Modeling, Identification and Control, Vol. 42, No. 3, 2021, pp. 99–112, ISSN 1890–1328

Estimation of Ship-Deck Motion using Lidar, Gyroscopes and Cameras

Hans K.R. Holen [1] Alexander M. Sjøberg [1] Olav Egeland [1]

[1] *Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway. E-mail:* {hans.k.r.holen, alexander.m.sjoberg, olav.egeland}@ntnu.no

Abstract

This paper presents a system for the estimation of ship deck motion using camera, lidar and gyroscopes. A camera is used in a vision system that is based on the detection of lines as input to a vanishing point detector. This is done under a Manhattan assumption for man-made structures where the majority of lines are along 3 orthogonal axes. Two sets of parallel orthogonal lines are detected for the ship deck, and this is used to estimate the attitude using a complementary filter with input from lidar and gyroscopes. Since the vision algorithm depends on lines rather than points, the system is more resistant to occlusions like vision algorithms based on point tracking. In addition, a lidar is used to measure the distance between the sensor frame and the plane, and gyroscopes are used to improved the accuracy of the estimates. The system is validated in real time in lab experiments on a model of a ship.

Keywords: Vanishing points, sensor fusion, vision, offshore motion compensation

1 Introduction

Crane operations from one ship to another is still heavily dependent of human interaction. The consequences of a failure could be grave, with potential damage on cargo and ship, and injuries on human operators involved in the process.

The topic of marine crane control has been an active topic of research in the last decades, as seen from the reviews in Abdel-Rahman et al. (2003) and Ramli et al. (2017). Also the topic of sensors for crane control in marine applications has been studied, where an important research field is the measurement of payload motion, which is considered in Rauscher et al. (2018) where a IMU was mounted in the crane hook, and in Tysse et al. (2021) where a camera is used to track the motion of markers on the wire. Also the tracking of the target ship motion can be important when there is significant wave motion to ensure that the payload of the crane is give a soft landing on the deck of the target ship. This was investigated in Tørdal and Hovland (2019), where an extended Kalman filter (EKF) was formulated to estimate the relative pose between a crane ship and a target ship for safe landing of the payload on the target ship. This system used an accurate industrial motion reference unit (MRU) in combination with a laser tracker.

A related sensor problem appears in the control and automatic landing of unmanned aerial vehicles (UAVs). In Lin et al. (2017) a system for the landing of a UAV on a cluttered ship-deck environment was presented, where the H marker of a helicopter landing pad was identified. In Patruno et al. (2019) a system for automatic landing of a UAV on ship deck was presented, where a vision system was used to identify the helicopter landing pad. Autonomous landing on a moving target with a camera system and fiducial markers was studied in Polvara et al. (2017), where a UAV was landed on a moving target in an experiment.

To track the motion of a target ship with camera it is required to have reliable feature extraction. A ship deck is an example of a man-made structure where

doi:10.4173/mic.2021.3.1

there are many parallel lines, and many of these lines will cluster orthogonal directions along the coordinate axes of the ship-fixed coordinate frame, which is referred to as the Manhattan assumption Coughlan and Yuille (2003). Line detection is less prone to problems with occlusions than point detection. As shown in works such as Zitnick and Dollár (2014), this may be beneficial when determining relative pose. Perspective-n-Lines (PnL) is a common way to determine the relative camera pose using the correspondences between 3D reference features and their 2D projections, where the features are lines Xu et al. (2017). To apply PnL it is required that the 3D lines are known in the scene. This is possible in target ship tracking if lines of the ship have been identified from the geometric design data of the ship. However, this is not a practical solution for a crane ship that is to serve a wide range of target ships, so some other technique would be useful to determine relative attitude without any prior knowledge of the target ship.

Points at infinity in the scene are mapped to vanishing points in the image Hartley and Zisserman (2004), Ma et al. (2003). Vanishing points have been important in calibration Caprile and Torre (1990), and are used in calibration toolboxes Bouguet (1999). Vanishing points are also used in combination with the Manhattan assumption in Coughlan and Yuille (2003). Then it assumed that the majority of lines are along three orthogonal directions. This result is then used to find the relative orientation between the camera and the Manhattan directions. Early work on the detection detection of vanishing points was based on the Hough transform Barnard (1983), Magee and Aggarwal (1984) where the direction of the intersection for each of the possible line pairs was computed and accumulated in the angular bins.

One of the major issues with estimating vanishing points is to know which line segments correspond to the relevant vanishing points, and which lines that are outliers. A way to detect vanishing points is to apply segmentation methods, based on random sampling methods such as multi-RANSAC Zuliani et al. (2005), however, this would require knowledge of a number of models in the image. Other solutions are based on J-linkage Toldo and Fusiello (2008) and T-linkage Magri and Fusiello (2014), which give good results for clustering of lines. An important contribution is due to Tardif Tardif who used J-linkage for clustering of lines to identify vanishing points according to the Manhattan assumption. This technique has certain limitations when it is used for tracking the motion of a target ship. In particular, J-linkage clustering method may be non-optimal when the majority of line segments are parallel and belong to only one of the vanishing points. In this case the Jaccard distance in the J-linkage method becomes very small for small line clusters. Moreover, line segments that coincide happen to disturb the clustering process and result in erroneous detected vanishing points.

In this paper we present a system for camera tracking of the deck of a target ship. The lines that characterize the ship deck are identified with a clustering technique based on J-linkage and the Manhattan assumption as used in Tardif. We replace the Jaccard distance used in Tardif with another similarity measure, and use preprocessing to reduce the number of line segments in the clustering to achieve real-time performance. The camera system is used in combination with a lidar, and one gyroscope in the crane tip, and one gyroscope on the target ship. The attitude estimator was based on the combination of the complementary filter on SO(3) Mahony et al. (2008), and a low pass filter on the lidar measurements. The main difference from Tørdal and Hovland (2019) is that we use inexpensive sensors like cameras, lidar and gyroscopes, which makes it necessary to develop the required vision algorithms. The paper does not address the control of the crane payload, only the sensor system. The proposed system was validated in experiments in a lab setup.

This paper is organized as follows: Section 2 contains the preliminaries, Section 3 describes the sensors used in our experiment setup, Section 4 describes the observer used for plane estimation, the algorithm for estimating vanishing points is described in Section 5 and finally experiments results are shown and presented in Section 6.

2 Preliminaries

2.1 The Special Orthogonal Group SO(3)

Consider the set of rotation matrices $SO(3)$, which is a matrix Lie group, and the corresponding Lie algebra $so(3)$ (Mueller, 2019). A rotation matrix $\boldsymbol{R} \in SO(3)$ is given by the exponential mapping

$$\boldsymbol{R} = \exp(\boldsymbol{u}^{\times}) \in SO(3) \tag{1}$$

where

$$\boldsymbol{u}^{\times} = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \in so(3) \tag{2}$$

is the skew symmetric form of the vector $\boldsymbol{u} = [u_x,\ u_y,\ u_z]^{\mathrm{T}}$. The vector form of the logarithm can be written $\boldsymbol{u} = \theta\boldsymbol{k}$ where $\theta = \|\boldsymbol{u}\|$ is the rotation angle and the unit vector $\boldsymbol{k}$ is along the axis of rotation. Then if $|\theta| < \pi$ the logarithm of $\boldsymbol{R}$ is

$$\boldsymbol{u}^{\times} = \log \boldsymbol{R} \in so(3) \tag{3}$$

The angular distance $d_a(\boldsymbol{R}, \boldsymbol{I})$ from $\boldsymbol{R}$ to the identity $\boldsymbol{I}$ is given by (Hartley et al., 2013)

$$d_a(\boldsymbol{R}, \boldsymbol{I})^2 = \|\boldsymbol{u}\| = \theta \tag{4}$$

The angular distance is also known as the geodesic distance. The angular distance is biinvariant, and it follows that the angular distance between two rotation matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ is $d_a(\boldsymbol{R}_1, \boldsymbol{R}_2) = d_a(\boldsymbol{R}_2^{\mathrm{T}}\boldsymbol{R}_1, \boldsymbol{I})$.

The exponential in $SO(3)$ can be computed for all $\boldsymbol{u}^\times$ from

$$\exp \boldsymbol{u}^\times = \boldsymbol{I} + \mathrm{sinc}(\theta)\boldsymbol{u}^\times + \frac{1}{2}\mathrm{sinc}^2\left(\frac{\theta}{2}\right)\boldsymbol{u}^\times\boldsymbol{u}^\times \tag{5}$$

This result is obtained from (Park, 1995) by inserting $\mathrm{sinc}(x) = \sin(x)/x$ and $\mathrm{sinc}^2(x/2) = 2(1 - \cos(x))/x^2$. Computation around $x = 0$ can be done with the Taylor series expansions $\mathrm{sinc}(x) = 1 - x^2/6 + \ldots$. The logarithm can be computed for $\|u\| < \pi$ from

$$\boldsymbol{u}^\times = \frac{\theta}{2\sin\theta}\left(\boldsymbol{R} - \boldsymbol{R}^{\mathrm{T}}\right), \quad \theta = \arccos\left(\frac{\mathrm{tr}\boldsymbol{R} - 1}{2}\right) \tag{6}$$

where the Taylor series expansion $x/\sin(x) = 1 + x^2/6 + \ldots$ can be used around $x = 0$.

The kinematic differential equation on $SO(3)$ is given as

$$\dot{\boldsymbol{R}} = \boldsymbol{\omega}_l^\times\boldsymbol{R} = \boldsymbol{R}\boldsymbol{\omega}_r^\times \tag{7}$$

where $\boldsymbol{\omega}_l$ is the left angular velocity, and $\boldsymbol{\omega}_r$ is the right angular velocity.

2.2 Computation of the Mean Element on SO(3)

Consider the set of rotation matrices $\boldsymbol{R}_i \in SO(3)$, $i = 1, \ldots, N$. Let $\boldsymbol{\mu} \in SO(3)$ be the average rotation with respect to the angular distance, and let $\tilde{\boldsymbol{R}}_i = \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{R}_i$ be the deviation between $\boldsymbol{\mu}$ and $\boldsymbol{R}_i$ (Hartley et al., 2013). The logarithm of $\tilde{\boldsymbol{R}}_i$ is denoted $\boldsymbol{u}_i^\times = \log(\tilde{\boldsymbol{R}}_i)$, where $\boldsymbol{u}_i = \theta_i\boldsymbol{k}_i$. Suppose that $|\theta_i| < \pi/2$. Then the mean $\boldsymbol{\mu}$ minimizes

$$\sum_{i=1}^N d_a(\boldsymbol{R}_i, \boldsymbol{\mu})^2 = \sum_{i=1}^N d_a(\tilde{\boldsymbol{R}}_i, \boldsymbol{I})^2 = \sum_{i=1}^N \theta_i^2 \tag{8}$$

The mean $\boldsymbol{\mu}$ will satisfy the condition (Moakher, 2002)

$$\sum_{i=1}^N \log(\tilde{\boldsymbol{R}}_i) = \boldsymbol{0} \tag{9}$$

and can be computed by iteration as follows (Manton, 2005). Initialize the mean with $\boldsymbol{\mu}$ as $\boldsymbol{\mu} := \boldsymbol{R}_j$ for an arbitrary rotation matrix $\boldsymbol{R}_j$ and choose a tolerance value $\lambda > 0$. The average rotation matrix $\boldsymbol{\mu}$ is then found from

$$\boldsymbol{\epsilon}_i^\times = \log(\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{R}_i) \tag{10}$$

$$\bar{\boldsymbol{\epsilon}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\epsilon}_i \tag{11}$$

$$\boldsymbol{\mu} : = \boldsymbol{\mu}\exp(\bar{\boldsymbol{\epsilon}}^\times) \tag{12}$$

where (10), (11) and (12) are iterated until $\sqrt{\bar{\boldsymbol{\epsilon}}^{\mathrm{T}}\bar{\boldsymbol{\epsilon}}} \leq \lambda$.

2.3 Points and Lines in 2D

A point $\boldsymbol{p} = [x, y]^{\mathrm{T}} \in \mathbb{R}^2$ in the plane can be represented by the homogeneous vector $\boldsymbol{x} = [\boldsymbol{p}^{\mathrm{T}}, 1]^{\mathrm{T}} \in \mathbb{P}^2$ in the projective space $\mathbb{P}^2$. The homogeneous point $\boldsymbol{x}$ and the scaled homogeneous point $\alpha\boldsymbol{x}$ represent the same Euclidean point $\boldsymbol{p}$ for $\alpha \neq 0$.

Consider the homogeneous point $\boldsymbol{x} = [\alpha\boldsymbol{a}^{\mathrm{T}}, 1]^{\mathrm{T}}$ where $\boldsymbol{a} \in \mathbb{R}^2$ is a unit vector. Then $(1/\alpha)\boldsymbol{x} = [\boldsymbol{a}^{\mathrm{T}}, 1/\alpha]^{\mathrm{T}}$ will represent the same point $\boldsymbol{p} = \alpha\boldsymbol{a}^{\mathrm{T}}$. In particular, if α tends to infinity, then $\boldsymbol{p}$ will tend to infinity in the direction of $\boldsymbol{a}$, and $(1/\alpha)\boldsymbol{x} = [\boldsymbol{a}^{\mathrm{T}}, 1/\alpha]^{\mathrm{T}}$ will tend to $\boldsymbol{z} = [\boldsymbol{a}^{\mathrm{T}}, 0]^{\mathrm{T}}$, which is said to be a point at infinity.

A line in the the plane can be represented by the homogeneous vector $\boldsymbol{\ell} = [a, b, c]^{\mathrm{T}}$ where $\boldsymbol{n} = [a, b]^{\mathrm{T}}$ is the normal vector to the line. Then a point $\boldsymbol{x}$ is on the line $\boldsymbol{\ell}$ if and only if $\boldsymbol{\ell}^{\mathrm{T}}\boldsymbol{x} = 0$. If it is given that two homogeneous points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are on the same line $\boldsymbol{\ell}$, then the line is given by $\boldsymbol{\ell} = \boldsymbol{x}_1^\times\boldsymbol{x}_2$. It is straightforward to verify that this implies $\boldsymbol{\ell}^{\mathrm{T}}\boldsymbol{x}_1 = \boldsymbol{\ell}^{\mathrm{T}}\boldsymbol{x}_2 = 0$.

Consider the two lines $\boldsymbol{\ell}_1 = [\boldsymbol{n}_1^{\mathrm{T}}, c_1]^{\mathrm{T}}$ and $\boldsymbol{\ell}_2 = [\boldsymbol{n}_2^{\mathrm{T}}, c_2]^{\mathrm{T}}$, and suppose that the angle between the two lines is $\alpha \in [-\pi/2, \pi/2]$. Then the angle between the two normal vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ will also be α, and it follows that $\boldsymbol{n}_1^\times\boldsymbol{n}_2 = \sin\alpha|\boldsymbol{n}_1||\boldsymbol{n}_2|$ and $\boldsymbol{n}_1^{\mathrm{T}}\boldsymbol{n}_2 = \cos\alpha|\boldsymbol{n}_1||\boldsymbol{n}_2|$. This can be used to define the distance measure

$$d(\boldsymbol{\ell}_1, \boldsymbol{\ell}_2) = \frac{\boldsymbol{n}_1^\times\boldsymbol{n}_2}{\boldsymbol{n}_1^{\mathrm{T}}\boldsymbol{n}_2} = \tan\alpha \tag{13}$$

between the two lines $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_2$.

2.4 Points and Planes in 3D

A point $\boldsymbol{t} = [x, y, z]^{\mathrm{T}} \in \mathbb{R}^3$ in the 3 dimensional Euclidean space $\mathbb{R}^3$ can be represented by the homogeneous point $\boldsymbol{X} = [\boldsymbol{t}^{\mathrm{T}}, 1]^{\mathrm{T}} \in \mathbb{P}^3$. A point $\boldsymbol{X} = [\boldsymbol{a}^{\mathrm{T}}, 0]^{\mathrm{T}}$ where $\boldsymbol{a} \in \mathbb{R}^3$ is a point at infinity in the direction of $\boldsymbol{a}$.

A plane is represented by the homogeneous vector $\boldsymbol{\pi} = [\boldsymbol{n}^{\mathrm{T}}, d]^{\mathrm{T}}$ where $\boldsymbol{n} = [n_1, n_2, n_3]^{\mathrm{T}}$ is the normal

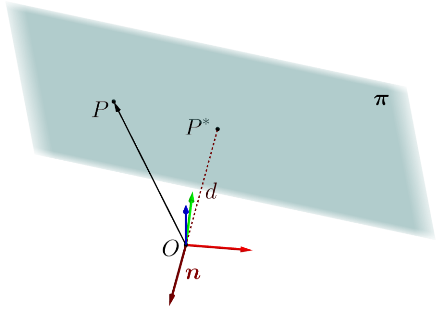

Figure 1: A plane $\boldsymbol{\pi}$ can be described in terms of a surface vector $\boldsymbol{n}$ of unit length and a scalar d which describes the shortest distance between the origin O of the selected coordinate frame and the point P^* which is the point on $\boldsymbol{\pi}$ being the closest to O.

vector of the plane, and d is related to the distance from the origin to the plane. The homogeneous point $\boldsymbol{X}$ is on the plane $\boldsymbol{\pi}$ if and only if

$$\boldsymbol{\pi}^{\mathrm{T}}\boldsymbol{X} = 0$$

It follows that if $\boldsymbol{X} = [\boldsymbol{t}^{\mathrm{T}}, 1]^{\mathrm{T}}$ is on the plane, then $d = -\boldsymbol{n}^{\mathrm{T}}\boldsymbol{t}$, and it follows that if $\boldsymbol{n}$ is a unit vector, then $-d$ is the distance from the origin to the point P^* on the plane that is closest to the origin. An illustration is provided in Figure 1.

Suppose that all the points $\boldsymbol{X}$ on the plane $\boldsymbol{\pi}$ undergo a motion to $\boldsymbol{X}' = \boldsymbol{T}\boldsymbol{X}$ where $\boldsymbol{T} \in SE(3)$. Then the plane will move to $\boldsymbol{\pi}' = \boldsymbol{T}^{-\mathrm{T}}\boldsymbol{\pi}$. This is seen from $(\boldsymbol{\pi}')^{\mathrm{T}}\boldsymbol{X}' = \boldsymbol{\pi}^{\mathrm{T}}\boldsymbol{T}^{-1}\boldsymbol{T}\boldsymbol{X} = \boldsymbol{\pi}^{\mathrm{T}}\boldsymbol{X} = 0$, which shows that $\boldsymbol{X}'$ is on $\boldsymbol{\pi}'$.

2.5 Camera model

Consider a point with position given by the homogeneous vector $\boldsymbol{X}_o \in \mathbb{P}^3$ in the object frame o. The position of this point in the camera frame c is given by the homogeneous vector $\boldsymbol{X}_c = [X_1, X_2, X_3, X_4] \in \mathbb{P}^3$ where $\boldsymbol{X}_c = \boldsymbol{T}\boldsymbol{X}_o$ and

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \in SE(3) \tag{14}$$

is the homogeneous transformation matrix from the camera frame c to the object frame o. This point is imaged to the homogeneous point $\boldsymbol{s} = [s_x, s_y, 1]^{\mathrm{T}}$ in the normalized image plane, where $s_x = X_1/X_3$ and $s_y = X_2/X_3$, and to the homogeneous point $\boldsymbol{x} = \boldsymbol{K}\boldsymbol{s}$ in pixel coordinates, where $\boldsymbol{x} = [u, v, 1]^{\mathrm{T}}$ and $\boldsymbol{K}$ is the camera parameter matrix. The camera model can then be written

$$\lambda\boldsymbol{x} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X} \tag{15}$$

where $\lambda = X_3$ is the depth coordinate.

2.6 Vanishing points

Suppose that a point in the camera frame is given by $\boldsymbol{X}_c = [\boldsymbol{a}^{\mathrm{T}}, 0]^{\mathrm{T}}$ where $\boldsymbol{a} = [a_1, a_2, a_3]^{\mathrm{T}}$ is a unit vector. Then $\boldsymbol{X}_c$ is a point at infinity in the 3D scene. This point at infinity is mapped to normalized image coordinates as the point

$$\lambda\boldsymbol{s} = \boldsymbol{a} \tag{16}$$

If $a_3 \neq 0$, then $\boldsymbol{s}$ is a finite point which is called a vanishing point. In this case the vanishing point $\boldsymbol{s}$ in the image represents a point at infinity $\boldsymbol{X}_c$ in the scene. In the special case $a_3 = 0$, the point has zero depth, and the point $\boldsymbol{s}$ will be a point at infinity.

2.7 Parallel lines in the scene

Consider a line in the 3D scene through the two points $\boldsymbol{X} = [\boldsymbol{x}^{\mathrm{T}}, x_4]^{\mathrm{T}}$ and $\boldsymbol{Y} = [\boldsymbol{y}^{\mathrm{T}}, y_4]^{\mathrm{T}}$ where $\boldsymbol{x} = [x_1, x_2, x_3]^{\mathrm{T}}$ and $\boldsymbol{y} = [y_1, y_2, y_3]^{\mathrm{T}}$. The line is given in Plücker coordinates as

$$L = (x_4\boldsymbol{y} - y_4\boldsymbol{x}, \boldsymbol{x}^{\times}\boldsymbol{y}) \tag{17}$$

where $\boldsymbol{y} - \boldsymbol{x}$ is the direction vector and $\boldsymbol{x}^{\times}\boldsymbol{y}$ is the moment. To find the representation of this line in the normalized image plane it is noted that $\boldsymbol{X}$ is imaged to $\boldsymbol{s}_x = (1/x_3)\boldsymbol{x}$ and $\boldsymbol{Y}$ is imaged to $\boldsymbol{s}_y = (1/y_3)\boldsymbol{y}$. The resulting line $\boldsymbol{\ell}$ in the image plane is through the points $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$, which gives the homogeneous line

$$\boldsymbol{\ell} = \boldsymbol{s}_x^{\times}\boldsymbol{s}_y = (1/(x_3y_3))\boldsymbol{x}^{\times}\boldsymbol{y} \tag{18}$$

in the normalized image plane, which can be scaled to $\boldsymbol{\ell} = \boldsymbol{x}^{\times}\boldsymbol{y}$, which is the moment of the 3D line L.

Next, suppose that the 3D line is through the point $\boldsymbol{X} = [\boldsymbol{x}^{\mathrm{T}}, 1]^{\mathrm{T}}$ and the point $[\boldsymbol{a}, 0]^{\mathrm{T}}$ at infinity. Then the line is

$$L = (\boldsymbol{a}, \boldsymbol{x}^{\times}\boldsymbol{a}) \tag{19}$$

where the direction vector is given by the point at infinity. It is noted that all lines that intersect at the same point at infinity will be parallel and vice versa. This line is imaged to the homogeneous line

$$\boldsymbol{\ell} = \boldsymbol{x}^{\times}\boldsymbol{a} \tag{20}$$

in the normalized image plane. It is recalled that the point $[\boldsymbol{a}, 0]^{\mathrm{T}}$ at infinity is imaged to the vanishing point $\boldsymbol{s} = \boldsymbol{a}$. This means that if there are n parallel lines L_i in the 3D scene, where L_i passes through the point $\boldsymbol{x}_i$ and the point $[\boldsymbol{a}^{\mathrm{T}}, 0]^{\mathrm{T}}$ at infinity, then the corresponding lines $\boldsymbol{\ell}_i = \boldsymbol{x}_i^{\times}\boldsymbol{a}$ in the normalized image plane will intersect at the vanishing point $\boldsymbol{s} = \boldsymbol{a}$.

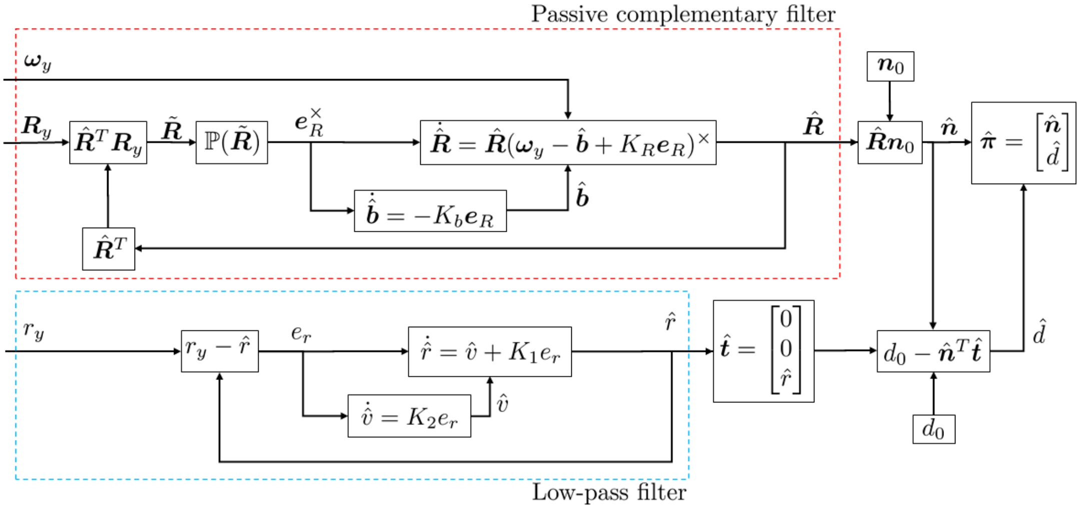
Passive complementary filter
$\boldsymbol{\omega}_y$
$\boldsymbol{R}_y$
$\hat{\boldsymbol{R}}^T\boldsymbol{R}_y$
$\tilde{\boldsymbol{R}}$
$\mathbb{P}(\tilde{\boldsymbol{R}})$
$\boldsymbol{e}_R^\times$
$\dot{\hat{\boldsymbol{R}}} = \hat{\boldsymbol{R}}(\boldsymbol{\omega}_y - \hat{\boldsymbol{b}} + K_R\boldsymbol{e}_R)^\times$
$\hat{\boldsymbol{R}}$
$\dot{\hat{\boldsymbol{b}}} = -K_b\boldsymbol{e}_R$
$\hat{\boldsymbol{b}}$
$\hat{\boldsymbol{R}}^T$
$\boldsymbol{n}_0$
$\hat{\boldsymbol{R}}\boldsymbol{n}_0$
$\hat{\boldsymbol{n}}$
$\hat{\boldsymbol{\pi}} = \begin{bmatrix}\hat{\boldsymbol{n}}\\ \hat{d}\end{bmatrix}$
r_y
$r_y - \hat{r}$
e_r
$\dot{\hat{r}} = \hat{v} + K_1 e_r$
$\hat{r}$
$\dot{\hat{v}} = K_2 e_r$
$\hat{v}$
$\hat{\boldsymbol{t}} = \begin{bmatrix}0\\0\\\hat{r}\end{bmatrix}$
$d_0 - \hat{\boldsymbol{n}}^T\hat{\boldsymbol{t}}$
$\hat{d}$
d_0
Low-pass filter

Figure 2: Block diagram of the passive complementary filter (in red) for the attitude estimation with bias correction, combined with the low pass filter (in blue) on the lidar measurements which provide the estimated plane.

3 Complementary Filter and Low-Pass Filter for Plane Estimation

3.1 Estimation of a Plane

Let a plane $\boldsymbol{\pi}$ at time instance t_k be described as

$$\boldsymbol{\pi}_k = \begin{bmatrix}\boldsymbol{n}_k\\ d_k\end{bmatrix} \in \mathbb{R}^4, \quad \|\boldsymbol{n}_k\| = 1 \tag{21}$$

where $\boldsymbol{n}_k \in \mathbb{R}^3$ is a unit surface normal of the observed plane, and $d_k \in \mathbb{R}$ is the shortest distance between the origin of the selected inertial frame and the plane.

We will assume that a plane evolves over time on the form

$$\boldsymbol{\pi}_k = \boldsymbol{T}_k^{-\mathrm{T}}\boldsymbol{\pi}_0 \tag{22}$$

where $\boldsymbol{\pi}_0 = [\boldsymbol{n}_0^{\mathrm{T}},\ d_0]^{\mathrm{T}}$ denote a plane initiated at time instance t_0. It is noted that unless anything else is specified, the surface vector of the initialized plane is set to $\boldsymbol{n}_0 = [0, 0, 1]^{\mathrm{T}}$, and the scalar term is set to $d_0 = 0$. However, $\boldsymbol{n}_0$ and d_0 can be arbitrarily chosen by the practitioner. The matrix

$$\boldsymbol{T}_k = \begin{bmatrix}\boldsymbol{R}_k & \boldsymbol{t}_k\\ \boldsymbol{0}^{\mathrm{T}} & 1\end{bmatrix} \in SE(3) \tag{23}$$

is the homogeneous transformation matrix at t_k, and $\boldsymbol{T}_k^{-\mathrm{T}}$ is the corresponding form for plane transformations (Hartley and Zisserman, 2004, p. 68). The plane $\boldsymbol{\pi}_k$ is then given on the form

$$\boldsymbol{\pi}_k = \boldsymbol{T}_k^{-\mathrm{T}}\boldsymbol{\pi}_0 = \begin{bmatrix}\boldsymbol{R}_k\boldsymbol{n}_0\\ d_0 - \boldsymbol{t}_k^{\mathrm{T}}\boldsymbol{R}_k\boldsymbol{n}_0\end{bmatrix} \tag{24}$$

where $\boldsymbol{t}_k$ can be any arbitrary point on the observed plane. The estimated plane is written in the form

$$\hat{\boldsymbol{\pi}}_k = \begin{bmatrix}\hat{\boldsymbol{n}}_k\\ \hat{d}_k\end{bmatrix} \tag{25}$$

where $\hat{\boldsymbol{n}}_k = \hat{\boldsymbol{R}}_k\boldsymbol{n}_0$ and $\hat{d}_k = d_0 - \boldsymbol{n}_0^{\mathrm{T}}\hat{\boldsymbol{R}}_k^{\mathrm{T}}\hat{\boldsymbol{t}}_k$. The plane $\boldsymbol{\pi}$ could be used as a state vector, but we choose to define a state element

$$\boldsymbol{x} = \{\boldsymbol{R}, d\} \tag{26}$$

which is an element of $SO(3)$ and a scalar $\mathbb{R}$. This allows us to employ the dynamics given in (7), and the complementary filter on $SO(3)$ can be used directly.

3.2 System Model

The rotation matrix to be estimated describes the relative attitude between the camera frame c and a coordinate frame fixed on the observed ship deck, s, and is denoted $\boldsymbol{R}_s^c$. The corresponding dynamics is, as in (7), given by the following kinematic differential equation

$$\dot{\boldsymbol{R}}_s^c = \boldsymbol{R}_s^c[\boldsymbol{\omega}_{sc}^s]^\times \tag{27}$$

where $\boldsymbol{\omega}_{sc}^s$ describes the relative angular velocity between s and c, given in s. The relative angular velocity between frame s and c, being expressed in s, is given as

$$\begin{aligned}\boldsymbol{\omega}^s_{sc} =&\boldsymbol{\omega}^s_{is} - \boldsymbol{\omega}^s_{ic} \\ =&\boldsymbol{\omega}^s_{is} - (\boldsymbol{R}^c_s)^{\mathrm{T}}\boldsymbol{\omega}^c_{ic}\end{aligned} \tag{28}$$

where i denotes the inertial frame.

Let $\boldsymbol{t}(t) = [0,\ 0,\ r(t)]$, where r denotes the true distance between the origin of c and the observed plane, along the z-axis of c. We will assume that the dynamics of r is random walk, such that $\dot{r}(t) = v(t)$ where $v(t)$ is unknown.

3.3 Measurement Models

The sensor configuration consists of a camera for measuring relative attitude, and a lidar to measure the distance between its sensor frame and the observed ship deck. Moreover, two IMUs are mounted on the sensor bracket and the observed ship.

Camera Measurements

The vision algorithm measures structural lines on the observed ship, which in turn will be used to construct a rotation matrix $\boldsymbol{R} \in SO(3)$ which describes the relative orientation between the camera frame c, which for brevity coincides with a, and the observed plane which will be denoted $\boldsymbol{\pi}$. The measured attitude $\boldsymbol{R}_y$ is assumed to be on the form

$$\boldsymbol{R}_y =\boldsymbol{R}\tilde{\boldsymbol{R}} \in SO(3) \tag{29}$$

where $\tilde{\boldsymbol{R}} = \exp(\boldsymbol{\eta}^{\wedge}_R)$ is the left-invariant measurement error of the attitude, $\boldsymbol{\eta}_R \in \mathbb{R}^3$ denotes white Gaussian measurement noise, and $\boldsymbol{R}$ is the true attitude.

Lidar Measurements

The lidar measures a distance r between the lidar fixed frame L and a point on $\boldsymbol{\pi}$. We assume that the measured value can be written as the sum

$$r_y = r + \eta_r \tag{30}$$

where r is the true distance, and η_r is a noise term. If the z axis of the lidar-fixed frame is parallel with the sensor direction, then we can represent the measurement as the vector $\boldsymbol{t}_y$ is given on the form

$$\boldsymbol{t}_y = (0,\ 0,\ r)^{\mathrm{T}} + (0,\ 0,\ \eta_r)^{\mathrm{T}} \tag{31}$$

Gyroscope Model

The IMUs are used to measure the angular rate of A and B, and when combined with the relative attitude $\boldsymbol{R}^A_B$, we can measure the relative angular velocity between A and B which is important for controlling purposes. The measured angular velocity $\boldsymbol{\omega}_y$ data will, as in (Kok et al., 2017), be assumed to be on the form

$$\boldsymbol{\omega}_y =\boldsymbol{\omega} + \boldsymbol{b}_{\omega} + \boldsymbol{\eta}_{\boldsymbol{\omega}} \in \mathbb{R}^3 \tag{32}$$

where $\boldsymbol{\omega}$ is the true angular velocity, $\boldsymbol{b}_{\omega} \in \mathbb{R}^3$ is a slowly time-varying bias term, and $\boldsymbol{\eta}_{\boldsymbol{\omega}}$ is the white, Gaussian noise.

3.4 Filter Design

The estimator is designed such that an attitude estimator and a low pass filter run independently, and their respective outputs are combined such that a full plane description can be provided, as illustrated in Fig. 2. The attitude estimator is the complementary filter formulated on $SO(3)$ (Mahony et al., 2008), while the low pass filter is simply an observer based on negative feedback with an observer gain.

Complementary Filter on SO(3)

The kinematics of the complementary filter are assumed to be given on the form

$$\dot{\hat{\boldsymbol{R}}} =\hat{\boldsymbol{R}}\left(\boldsymbol{\omega}_y - \hat{\boldsymbol{b}} + K_R\boldsymbol{e}_R\right)^{\times} \tag{33}$$

$$\dot{\hat{\boldsymbol{b}}} =- K_b\boldsymbol{e}_R \tag{34}$$

where $\boldsymbol{e}^{\times}_R = \frac{1}{2}\left(\tilde{\boldsymbol{R}} - \tilde{\boldsymbol{R}}^{\mathrm{T}}\right)$, and the estimated angular velocity is, based on the deterministic components of eq. (32), related to the measurements and the bias term as

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_y - \hat{\boldsymbol{b}} \tag{35}$$

The matrix $\tilde{\boldsymbol{R}}$ represents the left-invariant error rotation introduced in eq. (29), which in practice is found as the error between the measurement and the estimate, i.e. $\tilde{\boldsymbol{R}} = \hat{\boldsymbol{R}}^{\mathrm{T}}\boldsymbol{R}_y$. It is noted in systems where IMUs are not accessible, $\boldsymbol{\omega}_y = \boldsymbol{0}$, and $\dot{\hat{\boldsymbol{b}}} = \boldsymbol{0}$, which gives $\dot{\hat{\boldsymbol{R}}} = \hat{\boldsymbol{R}}(K_R\boldsymbol{e}_R)^{\times}$.

The filter configuration is illustrated in terms of a block diagram in Figure 2. The discretized form of eq. (33) and (34) gives

$$\hat{\boldsymbol{R}}_k =\hat{\boldsymbol{R}}_{k-1}\exp\left(h(\boldsymbol{\omega}_{y,k} - \hat{\boldsymbol{b}}_{\omega,k} + K_R\boldsymbol{e}^{\times}_{R,k})\right) \tag{36}$$

$$\hat{\boldsymbol{b}}_{\omega,k} =\hat{\boldsymbol{b}}_{\omega,k-1} - hK_b\boldsymbol{e}_{R,k} \tag{37}$$

where h is the time step.

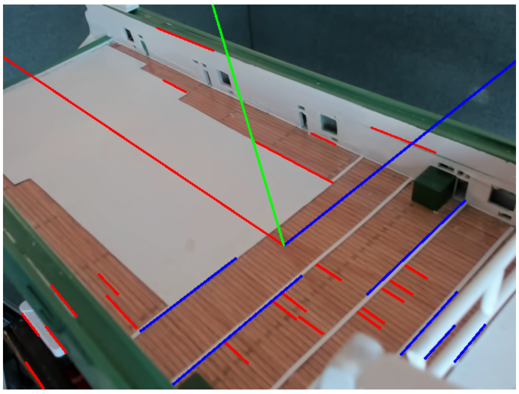

Figure 3: An illustration of clustered lines in the image. Lines of same color are assumed parallel, and the rotation from camera down to the ship deck is illustrated in the form of an axis.

Low Pass Filter for Lidar Measurements

Let r_k be the true distance at t_k, let $\dot{r}_k$ be the rate of change of the true distance, and let $\hat{r}_k$ and $\hat{v}_k = \dot{\hat{r}}_k$ represent the estimated distance and estimated rate of change, respectively. The discretized solution of the random walk model is $\hat{r}_{k+1} = \hat{r}_k + h\hat{v}_k$ over the time step $h = t_{k+1} - t_k$. Moreover, consider the error term $e_k = r_{y,k} - \hat{r}_k$, where $r_{y,k} = r_k + \eta_k$ is the measurement model in eq. (30), and η_r is the noise term. The filter can then be formulated as

$$\hat{r}_{k+1} = \hat{r}_k + h(\hat{v}_k + K_1 e_k) \tag{38}$$

$$\hat{v}_{k+1} = \hat{v}_k + hK_2 e_k \tag{39}$$

where K_1 and K_2 are positive, user defined parameters. It is noted that as the estimated velocity term $\hat{v}$ is driven by the error term e and not by a system model or any sensor measurements directly, it is actually a bias term.

The output $\hat{r}_k$ is used to define the estimated distance vector $\hat{\boldsymbol{t}}_k = [0,\ 0,\ \hat{r}_k]$ as in eq. (31), which in combination with the output from the attitude filter, $\hat{\boldsymbol{n}}_k = \hat{\boldsymbol{R}}_k \boldsymbol{n}_0$, the estimated plane distance is computed as $\hat{d}_k = -\hat{\boldsymbol{n}}_k^{\mathrm{T}} \hat{\boldsymbol{t}}_k$.

4 Vanishing point detection

The vanishing point detection method used in this paper is based on J-linkage method in (Tardif). The method is extended in the present paper for real-time performance with the following features. Firstly, we merge potential line segments from the line segment detector in a preprocessing stage. Secondly, instead of the Jaccard distance used in (Tardif) for clustering we use a different similarity measure which is more suited for the small line clusters, as the Jaccard distance gets very small for small clusters, which typically occurs in our experiments. Thirdly, we use a distance function between lines based on angle instead of translation.

Parallel lines in the 3D scene will intersect at a point at infinity. These parallel lines are mapped to the image plane as lines that intersect at the vanishing point corresponding to the point at infinity in the 3D scene. In a man-made 3D scene, it can be expected that the majority of the lines will belong to sets of parallel lines. These sets of parallel lines can be found by detecting the corresponding lines in the image, and then check if these lines intersect at a vanishing point. Lines in the image are found with a line segment detector, and then candidates for the vanishing points can be found as intersection points for the lines that are identified in the image.

The Line Segment Detector will give a set of M line segments where line segment i the two endpoints $\boldsymbol{e}_{i1}$ and $\boldsymbol{e}_{i2}$. The resulting line is then

$$\boldsymbol{\ell}_i = \boldsymbol{e}_{i1}^{\times} \boldsymbol{e}_{i2} \tag{40}$$

The midpoint of the line segment $\boldsymbol{\ell}_i$ is at $\boldsymbol{e}_{ic} = (1/2)(\boldsymbol{e}_{i1} + \boldsymbol{e}_{i2})$. A condition for the two lines $\boldsymbol{\ell}_i$ and $\boldsymbol{\ell}_j$ to coincide is that the midpoint $\boldsymbol{e}_{ic}$ of line $\boldsymbol{\ell}_i$ is on line $\boldsymbol{\ell}_j$, and that the angle between the two lines is zero. This is checked with the approximate conditions

$$\frac{\boldsymbol{\ell}_j^{\mathrm{T}} \boldsymbol{e}_{ic}}{\|\boldsymbol{n}_j\|} \leq \varepsilon \quad \text{and} \quad d(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j) \leq \psi \tag{41}$$

where $\boldsymbol{\ell} = [\boldsymbol{n}_j^{\mathrm{T}}, c_j]^{\mathrm{T}}$ and $d(\boldsymbol{\ell}_i, \boldsymbol{\ell}_j)$ is the distance function defined in (13), which equals the tangent of the angle between the lines. The threshold values are set to $\varepsilon = 4$ and $\psi = 0.01$, which is the maximum pixel offset. If the two lines are considered to coincide, then the two lines are merged, which will decrease the number of lines that needs to be evaluated from M to $K \leq M$.

4.1 Preference matrix

From the set of K lines, a set of N hypothetical vanishing points $\boldsymbol{v}_n$, $n = 1, \ldots, N$ are generated from

$$\boldsymbol{v}_n = \boldsymbol{\ell}_{i_n}^{\times} \boldsymbol{\ell}_{j_n} \tag{42}$$

where $i_n, j_n \in \{1, \ldots, K\}$ are randomly chosen indices for $n = 1, \ldots, N$.

It is then checked which of the remaining lines L_l, that pass through the hypothetical vanishing point $\boldsymbol{v}_n$. This is done by checking if the test line

$$\boldsymbol{\ell}_{ln} = \boldsymbol{e}_{lc}^{\times} \boldsymbol{v}_n \tag{43}$$

from the midpoint of line $\boldsymbol{\ell}_l$ to the vanishing point $\boldsymbol{v}_n$ is coincident with $\boldsymbol{\ell}_l$. This is checked with the condition

$$d(\boldsymbol{\ell}_{ln}, \boldsymbol{\ell}_l) \leq 0.05 \tag{44}$$

If this condition is satisfied, then $\boldsymbol{\ell}_l$ is considered to pass through $\boldsymbol{v}_n$.

This is encoded in a $K \times N$ preference matrix $\boldsymbol{P} = \{p_{ln}\}$ where $p_{ln} = 1$ if $\boldsymbol{\ell}_l$ passes through $\boldsymbol{v}_n$, and zero otherwise. This means that row l corresponds to line $\boldsymbol{\ell}_l$, and column n corresponds to vanishing point $\boldsymbol{v}_n$.

4.2 Clustering

It is assumed that all lines passing through the vanishing point $\boldsymbol{v}_n$ correspond to parallel lines in the scene. These lines are therefore grouped in the same set K_n, with elements encoded by column n in $\boldsymbol{P}$. Each row $\boldsymbol{P}_l$ in the preference matrix $\boldsymbol{P}$ encodes the vanishing points that are intersected by the line $\boldsymbol{\ell}_l$. If the rows $\boldsymbol{P}_{l_1}$ and $\boldsymbol{P}_{l_2}$ of $\boldsymbol{P}$ are sufficiently similar, then the corresponding two lines $\boldsymbol{\ell}_{l_1}$ and $\boldsymbol{\ell}_{l2}$ are considered to be coincident. The degree similarity between two rows was described with the Jaccard distance

$$d_J(\boldsymbol{P}_{l_1}, \boldsymbol{P}_{l_2}) = \frac{|\boldsymbol{P}_{l_1} \cup \boldsymbol{P}_{l_2}| - |\boldsymbol{P}_{l_1} \cap \boldsymbol{P}_{l_2}|}{|\boldsymbol{P}_{l_1} \cup \boldsymbol{P}_{l_2}|} \tag{45}$$

in (Tardif). Here $|\boldsymbol{P}_{l_1} \cup \boldsymbol{P}_{l_2}|$ denotes the number of columns where the rows $\boldsymbol{P}_{l_1}$ or $\boldsymbol{P}_{l_1}$ (or both) has a one, while $|\boldsymbol{P}_{l_1} \cap \boldsymbol{P}_{l_2}|$ is the number of rows where both rows $\boldsymbol{P}_{l_1}$ or $\boldsymbol{P}_{l_1}$ has a one.

Instead we used the similarity measure

$$d_S(\boldsymbol{P}_{l_1}, \boldsymbol{P}_{l_2}) = \frac{|\boldsymbol{P}_{l_1} \cap \boldsymbol{P}_{l_2}|}{|\boldsymbol{P}_{l_1} \oplus \boldsymbol{P}_{l_2}| + |\boldsymbol{P}_{l_1} \cap \boldsymbol{P}_{l_2}|} \tag{46}$$

where $|\boldsymbol{P}_{l_1} \oplus \boldsymbol{P}_{l_1}|$ is the number of rows where $\boldsymbol{P}_{l_1}$ or $\boldsymbol{P}_{l_1}$ (but not both) has a one. By using the exclusive disjunction $\boldsymbol{P}_{l_1} \oplus \boldsymbol{P}_{l_1}$ the system seems to be more sensitive to smaller clusters as the non-relevant elements in the preference sets are not considered when calculating the similarities.

The similarity measure is calculated for all combinations of two rows in $\boldsymbol{P}$, the two rows with the highest similarity measure are merged into one row where an element is zero where both rows has zeros, one where both rows have ones, and one where one row has zero and the other has one. An updated $\boldsymbol{P}$ matrix is stored where the two rows are replaced with the new row, and the two lines are put in the same cluster of lines. This is repeated with the updated matrix until the highest similarity measure between two lines is below 0.3, and the final result is stored in $\boldsymbol{P}_f$. Then each row in $\boldsymbol{P}_f$ will represent a cluster of lines with a potential vanishing point. It is assumed in this paper and experiment that the two clusters which contain the most lines will correspond to the two vanishing points in the x and y directions in agreement with the Manhattan assumption. These two clusters are denoted K_x and K_y, where the lines in the cluster K_x correspond to lines in the 3D scenes that are parallel to the x axis, while the lines in K_y correspond to 3D lines that are parallel to the y axis.

4.3 Refining the vanishing points

Consider the sets of lines K_n, $n = 1, 2$ obtained through the clustering process. If the cluster K_n only consists of two lines the vanishing point is found as $\boldsymbol{v}_n = \boldsymbol{\ell}_i \times \boldsymbol{\ell}_j$. If the set K_n consists of more than two lines the vanishing point can be found by minimizing the angle distance function according to

$$\boldsymbol{v}_n^* = \arg\min_{\boldsymbol{v}_n} \sum_{\boldsymbol{\ell}_j \in K_n} d(\boldsymbol{\ell}_j, \boldsymbol{e}_{jc}^{\times} \boldsymbol{v}_n) \tag{47}$$

where $\boldsymbol{e}_{jc}^{\times} \boldsymbol{v}_n$ is the line going through the estimated vanishing point and the centre point $\boldsymbol{e}_{jc}$ of line segment $\boldsymbol{\ell}_j$.

4.4 Rotation estimation

A homogeneous point $\boldsymbol{X}$ in the 3D scene is mapped to the corresponding homogeneous pixel point $\boldsymbol{x}$ in the image plane as

$$\boldsymbol{x} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X} \tag{48}$$

The Manhattan assumption states that all surfaces in the man-made environment are aligned with three dominant directions, corresponding to the points at infinity which is defined as $\boldsymbol{X} = [1, 0, 0, 0]^{\mathrm{T}}$, $\boldsymbol{Y} = [0, 1, 0, 0]^{\mathrm{T}}$ and $\boldsymbol{Z} = [0, 0, 1, 0]^{\mathrm{T}}$. As the points are not finite the last element in the direction representation is zero. The points at infinity can then be mapped to the image plane such that

$$\lambda \boldsymbol{v}_x = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{X} \tag{49}$$

$$\lambda \boldsymbol{v}_y = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{Y} \tag{50}$$

$$\lambda \boldsymbol{v}_z = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]\boldsymbol{Z} \tag{51}$$

where λ is the depth coordinate. Then the columns of $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3]^{\mathrm{T}}$ can be found from

$$\boldsymbol{r}_1 = \frac{\boldsymbol{K}^{-1}\boldsymbol{v}_x}{\|\boldsymbol{K}^{-1}\boldsymbol{v}_x\|},\ \boldsymbol{r}_2 = \frac{\boldsymbol{K}^{-1}\boldsymbol{v}_y}{\|\boldsymbol{K}^{-1}\boldsymbol{v}_y\|},\ \boldsymbol{r}_3 = \frac{\boldsymbol{K}^{-1}\boldsymbol{v}_z}{\|\boldsymbol{K}^{-1}\boldsymbol{v}_z\|} \tag{52}$$

However, as all of the above vectors are orthogonal only two vanishing points are needed, and the last column of the rotation matrix can be found as

$$\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2 \tag{53}$$

The computed rotation matrix $\boldsymbol{R}$ may not be orthogonal due to measurement inaccuracies. Then the closest orthogonal matrix $\boldsymbol{R}_o$ is found from the orthogonal Procrustes problem (Golub and Van Loan, 1996)

$$\min_{\boldsymbol{R}_o \in SO(3)} \|\boldsymbol{R}_o - \boldsymbol{R}\|_F^2 \tag{54}$$

which is equivalent to maximizing the trace of $\boldsymbol{R}_o\boldsymbol{R}^{\mathrm{T}}$. The solution is (Umeyama, 1991)

$$\boldsymbol{R}_o = \boldsymbol{U}\mathrm{diag}(1, 1, \det(\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}))\boldsymbol{V}^{\mathrm{T}} \tag{55}$$

where $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$ is the SVD of $\boldsymbol{R}$.

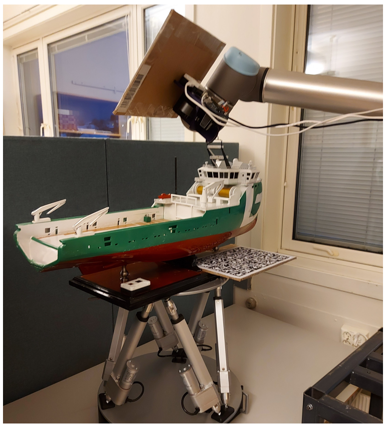

Figure 4: The experiment setup consisting of a Stewart platform with a ship and an IMU attached and a robotic arm with a camera, lidar and an IMU attached at the end effector.

5 Experiments and Results

This section presents experimental validation of the measurement systems, the vanishing point detection, and the performance of the total system with filtering. A small scale lab was made to test the vision algorithm along with the gyroscopes and the lidar. As seen in Figure 4, a ship model was mounted on the motion base of a Stewart platform, along with a gyroscope. The platform was controlled by a computer where an input signal based on a sum of sine and cosine functions was generated to simulate wave motions. Moreover, a robot arm was used to simulate the crane motion from another ship. A sensor package consisting of a monocular camera, a lidar distance sensor and a gyroscope was attached to the end effector of the robot.

The ground truth data was provided by an ArUco tracking system (Garrido-Jurado et al., 2014). The ArUco system tracked a printed set of 35 fiducial markers arranged in a 5 by 7 grid to ensure high accuracy. The measured position and attitude was obtained by computing a mean position and rotation based on which markers being available in the image frame.

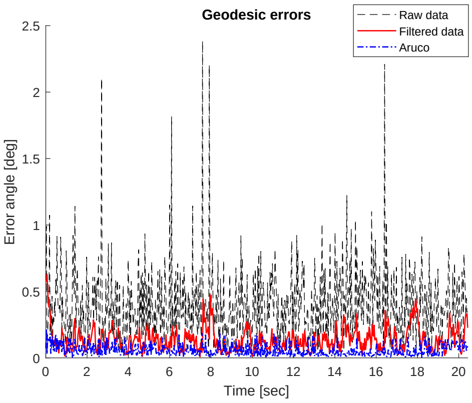

Figure 5: Angular distance with raw VPD data (dashed black curve), filtered VPD data (red curve) the found truth from the ArUco system (blue curve).

5.1 Validation of Measurements

In the first experiment the relative pose between the ship and the sensors was static. Attitude measurement were obtained with a camera running vanishing point detection (VPD) to identify the deck of the ship. The distance measurements were performed with a lidar. The measurements were filtered with the observer with the gains $K_R = 2.5 \cdot 10^{-2}$, $K_b = 2.5 \cdot 10^{-3}$, $K_1 = 0.1$, and $K_2 = 0.01$. In the experiment, the unfiltered raw measurements and the filtered measurements were compared to the ground truth from the ArUco system.

The uncertainty in the attitude measurements is described in terms of the standard deviation of the angular distance, which is computed from

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\theta_i^2 \tag{56}$$

Validation of Attitude Measurements

With reference to Figure 5 the ground truth from the ArUco system was very accurate with low noise. The results in Table 1 show that the raw data had a standard deviation of about 0.48° degrees in the angular distance, while the filtered data had a standard deviation of about 0.1655° in the angular distance using filter gain $K_R = 5$ and no bias correction. It is noted that a higher K_R gain gives faster system respone, but increased noise, while a lower gain gives slower response and reduced high frequency noise. The ground truth from the ArUco system had a standard deviation of 0.0727° in the angular distance.

Table 1: The standard deviations of each data set consisting of geodesic distances on $SO(3)$ are shown in degrees.

Data set	Raw data	Filtered data	ArUco
Std.dev.	0.4801	0.1655	0.0727

Table 2: Standard deviations of plane distances, given in centimeters. It is noted that the estimated distance was initiated by the first lidar and attitude measurements.

Data set	Raw data	Filtered data	ArUco
Std.dev.	1.7925	1.6694	0.0593

Validation of Lidar Measurements

The distance from camera sensor and to the plane of the ship deck was measured to be 40.2 cm. The raw data from the lidar had a standard deviation of 1.7925 cm. The filtered lidar measurement had a standard deviation of 1.6694 cm using the filter gains $K_R = 5$, $K_1 = 7$ and $K_2 = 2$. The ground truth from the AruCo system had a standard deviation of 0.0593 cm. The curves are shown in Figure 6, and the standard deviations are given in Table 2.

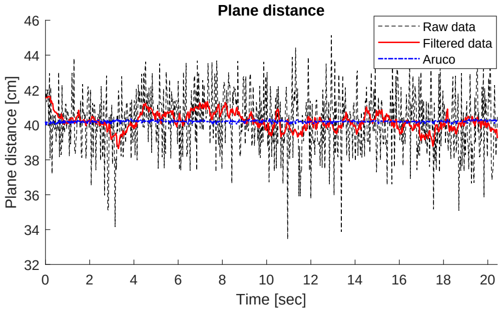

Figure 6: The black plot shows the plane distance obtained by the raw data from the lidar measurements and the VPD algorithm. The red plot shows the output from the filter shown in Fig. 2, while the blue plot is the distance based on the fiducial markers.

5.2 Vanishing point detection

In the second experiment it was investigated how the number N of hypothetical vanishing points influenced the the computational time and the accuracy of the vanishing point detection. In this investigation 4 different images were used where the deck of the ship was imaged from different angles. One of the 4 images is shown in Figure 7. The hypothetical vanishing points are generated from randomly selected line segments, so each image was processed 250 times and the average accuracy was found as the standard deviation in the angular distance as given by (56). In addition, the average computation time for the 4 images was found. The results are presented in Table 3. In (Tardif) 500 hypothetical vanishing point were used for vanishing point detection. The experimental results obtained here indicate the 200 is a better number for the hypothetical vanishing points. At this number the standard deviation was only slightly higher than with 500 or 1000 points, and the running time was significantly better. From our results it is seen that 200 points give a computational time of 25 ms, which allows for a frame rate up to 40 frames per second.

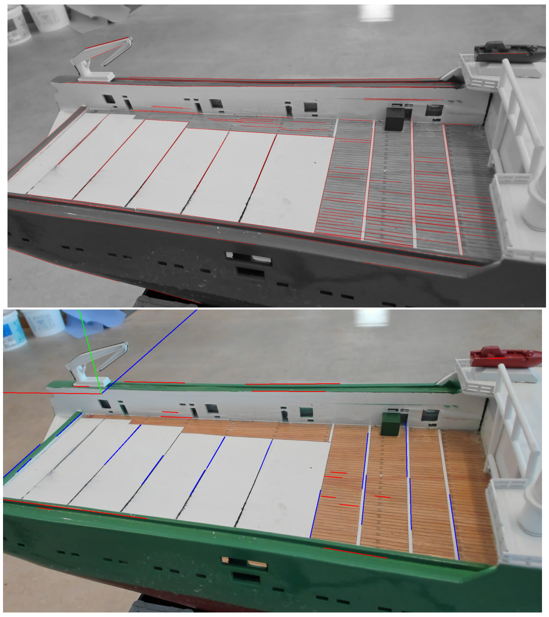

Figure 7: The top image shows the line segments which have been detected while the bottom image shows the vanishing points detection based on these line segments. The blue lines are considered to be parallel in the x direction, and the red lines are parallel in y direction.

The tests were conducted with an Intel Core i7 1.30 GHz computer with 32 GBs RAM on Ubuntu 21.04.

5.3 Case Studies

Three case studies were conducted on the experimental data. The ship was actuated with a Stewart platform which simulated a wave motion, and sensors were attached. The camera sampled line and marker data si-

Table 3: Standard deviation in degrees for each of the 4 test images as a function of the number of hypotheses for vanishing points. The average computation time in ms is shown in the last line.

N	50	100	200	500	1000
Image 1	0.6360	0.2406	0.1776	0.1203	0.09167
Image 2	2.481	2.286	1.719	1.432	1.381
Image 3	18.64	0.6474	0.4985	0.4870	0.4641
Image 4	11.60	0.8480	0.1891	0.1318	0.1031
Av. time	17	19	25	41	69

multaneously to build two comparable data sets. Moreover, the lidar measured the distance between the sensor bracket and the planar surface. One gyroscope was attached to the sensor package on the robot, and one gyroscope was attached to the ship deck.

Case 1

In Case 1, a time history was recorded for a predetermined ship motion over 60 s. Vanishing point detection and lidar data were used to track the ship deck. No gyroscopes were used in this experiment. The filter gains were $K_R = 10$, $K_b = 0.01$, $K_1 = 7$ and $K_2 = 2$. The unfiltered measurements and the filtered measurements were compared to the ground truth. The measured and estimated relative attitude between the ship deck and the camera frame in shown in Figure 8.

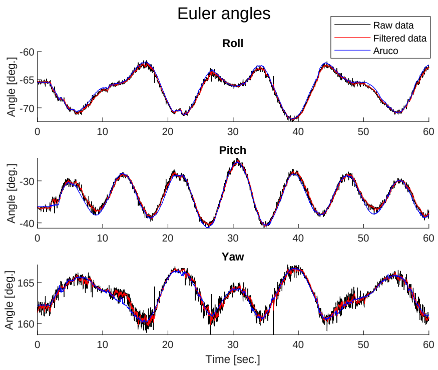

Figure 8: The plots show the measured, estimated and true Euler angles of the rotation matrices which represent the relative attitude between the sensor frame and the observed plane.

The raw measurements of the attitude had a standard deviation of 0.43° with a maximum error distance of 3.59° relative to the ground truth. The estimated attitude had a standard deviation of 0.38°, and a maximum angular erro of 2.14°. The time history of the error attitudes is shown in Figure 9.

The plane distances based on the unfiltered lidar and attitude data had a standard deviation of 1.72 cm, and the maximum error distance was 6.10 cm The estimated plane distances had a standard deviation of 0.93 cm, and a maximum error of 3.23 cm. The error distances over time is shown in Figure 10.

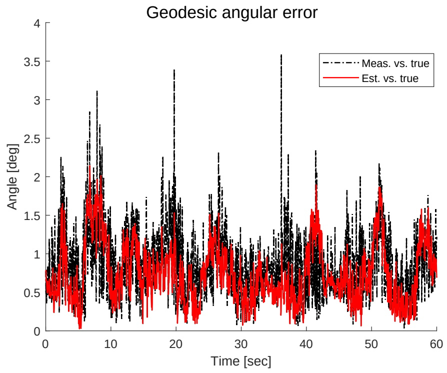

Figure 9: The black plot shows the measured geodesic distance between the measured rotation matrices and the true attitudes over time. The red plot shows error angles between the estimated attitudes, based on vision data only, and the true rotation matrices.

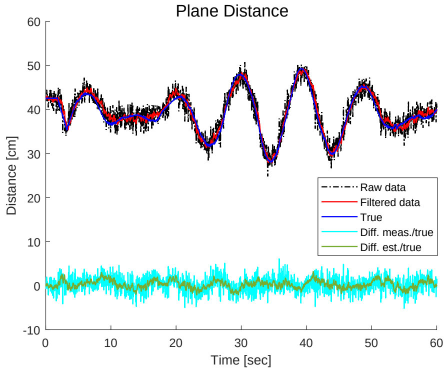

Figure 10: The plot shows the plane distances based on raw measurements, filtered data based on the scheme in Figure 2, and the true distance. Moreover, two error plots are added, namely the difference between the measured plane distances and the true values, and the estimated distances and the true values.

Case 2

In Case 2 the time history and the measurements of Case 1 was used, and in addition, measurements were included from one gyroscope at the robot sensor package and one gyroscope at the ship deck. The filter gains were $K_R = 6$, $K_\omega = 15$, $K_b = 0.01$, $K_1 = 7$ and

$K_2 = 2$. Figure 11 shows the geodesic angular errors of the measurements and the estimates in comparison with the ground truth.

The standard deviation of the estimated error attitude was 0.35° and the maximum angular error was 1.83°. The standard deviation of the estimated plane distance was 0.92 cm, with a maximum error of 3.05 cm. The estimated plane distances in Case 2 are shown in Figure 12.

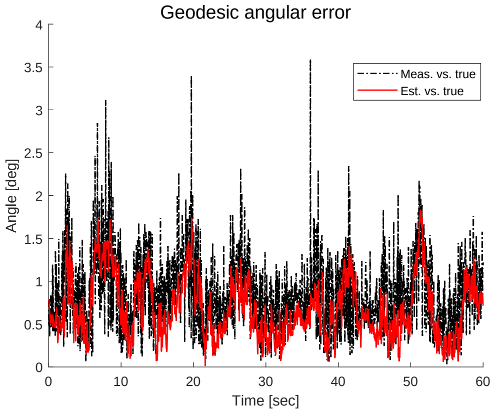

Figure 11: The black plot is the same as in Figure 9, while the red plot is the estimated error angle when the gyroscopes are used in addition to the camera data.

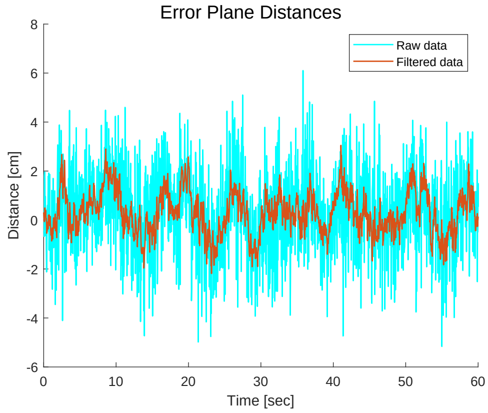

Figure 12: The difference between the estimated plane distances and the true values when gyroscopes are utilized is not significant. This is because K_1 and K_2 remained unchanged from the first case, and the gyroscope data only altered the surface normal.

Case 3

In Case 3, the sensor configuration was the same as the one in Case 2. However, in the time interval from 30 s to 40 s, only gyroscope and lidar measurements were available.

The absence of attitude correction made the estimate to drift from the ground truth. It was seen that the estimated roll had deviated 5.2° after 10 seconds. The estimated pitch and yaw parameters had deviated 6.6° and 3.1° respectively over the same time span, as shown in Figure 13. The corresponding geodesic distance was 8.6°, as shown in Figure 14.

The estimated plane distance was also affected by the loss of angular corrections. It was seen that the estimated plane distance had deviated 2.9 cm over the time span without camera measurements as shown in the time history of Figure 15. However, as the attitude measurements were available from 40 s, the plane distance and the angular offset was corrected.

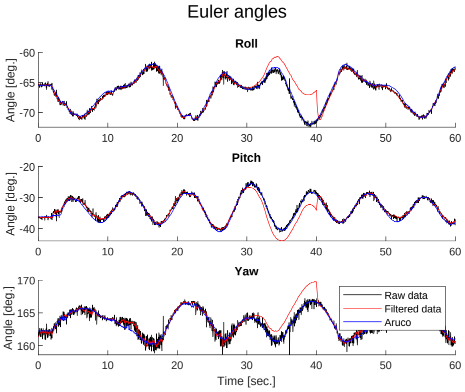

Figure 13: Estimated roll-pitch-yaw angles when the vanishing point detector is unavailable from 30 s to 40 s. It is seen that the angles drift off in the interval.

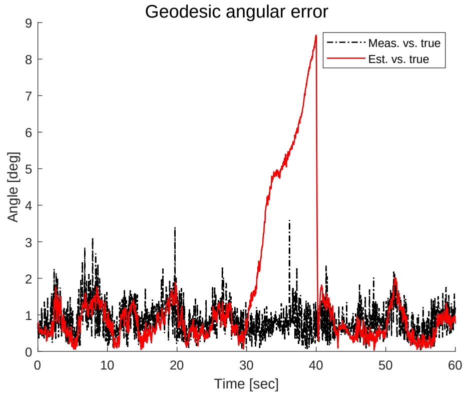

Figure 14: Estimated angular distance of the estimation error when the vanishing point detector is unavailable from 30 s to 40 s. It is seen that the angles drift off in the interval.

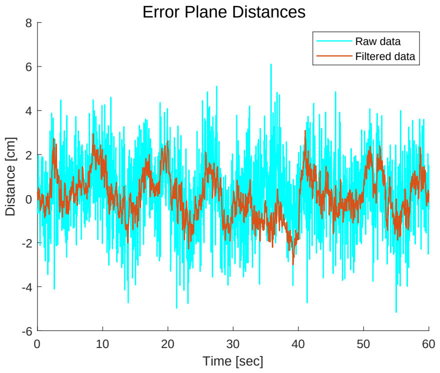

Figure 15: The plot illustrates how the estimated distance to the plane was affected by the loss of vanishing point detector.

6 Conclusion

We have in this paper presented a system for estimation of a ship deck using camera, lidar and gyroscopes. The camera has been used in a vision system that is based on the detection of lines as input to a vanishing point detector. This has been done under the Manhattan assumption where the majority of lines are along 3 orthogonal axes. Since the vision algorithm depends on lines rather than points,the system is more resistant to occlusions like vision algorithms based on point tracking. Two sets of parallel orthogonal lines are detected on the ship deck and used to construct an element of $SO(3)$ to describe the relative attitude, and we have validated the algorithm in experiments. We have shown that the algorithm has an average computational running time of 25 ms per frame, which allows for a frame rate up to 40 frames per second.

The vanishing point algorithm was used to estimate a ship deck using a complementary filter with input from lidar and gyroscopes. It was seen that the standard deviation in the angular distance of the VPD algorithm was 0.4801° in a static setup, and the measurements from the selected lidar sensor had a standard deviation of 1.7925 cm in the same case. Moreover, it was shown in both the static and dynamic cases that the observer was able to reduce the noise significantly.

Future work includes experiments performed on full scale systems in outdoor environments, as well as implementing the estimation technique in combination with a controlled crane, and using more accurate, industrial sensors.

Acknowledgments

The research presented in this paper was funded by the Norwegian Research Council under Project Number 237896, SFI Offshore Mechatronics.

References

Abdel-Rahman, E., Nayfeh, A., and Masoud, Z. Dynamics and control of cranes: A review. *Journal of Vibration and Control*, 2003. 9:863–908. doi:10.1177/1077546303009007007.

Barnard, S. T. Interpreting perspective images. *Artificial Intelligence*, 1983. 21(4):435–462. doi:10.1016/S0004-3702(83)80021-6.

Bouguet, J.-Y. *Visual methods for three-dimensional modeling*. Ph.D. thesis, California Institute of Technology, 1999. doi:10.7907/hc2c-sp47.

Caprile, B. and Torre, V. Using vanishing points for camera calibration. *International journal of computer vision*, 1990. 4(2):127–139. doi:10.1007/BF00127813.

Coughlan, J. M. and Yuille, A. L. Manhattan world: Orientation and outlier detection by bayesian inference. *Neural Computation*, 2003. 15(5):1063–1088. doi:10.1162/089976603765202668.

Garrido-Jurado, S., Muoz-Salinas, R., Madrid-Cuevas, F. J., and Marn-Jimnez, M. J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 2014. 47(6):2280–2292. doi:10.1016/j.patcog.2014.01.005.

Golub, G. H. and Van Loan, C. F. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

Hartley, R., Trumpf, J., Dai, Y., and Li, H. Rotation averaging. *International Journal of Computer Vision*, 2013. 103(3):267–305. doi:10.1007/s11263-012-0601-0.

Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. doi:10.1017/CBO9780511811685.

Kok, M., Hol, J., and Schn, T. Using inertial sensors for position and orientation estimation. *Foundation and Trends of Signal Processing*, 2017. 11. doi:10.1561/2000000094.

Lin, S., Garratt, M., and Lambert, A. Monocular vision-based real-time target recognition and tracking for autonomously landing an uav in a cluttered shipboard environment. *Autonomous Robots*, 2017. 41. doi:10.1007/s10514-016-9564-2.

Ma, Y., Soatto, S., Košecká, J., and Sastry, S. S. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003. doi:10.1017/S002555720017809X.

Magee, M. and Aggarwal, J. Determining vanishing points from perspective images. *Computer Vision, Graphics, and Image Processing*, 1984. 26(2):256–267. doi:10.1016/0734-189X(84)90188-9.

Magri, L. and Fusiello, A. T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 3954–3961, 2014. doi:10.1109/CVPR.2014.505.

Mahony, R., Hamel, T., and Pflimlin, J.-M. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 2008. 53(5):1203–1218. doi:10.1109/TAC.2008.923738.

Manton, J. A globally convergent numerical algorithm for computing the centre of mass on compact lie groups. volume 3. pages 2211 – 2216, 2005. doi:10.1109/ICARCV.2004.1469774.

Moakher, M. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 2002. 24. doi:10.1137/S0895479801383877.

Mueller, A. *Modern Robotics: Mechanics, Planning, and Control [Bookshelf]*, volume 39. 2019. doi:10.1109/MCS.2019.2937265.

Park, F. C. Distance metrics on the rigid-body motions with applications to mechanism design. *J. Mechanical Design*, 1995. 117(1):48–54. doi:10.1115/1.2826116.

Patruno, C., Nitti, M., Petitti, A., Stella, E., and D'Orazio, T. A vision-based approach for unmanned aerial vehicle landing. *Journal of Intelligent & Robotic Systems*, 2019. 95. doi:10.1007/s10846-018-0933-2.

Polvara, R., Sharma, S., Wan, J., Manning, A., and Sutton, R. Towards autonomous landing on a moving vessel through fiducial markers. 2017. doi:10.1109/ECMR.2017.8098671.

Ramli, L., Mohamed, Z., Abdullahi, A., Jaafar, H. I., and M. Lazim, I. Control strategies for crane systems: A comprehensive review. *Mechanical Systems and Signal Processing*, 2017. 95C:1–23. doi:10.1016/j.ymssp.2017.03.015.

Rauscher, F., Nann, S., and Sawodny, O. Motion control of an overhead crane using a wireless hook mounted IMU. In *Proceedings 2018 American Control Conference (ACC)*. IEEE, pages 5677–5682, 2018. doi:10.23919/ACC.2018.8431170.

Tardif, J.-P. Non-iterative approach for fast and accurate vanishing point detection. In *ICCV*. IEEE Computer Society, pages 1250–1257, 2019. doi:10.1109/ICCV.2009.5459328.

Toldo, R. and Fusiello, A. Robust multiple structures estimation with j-linkage. In *European conference on computer vision*. Springer, pages 537–547, 2008. doi:10.1007/978-3-540-88682-2_41.

Tørdal, S. S. and Hovland, G. Ship-to-Ship State Observer Using Sensor Fusion and the Extended Kalman Filter. *Journal of Offshore Mechanics and Arctic Engineering*, 2019. 141(4). doi:10.1115/1.4041643. 041603.

Tysse, G. O., Cibicik, A., and Egeland, O. Vision-based control of a knuckle boom crane with online cable length estimation. *IEEE Transactions on Mechatronics*, 2021. 26(1):416–426. doi:10.1109/TMECH.2020.3024637.

Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1991. 13(4):376–380. doi:10.1109/34.88573.

Xu, C., Zhang, L., Cheng, L., and Koch, R. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 39(6):1209–1222. doi:10.1109/TPAMI.2016.2582162.

Zitnick, C. L. and Dollár, P. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*. Springer, pages 391–405, 2014. doi:10.1007/978-3-319-10602-1_26.

Zuliani, M., Kenney, C., and Manjunath, B. The multiransac algorithm and its application to detect planar homographies. In *IEEE International Conference on Image Processing 2005*, volume 3. pages III–153, 2005. doi:10.1109/ICIP.2005.1530351.