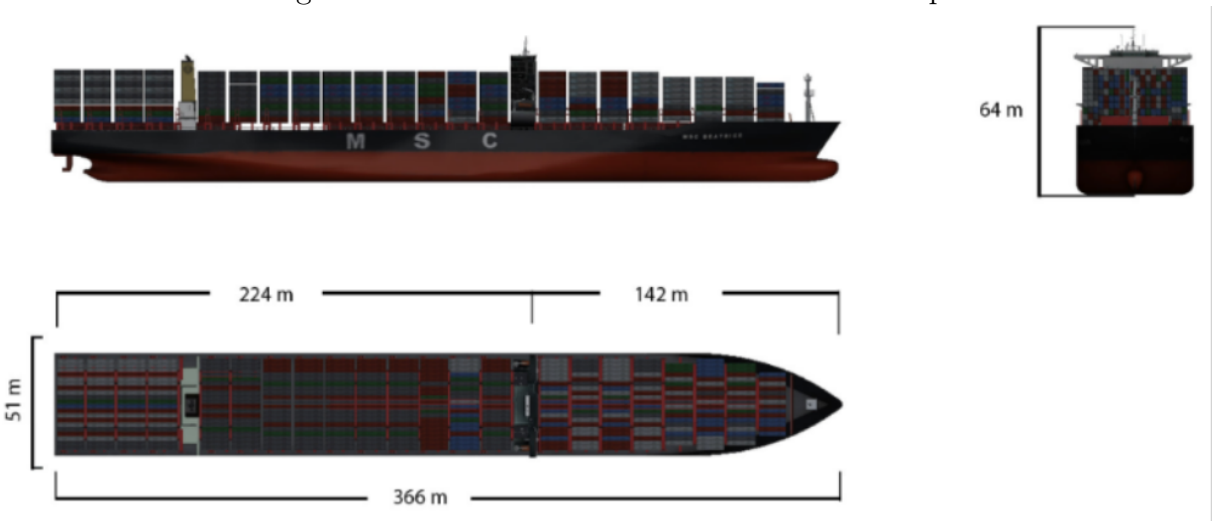
surrounding obstacles at the sea.

In this context, a camera is rigidly attached to the ship measuring the whole experiment. Nautical buoys are used as examples of obstacles to be highlighted in the navigation scene considering a *monitor augmented reality* setup. In order to investigate the development of a *monitor augmented reality* system to assist in the perception of the navigation scene, methods for rendering virtual visual elements in videos from navigation experiments are proposed. The methodology assumes that the ship geometry and trajectory is known alongside the camera position and orientation with respect to the ship frame. These methods may be incorporated into the *render* block of the aforementioned *monitor augmented reality* architecture. Alternatively, it is possible to simply investigate potential augmented visualizations for different scenarios without a fully operational augmented reality equipment.

6.2 Simulated Experiment

Consider a simulation from the TPN-USP ship maneuvering simulator. Figure 14 shows main dimensions of the simulated ship used for the experiments.

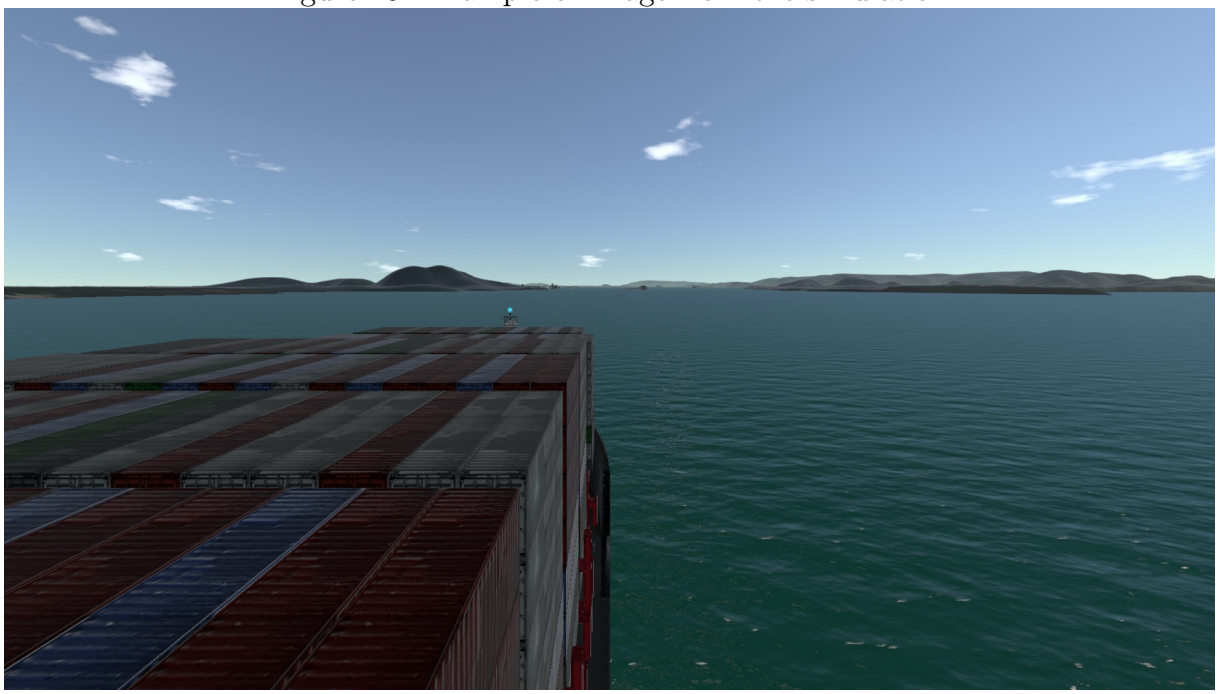
Figure 14: Main dimensions of the simulated ship



Source: Author

A camera is installed in the ship at a location p^s_{sc} and orientation R^s_c. The camera is able to see points from the ship with known three dimensional coordinates. Figure 15 shows an example of the image from the simulated experiment.

Figure 15: Example of image from the simulation



Source: Author

Numerical values for R_c^s and p_{sc}^s are given by:

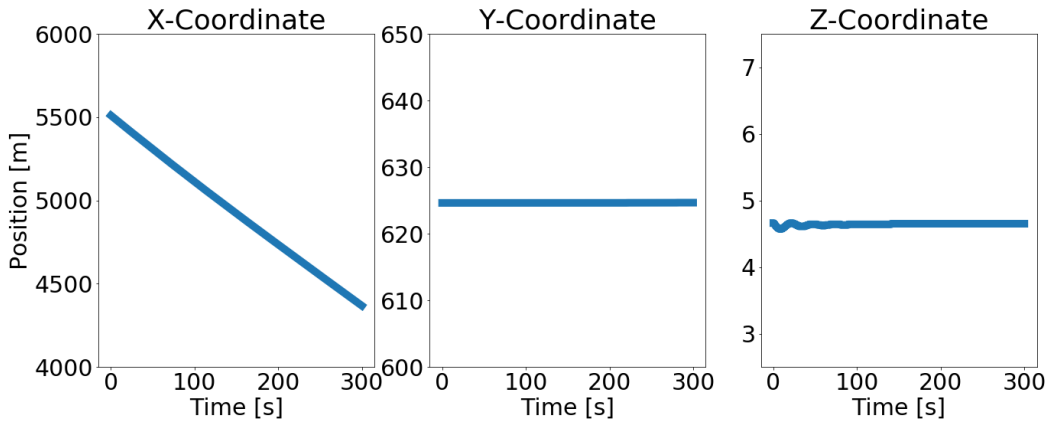
$$R_c^s = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \quad p_{sc}^s = \begin{bmatrix} 42.0 \\ -25.5 \\ 57.51 \end{bmatrix} \tag{6.1}$$

Let d_s be the draft of the ship in the simulation. In the following experiments, the draft is constant and known from the simulation output. An interesting installation parameter that may be computed from the simulation output is the distance h_w between the camera and the water. Considering the $z-$coordinate of the camera position, it is possible to compute the distance h_w between the camera and the water as:

$$h_w = p_{sc}^{s,k} - d_s = 57.51 - 13.3 = 44.2 \tag{6.2}$$

The ship trajectory is simulated with respect to a world reference coordinate system S_W as shown in Figure 16:

Figure 16: Ship position in the world frame as a function of time



Source: Author

6.2.1 Display

Parameters associated with the *display* block of the system are the camera intrinsic and extrinsic parameters. The extrinsic parameters p_{sc}^{s} and R_{c}^{s} are assumed to be known with considerably precision from the simulation outputs (Equation 6.1). In order to estimate the camera intrinsic parameters, consider points belonging to the containers that are visible to the camera as in the example of Figure 15. Let $[u_Q, v_Q]^T$ be the image projection of a vertex from a container. Further, let L_C, H_C and B_C be, respectively, the length, height and the basis size of each container:

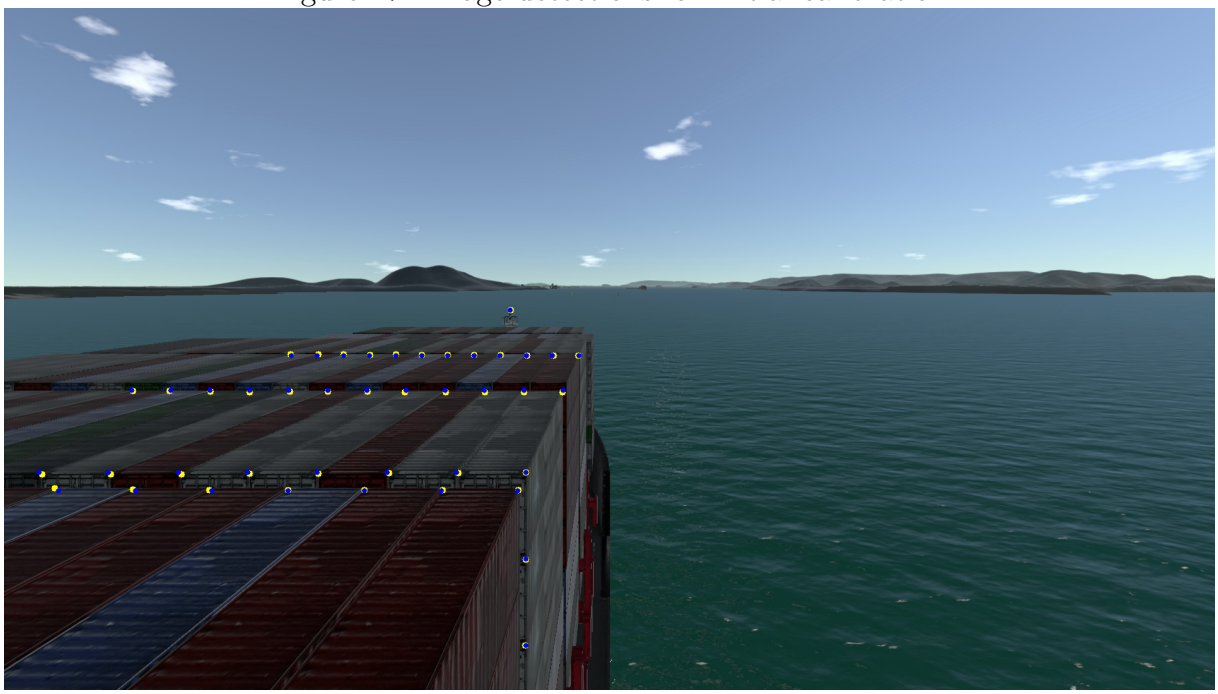
$$L_C = 12 \text{ m} \quad , \quad H_C = 3 \text{ m} \quad , \quad B_C = 2.27 \text{ m} \tag{6.3}$$

It is possible to determine three-dimensional coordinates with respect to the ship frame for each vertex of the containers from dimensions shown in Equation 6.3. As the extrinsic parameters of the camera are known, it is possible to transform these containers points to the camera frame and computing corresponding projections $[\hat{u}_Q, \hat{v}_Q]^T$ with Equation 4.19. Then, it is possible to estimate the camera intrinsic parameters with the minimization of the error between observations $[u_Q, v_Q]^T$ and estimations $[\hat{u}_Q, \hat{v}_Q]^T$. The camera intrinsic parameters obtained from the optimization are:

$$f_u^{OPTM} = 796.32 \qquad f_v^{OPTM} = 748.41 \qquad u_0^{OPTM} = 961.72 \qquad v_0^{OPTM} = 438.60 \tag{6.4}$$

Figure 17 compares the projection of points after optimization (in blue) with image detections (in yellow).

Figure 17: Image detections for initial calibration



Source: Author

Note that there is no onboard monitor for displaying the augmented view during experiments. Instead, the output of each experiment is simply an augmented video from the navigation, which may be used in further investigations of potential designs for such a navigational assistance equipment.

6.2.2 Tracker

Let O_w be the origin of the world coordinate system in which the ship trajectory is simulated. Assume that the origin O_w of the reference coordinate system is expressed by its curvilinear position in the earth frame; given by the following latitude L_w, longitude λ_w and height h_w:

$$L_w = -25.4927988057000^\circ = -0.444933274708117\ rad \tag{6.5}$$

$$\lambda_w = -48.4887732250436^\circ = -0.8462887430298802\ rad \tag{6.6}$$

$$h_w = 0\ m \tag{6.7}$$

$$p^e_{ew} = \begin{bmatrix} 3818020.96154852 & -4313784.14713004 & -2728450.48825468 \end{bmatrix}^T \tag{6.8}$$

Regarding each axis of the reference coordinate system, assume that axes $\vec{x}_w$ and $\vec{y}_w$ are parallel to the surface of the Earth, with $x-$axis pointing to the east and the

$y-$axis pointing to the north; while the $z-$axis is in the opposite direction as the gravity vector. Considering the *local navigation frame* described by axes $[\vec{x}_n, \vec{y}_n, \vec{z}_n]$, let R_w^n be the rotation matrix relating the reference coordinate frame with the local navigation frame. The rotation matrix describing the relation between the reference coordinate system and the earth frame can be computed with Equation 3.58:

$$R_w^n = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \Rightarrow R_w^e = R_n^e R_w^n \tag{6.9}$$

$$R_w^e = \begin{bmatrix} 0.748825870 & 0.285253270 & 0.598239407 \\ 0.662766789 & -0.322292896 & -0.675919723 \\ 0.0 & 0.902639386 & -0.430397652 \end{bmatrix} \tag{6.10}$$

Let p_{ws}^w be the ship position simulated with respect to the world coordinate system. The ship position p_{es}^e and orientation R_s^e with respect to the earth may be computed as:

$$p_{es}^e = p_{ew}^e + R_w^e p_{ws}^w \tag{6.11}$$

$$R_s^e = R_w^e R_s^w \tag{6.12}$$

Let v_{es}^w be the ship velocity with respect to the world coordinate system. Assuming that the world coordinate system is fixed with respect to the earth frame, the ship velocity v_{es}^e with respect to the earth may be computed as:

$$v_{es}^e = R_w^e v_{es}^w \tag{6.13}$$

A potential implementation for the *tracker* block of the system consists of an INS integrated with a GNSS and a compass in a loosely coupled architecture. For example, consider an INS with axes aligned with the ship frame and sampling interval ΔT_{INS}. Further, assume the following measurement model for the accelerometer defined from a simplification of Equation C.5:

$$\hat{f}_{is}^s = f_{is}^s + b_a + \eta_{fa} \tag{6.14}$$

where $\hat{f}_{is}^s$ is the measured specific force from the instrument, f_{is}^s is the real specific force, b_a is the accelerometer bias and η_{fa} is an additive Gaussian random noise with zero mean and covariance w_{fa}. The bias of the accelerometer is assumed to be affected by an additive Gaussian random noise with zero mean and covariance w_{ba}. Similarly, let $\hat{\omega}_{is}^s$ be the measured angular velocity and let ω_{is}^s be the truth angular velocity. From Equation

C.6:

$$\hat{\omega}_{is}^{s} = \omega_{is}^{s} + b_g + \eta_{\omega g} \tag{6.15}$$

where b_g is the gyrometer bias and $\eta_{\omega g}$ is an additive Gaussian random noise with zero mean and covariance $w_{\omega g}$. The bias of the gyrometer is also assumed to be affected by an additive Gaussian random noise with zero mean and covariance w_{bg}. Consider a vector Π_{INS} concatenating all error terms from the INS:

$$\Pi_{INS} = \left[w_{fa}, b_a, w_{ba}, w_{\omega g}, b_g, w_{bg}\right] \tag{6.16}$$

Further, consider a circular orbit with N_{sat} regularly distributed satellites simulating a GNSS constellation and a GNSS user antenna with a sampling interval ΔT_{GNSS}. The position of the antenna with respect to the ship is p_{sa}^{s}. Each measurement is modeled with Equations C.36 and C.45:

$$\rho_j = \bar{\rho}_j + \delta\rho_{rc} + \eta_{\rho p} \tag{6.17}$$

$$\dot{\rho}_j = \bar{\dot{\rho}}_j + \delta\dot{\rho}_{rc} + \eta_{\rho v} \tag{6.18}$$

where $\bar{\rho}_j$ is given by Equation C.33, $\bar{\dot{\rho}}_j$ is given by Equation C.46, $\eta_{\rho p}$ is an additive Gaussian random noise with zero mean and covariance $w_{\rho p}$ and $\eta_{\rho v}$ is an additive Gaussian random noise with zero mean and covariance $w_{\rho v}$. Parameters $\delta\rho_{rc}$ and $\delta\dot{\rho}_{rc}$ are related with offsets and drifts from the receiver clock. At each iteration, estimations $\hat{p}_{ea}^{e}$ and $\hat{v}_{ea}^{e}$ for the position and velocity of the user antenna with respect to the earth are computed with Equations C.41 and C.47. Similarly as before, consider the vector Π_{GNSS} concatenating these parameters from the GNSS equipment:

$$\Pi_{GNSS} = \left[p_{sa}^{s}, \delta\rho_{rc}, w_{\rho p}, \delta\dot{\rho}_{rc}, w_{\rho v}\right] \tag{6.19}$$

For the compass, consider the simulation of an instrument coincident with the INS position and aligned with the ship axes. Assume that measurements are provided at a sampling rate ΔT_{CMPS} accordingly to Equation C.55:

$$\hat{\psi}_{n\psi} = \psi_{n\psi} + \eta_{\psi} \tag{6.20}$$

where $\psi_{n\psi}$ is computed from Equation C.54 and η_{ψ} is an additive Gaussian noise with zero mean and nonzero covariance w_{ψ}. The vector Π_{CMPS} concatenates the parameters

from the compass model:

$$\Pi_{CMPS} = \left[p^s_{s\psi}, R^s_{\psi}, w_{\psi} \right] \tag{6.21}$$

The accuracy of the *tracker* depends on the errors from each sensor used in the implementation. Recall that in the *propagation* step from the proposed *loosely-coupled* integration filter depends on system matrices F_S, G_S and Q_β. The system deterministic matrix F_S is defined with Equation C.59. The system noise matrix Q_β is defined with Equation C.64 as a function of estimates for INS error parameters $[\hat{w}_{\omega g}, \hat{w}_{fa}, \hat{w}_{ba}, \hat{w}_{bg}]$. The *update* step depends on the measurement matrices H_Z and R_Z. The measurement deterministic matrix H_Z^{GNSS} for the GNSS is given by Equation C.68 and the measurement deterministic matrix H_Z^{CMPS} for the compass is given by Equation C.90. In the case of the GNSS, R_Z^{GNSS} is defined with Equation C.69 as a function of estimates $[\hat{w}_{\rho p}, \hat{w}_{\rho v}]$ for GNSS error parameters. For the compass, R_Z^{CMPS} is defined with Equation C.91 as a function of estimates for compass error parameter $[\hat{w}_{\psi}]$. Let Π_{LCI} be a vector concatenating all parameters from the *loosely coupled* integrator:

$$\Pi_{LCI} = \left[\hat{w}_{\omega g}, \hat{w}_{fa}, \hat{w}_{ba}, \hat{w}_{bg}, \hat{p}^s_{sa}, \hat{w}_{\rho p}, \hat{w}_{\rho v}, \hat{p}^s_{s\psi}, \hat{R}^s_{\psi}, \hat{w}_{\psi} \right] \tag{6.22}$$

Note that simulations with different sensor models may assist in the definition of precision requisites for each equipment from the tracker. Consider a simulation comparing the *tracker* performance from two different setups represented by sets of parameters $\Pi^1_{\dots}$ and $\Pi^2_{\dots}$.

For the inertial navigation system, which computes the reference solution for the *loosely coupled* integration, consider the following sets of parameters Π^1_{INS} and Π^2_{INS}:

$$\Pi^1_{INS} = \left[w_{fa} = 4 \cdot 10^{-5} \quad , \quad b_a = \begin{bmatrix} 4 \cdot 10^{-4} \\ 4 \cdot 10^{-4} \\ 4 \cdot 10^{-4} \end{bmatrix} \quad , \quad w_{ba} = 1 \cdot 10^{-5} \right. \tag{6.23}$$

$$\left. w_{\omega g} = 10^{-6} \quad , \quad b_g = \begin{bmatrix} 3 \cdot 10^{-5} \\ 3 \cdot 10^{-5} \\ 3 \cdot 10^{-5} \end{bmatrix} \quad , \quad w_{bg} = 10^{-7} \right] \tag{6.24}$$

$$\Pi^2_{INS} = \Big[w_{fa} = 4 \cdot 10^{-3} \quad , \quad b_a = \begin{bmatrix} 4 \cdot 10^{-2} \\ 4 \cdot 10^{-2} \\ 4 \cdot 10^{-2} \end{bmatrix} \quad , \quad w_{ba} = 1 \cdot 10^{-4} \tag{6.25}$$

$$w_{\omega g} = 10^{-4} \quad , \quad b_g = \begin{bmatrix} 3 \cdot 10^{-3} \\ 3 \cdot 10^{-3} \\ 3 \cdot 10^{-3} \end{bmatrix} \quad , \quad w_{bg} = 10^{-5} \Big] \tag{6.26}$$

The integrator of both *inertial navigation systems* are initialized from simulation outputs without initial errors.

For the GNSS equipment, consider the following sets of parameters Π^1_{GNSS} and Π^2_{GNSS}:

$$\Pi^1_{GNSS} = \Big[p^s_{sa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \delta\rho_{rc} = 0 \quad , \quad w_{\rho p} = 1 \quad , \quad \delta\dot{\rho}_{rc} = 0 \quad , \quad w_{\rho v} = 0.1 \Big] \tag{6.27}$$

$$\Pi^2_{GNSS} = \Big[p^s_{sa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \delta\rho_{rc} = 0 \quad , \quad w_{\rho p} = 10.0 \quad , \quad \delta\dot{\rho}_{rc} = 0 \quad , \quad w_{\rho v} = 1.0 \Big] \tag{6.28}$$

At each measurement from the GNSS user antenna, the antenna position and velocity with respect to the earth are computed from Equations C.41 and C.47 with initial estimates from the INS reference solution. Note that it is assumed that the user antenna position is coincident with the inertial navigation system, which considerably simplifies aforementioned GNSS models.

For the compass, consider the following sets of parameters Π^1_{GNSS} and Π^2_{GNSS}:

$$\Pi^1_{CMPS} = \Big[p^s_{s\psi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad R^s_{\psi} = I_{3\times 3} \quad , \quad w_{\psi} = 0.017453292519943295 \Big] \tag{6.29}$$

$$\Pi^2_{CMPS} = \Big[p^s_{s\psi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad R^s_{\psi} = I_{3\times 3} \quad , \quad w_{\psi} = 0.08726646259971647 \Big] \tag{6.30}$$

Similarly, note that it is assumed that the compass position and orientation are equal to the ship position and orientation.

Finally, consider the following sets of parameters Π^1_{LCI} and Π^2_{LCI} for the *loosely coupled* integrator:

$$\Pi^1_{LCI} = \Big[\hat{w}_{\omega g} = 10^{-6} \;\;,\;\; \hat{w}_{fa} = 4 \cdot 10^{-5} \;\;,\;\; \hat{w}_{ba} = 10^{-5} \;\;,\;\; \hat{w}_{bg} = 10^{-7} \tag{6.31}$$

$$\hat{p}^s_{sa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \;\;,\;\; \hat{w}_{\rho p} = 1.0 \;\;,\;\; \hat{w}_{\rho v} = 0.1 \;\;,\;\; \hat{p}^s_{s\psi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.32}$$

$$\hat{R}^s_{\psi} = I_{3\times 3} \;\;,\;\; \hat{w}_{\psi} = 0.017453292519943295\Big] \tag{6.33}$$

$$\Pi^2_{LCI} = \Big[\hat{w}_{\omega g} = 10^{-4} \;\;,\;\; \hat{w}_{fa} = 4 \cdot 10^{-3} \;\;,\;\; \hat{w}_{ba} = 10^{-4} \;\;,\;\; \hat{w}_{bg} = 10^{-5} \tag{6.34}$$

$$\hat{p}^s_{sa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \;\;,\;\; \hat{w}_{\rho p} = 10.0 \;\;,\;\; \hat{w}_{\rho v} = 1.0 \;\;,\;\; \hat{p}^s_{s\psi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.35}$$

$$\hat{R}^s_{\psi} = I_{3\times 3} \;\;,\;\; \hat{w}_{\psi} = 0.08726646259971647\Big] \tag{6.36}$$

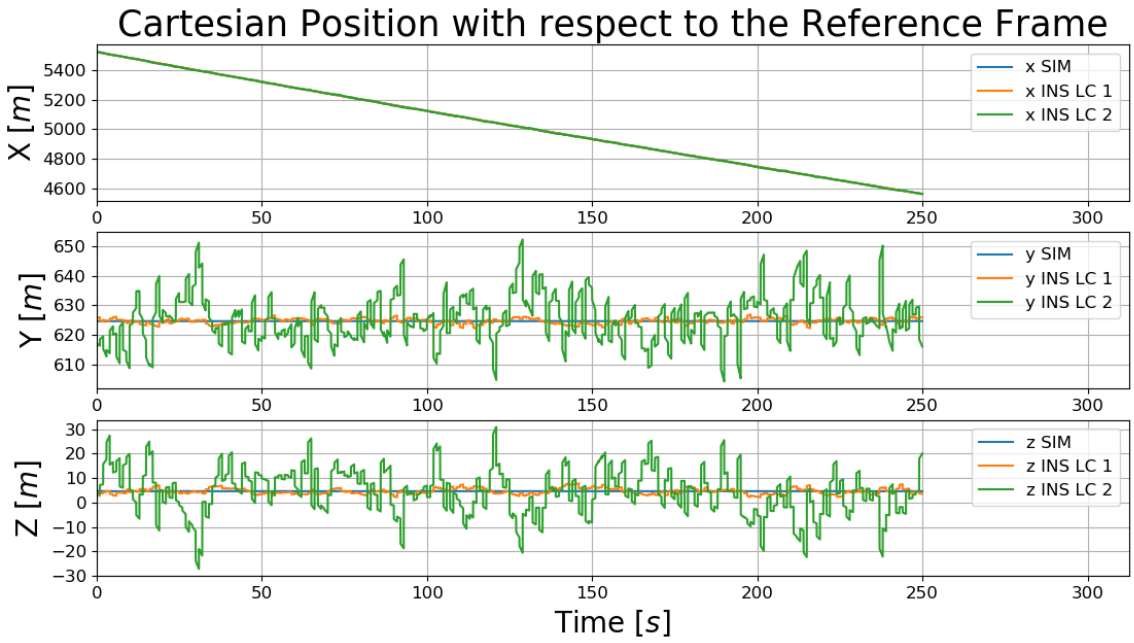
Both filters are initialized with zero mean and the same covariance matrix P_{S_0} accordingly to Equation C.58:

$$P_{S_0} = \begin{bmatrix} \delta\hat{\Psi} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \delta\hat{v} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & \delta\hat{p} I_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & \delta\hat{b_a} I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & \delta\hat{b_g} I_{3\times 3} \end{bmatrix} \tag{6.37}$$

$$\delta\hat{\Psi} = 0.017453 \;,\; \delta\hat{v} = 1 \;,\; \delta\hat{p} = 10 \;,\; \delta\hat{b_a} = 0.00098 \;,\; \delta\hat{b_g} = 0.0004848 \tag{6.38}$$

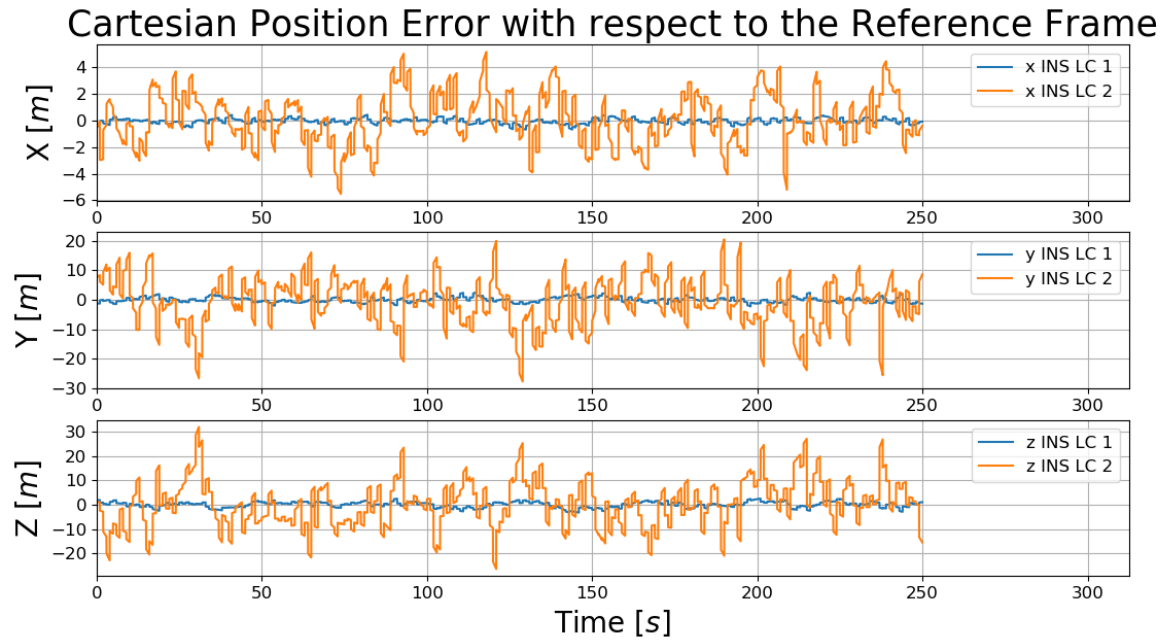
Note that parameters from the sets $\Pi^1_{...}$ represents an equipment setup with higher performance than parameters from the sets $\Pi^2_{...}$. Figures 18, 20 and 22 presents *tracker* outputs computed from parameters $\Pi^1_{...}$ (orange) and $\Pi^2_{...}$ (green) with blue lines representing the simulation outputs. Figures 19, 21 and 23 shows the errors between *tracker* outputs and simulation outputs. Blue lines represents errors from parameters $\Pi^1_{...}$ while orange lines represents parameters from $\Pi^2_{...}$.

Figure 18: Cartesian Position Computed from Tracker with Different Parameters



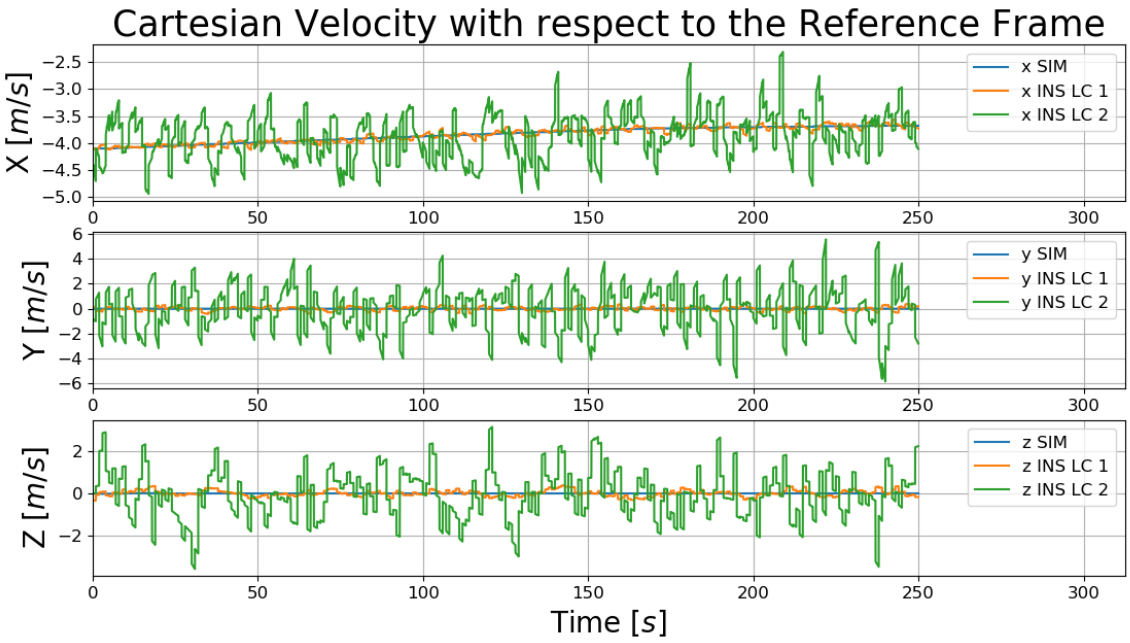
Source: Author

Figure 19: Cartesian Position Error Computed from Tracker with Different Parameters



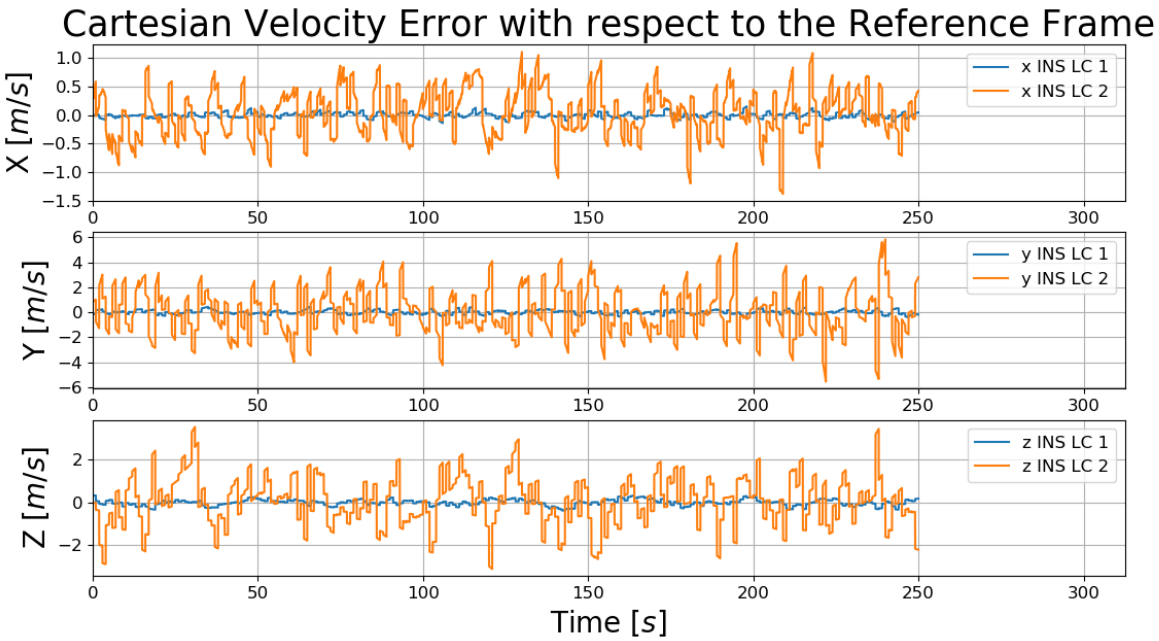
Source: Author

Figure 20: Cartesian Velocity Computed from Tracker with Different Parameters



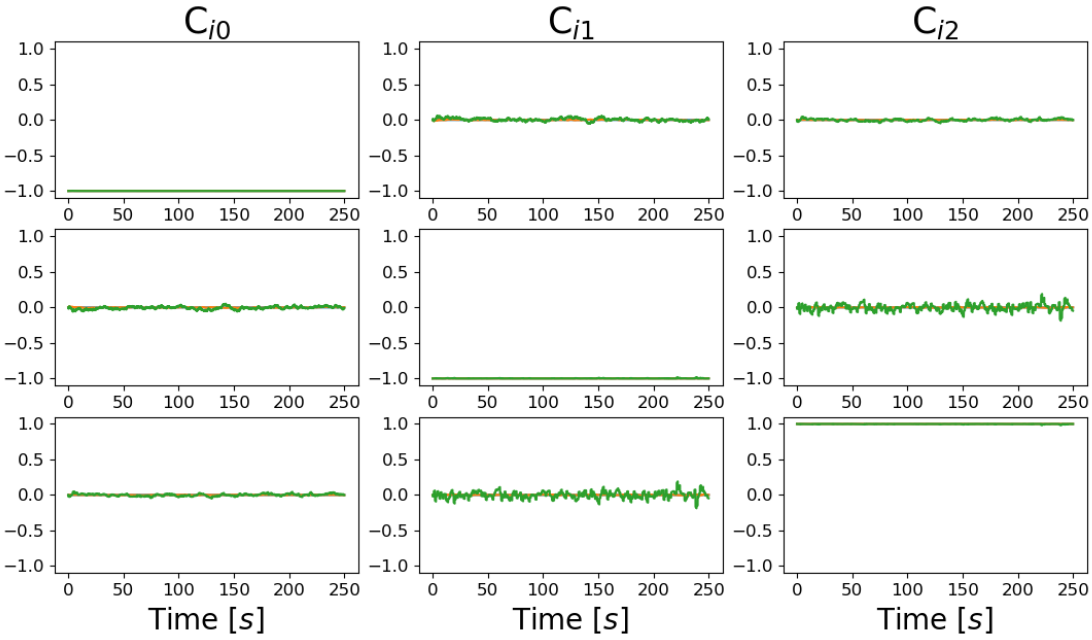
Source: Author

Figure 21: Cartesian Velocity Error Computed from Tracker with Different Parameters



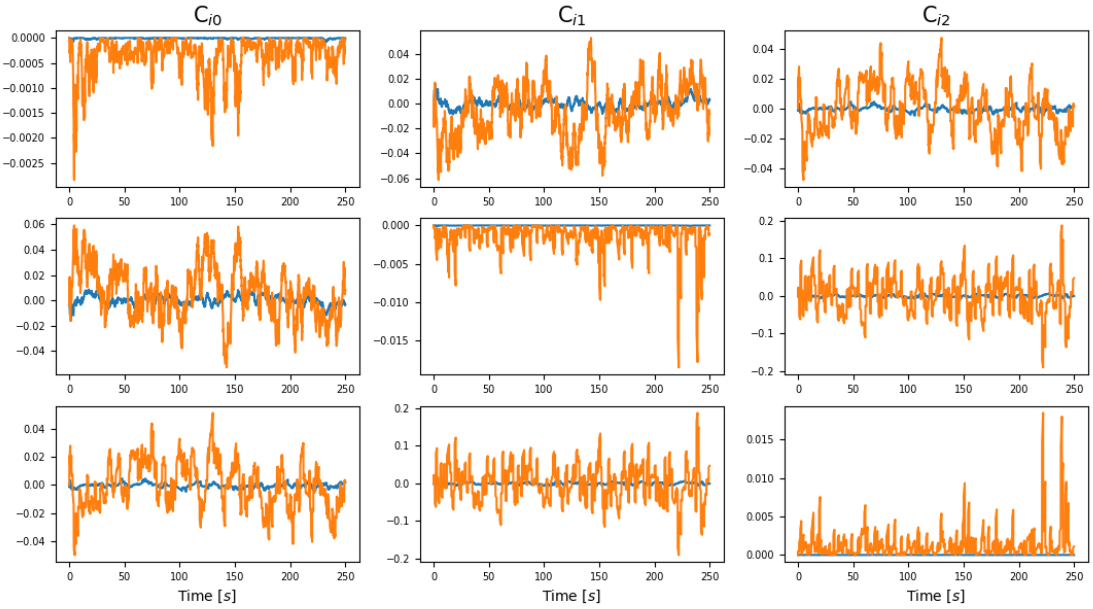
Source: Author

Figure 22: Attitude Computed from Tracker with Different Parameters



Source: Author

Figure 23: Attitude Error Computed from Tracker with Different Parameters



Source: Author

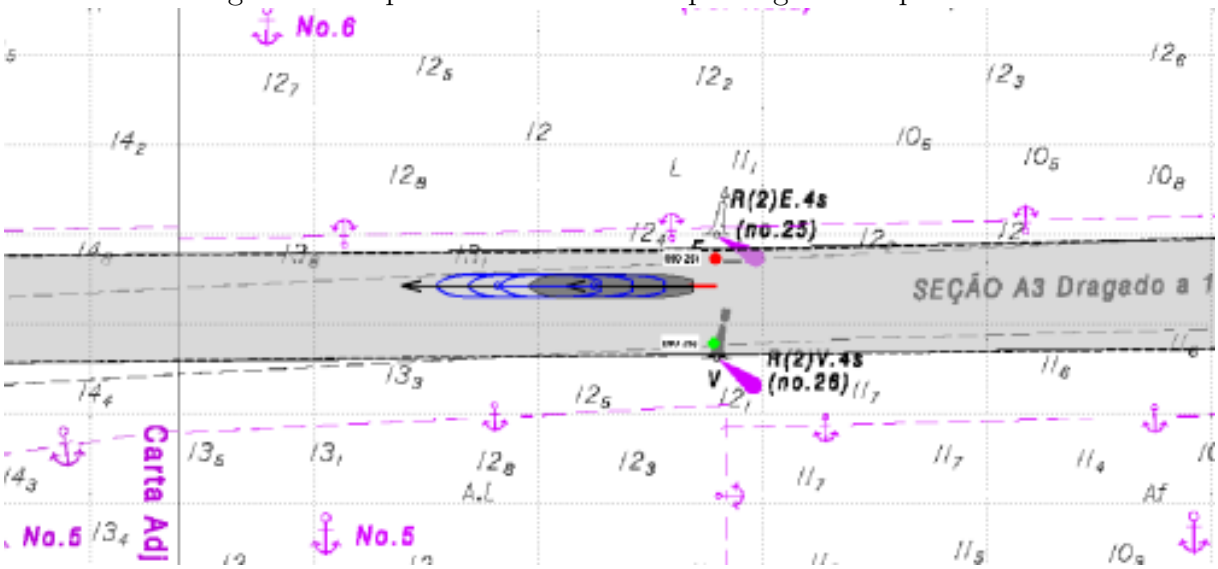
The performance of the *tracker* with the first set of parameters is considerably better than the performance of the *tracker* with the second set of parameters. Accordingly, sensors represented by the first set of parameters are more accurate than sensors from the second set of parameters. In order to define minimum requisites for sensors to be used in the *tracker* implementation, it is possible to repeat the simulations from this section comparing the filter performance with different sensor models. Note that a theoretical analysis describing how errors in each sensor affects the *tracker* performance is beyond the scope of the current work.

Further investigations regarding alternative *tracker* solutions are suggested as it is an extremely important part of the proposed *monitor augmented reality* system. Note that a simple *tracker* implementation in simulation experiments may be obtained by parsing the ship position and orientation from the simulation output at all times. In such implementation, there is no error between the truth and the estimated state of the ship. Thus, all virtual elements rendered on top of the navigation scene are possible to be adequately rendered. This scenario is particularly useful for designing optimal representations with respect to standardized ship operations.

6.2.3 Navigational Information

The ship navigates through a channel delimited by buoys. Figure 24 illustrates the navigation experiment with a superior view of the ship and its surroundings.

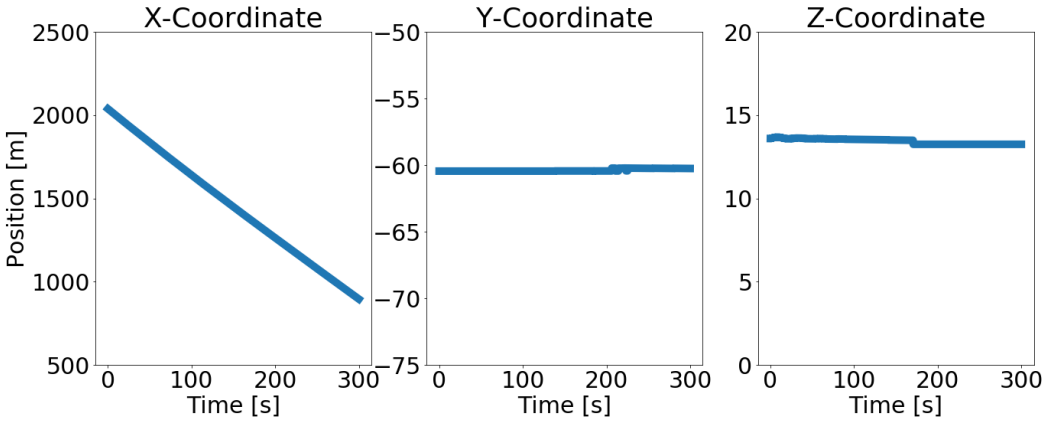
Figure 24: Superior view for the ship navigation experiment



Source: Author

The buoys are assumed to be fixed in the ground with known coordinates in the world frame. Thus, it is possible to compute the instantaneous relative position of each buoy with respect to the SCS. For each sample from the simulation output, the corresponding coordinates in the WCS was computed from the instantenous state of the ship and saved separately. Figure 25 presents the relative position of the right (red) buoy in the ship coordinate system as a function of time.

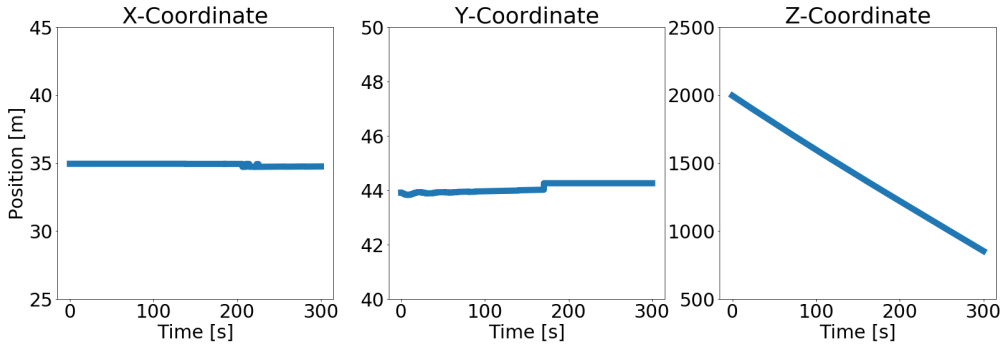
Figure 25: Right buoy position in the ship frame as a function of time



Source: Author

As the camera installation parameters are known from the simulation output, the correspondent coordinates of each buoy can be projected to the camera frame for all samples. Consider the observation by the right camera of the red buoy which is located at the right of the ship. Figure 26 presents the correspondent coordinates.

Figure 26: Right buoy position in the right camera frame as a function of time

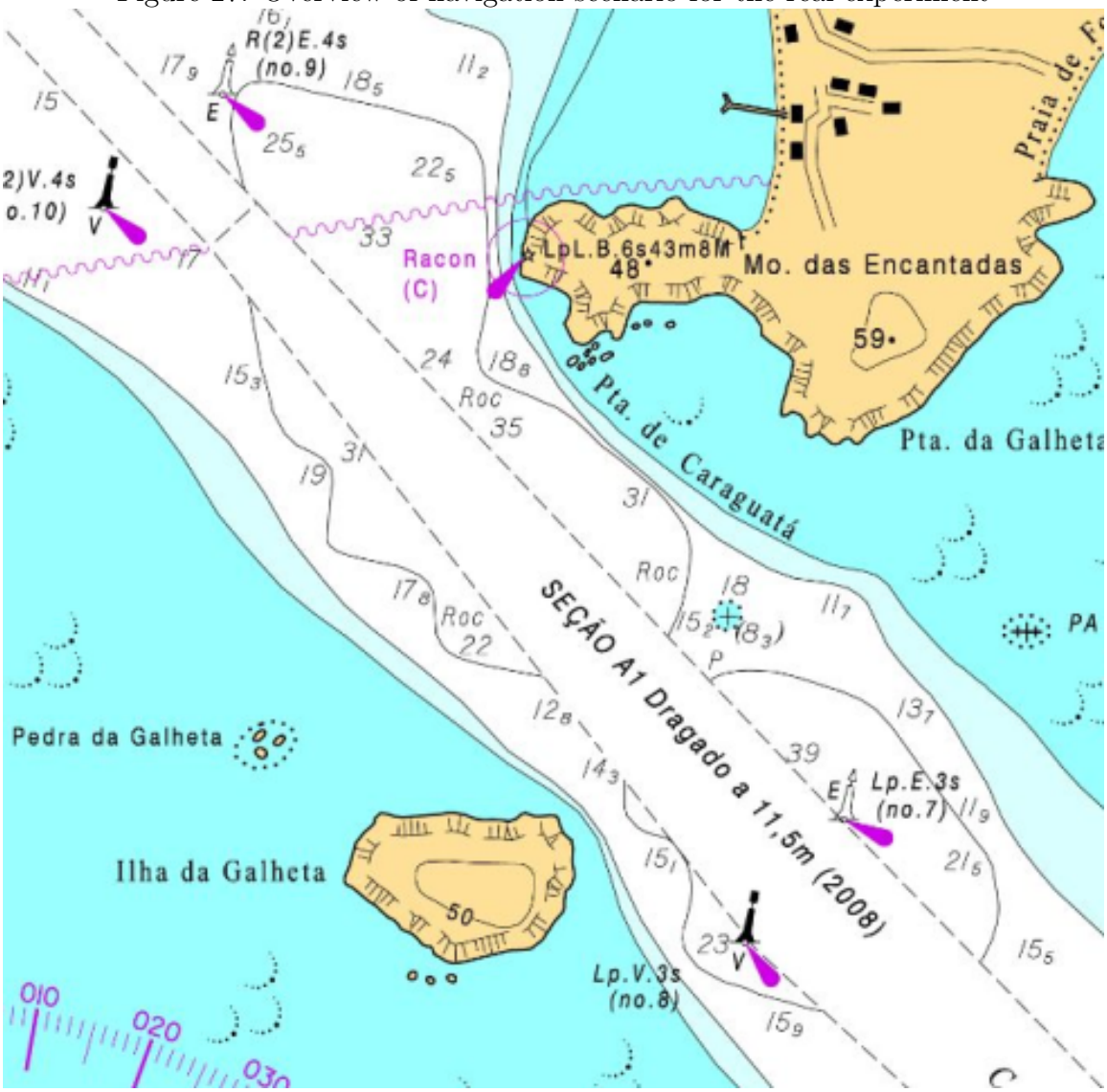


Source: Author

6.3 Real Experiment

Similarly as in the simulated experiment, real experiments from the current work consists of a ship navigating through a channel delimited by nautical buoys. Particularly, these experiments were recorded on a nautical channel around the *Paranagua port*. Figure 27 represents a part of the channel in which the ship navigates.

Figure 27: Overview of navigation scenario for the real experiment



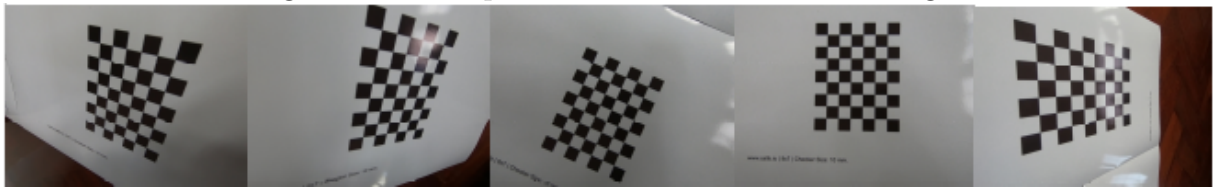
Source: Author

A commercial camera (*GoPro Black Hero 6*) is rigidly attached to the ship recording the entire experiment.

6.3.1 Display

Initially, the intrinsic parameters of the camera can be independently optimized with calibration patterns. Chessboard patterns with different sizes were measured by the camera in a digital video. Frames from the video were selected to be used as input images in the following intrinsic calibration. Figure 28 presents examples of such frames.

Figure 28: Examples of Intrinsic Calibration Images



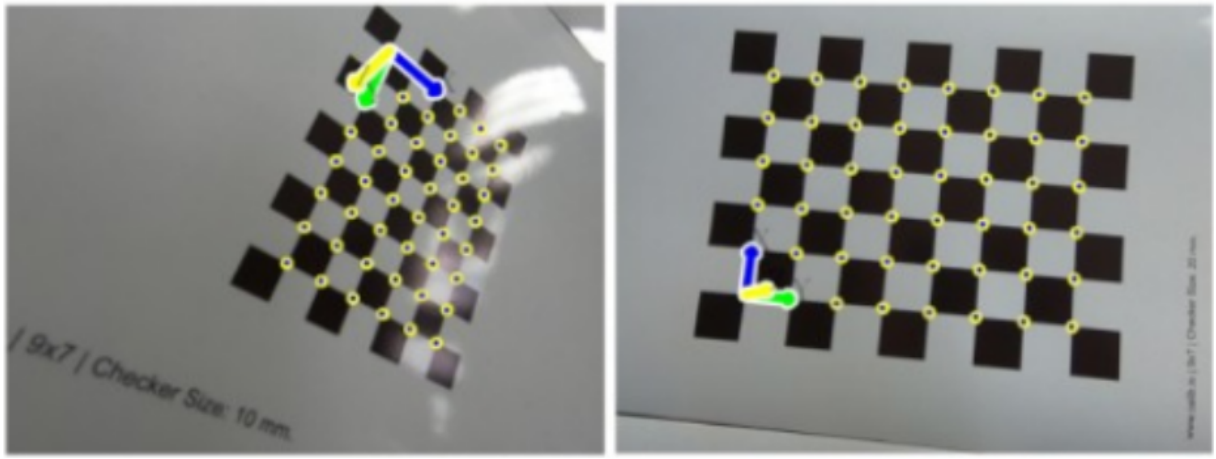
Source: Author

The size of each square from the chessboard is known, thus it is possible to define three-dimensional coordinates for each square intersection assuming an arbitrary point as the origin for the coordinate system. The implementation *calibrateCamera* available in the OpenCV library yields the following values for the intrinsic parameters:

$$f_u = 1044.4 \quad , \quad f_v = 1036.2 \quad , \quad u_0 = 961.9 \quad , \quad v_0 = 718.4$$

The intrinsic parameters may be validated by comparing image detections with correspondent projections of each annotated image. Figure 29 illustrates such comparison for the validation of the intrinsic calibration.

Figure 29: Validation of Intrinsic Calibration



Source: Author

After the intrinsic calibration the camera can be installed onboard of a ship. The

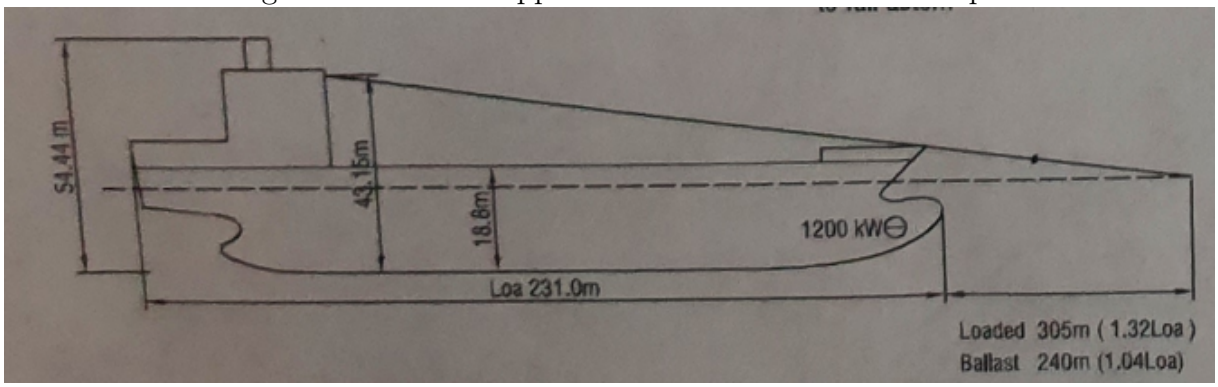
camera position and orientation with respect to the ship frame must be known. Thus, it is interesting to install the camera in a known location from the ship such as in its cabin as illustrated by Figure 30:

Figure 30: Installation of the camera at the ship cabin

Source: Author

Note that the axes of the camera are approximately aligned with the ship frame. Thus, neglecting installation errors, the camera orientation R_c^s may be assumed to be aligned with the ship frame. The camera position with respect to the ship frame may be estimated with geometrical information from the ship. Figure 31 presents ship main dimensions.

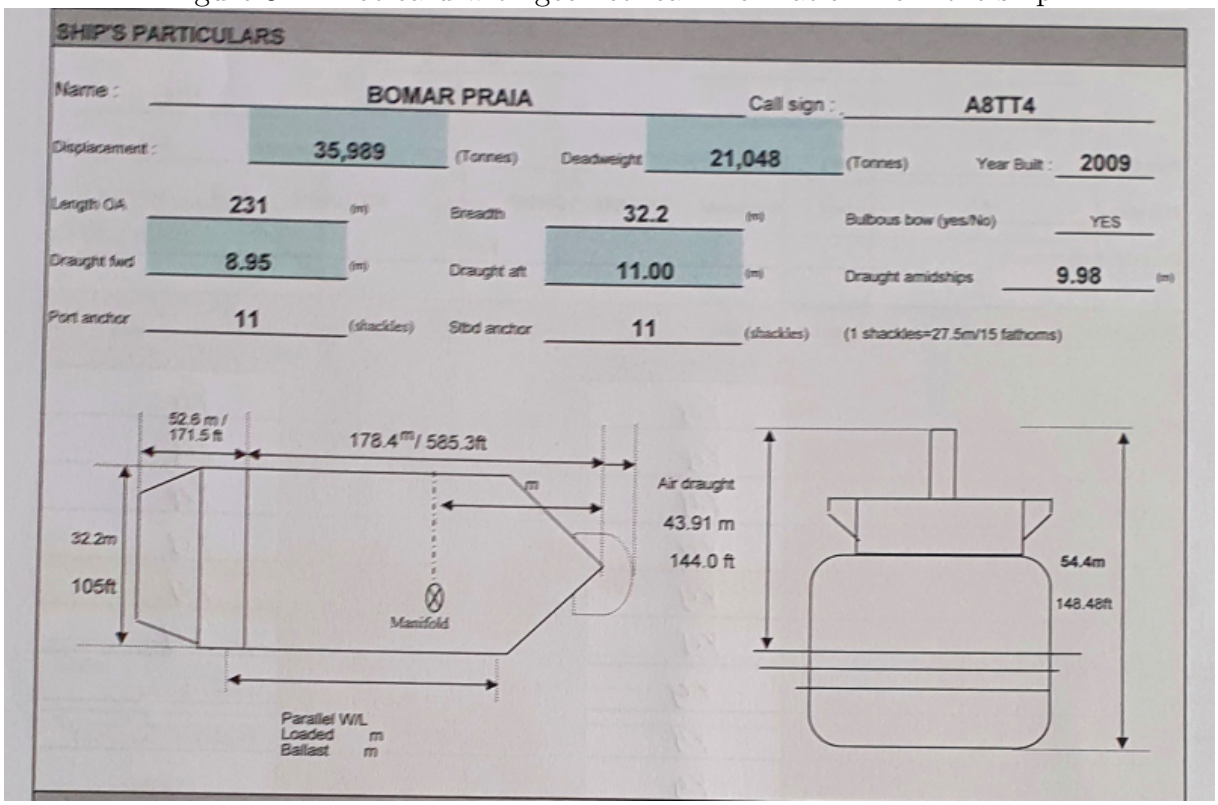
Figure 31: Vertical approximate dimensions of the ship



Source: Author

Thus, the $z-$coordinate of the camera position can be initially estimated as $p_{sc}^{s,k} \approx 43.15$. Further, considering the camera installation position on the window of the ship cabin, the camera position may be estimated as $p_{sc}^{s,k} \approx 43.15 - 0.35$. Considering that the cabin is positioned at the center of the ship, the $y-$coordinate of the camera position may be estimated as zero. Finally, the $x-$ coordinate of the camera position may be estimated from the pilot card of the ship as shown in Figure 32.

Figure 32: Pilot card with geometrical information from the ship



Source: Author

Therefore, the camera position and orientation with respect to the ship may be approximated by:

$$p_{sc}^{s} \approx \begin{bmatrix} 52.6 \\ 0.0 \\ 42.8 \end{bmatrix} \quad , \quad R_{c}^{s} \approx \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{6.39}$$

Recall that the distance between the camera and the water may be estimated from the ship draft and the camera $z-$coordinate position. Let d_s be the draft at the camera section of the ship. Assuming that the draft at the camera position is a linear combination between the aft draft d^A and the forward draft d^F, d_s may be computed as a function of

the length L_S of the ship and the camera position $p_{sc}^{s,i}$.

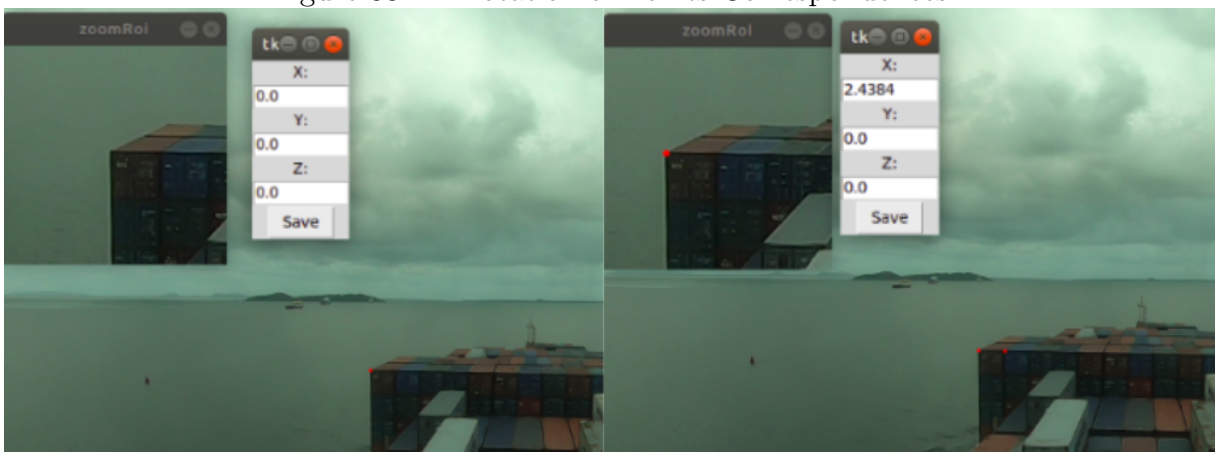
$$d_s = d^A + \frac{d^F - d^A}{L_S}(p_{sc}^{s,i}) = 11.0 + \frac{8.95 - 11.0}{231.0}(52.6) = 10.53 \tag{6.40}$$

Thus, the distance h_w between the camera and the water can be estimated:

$$h_w = (43.15 - 0.35) - (10.53) = 32.26 \tag{6.41}$$

Generally, it is possible to estimate the camera extrinsic parameters ship with points correspondences between observations and three-dimensional positions. In the experiment, note that there are visible containers in the image scene. As the dimensions of each visible container are known and standardized, it is possible to determine points correspondences between camera observations and three-dimensional positions in order to estimate extrinsic parameters for the camera after installation. Consider different *container* frames associated with each block of containers from the camera scene. Further, assume that the axes from the *container* frame are aligned with the ship frame. Each point correspondence may be defined manually or with the assistance of an auxiliary application as shown in Figure 33.

Figure 33: Annotation of Points Correspondences

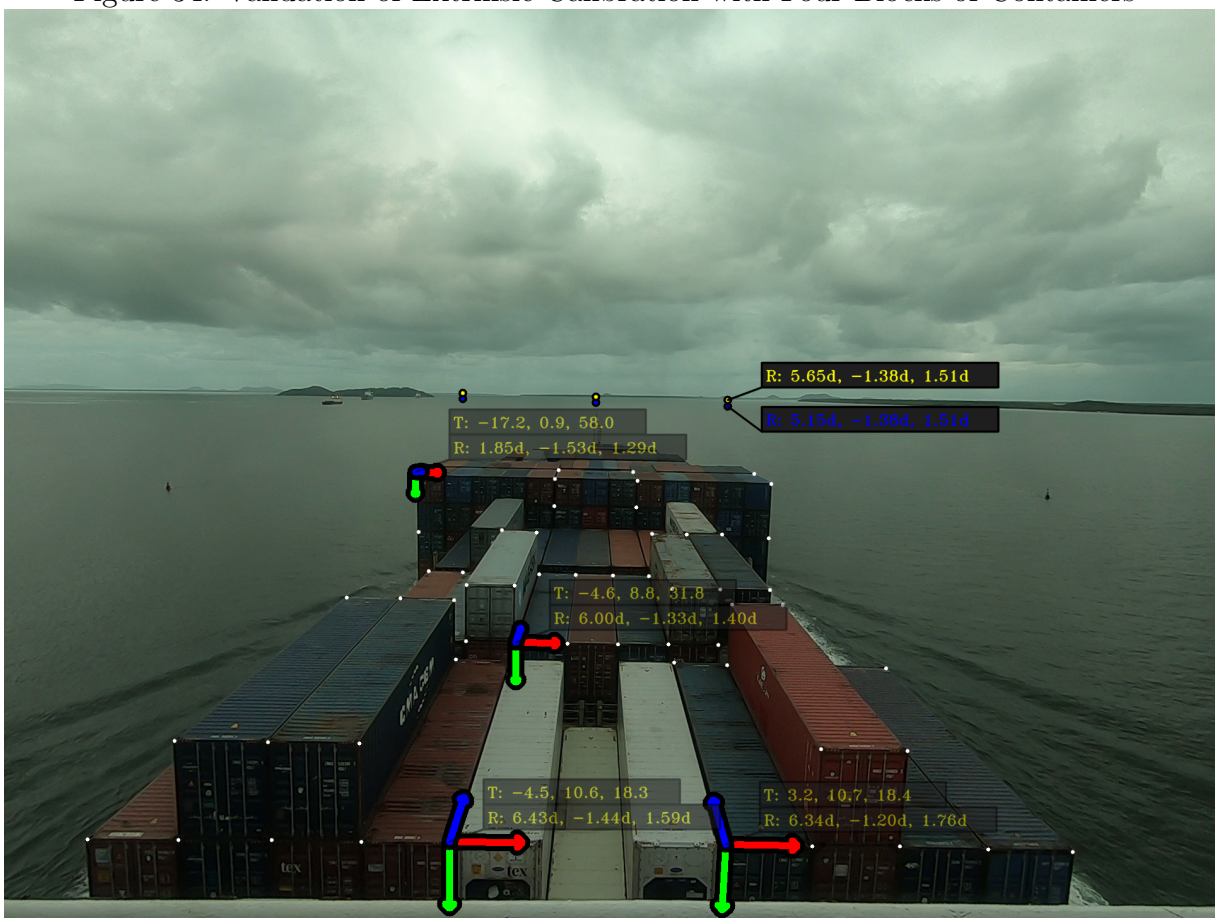


Source: Author

Four blocks of containers were defined, each one with a different *container* frame. For each block of containers with at least four points correspondences, it is possible to independently determine estimations for the camera position and orientation with respect to each *container* frame. An implementation for optimizing extrinsic parameters from points correspondences is available in the function *solvePnP* from the OpenCV library. Similarly as in the previous implementations, it is possible to validate the calibration

results by the comparison of image detections with the correspondent projection of each point as illustrated by Figure 34:

Figure 34: Validation of Extrinsic Calibration with Four Blocks of Containers



Source: Author

The camera orientation with respect to the SCS may be approximated by the mean orientation between all blocks of containers.

6.3.2 Tracker

No tracker solution were considered for the real experiment. Instead, it is assumed that the ship is navigating in a straight route without environment perturbations. Note that in order to implement a *tracker* solution based on the aforementioned loosely-coupled integration with an *inertial navigation system* aided by additional sensors, the required accurateness and robustness for all equipment may considerably increase the costs of the implementation. Although such *tracker* implementation needs to be addressed for general navigation and real-time assistance, it is possible to further research potential *monitor*

augmented reality designs with controlled experiments in which the ship state can be assumed to be known.

6.3.3 Navigational Information

The geometry of the ship is assumed to be known from information such as the pilot card and wheelhouse poster. This geometrical information may be used for displaying the expected area that will be occupied by the ship during its course. Furthermore, the route of the ship is assumed to be known from onboard equipment such as *Portable Pilot Unit* (PPUs). Examples of outputs from a PPU are shown in Figure 35.

Figure 35: Outputs from onboard equipment providing navigational information

Source: Author

Considering a navigation scenario in which the ship travels with approximately constant velocity, most of the navigational data does not change throughout the trajectory. This situation is particularly explored as it considerably simplifies the system implementation. Note that an automatic integration of such equipment with the proposed *monitor augmented reality* system is beyond the scope of the present work.

7 VISUAL ESTIMATION

This chapter presents methods related with the estimation of navigational information from videos recorded by a fixed onboard camera. These methods are potentially helpful for scenarios in which there are no additional information from other onboard equipment. As the proposed MAR architecture uses a fixed camera for recording the navigation scene, the proposed equipment enables the research of computer vision in scenarios of maritime navigation.

The first section discusses a potential application to assist in the perception of obstacles at sea. Next, a Kalman filter implementation for estimating the relative position from camera measurements is presented. Finally, the last section discusses the incorporation of camera measurements in the aforementioned *tracker* setup in a loosely coupled integration. Note that there are several factors that affects visibility conditions regarding the navigation scene. Thus, further research is needed before an incorporation of such methods into operational equipment.

7.1 Object Detector

Consider the scenario of an operator identifying an obstacle in the scene from an onboard camera. The obstacle region can be highlighted in the original image of the scene to assist the rest of the ship crew in the corresponding identification. Moreover, object detection algorithms can be used for further automatic tracking of the obstacle. Such feature can be embedded in an operational tool for the enhancement of perception or used for further researching computer vision in maritime environments.

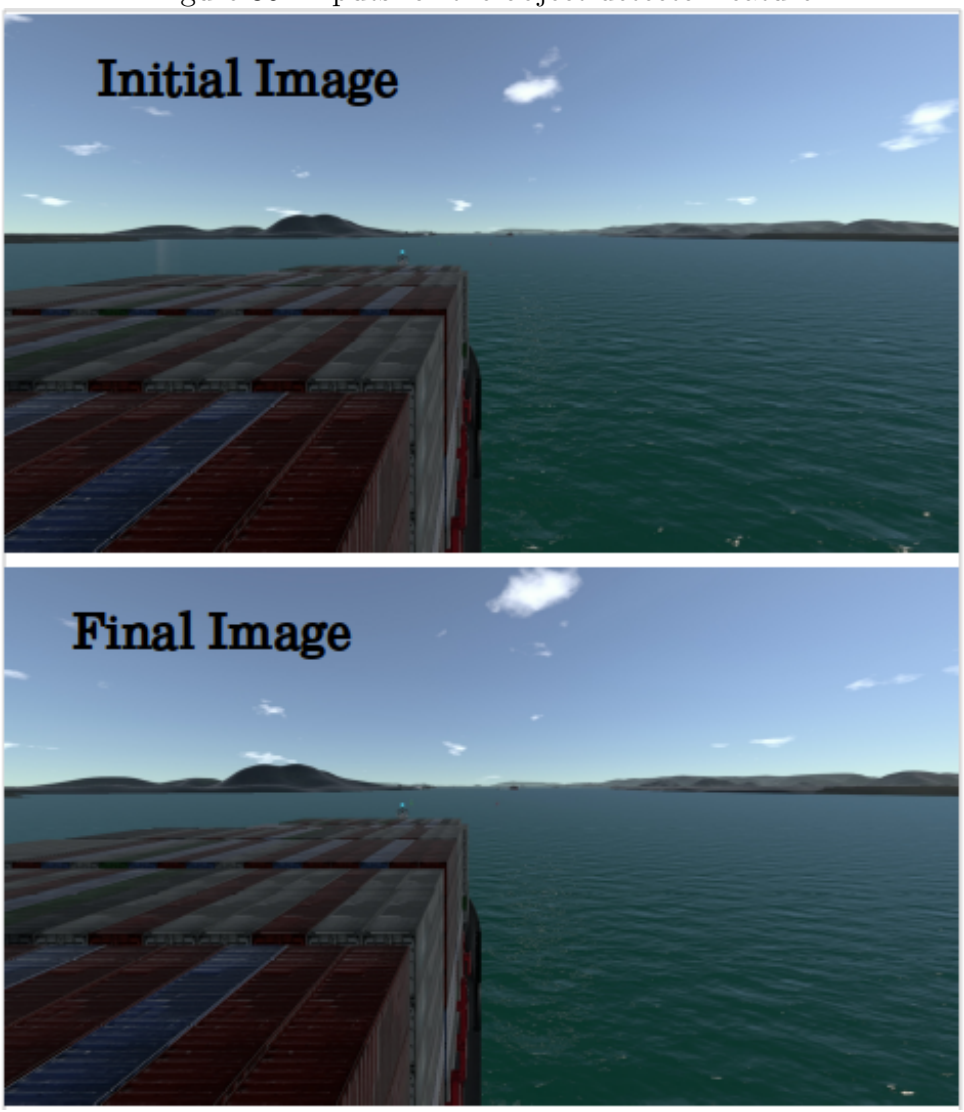
A potential implementation based on template matching and image segmentation may be summarized as follows. The user initially identifies the region of the obstacle in the image scene defining an initial template and a *region of interest* (**ROI**) around it. For each following iteration, a template matching algorithm is computed in the ROI. The output of the matching is further processed with an image segmentation algorithm. If both outputs

from consecutive iterations are valid and do not vary significantly, the input template is updated with the outputs from the most recent ROI. The rest of this section exemplifies this potential implementation with videos from the experimental setup described in the previous chapter. Note that there are different computer vision algorithms that could be further investigated for the same purpose.

7.1.1 Implementation with Data from Ship Maneuvering Simulator

Consider the navigation experiment from the ship maneuvering simulator. Figure 36 illustrates initial and final images for the implementation.

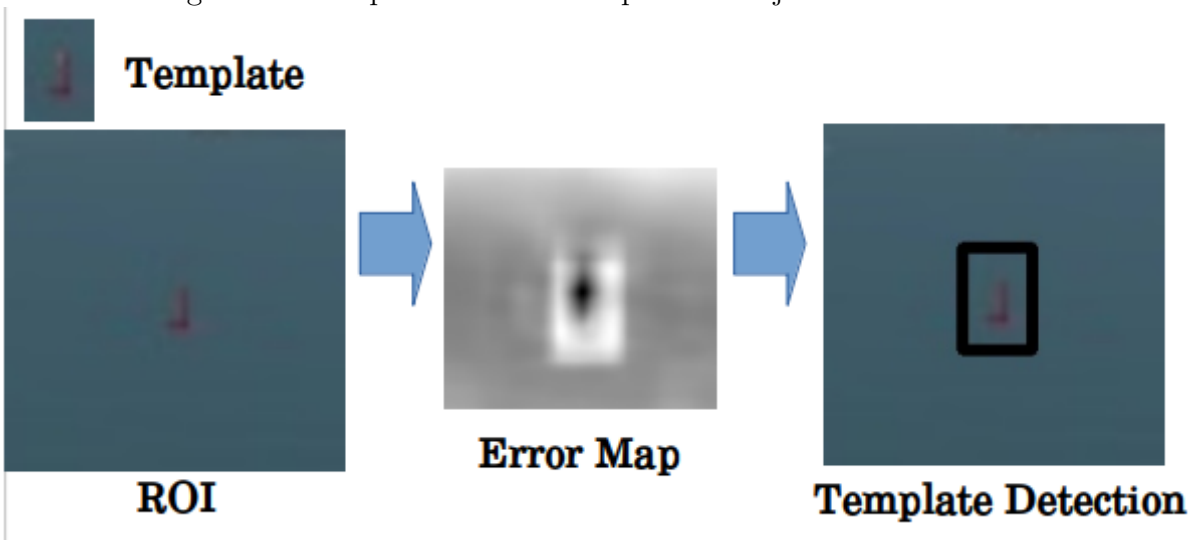
Figure 36: Inputs for the object detector feature



Source: Author

Initially, a manual initialization is required for the definition of an initial template. For each following iterations, the template is searched with the *matchTemplate* function of the *OpenCV* library in an amplified region around the detection position from the previous iteration. This step is illustrated by Figure 37.

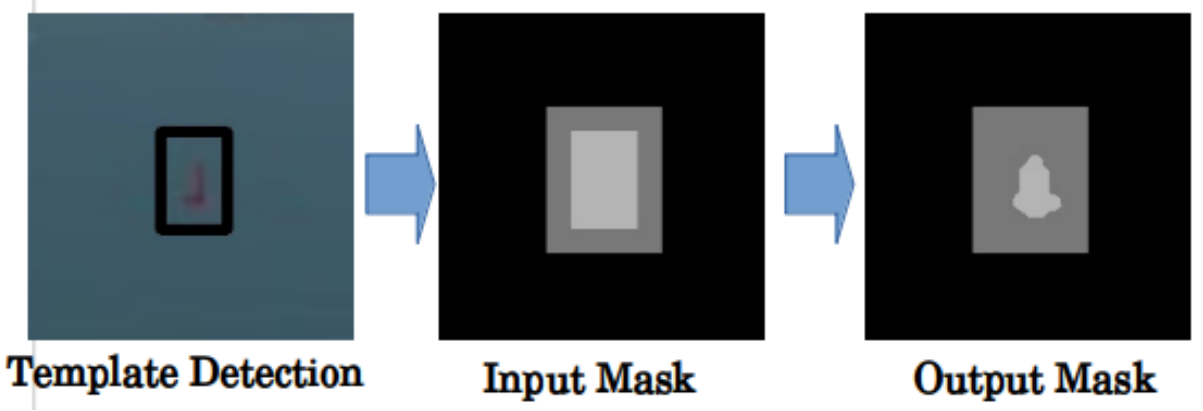
Figure 37: Template detection step in the object detector feature



Source: Author

The template detection region is further processed with the *grabCut* function of the *OpenCV* library. The initialization of the algorithm uses an input mask where the template detection region is marked as a region of possible foreground. The rest of the input mask is marked as background. This step is illustrated by Figure 38.

Figure 38: Image segmentation step in the object detector feature



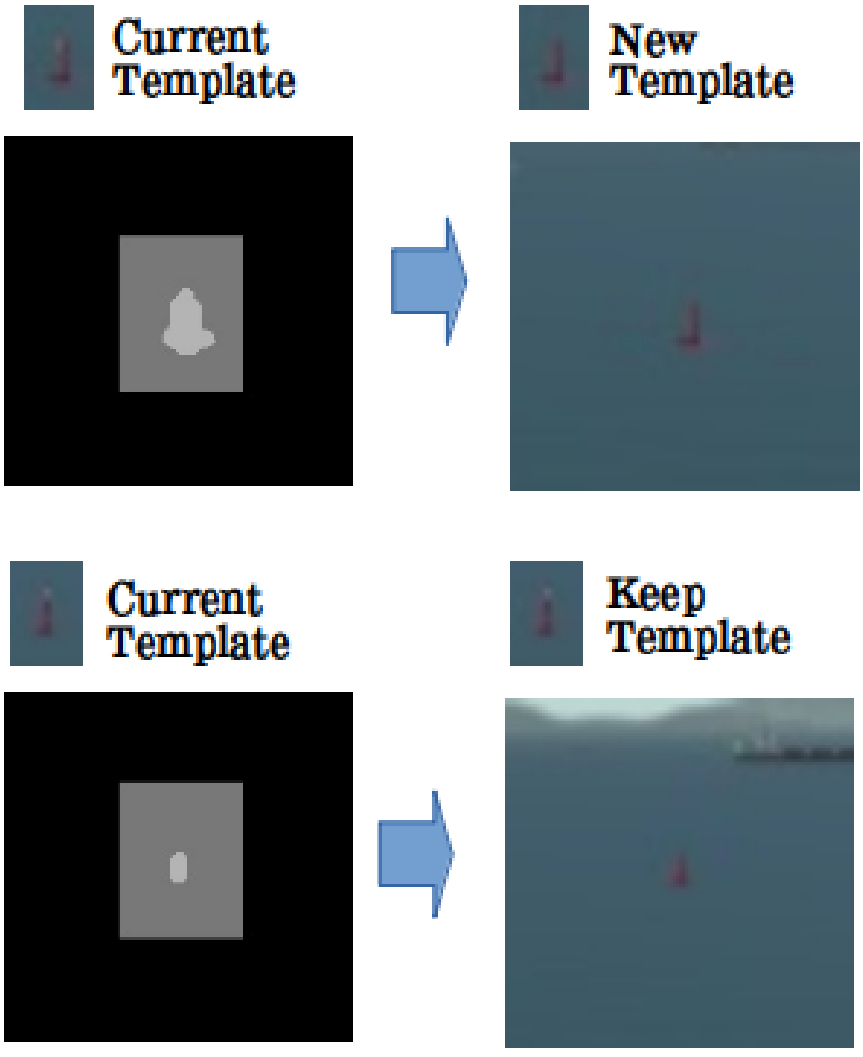
Source: Author

Two main conditions are checked regarding the update of the current template. The outputs from both algorithms are stored on an array with a fixed size. At each iteration,

oldest outputs are discarded to insert the most recent outputs. If the standard deviation of this array is smaller than a fixed threshold, then the first condition is verified. The objective of this condition is to update the template only if a stable output is available.

Additionally, the size of the mask output from the image segmentation must not be too small. If the number of pixels considered as object foreground is greater than a fixed threshold, the second condition for updating the current template is verified. This step is illustrated by Figure 39. If both conditions are verified, then the template is updated.

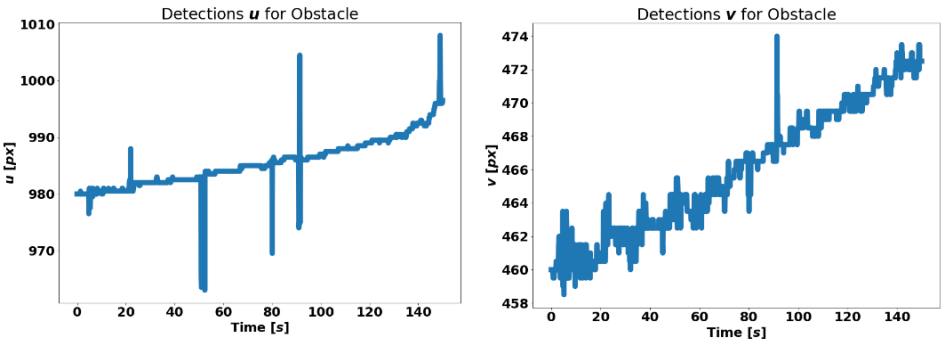
Figure 39: Template update step in the object detector feature



Source: Author

Finally, two-dimensional detections are determined by the mean of the upper-left and bottom-right points from the rectangle associated with each template detection. Figure 40 presents the corresponding detections as a function of time.

Figure 40: Two dimensional detections for the obstacle



Source: Author

7.1.2 Implementation with Data from Real Experiments

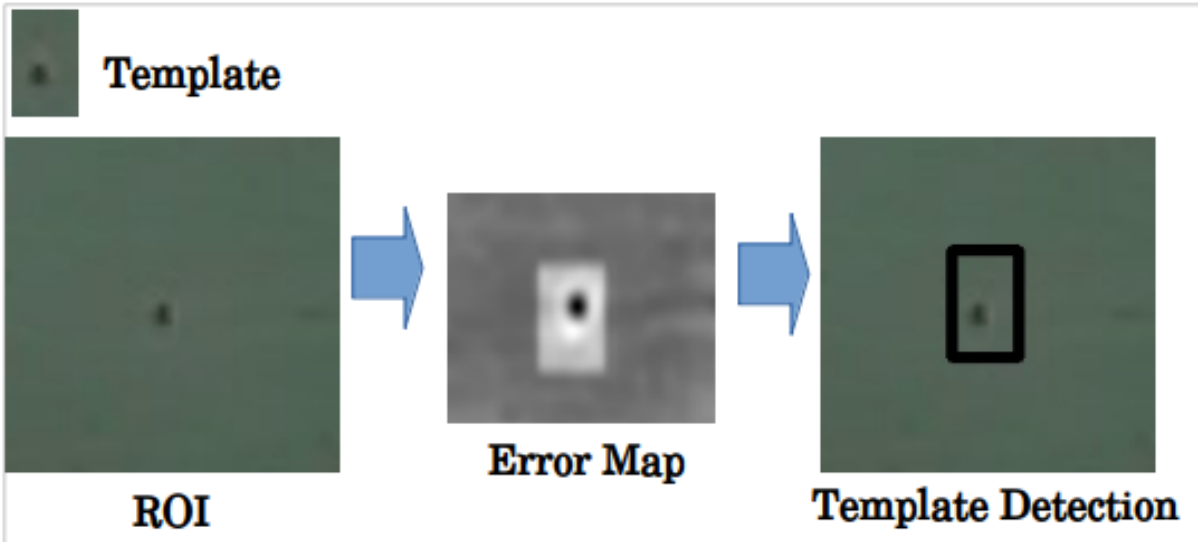
As in the previous implementation, consider the situation of tracking visible buoys. Figure 41 illustrates initial and final images of the video from real experiments.

Figure 41: Inputs for the object detector feature

Source: Author

The algorithm is initialized with a template that is further searched in the scene image. Figure 42 illustrates the implementation of the template detection step.

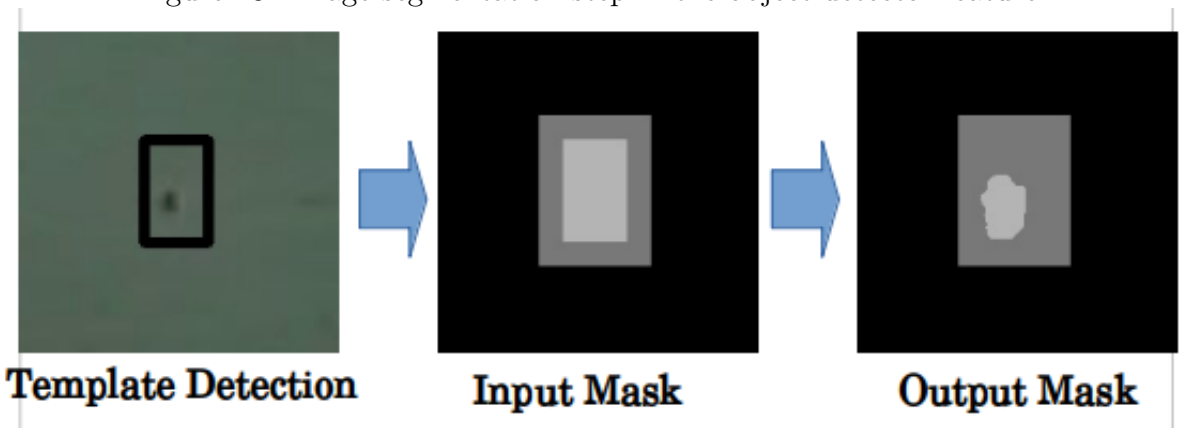
Figure 42: Template detection step in the object detector feature



Source: Author

Next, the detected region is further processed with the *grabCut* function of the *OpenCV* library to decide if the template should be updated. Figure 43 illustrates the implementation of this step in the experiment.

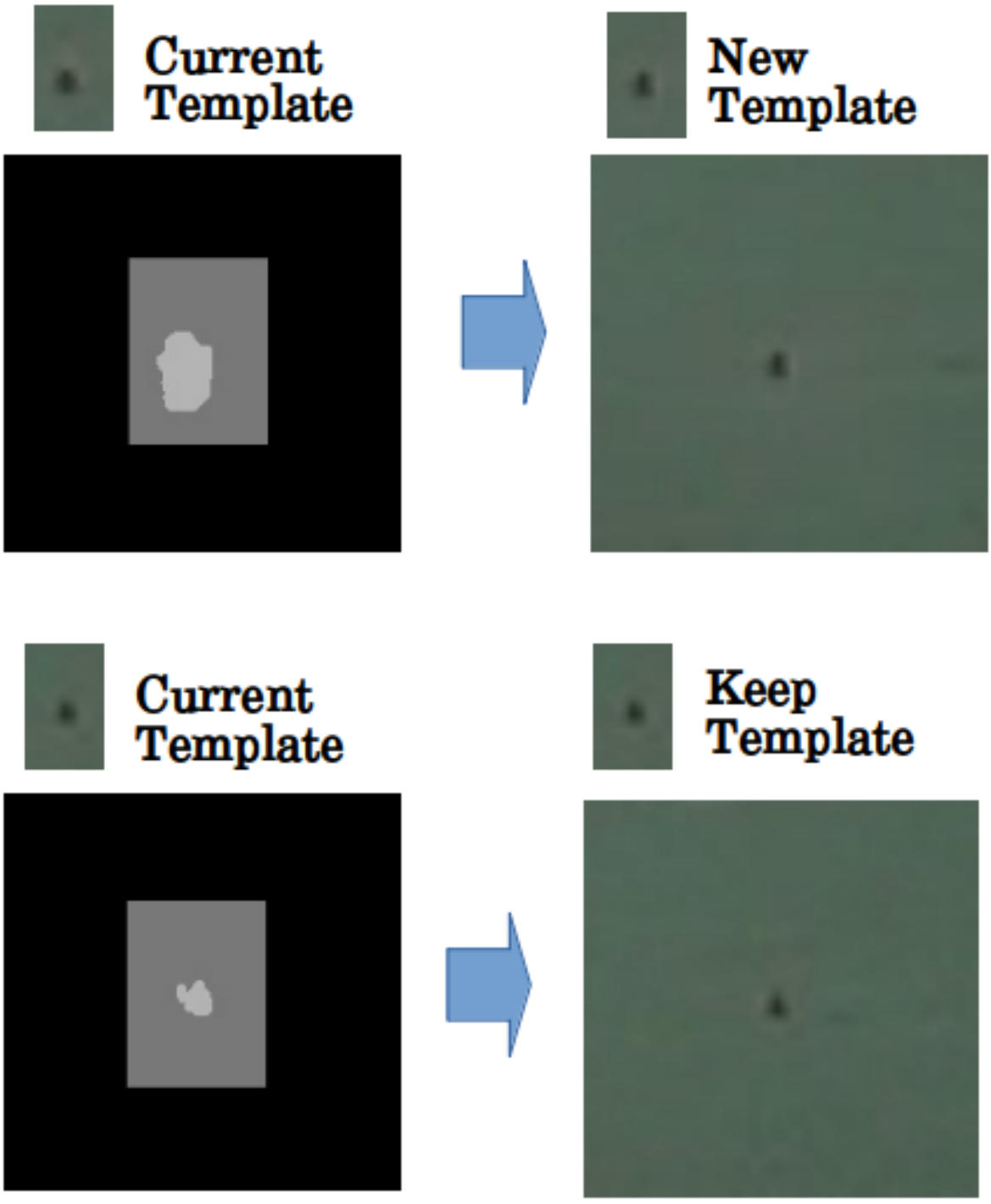
Figure 43: Image segmentation step in the object detector feature



Source: Author

Similarly as before, the template is updated if recent outputs are stable and the size of the mask output is not too small. Figure 44 illustrates the implementation of the two situations in the template update step. In one of them the template is updated and in the other the template is not updated.

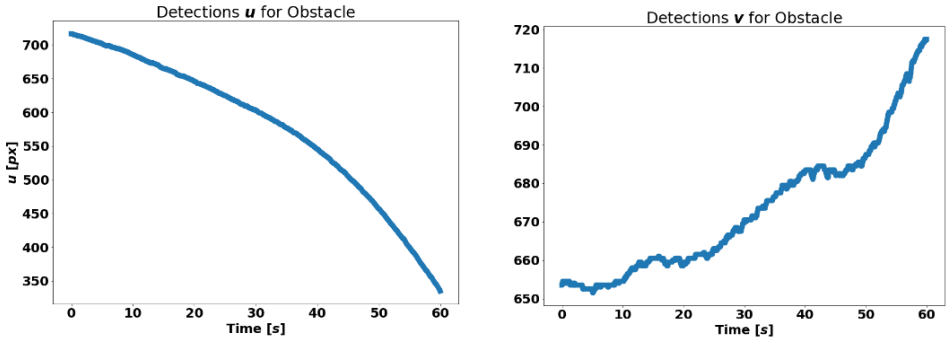
Figure 44: Template update step in the object detector feature



Source: Author

Finally, two-dimensional detections are computed for each iteration as the center of the rectangle associated with the template detection. Figure 45 presents the corresponding detections as a function of time.

Figure 45: Two dimensional detections for the obstacle



Source: Author

7.2 Relative Position Estimation

7.2.1 Formulation

As the camera image is a projection of relative measurements from the surrounding scene, it is possible to estimate relative geometrical information from image detections. Assume that a detector is able to provide image detections of a given obstacle. It is possible to estimate the relative position and velocity from such detections. These estimations can be particularly useful for the identification of obstacles without further external information from other systems.

Consider the following nonlinear dynamic system to represent the obstacle relative state.

$$d\vec{x}(t) = f(x(t), u(t), t)dt + G(t)d\vec{\beta}(t) \tag{7.1}$$

The state is further defined with the position and velocity of the obstacle with respect to the camera:

$$x(t) = \begin{bmatrix} p_\chi^c(t) & p_\gamma^c(t) & p_\kappa^c(t) & v_\chi^c(t) & v_\gamma^c(t) & v_\kappa^c(t) \end{bmatrix}^T \tag{7.2}$$

Let $u_\chi(t), u_\gamma(t), u_\kappa(t)$ be acceleration inputs for the state of the obstacle with respect to the CCS. Assume that such inputs are composed of two components: one related to its controls u^{CN} and another related to perturbations u^{PR}.

$$u = \begin{bmatrix} u_\chi(t) \\ u_\gamma(t) \\ u_\kappa(t) \end{bmatrix} = \begin{bmatrix} u_\chi(t)^{CN} + u_\chi(t)^{PR} \\ u_\gamma(t)^{CN} + u_\gamma(t)^{PR} \\ u_\kappa(t)^{CN} + u_\kappa(t)^{PR} \end{bmatrix}$$

Thus, the derivatives for the state can be represented in a vector form as:

$$f(x(t), u(t), t) = \begin{bmatrix} v_\chi^c(t) & v_\gamma^c(t) & v_\kappa^c(t) & u_\chi(t) & u_\gamma(t) & u_\kappa(t) \end{bmatrix}^T \tag{7.3}$$

Consider the situation of an obstacle moving with constant velocity with respect to the CCS. In such scenario, the acceleration inputs u^{CN} are zero. Further, assume that the perturbations u^{PR} may be represented as a random variable in a form of a Brownian motion $\vec{\beta}(t)$. The perturbations between each measurement from the camera may be represented as a Gaussian random vector with zero mean and covariance matrix $Q_\beta(t)$. The corresponding Jacobian matrices F and G that approximates the obstacle dynamics can be computed as:

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad G = \frac{\partial f}{\partial u^{PR}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.4}$$

Let ΔT_{dct} be the sampling period for the camera detector. The Kalman filter *updates* its estimates from each available detection at times $t + n\Delta T_{dct}$ for integers n. Between each sampling period ΔT_{dct}, the Kalman filter *propagates* its estimates with the respective model described by Jacobians F and G.

The measurement process is assumed to be described by a function $h(x(t), t)$ with an additive Gaussian noise $\vec{V}_{RV}$. The noise $\vec{V}_{RV}$ have zero mean and covariance P_V.

$$\begin{bmatrix} u_M(t) \\ v_M(t) \end{bmatrix} = h(x(t), t) + \vec{V}_{RV} \tag{7.5}$$

If distortion effects can be neglected, the projection of the relative state in the CCS can be computed with the following function:

$$h(x(t), t) = \begin{bmatrix} h_u(x(t), t) \\ h_v(x(t), t) \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}(p_\chi^c(t))(p_\kappa^c(t))^{-1} + u_0 \\ f\Delta_v^{-1}(p_\gamma^c(t))(p_\kappa^c(t))^{-1} + v_0 \end{bmatrix} \tag{7.6}$$

Equation 7.6 yields the following measurement Jacobian H:

$$H = \begin{bmatrix} f\Delta_u^{-1}(p_\kappa^c(t))^{-1} & 0 & -f\Delta_u^{-1}(p_\chi^c(t))(p_\kappa^c(t))^{-2} & 0 & 0 & 0 \\ 0 & f\Delta_v^{-1}(p_\kappa^c(t))^{-1} & -f\Delta_v^{-1}(p_\gamma^c(t))(p_\kappa^c(t))^{-2} & 0 & 0 & 0 \end{bmatrix}$$

Consider a point p_K described with coordinates in another coordinate system S_K. The state of the filter must be defined in the coordinate system S_K.

$$x(t) = \begin{bmatrix} p_x^K(t) & p_y^K(t) & p_z^K(t) & v_x^K(t) & v_y^K(t) & v_z^K(t) \end{bmatrix} \tag{7.7}$$

The corresponding measurement in the camera is computed from coordinates in the CCS. Thus, it is necessary to transform the coordinates from the coordinate system S_K to the camera frame with the corresponding rotation matrix R_K and camera center $\vec{c_K} = [c_x^K, c_y^K, c_z^K]^T$:

$$\begin{bmatrix} p_\chi^c(t) \\ p_\gamma^c(t) \\ p_\kappa^c(t) \end{bmatrix} = \begin{bmatrix} R_K & -R_K\vec{c_K} \end{bmatrix} \begin{bmatrix} p_x^K(t) \\ p_y^K(t) \\ p_z^K(t) \\ 1 \end{bmatrix}$$

Consider the representation of R_K by its rows R_K^{ri}:

$$R_K = \begin{bmatrix} r_{k1}^{r1} & r_{k2}^{r1} & r_{k3}^{r1} \\ r_{k1}^{r2} & r_{k2}^{r2} & r_{k3}^{r2} \\ r_{k1}^{r3} & r_{k2}^{r3} & r_{k3}^{r3} \end{bmatrix} = \begin{bmatrix} R_K^{r1} \\ R_K^{r2} \\ R_K^{r3} \end{bmatrix} \Rightarrow \begin{bmatrix} R_K & -R_K\vec{c_K} \end{bmatrix} = \begin{bmatrix} R_K^{r1} & -R_K^{r1}\vec{c_K} \\ R_K^{r2} & -R_K^{r2}\vec{c_K} \\ R_K^{r3} & -R_K^{r3}\vec{c_K} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} p_\chi^c(t) \\ p_\gamma^c(t) \\ p_\kappa^c(t) \end{bmatrix} = \begin{bmatrix} R_K^{r1} & -R_K^{r1}\vec{c_K} \\ R_K^{r2} & -R_K^{r2}\vec{c_K} \\ R_K^{r3} & -R_K^{r3}\vec{c_K} \end{bmatrix} \begin{bmatrix} p_x^K(t) \\ p_y^K(t) \\ p_z^K(t) \\ 1 \end{bmatrix} = \begin{bmatrix} R_K^{r1}\left(\vec{p}_K(t) - \vec{c}_K\right) \\ R_K^{r2}\left(\vec{p}_K(t) - \vec{c}_K\right) \\ R_K^{r3}\left(\vec{p}_K(t) - \vec{c}_K\right) \end{bmatrix}$$

$$h(x(t),t) = \begin{bmatrix} h_u(x(t),t) \\ \\ h_v(x(t),t) \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\left(R_K^{r1}\left(\vec{p}_K(t) - \vec{c}_K\right)\right)\left(R_K^{r3}\left(\vec{p}_K(t) - \vec{c}_K\right)\right)^{-1} + u_0 \\ \\ f\Delta_v^{-1}\left(R_K^{r2}\left(\vec{p}_K(t) - \vec{c}_K\right)\right)\left(R_K^{r3}\left(\vec{p}_K(t) - \vec{c}_K\right)\right)^{-1} + v_0 \end{bmatrix} \tag{7.8}$$

The corresponding Jacobian H can be computed from symbolic methods. Note that if the distortion of the installed camera can not be neglected, the corresponding distor-

tion terms needs to be considered in the projection equation. The initialization of the filter is made from an initial set of N_d detections. Consider initially the implementation with points described with respect to the CCS. Let $[u_M^k, v_M^k]^T$ be the coordinate of each detection from the initial set with $1 \leq k \leq N_d$. If one of the position coordinate is known, it is possible to compute estimates for the rest of the position state from the mean of all N_d detections.

$$\begin{bmatrix} \bar{u}_M \\ \bar{v}_M \end{bmatrix} = \frac{1}{N_d} \sum_{k=1}^{N_d} \begin{bmatrix} u_M^k \\ v_M^k \end{bmatrix} \Rightarrow \tag{7.9}$$

$$\begin{bmatrix} \hat{p}_\chi(t_0) \\ \hat{p}_\gamma(t_0) \end{bmatrix} = \frac{1}{f} \begin{bmatrix} \Delta_u(\bar{u}_M - u_0) \\ \Delta_v(\bar{v}_M - v_0) \end{bmatrix} \hat{p}_\kappa(t_0) \tag{7.10}$$

Further, the covariance for the noise of the sensor can be estimated from the initial set of N_d detections.

$$P_V = \begin{bmatrix} P_{Vu} & 0 \\ 0 & P_{Vv} \end{bmatrix} \Rightarrow \; P_{Vu} = \frac{1}{N_d} \sum_{k=1}^{N_d} \Big(u_M^k - \bar{u}_M \Big)^2 \;\; , \;\; P_{Vv} = \frac{1}{N_d} \sum_{k=1}^{N_d} \Big(v_M^k - \bar{v}_M \Big)^2 \tag{7.11}$$

Assume that the position $\hat{p}_\gamma(t_0)$ is known as γ_0 with an uncertainty δ_γ:

$$\hat{p}_\gamma(t_0) = \gamma_0 + N(0, \delta_\gamma) \tag{7.12}$$

where $N(0, \delta_\gamma)$ denotes a Gaussian random variable with zero mean and covariance δ_γ. The other coordinates can be computed as a function of measurements $[\bar{u}_M, \bar{v}_M]^T$ and γ_0 as:

$$\hat{p}_\kappa(t_0) = \Big[\frac{f}{\Delta_v} \Big(\bar{v}_M - v_0 \Big)^{-1} \Big] \gamma_0 \tag{7.13}$$

$$\hat{p}_\chi(t_0) = \frac{\Delta_u}{f} \Big(\bar{u}_M - u_0 \Big) \Big[\frac{f}{\Delta_v} \Big(\bar{v}_M - v_0 \Big)^{-1} \Big] \gamma_0 \tag{7.14}$$

Consider the implementation of N_K filters for a parallel estimation of the state. Let $m_{X_0}^i$ denotes the initial state of the $i-$th filter. Assume that there is a three-dimensional estimation $\vec{v}_{\chi,\gamma,\kappa}$ for the velocity with corresponding uncertainty $\vec{\delta} v_{\chi,\gamma,\kappa}$. Each initial

estimate $m^i_{X_0}$ is randomized considering the uncertainties δ_γ and $\vec{\delta}v_{\chi,\gamma,\kappa}$:

$$m^i_{X_0} = \begin{bmatrix} m^i_{p_\chi} & m^i_{p_\gamma} & m^i_{p_\kappa} & m^i_{v_\chi} & m^i_{v_\gamma} & m^i_{v_\kappa} \end{bmatrix}^T \Rightarrow m^i_{p_\gamma} = \gamma_0 + N(0,\delta_\gamma) \quad (7.15)$$

$$m^i_{p_\kappa} = \Big[\frac{f}{\Delta_v}\Big(\bar{v}_M - v_0\Big)^{-1}\Big] m^i_{p_\gamma} \quad (7.16)$$

$$m^i_{p_\chi} = \frac{\Delta_u}{f}\Big(\bar{u}_M - u_0\Big)\Big[\frac{f}{\Delta_v}\Big(\bar{v}_M - v_0\Big)^{-1}\Big] m^i_{p_\gamma} \quad (7.17)$$

$$m^i_{v_\chi} = v_{\chi 0} + N(0,\delta_{v\chi}) \quad , \quad m^i_{v_\gamma} = v_{\gamma 0} + N(0,\delta_{v\gamma}) \quad , \quad m^i_{v_\kappa} = v_{\kappa 0} + N(0,\delta_{v\kappa}) \quad (7.18)$$

The corresponding covariance matrix P_{X_0} is initialized accordingly to the expressions:

$$P^i_{p_\gamma} = 0 \quad , \quad P^i_{p_\kappa} = \Big(\Big[\frac{f}{\Delta_v}\Big(\bar{v}_M - v_0\Big)^{-1}\Big]\Big)^2 \delta_\gamma \quad (7.19)$$

$$P^i_{p_\chi} = \Big(\frac{\Delta_u}{f}\Big(\bar{u}_M - u_0\Big)\Big[\frac{f}{\Delta_v}\Big(\bar{v}_M - v_0\Big)^{-1}\Big]\Big)^2 \delta_\gamma \quad (7.20)$$

$$P^i_{v_\chi} = \delta_{v\chi} \quad , \quad P^i_{v_\gamma} = \delta_{v\gamma} \quad , \quad P^i_{v_\kappa} = \delta_{v\kappa} \quad (7.21)$$

Matricially:

$$P^i_{X_0} = \begin{bmatrix} P^i_{p_\chi} & 0 & 0 & 0 & 0 & 0 \\ 0 & P^i_{p_\gamma} & 0 & 0 & 0 & 0 \\ 0 & 0 & P^i_{p_\kappa} & 0 & 0 & 0 \\ 0 & 0 & 0 & P^i_{v_\chi} & 0 & 0 \\ 0 & 0 & 0 & 0 & P^i_{v_\gamma} & 0 \\ 0 & 0 & 0 & 0 & 0 & P^i_{v_\kappa} \end{bmatrix} \quad (7.22)$$

If points coordinates are expressed with respect to the coordinate system S_K, the initialization can be similarly computed from the knowledge of one of the coordinates. Recall the inverse transformations between coordinate systems:

$$\begin{bmatrix} p^K_x(t) \\ p^K_y(t) \\ p^K_z(t) \end{bmatrix} = \begin{bmatrix} R^T_K & \vec{c}_K \end{bmatrix} \begin{bmatrix} p^c_\chi(t) \\ p^c_\gamma(t) \\ p^c_\kappa(t) \\ 1 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} r^{r1}_{k1} & r^{r1}_{k2} & r^{r1}_{k3} \\ r^{r2}_{k1} & r^{r2}_{k2} & r^{r2}_{k3} \\ r^{r3}_{k1} & r^{r3}_{k2} & r^{r3}_{k3} \end{bmatrix}^T \begin{bmatrix} c^K_x \\ c^K_y \\ c^K_z \end{bmatrix} \end{bmatrix} \begin{bmatrix} p^c_\chi(t) \\ p^c_\gamma(t) \\ p^c_\kappa(t) \\ 1 \end{bmatrix} \Rightarrow \quad (7.23)$$

$$\begin{bmatrix} p^K_x(t) \\ p^K_y(t) \\ p^K_z(t) \end{bmatrix} = \begin{bmatrix} r^{r1}_{k1} & r^{r2}_{k1} & r^{r3}_{k1} & c^K_x \\ r^{r1}_{k2} & r^{r2}_{k2} & r^{r3}_{k2} & c^K_y \\ r^{r1}_{k3} & r^{r2}_{k3} & r^{r3}_{k3} & c^K_z \end{bmatrix} \begin{bmatrix} p^c_\chi(t) \\ p^c_\gamma(t) \\ p^c_\kappa(t) \\ 1 \end{bmatrix} \quad (7.24)$$

Consider that the coordinate p_y^K is known:

$$p_y^K - c_y^K = r_{k2}^{r1} p_\chi^C + r_{k2}^{r2} p_\gamma^C + r_{k2}^{r3} p_\kappa^C \tag{7.25}$$

For points in front of the camera, $p_\kappa^C > 0$:

$$\frac{p_y^K - c_y^K}{p_\kappa^C} = r_{k2}^{r1}\frac{p_\chi^C}{p_\kappa^C} + r_{k2}^{r2}\frac{p_\gamma^C}{p_\kappa^C} + r_{k2}^{r3} \Rightarrow p_\kappa^C = \frac{p_y^K - c_y^K}{r_{k2}^{r1}\frac{p_\chi^C}{p_\kappa^C} + r_{k2}^{r2}\frac{p_\gamma^C}{p_\kappa^C} + r_{k2}^{r3}} \Rightarrow \tag{7.26}$$

$$p_\kappa^C = (p_y^K - c_y^K)\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)^{-1} \tag{7.27}$$

$$p_\chi^C = \frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}(p_y^K - c_y^K)\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)^{-1} \tag{7.28}$$

$$p_\gamma^C = \frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}(p_y^K - c_y^K)\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)^{-1} \tag{7.29}$$

Thus, it is possible to compute the rest of the coordinates p_x^K and p_z^K from the parameters p_y^K and $[\bar{u}_M, \bar{v}_M]$ according to:

$$p_x^K - c_x^K = r_{k1}^{r1} p_\chi^c + r_{k1}^{r2} p_\gamma^c + r_{k1}^{r3} p_\kappa^c \Rightarrow \tag{7.30}$$

$$p_x^K - c_x^K = r_{k1}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big)p_\kappa^c + r_{k1}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big)p_\kappa^c + r_{k1}^{r3} p_\kappa^c \Rightarrow \tag{7.31}$$

$$p_x^K - c_x^K = \Big(r_{k1}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k1}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k1}^{r3}\Big)p_\kappa^c \Rightarrow \tag{7.32}$$

$$p_x^K - c_x^K = \frac{\Big(r_{k1}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k1}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k1}^{r3}\Big)\Big(p_y^K - c_y^K\Big)}{\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)} \Rightarrow \tag{7.33}$$

$$p_x^K = \frac{\Big(r_{k1}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k1}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k1}^{r3}\Big)\Big(p_y^K - c_y^K\Big)}{\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)} + c_x^K \tag{7.34}$$

$$p_z^K - c_z^K = r_{k3}^{r1} p_\chi^c + r_{k3}^{r2} p_\gamma^c + r_{k3}^{r3} p_\kappa^c \Rightarrow \tag{7.35}$$

$$p_z^K - c_z^K = \frac{\Big(r_{k3}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k3}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k3}^{r3}\Big)\Big(p_y^K - c_y^K\Big)}{\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)} \Rightarrow \tag{7.36}$$

$$p_z^K = \frac{\Big(r_{k3}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k3}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k3}^{r3}\Big)\Big(p_y^K - c_y^K\Big)}{\Big(r_{k2}^{r1}\Big(\frac{\bar{u}_M - u_0}{f\Delta_u^{-1}}\Big) + r_{k2}^{r2}\Big(\frac{\bar{v}_M - v_0}{f\Delta_v^{-1}}\Big) + r_{k2}^{r3}\Big)} + c_z^K \tag{7.37}$$

Note that the position coordinates p_x^K and p_z^K are related with p_y^K by a multiplicative factor. Similarly as before, it is possible to initialize the mean $m_{X_0}^i$ and covariance $P_{X_0}^i$ of

N_k filters for a parallel estimation of the state.

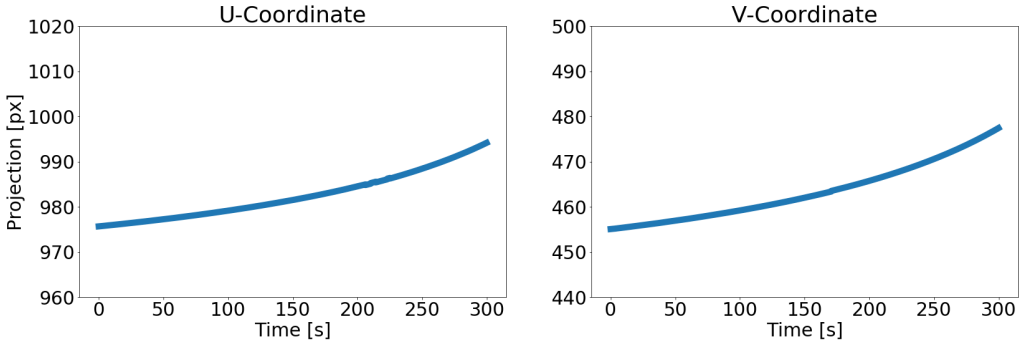
7.2.2 Implementations with Synthetic Data

In order to investigate the ideal performance of the filter, consider a situation with ideal measurements computed from the following camera intrinsic parameters:

$$f_u = \frac{f}{\Delta_u} = 796.32 \quad , \quad f_v = \frac{f}{\Delta_v} = 748.41 \quad , \quad u_0 = 961.72 \quad , \quad v_0 = 438.6$$

Figure 46 presents the resulting projection of the red buoy in the right camera as a function of time.

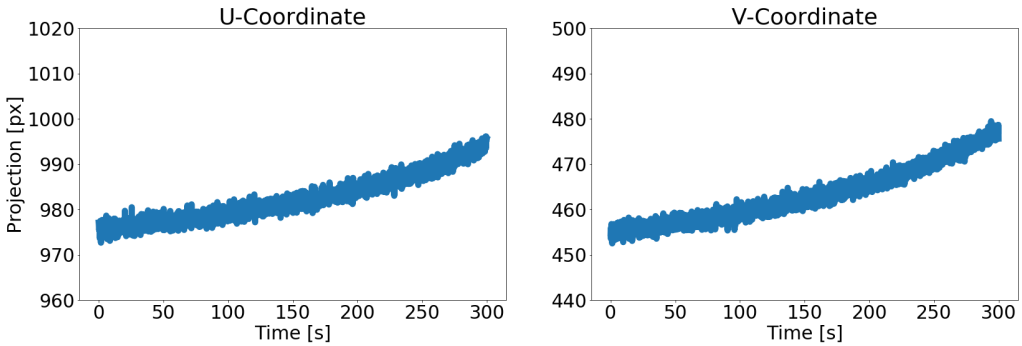
Figure 46: Ideal measurements of the red buoy in the right camera



Source: Author

The ideal measurements shown in Figure 46 are combined with an additive Gaussian noise $\vec{V}_{RV}$ with zero mean and covariance P_V. Figure 47 presents the respective simulated camera measurements with unitary covariance for the additive noise.

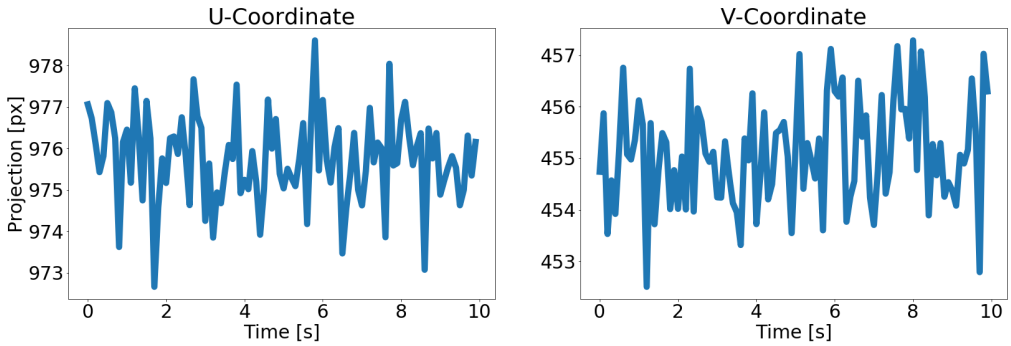
Figure 47: Simulated camera measurements with $P_V \sim \begin{bmatrix} 1.0 & 0 \\ 0 & 1.0 \end{bmatrix}$



Source: Author

The filter is initialized with the distance between the camera and the surface of the sea and an initial set of detections. The distance between the camera and the surface of the sea was defined as 43.8 m. Figure 48 shows the initial set of measurements with 10 seconds of duration.

Figure 48: Initial set of ideal measurements



Source: Author

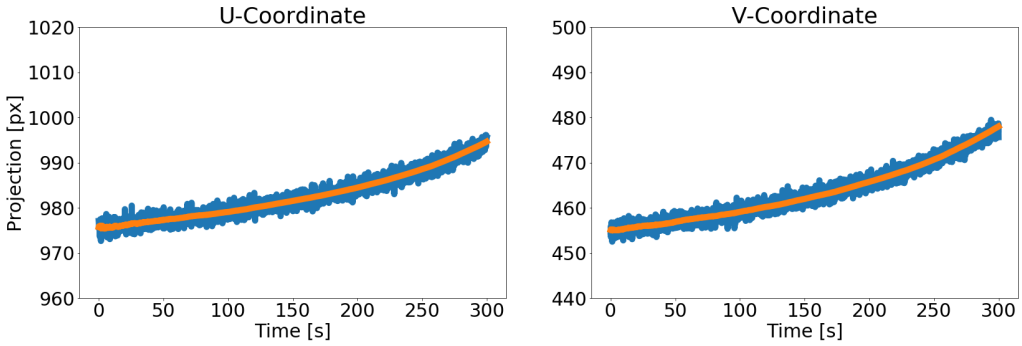
The initial covariance P_V for the error of the sensor is estimated from the initial set of measurements. Summarizing the initial parameters:

$$Q_\beta = 0 \quad , \quad \gamma_0 = 43.8 \quad , \quad \delta_\gamma = 0.5 \quad , \quad v_{\gamma 0} = 0.0 \quad , \quad \delta_{v\gamma} = 0.0 \tag{7.38}$$

$$v_{\chi 0} = 0.0 \quad , \quad \delta_{v\chi} = 1.0 \quad , \quad v_{\kappa 0} = -4.0 \quad , \quad \delta_{v\kappa} = 1.0 \tag{7.39}$$

For each following iteration, the estimated state is projected into image coordinates and compared with each incoming detection. Figure 49 presents the projection of one filter alongside each measurement as a function of time.

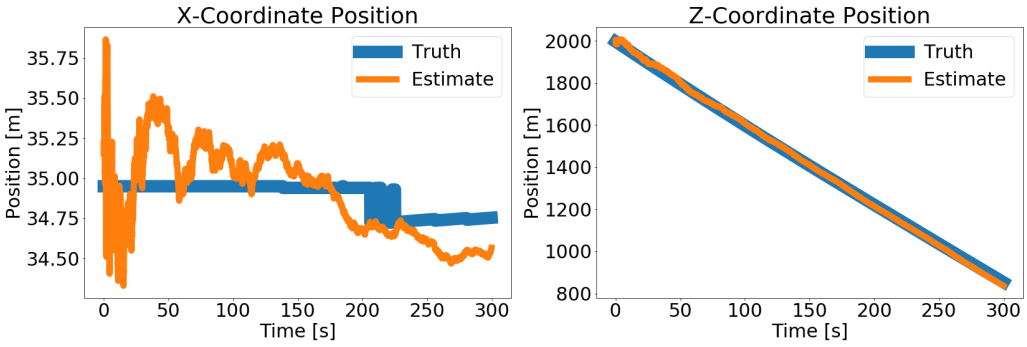
Figure 49: Estimated projections and measurements



Source: Author

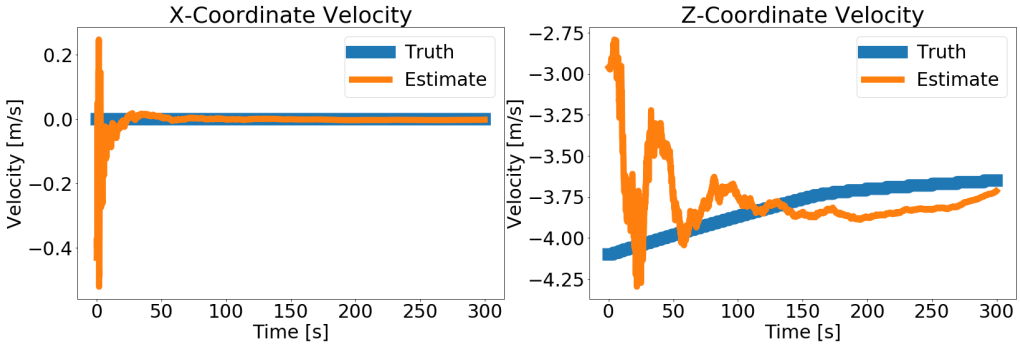
Figures 50 and 51 presents the filter estimations for the buoy position and velocity alongside the truth relative state as a function of time, respectively.

Figure 50: Estimated position as a function of time



Source: Author

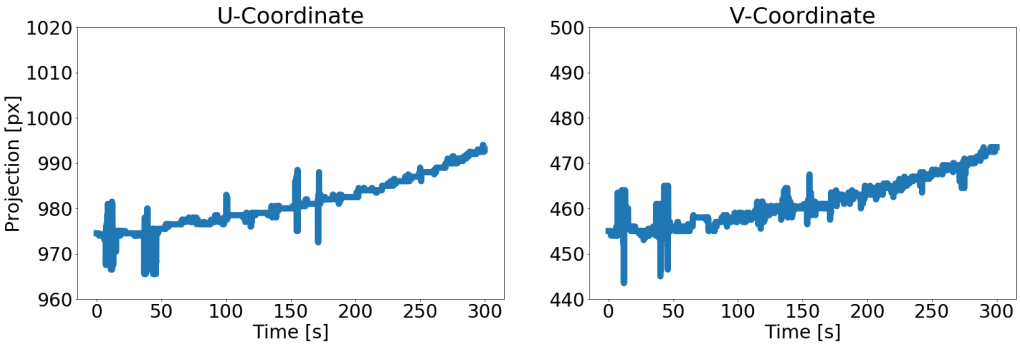
Figure 51: Estimated velocity as a function of time



Source: Author

As one can see, the filter is able to estimate the relative state of the buoy from ideal measurements. Consider now the same implementation with measurements from image detection methods as the one described in the last chapter. Figure 52 presents corresponding measurements for the same period as in previous results.

Figure 52: Measurements from image detection algorithms

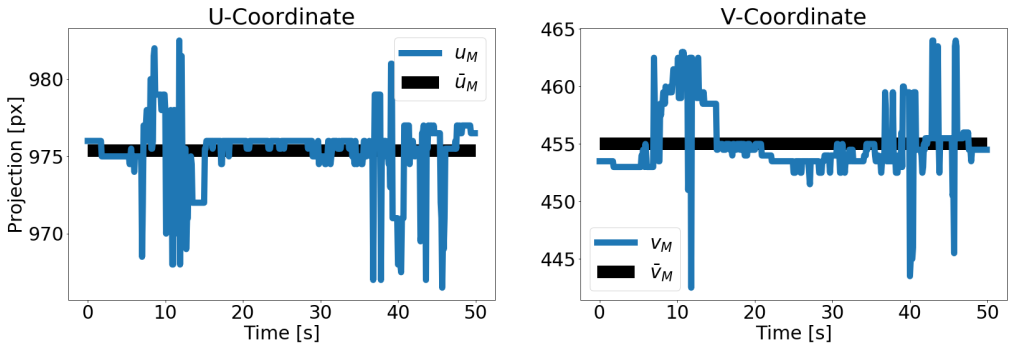


Source: Author

Similarly as before, the filter is initialized with an initial set of detections. Figure

53 presents an initial set of measurements corresponding to the initial 50 seconds of the experiment.

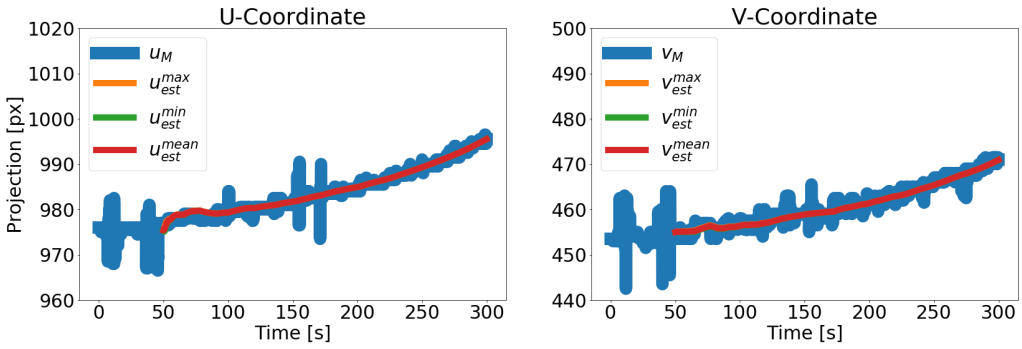
Figure 53: Initial set of measurements from image detections



Source: Author

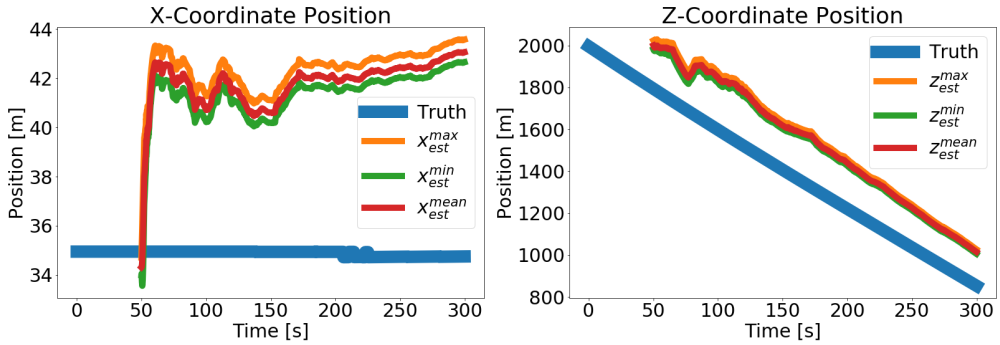
Let $N_k = 10$, consisting in an implementation with 10 filters for the estimation of the state from simulated measurements. The initialization of each one of the filters used parameters from Equation 7.39. For the following iterations, the image projection of each filter estimate is compared with incoming detections, yielding N_k estimates per iteration. Figure 54 presents filter estimates alongside the measurements of the experiment. Figures 55 and 56 presents the comparison between the estimated state and the relative state of the buoy computed from the simulation output. Note that the mean, maximum and minimum estimate between all filters are indicated in Figures 55 and 56.

Figure 54: Estimated projections and measurements



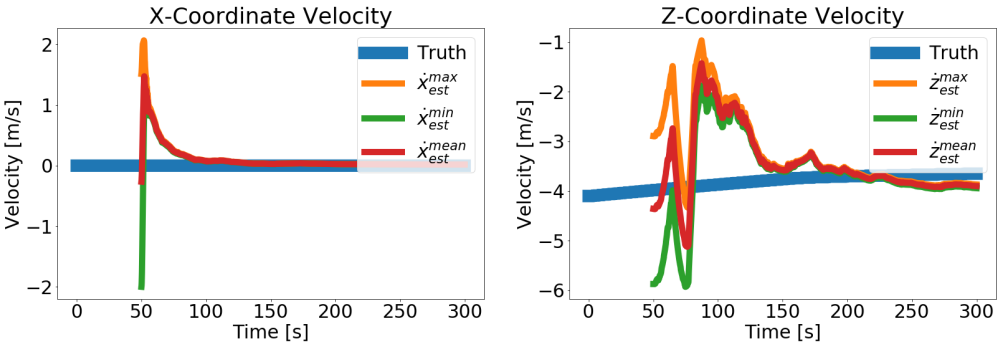
Source: Author

Figure 55: Estimated position as a function of time



Source: Author

Figure 56: Estimated velocity as a function of time



Source: Author

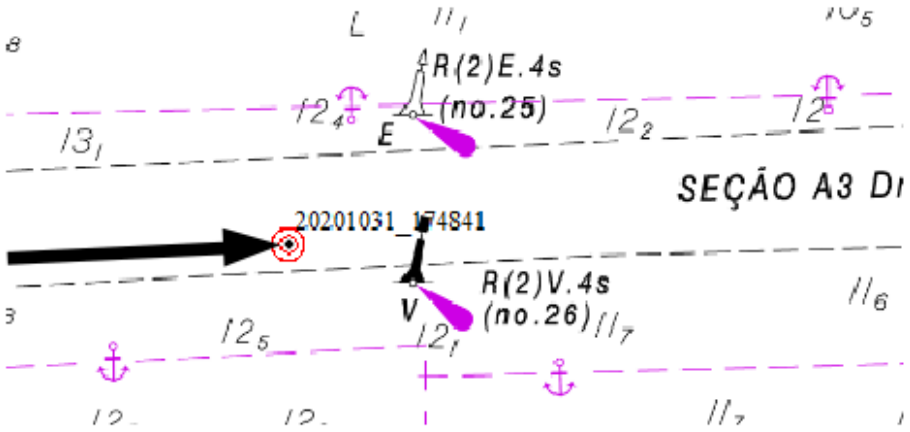
Note that the error between the filter estimate state and the truth relative state is higher than from previous experiments with ideal measurements. Thus, one source of potential improvements refers to improvements in the image detection algorithms used for the definition of input measurements. Moreover, recall that in the filter formulation the camera was assumed to be with constant orientation throughout the filter computation. The proposed filter may be implemented in more general scenarios with improvements in the dynamic model for the obstacle dynamics. Such considerations are beyond the scope of the present work.

7.2.3 Implementations with Real Data

Consider now the implementation of the filter in the aforementioned real experiments of navigation. The ship is navigating through a channel delimited by buoys similarly as in the video from the ship maneuvering simulator. The velocity of the ship is approximately constant with value $v_{SHIP} = 14.7$ knot. The theoretical size of the channel is $280m$.

Figure 57 shows a superior view for the surrounding region of the experiment.

Figure 57: Superior view for the ship navigation experiment



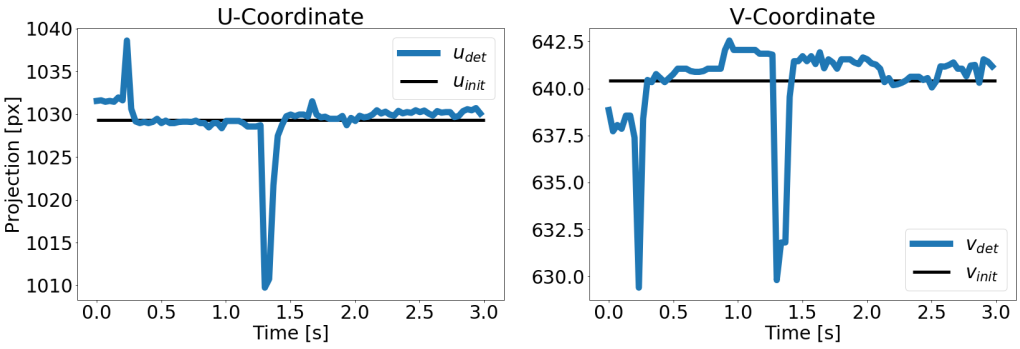
Source: Author

Consider the estimation of the relative state for the right buoy. Assume an additional coordinate system S_K with $\vec{c}_K = \vec{0}$ and a rotation matrix R_K defined from the installation procedure of the previous chapter.

$$\vec{c}_K = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad R_K = \begin{bmatrix} 0.99955 & -0.01871 & -0.02311 \\ 0.01636 & 0.99505 & -0.09800 \\ 0.02483 & 0.09758 & 0.99492 \end{bmatrix} \tag{7.40}$$

Similarly as before, the filter is initialized from an initial set of detections. Figure 58 presents the initial set of detections used for the initialization of the filter consisting of 3 seconds of measurements.

Figure 58: Initial set of measurements from image detections



Source: Author

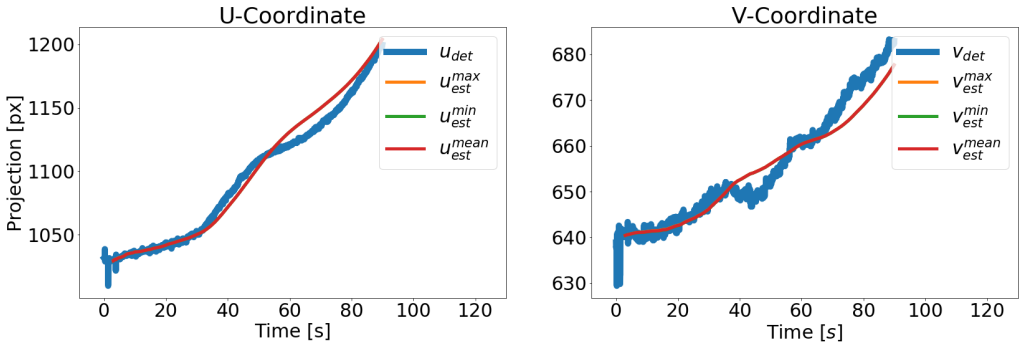
The initial covariance P_V is similarly estimated from the initial set of measurements. Ten different filters were initialized. Summarizing the initial parameters for this experiment:

$$Q_\beta = 0 \quad , \quad \gamma_0 = 32.26 \quad , \quad \delta_\gamma = 1.0 \quad , \quad v_{\gamma 0} = 0.0 \quad , \quad \delta_{v\gamma} = 0.0 \tag{7.41}$$

$$v_{\chi 0} = 0.0 \quad , \quad \delta_{v\chi} = 0.25 \quad , \quad v_{\kappa 0} = -7.0 \quad , \quad \delta_{v\kappa} = 0.25 \tag{7.42}$$

Figure 59 shows input detections alongside the corresponding mean projection of the filters for each time.

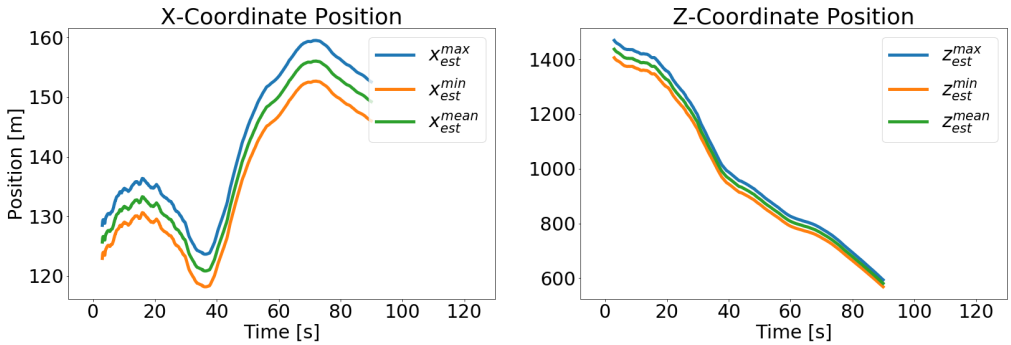
Figure 59: Input detections and corresponding image projections



Source: Author

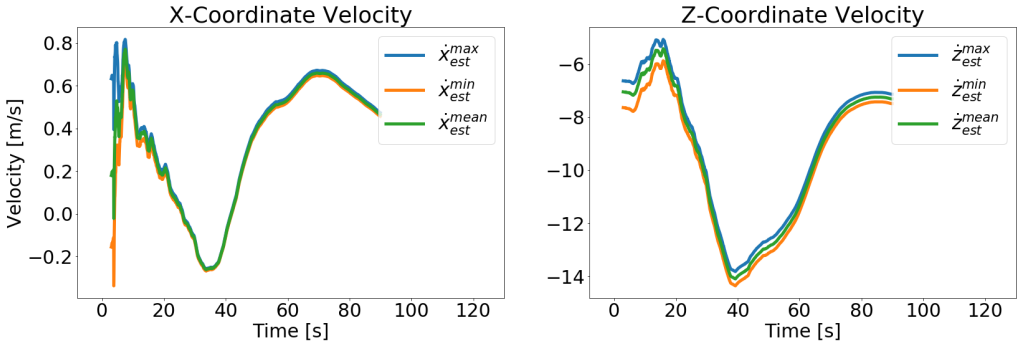
Figures 60 and 61 presents the estimated relative position and velocity of the obstacle respectively.

Figure 60: Estimation of relative position for the right buoy



Source: Author

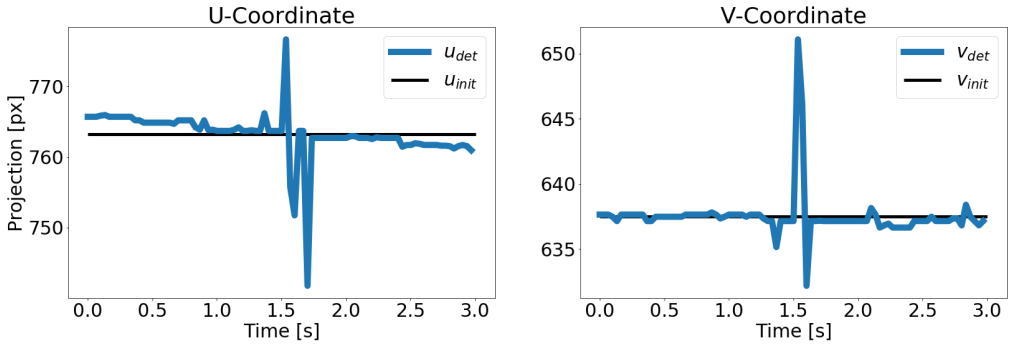
Figure 61: Estimation of relative velocity for the right buoy



Source: Author

Consider now the estimator implementation for the left buoy. Similarly, the filter is initialized with a set of initial detections as indicated by Figure 62.

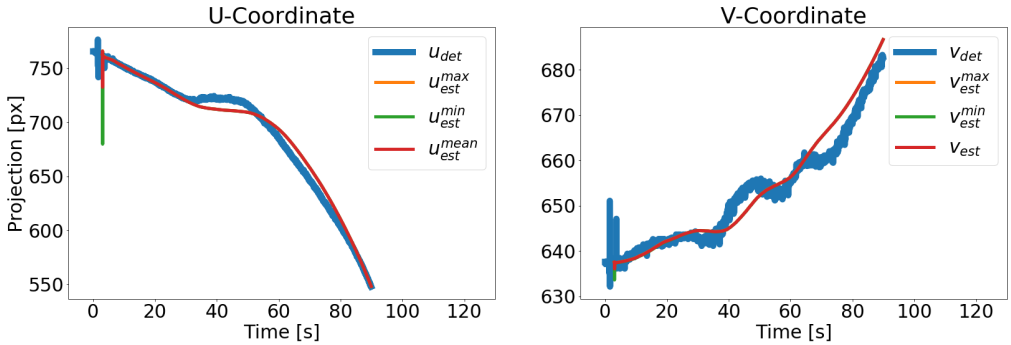
Figure 62: Initial set of measurements from image detections



Source: Author

Figure 63 shows the corresponding detections and projections of the filters for each time.

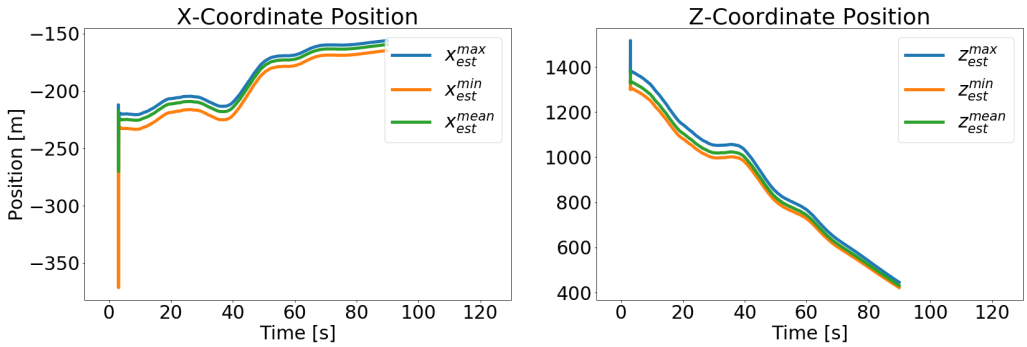
Figure 63: Input detections and corresponding image projections



Source: Author

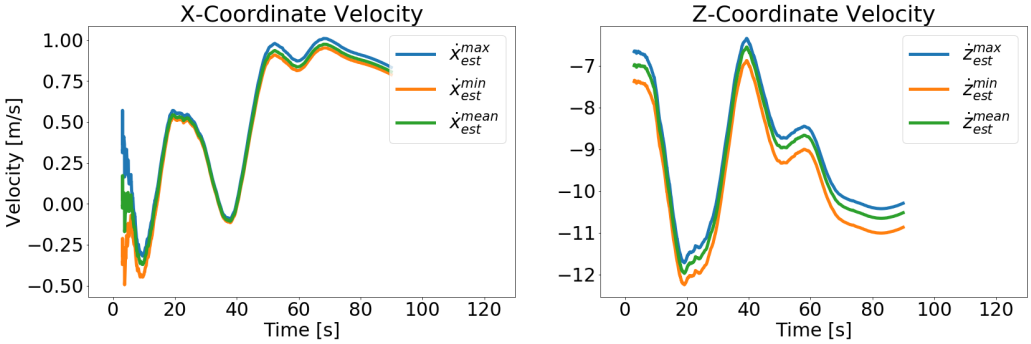
Figures 64 and 65 presents the estimated relative position and velocity for the left buoy, respectively.

Figure 64: Estimation of relative position for the left buoy



Source: Author

Figure 65: Estimation of relative velocity for the left buoy



Source: Author

As the truth relative state of the buoy is unknown, it is difficult to validate the estimation results independently. However, as the size of the channel is theoretically known, a simple validation for both estimates consists in the verification of the distance between estimates. In the previous case, for example, the difference between the x-coordinate of both estimates resulted in approximately $300m$, which is close to the theoretical channel size of $280m$.

7.3 Loosely Coupled Tracker Integration

The *tracker* setup from the previous chapter assumes the integration of an inertial measurement unit, a compass and a GNSS receiver. Generally, for a *loosely-coupled* integration, it is necessary to provide an estimation in terms of error components from the

filter state. Note that it is possible to estimate the camera position and orientation from the projection of points with known three-dimensional coordinates. Thus, assuming that the camera position and orientation with respect to the ship frame is known, it is possible to compute the ship position and orientation from the camera estimation alongside the corresponding error components. Then, these error components computed from camera measurements may be integrated into the *tracker* with a *loosely-coupled* integration.

7.3.1 Formulation

Consider a world coordinate system S_w fixed with respect to the earth frame. Let p^e_{ew} be the origin of the reference coordinate system with respect to the earth frame, and let R^e_w be the rotation matrix relating the reference frame and the earth frame. Further, consider a set of N visible points $\mathbf{P}^r$ with known coordinates in this reference frame. Each element of $\mathbf{P}^w$ is a point expressed in the reference frame as a three-dimensional vector.

$$\mathbf{P}^w = \begin{bmatrix} p^w_{wp1}, p^w_{wp2}, ..., p^w_{wpN} \end{bmatrix} \tag{7.43}$$

Let $\mathbf{Z}$ be the corresponding set of camera observations from $\mathbf{P}^w$. Each element of $\mathbf{Z}$ represents an image coordinate as a two-dimensional vector. Consider a *pinhole* model described by intrinsic parameters $[f_u, f_v, u_0, v_0]$.

$$\mathbf{Z} = \begin{bmatrix} [u_{p1}, v_{p1}]^T, [u_{p2}, v_{p2}]^T, ..., [u_{pN}, v_{pN}]^T \end{bmatrix} \tag{7.44}$$

Similarly as in the previous chapter, consider a vector Π_{CAM} concatenating the real parameters from the camera:

$$\Pi_{CAM} = \begin{bmatrix} [f_u, f_v, u_0, v_0], p^s_{sc}, R^s_c \end{bmatrix} \tag{7.45}$$

Let $\hat{p}^e_{es}$ and $\hat{R}^e_s$ be the *inertial navigation system* outputs for position and orientation of the ship with respect to the earth system. An estimation for the position $\hat{p}^w_{ws}$ and orientation $\hat{R}^w_s$ with respect to the reference coordinate system may be computed from p^e_{ew} and R^e_w:

$$\hat{p}^w_{ws} = \left(R^w_e\right)\left(\hat{p}^e_{es} - p^e_{ew}\right) = \left(R^e_w\right)^T\left(\hat{p}^e_{es} - p^e_{ew}\right) \tag{7.46}$$

$$\hat{R}^w_s = \left(R^w_e\right)\hat{R}^e_s = \left(R^e_w\right)^T\hat{R}^e_s \tag{7.47}$$

Let $\hat{p}^s_{sc}$ and $\hat{R}^s_c$ be estimates for the camera position and orientation with respect to the ship frame. Corresponding estimates for the camera position $\hat{p}^w_{wc}$ and orientation $\hat{R}^w_c$ with respect to the reference frame may be computed from $\hat{p}^w_{ws}$ and $\hat{R}^w_s$:

$$\hat{p}^w_{wc} = \hat{p}^w_{ws} + \hat{R}^w_s \hat{p}^s_{sc} \tag{7.48}$$

$$\hat{R}^w_c = \hat{R}^w_s \hat{R}^s_c \tag{7.49}$$

The set $\mathbf{P}^w$ of points with known coordinates in the reference frame may be transformed to the camera frame:

$$\hat{\mathbf{P}}^c = \left[..., \hat{p}^c_{cpi}, ...\right] = \left[..., (\hat{R}^w_c)^T \left(p^w_{wpi} - \hat{p}^w_{wc}\right), ...\right] \tag{7.50}$$

Assume a *pinhole* model described by intrinsic parameters $[\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0]$. Estimations $[\hat{u}_{pi}, \hat{v}_{pi}]$ may be computed from each point $\hat{p}^c_{cpi}$ of $\hat{\mathbf{P}}^c$:

$$\hat{p}^c_{cpi} = \begin{bmatrix} \hat{p}^{c,\chi}_{cpi} \\ \hat{p}^{c,\gamma}_{cpi} \\ \hat{p}^{c,\kappa}_{cpi} \end{bmatrix} \Rightarrow \begin{bmatrix} \hat{u}_{pi} \\ \hat{v}_{pi} \end{bmatrix} = \begin{bmatrix} \hat{f}_u \left(\hat{p}^{c,\chi}_{cpi}\right) \left(\hat{p}^{c,\kappa}_{cpi}\right)^{-1} + \hat{u}_0 \\ \hat{f}_v \left(\hat{p}^{c,\gamma}_{cpi}\right) \left(\hat{p}^{c,\kappa}_{cpi}\right)^{-1} + \hat{v}_0 \end{bmatrix} \tag{7.51}$$

Let $\hat{\mathbf{Z}}$ be the set of camera projection estimations computed from $\hat{\mathbf{P}}^c$ and $[\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0]$:

$$\hat{\mathbf{Z}} = \left[[\hat{u}_{p1}, \hat{v}_{p1}]^T, [\hat{u}_{p2}, \hat{v}_{p2}]^T, ..., [\hat{u}_{pN}, \hat{v}_{pN}]^T \right] \tag{7.52}$$

Further, let $\mathbf{E}$ be the error between observations $\mathbf{Z}$ and the estimated set $\hat{\mathbf{Z}}$.

$$\mathbf{E} = \left[u_{p1} - \hat{u}_{p1}, v_{p1} - \hat{v}_{p1}, ..., u_{pN} - \hat{u}_{pN}, v_{pN} - \hat{v}_{pN} \right] = \left[e_{u1}, e_{v1}, ..., e_{uN}, e_{vN} \right] \tag{7.53}$$

The observation set $\mathbf{Z}$ is assumed to be known alongside the camera intrinsic parameters $[\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0]$. The camera extrinsic parameters are initially set to $\hat{p}^w_{wc}$ and $\hat{R}^w_c$ accordingly to Equations 7.48 and 7.49. Then, the camera position and orientation is optimized with the minimization of $\mathbf{E}$. Typical procedures for optimizing camera extrinsic parameters from the minimization of $\mathbf{E}$ requires at least 4 points correspondences. Note that there are different methods for optimizing the error $\mathbf{E}$.

Let $\tilde{p}^w_{wc}$ and $\tilde{R}^w_c$ be the optimized position and orientation of the camera after minimizing $\mathbf{E}$. It is possible to compute the corresponding ship position $\tilde{p}^w_{ws}$ and orientation

$\tilde{R}_s^w$ from the optimized parameters $\tilde{p}_{wc}^w$ and $\tilde{R}_c^w$.

$$\tilde{R}_s^w = \tilde{R}_c^w \hat{R}_s^c = \tilde{R}_c^w \left(\hat{R}_c^s\right)^T \tag{7.54}$$

$$\tilde{p}_{ws}^w = \tilde{p}_{wc}^w - \left(\tilde{R}_s^w\right)\hat{p}_{sc}^s \tag{7.55}$$

Hence, it is possible to define error components in the reference coordinate system by the comparison of the INS outputs $[\hat{p}_{ws}^w, \hat{R}_s^w]$ with $[\tilde{p}_{ws}^w, \tilde{R}_s^w]$. However, as the *loosely coupled* integration equations from previous chapters are developed in the earth coordinate system, it is necessary to express the errors with respect to the earth coordinate system. Let $\hat{p}_{es}^e$ be the *INS* position described with respect to the earth frame without corrections. The position error δp^e in the earth frame may be computed as:

$$\delta p_{es}^e = \hat{p}_{es}^e - \left(p_{ew}^e + R_w^e \hat{p}_{ws}^w\right) \tag{7.56}$$

Further, let $\tilde{R}_s^e$ be the orientation of the ship with respect to the earth frame computed from the optimized orientation $\tilde{R}_s^w$:

$$\tilde{R}_s^e = R_w^e \tilde{R}_s^w \tag{7.57}$$

Let $\hat{R}_s^e$ be the *tracker* orientation described with respect to the earth frame without corrections. Consider the following matricial error δR_s^e between between $\tilde{R}_s^e$ and R_s^e:

$$\delta R_s^e = \tilde{R}_s^e - \hat{R}_s^e \tag{7.58}$$

Similarly as with the compass measurements, an angular displacement $\tilde{\Psi}^e$ that transforms the estimated rotation matrix $\hat{R}_s^e$ to the optimized rotation matrix $\tilde{R}_s^e$ may be determined by the minimization of δR_s^e with Equation C.78. Then, an angular error $\delta\psi_{es}^e$ may be determined comparing $\tilde{\Psi}^e$ with the angular displacement from consecutive gyrometer measurements. The position error and angular error are combined into a single observation vector z^{CAM}:

$$z^{CAM} = \begin{bmatrix} -\delta\psi^e \\ -\delta p^e \end{bmatrix} \Rightarrow H_Z^{CAM} = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{7.59}$$

The measurement noise covariance matrix R_Z^{CAM} should correspond with the uncer-

tainty of the error components. A simple form for R_Z^{CAM} is given by:

$$R_Z^{CAM} = \begin{bmatrix} \hat{w}_{c\psi} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \hat{w}_{cp} I_{3\times3} \end{bmatrix} \tag{7.60}$$

where $\hat{w}_{c\psi}$ and $\hat{w}_{cp}$ are associated with the uncertainty for estimates $\delta\psi^e$ and δp^e, respectively. Consider a vector Π_{LCI} concatenating parameters from the *loosely coupled* integrator similarly as in Equation 6.22:

$$\Pi_{LCI} = \left[\hat{w}_{\omega g}, \hat{w}_{fa}, \hat{w}_{ba}, \hat{w}_{bg}, \hat{p}^s_{sa}, \hat{w}_{\rho p}, \hat{w}_{\rho v}, \hat{p}^s_{s\psi}, \hat{R}^s_{\psi}, \hat{w}_{\psi}, [\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0], \hat{p}^s_{sc}, \hat{R}^s_c, \hat{w}_{c\psi}, \hat{w}_{cp}\right] \tag{7.61}$$

7.3.2 Implementations with Synthetic Data

In this section, the nautical buoys from the channel are used as reference points for the *loosely coupled* integration. Consider the same world coordinate system from the previous chapter given by Equations 6.8 and 6.10. Let $\mathbf{P}^w$ be the set of known points described with respect to the reference coordinate system. Four nautical buoys from the channel are used as reference points with the following coordinates:

$$p^w_{wp1} = \begin{bmatrix} 3472.1 \\ 685.1 \\ 17.6 \end{bmatrix}, \; p^w_{wp2} = \begin{bmatrix} 3470.2 \\ 499.5 \\ 17.6 \end{bmatrix}, \; p^w_{wp3} = \begin{bmatrix} 1544.7 \\ 671.5 \\ 17.6 \end{bmatrix}, \; p^w_{wp4} = \begin{bmatrix} 1542.3 \\ 485.5 \\ 17.6 \end{bmatrix} \tag{7.62}$$

The camera instantaneous position and orientation with respect to the reference system is computed from the simulation outputs to determine the set of observations $\mathbf{Z}$. A *pinhole* model described by intrinsic parameters $[f_u, f_v, u_0, v_0]$ is assumed. The set of estimations $\hat{\mathbf{Z}}$ are computed from estimated *intrinsic* parameters $[\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0]$ and estimated *extrinsic* parameters from the INS outputs. The projection error $\mathbf{E}$ is defined from the difference between observations $\mathbf{Z}$ and estimations $\hat{\mathbf{Z}}$ with Equation 7.53. An improved estimation for the camera position and orientation with respect to the world reference frame is computed by the minimization of the projection error $\mathbf{E}$. Finally, the *tracker* incorporates such improved estimations in a *loosely coupled* integration.

Similarly as in the loosely coupled integration from the previous chapter, consider a simulation comparing *tracker* performance from two different setups represented by sets of parameters $\Pi^1_{\dots}$ and $\Pi^2_{\dots}$. The only difference between both *tracker* setups is that $\Pi^1_{\dots}$ consider camera measurements while $\Pi^2_{\dots}$ does not. Parameters from the INS, GNSS and

compass are equal in both simulations:

$$\Pi^1_{INS} = \Pi^2_{INS} = \Big[w_{fa} = 4 \cdot 10^{-3} \quad , \quad b_a = \begin{bmatrix} 4 \cdot 10^{-2} \\ 4 \cdot 10^{-2} \\ 4 \cdot 10^{-2} \end{bmatrix} \quad , \quad w_{ba} = 1 \cdot 10^{-4} \tag{7.63}$$

$$w_{\omega g} = 10^{-4} \quad , \quad b_g = \begin{bmatrix} 3 \cdot 10^{-3} \\ 3 \cdot 10^{-3} \\ 3 \cdot 10^{-3} \end{bmatrix} \quad , \quad w_{bg} = 10^{-5} \Big] \tag{7.64}$$

$$\Pi^1_{GNSS} = \Pi^2_{GNSS} = \Big[p^s_{sa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \delta\rho_{rc} = 0 \quad , \quad w_{\rho p} = 10.0 \quad , \quad \delta\dot{\rho}_{rc} = 0 \quad , \quad w_{\rho v} = 1.0 \Big] \tag{7.65}$$

$$\Pi^1_{CMPS} = \Pi^2_{CMPS} = \Big[p^s_{s\psi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad R^s_{\psi} = I_{3\times 3} \quad , \quad w_{\psi} = 0.08726646259971647 \Big] \tag{7.66}$$

Consider the following set of camera parameters Π^1_{CAM}:

$$\Pi^1_{CAM} = \Big[[f_u = 790.0, f_v = 750.0, u_0 = 960.0, v_0 = 440.0], \tag{7.67}$$

$$p^s_{sc} = \begin{bmatrix} 42.0 \\ -25.5 \\ 57.51 \end{bmatrix} \quad , \quad R^s_c = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \Big] \tag{7.68}$$

Finally, consider the following sets of parameters Π^1_{LCI} and Π^2_{LCI} for the *loosely coupled* integrator:

$$\Pi^2_{LCI} = \Big[\hat{w}_{\omega g} = 10^{-4} \quad , \quad \hat{w}_{fa} = 4 \cdot 10^{-3} \quad , \quad \hat{w}_{ba} = 10^{-4} \quad , \quad \hat{w}_{bg} = 10^{-5} \tag{7.69}$$

$$\hat{p}^s_{sa} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad , \quad \hat{w}_{\rho p} = 10.0 \quad , \quad \hat{w}_{\rho v} = 1.0 \quad , \quad \hat{p}^s_{s\psi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.70}$$

$$\hat{R}^s_{\psi} = I_{3\times 3} \quad , \quad \hat{w}_{\psi} = 0.0873 \Big] \tag{7.71}$$

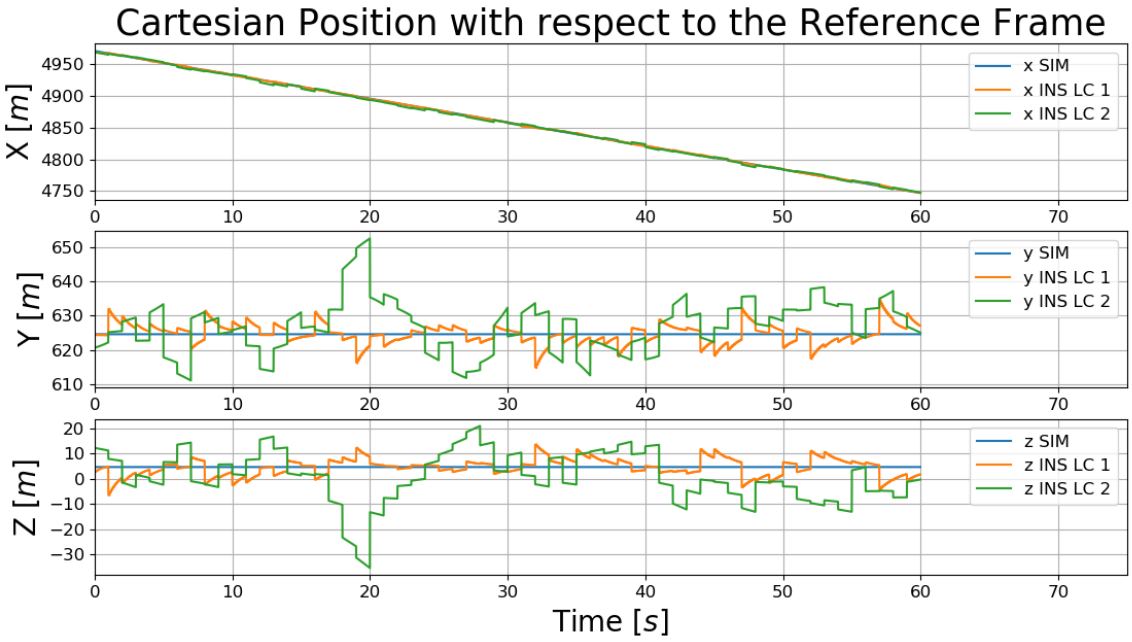
$$\Pi^1_{LCI} = \Big[\Pi^2_{LCI}, [\hat{f}_u = 790.0, \hat{f}_v = 750.0, \hat{u}_0 = 960.0, \hat{v}_0 = 440.0], \tag{7.72}$$

$$\hat{p}^s_{sc} = \begin{bmatrix} 42.0 \\ -25.5 \\ 57.51 \end{bmatrix} \quad , \quad \hat{R}^s_c = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{7.73}$$

$$\hat{w}_{c\psi} = 0.00174533 \quad , \quad \hat{w}_{cp} = 0.01\Big] \tag{7.74}$$

Figures 66, 68 and 70 presents *tracker* outputs computed from parameters $\Pi^1_{...}$ and $\Pi^2_{...}$. In this set of Figures, blue lines represents the simulation outputs, orange lines represents the outputs computed from the integrator with parameters $\Pi^1_{...}$ and green lines represents the outputs computed from the integrator with parameters $\Pi^2_{...}$. Figures 67, 69 and 71 shows the errors between *tracker* outputs and simulation outputs. Blue lines represents errors from the integrator with parameters $\Pi^1_{...}$ while orange lines represents errors from the integrator with parameters from $\Pi^2_{...}$.

Figure 66: Cartesian Position Computed from Tracker with Different Parameters

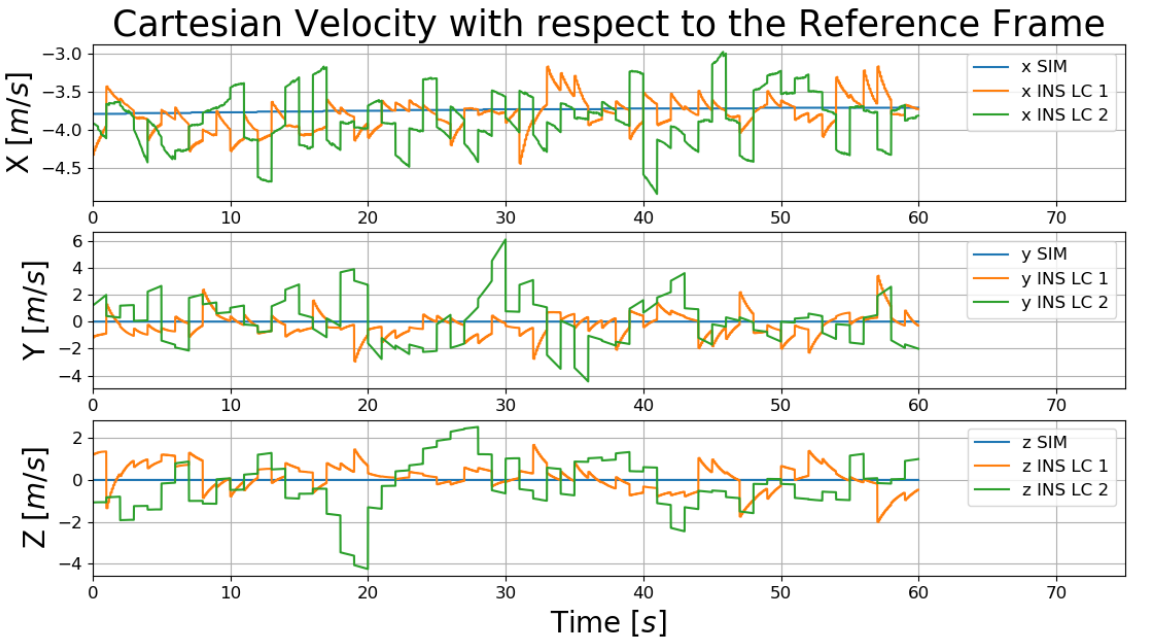


Source: Author

Figure 67: Cartesian Position Error Computed from Tracker with Different Parameters

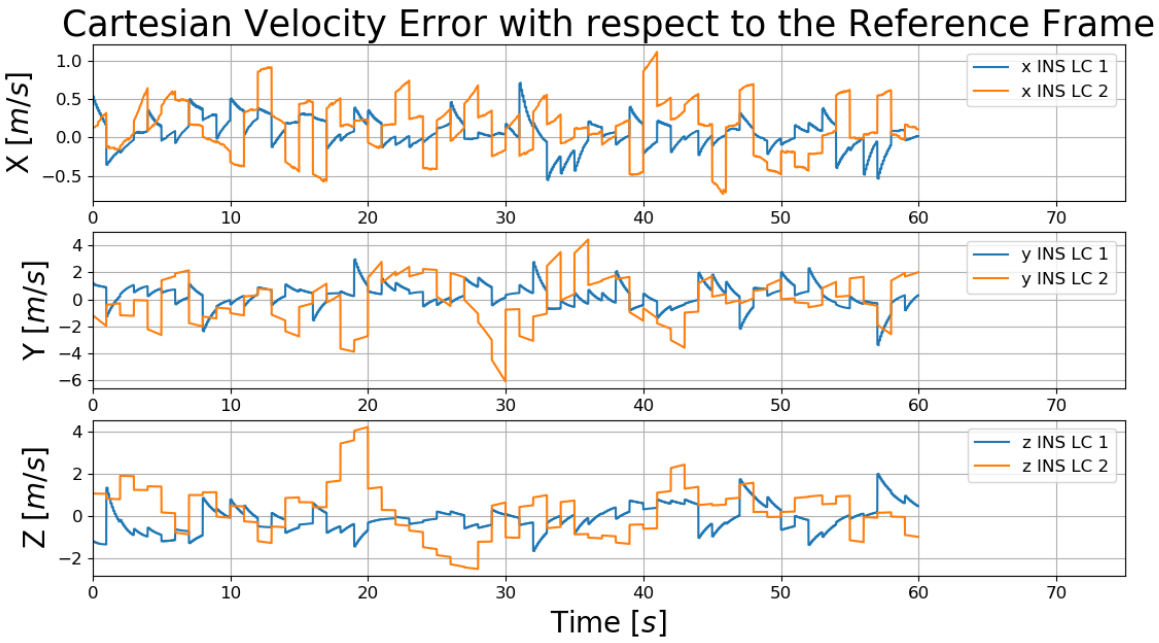
Source: Author

Figure 68: Cartesian Velocity Computed from Tracker with Different Parameters



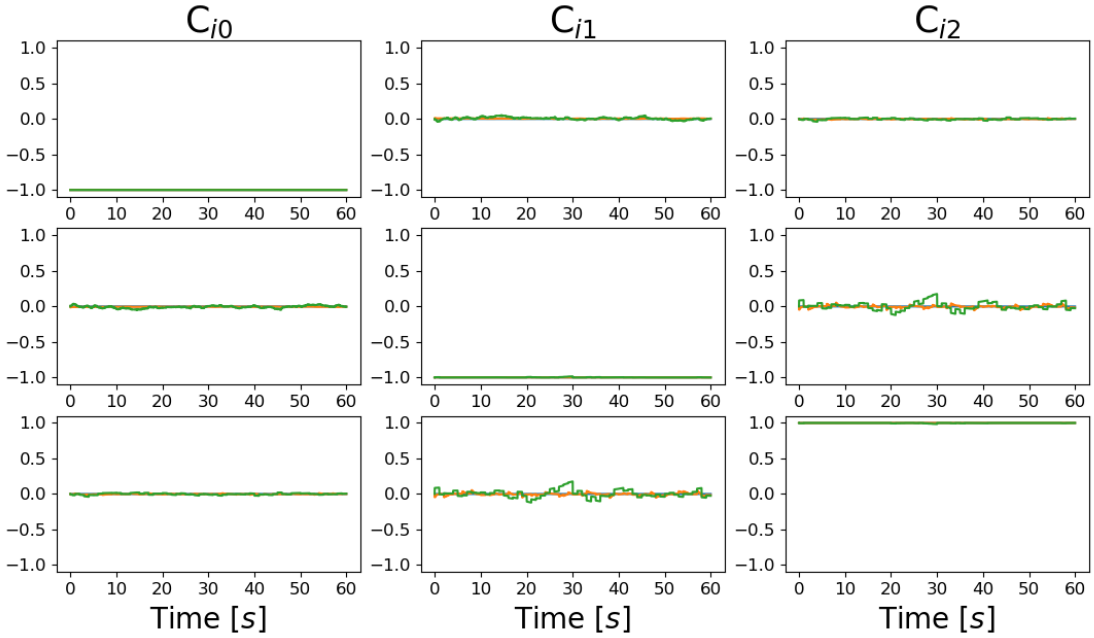
Source: Author

Figure 69: Cartesian Velocity Error Computed from Tracker with Different Parameters



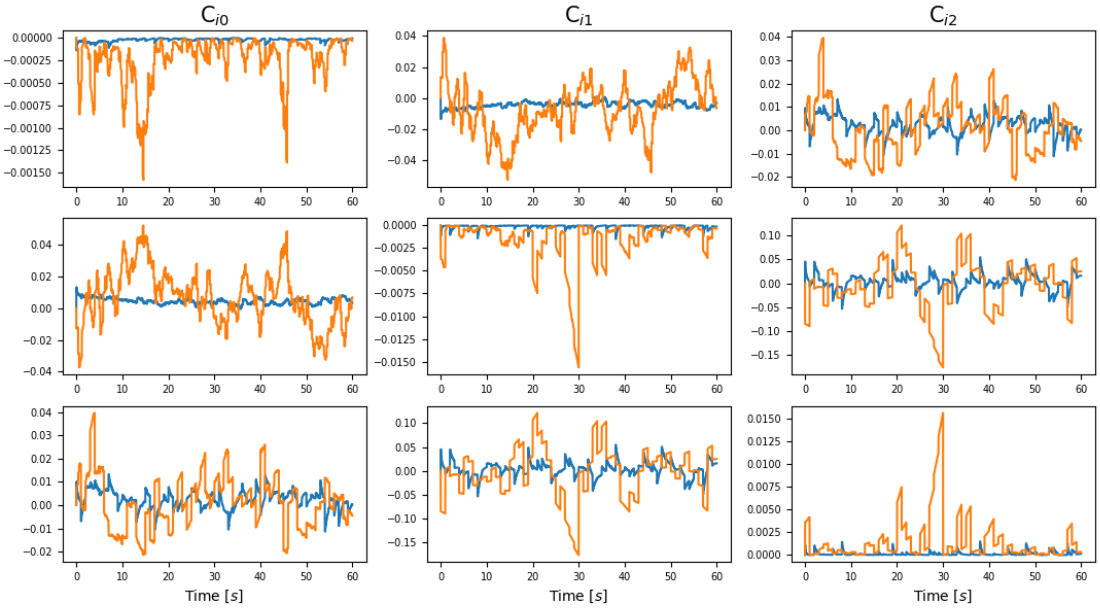
Source: Author

Figure 70: Attitude Computed from Tracker with Different Parameters



Source: Author

Figure 71: Attitude Error Computed from Tracker with Different Parameters



Source: Author

Errors from the *tracker* represented by the first set of parameters $\Pi^1_{...}$ are smaller than $\Pi^2_{...}$. This observation indicates that the incorporation of camera measurements into the loosely coupled integration is interesting to be pursued. Note that in this simulation camera measurements are simulated assuming that camera parameters and reference points are perfectly known. Although this setup is interesting for a preliminary evaluation of the loosely coupled integration algorithm, a more realistic setup is needed before the incorporation of such algorithm into operational equipment.

8 VISUAL AUGMENTATION

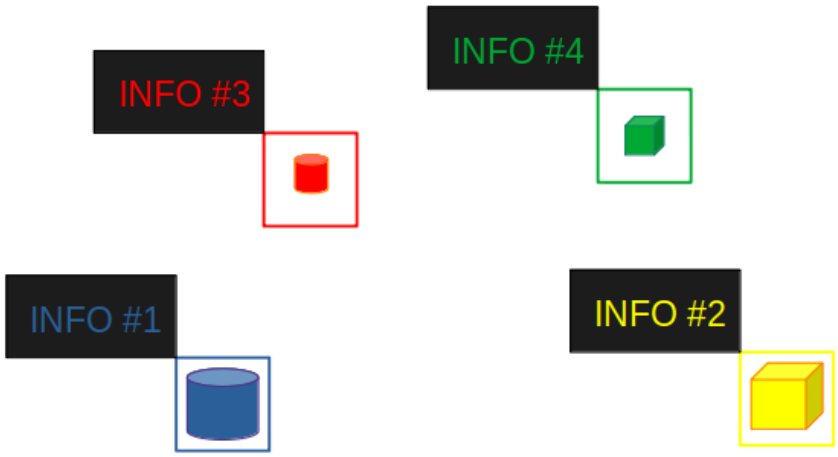
This chapter discusses applications for displaying virtual elements in maritime scenes. These features are concerned with the visual presentation of navigational information. Most of the proposed features assumes that external information about the scene is available from other systems of the ship. It is assumed that the installation parameters from the previous chapter are available. The methods are pursued for assisting ship operators in an efficient perception of their surroundings and immediate tasks.

8.1 Simple Highlight

8.1.1 Feature Description

A simple form of highlight is the synthesis of a rectangle around a given point from the navigation scene. In order to facilitate the understanding of what is being highlighted, it is interesting to render an information summary next to the point projection in the navigation scene, as exemplified by Figure 72.

Figure 72: Four simple highlights

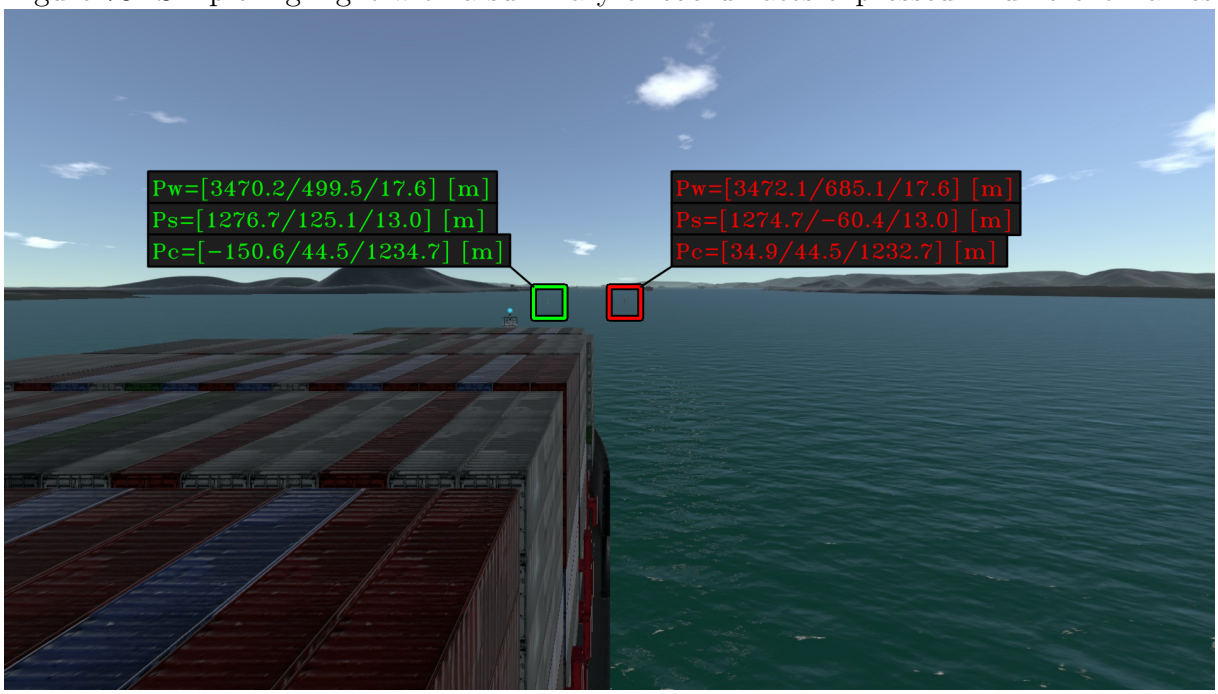


Source: Author

8.1.2 Implementation with Data from Ship Maneuvering Simulator

In the simulation experiment, as the position of each buoy in the camera coordinate system is known at all times, it is possible to determine each buoy corresponding projection in the camera scene. These coordinates may be used for an automatic highlight of such obstacles. Figure 73 illustrates a simple highlight with obstacles coordinates as information elements.

Figure 73: Simple highlight with a summary of coordinates expressed in different frames



Source: Author

8.1.3 Implementation with Data from Real Experiments

Similarly as before, buoys are used for the implementation of the proposed feature in videos from real experiments. If the buoy position is not known as in the case of the experiment, the corresponding region can be automatically obtained from obstacle detection algorithms such as the one described in the previous chapter. A simple highlight with just two attributes is shown in Figure 74.

Figure 74: Highlight of information from buoy

Source: Author

8.2 Zoom View

8.2.1 Feature Description

For very far obstacles, the corresponding projection in the image scene may be very difficult to visualize. Thus, another useful feature that can be implemented consists in the presentation of a *zoom* region alongside the image scene. If relative coordinates of an obstacle are available from an external system of the ship, the corresponding projection region can be automatically *amplified* and presented next to the obstacle projection. Alternatively, such *zoom views* may be defined from operators inputs.

8.2.2 Implementation with Data from Ship Maneuvering Simulator

Figure 75 illustrates the proposed feature with buoys from the simulation. As the position of the buoys are known alongside the ship position and orientation, the corresponding projection region in the image can be determined and amplified automatically. The amplified region of the image can assist operators in the visual identification of far obstacles.

Figure 75: Example of zoom view for far obstacles

Source: Author

8.2.3 Implementation with Data from Real Experiments

If the relative position of the obstacle is unknown, the region to be amplified can be defined from an obstacle tracker feature such as from the previous section. After an initial identification by the user, the corresponding region of the obstacle may be determined automatically. Figure 76 illustrates the proposed feature with buoys from the real experiments.

Figure 76: Example of zoom view for far obstacles

Source: Author

8.3 Planar Rectangular Highlight

8.3.1 Feature Description

Consider a plane Π described with a *plane* coordinate system S_Π as in Equation 4.17. Let $[\vec{x}_\Pi, \vec{y}_\Pi, \vec{z}_\Pi]$ be the axes of the plane frame S_Π and let $p^c_{c\Pi}$ be a point belonging to plane Π expressed with coordinates in the camera coordinate system.

$$p^c_{cP_\pi} = p^c_{c\Pi} + \mu_1 \vec{x}^c_\Pi + \mu_2 \vec{y}^c_\Pi = p^c_{c\Pi} + \begin{bmatrix} \vec{x}^c_\Pi & \vec{y}^c_\Pi & \vec{z}^c_\Pi \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ 0 \end{bmatrix} \tag{8.1}$$

Note that μ_1 and μ_2 from Equation 8.1 represents points coordinates in the plane frame. The origin of this plane coordinate system is the point $p^c_{c\Pi}$, while the orientation is given by vectors $[\vec{x}^c_\Pi, \vec{y}^c_\Pi, \vec{z}^c_\Pi]$. It is possible to define a rectangular grid in the plane frame

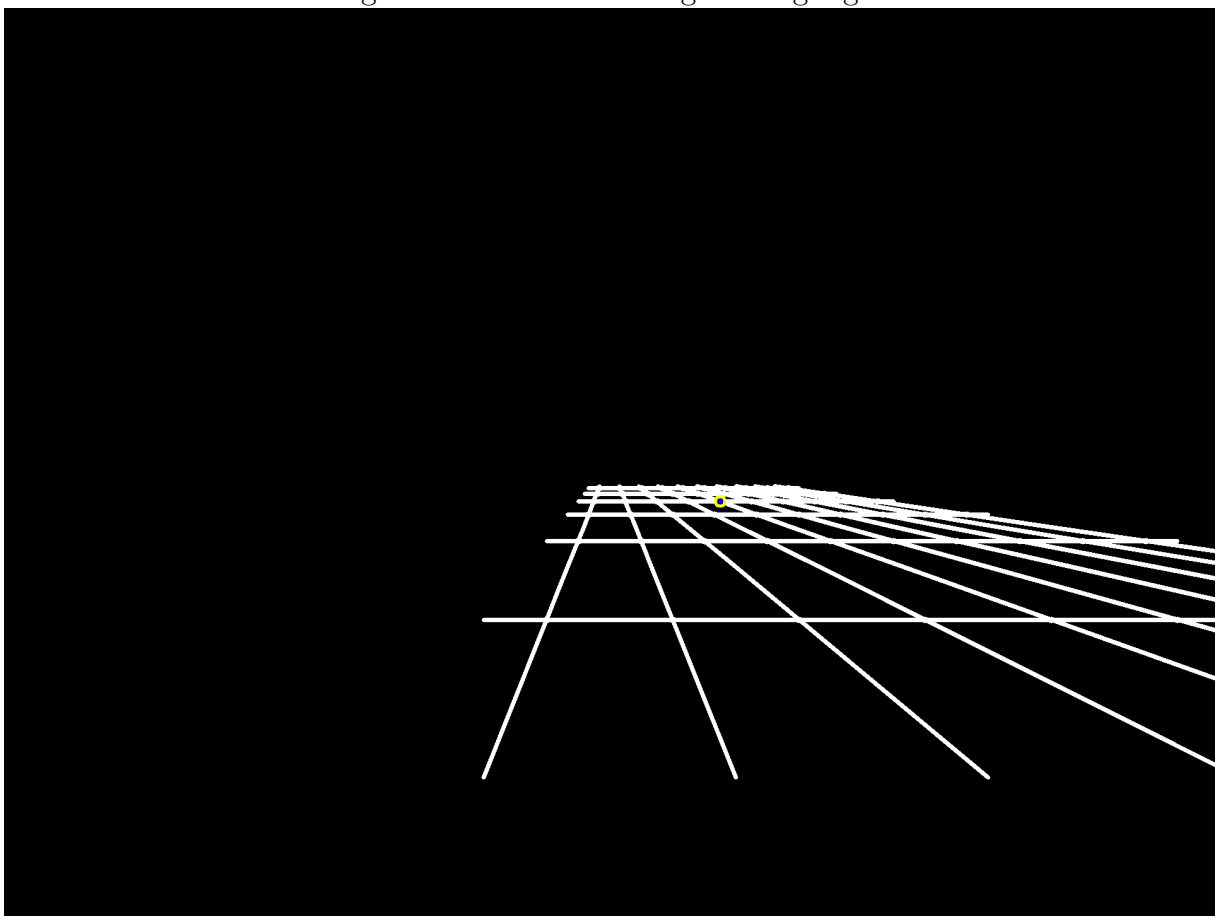
by iterating different $[\mu_1, \mu_2]$ according to vertexes of different rectangles.

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} \eta_x \Delta L_x \\ \eta_y \Delta L_y \end{bmatrix} \quad , \quad -n_x \leq \eta_x \leq n_x \quad , \quad -n_y \leq \eta_y \leq n_y \quad , \quad \eta_x, n_x, \eta_y, n_y \in \mathbb{Z} \tag{8.2}$$

where L_x and L_y are the size of each rectangle. Note that η_x and η_y are iterated according to n_x and n_y, which are the number of rectangles in each direction. The connection of adjacent points computed from different η_x and η_y yields a rectangular grid. Figure 77 illustrates a rectangular highlight in a standalone simulation for intrinsic parameters $[f_u, f_v, u_0, v_0] = [1000, 1000, 960, 720]$ and following plane parameters:

$$p^c_{c\Pi} = \begin{bmatrix} 70.0 \\ 25.0 \\ 400.0 \end{bmatrix} \ , \ \vec{x}^c_{\Pi} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ , \ \vec{y}^c_{\Pi} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \ , \ \vec{z}^c_{\Pi} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

Figure 77: Planar Rectangular Highlight



Source: Author

8.3.2 Implementation with Data from Ship Maneuvering Simulator

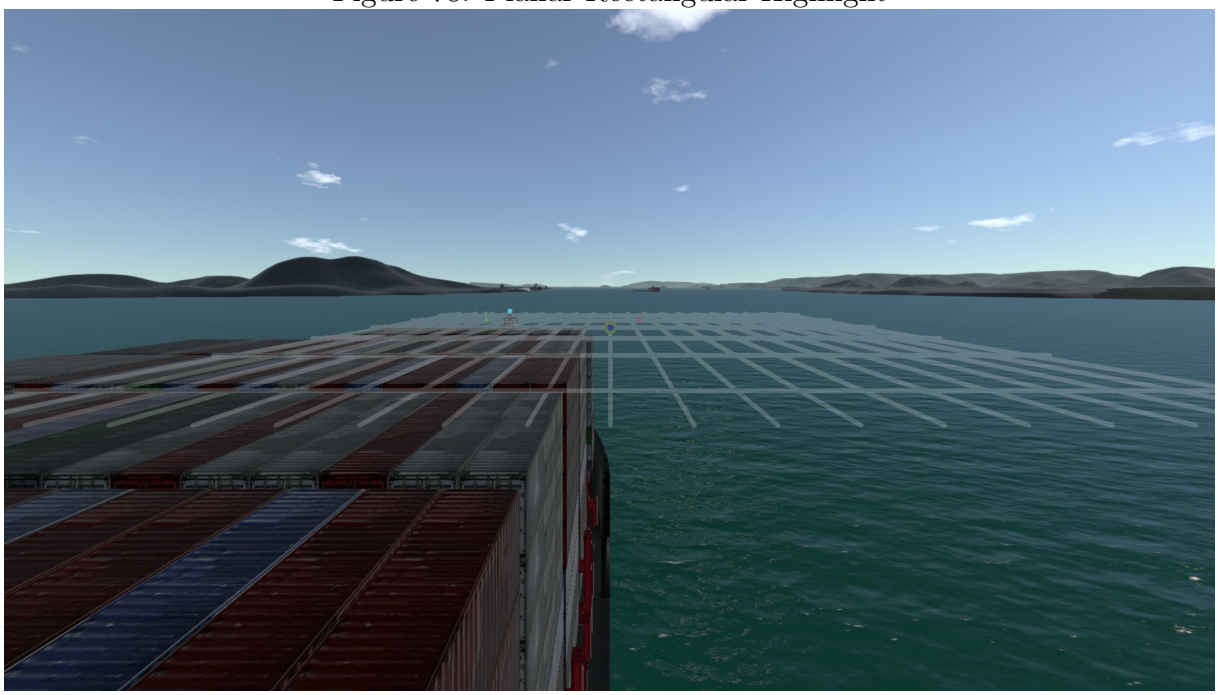
Consider a rectangular grid with points belonging to a plane at the surface of the sea. Assume that the distance h_w between the camera and the water is known from the installation procedure described in the previous chapter. By overlaying such lines of the plane in the image of the scene, the corresponding region is highlighted and better perceived spatially. Figure 78 illustrates a planar rectangular grid with the following parameters:

$$p_{c\Pi}^{c} = \begin{bmatrix} 0 \\ h_w \\ 500.0 \end{bmatrix} = \begin{bmatrix} 0 \\ 44.2 \\ 500.0 \end{bmatrix} \;,\; \vec{x}_{\Pi}^{c} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \;,\; \vec{y}_{\Pi}^{c} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \;,\; \vec{z}_{\Pi}^{c} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\Delta L_x = 100.0 \quad , \quad n_x = 3$$

$$\Delta L_y = 25.0 \quad , \quad n_y = 10$$

Figure 78: Planar Rectangular Highlight



Source: Author

The center of the rectangular grid is the point $p_{c\Pi}^{c}$. For each direction of the plane given by vectors $\vec{x}_{\Pi}^{c}$ and $\vec{y}_{\Pi}^{c}$, n_x and n_y determines the number of divisions of the rectangular grid. The size of each division is given by ΔL_x and ΔL_y.

Note that is possible to estimate the relative position of visible obstacles according to the intersection of its projection with each auxiliary line from the grid.

8.3.3 Implementation with Data from Real Experiments

Similarly as before, consider a rectangular grid with points belonging to a plane at the surface of the sea. The distance h_w between the camera and the surface of the sea is assumed to be known from the installation of the previous chapter. Figure 79 illustrates a planar rectangular grid with the following parameters:

$$p^c_{c\Pi} = \begin{bmatrix} 0 \\ h_w \\ 500 \end{bmatrix} = \begin{bmatrix} 0 \\ 32 \\ 500 \end{bmatrix} \;,\; \vec{x}^c_{\Pi} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \;,\; \vec{y}^c_{\Pi} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \;,\; \vec{z}^c_{\Pi} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\Delta L_x = 100.0 \quad , \quad n_x = 3$$

$$\Delta L_y = 25.0 \quad , \quad n_y = 10$$

Figure 79: Planar Rectangular Highlight

Source: Author

Note that considering the world frame defined by Equation 6.10, the plane shown in

Figure 79 would match the sea plane only if the camera is installed without considerable errors and if the ship is aligned with the WCS. More generally, the synthetic plane will match the sea plane if $\vec{z}_{\Pi}^{c}$ is aligned with the direction of the gravity, which may be approximated by axis $\vec{z}_{n}^{c}$ from the local navigation frame. Figure 80 shows a planar rectangular grid with the following parameters:

$$p_{c\Pi}^{c} = \begin{bmatrix} 0.9996 & -0.0187 & -0.0231 \\ 0.0164 & 0.9950 & -0.0980 \\ 0.0248 & 0.0976 & 0.9949 \end{bmatrix} \begin{bmatrix} 0 \\ 32 \\ 500 \end{bmatrix} = \begin{bmatrix} -12.152 \\ -17.157 \\ 500.582 \end{bmatrix}$$

$$\vec{x}_{\Pi}^{c} = \begin{bmatrix} -0.0231064 \\ -0.0979969 \\ 0.9949184 \end{bmatrix} , \ \vec{y}_{\Pi}^{c} = \begin{bmatrix} -0.9995578 \\ -0.0163644 \\ -0.0248260 \end{bmatrix} , \ \vec{z}_{\Pi}^{c} = \begin{bmatrix} 0.0187141 \\ -0.9950522 \\ -0.0975755 \end{bmatrix}$$

$$\Delta L_x = 100.0 \quad , \quad n_x = 3$$

$$\Delta L_y = 25.0 \quad , \quad n_y = 10$$

Figure 80: Planar Rectangular Highlight

Source: Author

8.4 Planar Circular Highlight

8.4.1 Feature Description

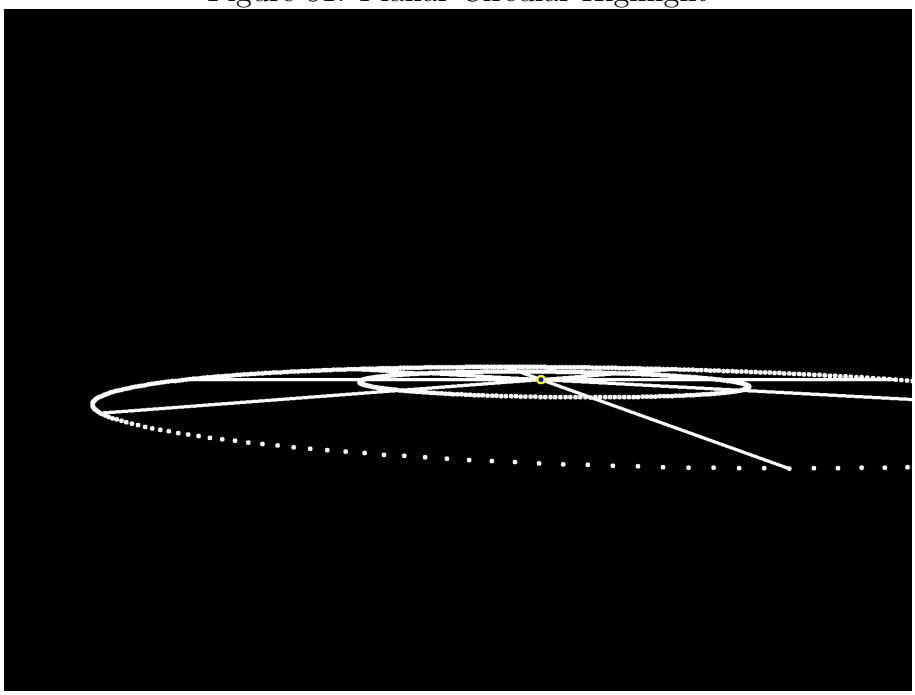
Consider the previous plane definition described by a *plane* frame S_{Π}. Instead of computing coordinates $[\mu_1, \mu_2]$ as vertices from rectangles yielding a rectangular grid, the definition of coordinates $[\mu_1, \mu_2]$ according to circles yields a planar circular highlight:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} r_i \cos(\theta) \\ r_i \sin(\theta) \end{bmatrix} \qquad -\pi \leq \theta \leq \pi$$

Let $r_c = [r_1, ..., r_i, ..., r_M]$ denotes the radius of different circles to be highlighted in the region of interest and $d\theta$ denotes each incremental step for the angle θ. Additionally, each circle from the highlight is divided in N_θ sections. Figure 81 illustrates a planar circular highlight in a standalone simulation with the following plane parameters:

$$p^c_{c\Pi} = \begin{bmatrix} 70.0 \\ 25.0 \\ 400.0 \end{bmatrix} \ , \ \vec{x}^{\,c}_{\Pi} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ , \ \vec{y}^{\,c}_{\Pi} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \ , \ \vec{z}^{\,c}_{\Pi} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \ , \ r_c = [150.0, 300.0] \ , \ N_\theta = 8$$

Figure 81: Planar Circular Highlight



Source: Author

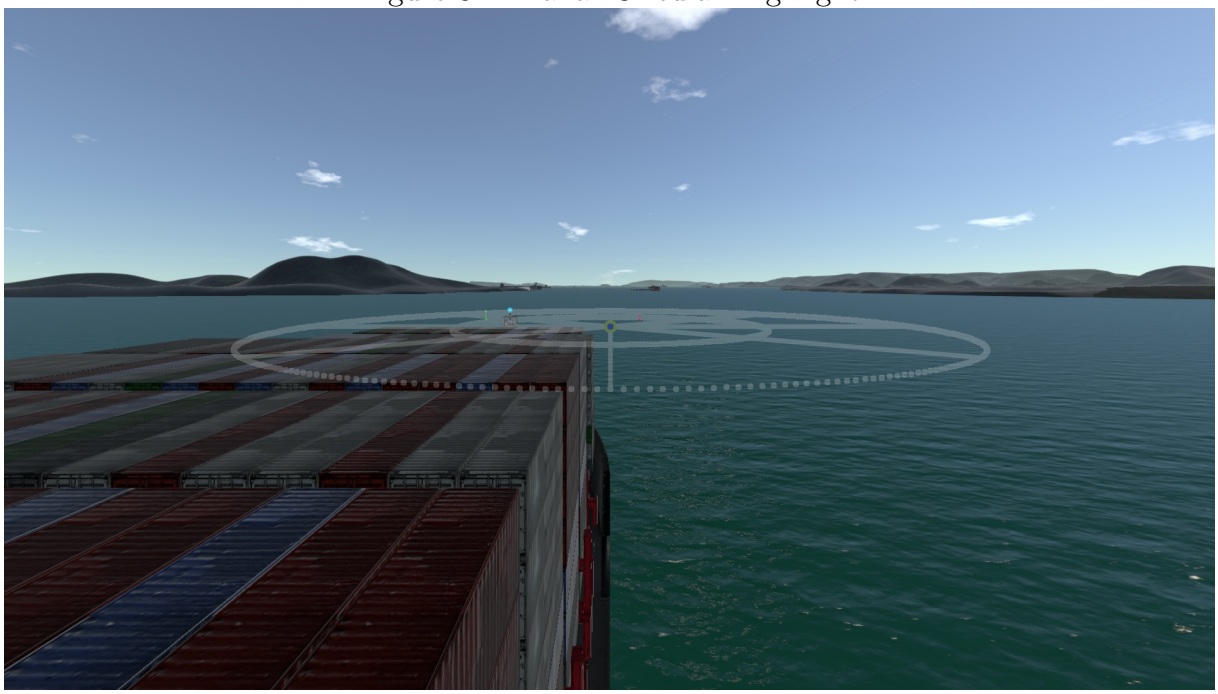
8.4.2 Implementation with Data from Ship Maneuvering Simulator

Figure 82 illustrates a planar circular grid with the following parameters:

$$p_{c\Pi}^{c} = \begin{bmatrix} 0 \\ h_w \\ 500.0 \end{bmatrix} = \begin{bmatrix} 0 \\ 44.2 \\ 500.0 \end{bmatrix} \ , \ \vec{x}_{\Pi}^{c} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ , \ \vec{y}_{\Pi}^{c} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \ , \ \vec{z}_{\Pi}^{c} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$r_c = [150.0, 300.0] \quad , \quad d\theta = 1^{\circ} = \frac{\pi}{180} \quad , \quad N_\theta = 8$$

Figure 82: Planar Circular Highlight



Source: Author

8.4.3 Implementation with Data from Real Experiments

Figure 83 illustrates a planar circular grid with the following parameters:

$$p_{c\Pi}^{c} = \begin{bmatrix} 0 \\ h_w \\ 500 \end{bmatrix} = \begin{bmatrix} 0 \\ 32 \\ 500 \end{bmatrix} \ , \ \vec{x}_{\Pi}^{c} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ , \ \vec{y}_{\Pi}^{c} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \ , \ \vec{z}_{\Pi}^{c} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

$$r_c = [150.0, 300.0] \quad , \quad d\theta = 1^\circ = \frac{\pi}{180} \quad , \quad N_\theta = 8$$

Figure 83: Planar Circular Highlight

Source: Author

Similarly as before, the highlighted plane shown in Figure 83 matches the surface of the sea if the camera is installed without considerable errors and if the ship axes are aligned with the WCS. Figure 84 presents a planar circular highlight computed with the following plane parameters:

$$p^c_{c\Pi} = \begin{bmatrix} 0.9996 & -0.0187 & -0.0231 \\ 0.0164 & 0.9950 & -0.0980 \\ 0.0248 & 0.0976 & 0.9949 \end{bmatrix} \begin{bmatrix} 0 \\ 32 \\ 500 \end{bmatrix} = \begin{bmatrix} -12.152 \\ -17.157 \\ 500.582 \end{bmatrix}$$

$$\vec{x}^c_\Pi = \begin{bmatrix} -0.0231064 \\ -0.0979969 \\ 0.9949184 \end{bmatrix} \ , \ \vec{y}^c_\Pi = \begin{bmatrix} -0.9995578 \\ -0.0163644 \\ -0.0248260 \end{bmatrix} \ , \ \vec{z}^c_\Pi = \begin{bmatrix} 0.0187141 \\ -0.9950522 \\ -0.0975755 \end{bmatrix}$$

$$r_c = [150.0, 300.0] \quad , \quad d\theta = 1^\circ = \frac{\pi}{180} \quad , \quad N_\theta = 8$$

Figure 84: Planar Rectangular Highlight

Source: Author

8.5 Expected Trajectory Visualizer

8.5.1 Feature Description

Consider an arbitrary point Q in the navigation scene described by its position p^c_{cQ} with respect to the camera frame. Further, let v^c_{cQ} be its velocity with respect to the camera frame. It is possible to compute a simple estimation for the future trajectory of point Q by the extrapolation of its velocity v^c_{cQ} around its position p^c_{cQ}. Let $p^c_{cQ} = [p^{c,\chi}_{cQ}, p^{c,\gamma}_{cQ}, p^{c,\kappa}_{cQ}]^T$ and $v^c_{cQ} = [v^{c,\chi}_{cQ}, v^{c,\gamma}_{cQ}, v^{c,\kappa}_{cQ}]^T$ denotes the instantaneous position and velocity of the obstacle with respect to the camera frame. A simple expected trajectory $p^c_{traj}(t)$ for the obstacle can be computed with the assumption that the obstacle velocity v^c_{cQ} is

approximately constant for an interval $t \in [0, \Delta T_{traj}]$:

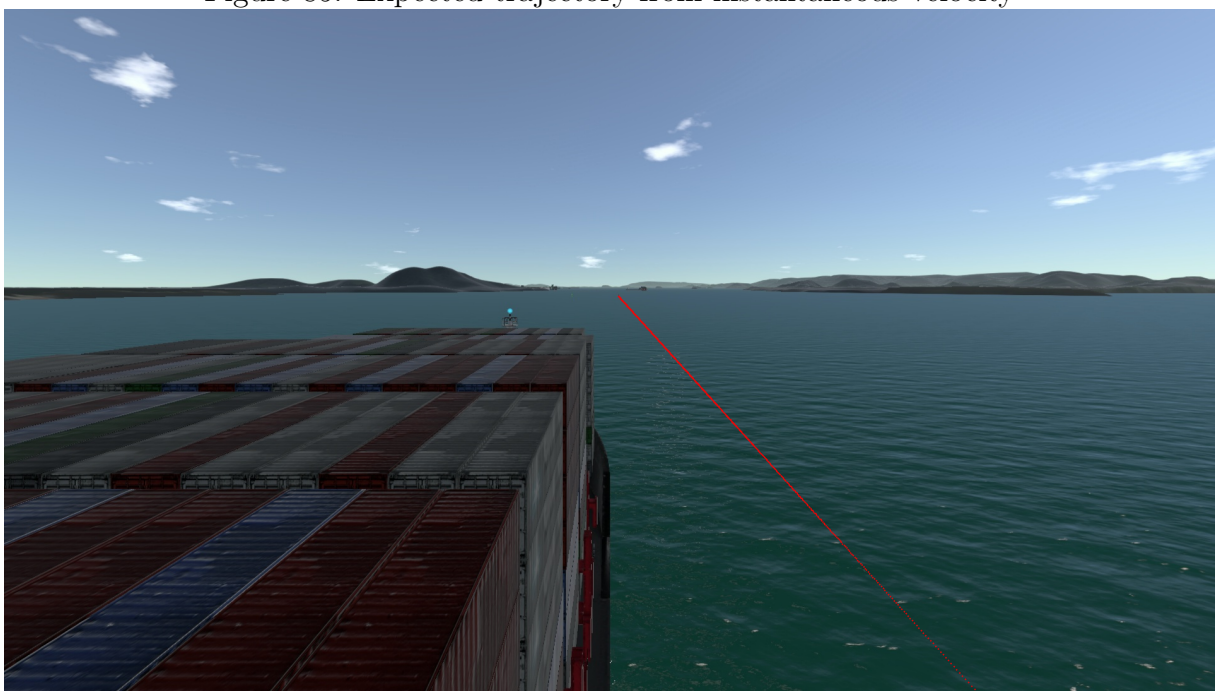
$$p^{c}_{traj}(t) \approx \begin{bmatrix} p^{c,\chi}_{cQ} \\ p^{c,\gamma}_{cQ} \\ p^{c,\kappa}_{cQ} \end{bmatrix} + t \begin{bmatrix} v^{c,\chi}_{cQ} \\ v^{c,\gamma}_{cQ} \\ v^{c,\kappa}_{cQ} \end{bmatrix} \qquad 0 \leq t \leq \Delta T_{traj}$$

A set of points belonging to the trajectory may be defined with respect to the camera frame. Then, the corresponding image projections for such points can be shown on top of the image of the scene. Note that if an obstacle is represented in another known coordinate system, it is possible to transform the obstacle position and velocity to the camera frame before computing the expected trajectory with the same expressions.

8.5.2 Implementation with Data from Ship Maneuvering Simulator

Considering that the ship velocity is known and assuming that the buoy does not move, the relative velocity of the buoy can be computed from the simulation outputs. Figure 85 presents the corresponding image projections for the expected trajectory of the buoy in the scene of the onboard camera.

Figure 85: Expected trajectory from instantaneous velocity



Source: Author

Note that such estimations are particularly useful if the ship and the obstacle are

moving with constant velocity. If there are significant rotations in the relative trajectory of the obstacle, the computed expected trajectory will not coincide with the actual trajectory of the obstacle in the image scene.

8.5.3 Implementation with Data from Real Experiments

Similarly as before, Figure 86 illustrates this feature assuming that the ship velocity is known and that the buoy does not move.

Figure 86: Expected trajectory from instantaneous velocity

Source: Author

9 DISCUSSION

This chapter discusses the utilization of aforementioned methods for navigational assistance. Initially, augmented visualizations that may assist ship operators are presented. The integration of the proposed MAR system with typical onboard navigational equipment is briefly discussed in the subsequent section. Finally, further research is suggested to formally define accuracy requirements and operational capabilities for the proposed MAR system.

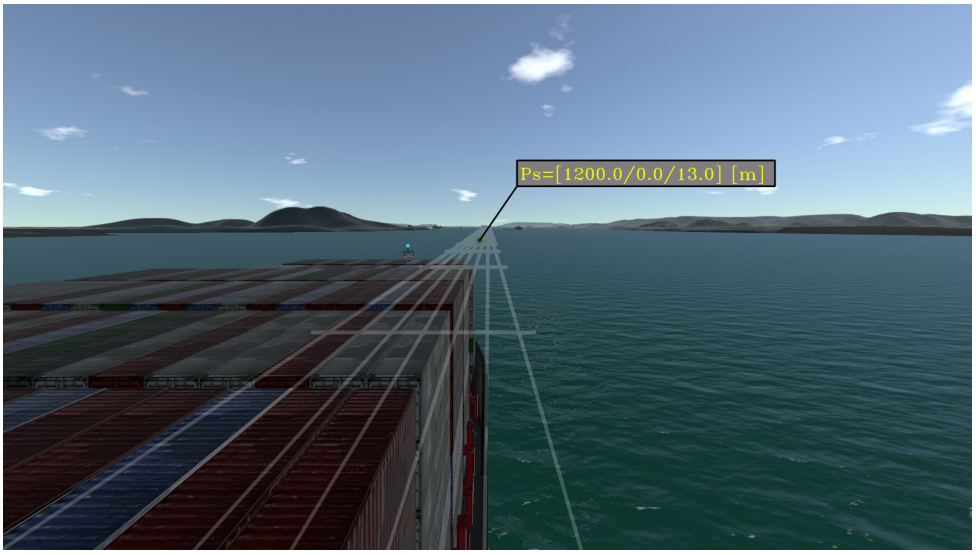
9.1 Assistance by Visual Augmentation

Visualizations from this section assumes a scenario of navigation in restricted waters with fixed obstacles as presented in [LEITE et al., 2022]. In such scenarios, in order to prevent collisions, it is important that operators acknowledge the relative position of surrounding obstacles along the expected trajectory of the ship. Geometrical information regarding the ship and the navigation scene is assumed to be known. Three types of navigational assistance situations are discussed throughout this section: *route highlight*, *obstacle highlight* and *spatial perception*.

9.1.1 Route Highlight

In the previous simulation, the ship is navigating in a straight line with known velocity. The camera position and orientation with respect to the ship coordinate system is known alongside the ship state with respect to the world coordinate system. Thus, it is possible to define coordinates for points at the surface of the sea in the camera coordinate system. In such situations, a rectangular highlight in front of the ship could be defined to highlight the expected trajectory for the ship. The rectangle width is defined accordingly to the ship course and dimensions. Figure 87 exemplifies this implementation in the previous simulation experiment with a rectangular highlight delimiting the expected trajectory of the ship.

Figure 87: Rectangular highlight showing expected trajectory for the ship in the simulation



Source: [LEITE et al., 2022]

A similar rectangular highlight may be drawn on top of the video from the real experiment as the ship navigates with approximately constant velocity in a straight line. Figure 88 illustrates this implementation.

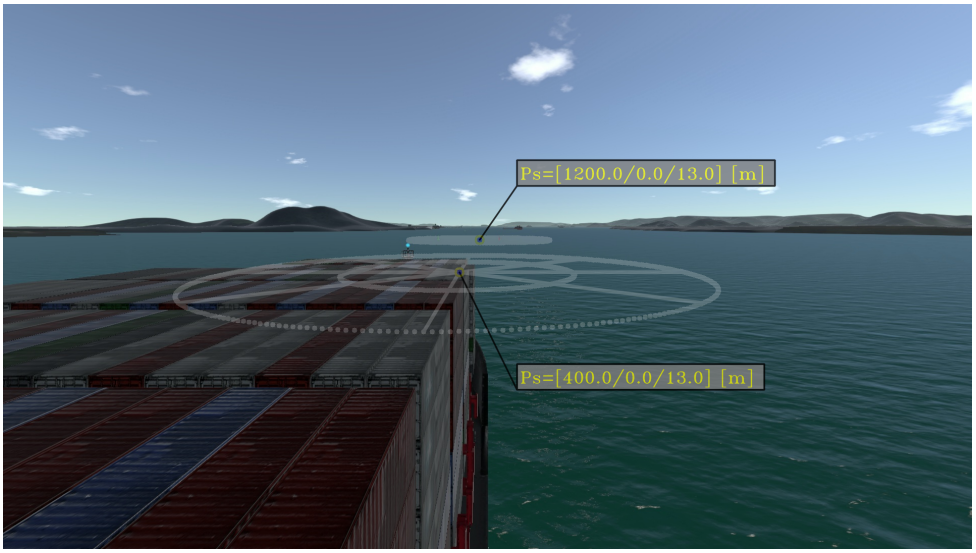
Figure 88: Rectangular highlight showing expected trajectory for the ship in the reality

Source: [LEITE et al., 2022]

Another useful representation for the expected trajectory of the ship is in the form of waypoints. Each waypoint may be projected into the image scene with a circular highlight.

As the ship is traveling with constant velocity, its trajectory may be represented with colinear waypoints. Figure 89 illustrates the visualization with two waypoints in front of the ship. Figure 90 illustrates a similar implementation in the recorded video from the aforementioned real experiment.

Figure 89: Circular highlight representing waypoints of the expected trajectory for the ship in the simulation



Source: [LEITE et al., 2022]

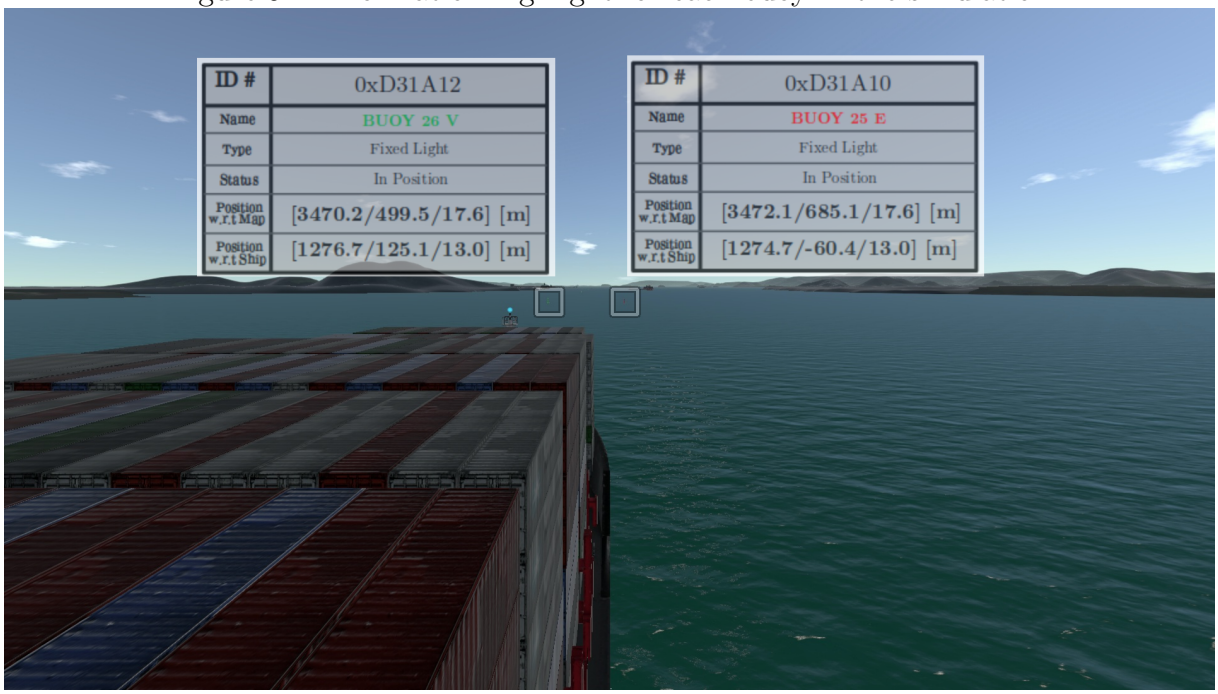
Figure 90: Circular highlight representing waypoints of the expected trajectory for the ship in the reality

Source: [LEITE et al., 2022]

9.1.2 Obstacle Highlight

Assume that each obstacle can be characterized by a set of N_a attributes such as *ID*, *name*, *type*, *status* and *position*. Further, assume that the corresponding obstacle attributes are automatically rendered into a graphical summary. These attributes can be summarized and presented accordingly to their relative position in the scene to assist in the perception of these obstacles as shown in Figure 91.

Figure 91: Information highlight for each buoy in the simulation



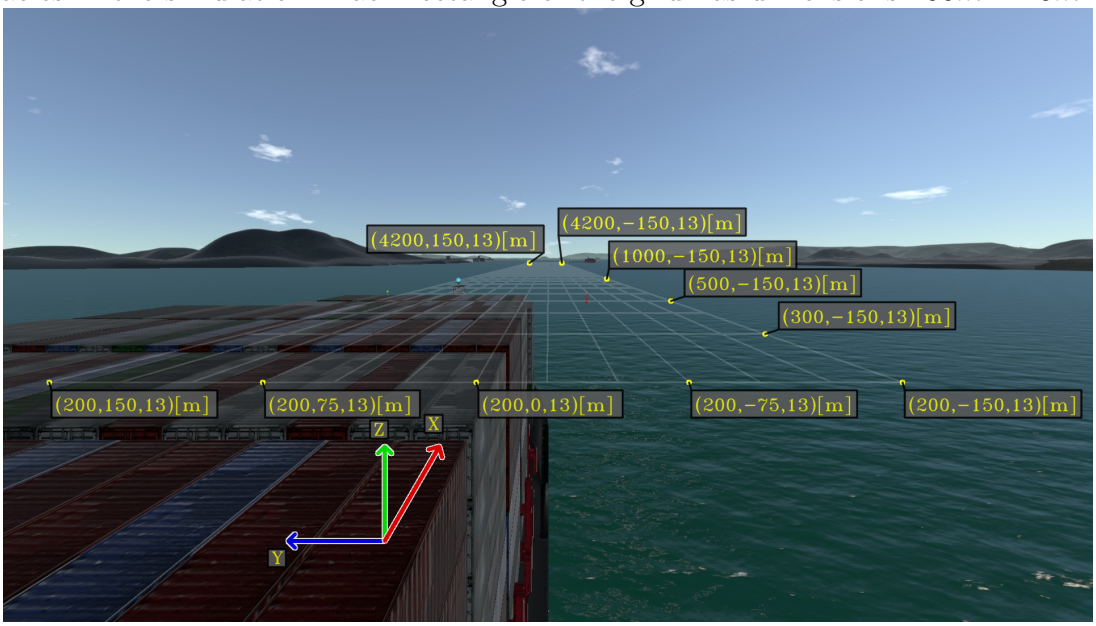
Source: [LEITE et al., 2022]

Note that an optimal definition of pertinent attributes considering each ship task may be developed in view of requirements for user interfaces. These usability considerations are beyond the scope of the present work.

9.1.3 Spatial Perception

In cases in which external information about surrounding obstacles is not available, it is possible to estimate the relative position of visible obstacles in the image scene with planar highlights. Figure 92 shows a rectangular highlight with points described in the ship coordinate system. An estimation for the buoy relative position may be inferred by the intersection of the buoy projection in the image scene with each auxiliary line of the rectangular grid.

Figure 92: Auxiliary rectangular grid for estimating the relative position of surrounding obstacles in the simulation. Each rectangle of the grid has dimensions $100m \times 25m$.



Source: [LEITE et al., 2022]

Figure 93 shows a rectangular highlight in the real experiments with a grid of dimensions $100m \times 25m$.

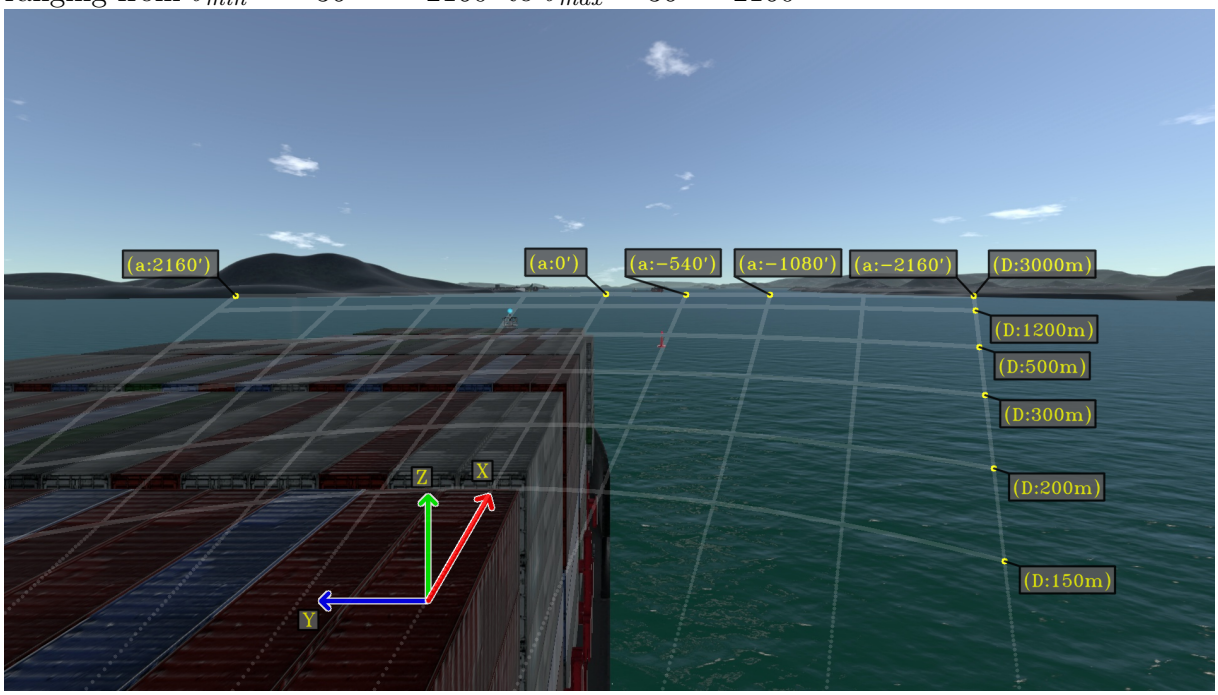
Figure 93: Auxiliary rectangular grid for estimating the relative position of surrounding obstacles in the reality. Each rectangle of the grid has dimensions $100m \times 25m$.

Source: [LEITE et al., 2022]

Note that the right buoy intersects the rectangular grid at the sixth parallel line of the grid, representing an approximate lateral distance of $25 \cdot 6 = 150m$. Correspondingly, the left buoy intersects the rectangular grid at the fifth parallel line of the grid, yielding an approximate lateral distance of $25 \cdot 5 = 125m$. Combining both results yields estimates of $275m$ for the distance between both buoys. Accordingly, the theoretical distance between them is approximately $280m$ when both buoys are in their map position.

Another useful representation for estimating the buoy relative position may be displayed with a circular highlight. Figure 94 shows a circular highlight with points described in the ship coordinate system.

Figure 94: Auxiliary lines described with respect to the ship-frame. Lines are drawn ranging from $\theta_{min} = -36° = -2160'$ to $\theta_{max} = 36° = 2160'$



Source: [LEITE et al., 2022]

Similarly, an estimation for the buoy relative position may be determined from the intersection of its projection with each auxiliary line. From the augmented scene of the above example, it is possible to estimate that the buoy is at an approximate distance of 500 m from the ship with an angle of $-9°$ with respect to the ship $x-$axis.

Note that alternative enhanced scenes may be adapted from the current work towards other navigation scenarios. As the development of optimal visualizations combining pertinent virtual elements depends on the particular operation of the user, further research is suggested to cover other types of standardized navigation operations.

9.2 MAR System Development

Consider a camera rigidly fixed on a ship that navigates through a channel delimited by nautical buoys. The ship position and orientation is assumed to be known at all times with respect to a reference coordinate system. Furthermore, information about the ship surroundings and expected route is assumed to be available from other onboard ship systems. A centralized monitor screen displays the video stream from the camera with virtual assets representing the information from other ship systems.

In order to define operational capabilities of the system, it is interesting to estimate the maximum distance in which an obstacle is still observable by the camera. A simplified analysis may be carried out as follows. Consider a *pinhole* model expressed by the set of intrinsic parameters $[f_u, f_v, u_0, v_0]$. The projection of an arbitrary point Q with position p^c_{cQ} with respect to the camera is given by:

$$p^c_{cQ} = \begin{bmatrix} p^{c,\chi}_{cQ} \\ p^{c,\gamma}_{cQ} \\ p^{c,\kappa}_{cQ} \end{bmatrix} \Rightarrow \begin{bmatrix} u_Q \\ v_Q \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\frac{p^{c,\chi}_{cQ}}{p^{c,\kappa}_{cQ}} + u_0 \\ f\Delta_v^{-1}\frac{p^{c,\gamma}_{cQ}}{p^{c,\kappa}_{cQ}} + v_0 \end{bmatrix} \tag{9.1}$$

Further, assume that the obstacle is represented as a simple planar square object with size L positioned at p^c_{cQ}. Each vertex $p^c_{cL_i}$ of this square may be projected into the image scene:

$$p^c_{cL_1} = p^c_{cQ} + \frac{L}{2}\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p^{c,\chi}_{cQ} + 0.5L \\ p^{c,\gamma}_{cQ} + 0.5L \\ p^{c,\kappa}_{cQ} \end{bmatrix} \Rightarrow \begin{bmatrix} u_{L_1} \\ v_{L_1} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\frac{p^{c,\chi}_{cQ}+0.5L}{p^{c,\kappa}_{cQ}} + u_0 \\ f\Delta_v^{-1}\frac{p^{c,\gamma}_{cQ}+0.5L}{p^{c,\kappa}_{cQ}} + v_0 \end{bmatrix} \tag{9.2}$$

$$p^c_{cL_2} = p^c_{cQ} + \frac{L}{2}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} p^{c,\chi}_{cQ} + 0.5L \\ p^{c,\gamma}_{cQ} - 0.5L \\ p^{c,\kappa}_{cQ} \end{bmatrix} \Rightarrow \begin{bmatrix} u_{L_2} \\ v_{L_2} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\frac{p^{c,\chi}_{cQ}+0.5L}{p^{c,\kappa}_{cQ}} + u_0 \\ f\Delta_v^{-1}\frac{p^{c,\gamma}_{cQ}-0.5L}{p^{c,\kappa}_{cQ}} + v_0 \end{bmatrix} \tag{9.3}$$

$$p^c_{cL_3} = p^c_{cQ} + \frac{L}{2}\begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} p^{c,\chi}_{cQ} - 0.5L \\ p^{c,\gamma}_{cQ} - 0.5L \\ p^{c,\kappa}_{cQ} \end{bmatrix} \Rightarrow \begin{bmatrix} u_{L_3} \\ v_{L_3} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\frac{p^{c,\chi}_{cQ}-0.5L}{p^{c,\kappa}_{cQ}} + u_0 \\ f\Delta_v^{-1}\frac{p^{c,\gamma}_{cQ}-0.5L}{p^{c,\kappa}_{cQ}} + v_0 \end{bmatrix} \tag{9.4}$$

$$p^c_{cL_4} = p^c_{cQ} + \frac{L}{2}\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p^{c,\chi}_{cQ} - 0.5L \\ p^{c,\gamma}_{cQ} + 0.5L \\ p^{c,\kappa}_{cQ} \end{bmatrix} \Rightarrow \begin{bmatrix} u_{L_4} \\ v_{L_4} \end{bmatrix} = \begin{bmatrix} f\Delta_u^{-1}\frac{p^{c,\chi}_{cQ}-0.5L}{p^{c,\kappa}_{cQ}} + u_0 \\ f\Delta_v^{-1}\frac{p^{c,\gamma}_{cQ}+0.5L}{p^{c,\kappa}_{cQ}} + v_0 \end{bmatrix} \tag{9.5}$$

The projection area a_p in pixel^2 may be computed as a function of the projection of

each vertex of the square:

$$\begin{bmatrix} u_{L_M} \\ v_{L_M} \end{bmatrix} = \sum_{i=1}^{4} \frac{1}{4} \begin{bmatrix} u_{L_i} \\ v_{L_i} \end{bmatrix} \Rightarrow \begin{bmatrix} \bar{u}_{L_i} \\ \bar{v}_{L_i} \end{bmatrix} = \begin{bmatrix} u_{L_i} - u_{L_M} \\ v_{L_i} - v_{L_M} \end{bmatrix}$$

$$a_p = \frac{1}{2}\Big(\big|\big|\bar{u}_{L_1}\bar{v}_{L_2} + \bar{u}_{L_2}\bar{v}_{L_3} + \bar{u}_{L_3}\bar{v}_{L_4} + \bar{u}_{L_4}\bar{v}_{L_1} - \bar{u}_{L_2}\bar{v}_{L_1} - \bar{u}_{L_3}\bar{v}_{L_2} - \bar{u}_{L_4}\bar{v}_{L_3} - \bar{u}_{L_1}\bar{v}_{L_4}\big|\big|\Big) \tag{9.6}$$

Note that visible objects should have a projection area a_p bigger than 1 px^2. Still, if a_p is not sufficiently greater than 1 px^2, its projection in the image scene may be very difficult to acknowledge. An unitary pixel typically represents a small fraction of the camera image. Assuming an image with size 1920 $px \times 880$ px, the total area would be 1920 $px \times 880$ $px = 1689600$ px^2. Hence, 1 px^2 would represent less than 0.0001% of the total area from the image:

$$\frac{1\ px^2}{1920\ px \times 880\ px} = 0.0000591856\% \tag{9.7}$$

For fixed intrinsic parameters, it is possible to determine the maximum distance in which the square is observable by the computation of a_p around different points of operation. Table 6 presents a_p as a function of different L at points $p^c_{cQ_1} = [50, 15, 200]$, $p^c_{cQ_2} = [50, 15, 2000]$ and $p^c_{cQ_3} = [50, 15, 10000]$ for fixed intrinsic camera parameters $[f_u, f_v, u_0, v_0] = [790, 770, 960, 440]$.

Table 6: Projection area a_p calculated at $p^c_{cQ_i}$ for different values of L

L $[m]$	a_p at $p^c_{cQ_1}$ [px^2]	a_p at $p^c_{cQ_2}$ [px^2]	a_p at $p^c_{cQ_3}$ [px^2]
1	15.306	0.153	0.006
5	382.656	3.827	0.153
10	1530.625	15.306	0.612
20	6122.500	61.225	2.449
50	38265.625	382.656	15.306

For example, a square with an area of 1 m^2 would not be visible at a distance of 2000 m but would be detectable at a distance of 200 m from the camera. Similarly, a bigger square with an area of 100 m^2 would be visible at a distance of 2000 m but would not be detectable at a distance of 10000 m from the camera. Generally, the maximum distance in which each object is detectable depends on its size. Hence, in the context of this preliminary analysis, smaller objects would be detectable at distances as far as 200 m, while bigger objects would be detectable at distances as far as 2000 m.

Note that the same methodology could be repeated with different intrinsic camera parameters for the comparison of different camera devices. Besides, further research is suggested in order to extend the above methodology with more realistic object models. A simple approach would be the definition of a planar rectangle with size equivalent to the geometry of potential obstacles. Alternatively, the definition of three-dimensional models for potential obstacles could be more adequate for a thorough analysis.

Recall that the camera position and orientation with respect to the ship coordinate system should be known in order to project any virtual point into the camera scene. Thus, the camera should be installed around an expected position and orientation with precision to avoid errors in each virtual point projection. The rest of this section discusses how errors in the determination of the camera state with respect to the ship affects the virtual points projection in the camera scene. A detailed systematic error analysis is beyond the scope of the present work.

Consider an arbitrary point with position p^s_{sp} described with respect to the ship coordinate system. Further, let p^s_{sc} be the position of the camera and R^c_s be the rotation matrix describing the camera frame with respect to the ship coordinate system:

$$p^s_{sp} = \begin{bmatrix} p^{s,i}_{sp} \\ p^{s,j}_{sp} \\ p^{s,k}_{sp} \end{bmatrix} \ , \ R^c_s = \begin{bmatrix} r^c_{s11} & r^c_{s12} & r^c_{s13} \\ r^c_{s21} & r^c_{s22} & r^c_{s23} \\ r^c_{s31} & r^c_{s32} & r^c_{s33} \end{bmatrix} \ , \ p^s_{sc} = \begin{bmatrix} p^{s,i}_{sc} \\ p^{s,j}_{sc} \\ p^{s,k}_{sc} \end{bmatrix} \tag{9.8}$$

Transforming p^s_{sp} to the camera coordinate system:

$$p^c_{cp} = \begin{bmatrix} p^{c,\chi}_{cp} \\ p^{c,\gamma}_{cp} \\ p^{c,\kappa}_{cp} \end{bmatrix} = \begin{bmatrix} R^c_s & -R^c_s p^s_{sc} \end{bmatrix} \begin{bmatrix} p^s_{sp} \\ 1 \end{bmatrix} \Rightarrow \tag{9.9}$$

$$\begin{bmatrix} p^{c,\chi}_{cp} \\ p^{c,\gamma}_{cp} \\ p^{c,\kappa}_{cp} \end{bmatrix} = \begin{bmatrix} r^c_{s11}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s12}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s13}(p^{s,k}_{sp} - p^{s,k}_{sc}) \\ r^c_{s21}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s22}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s23}(p^{s,k}_{sp} - p^{s,k}_{sc}) \\ r^c_{s31}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s32}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s33}(p^{s,k}_{sp} - p^{s,k}_{sc}) \end{bmatrix} \tag{9.10}$$

Consider a set of estimated parameters $\hat{p}^s_{sc}$ and $\hat{R}^c_s$ for the camera position and orientation with respect to the ship coordinate system:

$$\hat{R}^c_s = \begin{bmatrix} \hat{r}^c_{s11} & \hat{r}^c_{s12} & \hat{r}^c_{s13} \\ \hat{r}^c_{s21} & \hat{r}^c_{s22} & \hat{r}^c_{s23} \\ \hat{r}^c_{s31} & \hat{r}^c_{s32} & \hat{r}^c_{s33} \end{bmatrix} \ , \ \hat{p}^s_{sc} = \begin{bmatrix} \hat{p}^{s,i}_{sc} \\ \hat{p}^{s,j}_{sc} \\ \hat{p}^{s,k}_{sc} \end{bmatrix} \tag{9.11}$$

Transforming p^s_{sp} to the camera coordinate system with estimated parameters $\hat{p}^s_{sc}$ and $\hat{R}^c_s$ yields $\hat{p}^c_{cp}$:

$$\hat{p}^c_{cp} = \begin{bmatrix} \hat{p}^{c,\chi}_{cp} \\ \hat{p}^{c,\gamma}_{cp} \\ \hat{p}^{c,\kappa}_{cp} \end{bmatrix} = \begin{bmatrix} \hat{R}^c_s & -\hat{R}^c_s \hat{p}^s_{sc} \end{bmatrix} \begin{bmatrix} p^s_{sp} \\ 1 \end{bmatrix} \Rightarrow \tag{9.12}$$

$$\begin{bmatrix} \hat{p}^{c,\chi}_{cp} \\ \hat{p}^{c,\gamma}_{cp} \\ \hat{p}^{c,\kappa}_{cp} \end{bmatrix} = \begin{bmatrix} \hat{r}^c_{s11}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s12}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s13}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc}) \\ \hat{r}^c_{s21}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s22}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s23}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc}) \\ \hat{r}^c_{s31}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s32}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s33}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc}) \end{bmatrix} \tag{9.13}$$

Assuming the *pinhole* model described by camera parameters $[f_u, f_v, u_0, v_0]$, the projection of p^s_{sp} may be computed as:

$$\begin{bmatrix} u \\ \\ v \end{bmatrix} = \begin{bmatrix} f_u \frac{r^c_{s11}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s12}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s13}(p^{s,k}_{sp} - p^{s,k}_{sc})}{r^c_{s31}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s32}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s33}(p^{s,k}_{sp} - p^{s,k}_{sc})} + u_0 \\ \\ f_v \frac{r^c_{s21}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s22}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s23}(p^{s,k}_{sp} - p^{s,k}_{sc})}{r^c_{s31}(p^{s,i}_{sp} - p^{s,i}_{sc}) + r^c_{s32}(p^{s,j}_{sp} - p^{s,j}_{sc}) + r^c_{s33}(p^{s,k}_{sp} - p^{s,k}_{sc})} + v_0 \end{bmatrix} \tag{9.14}$$

Assuming estimated camera parameters $[\hat{f}_u, \hat{f}_v, \hat{u}_0, \hat{v}_0]$, the estimated projection $[\hat{u}, \hat{v}]$ may be computed as:

$$\begin{bmatrix} \hat{u} \\ \\ \hat{v} \end{bmatrix} = \begin{bmatrix} \hat{f}_u \frac{\hat{r}^c_{s11}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s12}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s13}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc})}{\hat{r}^c_{s31}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s32}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s33}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc})} + \hat{u}_0 \\ \\ \hat{f}_v \frac{\hat{r}^c_{s21}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s22}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s23}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc})}{\hat{r}^c_{s31}(p^{s,i}_{sp} - \hat{p}^{s,i}_{sc}) + \hat{r}^c_{s32}(p^{s,j}_{sp} - \hat{p}^{s,j}_{sc}) + \hat{r}^c_{s33}(p^{s,k}_{sp} - \hat{p}^{s,k}_{sc})} + \hat{v}_0 \end{bmatrix} \tag{9.15}$$

Similarly, let δu and δv be the error between $[u, v]$ and $[\hat{u}, \hat{v}]$, and consider the following projection error e_p:

$$\delta u = u - \hat{u} \ , \ \delta v = v - \hat{v} \tag{9.16}$$

$$e_p = \sqrt{\delta u^2 + \delta v^2} \tag{9.17}$$

Furthermore, let $[\delta f_u, \delta f_v, \delta u_0, \delta v_0]$ be the errors in the intrinsic camera parameters, $[\delta p^{s,i}_{sc}, \delta p^{s,j}_{sc}, \delta p^{s,k}_{sc}]$ be the errors in the camera position and $[\delta\phi, \delta\theta, \delta\psi]$ be the errors in the camera orientation:

$$\hat{R}^c_s = R^{ZYX}(\delta\phi, \delta\theta, \delta\psi) R^c_s \ , \ \hat{p}^s_{sc} = \begin{bmatrix} 42.0 + \delta p^{s,i}_{sc} \\ -25.5 + \delta p^{s,j}_{sc} \\ 57.51 + \delta p^{s,k}_{sc} \end{bmatrix} \ , \ \begin{bmatrix} \hat{f}_u \\ \hat{f}_v \\ \hat{u}_0 \\ \hat{v}_0 \end{bmatrix} = \begin{bmatrix} 790 + \delta f_u \\ 770 + \delta f_v \\ 960 + \delta u_0 \\ 940 + \delta v_0 \end{bmatrix} \tag{9.18}$$

where $R^{ZYX}(\delta\phi, \delta\theta, \delta\psi)$ is given by Equation 3.20. Consider each error component as a Gaussian random variable with zero mean and non-zero covariance w. Let $[w_{fu}, w_{fv}, w_{u0}, w_{v0}]$ be the corresponding covariances of errors $[\delta f_u, \delta f_v, \delta u_0, \delta v_0]$, let $[w_{pi}, w_{pj}, w_{pk}]$ be the covariances of errors $[\delta p_{sc}^{s,i}, \delta p_{sc}^{s,j}, \delta p_{sc}^{s,k}]$, and let $[w_{r\phi}, w_{r\theta}, w_{r\psi}]$ be the covariances of errors $[\delta\phi, \delta\theta, \delta\psi]$. Recall that the probability density of a Gaussian random variable η_X have a peak at its mean μ and is almost null for points that are far away from μ. The width in which the probability density is significant is given by its covariance σ. It is possible to numerically determine the probability that an observation of η_X is contained in intervals given by μ and σ:

$$P\Big(\mu - 3\sigma < x < \mu + 3\sigma\Big) \approx 99.73\% \tag{9.19}$$

Therefore, the following covariances yields observations for $[\delta f_u, \delta f_v, \delta u_0, \delta v_0]$ that are probably smaller than an arbitrary constant Π_{px} (with probability higher than 99.73%):

$$w_{fu} = w_{fv} = w_{u0} = w_{v0} = \frac{\Pi_{px}}{3} \tag{9.20}$$

Similarly, the following covariances yields observations for the extrinsic camera parameters that are probably smaller than arbitrary constants Π_m and Π_{deg}:

$$w_{pi}, w_{pj}, w_{pk} = \frac{\Pi_m}{3} \;\; ; \;\; w_{r\phi}, w_{r\theta}, w_{r\psi} = \frac{\Pi_{deg}}{3}\frac{\pi}{180} \tag{9.21}$$

Consider N observations for each error component yielding N different projection errors e_p. Useful statistics such as *minimum*, *maximum*, *mean*, and *standard deviation* from the set of projection errors may assist in the determination of preliminary requirements for the system. Table 7 presents projection error e_p in *pixels* computed for points $p_{sc}^s = [2000, -500, 13]^T$ with $N = 1000$ and different $[\Pi_{px}, \Pi_m, \Pi_{deg}]$.

Table 7: Projection error e_p calculated at p for different values of Π_{px}, Π_m and Π_{deg}

Π_{px}	Π_m	Π_{deg}	$min(e_p)$ [px]	$max(e_p)$ [px]	$mean(e_p)$ [px]	$std(e_p)$ [px]
0.1	1.0	0.1°	0.0125	1.9427	0.6116	0.3285
0.1	5.0	0.1°	0.04620	3.1401	1.0327	0.5126
0.1	10.0	0.1°	0.0258	5.4089	1.7966	0.9133
1.0	1.0	0.1°	0.0202	2.2557	0.7577	0.4097
10.0	1.0	0.1°	0.1784	11.9372	4.3873	2.2654
25.0	1.0	0.1°	0.5824	29.2537	10.7294	5.6226
1.0	1.0	1.0°	0.1787	18.3374	5.9398	3.1045
1.0	1.0	2.5°	0.2719	45.1124	15.2081	7.8314
1.0	1.0	5.0°	1.7297	88.4720	30.3552	15.5040

The initial three lines from the table compares results with variation in the error for the camera position. The next three lines compares the projection error with different uncertainty for the camera intrinsic parameters. Finally, the last three lines presents the projection error results with different uncertainty for the camera orientation. Above results indicates that the projection error is not considerably affected by the introduction of bigger errors in the camera position, but it is significantly affected by errors in the camera orientation.

Given a desirable minimum projection error e_p^{MIN}, it is possible to determine simplified installation requirements for the camera installation accuracy. For example, if the projection errors for the system are constrained to be at a maximum of 5 px, Table 7 would indicate that the camera orientation could be determined within 0.1° accuracy, the camera position could be determined within 1 m accuracy, and camera intrinsic parameters could be determined within 1 px accuracy. Similarly, if the projection errors for the system are constrained to be at a maximum of 20 px, above results would indicate that the camera orientation should be determined within 1.0° accuracy, the camera position should be determined within 1 m accuracy, and camera intrinsic parameters should be determined within 1 px accuracy. For a image of size 1920 $px \times 880$ px, an error of 5 px represents less than 1% of the image total width and height; while an error of 20 px represents less than 3% of the image total width and height:

$$\frac{5px}{1920px} \approx 0.260\% \ , \ \frac{5px}{880px} \approx 0.568\% \tag{9.22}$$

$$\frac{20px}{1920px} \approx 1.042\% \ , \ \frac{20px}{880px} \approx 2.273\% \tag{9.23}$$

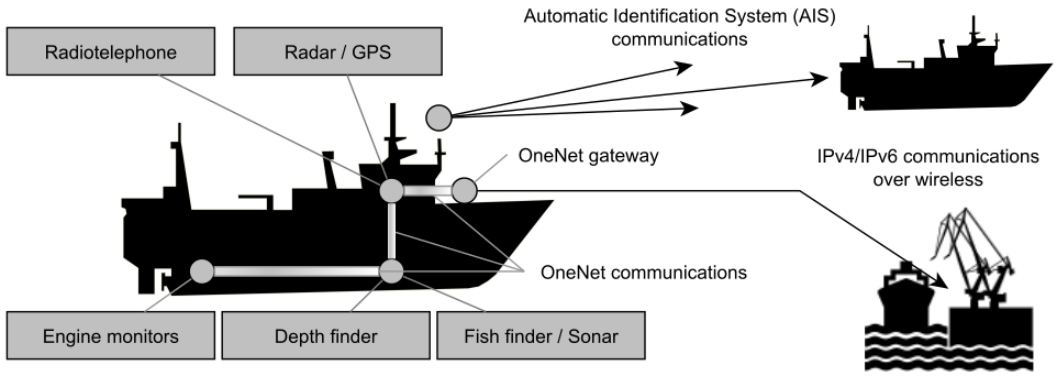
Note that above tables are computed around a single point of operation. The proposed methodology may be extended with similar computations around other points of operation. Alternative investigations regarding requirements definition are also suggested as these considerations are extremely important for enabling the development of a real MAR prototype. Moreover, a similar analysis needs to be performed for the aforementioned *tracker* component of the system.

9.3 External Integration

There are multiple onboard equipment that assists operators with the ship conduct. These multiple onboard devices may be produced by one or many manufacturers. All equipment must be interconnected with a common protocol for a complete integration.

Figure 95 exemplifies equipment and protocols that may exist on a single marine vessel [TRAN et al., 2021].

Figure 95: Network protocols utilized by a marine vessel and their interaction with other adjacent vessel or coastal services.



Source: [TRAN et al., 2021]

Several considerations must be addressed in order to implement a network for interconnecting marine equipment. As with almost any network protocol, marine network protocols are potentially vulnerable against malicious attacks or unintentional human errors. Table 8 presents major security considerations that are typically used for evaluating the security of a network protocol [TRAN et al., 2021].

Table 8: Summary of security considerations

Security Consideration	**Primary Risk**
Confidentiality	Access to private data
Integrity	Modification of data
Availability	Inability to access data or resource
Authentication	Inability to confirm identity
Authorization	Improper access to resource or data
Non-repudiation	Inability to confirm an action made by an identity

Source: [TRAN et al., 2021]

Examples of protocols that enables the interchange of information between different equipment are standards NMEA-0183 [NATIONAL MARINE ELECTRONICS ASSOCIATION, 2002] and NMEA-2000 [NATIONAL MARINE ELECTRONICS ASSOCIATION, 2016] published by the National Marine Electronics Association (**NMEA**). The integration of the proposed MAR system with current navigational equipment may be

facilitated with the investigation of such maritime protocols. Note that as the overall cost of the prototype might be significant, it is interesting to further investigate the viability of the proposed prototype before its full development. In this context, it is possible to research for *render* techniques with recorded video from navigation experiments. Furthermore, it is possible to investigate potential *interfaces* for the system with any full-mission *ship maneuvering simulator*. As a general recommendation, the development of a *monitor augmented reality* system for navigational assistance should be initially performed with a *ship maneuvering simulator*. In such simulations, all geometrical parameters are known, which facilitates the development of ideal interfaces and visualizations without the need of extremely accurate equipment.

10 CONCLUSION

Foundations regarding augmented reality methods were discussed throughout the dissertation. A simple conceptual design for a monitor AR solution were proposed. Important components of the proposed design were illustrated with numerical implementations. Examples of enhanced scenes for a monitor AR solution were proposed in a context of a ship navigating with constant velocity in restricted waters. As each augmented scene assist in the perception of the navigation environment, these visualizations are potentially helpful for the corresponding ship operators. However, in order to adequately embed these augmented scenes into helpful equipment, designed visualizations shall be further validated considering usability requirements from typical operations.

The development of an operational augmented reality system for navigational assistance is still a complex task. For a real implementation, the ship state with respect to the world coordinate system needs to be accurately determined along with the camera parameters. In order to ensure a proper alignment between real and virtual marks, information regarding the truth position of nearby obstacles and ship surroundings needs to be accurately determined by onboard ship systems. Besides that, the integration of such different equipment needs to robustly address real time constraints which may be inherently complex.

Despite the challenges, the development of equipment with augmented reality methods is further suggested as it has the potential to significantly contribute to the overall efficiency and safety of maritime operations. It should be noted that although further research with real implementations may be extremely expensive due to the requirement of high-precision sensors, usability research may be appropriately performed with any full-mission ship maneuvering simulator as presented in the present work. Therefore, as a general recommendation, the authors suggests that user interface research for MAR systems should be preliminary performed with a ship maneuvering simulator. In such simulations all geometrical parameters are known, which facilitates the development of ideal visualizations.

REFERENCES

AZUMA, R. T. A survey of augmented reality. **Presence: Teleoperators & Virtual Environments**, Cambridge, United States, v. 6, n. 4, p. 355–385, 1997.

AZUMA, R. T. et al. Recent advances in augmented reality. **IEEE Computer Graphics and Applications**, Washington, United States, v. 21, n. 6, p. 34–47, Nov 2001. ISSN 1558-1756.

BILLINGHURST, M.; CLARK, A.; LEE, G. A survey of augmented reality. **Foundations and Trends® in Human–Computer Interaction**, Hanover, United States, v. 8, n. 2-3, p. 73–272, 2015. ISSN 1551-3955. Disponível em: ⟨http://dx.doi.org/10.1561/1100000049⟩.

BRADSKI, G.; KAEHLER, A. **Learning OpenCV: Computer vision with the OpenCV library**. Sebastopol, United States: O'Reilly Media, Inc., 2008. Disponível em: ⟨https://opencv.org/⟩.

BRANCH, M. A.; COLEMAN, T. F.; LI, Y. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. **SIAM Journal on Scientific Computing**, Philadelphia, United States, v. 21, n. 1, p. 1–23, 1999.

BROWN, D. C. Decentering distortion of lenses. **Photogrammetric Engineering**, [S.l.], v. 32, n. 3, p. 444–462, 1966.

BYRD, R. H.; SCHNABEL, R. B.; SHULTZ, G. A. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. **Mathematical programming**, Berlin, Germany, v. 40, n. 1, p. 247–263, 1988.

CLARKE, T. A.; FRYER, J. G. The development of camera calibration methods and models. **The Photogrammetric Record**, Nottingham, United Kingdom; Hoboken, United States, v. 16, n. 91, p. 51–66, 1998.

COMMISSION DECISION C(2014)4995. **Technology readiness levels (TRL).** Europe, 2014. Disponível em: ⟨https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf⟩.

CONRADY, A. E. Decentred lens-systems. **Monthly notices of the royal astronomical society**, London, United Kingdom, v. 79, n. 5, p. 384–390, 1919.

FENG ZHOU; DUH, H. B.; BILLINGHURST, M. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In: **2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality**. Cambridge, United Kingdom: Institute of Electrical and Electronics Engineers (IEEE), 2008. p. 193–202.

GROVES, P. D. **Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems**. Boston, USA: Artech House, 2008.

HARTLEY, R.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. 2. ed. Cambridge, United Kingdom: Cambridge University Press, 2003. ISBN 0521540518.

HEILIG, L.; LALLA-RUIZ, E.; VOSS, S. Digital transformation in maritime ports: analysis and a game theoretic framework. **Netnomics: Economic Research and Electronic Networking**, Springer, [S.l], v. 18, n. 2-3, p. 227–254, dez. 2017. ISSN 1385-9587.

HEILIG, L.; SCHWARZE, S.; VOSS, S. An analysis of digital transformation in the history and future of modern ports. In: **Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)**. Hawaii, United States: ScholarSpace / AIS Electronic Library (AISeL), 2017. p. 1–10. Disponível em: ⟨http://dblp.uni-trier.de/db/conf/hicss/hicss2017.html#HeiligSV17⟩.

HONG, T. C.; ANDREW, H. S. Y.; KENNY, C. W. L. Assessing the situation awareness of operators using maritime augmented reality system (MARS). **Proceedings of the Human Factors and Ergonomics Society Annual Meeting**, [S.l], v. 59, n. 1, p. 1722–1726, 2015. Disponível em: ⟨https://doi.org/10.1177/1541931215591372⟩.

INTERNATIONAL MARITIME ORGANIZATION. **International Convention for the Prevention of Pollution from Ships**. [S.l.], 1973. Disponível em: ⟨https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Prevention-of-Pollution-from-Ships-(MARPOL).aspx⟩. Acesso em: 7 mar. 2021.

INTERNATIONAL MARITIME ORGANIZATION. **International Convention for the Safety of Life At Sea**. [S.l.], 1974. Disponível em: ⟨https://www.imo.org/en/About/Conventions/Pages/International-Convention-for-the-Safety-of-Life-at-Sea-(SOLAS),-1974.aspx⟩. Acesso em: 7 mar. 2021.

INTERNATIONAL MARITIME ORGANIZATION. **Maritime Safety Committee (MSC)**: 98th session. London, United Kingdom, 2017. Disponível em: ⟨https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-98th-session.aspx⟩. Acesso em: 5 jan. 2021.

INTERNATIONAL MARITIME ORGANIZATION. **Maritime Safety Committee (MSC)**: 99th session. London, United Kingdom, 2018. Disponível em: ⟨https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-99th-session.aspx⟩. Acesso em: 5 jan. 2021.

INTERNATIONAL MARITIME ORGANIZATION. **Maritime Safety Committee (MSC)**: 100th session. London, United Kingdom, 2018. Disponível em: ⟨https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-100th-session.aspx⟩. Acesso em: 5 jan. 2021.

INTERNATIONAL MARITIME ORGANIZATION. **Maritime Safety Committee (MSC)**: 101th session. London, United Kingdom, 2019. Disponível em: ⟨https://www.imo.org/en/MediaCentre/MeetingSummaries/Pages/MSC-101st-session.aspx⟩. Acesso em: 5 jan. 2021.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 8468:2007**: Ships and marine technology — ship's bridge layout and associated equipment — requirements and guidelines. Geneva, Switzerland, 2007. Disponível em: ⟨https://www.iso.org/standard/43040.html⟩. Acesso em: 5 jan. 2021.

KAEHLER, A.; BRADSKI, G. **Learning OpenCV 3: computer vision in C++ with the OpenCV library**. Sebastopol, United States: O'Reilly Media, Inc., 2016. Disponível em: ⟨https://opencv.org/⟩.

KOPACZ, Z.; MORGAŚ, W.; URBAŃSKI, J. The maritime safety system, its main components and elements. **Journal of Navigation**, Cambridge, United Kingdom, v. 54, n. 2, p. 199–211, 2001.

KOPACZ, Z.; MORGAŚ, W.; URBAŃSKI, J. The changes in maritime navigation and the competences of navigators. **Journal of Navigation**, Cambridge, United Kingdom, v. 57, n. 1, p. 73–83, 2004.

LAERA, F. et al. Augmented reality for maritime navigation data visualisation: a systematic review, issues and perspectives. **Journal of Navigation**, Cambridge, United Kingdom, p. 1–18, 2021.

LEITE, B. G. et al. Maritime navigational assistance by visual augmentation. **Journal of Navigation**, Cambridge University Press, v. 75, n. 1, p. 57–75, 2022.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. **Quarterly of applied mathematics**, Providence, United States, v. 2, n. 2, p. 164–168, 1944.

MAKIYAMA, H. S. et al. Computational graphics and immersive technologies applied to a ship maneuvering simulator. In: **19th International Conference on Geometry and Graphics - ICGG 2020, 2021**. [S.l.: s.n.], 2020. p. 626–635.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the Society for Industrial and Applied Mathematics**, Philadelphia, United States, v. 11, n. 2, p. 431–441, 1963.

MAYBECK, P. S. **Stochastic Models, Estimation, and Control: Chapter 5 Optimal filtering with linear system models**. Cambridge, United States: Elsevier - Academic Press, 1979. v. 1. 203 - 288 p. (Mathematics in Science and Engineering, v. 1). ISSN 0076-5392. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S0076539208621700⟩.

MAYBECK, P. S. **Stochastic Models, Estimation, and Control: Chapter 9 Compensation of linear model inadequacies**. Cambridge, United States: Elsevier - Academic Press, 1982. v. 3. 23 - 67 p. (Mathematics in Science and Engineering, v. 3). ISSN 0076-5392. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S0076539208632221⟩.

MIHOC, A.; CATER, K. Augmenting navigational aids: The development of an assistive maritime navigation application. **International Journal of Computer and Information Engineering**, World Academy of Science, Engineering and Technology, [S.l], v. 11, n. 1, p. 146 – 151, 2017. ISSN eISSN: 1307-6892. Disponível em: ⟨https://publications.waset.org/vol/121⟩.

MILGRAM, P. et al. Augmented reality: a class of displays on the reality-virtuality continuum. In: **Telemanipulator and Telepresence Technologies**. [S.l]: Society of Photo-Optical Instrumentation Engineers (SPIE), 1995. v. 2351, p. 282 – 292. Disponível em: ⟨https://doi.org/10.1117/12.197321⟩.

MORÉ, J. J. The levenberg-marquardt algorithm: Implementation and theory. In: WATSON, G. A. (Ed.). **Numerical Analysis**. Berlin, Heidelberg; Germany: Springer Berlin Heidelberg, 1978. p. 105–116. ISBN 978-3-540-35972-2.

MORGÈRE, J.; DIGUET, J.; LAURENT, J. Electronic navigational chart generator for a marine mobile augmented reality system. In: **2014 Oceans - St. John's**. St. John's, Canada: Institute of Electrical and Electronics Engineers (IEEE), 2014. p. 1–9. ISSN 0197-7385.

NATIONAL MARINE ELECTRONICS ASSOCIATION. **NMEA 0183–Standard for Interfacing Marine Electronic Devices (NMEA-0183)**. Reston, VA, USA, 2002.

NATIONAL MARINE ELECTRONICS ASSOCIATION. **NMEA 2000 ® Interface Standard Standard for Serial-Data Networking of Marine Electronic Devices**. Reston, VA, USA, 2016.

OH, J.; PARK, S.; KWON, O.-S. Advanced navigation aids system based on augmented reality. **International Journal of e-Navigation and Maritime Economy**, [S.l], v. 5, p. 21–31, 12 2016.

SANCHEZ-GONZALEZ, P.-L. et al. Toward digitalization of maritime transport? **Sensors**, Basel, Switzerland, v. 19, n. 4, 2019. ISSN 1424-8220.

SAROLIC, A. A review of maritime navigation and radiocommunication equipment and systems standardization. In: **Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine**. Zadar, Croatia: Institute of Electrical and Electronics Engineers (IEEE), 2004. p. 380–383. ISSN 1334-2630.

SUTHERLAND, I. E. A head-mounted three dimensional display. In: **Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I**. New York, NY, USA: Association for Computing Machinery, 1968. (AFIPS '68 (Fall, part I)), p. 757–764. ISBN 9781450378994. Disponível em: ⟨https://doi.org/10.1145/1476589.1476686⟩.

SZELISKI, R. **Computer vision: algorithms and applications**. [S.l]: Springer Science & Business Media, 2010. Disponível em: ⟨https://szeliski.org/Book/⟩.

TANNURI, E. A. et al. Modular mathematical model for a low-speed maneuvering simulator. In: **ASME 33th International Conference on Ocean, Offshore and Arctic Engineering OMAE 2014**. [S.l.: s.n.], 2014.

TITTERTON, D. H.; WESTON, J. L. **Strapdown Inertial Navigation Technology**. 2. ed. Stevenage, United Kingdom and Reston, USA: The Institution of Electrical Engineers and The American Institute of Aeronautics and Astronautics, 2004. ISBN 0521540518.

TRAN, K. et al. Marine network protocols and security risks. **Journal of Cybersecurity and Privacy**, v. 1, n. 2, p. 239–251, 2021. ISSN 2624-800X. Disponível em: ⟨https://www.mdpi.com/2624-800X/1/2/13⟩.

VASILIJEVIC, A.; BOROVIĆ, B.; VUKIĆ, Z. Augmented reality in marine applications. **Brodogradnja**, Zagreb, Croatia, v. 62, p. 136–142, 06 2011.

VENKATRAMAN, N. IT-enabled business transformation: from automation to business scope redefinition. **MIT Sloan Management Review**, Cambridge, United States, v. 35 (2), p. 73–87, 1994.

VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. **Nature Methods**, New York, United States, v. 17, p. 261–272, 2020. Disponível em: ⟨https://www.scipy.org/⟩.

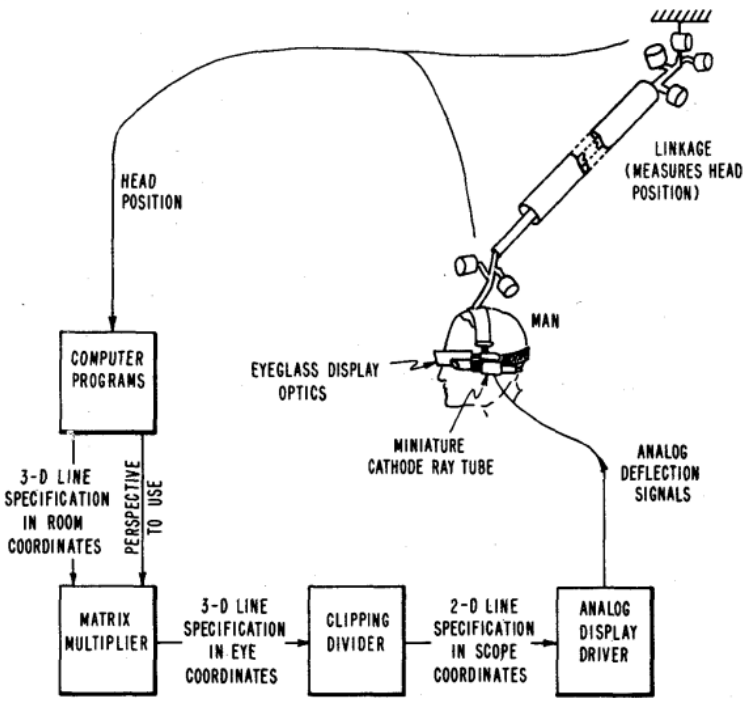
WANG, J. et al. A new calibration model of camera lens distortion. **Pattern Recognition**, Elsevier, v. 41, n. 2, p. 607 – 615, 2008. ISSN 0031-3203. Disponível em: ⟨http://www.sciencedirect.com/science/article/pii/S0031320307003020⟩.

ZHANG, Z. A flexible new technique for camera calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Washington, United States, v. 22, n. 11, p. 1330–1334, Nov 2000. ISSN 1939-3539.

APPENDIX A – AR HISTORY

The beginnings of *Augmented-Reality* technology can be referred to a pioneer work in the 1960s. A see through head-mounted device (**HMD**) was designed to present three-dimensional graphics [SUTHERLAND, 1968]. The fundamental idea behind the three-dimensional display proposed in the work is to present the user with a perspective image which changes as he moves. As the retinal images measured by eyes are two-dimensional projections, it is possible to create three-dimensional object illusions by placing suitable two-dimensional images on the observer's retina. Figure 96 illustrates the proposed device with a diagram of its constitutive parts.

Figure 96: The parts of the three-dimensional display system



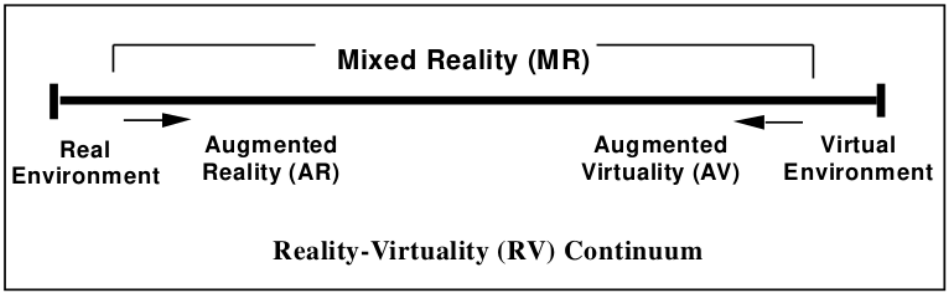
Source: [SUTHERLAND, 1968]

A tracking device was designed for measuring the head movement of a user in real-time. Two different coordinate systems were defined: the *room coordinate system* and

the *eye coordinate system*. Fixed points were stored numerically in the *room* coordinate system. Considering the viewer position known at all times with the tracking device, the *eye coordinate system* can be determined. A digital matrix multiplier was built to dynamically compute the transformation between the *room* and the *eye coordinate system*. The corresponding image projection of the virtual three-dimensional points were computed with perspective projection models.

Over the next years, further research on the proposed concept resulted in the emergence of *augmented reality* as a research field. In [MILGRAM et al., 1995], *augmented-reality* displays were categorized in a general sense within the context of a *Reality-Virtuality Continuum*. This *continuum* have been proposed as a unified framework for studying environments in which real and virtual world objects are presented in the same display. These environments have been defined as *mixed-reality* environments. Figure 97 presents a diagram illustrating the proposed concept.

Figure 97: Simplified representation of a RV Continuum



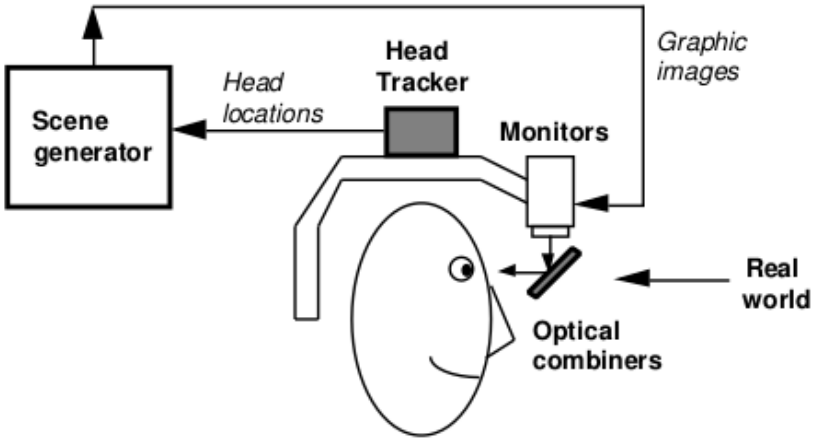
Source: [MILGRAM et al., 1995]

In this framework, a graphical simulation is classified as a *purely virtual environment* while a recorded video without further graphical processing is considered as a *purely real environment*. As *augmented-reality* methods are concerned with showing scenes of real environments with synthetic elements, most of the presented view elements are going to be real. Thus, it is located closer to the *purely real* side of the continuum. On the other side, *augmented-virtuality* methods are located closer to the *purely virtual environment* as such displays presents simulated environments with few enhancements based on real data. Two subclasses for *augmented-reality* displays have been defined: *see-through* and *monitor-based* displays. *See-through* displays are inspired on the earlier work of [SUTHERLAND, 1968], while *monitor-based* refers to displays in which computer generated images are overlaid onto live or stored video images.

In [AZUMA, 1997], an important survey have been published summarizing main de-

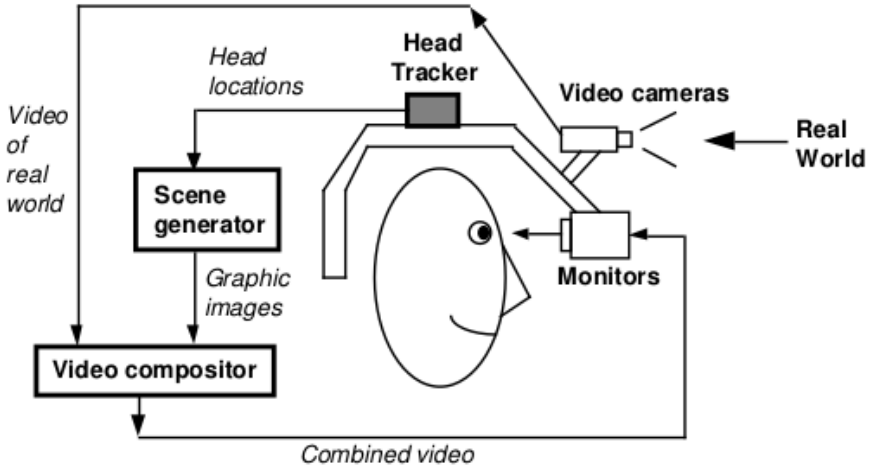
velopments up to that point. The work proposed definitions for the field, including main characteristics, challenges and examples of applications. Obstacles for the development of *augmented-reality* systems were identified at the time. The survey defines *augmented-reality* as systems with three main characteristics: *combines real and virtual information*, *interactive in real time* and *registered in 3-D*. As in [SUTHERLAND, 1968], the work proposes concepts for see-through head mounted devices. Two particular types of see-through HMD were proposed. One type is built with optical combiners as illustrated by Figure 98. The other type is built with video cameras and monitors as illustrated by Figure 99.

Figure 98: Optical see-through HMD conceptual diagram



Source: [AZUMA, 1997]

Figure 99: Video see-through HMD conceptual diagram

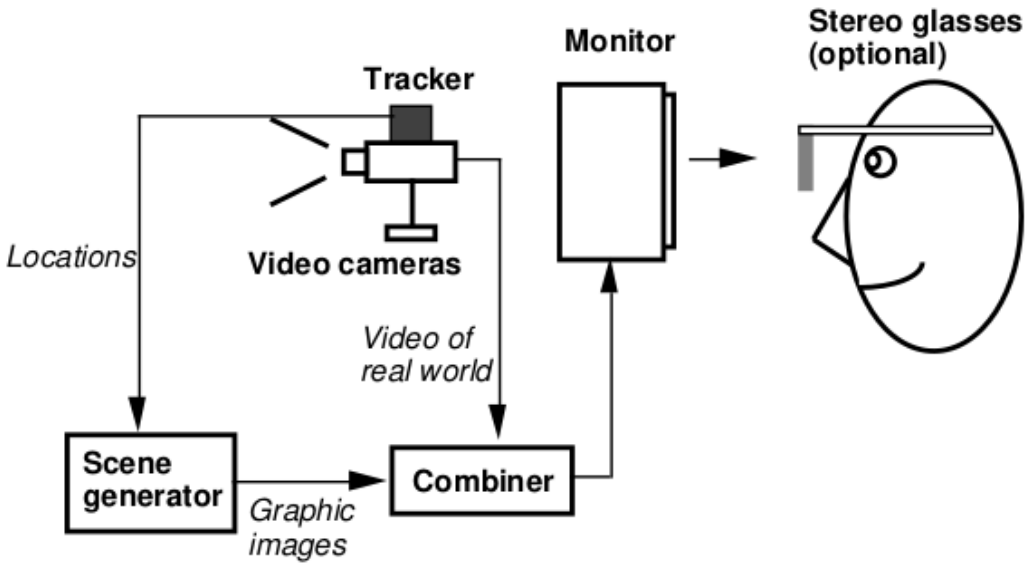


Source: [AZUMA, 1997]

As previously described in [MILGRAM et al., 1995], the work further addresses

monitor-based configurations with a conceptual diagram exemplified by Figure 100.

Figure 100: Monitor-based AR conceptual diagram



Source: [AZUMA, 1997]

The major obstacle identified in the work is defined as *the registration problem*. It refers to the proper alignment between real and projected virtual objects from a scene. There is a high requirement for accurate, long-range sensors and trackers to provide reliable information about the viewpoint and surrounding objects. In order to ensure a three-dimensional perception from image projections, the position and orientation of the viewpoint needs to be accurately known alongside the location of any virtual element to be displayed in the scene.

Note that there are numerous sources of errors that impact the registration problem. These errors have been further classified in two categories: *static* and *dynamic* errors. Static errors have been defined as the errors induced by optical distortions, failures in the tracking system, mechanical misalignment, and incorrect viewing parameters. Dynamic errors have been related with system delays.

In the next years, research conferences dedicated to augmented reality began: the IEEE/ACM International Workshop on Augmented Reality (IWAR) in 1998; International Symposium on Mixed Reality (ISMR) in 1999; and the International Symposium on Augmented Reality (ISAR) in 2000. These conferences promoted considerable advancements for the development of the field.

Further advancements in camera models and respective camera devices enabled the

development of solutions for the registration problem. In [CLARKE; FRYER, 1998], it is presented an overview of the development and evolution of models for lens distortion. The earliest applications for such methods were in mapping. In the 1950s, those involved with the manufacture and calibration of aerial cameras were concerned with the investigation and discussion of the characteristics of lens distortion. Different models have been proposed by different manufacturers, with corresponding different calibration methods. In [BROWN, 1966], an important analytical model for describing lens distortion have been proposed.

One particular class of calibration methods that were pursued by the time have been defined as *on-the-job* calibration. This technique places control points in the surrounding area of the object to be imaged. Alternatively, techniques that does not rely on control points have been referred as *self-calibration* methods within the work. With analytical modeling such as the one presented in [BROWN, 1966], it is theoretically possible to use optimization algorithms for determining lens calibration parameters and three-dimensional coordinates of measured targets simultaneously.

[ZHANG, 2000], an important flexible new technique for camera calibration have been proposed. The technique consists in the observation of a planar pattern for the camera at a few different orientations. Either the camera or the planar pattern can be freely moved. The corresponding calibration software have been made available for public use in order to advance the research of 3D computer vision from laboratory environments towards real world applications. Besides, standardized planar patterns have been made available to assist researchers in repeating the proposed methodology with any commercial printer.

In [AZUMA et al., 2001], a complement to the earlier work in [AZUMA, 1997] is proposed presenting representative examples of technological advances at that time. These have been referred as *enabling technologies*, as these developments were crucial for *enabling* the design of AR systems. Examples of such technologies that were further discussed include displays, tracking, registration and calibration. Despite all advancements reported by the time, there were still major technical obstacles limiting the wider use of AR. It was further recommended that displays, trackers, and AR systems required improvements to be more accurate, lighter, and less power consuming. Tracking in unprepared environments remained an enormous challenge.

In [FENG ZHOU; DUH; BILLINGHURST, 2008], a review of the principal works reported in earlier international conferences on *augmented reality* provided a road-map for future augmented reality research in the emerging field. The analysis was conducted over

276 papers presented in these conferences over ten years. There was significant work in core research areas such as *tracking techniques*, *interaction techniques*, *calibration and registration*, *AR applications* and *display techniques*. Other topics such as *evaluation/testing*, *mobile/wearable AR* and *visualization* were also reported indicating emerging research interests towards real world applications.

Tracking techniques have been further classified into *sensor-based*, *vision-based* or hybrid methods. Sensor-based tracking techniques are based on external sensors such as magnetic, acoustic, inertial, optical and/or mechanical sensors. These methods do not use camera measurements for tracking the user and/or environment. Among the works analyzed by the review, there were very few published papers on this type of tracking that uses only non-camera based sensors.

Vision-based tracking techniques uses image processing methods to calculate the camera pose relative to real world objects. Earlier conferences began by exploring marker based methods which depended on the instrumentation of the scenes. A notable trend was identified at the time towards the research on model-based tracking methods. These are techniques that explicitly uses geometrical models for tracking each object.

Hybrid methods combine several sensing technologies in an integrated tracking system. In earlier conferences, hybrid methods usually relied on markers. Later there was a growing consensus to combine inertial and computer vision technologies to provide closed-loop-type tracking. These two technologies offers complementary features. Inertial tracking is usually fast and robust but tends to drift due to noise accumulation. On the other hand, vision-based tracking has low jitter and no drift, but is slow, and outliers can occur. There is ongoing research on the development of hybrid tracking systems as the registration problem is still a major challenge for the popularization of augmented reality applications [BILLINGHURST; CLARK; LEE, 2015].

APPENDIX B – KALMAN FILTERS

This appendix presents the Kalman filter formulation as proposed in books about estimation and control [MAYBECK, 1979] and [MAYBECK, 1982]. Consider a normally distributed $n-$dimensional random vector $\vec{\eta}_X$ described by its mean $\vec{\mu}_X$ and covariance P_X. The corresponding probability density of the normally distributed random variable $\vec{\eta}_X$ is denoted as $pd_X(\varepsilon)$:

$$pd_X(\vec{\varepsilon}) = \frac{1}{(2\pi)^{n/2}\sqrt{(\det P_X)}} \exp\left(-\frac{1}{2}\left[\vec{\varepsilon} - \vec{\mu}_X\right]^T P_X^{-1}\left[\vec{\varepsilon} - \vec{\mu}_X\right]\right) \tag{B.1}$$

Note that the mean $\vec{\mu}_X$ is a vector and the covariance P_X is a matrix. A normally distributed random vector has the following statistics:

$$\mathbb{E}[\vec{\eta}_X] = \vec{\mu}_X \tag{B.2}$$

$$\mathbb{E}[\vec{\eta}_X \vec{\eta}_X^T] = P_X + \vec{\mu}_X \vec{\mu}_X^T \tag{B.3}$$

$$\mathbb{E}[(\vec{\eta}_X - \vec{\mu}_X)(\vec{\eta}_X - \vec{\mu}_X)^T] = P_X \tag{B.4}$$

The linear combination of Gaussian random vectors is also Gaussian. Let $\vec{\eta}_Y$ be another random vector described by a linear transformation A from $\vec{\eta}_X$. The mean $\vec{\mu}_Y$ and covariance P_Y of the random vector $\vec{\eta}_Y$ may be computed from μ_X and P_X:

$$\vec{\eta}_Y = A\vec{\eta}_X \Rightarrow \tag{B.5}$$

$$\vec{\mu}_Y = A\vec{\mu}_X \ , \ P_Y = AP_XA^T \tag{B.6}$$

A white Gaussian process is defined as a sequence of independent Gaussian random variables $\vec{\eta}_{X1}$, $\vec{\eta}_{X2}$, ... , $\vec{\eta}_{XN}$. Consider $\delta\vec{\eta}_k$ as the increment between two subsequent observations of the process:

$$\delta\vec{\eta}_k = \vec{\eta}_{Xk} - \vec{\eta}_{Xk-1} \tag{B.7}$$

Let $\vec{\beta}(t_k)$ denotes a random variable comprised by the sum of such Gaussian normal

increments:

$$\vec{\beta}(t_k) = \sum_{\tilde{k}=1}^{k} \delta\vec{\eta}_{\tilde{k}} \tag{B.8}$$

The random variable defined in Equation B.8 is commonly referred as a *Brownian motion*. A *Brownian motion* has the following statistics:

$$\mathbb{E}[\vec{\beta}(t)] = \vec{0} \tag{B.9}$$

$$\mathbb{E}[[\vec{\beta}(t_2) - \vec{\beta}(t_1)][\vec{\beta}(t_2) - \vec{\beta}(t_1)]^T] = \int_{t_1}^{t_2} Q_\beta(t)dt \qquad \forall t_1, t_2 \;\; ; \;\; t_1 \leq t_2 \tag{B.10}$$

where $Q_\beta(t)$ is a symmetric and positive semi-definite matrix. Considering these definitions of a *Gaussian* random variable and a *Brownian* motion, it is possible to define the Kalman filter formulation. Let $\vec{s}(t)$ be the state to be estimated by the filter. Assume the following linear system model for the state:

$$d\vec{s}(t) = F_S(t)\vec{s}(t)dt + B_S(t)\vec{u}(t)dt + G_S(t)d\vec{\beta}(t) \tag{B.11}$$

where $F_S(t)$ is an $n \times n$ dynamics matrix; $B_S(t)$ is an $n \times r$ input matrix; $u(t)$ is a $r-$vector comprising the system inputs; $G_S(t)$ is an $n \times s$ noise input matrix; β is a $s-$vector Brownian motion with statistics as indicated by Equations B.9 and B.10. The estimated state is modeled as a Gaussian random vector $\vec{\eta}_S(t)$, described by its mean $\vec{\mu}_S(t)$ and covariance $P_S(t)$. The mean of the Gaussian random vector $\vec{\eta}_S(t)$ corresponds to the state $\vec{s}(t)$ while the covariance $P_S(t)$ corresponds to the uncertainty of the filter estimation. The initial mean and covariance of the state is defined as $\vec{\mu}_{S_0}$ and P_{S_0}, respectively:

$$\mathbb{E}[\vec{\eta}_S(t_0)] = \vec{\mu}_S(t_0) = \vec{\mu}_{S_0} \tag{B.12}$$

$$\mathbb{E}[[\vec{\eta}_S(t_0) - \vec{\mu}_S(t_0)][\vec{\eta}_S(t_0) - \vec{\mu}_S(t_0)]^T] = P_S(t_0) = P_{S_0} \tag{B.13}$$

Measurements are available at discrete time points $t_1, t_2, ..., t_k$ and are modeled by a linear transformation from the state $\vec{s}(t)$ affected by an additive Gaussian noise $\vec{\eta}_Z$:

$$\vec{z}(t_k) = H_Z(t_k)\vec{s}(t_k) + \vec{\eta}_Z(t_k) \tag{B.14}$$

where $\vec{z}$ is an $m-$vector comprising measurements, H_Z is an $m \times n$ measurement matrix modeling the relationship between state and measurements, $\vec{s}(t_k)$ is the state vector at

time t_k; and $\vec{\eta}_Z$ is an random $m-$vector with statistics:

$$E[\vec{\eta}_Z(t_k)] = \vec{0} \tag{B.15}$$

$$E[\vec{\eta}_Z(t_k)\vec{\eta}_Z^T(t_k)] = R_Z(t_k) \tag{B.16}$$

At each time instant, filter estimates for the state $\vec{s}$ are improved with incoming measurements. Let $\vec{\mu}_S(t_k^-)$ and $P_S(t_k^-)$ be the mean and covariance of the estimator before updating with measurement $\vec{z}(t_k)$ at time t_k and let $\vec{\mu}_S(t_k^+)$ and $P_S(t_k^+)$ be the respective mean and covariance after the incorporation of $\vec{z}(t_k)$. Estimates are *updated* from incoming measurement $\vec{z}(t_k)$ with the following expression:

$$K(t_k) = P_S(t_k^-)H_Z^T(t_k)\Big[H_Z(t_k)P_S(t_k^-)H_Z^T(t_k) + R_Z(t_k)\Big]^{-1} \tag{B.17}$$

$$\mu_S(t_k^+) = \mu_S(t_k^-) + K(t_k)\Big[\vec{z}(t_k) - H_Z(t_k)\vec{\mu}_S(t_k^-)\Big] \tag{B.18}$$

$$P_S(t_k^+) = P_S(t_k^-) - K(t_k)H_Z(t_k)P_S(t_k^-) \tag{B.19}$$

The mean and covariance of the estimation is propagated between discrete times $t_1, ..., t_k$ with the dynamic model defined by matrices $F_S(t)$, $B_S(t)$, $G_S(t)$ and $Q_\beta(t)$. These estimates are propagated between each time interval $[t_k, t_{k+1}]$ by the integration of the following differential equations:

$$\dot{\vec{\mu}}_S(t) = F_S(t)\vec{\mu}_S(t) + B_S(t)\vec{u}(t) \tag{B.20}$$

$$\dot{P}_S(t) = F_S(t)P_S(t) + P_S(t)F_S^T(t) + G_S(t)Q_\beta(t)G_S^T(t) \tag{B.21}$$

$$t_k \leq t < t_{k+1}$$

For each time interval, initial conditions are the current estimate mean $\vec{\mu}_S(t_k)$ and covariance $P_S(t_k)$. The integration can be performed with a first order method over partitions of this interval. The previous formulations can be adapted for a particular class of nonlinear system with the *Extended Kalman Filter*. Let the dynamics of the state be described in the following nonlinear form:

$$d\vec{s}(t) = f(\vec{s}(t), \vec{u}(t), t)dt + G_S(t)d\vec{\beta}(t) \tag{B.22}$$

where $f(.)$ is a known $n-$vector of functions representing the state dynamics. Note that this replaces the linear model $[F_S(t)\vec{s}(t) + B_S(t)\vec{u}(t)]$ with $f(\vec{s}(t), \vec{u}(t), t)$. Similarly as in the previous development, the state is modeled with a Gaussian random vector $\vec{\eta}_S(t)$ defined by its mean $\vec{\mu}_S(t)$ and covariance $P_S(t)$. Initial estimates for the mean are given

by $\vec{\mu}_{S_0}$ and P_{S_0}:

$$\mathbb{E}[\vec{\eta}_S(t_0)] = \vec{\mu}_S(t_0) = \vec{\mu}_{S_0} \tag{B.23}$$

$$\mathbb{E}[[\vec{\eta}_S(t_0) - \mu_{S_0}][\vec{\eta}_S(t_0) - \mu_{S_0}]^T] = P_S(t_0) = P_{S_0} \tag{B.24}$$

These estimates are propagated between time intervals $t_1, t_2, ..., t_k$ similarly as before with the integration of the following differential equations:

$$\Rightarrow F_S(\vec{\mu}_S(t), t) = \left.\frac{\partial f}{\partial s}\right|_{\vec{s}(t)=\vec{\mu}_S(t)} \tag{B.25}$$

$$\dot{\vec{\mu}}_S(t) = f(\vec{\mu}_S(t), u(t), t) \tag{B.26}$$

$$\dot{P}_S(t) = F_S(\vec{\mu}_S(t), t)P_S(t) + P_S(t)F_S^T(\vec{\mu}_S(t), t) + G_S(t)Q_\beta(t)G_S^T(t) \tag{B.27}$$

$$t_k \leq t < t_{k+1}$$

At each discrete time instant t_k measurements are available. These are modeled with a nonlinear function $h(\vec{s}(t), t)$ affected by an additive Gaussian noise $\vec{\eta}_Z$ with zero mean and covariance $R_Z(t)$:

$$\vec{z}(t_k) = h(\vec{s}(t_k), t_k) + \vec{\eta}_Z \tag{B.28}$$

Assuming a similar notation as before, let $\vec{\mu}_S(t_k^-)$ and $P_S(t_k^-)$ be the mean and covariance of the estimator before updating with measurement $\vec{z}(t_k)$ at time t_k and let $\vec{\mu}_S(t_k^+)$ and $P_S(t_k^+)$ be the respective mean and covariance after the incorporation of $\vec{z}(t_k)$. The estimates are *updated* from incoming measurement $\vec{z}(t_k)$ with the following expressions:

$$\Rightarrow H_Z(\vec{\mu}_S(t_k^-), t_k^-) = \left.\frac{\partial h}{\partial s}\right|_{s(t)=\vec{\mu}_S(t_k^-)} = H_Z(t_k^-) \tag{B.29}$$

$$K(t_k) = P_S(t_k^-)H_Z^T(t_k^-)\Big[H_Z(t_k^-)P_S(t_k^-)H_Z^T(t_k^-) + R_Z(t_k)\Big]^{-1} \tag{B.30}$$

$$\vec{\mu}_S(t_k^+) = \vec{\mu}_S(t_k^-) + K(t_k)\Big[\vec{z}(t_k) - h(\vec{\mu}_S(t_k^-), t_k)\Big] \tag{B.31}$$

$$P_S(t_k^+) = \Big[I - K(t_k)H_Z(t_k^-)\Big]P_S(t_k^-)\Big[I - K(t_k)H_Z(t_k^-)\Big]^{-1} + K(t_k)R_Z(t_k)K^T(t_k) \tag{B.32}$$

From a computational perspective, the implementation of the filter requires the definition of a model for the system with functions $f(\vec{s}(t), \vec{u}(t), t)$ and $h(\vec{s}(t), t)$ that are linearized yielding matrices F_S and H_Z. The matrix R_Z is related with the Gaussian noise model for the sensor measurements and Q_β is related with the system dynamic noise modeled as Brownian perturbations. An initial estimate for the filter state must be

provided with mean $\vec{\mu}_{S_0}$ and covariance P_{S_0}. The initial mean $\vec{\mu}_{S_0}$ of the estimator is the expected initial state for the system, while P_{S_0} provides an indication of the uncertainty associated with this initial estimate.

APPENDIX C – MARITIME NAVIGATIONAL ASSISTANCE

C.1 Inertial Navigation Systems

An *inertial navigation system* (**INS**) is an equipment with an inertial measuring unit that provides specific force and angular rate measurements with respect to the instrument case. A navigation processor integrates Equations 3.80, 3.81 and 3.84 to provide the instrument state with respect to a reference coordinate system.

Ideally, measurements of specific force are composed by the acceleration subtracted by the gravity force. Considering that the axes of the instrument are aligned with the ship coordinate system, let f_{is}^s be the measured specific force from the instrument:

$$f_{is}^s = a_{is}^s - g_S^s = a_{is}^s - R_e^s g_S^e \tag{C.1}$$

where g_S^s, g_S^e denotes the gravity vector at the ship position expressed in the ship and earth coordinate system, respectively. Thus, the acceleration in the Earth may be computed from the specific force with knowledge of the gravity vector and ship orientation with respect to the Earth:

$$R_s^e f_{is}^s = R_s^e(a_{is}^s - g_S^s) = R_s^e a_{is}^s - R_s^e g_S^s = a_{is}^e - g_S^e \Rightarrow \tag{C.2}$$

$$a_{is}^e = R_s^e f_{is}^s + g_S^e \tag{C.3}$$

Let $\hat{g}_S^e$ and $\hat{R}_s^e$ be estimates for the gravity vector and the orientation of the ship with respect to the Earth coordinate system. The corresponding estimation $\hat{a}_{is}^e$ be for the acceleration with respect to the Earth frame may be computed as:

$$\hat{a}_{is}^e = \hat{R}_s^e f_{is}^s + \hat{g}_S^e = \hat{R}_s^e(a_{is}^s - g_S^s) + \hat{g}_S^e = \left(\hat{R}_s^e a_{is}^s\right) + \left(\hat{g}_S^e - \hat{R}_s^e g_S^s\right) \tag{C.4}$$

Neglecting errors in the Earth gravity model, note that the estimated acceleration $\tilde{a}^e_{is}$ is equal to the truth acceleration a^e_{is} if the estimated rotation matrix $\hat{R}^e_s$ is equal to the truth rotation matrix R^e_s. Furthermore, typical sensors deviates from the truth specific force due to multiple source of errors. A common error model may be expressed as follows. Let f^s_{is} be the truth specific force and let $\hat{f}^s_{is}$ be the measurement from a typical instrument:

$$\hat{f}^s_{is} = b_a + (I_{3\times3} + M_a)f^s_{is} + \eta_{fa} \tag{C.5}$$

where b_a, M_a and η_{fa} are the accelerometer *bias*; *scale-factor and cross-coupling matrix*; and *random noise*; respectively. The random noise η_{fa} may be modeled as a Gaussian random variable with zero mean and nonzero covariance w_{fa}. Typically, the accelerometer *bias* b_a ranges from 10^{-4} m/s^2 to 10^{-1} m/s^2. It is customary to express the accelerometer *bias* in terms of gravity units, yielding values from 0.01 mg to 10 mg. The terms of the dimensionless *scale-factor and cross-coupling matrix* ranges from 10^{-4} to 10^{-2}. Random noise from the accelerometer is usually characterized in terms of its root power spectral density (**PSD**), which is customary given by $\mu g/\sqrt{Hz}$. For typical accelerometers, the root PSD ranges from 20 $\mu g/\sqrt{Hz}$ to 1000 $\mu g/\sqrt{Hz}$. Note that in order to obtain the corresponding covariance for the random noise samples, the root PSD needs to be multiplied by the root of the sampling rate (or divided by the root of the sampling interval) [GROVES, 2008].

Similarly, let ω^s_{is} be the truth angular rate of the ship and $\hat{\omega}^s_{is}$ be the corresponding measurement affected by errors:

$$\hat{\omega}^s_{is} = b_g + (I_{3\times3} + M_g)\omega^s_{is} + G_g f^s_{is} + \eta_{\omega g} \tag{C.6}$$

where b_g, M_g, G_g and $\eta_{\omega g}$ are the gyrometer *bias*; *scale-factor and cross-coupling matrix*; *gravity-cross-coupling matrix*; and *random noise*; respectively. Typical gyrometer *bias* b_g ranges from 10^{-3} $^\circ/hr$ to 10^2 $^\circ/hr$, where $1^\circ/hr = 4.848 \times 10^{-6}$ rad/s. Similarly as with the accelerometer error model, the terms of the dimensionless *scale-factor and cross-coupling matrix* ranges from 10^{-4} to 10^{-2}. The root PSD of typical gyrometers ranges from 0.002 $^\circ/\sqrt{hr}$ to 1 $^\circ/\sqrt{hr}$, where $1^\circ/\sqrt{hr} = 2.909 \times 10^{-4}$ $rad/\sqrt{s}$ [GROVES, 2008]. Procedures for estimating error parameters from Equations C.5 and C.6 are commonly referred as *IMU calibration*. Such *calibration* is beyond the scope of the present work.

Let $\hat{f}^s_{is}(t)$ and $\hat{\omega}^s_{is}(t)$ be the IMU measurements at time t, and consider estimations $\hat{p}^e_{es}(t)$, $\hat{v}^e_{es}(t)$ and $\hat{R}^e_s(t)$ for the ship position, velocity and orientation with respect to the Earth, respectively. Additionally, consider a regular sampling interval ΔT for the IMU

and let $\alpha_{is}^{s}(t)$ be the angle increment measured by the gyrometer during the sampling interval ΔT:

$$\alpha_{is}^{s}(t) = \begin{bmatrix} \alpha_{is}^{s,i}(t) \\ \alpha_{is}^{s,j}(t) \\ \alpha_{is}^{s,k}(t) \end{bmatrix} = \int_{t-\Delta T}^{t} \hat{\omega}_{is}^{s}(\tau)d\tau = \begin{bmatrix} \int_{t-\Delta T}^{t} \hat{\omega}_{is}^{s,x}(\tau)d\tau \\ \int_{t-\Delta T}^{t} \hat{\omega}_{is}^{s,y}(\tau)d\tau \\ \int_{t-\Delta T}^{t} \hat{\omega}_{is}^{s,z}(\tau)d\tau \end{bmatrix} \approx \begin{bmatrix} \hat{\omega}_{is}^{s,x}(t)\Delta T \\ \hat{\omega}_{is}^{s,y}(t)\Delta T \\ \hat{\omega}_{is}^{s,z}(t)\Delta T \end{bmatrix} \tag{C.7}$$

$$\alpha_{is}^{s}(t) \approx \hat{\omega}_{is}^{s}(t)\Delta T \tag{C.8}$$

Let $\hat{R}_{s}^{e}(t_k)$ be the rotation matrix after IMU measurements at a time t_k and $\hat{R}_{s}^{e}(t_{k-1})$ be the rotation matrix from previous INS measurement t_{k-1}. The following expressions may be used for updating $\hat{R}_{s}^{e}(t_k)$ from gyrometer measurements [GROVES, 2008]:

$$\left[\alpha_{is}^{s}\hat{\times}\right] = \begin{bmatrix} 0 & -\alpha_{is,k}^{s} & \alpha_{is,j}^{s} \\ \alpha_{is,k}^{s} & 0 & -\alpha_{is,i}^{s} \\ -\alpha_{is,j}^{s} & \alpha_{is,i}^{s} & 0 \end{bmatrix} \approx \begin{bmatrix} 0 & -\hat{\omega}_{is}^{s,k}\Delta T & \hat{\omega}_{is}^{s,j}\Delta T \\ \hat{\omega}_{is}^{s,k}\Delta T & 0 & -\hat{\omega}_{is}^{s,i}\Delta T \\ -\hat{\omega}_{is}^{s,j}\Delta T & \hat{\omega}_{is}^{s,i}\Delta T & 0 \end{bmatrix} \tag{C.9}$$

$$\hat{R}_{s+}^{s-}\left(\alpha_{is}^{s}(t_k)\right) = I_{3\times 3} + \frac{\sin|\alpha_{is}^{s}|}{|\alpha_{is}^{s}|}\left[\alpha_{is}^{s}\hat{\times}\right] + \frac{1-\cos|\alpha_{is}^{s}|}{|\alpha_{is}^{s}|^2}\left[\alpha_{is}^{s}\hat{\times}\right]^2 \tag{C.10}$$

$$\hat{R}_{s}^{e}(t_k) = \hat{R}_{s}^{e}(t_{k-1})\hat{R}_{s+}^{s-}\left(\alpha_{is}^{s}(t_k)\right) - \Omega_{ie}^{e}\hat{R}_{s}^{e}(t_{k-1})\Delta T \tag{C.11}$$

As any rotation matrix should be orthogonal with an unitary determinant, it is interesting to ensure such property with the following additional operations. Orthogonalization:

$$\hat{R}_{s}^{e} = \begin{bmatrix} \hat{i}_{s}^{e,x} & \hat{j}_{s}^{e,x} & \hat{k}_{s}^{e,x} \\ \hat{i}_{s}^{e,y} & \hat{j}_{s}^{e,y} & \hat{k}_{s}^{e,y} \\ \hat{i}_{s}^{e,z} & \hat{j}_{s}^{e,z} & \hat{k}_{s}^{e,z} \end{bmatrix} = \begin{bmatrix} \hat{r}_{l1} \\ \hat{r}_{l2} \\ \hat{r}_{l3} \end{bmatrix} \Rightarrow \Delta_{ij} = \hat{r}_{li}\hat{r}_{lj}^{T} \Rightarrow \tag{C.12}$$

$$\hat{r}_{l1}(+) = \hat{r}_{l1}(-) - 0.5\Delta_{12}\hat{r}_{l2}(-) - 0.5\Delta_{13}\hat{r}_{l3}(-) \tag{C.13}$$

$$\hat{r}_{l2}(+) = \hat{r}_{l2}(-) - 0.5\Delta_{12}\hat{r}_{l1}(-) - 0.5\Delta_{23}\hat{r}_{l3}(-) \tag{C.14}$$

$$\hat{r}_{l3}(+) = \hat{r}_{l3}(-) - 0.5\Delta_{13}\hat{r}_{l1}(-) - 0.5\Delta_{23}\hat{r}_{l2}(-) \tag{C.15}$$

Note that if $\hat{R}_{s}^{e}$ is already orthogonal, the corresponding terms Δ_{ij} are going to be zero yielding no modifications to $\hat{R}_{s}^{e}$. Normalization:

$$\hat{r}_{li}(+) = \left(\frac{2}{1+\hat{r}_{li}(-)\hat{r}_{li}(-)^{T}}\right)\hat{r}_{li}(-) \tag{C.16}$$

Further, regarding the velocity and position of the system, the specific force must be transformed from the INS axes to the earth frame. Consider the *average coordinate*

transformation matrix $\bar{R}_s^e(t)$ with respect to the sampling interval ΔT [GROVES, 2008]:

$$\hat{R}_{\bar{s}}^{s^-}\left(\alpha_{is}^s(t_k)\right) = I_{3\times 3} + \frac{1-\cos|\alpha_{is}^s|}{|\alpha_{is}^s|^2}\left[\alpha_{is}^s\hat{\times}\right] + \frac{1}{|\alpha_{is}^s|^2}\left(1-\frac{\sin|\alpha_{is}^s|}{|\alpha_{is}^s|}\right)\left[\alpha_{is}^s\hat{\times}\right]^2 \tag{C.17}$$

$$\bar{R}_s^e(t_k) = \hat{R}_s^e(t_{k-1})\hat{R}_{\bar{s}}^{s^-}\left(\alpha_{is}^s(t_k)\right) - 0.5\Omega_{ie}^e\hat{R}_s^e(t_{k-1})\Delta T \tag{C.18}$$

The average rotation matrix $\bar{R}_s^e(t)$ must be orthonormalized similarly as with the previous additional computations given by Equations C.12 and C.16. The specific force in the earth frame may be approximated from the accelerometer measurement $\tilde{f}_{is}^s(t)$ and $\bar{R}_s^e(t)$ as:

$$\hat{f}_{is}^e(t_k) = \bar{R}_s^e(t_k)\hat{f}_{is}^s(t_k) \tag{C.19}$$

Let $\hat{v}_{es}^e(t_k)$ and $\hat{p}_{es}^e(t_k)$ be the velocity and position of the system with respect to the Earth after a INS measurement at time t_k, respectively; and let $\hat{v}_{es}^e(t_{k-1})$ and $\hat{p}_{es}^e(t_{k-1})$ be the corresponding velocity and position from the previous measurement. The following expressions may be used for updating the system velocity:

$$\hat{v}_{es}^e(t_k) = \hat{v}_{es}^e(t_{k-1}) + \left(\hat{a}_{is}^e(t_k) - \left(\Omega_{ie}^e\Omega_{ie}^e\right)\hat{p}_{es}^e(t_{k-1}) - 2\Omega_{ie}^e\hat{v}_{es}^e(t_{k-1})\right)\Delta T \tag{C.20}$$

$$\begin{aligned}\hat{v}_{es}^e(t_k) =& \hat{v}_{es}^e(t_{k-1}) \\ &+ \left(\hat{f}_{is}^e(t_k) + \left(\tilde{g}_S^e(\hat{p}_{es}^e(t_{k-1})) - \Omega_{ie}^e\Omega_{ie}^e\hat{p}_{es}^e(t_{k-1})\right) - 2\Omega_{ie}^e\hat{v}_{es}^e(t_{k-1})\right)\Delta T\end{aligned} \tag{C.21}$$

where the Earth gravity model is a function of the system position $\hat{p}_{es}^e$ as in Equation 3.54. The system position may be updated with the following expression:

$$\hat{p}_{es}^e(t_k) = \hat{p}_{es}^e(t_{k-1}) + \left(\hat{v}_{es}^e(t_{k-1}) + \hat{v}_{es}^e(t_k)\right)\frac{\Delta T}{2} \tag{C.22}$$

C.2 Global Navigation Satellite Systems

Considering that each signal propagates from the satellite to the user antenna with known velocity, the distance ρ traveled by the signal can be determined from the time difference between the signal transmission and arrival. This distance traveled by each signal is usually referred as *range* measurements. Let t_{st} be the time of signal transmission,

t_{sa} be the time of signal arrival and c be the signal propagation velocity.

$$\rho = c\Big(t_{sa} - t_{st}\Big) \tag{C.23}$$

Note that t_{st} is determined by the clock onboard of the satellite while t_{sa} is determined by the clock onboard of the user antenna. Thus, both clocks needs to be adequately synchronized. Note that an offset δt_{at} between both clocks yields an error $\delta\rho_{rc}$ in the *range* measurement:

$$\delta\rho_{rc} = c\delta t_{as} \tag{C.24}$$

$$\rho = c\Big(t_{sa} - t_{st} + \delta t_{at}\Big) = c\Big(t_{sa} - t_{st}\Big) + \delta\rho_{rc} \tag{C.25}$$

Assume that the position p^s_{sa} of the antenna with respect to the ship frame is known and time-invariant. The antenna position p^e_{ea} with respect to Earth frame may be computed from the ship state as:

$$p^e_{ea} = p^e_{es} + R^e_s p^s_{sa} \tag{C.26}$$

Consider a constellation with N_s satellites regularly distributed in a circular orbit and let $p^e_{es,j}$ be the position of the $j-$th satellite with respect to the Earth at the signal transmission. Further, let ρ_j be the corresponding range measurement between the $j-$th satellite and the user antenna. Neglecting the rotation of the Earth during signal propagation, a simple approximation $\tilde{\rho}_j$ for the range measurement ρ_j may be expressed as:

$$\rho_j \approx \tilde{\rho}_j = \sqrt{\Big[p^e_{es,j} - p^e_{ea}\Big]^T \Big[p^e_{es,j} - p^e_{ea}\Big]} \tag{C.27}$$

$$t_{sa} - t_{st,j} \approx \frac{\tilde{\rho}_j}{c} \tag{C.28}$$

Equation C.27 would be exact if both positions were described with respect to an inertial coordinate system. Consider an inertial coordinate system I aligned with the Earth frame at the time of signal arrival, and let $R^I_e(t)$ be the rotation matrix relating

this inertial coordinate system with the Earth frame at time t:

$$R_e^I(t_{sa}) = I_{3\times 3} \tag{C.29}$$

$$R_e^I(t) = \begin{bmatrix} \cos\left(\omega_{ie}(t-t_{sa})\right) & -\sin\left(\omega_{ie}(t-t_{sa})\right) & 0 \\ \sin\left(\omega_{ie}(t-t_{sa})\right) & \cos\left(\omega_{ie}(t-t_{sa})\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.30}$$

Note that for small rotations, the rotation matrix from Equation C.30 may be approximated:

$$R_e^I(t) \approx \begin{bmatrix} 1 & -\omega_{ie}(t-t_{sa}) & 0 \\ \omega_{ie}(t-t_{sa}) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.31}$$

At the time of signal transmission, from Equation C.28:

$$R_e^I(t_{st,j}) \approx \begin{bmatrix} 1 & -\omega_{ie}(t_{st,j}-t_{sa}) & 0 \\ \omega_{ie}(t_{st,j}-t_{sa}) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & \omega_{ie}(\frac{\tilde{\rho}_j}{c}) & 0 \\ -\omega_{ie}(\frac{\tilde{\rho}_j}{c}) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.32}$$

Thus, an approximation $\bar{\rho}_j$ for the range measurement ρ_j that includes the Earth rotation may be expressed as:

$$\rho_j \approx \bar{\rho}_j = \sqrt{\left[R_e^I(t_{st,j})p_{es,j}^e - p_{ea}^e\right]^T \left[R_e^I(t_{st,j})p_{es,j}^e - p_{ea}^e\right]} \tag{C.33}$$

Considering a time offset δt_{as} between the receiver clock from the user equipment and the satellite constellation and the corresponding range offset $\delta\rho_{rc}$:

$$\rho_j \approx \bar{\rho}_j + \delta\rho_{rc} = \sqrt{\left[R_e^I(t_{st,j})p_{es,j}^e - p_{ea}^e\right]^T \left[R_e^I(t_{st,j})p_{es,j}^e - p_{ea}^e\right]} + \delta\rho_{rc} \tag{C.34}$$

Finally, the range measurement is assumed to be affected by an additive Gaussian noise $\eta_{\rho p}$ with zero mean and covariance $w_{\rho p}$:

$$\rho_j = \bar{\rho}_j + \delta\rho_{rc} + \eta_{\rho p} \tag{C.35}$$

$$\rho_j = \sqrt{\left[R_e^I(t_{st,j})p_{es,j}^e - p_{ea}^e\right]^T \left[R_e^I(t_{st,j})p_{es,j}^e - p_{ea}^e\right]} + \delta\rho_{rc} + \eta_{\rho p} \tag{C.36}$$

Note that there are additional error terms that affects *range* measurements such as ionosphere and troposphere delays. Furthermore, there are other errors related with the

receiver such as thermal noise and RF interference [GROVES, 2008]. Such additional effects are neglected in the present work.

The antenna state with respect to the Earth frame may be determined by the minimization of errors between *range* measurements and corresponding estimates. Note that there are three parameters regarding the antenna position and one parameter regarding the receiver offset. Thus, it is necessary at least four simultaneous measurements for the determination of the antenna state with respect to the Earth frame. An analytical solution for such computation is hard to be defined. Typically, an iterative procedure is implemented around a predicted antenna position and receiver offset. One iteration of such procedure may be summarized as follows.

Let $\hat{p}^e_{ea}$ and $\hat{p}^e_{es,j}$ be predicted estimates for the position of the user antenna and the $j-$th satellite, respectively. Additionally, let $\delta\hat{\rho}_{rc}$ be an initial estimation for the receiver clock offset. Consider a column vector $s_a = [\hat{p}^e_{ea}, \delta\hat{\rho}_{rc}]^T$ representing the predicted antenna state as a concatenation of antenna position and receiver clock offset.

An initial estimation $\hat{\rho}_j$ for the *range* measurement may be computed as:

$$\hat{\tilde{\rho}}_j = \sqrt{\left[\hat{p}^e_{es,j} - \hat{p}^e_{ea}\right]^T \left[\hat{p}^e_{es,j} - \hat{p}^e_{ea}\right]} \Rightarrow \hat{R}^I_e(t_{st,j}) = \begin{bmatrix} 1 & \omega_{ie}\hat{\tilde{\rho}}_j c^{-1} & 0 \\ -\omega_{ie}\hat{\tilde{\rho}}_j c^{-1} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.37}$$

$$\hat{\rho}_j = \sqrt{\left[\hat{R}^I_e(t_{st,j})\hat{p}^e_{es,j} - \hat{p}^e_{ea}\right]^T \left[\hat{R}^I_e(t_{st,j})\hat{p}^e_{es,j} - \hat{p}^e_{ea}\right]} + \delta\hat{\rho}_{rc} \tag{C.38}$$

Consider a column vector $\vec{\rho}$ concatenating N_s measurements ρ_j:

$$\vec{\rho} = \begin{bmatrix} \rho_1 & \dots & \rho_j & \dots & \rho_{N_s} \end{bmatrix}^T \tag{C.39}$$

Note that partial derivatives of $\vec{\rho}$ are function of an unitary vector $u^e_{as,j}$ pointing from the user antenna towards the $j-$th satellite:

$$u^e_{as,j} \approx \frac{p^e_{es,j} - p^e_{ea}}{\left|p^e_{es,j} - p^e_{ea}\right|} \ , \ \hat{u}^e_{as,j} \approx \frac{\hat{p}^e_{es,j} - \hat{p}^e_{ea}}{\left|\hat{p}^e_{es,j} - \hat{p}^e_{ea}\right|} \ \Rightarrow \ \frac{\partial\vec{\rho}}{\partial s_a} \approx G_e = \begin{bmatrix} -(\hat{u}^e_{as,1})^T & 1 \\ \dots & 1 \\ -(\hat{u}^e_{as,j})^T & 1 \\ \dots & 1 \\ -(\hat{u}^e_{as,N_s})^T & 1 \end{bmatrix} \tag{C.40}$$

Therefore, let $[\hat{p}^e_{ea+}, \delta\hat{\rho}_{rc+}]$ be improved estimates for the antenna state after one iter-

ation.

$$\begin{bmatrix} \hat{p}^e_{ea+} \\ \delta\hat{\rho}_{rc+} \end{bmatrix} = \begin{bmatrix} \hat{p}^e_{ea} \\ \delta\hat{\rho}_{rc} \end{bmatrix} + (G_e^T G_e)^{-1} G_e^T \begin{bmatrix} \rho_1 - \hat{\rho}_1 \\ \cdots \\ \rho_j - \hat{\rho}_j \\ \cdots \\ \rho_{N_s} - \hat{\rho}_{N_s} \end{bmatrix} \tag{C.41}$$

Note that the procedure described by Equations C.38, C.40 and C.41 needs to be repeated until convergence. For each new set of measurements from the GNSS, it is customary to initialize the current predicted antenna state s_a with the solution from the previous iteration.

Another type of measurement provided by GNSS user equipment is commonly referred as *range-rate* measurements, which is theoretically related with the derivative of each *range* measurement. Let ω^s_{es} be the angular velocity of the ship with respect to the Earth frame. The velocity of the antenna with respect to the Earth frame may be computed as:

$$v^e_{ea} = v^e_{es} + R^e_s\Big(\omega^s_{es} \times p^s_{sa}\Big) = v^e_{es} + R^e_s\Big(\omega^s_{is} \times p^s_{sa}\Big) - R^e_s\Big(\omega^s_{ie} \times p^s_{sa}\Big) \tag{C.42}$$

where $\times$ is the cross product operator. Note that gyrometers typically measures the angular velocity ω^s_{is} of the ship with respect to a inertial frame. The angular velocity of the Earth frame is assumed to be known and constant. Let $\dot{\rho}_j$ be the *range-rate* measurement between the $j-$th satellite and the user antenna and $v^e_{es,j}$ be the velocity of the $j-$th satellite with respect to the Earth frame. An approximation $\bar{\dot{\rho}}_j$ for the *range-rate* measurement $\dot{\rho}_j$ may be expressed as:

$$\dot{\rho}_j \approx \bar{\dot{\rho}}_j = (u^e_{as,j})^T \Big[R^I_e(t_{st,j})(v^e_{esj} + \Omega^e_{ie} p^e_{es,j}) - (v^e_{ea} + \Omega^e_{ie} p^e_{ea})\Big] \tag{C.43}$$

where $R^I_e(t_{st,j})$ is the rotation matrix defined by C.32, $u^e_{as,j}$ is the unitary vector defined by Equation C.40 and Ω^e_{ie} is the skew-symmetric form for the angular velocity of the Earth with respect to the inertial frame. Furthermore, consider two additional error terms: (1) a *range-rate* drift error $\delta\dot{\rho}_{rc}$ associated with the derivative of the receiver clock offset; (2) an additive Gaussian noise $\eta_{\rho v}$ with zero mean and covariance $w_{\rho v}$:

$$\dot{\rho}_j = \bar{\dot{\rho}}_j + \delta\dot{\rho}_{rc} + \eta_{\rho v} \tag{C.44}$$

$$\dot{\rho}_j = (u^e_{as,j})^T \Big[R^I_e(t_{st,j})(v^e_{esj} + \Omega^e_{ie} p^e_{es,j}) - (v^e_{ea} + \Omega^e_{ie} p^e_{ea})\Big] + \delta\dot{\rho}_{rc} + \eta_{\rho v} \tag{C.45}$$

Note that there are additional sources of error that affect *range-rate* measurements

[GROVES, 2008]. These additional effects are neglected in the present work. Similarly as before, the user antenna velocity may be determined by the minimization of errors between *range-rate* measurements and corresponding estimates. In addition to the previous predicted parameters from the user antenna position estimation, let $\hat{v}^e_{ea}$ and $\hat{v}^e_{es,j}$ be predicted estimates for the velocity of the user antenna and the $j-$th satellite, respectively. Further, let $\delta\hat{\dot{\rho}}_{rc}$ be the predicted *range-rate* drift error.

$$\hat{\dot{\rho}}_j = (\hat{u}^e_{as,j})^T \left[\hat{R}^I_e(t_{st,j})(\hat{v}^e_{es,j} + \Omega^e_{ie}\hat{p}^e_{es,j}) - (\hat{v}^e_{ea} + \Omega^e_{ie}\hat{p}^e_{ea}) \right] + \delta\hat{\dot{\rho}}_{rc} \tag{C.46}$$

Similarly as before, consider a column vector $[\hat{v}^e_{ea}, \delta\hat{\dot{\rho}}_{rc}]^T$ representing predicted parameters and let $[\hat{v}^e_{ea+}, \delta\hat{\dot{\rho}}_{rc+}]^T$ be improved estimates after one iteration.

$$\begin{bmatrix} \hat{v}^e_{ea+} \\ \delta\hat{\dot{\rho}}_{rc+} \end{bmatrix} = \begin{bmatrix} \hat{v}^e_{ea} \\ \delta\hat{\dot{\rho}}_{rc} \end{bmatrix} + (G_e^T G_e)^{-1} G_e^T \begin{bmatrix} \dot{\rho}_1 - \hat{\dot{\rho}}_1 \\ \cdots \\ \dot{\rho}_j - \hat{\dot{\rho}}_j \\ \cdots \\ \dot{\rho}_{N_s} - \hat{\dot{\rho}}_{N_s} \end{bmatrix} \tag{C.47}$$

Note that there are alternative procedures for the determination of the position and velocity of the user antenna, which are beyond the scope of the present work.

C.3 Compass

Consider a *compass coordinate system* S_ψ (index ψ) associated with the instrument axes. Assume that the position $p^s_{s\psi}$ and orientation R^s_ψ of the instrument case is time-invariant and known with respect to the ship frame. The compass position $p^e_{e\psi}$ and orientation R^e_ψ with respect to the Earth frame may be computed as:

$$p^e_{e\psi} = p^e_{es} + R^e_s p^s_{s\psi} \tag{C.48}$$

$$R^e_\psi = R^e_s R^s_\psi \tag{C.49}$$

Let $[L_\psi, \lambda_\psi, h_\psi]$ be the curvilinear position of the compass computed from the cartesian position $p^e_{e\psi}$ with Equations 3.49, 3.46, 3.47 and 3.48. The rotation matrix relating the Earth frame with the Navigation frame may be computed from the curvilinear position with expression 3.50:

$$R_e^n(L_\psi, \lambda_\psi, h_\psi) = \begin{bmatrix} -\sin(L_\psi)\cos(\lambda_\psi) & -\sin(L_\psi)\sin(\lambda_\psi) & \cos(L_\psi) \\ -\sin(\lambda_\psi) & \cos(\lambda_\psi) & 0 \\ -\cos(L_\psi)\cos(\lambda_\psi) & -\cos(L_\psi)\sin(\lambda_\psi) & -\sin(L_\psi) \end{bmatrix} \tag{C.50}$$

Thus, the rotation matrix relating the compass and the navigation frame may be computed as:

$$R_\psi^n = R_e^n R_\psi^e = R_e^n R_s^e R_\psi^s \tag{C.51}$$

Recall that the rotation matrix with respect to the navigation frame may be computed with Equation 3.20 as a function of three sequential rotations by angles $[\phi_{n\psi}, \theta_{n\psi}, \psi_{n\psi}]$:

$$R_\psi^n(\phi_{n\psi}, \theta_{n\psi}, \psi_{n\psi}) = \tag{C.52}$$

$$\begin{bmatrix} c(\theta_{n\psi})c(\psi_{n\psi}) & -c(\phi_{n\psi})s(\psi_{n\psi}) + s(\phi_{n\psi})s(\theta_{n\psi})c(\psi_{n\psi}) & s(\phi_{n\psi})s(\psi_{n\psi}) + c(\phi_{n\psi})s(\theta_{n\psi})c(\psi_{n\psi}) \\ c(\theta_{n\psi})s(\psi_{n\psi}) & c(\phi_{n\psi})c(\psi_{n\psi}) + s(\phi_{n\psi})s(\theta_{n\psi})s(\psi_{n\psi}) & -s(\phi_{n\psi})c(\psi_{n\psi}) + c(\phi_{n\psi})s(\theta_{n\psi})s(\psi_{n\psi}) \\ -s(\theta_{n\psi}) & s(\phi_{n\psi})c(\theta_{n\psi}) & c(\phi_{n\psi})c(\theta_{n\psi}) \end{bmatrix} \tag{C.53}$$

Let $\psi_{n\psi}$ be the real angle between the instrument case and the north direction from the navigation frame. A simple model for $\psi_{n\psi}$ may be defined as a function of the rotation matrix R_ψ^n between the instrument frame and the navigation frame:

$$R_\psi^n = \begin{bmatrix} r_{\psi 11}^n & r_{\psi 12}^n & r_{\psi 13}^n \\ r_{\psi 21}^n & r_{\psi 22}^n & r_{\psi 23}^n \\ r_{\psi 31}^n & r_{\psi 32}^n & r_{\psi 33}^n \end{bmatrix} \Rightarrow \psi_{n\psi} = \arctan 2(r_{\psi 21}^n, r_{\psi 11}^n) \tag{C.54}$$

Henceforth, the corresponding measurement of $\psi_{n\psi}$ is assumed to be affected by an additive Gaussian noise η_ψ with zero mean and nonzero covariance w_ψ:

$$\hat{\psi}_{n\psi} = \psi_{n\psi} + \eta_\psi \tag{C.55}$$

C.4 Integrated Navigation Systems

Let $\hat{p}_{es}^e$, $\hat{v}_{es}^e$ and $\hat{R}_s^e$ be the inertial navigation solution for the ship position, velocity and orientation with respect to the Earth frame, respectively; and let δp_{es}^e, δv_{es}^e and $\delta \psi_{es}^e$ be the corresponding error of the inertial solution with respect to the truth state of the

ship. Consider a *Kalman* filter for the estimation of errors δp^e_{es}, δv^e_{es} and $\delta \psi^e_{es}$ alongside errors in IMU biases δb_a and δb_g. Let $\vec{s}$ be a state vector concatenating such error terms:

$$\vec{s} = \begin{bmatrix} \delta\psi^e_{es} & \delta v^e_{es} & \delta p^e_{es} & \delta b_a & \delta b_g \end{bmatrix}^T \tag{C.56}$$

In a *Kalman* filter, the estimated state is a *Gaussian* random variable described by its mean $\vec{\mu}_S$ and covariance P_S. Let ΔT_Z be the time interval between filter iterations. The initial state $\vec{\mu}_{S_0}$ of the filter is initialized with a null column vector:

$$\vec{\mu}_{S_0} = \vec{0} \tag{C.57}$$

The covariance matrix of the filter needs to be initialized with estimates for the uncertainties of each state component:

$$P_{S_0} = \begin{bmatrix} \delta\hat{\Psi} I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \delta\hat{v} I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \delta\hat{p} I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \delta\hat{b_a} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \delta\hat{b_g} I_{3\times3} \end{bmatrix} \tag{C.58}$$

Recall that the *Kalman* filter is implemented with two sequential operations. In the *propagation* step, which is given by Equations B.25, B.26 and B.27, the filter mean $\vec{\mu}_S$ and covariance P_S are *propagated* between two consecutive measurement instants t_k and t_{k-1} with system matrices F_S, G_S and Q_β. In the *update* step, which is given by Equations B.29, B.30, B.31 and B.32, the filter mean $\vec{\mu}_S$ and covariance P_S are *updated* at time t_k with measurement matrices H_Z and R_Z.

The system matrix F_S is a 15×15 matrix that express how the error in the estimation is propagated through time. This matrix depends on the current IMU measurements ($\hat{f}^s_{is}$ and $\hat{\omega}^s_{is}$) as well as the current inertial solution $\hat{p}^e_{es}$, $\hat{v}^e_{es}$ and $\hat{R}^e_s$. Let $\hat{L}_s$ be the latitude associated with the estimated position $\hat{p}^e_{es}$. An expression for the system matrix F_S is given by [GROVES, 2008]:

$$F_S = \begin{bmatrix} -\Omega_{ie}^e & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \hat{R}_s^e \\ F_{S21} & -2\Omega_{ie}^e & F_{S23} & \hat{R}_s^e & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{C.59}$$

where F_{S21} and F_{S23} are 3×3 matrices given by:

$$\hat{f}_{is}^e = \hat{R}_s^e \hat{f}_{is}^s \Rightarrow \begin{bmatrix} \hat{f}_{is}^{e,x} \\ \hat{f}_{is}^{e,y} \\ \hat{f}_{is}^{e,z} \end{bmatrix} = \begin{bmatrix} i_s^{e,x} & j_s^{e,x} & k_s^{e,x} \\ i_s^{e,y} & j_s^{e,y} & k_s^{e,y} \\ i_s^{e,z} & j_s^{e,z} & k_s^{e,z} \end{bmatrix} \begin{bmatrix} \hat{f}_{is}^{s,i} \\ \hat{f}_{is}^{s,j} \\ \hat{f}_{is}^{s,k} \end{bmatrix} \Rightarrow \tag{C.60}$$

$$F_{S21} = \Big[-\hat{f}_{is}^e \hat{\times} \Big] = \Big[-\hat{R}_s^e \hat{f}_{is}^s \hat{\times} \Big] = \begin{bmatrix} 0 & \hat{f}_{is}^{e,z} & -\hat{f}_{is}^{e,y} \\ -\hat{f}_{is}^{e,z} & 0 & \hat{f}_{is}^{e,x} \\ \hat{f}_{is}^{e,y} & -\hat{f}_{is}^{e,x} & 0 \end{bmatrix} \tag{C.61}$$

$$r_{eS}^e(\hat{L}_s) = \frac{R_0^{WGS}}{\sqrt{1-(e^{WGS})^2 \sin^2(\hat{L}_s)}} \sqrt{\cos^2(\hat{L}_s) + (1-(e^{WGS})^2)^2 \sin^2(\hat{L}_s)} \tag{C.62}$$

$$F_{S23} = -\Big(\frac{2}{r_{eS}^e(\hat{L}_s)|\hat{p}_{es}^e|} \Big) \Big[\Big(\bar{g}_S^e(\hat{p}_{es}^e) \Big) \Big(\hat{p}_{es}^e \Big)^T \Big] \tag{C.63}$$

The system noise described by matrices G_S and Q_β can be defined as:

$$G_S = I_{15\times15} \quad , \quad Q_\beta = \begin{bmatrix} \hat{w}_{\omega g}^2 I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \hat{w}_{fa}^2 I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \hat{w}_{ba}^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \hat{w}_{bg}^2 I_{3\times3} \end{bmatrix} \Delta T_Z \tag{C.64}$$

where $\hat{w}_{\omega g}$ and $\hat{w}_{fa}$ are estimates for the power spectral density of the gyrometer and accelerometer random noise, respectively; $\hat{w}_{bg}$ and $\hat{w}_{ba}$ are estimates for the power spectral density of the gyrometer and accelerometer bias variation, respectively.

In a loosely-coupled integration, each measurement corresponds to one of the filter states. Thus, the filter measurement matrix may be expressed in terms of blocks of null matrices $0_{3\times3}$ and identity matrices $I_{3\times3}$. Consider initially a *loosely coupled* integration with measurements from an user antenna of a GNSS. This equipment outputs position

and velocity estimations of the user antenna with respect to the Earth frame. Let $\hat{p}^e_{ea}$ be an estimation for the user antenna position with respect to the Earth, and let $\hat{p}^s_{sa}$ be an estimation for the user antenna position with respect to the ship frame. The ship position $\tilde{p}^e_{es}$ with respect to the Earth may be computed from the GNSS output with the inverse of Equation C.26:

$$\tilde{p}^e_{es} = \hat{p}^e_{ea} - \hat{R}^e_s \hat{p}^s_{sa} \tag{C.65}$$

Similarly, the ship velocity $\tilde{v}^e_{es}$ may be determined with the inverse of Equation C.42:

$$\tilde{v}^e_{es} = \hat{v}^e_{ea} - \hat{R}^e_s \Big(\hat{\omega}^s_{es} \times \hat{p}^s_{sa} \Big) \tag{C.66}$$

Considering $\hat{p}^e_{es}$ and $\hat{v}^e_{es}$ as the reference position and velocity of the ship estimated by the INS, the measurement innovation z^{GNSS} feed to the filter is given by:

$$z^{GNSS} = \begin{bmatrix} \tilde{p}^e_{es} - \hat{p}^e_{es} \\ \tilde{v}^e_{es} - \hat{v}^e_{es} \end{bmatrix} = \begin{bmatrix} -\delta p^e_{es} \\ -\delta v^e_{es} \end{bmatrix} \tag{C.67}$$

The corresponding measurement matrix H^{GNSS}_Z is given by:

$$H^{GNSS}_Z = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & -I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{C.68}$$

The measurement noise covariance matrix R^{GNSS}_Z should corresponds with the uncertainty of the GNSS estimations for the INS state. Let $\hat{w}_{\rho p}$ be the uncertainty of the GNSS position estimation and $\hat{w}_{\rho v}$ be the uncertainty of the GNSS velocity estimation. The measurement noise matrix R^{GNSS}_Z for the error measurements can be computed as:

$$R^{GNSS}_Z = \begin{bmatrix} \hat{w}_{\rho p} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \hat{w}_{\rho v} I_{3\times3} \end{bmatrix} \tag{C.69}$$

An additional type of sensor that are useful for *tracker* integration are compasses. Compasses provides the angle ψ_c between the instrument case with respect to the true north, which is the first axis of the local navigation frame. Assume that the instrument case is aligned with the ship axes. The rotation matrix $\hat{R}^e_s$ estimated by the INS with respect to the Earth frame is transformed to the local navigation frame with the computation of $\hat{R}^n_e$ from current INS position $\hat{p}^e_{es}$:

$$\hat{R}^n_s = \hat{R}^n_e \hat{R}^e_s \tag{C.70}$$

Recall that the rotation matrix $\hat{R}_s^n$ between the ship and the local navigation frame may be expressed as a function of three Euler angles $[\phi_{ns}, \theta_{ns}, \psi_{ns}]$:

$$R_s^n(\phi_{ns}, \theta_{ns}, \psi_{ns}) = \tag{C.71}$$

$$\begin{bmatrix} c(\theta_{ns})c(\psi_{ns}) & -c(\phi_{ns})s(\psi_{ns}) + s(\phi_{ns})s(\theta_{ns})c(\psi_{ns}) & s(\phi_{ns})s(\psi_{ns}) + c(\phi_{ns})s(\theta_{ns})c(\psi_{ns}) \\ c(\theta_{ns})s(\psi_{ns}) & c(\phi_{ns})c(\psi_{ns}) + s(\phi_{ns})s(\theta_{ns})s(\psi_{ns}) & -s(\phi_{ns})c(\psi_{ns}) + c(\phi_{ns})s(\theta_{ns})s(\psi_{ns}) \\ -s(\theta_{ns}) & s(\phi_{ns})c(\theta_{ns}) & c(\phi_{ns})c(\theta_{ns}) \end{bmatrix} \tag{C.72}$$

Therefore, it is possible to estimate Euler angles $[\hat{\phi}_{ns}, \hat{\theta}_{ns}, \hat{\psi}_{ns}]$ from the rotation matrix $\hat{R}_s^n$:

$$\hat{R}_s^n = \begin{bmatrix} \hat{r}_{s11}^n & \hat{r}_{s12}^n & \hat{r}_{s13}^n \\ \hat{r}_{s21}^n & \hat{r}_{s22}^n & \hat{r}_{s23}^n \\ \hat{r}_{s31}^n & \hat{r}_{s32}^n & \hat{r}_{s33}^n \end{bmatrix} \Rightarrow \begin{bmatrix} \hat{\phi}_{ns} \\ \hat{\theta}_{ns} \\ \hat{\psi}_{ns} \end{bmatrix} = \begin{bmatrix} \arctan 2(\hat{r}_{s32}^n, \hat{r}_{s33}^n) \\ -\arcsin(\hat{r}_{s31}^n) \\ \arctan 2(\hat{r}_{s21}^n, \hat{r}_{s11}^n) \end{bmatrix} \tag{C.73}$$

Let $\hat{R}_s^n$ be the rotation matrix computed from the INS position $\hat{p}_{es}^e$ and orientation $\hat{R}_s^e$. Further, let $\tilde{R}_s^n$ be the rotation matrix computed from the compass measurement ψ_c along with remaining Euler angles $[\hat{\phi}_{ns}, \hat{\theta}_{ns}]$:

$$\tilde{R}_s^n(\phi_{ns}, \theta_{ns}, \psi_c) =$$

$$\begin{bmatrix} c(\theta_{ns})c(\psi_c) & -c(\phi_{ns})s(\psi_c) + s(\phi_{ns})s(\theta_{ns})c(\psi_c) & s(\phi_{ns})s(\psi_c) + c(\phi_{ns})s(\theta_{ns})c(\psi_c) \\ c(\theta_{ns})s(\psi_c) & c(\phi_{ns})c(\psi_c) + s(\phi_{ns})s(\theta_{ns})s(\psi_c) & -s(\phi_{ns})c(\psi_c) + c(\phi_{ns})s(\theta_{ns})s(\psi_c) \\ -s(\theta_{ns}) & s(\phi_{ns})c(\theta_{ns}) & c(\phi_{ns})c(\theta_{ns}) \end{bmatrix} \tag{C.74}$$

As the filter is modeled with respect to the Earth frame, it is necessary to represent each orientation with respect to the Earth frame as well. The rotation matrix from Equation C.74 may be transformed to the Earth frame with $\hat{R}_e^n$:

$$\tilde{R}_s^e = \hat{R}_n^e \tilde{R}_s^n \tag{C.75}$$

For a given pair $\tilde{R}_s^e$ and $\hat{R}_s^e$, it is interesting to determine an angular displacement Ψ^e that transforms $\hat{R}_s^e$ to $\tilde{R}_s^e$. Let $\delta\hat{R}_s^e$ be the error between the reference orientation $\hat{R}_s^e$

estimated by the INS and the orientation $\tilde{R}_s^e$ computed with the compass:

$$\Psi^e = \begin{bmatrix} \Psi_x \\ \Psi_y \\ \Psi_z \end{bmatrix} \Rightarrow \left[\Psi^e \hat{\times}\right] = \begin{bmatrix} 0 & -\Psi_z & \Psi_y \\ \Psi_z & 0 & -\Psi_x \\ -\Psi_y & \Psi_x & 0 \end{bmatrix} \tag{C.76}$$

$$\delta\hat{R}_s^e = \left(\tilde{R}_s^e\right) - \left(\left(I_{3\times 3} + \left[\Psi^e \hat{\times}\right]\right)\hat{R}_s^e\right) \tag{C.77}$$

$$\delta\hat{R}_s^e = \begin{bmatrix} \tilde{r}_{s11}^n & \tilde{r}_{s12}^n & \tilde{r}_{s13}^n \\ \tilde{r}_{s21}^n & \tilde{r}_{s22}^n & \tilde{r}_{s23}^n \\ \tilde{r}_{s31}^n & \tilde{r}_{s32}^n & \tilde{r}_{s33}^n \end{bmatrix} - \begin{bmatrix} 1 & -\Psi_z & \Psi_y \\ \Psi_z & 1 & -\Psi_x \\ -\Psi_y & \Psi_x & 1 \end{bmatrix} \begin{bmatrix} \hat{r}_{s11}^n & \hat{r}_{s12}^n & \hat{r}_{s13}^n \\ \hat{r}_{s21}^n & \hat{r}_{s22}^n & \hat{r}_{s23}^n \\ \hat{r}_{s31}^n & \hat{r}_{s32}^n & \hat{r}_{s33}^n \end{bmatrix} \tag{C.78}$$

Note that the error δR_s^e is a 3×3 matrix where each element corresponds to a different error component. Thus, it is possible to determine the angular displacement Ψ^e by the minimization of δR_s^e:

$$\delta\hat{R}_s^e = \begin{bmatrix} \delta\hat{r}_{s11}^n & \delta\hat{r}_{s12}^n & \delta\hat{r}_{s13}^n \\ \delta\hat{r}_{s21}^n & \delta\hat{r}_{s22}^n & \delta\hat{r}_{s23}^n \\ \delta\hat{r}_{s31}^n & \delta\hat{r}_{s32}^n & \delta\hat{r}_{s33}^n \end{bmatrix} \Rightarrow \tag{C.79}$$

$$\delta\hat{r}_{s11}^n = \tilde{r}_{s11}^n - \left(\hat{r}_{s11}^n - \Psi_z\hat{r}_{s21}^n + \Psi_y\hat{r}_{s31}^n\right) \tag{C.80}$$

$$\delta\hat{r}_{s12}^n = \tilde{r}_{s12}^n - \left(\hat{r}_{s12}^n - \Psi_z\hat{r}_{s22}^n + \Psi_y\hat{r}_{s32}^n\right) \tag{C.81}$$

$$\delta\hat{r}_{s13}^n = \tilde{r}_{s13}^n - \left(\hat{r}_{s13}^n - \Psi_z\hat{r}_{s23}^n + \Psi_y\hat{r}_{s33}^n\right) \tag{C.82}$$

$$\delta\hat{r}_{s21}^n = \tilde{r}_{s21}^n - \left(\Psi_z\hat{r}_{s11}^n + \hat{r}_{s21}^n - \Psi_x\hat{r}_{s31}^n\right) \tag{C.83}$$

$$\delta\hat{r}_{s22}^n = \tilde{r}_{s22}^n - \left(\Psi_z\hat{r}_{s12}^n + \hat{r}_{s22}^n - \Psi_x\hat{r}_{s32}^n\right) \tag{C.84}$$

$$\delta\hat{r}_{s23}^n = \tilde{r}_{s23}^n - \left(\Psi_z\hat{r}_{s13}^n + \hat{r}_{s23}^n - \Psi_x\hat{r}_{s33}^n\right) \tag{C.85}$$

$$\delta\hat{r}_{s31}^n = \tilde{r}_{s31}^n - \left(-\Psi_y\hat{r}_{s11}^n + \Psi_x\hat{r}_{s21}^n + \hat{r}_{s31}^n\right) \tag{C.86}$$

$$\delta\hat{r}_{s32}^n = \tilde{r}_{s32}^n - \left(-\Psi_y\hat{r}_{s12}^n + \Psi_x\hat{r}_{s22}^n + \hat{r}_{s32}^n\right) \tag{C.87}$$

$$\delta\hat{r}_{s33}^n = \tilde{r}_{s33}^n - \left(-\Psi_y\hat{r}_{s13}^n + \Psi_x\hat{r}_{s23}^n + \hat{r}_{s33}^n\right) \tag{C.88}$$

Let $\tilde{\Psi}^e$ be the angular displacement that minimizes the matricial error δR_s^e from Equation C.79. Further, let $\hat{\psi}^e$ be an angular displacement computed from consecutive gyrometer measurements. The measurement innovation z^{CMPS} feed to the integration filter is given by:

$$z^{CMPS} = \left[\tilde{\Psi}^e - \hat{\psi}^e\right] = \left[-\delta\psi_{es}^e\right] \tag{C.89}$$

The corresponding measurement matrix is given by:

$$H_Z^{CMPS} = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{C.90}$$

Similarly as in the previous case of GNSS integration, the measurement noise covariance matrix R_Z^{CMPS} should correspond with the uncertainty of the compass estimations. Let $\hat{w}_\psi$ be the uncertainty associated with the compass measurements and consider R_Z^{CMPS} as the corresponding measurement noise matrix:

$$R_Z^{CMPS} = \hat{w}_\psi I_{3\times3} \tag{C.91}$$

Note that there are alternative procedures for the determination of noise matrices G_S, Q_β, R_Z^{CMPS} and R_Z^{GNSS}. Such procedures are beyond the scope of the current work. Thus, further error modeling is suggested for future research.

Periodically, the reference solution computed from the INS is *corrected* with filter error estimates. These estimates are obtained from the mean $\vec{\mu}_S^*$ of the filter solution after the *update* step:

$$\vec{\mu}_S^* = \begin{bmatrix} \delta^*\psi_{es}^e & \delta^*v_{es}^e & \delta^*p_{es}^e & \delta^*b_a & \delta^*b_g \end{bmatrix}^T \tag{C.92}$$

After each *correction* of the reference solution provided by the INS, all error estimates from the filter are set to zero. Let $\acute{p}_{es}^e$, $\acute{v}_{es}^e$ and $\acute{R}_s^e$ be the corrected ship position, velocity and orientation; respectively:

$$\acute{p}_{es}^e = \hat{p}_{es}^e - \delta^*p_{es}^e \tag{C.93}$$

$$\acute{v}_{es}^e = \hat{v}_{es}^e - \delta^*v_{es}^e \tag{C.94}$$

$$\acute{R}_s^e \approx \left(I_{3\times3} - \left[\delta^*\psi_{es}^e\right]_\times\right)\hat{R}_s^e \tag{C.95}$$