THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING IN MACHINE AND VEHICLE SYSTEMS

Beyond-application datasets and automated fair benchmarking

KRISTER BLANCH

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2023

Beyond-application datasets and automated fair benchmarking
KRISTER BLANCH

© KRISTER BLANCH, 2023

Thesis for the degree of Licentiate of Engineering 2023:12
ISSN 1652-8565
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers digitaltryck
Göteborg, Sweden 2023

Beyond-application datasets and automated fair benchmarking

KRISTER BLANCH
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Beyond-application perception datasets are generalised datasets that emphasise the fundamental components of good machine perception data. When analysing the history of perception datatsets, notable trends suggest that design of the dataset typically aligns with an application goal. Instead of focusing on a specific application, beyond-application datasets instead look at capturing high-quality, high-volume data from a highly kinematic environment, for the purpose of aiding algorithm development and testing in general. Algorithm benchmarking is a cornerstone of autonomous systems development, and allows developers to demonstrate their results in a comparative manner. However, most benchmarking systems allow developers to use their own hardware or select favourable data. There is also little focus on run time performance and consistency, with benchmarking systems instead showcasing algorithm accuracy. By combining both beyond-application dataset generation and methods for fair benchmarking, there is also the dilemma of how to provide the dataset to developers for this benchmarking, as the result of a high-volume, high-quality dataset generation is a significant increase in dataset size when compared to traditional perception datasets.

This thesis presents the first results of attempting the creation of such a dataset. The dataset was built using a maritime platform, selected due to the highly dynamic environment presented on water. The design and initial testing of this platform is detailed, as well as as methods of sensor validation. Continuing, the thesis then presents a method of fair benchmarking, by utilising remote containerisation in a way that allows developers to present their software to the dataset, instead of having to first locally store a copy. To test this dataset and automatic online benchmarking, a number of reference algorithms were required for initial results. Three algorithms were built, using the data from three different sensors captured on the maritime platform. Each algorithm calculates vessel odometry, and the automatic benchmarking system was utilised to show the accuracy and run-time performance of these algorithms. It was found that the containerised approach alleviated data management concerns, prevented inflated accuracy results, and demonstrated precisely how computationally intensive each algorithm was.

Keywords: Beyond-application datasets, automatic fair benchmarking, algorithm evaluation, autonomous systems, vehicle odometry, containerisation

There is nothing—absolutely nothing—half so much worth doing as simply messing about in boats.

—Kenneth Grahame, The Wind in the Willows

Acknowledgements

To my wonderful wife, Talia, I thank you for ensuring my happiness and sanity.

To Ola, Christian, Björnborg, Ted, the Chalmers Revere vehicle laboratory, and to all those who have helped get and keep Seahorse afloat, I am truly grateful for the energy and effort you have given.

To my friends, family, and everyone else who has supported this undertaking, I appreciate you far more than I can ever express on a single acknowledgement page.

Thank you.

List of included papers

This thesis consists of the following papers:

Paper A A. Engström et al. "A lidar-only SLAM algorithm for marine vessels and autonomous surface vehicles". *Proceedings of the 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, (CAMS 2022).* 2022

Paper B B. Nguyen et al. "Application and evaluation of direct sparse visual odometry in marine vessels". *Proceedings of the 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, (CAMS 2022).* 2022

Paper C K. Blanch and O. Benderius. Topographic flow based odometry. *Submitted to the Journal of Intelligent & Robotic Systems* (2023)

Table of contents

Chapter 1

Introduction and motivation

Autonomous systems development is currently caught between the constant improvement of sensor technology and the need for fast algorithm execution for perception and navigation. Perception algorithms are the cornerstone of machines being able to understand the environment around them, and are therefore inherently linked to the safe operation of an autonomous platform. These algorithms are built and trained on datasets which are made by capturing the environment with a number of varying sensors, in order to replicate the world the machine will operate in.

The result is that, historically, sensor selection and dataset generation has erred towards application-centric approaches [29], which arguably prioritises algorithm development over the long term viability of the dataset. Whilst not inherently wrong, it is now possible to see the technical trends within these datasets, and the biases that they now present. In the case for autonomous road vehicles, this has manifested as two clear clusters of perception datasets, with the first in 2009 corresponding to the increased development and adoption for *advanced driver-assistance systems* (ADAS), and the following in 2016 corresponding to the accelerated race around *autonomous drive* (AD) [29].

Datasets are also limited by the environments where they are collected, which tends to be either from on-road scenarios using conventional passenger cars or from flying drones. In the case of land vehicles, they are driven through urban environments or on highways, with very limited and predictable kinematics. In the case of flying vehicles, they are limited by payload size and cost to operate. There are some exceptions, but as of the time of writing, the author has found that when requesting datasets falling outside of this generalisation, there is no response, or that they continue to be designed for a specific application.

Arguably, to fully test the accuracy and performance of robot perception, a state-of-the-art dataset should enable both: a kinematically demanding agent, as well as a sensor package providing high-volume and high-quality data of diverse scenarios. Marine vehicles would likely be the best agent to fulfil these requirements, as weather conditions directly impact the kinematics, and that they are typically large enough to support the payload required for a high quality sensor suite. In addition, marine environments also provide both sparse and dense visual and geometric sensor feedback, a large variety of discrete man-made and biological objects, and predictable behavioural patterns, which are all relevant for the typical types of perception algorithms. Uniquely, it also provides morphological geometry as given by the surrounding water, giving further opportunities for algorithms.

This method of building a dataset based on these merits is considered a *beyond-application* approach,

whereby the focus is less on the outcome and use-cases, and instead on these fundamentals. There are cases where a tailored dataset is still preferable, and for the majority of contemporary algorithm developers, working on a dataset within their domain would initially yield better results. However, when working on algorithms that are abstract or fundamental in nature, there is a theoretical benefit from working with a beyond-application dataset.

To prove this, there needs to be a method of *fair benchmarking* that is available to all who wish to use the dataset. Benchmarking has been a cornerstone of open source dataset development, and allows users to present their code for comparison against others for specific tasks. Whilst necessary for continual improvements to autonomous tasks, it has been shown that submissions to these benchmarks often overlook the importance of local hardware when developing [47]. Users take a non-systematic approach to evaluation, and as a result, have developed their systems to favour results over computational costs. To ensure the validity of a generalised dataset, it therefore needs to have a non-domain and non-application approach to benchmarking, whilst providing a method that ensures that every algorithm is subjected to the same environmental factors.

Due to the goal of high-volume data (i.e. larger than what is practical to download and use locally), a number of technical problems were established. Firstly, even with the payload limitations being relaxed, there is still a significant limitation of both power and space on maritime vessels. The highly kinematic nature of the marine environment also provides challenges, being that equipment is constantly subjected to changed acceleration forces. Migrating the data from a floating platform to an established and secure hosting facility requires the development of said secure housing, and a method of transport that is safe. And finally, with the implication that algorithms need to be evaluated close to the data, an important technical problem of this work was to find methods to evaluate algorithms towards the data as efficiently as possible.

1.1 Research problem and thesis outline

This thesis presents multiple sections of work. Chapter 2 presents contemporary dataset studies across multiple domains. Chapter 3 presents an alternative to these studies, being the implementation of a beyond-application approach for developing autonomous datasets. This has been done on a maritime sensor platform suitable for littoral waters, and in Chapter 4 the platform is presented, detailing the sensors, system, power and network architecture. It also discusses the novel solutions developed to overcome the problems associated with maritime development. Chapter 5 then presents the work on the validation, storage and distribution of a high-volume dataset, and specifically addresses the problems put forward of how to overcome data modernisation and still have comprehensive, high-fidelity data available for global use. Further on, the chapter presents the work and results of three papers that used this platform and dataset to design, develop and test algorithms for autonomous driving. This chapter continues by establishing methods for sensor and data validation across domains. Finally, this chapter demonstrates these algorithms in a novel system for automated performance benchmarking, outlining the best practice for algorithm comparisons.

From this, the research questions can be broken into the following research questions:

1. **Datasets:** Are beyond-application, high-volume, high-quality datasets suitable for autonomous systems development?

2. **Data collection:** Is the maritime domain suitable for capture of high-volume data, and how do we develop a practical system for this environment that can maintain data integrity?

3. **Deployment:** When working with high-volume data, what are the best practices for global distribution?

4. **Benchmarking:** What is the best way to ensure fair, accurate and reproducible benchmarking for algorithm development?

To conclude the thesis, Chapter 6 contains the author's discussion on the works concluded so far, as well as the future direction for the project.

Chapter 2

Related works

In 2012, the Kitti dataset was published and whilst not the first autonomous systems dataset, it is arguably the most impacting with applied systems benchmarking [13, 15]. The paper itself details the pipeline for users, being the collection, the annotation, and the storage of data, and then finally, a system where developers can submit the results of their algorithms for comparison. Specifically, the Kitti dataset allows users to develop sensor-fusion methodologies for autonomous systems, as it layered a multitude of sensors with time synchronisation and calibration protocols. In 2015, the Kitti authors demonstrated their comparative leaderboard, the methods used, and any papers published in relation to its submission [14].

Since this implementation, modern autonomous systems dataset generation has fallen into two categories, being the emulation of the Kitti dataset [34] or being tailored for a specific application. In the earlier days of autonomous systems this application was the development of ADAS algorithms, but since 2016 this shifted towards complete AD [58], of which a large number of perception datasets have built their collection methods for [16, 8, 39, 37]. The emphasis of these datasets has been to completely replace human drivers, and thus the datasets carry facets from a multitude of sensors designed specifically to replace human perception.

In the maritime sector, this level of development has not been as forthcoming, with focus on developing for skipper assistance instead of true autonomous drive. This is due to the rigidness of the international communities legislation regarding *maritime autonomous surface ships* (MASS). The current regulatory standards prohibit the deployment of true AD within the maritime framework [9], and thus the developmental focus has been on skipper assistance, similar to the 2009 trend of developing ADAS. Whilst the land based datasets are publishing comprehensive sensor suites, contemporary maritime datasets are limited by either size or sensor selection, which can be seen in a number of high profile contemporary datasets [59, 18, 31, 21, 72]. It has only been in the last few years where maritime datasets are beginning to approach the Kitti model [30], but these are typically focusing on integrating a single sensor into an established sensor system. Recently, there has also been a trend of utilising external data networks, with both GNSS and the maritime *automatic identification system* (AIS) being able to provide navigational data to a vessel by receiving information provided by other parties. However, similar to the other contemporary marine datasets, the emphasis is on the integration of this system into an already established platform [63, 66].

There are also datasets created using *unmanned aerial drones* (UAVs) for similar purposes, such as UAV123 [44] and MOR-UAV [38]. For the purposes of developing perception algorithms, these datasets provide annotations and benchmarking utilities specific for object detection and tracking. These datasets are again tailored specifically for a single application, and whilst similar to Kitti in the approach to

dissemination, lack a diverse sensor suite. This is quite common within UAV datasets, again due to legislative concerns [42], but also due to the excessive cost of payload size and operation [65].

Looking at these three domains in parallel, there are two main tracks to the research being done on cross-domain deployability. The first track looks at robots that operate across multiple domains [71]. However, these often have multiple independent algorithmic suites which are deployed as the vehicle enters each domain. There has been some recent works that look at the transferability of algorithms across domains [41, 53], with focus on the conceptual states that are generalised across robotics.

Regardless of domain, there has been a significant gap in providing the user with accurate timestamping methods. In 2003, an overview was presented detailing how best to approach data synchronicity for sensor-fusion [28], however an analysis in 2020 showed that system designers are often left to build their own synchronisation methods, which are often error prone [70]. For the application-centric datasets that were designed with a specific goal in mind, this is quite often the norm. However, this does limit the transferability of those dataset between domains, as the vehicle dynamics determine if the sensor noise or the time synchronisation is the limiting factor [25].

Aside from the actual generation of data, there has been limited research in developing fair benchmarking. Since 2010, the trend has been to develop a dataset, the tools to utilise the dataset, and then provide a method of benchmarking. This has lead to prioritising hardware elements for benchmarking [55]. Alternatively, some online benchmarking systems allow users to focus on the algorithm without having to worry about the hardware. For example, this is the case within the data science platform *Kaggle*, which allows users to benchmark algorithms across multiple fields with virtualised hardware [7, 69]. Unlike Kitti, this platform does not actively collect data in an experimental sense, instead it is providing a storage medium for all datatypes, as well as a method of integrating that data into benchmarking competitions. This system provides a flexibility to datasets that may be viable for benchmarking competitions otherwise developed for specific robotic systems, which was the case for many of the early autonomous systems datasets. However, by taking the emphasis on hardware away from the user, algorithm development has been shown to focus on accuracy rather than run time performance and deployability [47].

Chapter 3

Beyond-application, high-volume, high-quality data

The terms *high-volume*, and *high-quality* are the summation of a number of parts that need to work together to produce an outcome suitable for the beyond-application scope of a generalised dataset. High-quality not only refers to the data quality, which encompasses data fidelity and the steps required to capture and store the data, but also ensures that the data is synchronised and deployed in a manner that preserves that quality. Volume refers to the number of scenarios, the different environments, and the total culmination of the data. In the case of the beyond-application approach, the goal is therefore to produce volume in excess of what may be needed for algorithm development. This is achieved by deploying a high number of sensors (spatial coverage), with a high sampling rate (temporal coverage), and recording a multitude of scenarios over a large number of collection runs. This dictates the approach of building a collection vehicle, with sensors being selected first, before any other infrastructure. Over this chapter, the factors regarding digital sensor technology is covered, as well as how they are applied to autonomous systems development.

3.1 Data

At the core of this thesis is data. Fundamentally, this work involves finding the digital representation of the physical world. A sensor interprets a physical element, such as light, sound, forces, signals, or some other element of the tangible universe, and quantifies it so that it can be represented it as a measurement. This is an *analogue* data representation of the physical world [62]. However, a computer cannot capture these values continuously. The logical processor operates cyclically on a clock and there are moments in time between each cycle where nothing occurs and the signal at that moment is lost. This new, gap filled signal is simply the *digital* representation of the data source, and this process is how modern data is captured and processed. The analogue signal is processed into a digital format, and placed into a storage medium where it can be accessed in the future.

The quality of this digitised data is dependent on a number of elements, which can be divided into the steps taken before the data is processed (pre-processing), during (processing) and afterwards (post-processing). Pre-processing steps include ensuring that the source quality is of a high enough standard, the sensors are calibrated and tuned, and that the capture method is sufficient for the type of source that is collected. Post-processing ensures that the data is moved and stored in a manner that is suitable. If the data is

then used, this would also include any transformations on the data itself. This schema of data usage is the basis on how this thesis captures, manages and utilises the data.

By far the biggest impact on quality is the act of processing the analogue data into the digital representation. The following sections detail how the different aspects impact the overall digital data, and how the size and quality change, but there is one concept that is crucial to all of this and needs to be discussed first, which is *quantising*. Quantising is the method of taking infinitely changing analogue values and making them fit to a finite scale [60, 50]. This limit is imposed by a sensors resolution and sampling rate.

3.1.1 Resolution

The numerical range of data that can be captured is usually represented as the number of bits of data within a single data point, *bit depth*, where a bit is the smallest conventional piece of data representation and allows for the values zero and one. Whilst quite limited in what can be done with a single bit, it does allow for a succinct explanation on what happens as the bit depth increases. Using a camera with a single bit of depth, a pixel of data would show as either a `1` or a `0`, and will return a black pixel or a white pixel. If the camera was pointed at a grey wall, each pixel would fall to either the black or the white value, and the actual representation of the wall itself will be quite disfigured. If the bit depth was increased to two, there would be four potential values—`00`, `01`, `10` and `11`, or zero, one, two and three. If the range of colours is still between black and white, the values represented can be black (0), white (3) and two grey values in between (1 and 2). The grey wall will still be disfigured, but the difference between the true wall and the values in the data is less. At three bits, there are eight potential values between one and zero, and the difference gets smaller again.

Bit depth is only one part of the resolution of a sensor, and dictates the total range of a data a single point can hold. However, some sensors are not limited to capturing only one single point of data per sample. In this case, the resolution refers to both the bit depth, as well as the spatial coverage, which, for example, may be the height and width of a camera image. As resolution increases, it allows for greater separation and clarity of the space it is capturing. However, as with sample rate, increasing resolution increases the size of the data sample, and thus puts pressure on the throughput of the system.

3.1.2 Sample rate

The prior sections have dealt with how to quantify the measurements of the data, which are the tangible values of the world that have been captured at a specific point in time. This time domain is equally important in determining what can be captured. Part of this is due to the work pioneered by Nyquist, Shannon and Whittaker, and is referred to the *cardinal theorem of interpolation* [32]. This theorem establishes the link between the how fast a sensor captures data, and the maximum frequency of a continuous signal that can be captured. This capture rate, noted as *sample rate*, dictates the upper maximum of what a sensor can resolve.

The actual theorem states that perfect reconstruction of a signal can only occur at twice the frequency of the highest frequency, which establishes that the sample rate should be twice the highest frequency that warrants capturing [32]. As with resolution and depth, this sample rate is a key component in data population. At the low end of the frequency spectrum, sound waves (for human hearing) start at roughly 20 Hz and go to 20 kHz, thus putting the sample rate at 40 kHz to capture the entire range. At marine VHF radio frequencies, which is one of the sensors utilised within this work, the frequencies begin in the 150 MHz range, which would entail a sampling above 300 MHz to capture correctly. With a higher

sampling rate, there is the capability of discerning higher signals within an object, allowing for separation of signal components within an object [36]. At the top end, the light and radio wave sensors move into spectrums where conventional sampling per second is beyond practical, and therefore requires electrical engineering to step the captured frequency range to a range which a computer can handle. The actual scope of this design and construction is not required for this thesis, as this is thankfully handled by the sensors themselves, however this becomes quite a real problem during sensor validation, which is covered in Chapter 6, and is used to establish a method for radio sensor validation.

3.2 Automation

For autonomous systems, datasets play a crucial role in the development of robotic perception, with the goal being that these systems can correctly perceive the environment around them, and for that, they need to be able to correctly interpret and analyse the captured data. Whilst not exhaustive, a comparison between varying contemporary datasets [29], shows a clear definition between the complexity of these datasets use cases, based on which category the authors intend the data to be used for. This clear separation implies that the trend for dataset development has been to focus on application-centric development, with the intent to develop the dataset for a specific branch of autonomous system design. Referring back to how a dataset is developed, this typically manifests as tailored pre-processing.

This is not an problem for the one-to-one research projects which require this specific pre-processing, but this becomes a limiting factor when attempting to use the dataset for projects outside the original scope. The solution is to approach dataset generation outside the application specific requirements, and instead develop a robust, generalised dataset, that focuses on the strength of overlapping sensor technology without curbing or tailoring the collection. Instead of breaking datasets into groups like *automotive* or *aeronautic*, a more straightforward approach is to ensure that the dataset provides the fundamental components of autonomous research – which, simply put, is high-volume, high-quality data in a kinematically diverse environment.

The volume and quality is dependent upon the sensor selection, but the kinematics are dictated by the research craft and the environment it is deployed in. For environments, there are a number of possible domains. Of these, typical examples include air, land and sea. Here, the maritime domain was selected for evaluation, with a comparison of acceleration dynamics undertaken against road vehicles to demonstrate the kinematic differences.

The vehicles velocity and position was measured using an ANavS Multi-Sensor RTK global navigation satellite system (GNSS), and the acceleration was measured using an KVH P-1775 inertial measurement unit (IMU). The sensors were mounted as centrally as possible on a 5 m trailerable surface vessel, which was towed behind a standard road car for the on-road tests, and driven in calm, littoral waters for the on-water tests. In both cases, the vehicle was driven normally, with the operator using the vehicle as intended. Fig. 3.1 is a snippet taken to highlight the vertical axis acceleration differences, demonstrating that the morphological surface of the water provided significant kinematics, even in calmer waters. This dynamic environment was further compounded with other vehicles in the area, as is shown when the vessel moved into waters that had been recently disturbed by a heavier vessel and subsequently finding itself without water to move on, seen with the maximum vertical acceleration obtained by the boat in Table 3.1. This table also shows the full comparison of both road and water tests.

Table 3.1: A comparison of acceleration forces on road and water vehicles. Total run time for the on-road test was 19.4 min and the total run time for the on-water test was 17.4 min

	Minimum ($\mathrm{m\,s^{-2}}$)	Maximum ($\mathrm{m\,s^{-2}}$)	Standard deviation ($\mathrm{m\,s^{-2}}$)
Road longitudinal	-5.271	5.548	0.701
Road lateral	-6.175	6.242	0.797
Road vertical	-1.136	22.322	1.006
Boat longitudinal	-18.373	8.082	0.521
Boat lateral	-18.811	6.179	0.575
Boat vertical	-5.655	88.734	1.271

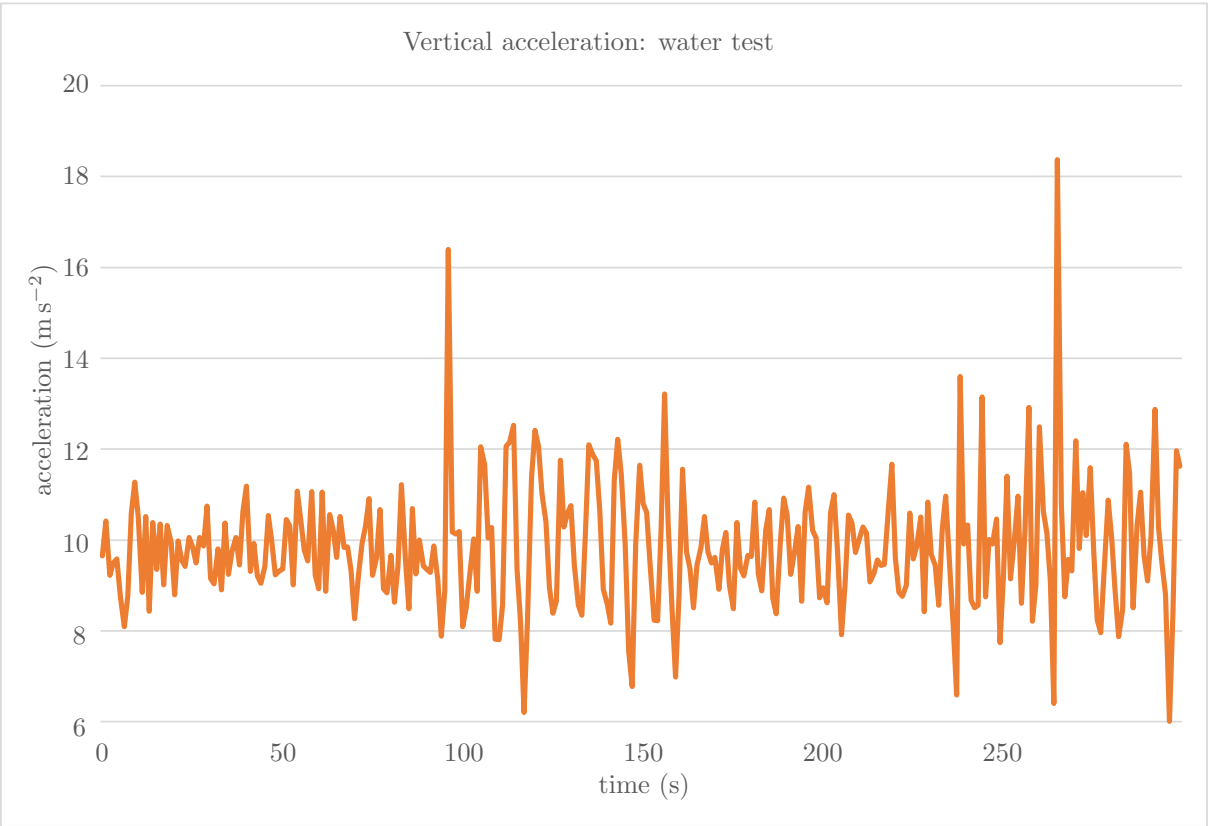

Figure 3.1: *A plot of acceleration forces in the vertical orientation over* 5 min*, taken from a surface vessel travelling at approximately* $40\,\mathrm{km\,h^{-1}}$*, in typical operation.*

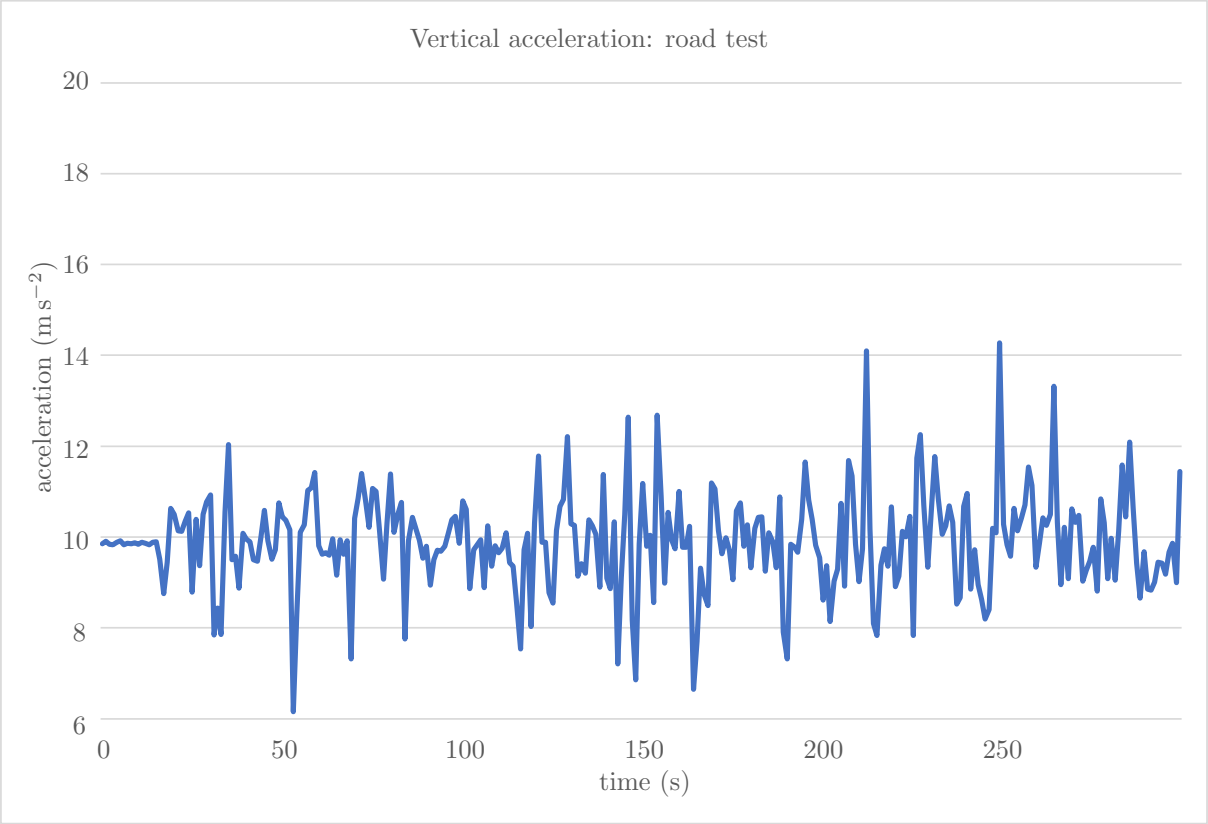

Figure 3.2: *A comparison of acceleration forces in the vertical orientation over* $5\,\mathrm{min}$*, taken from a road vehicle travelling at approximately* $40\,\mathrm{km\,h^{-1}}$*, in typical operation.*

3.3 Online distribution, benchmarking and evaluation

As mentioned, over the last few years there has been an increase in development of maritime datasets [43, 22]. The drive behind this has been mostly due to the lack of work in the domain and the focus is still on the maritime application-centric nature of the data capture. Whilst the motivation behind this research differs from the beyond-application motivation of this project, these datasets do highlight the difficulties of procuring data in an environment that is both unforgiving on equipment, and highly kinematic. It should also be noted that a comparison of established automotive datasets, and these relatively new marine datasets [13, 22, 45, 63] shows significant skewing in quality and quantity of the data. There is also a clear lack of scalability, as modern computation ability has vastly outgrown the quality the sensors can provide.

The first problem then, when developing a sensor platform that takes advantage of modern computing capabilities, is that of data storage and data distribution. As sensors improve resolution, depth and sample rate, the data generation rate increases. The introduction of datalinking, edge computing and wireless sensor networks [26], has only increased this generation rate, as data can be collated from multiple remote systems, as is the case of the Piraeus AIS dataset [63]. The result is that datasets are becoming inherently more complex and larger in size, which in turn makes distribution more difficult. Storage mediums have continuously grown, and costs have reduced, but there is a practical upper limit for personal computing storage and bandwidth, and whilst moving to an enterprise level for storage is feasible, it is costly, and limits the accessibility of open-access data to only those who can afford it.

Another problem is fair benchmarking, which has been shown to be missing in contemporary algorithm development [47]. It was also found that benchmarks towards open datasets were primarily focusing on *accuracy* rather than run-time performance such as evaluation of formal real-time capabilities (i.e., suitability to run in embedded robotic systems). Furthermore, it was found that the reported results from a few research studies could not be confirmed due to lack of conserved and fully linked software, missing documentation, or missing source code. In addition, since benchmarks are typically computed only when the publication in question is submitted, or possibly in some cases occasionally by the database providers for the purpose of updating the leaderboards over time, the specific numbers can rarely be trusted to be fair, for instance, due to differences in computational platforms. To allow fairness, benchmark results should always be computed and presented in a common centralised system, with one such proposal being the use of automated containerised methods [47].

The question therefore, is how to simultaneously solve data throughput limitations, and level the benchmarking to encourage computational cost reduction as a factor. This thesis does this through the use of containerised distribution where the users run their evaluations and benchmarking on a cluster provided by the project, outlined in Chapter 5. This online distribution allows the users to develop without moving data to a local machine, and each user has access to the exact same hardware infrastructure, with performance measured in both accuracy, as is typical of current benchmarks, and computational cost.

Chapter 4

Data collection

There are specific challenges when building a data collection surface vessel that are intrinsically linked to the maritime domain. There are no fixed lines for power or data management, and space and weight are still luxuries when compared to a land based system. Not only do these limitations dictate the power, computational and network architecture, but it also limits the amount of locations for sensor placement. Thus, the first decision when building a surface vessel is to determine the size of the platform. A larger vessel allows for a larger payload, offsetting many of the prior problems. However, a larger vessel also has higher maintenance and running costs. There is also the problem of practical maintenance, being that if a vessel requires a crane or drydock facilities to perform general works it can limit the usability of the vessel. If the sensors or systems fail during a data collection run, then this problem compounds the repair procedures. As the goal is to provide a highly kinematic data set, should be seriously considered as a possibility.

With this in mind, the decision was made to procure a trailerable data collection platform. The vessel, named Seahorse, seen in Fig. 4.1, is a 5 m Ockelbo B16AL. This vessel is not without limitations, as it lacks an on board power supply suitable for high-volume data procurement. This is compensated by the flexibility this vessel size confers, as the entire platform is able to be moved to a laboratory with relative ease. This chapter, therefore, looks at the resolving the problems with selecting a vessel this size, and then looks at validating the sensor selection by analysing the inertial sensors for maritime suitability.

4.1 Building Seahorse

When building Seahorse, the initial concern was the priority of development. The vessel had an upper payload limit, and this had to be distributed between the sensor and server selection, the power and network architecture, the weatherproofing, and temperature control. Power had to be sufficient enough to power the sensors and servers for an extended period, but by increasing the battery system the number of sensors would have to be reduced to stay within the weight limit. Motivated by the datasets goal of high-volume and high-quality data, the emphasis of the design was placed on sensor selection, with processing and networking secondary. Weatherproofing was established after the server and sensor size was determined, and finally, the remaining weight was given to the power architecture.

Figure 4.1: *An image of the research vessel Seahorse.*

4.1.1 Sensors

From the autonomous systems perspective, there are two key components to making any sort of navigation decision. The first is that the decision maker needs to know where things are, and where they are going. The second is to know what those things are. Traditionally, the dataset collection vehicle would be built with an application in mind, and this would dictate the sensor selection. In the case of an abstract dataset, this selection is more difficult, but not impossible. The sensors need to be applicable to multi-domain environments, be kinetically robust, and be able to provide the solutions to the *where* and the *what* [54]. With this in mind, the decision was made to include a 360° camera suite, a 240° high-quality camera suite, a 360° *light detection and ranging unit* (lidar), a 360° *radio detection and ranging unit* (radar), a GNSS, and an IMU. These, at the bare minimum, provided the position of the vessel, and short, medium and long range data of the objects and space around it.

The sensors selected can be seen in Table 4.1. In the following sections where discussion is given on the decision making for the various components, a top down approach was taken, where the most computationally intense and complex sensor was used for explanation.

4.1.2 Sensor validation

Each sensor was selected based on the goal to deliver high-quality data. However, to ensure each sensor is actually both precise and accurate, experimentation was undertaken to validate the assumed qualities that made these sensors high-quality. Whilst the datasheet provided by the manufacturers offers a level of guarantee, there is always the possibility of mechanical, communication or software errors, and there needs to be a method for ensuring each sensor can meet the manufacturers guarantee, or at least have a method to generate a configuration file that shows exactly what that particular sensors error is.

The *Allan variance test* is one such method to determine the fundamental components of the cumulative

Table 4.1: A table outlining sensor selection of the Seahorse platform

Type	**Qty.**	**Sensor**
IMU	1	KVH P-1775
GNSS	1 (3 ant.)	Anavs MSRTK
Monochrome camera*	4	Flir ORX-10G-71S7M-C
RGB camera*	2	Flir ORX-10G-71S7C-C
360° documentation camera	2 (8 cam.)	Axis F44
Lidar	1	Ouster OS1
Lidar	2	Ouster OS2
Marine radar (X-band)	1	Simrad Halo 20+
*Camera lens	6	EO 16mm f/4 1″ Pr lens (43.2°)

Table 4.2: A table outlining each unit that was considered as an IMU for Seahorse, and the length of the test undertaken to determine Allan variance

Unit	Sample rate (Hz)	Sample size (seconds)
Panasonic MEMs	100	86490
Anello A1	150	123685
KVH P1775	1000	41306
OxTS RT3000GG	65	68272

error of a time based sensor, which was undertaken to show the reliability of the inertial sensor, being the KVH P-1775 fibre optic gyroscopic (FOG) IMU. The power spectrum and Allan variance methods are modelled off the work of El-Sheimy, Hou, and Niu [57], and the 1998 IEEE standard for FOG testing [23]. These methods develop a signal spectrum analysis, and use this to justify a relationship of the deviation of the signal, which is made of the varying noise signals and this spectrum analysis. This formula, shown in Eq. 4.1, is the basis for all noise signals, and allows for extraction of the signals through the analysis of a log–log plot of $\sigma(T)$ versus T, where $\sigma(T)$ is the Allan deviation, and T is the total time of an consecutive cluster taken from a sample of the sensor.

$$\sigma^2(T) = 4 \int_0^\infty S_\Omega(f) \frac{\sin^4(\pi f T)}{(\pi f T)^2} df \tag{4.1}$$

where $S_\Omega(f)$ is the spectrum analysis of the random process $\Omega(T)$.

Experimentation

Four various IMUs, each a contender to be used for maritime applications, were left stationary in a sealed lab and data was recorded for no less than eight hours. Table 4.2 specifies the units, their sample rate and length of the sample in seconds. To determine noise, random walk and bias, the following algorithms were applied. As each sensor has a different sample rate, the algorithm works with sample *clusters*, of which the total sample N is split into consecutive data points n (with $n < \frac{N}{2}$). Each cluster has a time T, which is the number of samples in the cluster, multiplied by the length of a single sample. The Allan variance σ, for T was obtained with Eq. 4.2, where Ω is the value in the sample:

$$\sigma^2(T) = \frac{1}{2(N-2n)} \sum_{k=1}^{N-2n} \left[\overline{\Omega}_{k+1}(T) - \overline{\Omega}_k(T)\right]^2 \tag{4.2}$$

Velocity and angle random walk

The random walk of the sensor is accumulated drift as the sensor integrates a signal. In IMUs this is split into angle random walk for the gyroscope and velocity random walk for the acclerometers. In both cases, the equation to resolve both is found by integrating within Eq. 4.2, giving

$$\sigma^2(T) = \frac{N^2}{T} \tag{4.3}$$

where N is the random walk coefficient. This can be represented by a $-\frac{1}{2}$ slope in a log–log plot of $\sigma(T)$ versus T. The magnitude of the walk can be fount at $T = 1$.

Bias

Bias requires a more complex resolution, and involves Fourier transforms to be modelled [57]. For this, the relationship between σ and the power spectral density (PSD) must be determined, as found in Eq. 4.1. The signal of the bias can be determined using

$$S_\Omega(f) = \begin{cases} \frac{B^2}{2\pi}\frac{1}{f} : f \leq f_0 \\ 0 : f > f_0 \end{cases} \tag{4.4}$$

where B is bias instability coefficient, and f_0 is the cutoff frequency. Substituting this into 4.1 gives

$$\sigma^2(T) = \frac{2B^2}{\pi}\left[\ln 2 - \frac{\sin^3 x}{2x^2}(\sin x + 4x\cos x) + C_\mathrm{i}(2x) - C_\mathrm{i}(4x)\right] \tag{4.5}$$

where $x = \pi f_0 T$ and C_i is the cosine-integral function [57].

Using the same plot of $\sigma(T)$ versus T, bias is determined by determining the shelf located at $\sigma(T) = \sqrt{2\ln\frac{2}{\pi}}$, (0.664).

Rate random walk

Rate random walk can be determined using the following

$$\sigma^2(T) = \frac{K^2 T}{3} \tag{4.6}$$

Where K is the rate random walk coefficient. On a log–log slope of $+\frac{1}{2}$, on the aforementioned plot of $\sigma(T)$ versus T, the magnitude can be found at $T = 3$.

Industry standards

Using the definitions provided by SOLAS [51], the ISO standards [24] and the definitions provided by IEEE [23], the results found were adjusted to the defining units, and a comparison was drawn to determine usability, as per the 1998 IEEE standard for FOG units [23].

Analysis and interpretation

A succinct overview of the results can be seen in Tables 4.3 and 4.4. For simplicity, the goal of each value is to be as close to zero as possible. As the manufacturers have presented their datasheets in varying units, the results shown here have been standardised to seconds (s), radians (rad), and metres (m).

Whilst informative as a comparison for varying IMU sensors and their technology, the primary goal was to determine how trustworthy the sensors are. In the case of the KVH P-1775, it was determined that the unit did not perform to the datasheet specifications [52], but that the unit was still quite trustworthy, as the accumulated errors were consistent enough to be modelled, and that the accumulated error growth was so minuscule it did not impact a typical data run due to the power limitations of Seahorse.

Table 4.3: IMU results: Acceleration. The best values are highlighted as follows: Velocity random walk, N ($\frac{ms^{-1}}{\sqrt{s}}$) is in italics; rate random walk, K ($ms^{-1}\sqrt{s}$) is in boldface; bias, B (ms^{-1}/s) is underlined

	X	Y	Z
Panasonic			
N	1.39E-03	1.30E-03	1.29E-03
K	3.25E-05	4.76E-02	6.01E-05
B	2.54E-03	1.46E-03	8.46E-04
Anello			
N	4.80E-04	4.54E-04	4.70E-04
K	8.16E-06	8.37E-06	1.17E-05
B	3.87E-04	3.84E-04	2.83E-04
KVH			
N	3.16E-04	3.17E-04	3.14E-04
K	5.28E-04	2.50E-04	9.27E-06
B	2.09E-03	4.16E-04	4.41E-04
OxTS			
N	*2.95E-04*	*1.54E-04*	*1.79E-04*
K	**2.29E-06**	**5.88E-07**	**2.67E-06**
B	<u>6.88E-05</u>	<u>6.62E-05</u>	<u>6.55E-05</u>

Table 4.4: IMU results: Angular rotation. The best results are highlighted as follows: Angle random walk, N ($\frac{rad/s}{\sqrt{s}}$) is in italics; rate random walk, K ($rad/s\sqrt{s}$) is in boldface; bias, B (rad/s) is underlined

	Roll	Pitch	Yaw
Panasonic			
N	8.70E-03	1.03E-02	7.41E-03
K	2.72E-03	1.63E-06	2.60E-03
B	1.01E-04	1.02E-04	8.50E-05
Anello			
N	1.08E-04	1.47E-04	6.12E-05
K	4.99E-07	8.66E-07	1.96E-07
B	2.07E-05	2.19E-05	8.52E-06
Anello (FOG)			
N	-	-	1.44E-05
K	-	-	**1.22E-08**
B	-	-	5.91E-06
KVH			
N	*6.52E-06*	*6.24E-06*	*6.51E-06*
K	**1.50E-08**	**9.43E-08**	2.08E-08
B	<u>5.92E-07</u>	<u>7.71E-07</u>	<u>6.49E-07</u>
OxTS			
N	1.09E-04	1.13E-04	1.14E-04
K	5.58E-07	4.28E-07	6.30E-07
B	4.35E-05	3.75E-05	3.67E-05

4.1.3 Servers

The various components of the servers can be broken down into the minimum requirements based upon the sensor selection and power availability. An attractive approach that maintains as much battery reserve as possible would be to use long term power management systems [1]. In practice, this often leaves spooling gaps as the central processing unit (CPU) switches between lower power and higher power states. In fact, when utilising the FLIR Oryx cameras, there is considerable on/off switching of CPU cores. This introduce flickers of cycle skips, and the subsequent sample rate analysis of a brief capture shows this when running both a colour and monochrome camera in parallel, which can be seen in Fig. 4.2. The solution, therefore, is to ensure that the system is spooled before and during the entire data capture, especially when presented with high bandwidth devices. CPU switching should also be discouraged, and therefore, a high cyclic, multiple thread CPU architecture was chosen, with the AMD EPYC 7352 selected, in a 2x2 server configuration (two servers, with two CPUs each). The higher bandwidth sensors could be provided dedicated CPU threads, and were isolated from all other logical operations. Short term storage, in the form of random access memory (RAM), was based on the minimum amount required to provide high throughput buffers for these sensors. Mathematically, the FLIR camera can produce a depth, resolution and sample rate at the following:

$$[3208\,\mathrm{px} \times 2200\,\mathrm{px}] \times 12\,\mathrm{bit} \times 77\,\mathrm{Hz} \tag{4.7}$$

Totalling 815 MB s^{-1}, with each frame using 11 MB of data.

To ensure each sensor had ample room for short term storage, each sensor was given a 200 frame overhead, which adds to 2.2 GB per camera. Factoring in the other sensors, each server was given 128 GB of RAM. The last section is graphical processing. As mentioned, the FLIR cameras can produce 12 bit images. There are considerable throughput savings to be made with 12 bit video encoding on a graphical processing unit, which would drastically increase the long term usability of the work, as well as data collection run time, due to the compression abilities reducing storage needs. For this then, each server was given a Nvidia GeForce RTX 2080 and a Nvidia Quadro RTX 4000 which is capable of transcoding six streams, each with video at 3208×2200px and with a bit depth of ten. At the time of writing there are limitations to this capability, but as stated, the entire system is built around CPU computation for the data recording, so the benefits of having these GPUs will be fully utilised in future works. Examples of such use cases, in addition to data compression and transcoding, could be hardware-accelerated visualisation, on-line data quality assessment, or machine learning-based object detection and tracking giving real-time suggestions during the data collection.

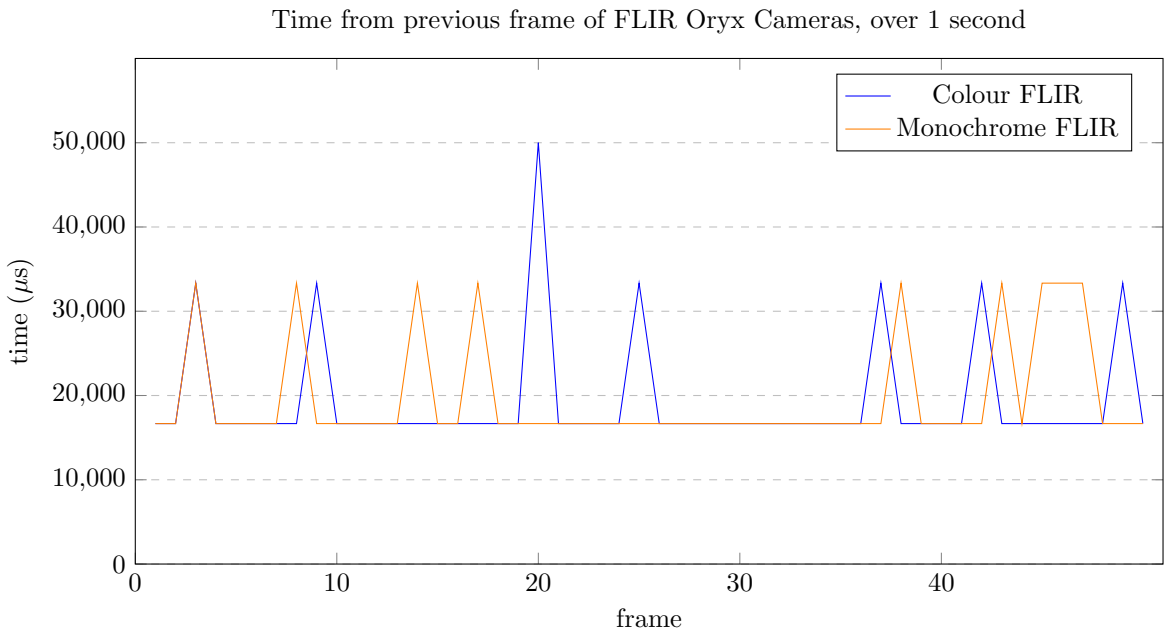

Figure 4.2: *An analysis of FLIR camera frame capture, with no CPU protocols in place to prevent spooling or switching. Shows the time, in microseconds, from the prior recorded frame. For this recorded second, the frame rate was set to* 60 Hz. *This indicates about a* 10 % *loss of data capture. When sampling at* 30 Hz *there was no frame loss.*

4.1.4 Storage

Finding a method of storage for high-volume data that can sustain the high kinematic state of the environment was in itself a challenge. For starters, the FLIR cameras, working at full resolution, bit depth and sample rate, generate data faster than a high volume mechanical drive can write. The Seagate Exos X series of drives were selected for long term storage, and each can reach a maximum speed of 270 MB s^{-1}, which is dwarfed by the production speed of the FLIR cameras as seen in Eq. 4.7. To resolve

this, a solid storage solution that uses a U.2 connection was utilised. The Micron 9300 provides an initial 15 TB of flash storage, and can be written to at 3500 MB s^{-1}. With this still unable to handle the three FLIR cameras per server, the sample rate of each FLIR camera is held at 60 Hz.

Each server utilises their own U.2 flash storage, which provides a total of an hour of recording before it needs to flushed and written to the mechanical drives. Final storage is to be held off the vessel in an enterprise level data centre, which is required for the vast amount of the total dataset, as well as providing the infrastructure for the deployment methods for the benchmark.

4.1.5 Networking and synchronicity

The network map of the Seahorse platform is developed to resolve a number of problems that occur with high-volume and high-quality data rates. Firstly, each FLIR camera requires a dedicated 10 Gbit link, to maintain the 815 MB s^{-1} (6.5 Gbit s). Each of the two SuperMicro server motherboards provide two of these links, but to provide links for all sensors each server was then equipped with Intel x710-T4 Network Interface Cards (NIC), which each provide four 10 Gbit links. To ensure equal working load, each server was given three FLIR cameras, as well as dividing the remaining sensors between the two. Remote ingress was done through the PC Engines APU, which also provides a serial link to the IMU.

To ensure each subsystem of Seahorse maintains synchronicity, there are two different methods of clock management. Network time protocol (NTP) and precision time protocol (PTP) are both utilised to ensure synchronicity between the different servers and sensors. Both methods work on a stratum schema, whereby a reference clock is pushed as a hierarchy through the network [46]. In NTP, the reference clock is a trusted time-server, and each client requests an update from the step above for their local time. PTP uses the most precise clock on the network as the reference clock, and pushes this to all clients, with offsets generated between the client and the reference clock, regardless of how many steps are between the two. Ultimately, both methods provide millisecond accurate synchronicity between two clients. PTP has the benefit of providing nanosecond precision when configured with a reference clock that can support atomic precision. As Seahorse does not carry one of these reference clocks, a novel hybrid method was introduced to provide synchronicity.

The Seahorse server cluster (Seahorse 0, 1 and 2) are all PTP compatible, as are the FLIR cameras and the Ouster Lidars. For these devices, a typical PTP network is established, with Seahorse 1 providing a grandmaster clock, using one of the Intel NIC cards as the precision clock. The average offset detected by the network cards for this PTP system between the servers sits at 50 ns. However, the network card, not being atomic precise, will introduce drift that will impact the entire network. For the purposes of data collection, this drift will not have time to accumulate during a typical data recording session, but it was notable during maintenance downtime. To prevent this, the ANAVS GNSS system provides a NTP server, corrected by GNSS time, which is used to periodically correct the grandmaster PTP clock when required. When the GNSS is under cover, a backup link to online NTP servers is maintained to provide the same functionality. To ensure that the NTP server does not cause the PTP layer to flicker when recording, the NTP update is disabled and instead takes its reference time from the PTP layer. The AXIS documentation cameras have their own network devices, which are not PTP compatible. To solve this, a second NTP layer is introduced inside Seahorse 1 and 2, which takes the PTP time as the source, and provides this to the AXIS network units as an NTP message. This time management can be seen as a whole in Fig. 4.3.

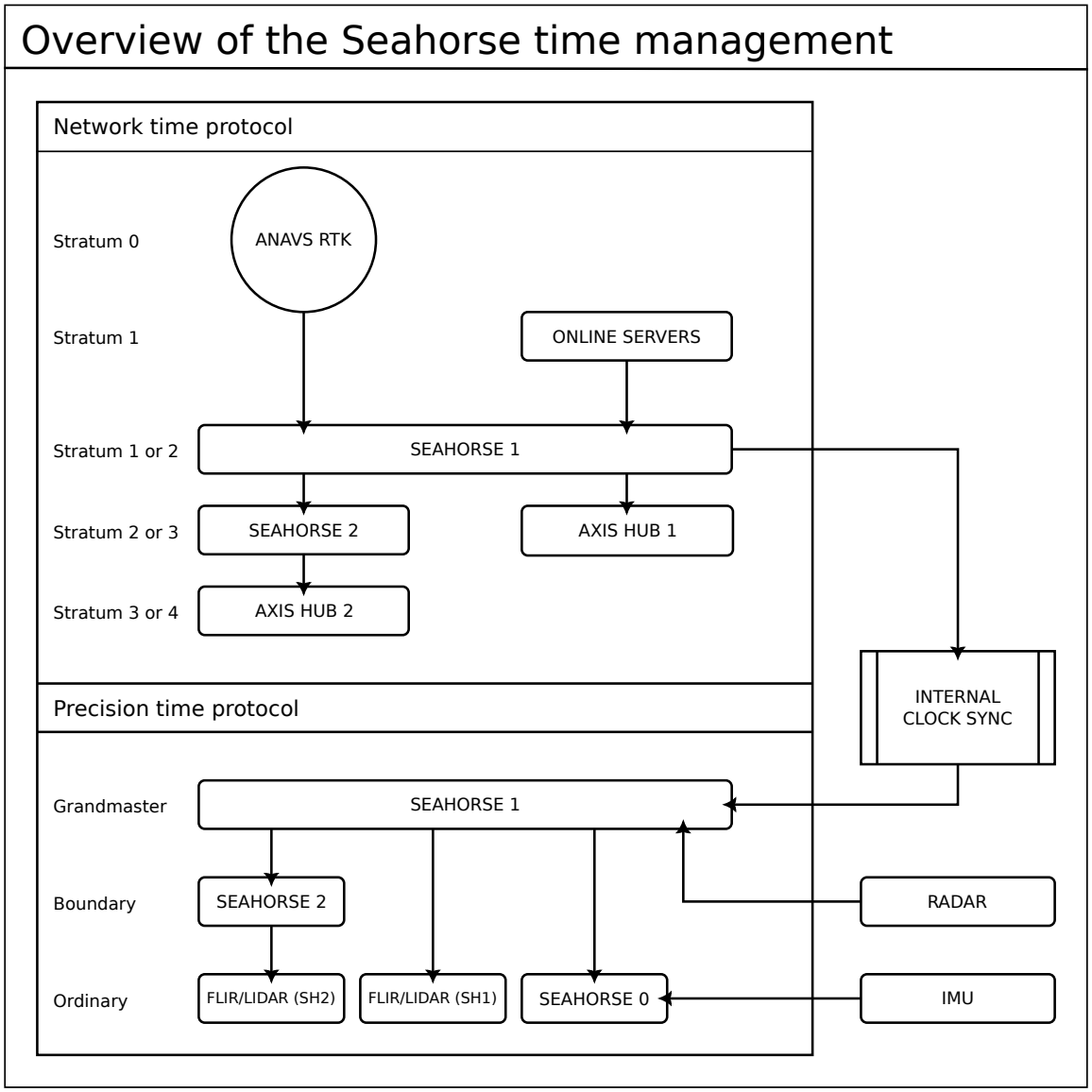

Figure 4.3: *The time hierarchy of the PTP and NTP system with the Seahorse platform.*

4.1.6 Software architecture

Each sensor requires logic to communicate and send commands, as well as receive the sensor data. There is also a need for logic to process and store this data. There is the option of doing this as an entire software package, but another approach demonstrates that there are significant benefits to a containerised method of software development for the Seahorse platform [47].

Containerisation simply separates all of the steps of this structure into their own software package, which is deployed inside an isolated environment. The environment is allowed to use the parts of the host required, such as an Ethernet port to communicate to a camera, but otherwise stays separate from every

other container running at the same time. This was selected as the method to build all the components of the Seahorse system. For this, the *Docker* software was used to deploy each component, and each container can be placed linearly between a sensor and the final recording. Using the OpenDLV software architecture, each container communicates using a standard message template, allowing for interchangeable logic blocks [4].

For data collection, each sensor has a container that handles the communication between the sensor and the server, and then processes the data stream into the standard message template, which is then openly broadcast to the network. In some cases, the sensor data may be too large to broadcast as a whole sample, and is instead stored straight to disk. In all cases, every message receives an individual timestamp.

4.1.7 Power

There is an argument to be made that the power requirements should be considered before the sensor and server selection. However, as the goal of this project is to develop the highest fidelity dataset, the power selection should compliment the sensor selection. The platform is powered by a forty-five horsepower engine, and whilst there is an alternator system on board to charge the starting batteries, this charger is separated from the payload location and unable to provide a long term power solution.

With sensors and servers selected, the remaining payload weight was given to the vessels architecture to support weather and shock proofing, and finally, the last factor was selecting a power source. For this, the only particular was that it had to, as a minimum, power all the previous components for the time taken to move the vessel from the Chalmers Revere vehicle laboratory to the test area, and then run a single test. Ideally, and with the capability to quickly remove data from the U.2 drives, the total power should be able to handle multiple test runs, in case of data corruption or other sensor problems that may be resolvable whilst underway.

The selected unit is a 6000 kW h Mastervolt battery, which provides a 24 V line. This line is stepped down via a transformer bank to provide continuous 12 V to the sensors that require the lower voltage. Total run time with the entire setup is temperature dependent, and it was noted that through the colder months of the Swedish winter, the battery system lost 20 % of the charge before the vessel had even reached the launching spot.

4.1.8 Temperature control

Most of the components within the servers and sensor suite are rated for quite a high temperature. However, the entire system is enclosed to prevent weather damage and when under stress, the system quickly reaches a temperature that puts strain on various components. As the servers have closed loop water cooling solutions, the network interface cards, which are not part of that loop, are limited with the amount of active cooling on them. When under load, they reached their maximum operating temperature of 55 °C, where the units throttled the datalink to prevent overheating and damage. This, plus the problem mentioned prior with the batteries being too cold, led to the development of a hot-cold aisle system, where cool air is rotated to the front of the hardware, and the server exhaust is then vented into a contained aisle housing the sensors and battery that require a constant temperature. An overpressure outlet allows extra heat to be vented back to the operator, which is graciously appreciated during the cooler weather.

4.2 Collection runs

The data collection runs that have been undertaken have been through the littoral waters of the western coast of Sweden. These waters were mostly the Göteborg river, as well as the river mouth and shipping regions. Each run, as limited by the power and storage capabilities, was roughly an hour in length. The operators of the vessel would first place the vessel in a relatively stationary position, with sensors angled in a way that there was some static object in a frame to produce a starting point.

Due to manufacturer protocols, each sensor had a unique spooling time, which was the time taken between starting a recording and the actual first value being captured. As each sensor had time synchronicity, this did not impact the actual data collection, however this did tend to leave the first minute of the dataset with a staggered sensor deployment. To overcome this, another container was built that allowed an operator to inject each other container with a 'record' message, and was broadcast once the slowest sensor was spooled. The OpenDLV architecture allowed for this to be sent across the entire network, which effectively synchronised the recording time between each server.

The current state of the collected data includes runs in various weather systems, in both calm and rough waters, and in both light and dark conditions. An example of a complex littoral scene can be seen in Fig. 4.4 taken with a FLIR Oryx 10GigE.

Figure 4.4: *A vertically trimmed colour image demonstrating pedestrians, vehicles, buildings, infrastructure, water morphology and weather deviations. Note the kinematic state of Seahorse skewing the image due to water-induced motion.*

Chapter 5

Data validation and benchmarking

With the data captured and in a state where it can be used, the next topic of this work is how it can be used in fair benchmarking of perception algorithms. Regardless of which algorithm is being deployed or tested, there needs to be a method of validation. The *ground truth* of the data is the data that can be accurately relied upon to make this validation. As exemplified in Sect. 3.1, all measured data will be disfigured (i.e., a digitalised approximation), but the ground truth is the data commonly agreed to be the best possible representation of a specific measurement, such as position or vehicle motion. The task of the perception algorithms using the dataset is then to try recreate the ground truth data exclusively using other data present in the dataset. Papers A, B and C introduce three different algorithmic methods that attempt to replicate this ground truth, which are different aspects of the GNSS data and the IMU data. They do this by using lidar, camera, and radar data, respectively. This chapter covers the the trialling of the initial dataset and reference algorithms, and the automated benchmarking facilities built to support the testing of these.

5.1 Online benchmarking

The initial online benchmarking dataset, henceforth called the *shakedown dataset*, was built during the shakedown cruise of Seahorse, which is the period of initial testing before regular use. This dataset is the primary dataset used for the online development and testing, and is also the dataset used within Papers A, B and C. Table 5.1 outlines the sensors, the datarate for each sensor group and the total datarate. With each test run aiming to be between 30 min and 1 h, the total summation of data can be calculated to be between 400 TB and 700 TB. The final goal of the project is to have this data be internationally disseminated to researchers in all domains. With a dataset of this size it would be impractical to have the users pull the data for each individual use case, and so the projects cloud backend was formulated, shown in Fig. 5.1.

Firstly, the data is transferred from Seahorse after a data collection run. There are two methods of annotation, which is dependant upon the verification of the sensors. AIS data can be used to show GNSS locations of other vessels, and when combined with on board RTK GNSS, allows for the automatic generation of an annotated topographic map. Annotations can also provided by the integrated online tool, where camera, lidar, and radar data is presented to external users. Algorithms can be uploaded by external researchers and developers. When a new algorithm, a new annotation set, or a new dataset was

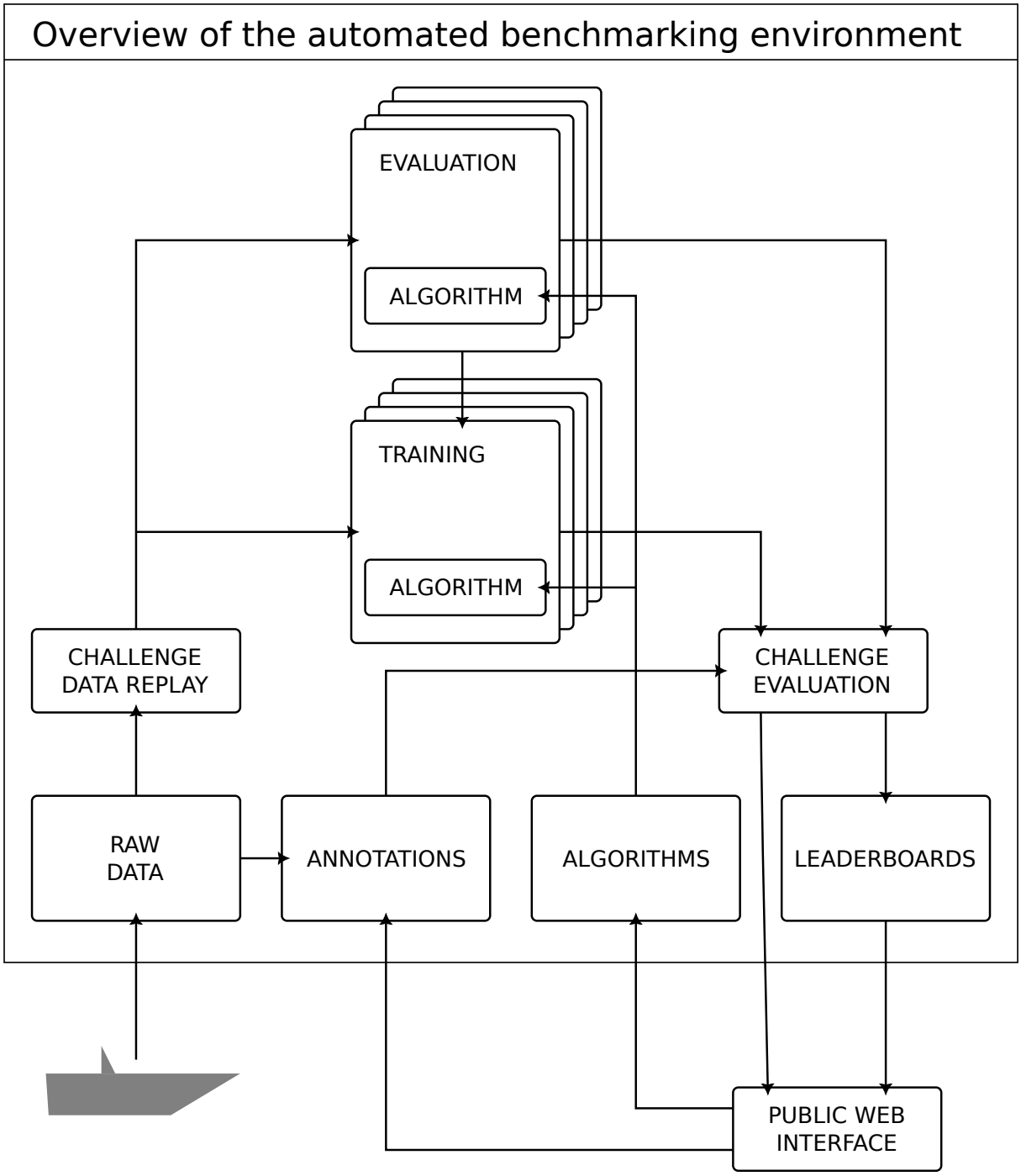

Figure 5.1: *The online cloud environment, detailing the method behind user-to-data principles.*

Table 5.1: Table outlining sensor data rates of the shakedown dataset

Type	**Sensor**	**Datarate** $(Qty.) \times [Res.] \times bit\ dep. \times freq.$(Hz)
IMU	KVH P-1775	$(1) \times [9] \times 32 \times 1000 =$ 288000 (0.036 MB/s)
GNSS	Anavs MSRTK	$(1) \times [4] \times 32 \times 100 =$ 12800 (0.0016 MB/s)
Monochrome camera	Flir ORYX-M	$(4) \times [3208 \times 2200] \times 12 \times 60 =$ 20325888000 (2540.74 MB/s)
RGB camera	Flir ORYX-C	$(2) \times [3208 \times 2200] \times 12 \times 60 =$ 8315136000 (1039.392 MB/s)
RGB camera	Axis F44	$(8) \times [1920 \times 1080] \times 8 \times 15 =$ 1990656000 (248.832 MB/s)
Lidar	Ouster OS1	$(1) \times [2048 \times 128] \times 60 \times 10 =$ 157286400 (19.5 MB/s)
Lidar	Ouster OS2	$(2) \times [2048 \times 128] \times 60 \times 10 =$ 314572800 (39 MB/s)
Radar	Simrad Halo 20+	$(1) \times [2048 \times 512] \times 8 \times 1 =$ 8388608 (1.049 MB/s)

added, then an automated evaluation is initiated and the results are registered on the leaderboards.

Note that the full envisioned system is not yet implemented. An early version was however demonstrated at the workshop *Autonomous Maritime Robotics: Digital Twins with Simulations & Cloud-enabled Massive Datasets*, at the 2023 ICRA conference, London.

5.2 Reference algorithms

Using the shakedown dataset, there has been considerable work in developing the first of the *reference* algorithms that will make use of the automated deployment. When deciding upon which algorithms to build, the emphasis was on the high-quality and high-volume sensors, being the FLIR cameras and the Ouster lidars. These sensors allowed for immediate testing of both object detection and vehicle odometry systems, while object detection, for example, would require annotation before validation. Vehicle odometry systems could be trialled directly against the ground truth sensors, with validation being done with the aforementioned Allan variance tests to ensure the truth was maintained. This led to the decision to build pure odometry systems, which allowed for direct comparison of accuracy and performance between the different algorithms.

The following three algorithms are covered far more in depth within their papers. However, there are key points that were established with their development that shaped the development of the online benchmarking tools, and warrant the use of a maritime platform for the creation of the beyond-application dataset.

5.2.1 Lidar kinematics

The first of the three papers, Paper A, looked at the generation of a *simultaneous localisation and mapping* (SLAM) system using one of the on board lidars. As mentioned, to ensure an autonomous vehicle has sufficient information to make intelligent navigation choices, it requires to know where it is and what is around it. SLAM is the application of generating a map of the surrounding area around the sensor, and then continuously placing itself within the map. Over time, as the position changes, the map is updated to include new data points, and the vehicle develops local odometry to update its position in the map, even without GNSS.

The use of GNSS allows for considerably accurate positioning, and the implementation of GNSS within contemporary perception algorithms has been covered in a number of studies over the past few decades [61, 68]. However, as shown with Table 3.1, there is the significant chance of kinematic upheaval which can cause sensor damage. This alone is significant enough to warrant exploring the development of multi-redundant algorithms, however it has been shown that the current trend for autonomous vehicle development is to assume that they will be operating in GNSS denied environments, and to build decision based systems that do not rely on external sensors [20, 19].

The method to develop this SLAM relies upon the three-dimensional (3D) representation of space generated by the lidar, which are commonly referred to as *pointclouds*. From this, each pointcloud is first transformed roughly to the global map through *normal distribution transformation* (NDT) [56], and then refined with a method of point iteration, called *iterative-closest-point* (ICP) [5]. Each frame is considered against the current map, initially transformed with NDT, and then refined with ICP. The final transform provides the current sensor offset in rotation, pitch and yaw from the prior, as well as the translation within the map. These factors allow the user to determine the vessels odometry.

These results are based upon the light reflection in the environment around the sensor, and this presents the first found problem being that the properties of the water presented a surface that not only did not provide clear reflection, but also changed dynamically with the weather [40]. The results also demonstrated that there are gaps between the three primary domains of autonomous vehicles, as the application of an algorithm that was developed for on a land based dataset failed when applied to the maritime domain. This was due to the vertical motion of the vessel.

5.2.2 Visual optic flow odometry

The second method utilises a method of image tracking called optic flow [47]. Optic flow looks at pixel intensity within specific clusters of an image, and attempts to find the same pixel intensity in a subsequent frame. This was applied within Paper B, which uses a single FLIR camera, downsampled to $[1920 \times 1200]$px [48]. After camera calibration, a direct sparse optic flow method was applied, which uses a keyframe gathering approach [33], which strips the image down to significant points. These points become the clusters of interest in the next frame, and then uses a direct approach to image comparison to find the points in subsequent images [10].

As with the SLAM method, the various components of odometry can be determined by how these points move between images, however this method requires transforming the two-dimensional image into three-dimensional space. The Sim3 method was used for validation [27, 64], but this relied upon the ground truth to find the smallest error margin. The final results showed that within the maritime environment, the morphology of the waves, and the lack of fixed points when looking only at the sea caused severe deviations in the determined odometry. Likewise, when the only thing in the frame was another non-static object moving similarly to the camera, the found velocity tended to drift towards zero. Consequently, without static details in the image, it was impossible to determine true velocity, however there was success in determining angular rotation.

5.2.3 Topographic flow

Utilising the method derived from the visual optic flow paper, a novel method of determining odometry was proposed in Paper C. *Topographic flow* first converts top-down sensor data into a visual image, and then applies various flow methods to determine vessel odometry. The benefits of using a time-of-flight sensor in this manner is that the range is a known value, which allows for easier conversion to velocity without seeing a known object in the frame. In this paper, a 360° radar was used, which allowed for full coverage of the static environment along a riverside as the vessel traversed through it.

This static environment allowed for the sparse method [35] to perform exceptionally well with determining angular rotation, however the novel method of determining velocity required key points to be in specific orientations, and was not able to maintain a consistent track. The dense method [12] also performed quite well with determining angular rotation, but the amount of deadspace within the frame, due to the lack of return from water and clear radar lines, also caused the velocity track to fail.

5.3 Automated benchmarking

With this in consideration, all three of these algorithms were tested for the first run of the automated benchmarking suite. The following details the methods of standardising the algorithm deployment within the OpenDLV framework, as well as the results found when comparing system performance. The main outcome from this is the ability to fairly benchmark different algorithms, using different sensors on an automated and standardised platform. Emphasis of this system is not the accuracy of the algorithms, as that is already established within the included papers, but is instead on the computational run-time performance of the algorithms.

5.3.1 Method

The input for this system uses the OpenDLV message format that was captured using the device containers outlined in Chapter 4. Each message contains timestamps as well as the sensor data, and using the cluon library it is possible to rebroadcast these messages exactly as they were captured. This standardises the time and sensor data inputs. The output, as it is monitoring both accuracy and run time performance, requires the determined value, as well as the time delta between the timestamp within the sample and the time output. A standardised template for algorithm development can therefore be seen as a classic input-output schema of containerised software development [4].

The three algorithms detailed in Papers A, B and C each use a different sensor, however, the goal of each algorithm is similar, which is to produce a method of vehicle odometry. Looking at the results and

Figure 5.2: *A monochrome image of the run, taken from a FLIR camera, and presented to the DSO algorithm.*

limitation shown from Papers A, B and C, it was shown that whilst each system presented a solution, the best testing factor for this demonstration benchmarking suite is to compare the run time performance and accuracy of the heading component. For ease of comparison, the yaw rate, which is presented as an angular velocity in $\mathrm{rad\,s^{-1}}$, is presented below after running through an automated benchmarking system.

For the purpose of this thesis, the benchmarking system was tested on an isolated server. The server used an an Intel(R) Core(TM) i5-6600K CPU and 16 GB of memory, and is equipped with a NVIDIA GeForce GTX 1060 with 6 GB of memory. The server is running Linux RT 6.3.3-rt15 (i.e., a real-time kernel), offering the best CPU prioritisation and standardisation for the different algorithms. Each algorithm is run individually, using the same data run, and continues till the end of the run, or until the algorithm fails. Figures 5.2 to 5.4 showcase the different inputs being presented to the algorithms, all taken from the same dataset, which is a 533 s run through the Göteborg river, shown in Fig. 5.5.

5.3.2 Results

Each of the three algorithms were evaluated one after the other in the automated benchmarking system. For the purpose of this demonstration, two benchmarking results are presented here, namely the angular velocity (in $\mathrm{rad\,s^{-1}}$) on the vertical axis (i.e., yaw rate), and the computational time (in µs) from each input (sensor measurement) to each output (given by the algorithm). Even though not important for the demonstration in this thesis, the angular velocity and other estimated signals would be automatically compared towards the ground truth signals. The result from the comparison would then be the basis for an algorithm *score* to be reported on the public leader boards. The computational time on the other hand

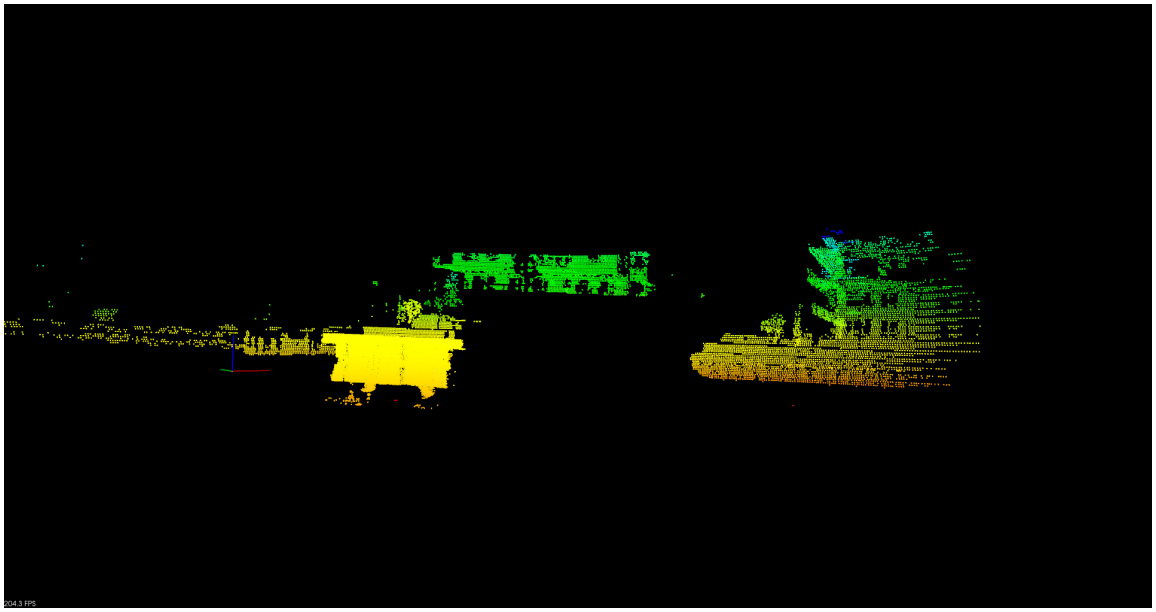

Figure 5.3: *A lidar frame captured by an Ouster lidar and presented to the lidar SLAM algorithm. This is the same frame as Fig. 5.2.*

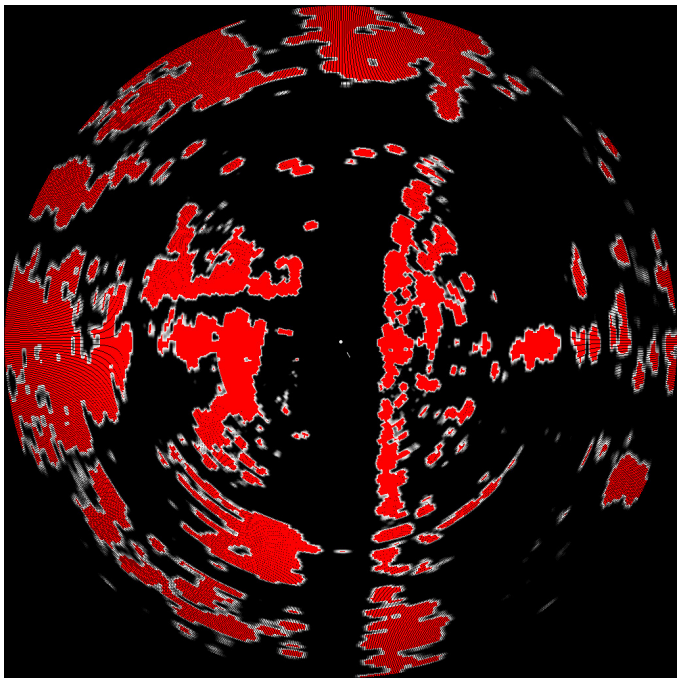

Figure 5.4: *An example of the image captured by the radar and presented from a topographic perspective, and used by the topographic flow algorithm. This is the same environment as Fig. 5.2.*

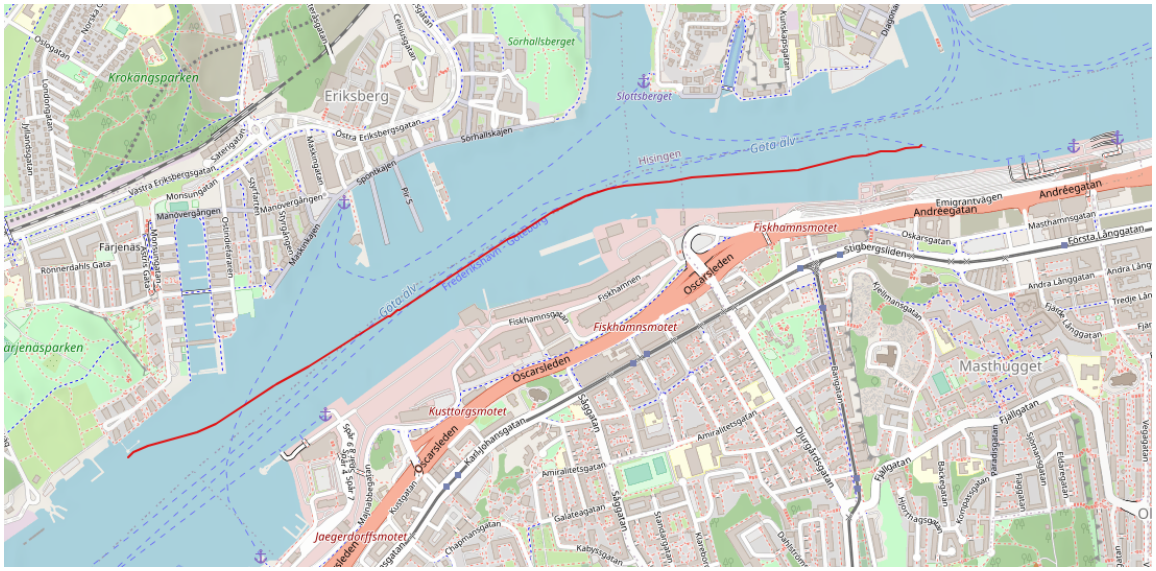

Figure 5.5: *The GNSS route used for the first automated benchmarking appraisal [49]. The route is left to right, over* 533 s*.*

would be reported as a basis to assess the algorithm's suitability as part of a formal real time system, where time predictability is crucial.

The angular velocity results are presented in Fig. 5.6, which are presented alongside the ground truth for comparison. The performance results are shown in Fig. 5.7.

Figure 5.6: *The angular velocity* ($\mathrm{rad\,s^{-1}}$) *of the ground truth, lidar SLAM, topographic flow, and DSO algorithms. Note that in the case of the DSO, no more data was available after* 210 s*, due to a logging error.*

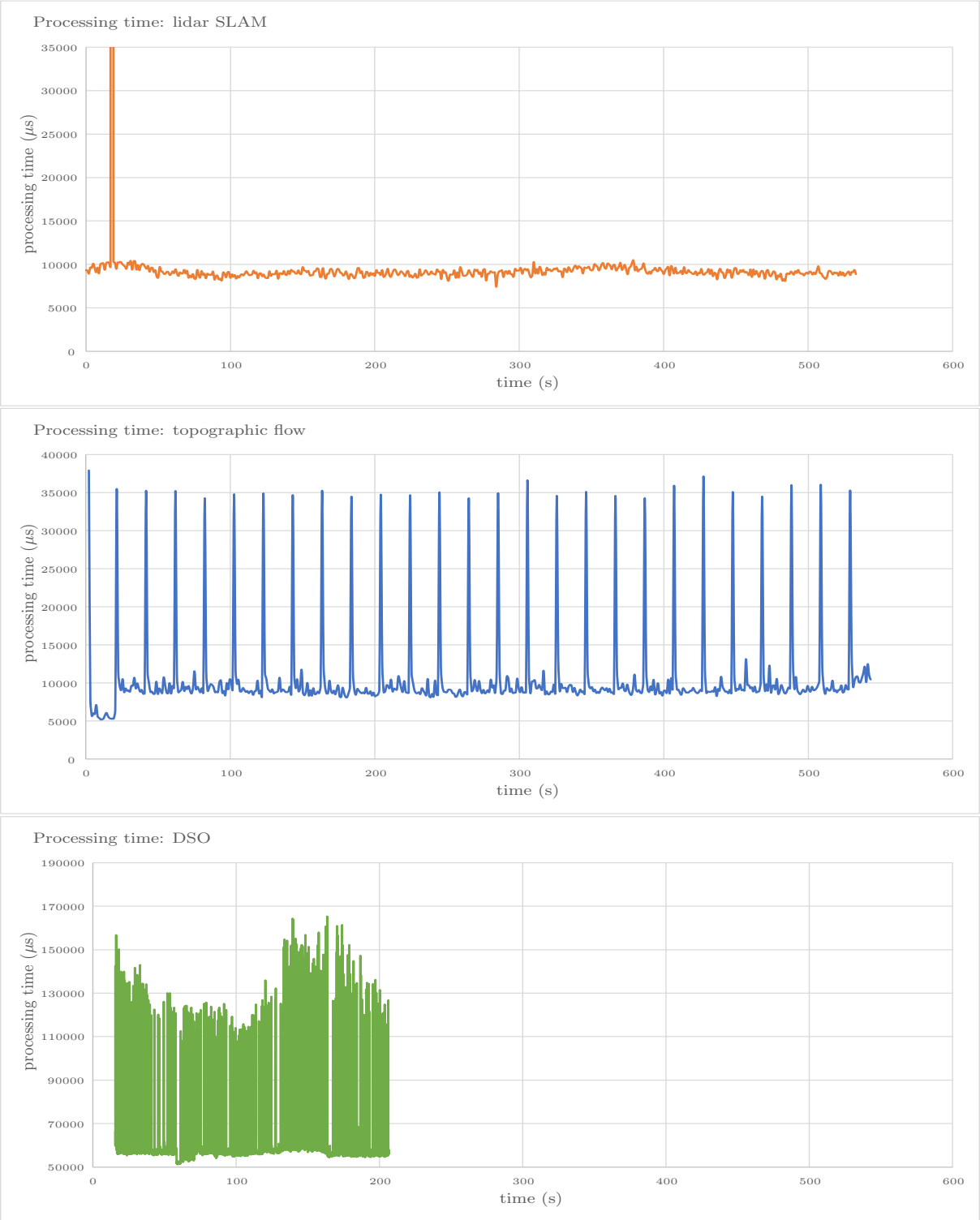

Figure 5.7: *The performance time of each processed result for the lidar SLAM, topographic flow, and DSO algorithms. At* 19 s, *there is a spike in processing time of the lidar SLAM which reached* 1.2 s. *As with the angular rotation shown in Fig. 5.6, the DSO results stop after* 210 s.

5.3.3 Analysis and interpretation

The automated benchmarking suite results demonstrated that an online approach is feasible for algorithm comparisons. With both the lidar SLAM and the topographic flow, the angular velocity trends matched those demonstrated by the ground truth, as shown in Fig. 5.6. As mentioned, this was not the focus of the benchmarking, instead, the emphasis was on the performance comparison. Fig. 5.7 shows a clear indication that the timing is working as intended, as the topographic flow algorithm reinitialises the tracks every twenty frames. With the radar operating at roughly 1 Hz, this can be seen with the spike in processing time every 20 s.

Importantly, there is no possibility for selecting favourable input data, as seen with the results of the DSO algorithms. There are six FLIR cameras, each covering a different angle off the front of the vessel. As mentioned in Paper B, the algorithm tends to fail when presented with a high lateral movements, which was seen when the DSO algorithm was used on outside cameras, and did not yield any plotable results. When run on one of the middle cameras the DSO managed to initialise and run longer. In this case the run ended due to an image capture failure in the shakedown dataset, but the results to that point were erratic and required considerably longer processing time when compared to the lidar SLAM or topographic flow methods.

When submitting an algorithm to the online benchmarking, the developer presents their algorithm to the dataset, without creating a local copy of the data to work on. This prevents tailoring the dataset to the algorithm to inflate accuracy results. There is also another benefit to this approach which was highlighted with the DSO algorithm. As mentioned, the longer DSO results failed due to a corruption of the dataset. The dataset can be swapped in place, which will rerun all the presented algorithms without each author having to download the new data and present their new results. The only limitation at this point is the storage capabilities of the automatic benchmarking system. This is still a far more attractive approach than requiring each user have TB of local storage, and highlights the importance of the online benchmarking system.

Chapter 6

Discussion and conclusion

This thesis initiated the concept of *beyond-application* datasets for machine perception. Since perception datasets have historically been associated with a specific application when being developed, the abstract methods of beyond-application dataset generation is fundamentally different. High-quality, high-volume and highly kinematic dataset design takes the emphasis away from the algorithms and gives considerably more freedom with the decisions of sensor selection, with the goal to capture measurements of the world as best as possible, in as many different environments and scenarios as possible. There will always be a benefit to tailoring a dataset for an application, if the goal is to produce an algorithm specifically to solve a problem that has a known sensor and computation structure beforehand. As an example, if the goal is to build and test a perception algorithm that was compliant with the 1972 convention on the international regulations for preventing collisions at sea [3], then the developers would require a dataset that contains both visual and acoustic data.

However, there are other ways to approach perception. Stepping back from the bounds of these international regulations allows for a more abstract based approach, and it is here the beyond-application method shows merit. Paper C demonstrated that object detection and tracking can be a potential byproduct of the topographic flow method. This method was found by first taking advantage of above standard data quality captured by Seahorse. This in itself is a key point of the beyond-application dataset, in that a user may not necessarily need the highest quality data, but may discover that there are performance and accuracy benefits, as well as completely novel solutions when the high-quality data is used. In the case of Paper B, the initial pixel resolution and bit depth was set quite high, at 3208×2200px and 12 bit. There are significant improvements with determining object edges when using the higher bit depth [2], but for this specific algorithm, the extra resolution and bit depth drastically increased the run time and were not necessary. At run time, only the top left quadrant of the image was required, with a resolution of 1920×1080px, with the bit depth reduced to 10 bit. This highlights the second part of the beyond-application approach, which is that there is always the option of stepping the dataset down to a lower quality when necessary. The opposite is far more difficult [67].

The emphasis on high-quality also forced a much higher scrutiny of time management systems compared to other contemporary datasets. Whilst every effort was made to ensure that sensors used the PTP layer, which offered significant precision and accuracy, some sensors simply did not have timestamping capability [25]. The shakedown dataset did not escape this, with the radar being one of these devices. An option that other datasets use is to have a mechanical system that triggers an electronic timestamp when a spinning sensor completes a cycle. However, the radar is enclosed and this option is not practical. The

next option proposed requires an external detector to determine the active beam angle and match it to the system clock. Chapter 3 discusses the difficulties with this, as the sensor in question uses a signal frequency that is highly impractical for any off the shelf solution [17]. This, along with refining methods to ensure data stability from the FLIR cameras, is the immediate next step for this work.

On the topic of FLIR stability, the data corruption was due in part to the highly kinematic environment the maritime platform is subjected to. In some instances, the vessel was subjected to acceleration forces exceeding $85\,\mathrm{m\,s^{-2}}$. The question is if this highly kinematic state helps solve the beyond-application approach to autonomous systems development, or if it hinders the data collection and make it unfeasible as a platform. The answer to this is that it does both. Developing an algorithm that can handle sharp impulses increases the real world applicability. As shown in Table. 3.1, the comparison between the road and water vehicle acceleration demonstrates that the water based platform has higher vertical kinetics, but both road and water shared similar kinetic factors. Whilst not the focus of the beyond-application dataset, this does have the benefit of allowing algorithms to be deployed across domains. To highlight this, the DSO algorithms presented in Paper B and the lidar SLAM algorithms developed in Paper A, were initially developed for land vehicles. To that end, these algorithms have been shown to work in a kinematic environment, regardless of which domain. The high dynamic range of the kinematics also has the benefit of demonstrating the limitations of the algorithms, allowing developers to improve upon them. Paper C shows that with high yaw rates, the algorithm slips and loses tracking.

However, this also carries the difficulties of building a system that can withstand those kinematics. Using an enterprise level server solves storage limitations, however these are typically not subjected to constant vertical shifts in acceleration, which is the case of a small, high-speed maritime platform. The shakedown dataset shows these limitations, as one of the servers shut itself down when acceleration in the vertical direction exceeded $25\,\mathrm{m\,s^{-2}}$. This was the biggest hindrance to continuous dataset collection, and has only recently been solved.

Lastly, the next group of benchmarking algorithms will attempt to solve the object detection components. This requires object annotation for accuracy, which has already begun. In the case for most other datasets, objects are split quite generically [29]. Due to the high-quality nature of the beyond-application approach, there is the possibility to derive individual components of an object, which highlights the annotation facilities of the online benchmarking suite. The public user interface, shown in Fig. 5.1, allows a developer to annotate the dataset for their own testing. As an example, a container ship could be categorised as a *ship*, or it could be split into various components, such as *bridge*, *hold* or *hatch*, depending on how the developer wishes to test their algorithm. For this project, the next benchmarking algorithm will begin by attempting to separate these various components of standard shipping designs.

6.1 Conclusion

There are four main components of this project, summarised by the four research questions outlined in Chapter 1. The first, is to determine if a beyond-application dataset is suitable for autonomous systems development. Papers A, B and C all show that autonomous navigational systems can be developed with such a dataset. The next question refers to the maritime platform, and if it is suitable for the development of this dataset. Whilst not conclusive, the highly kinematic environment demonstrates merit in allowing developers to refine algorithm development. Unfortunately, there are also problems with building a platform that can withstand these kinetics, but data collected after the initial shakedown has so far proven to be continuously stable. This has led to the first successful 90 min continuous run, and has also allowed for an in depth exploration of the third question, which looks at the best practices

for the deployment of such a large volume of data. For this, it has been shown how having an online benchmarking system where the developer can push an algorithm towards the data, rather than having the developer download the data, allows for continuously growing data without negatively impacting said developer. Using the same hardware with a standardised remote benchmarking pipeline also enables great opportunities for run-time evaluation. Such evaluation could for instance result in fair indications on how suitable algorithms are in an industrial (i.e., embedded) setting. Lastly, this online benchmarking approach removes the ability to select favourable data, and forces the same hardware for testing across every test. This is the best way to ensure fair, accurate and reproducible benchmarking.

Bibliography

[1] H. Aydin et al. "Determining optimal processor speeds for periodic real-time tasks with different power characteristics". *Proceedings 13th Euromicro Conference on Real-Time Systems*. IEEE. 2001, pp. 225–232.

[2] S. Battiato, A. Castorina, and M. Mancuso. High dynamic range imaging for digital still camera: an overview. *Journal of electronic Imaging* **12**.3 (2003), 459–469.

[3] M. R. Benjamin and J. A. Curcio. "COLREGS-based navigation of autonomous marine vehicles". *2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No. 04CH37578)*. IEEE. 2004, pp. 32–39.

[5] P. J. Besl and N. D. McKay. "Method for registration of 3-D shapes". *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. Spie. 1992, pp. 586–606.

[6] K. Blanch and O. Benderius. Topographic flow based odometry. *Submitted to the Journal of Intelligent & Robotic Systems* (2023).

[7] C. S. Bojer and J. P. Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting* **37**.2 (2021), 587–603.

[8] H. Caesar et al. "nuscenes: A multimodal dataset for autonomous driving". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[9] J. Choi and S. Lee. Legal status of the remote operator in Maritime Autonomous Surface Ships (MASS) under maritime law. *Ocean Development & International Law* **52**.4 (2022), 445–462.

[10] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence* **40**.3 (2017), 611–625.

[11] A. Engström et al. "A lidar-only SLAM algorithm for marine vessels and autonomous surface vehicles". *Proceedings of the 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, (CAMS 2022)*. 2022.

[12] G. Farnebäck. "Two-frame motion estimation based on polynomial expansion". *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*. Springer. 2003, pp. 363–370.

[13] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3354–3361.

[14] A. Geiger et al. The kitti vision benchmark suite. *URL http://www. cvlibs. net/datasets/kitti* **2**.5 (2015).

[15] A. Geiger et al. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* **32**.11 (2013), 1231–1237.

[16] J. Geyer et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320* (2020).

[17] R. Goncalves Licursi de Mello, F. R. de Sousa, and C. Junqueira. "SDR-based radar-detectors embedded on tablet devices". *2017 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*. 2017, pp. 1–5. DOI: 10.1109/IMOC.2017.8121126.

[18] E. Gundogdu et al. "Marvel: A large-scale image dataset for maritime vessels". *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*. Springer. 2017, pp. 165–180.

[19] J. Han et al. GPS-less coastal navigation using marine radar for USV operation. *Ifac-papersonline* **49**.23 (2016), 598–603.

[20] N. Hudson et al. Heterogeneous ground and air platforms, homogeneous sensing: Team CSIRO Data61's approach to the DARPA subterranean challenge. *arXiv preprint arXiv:2104.09053* (2021).

[21] B. Iancu et al. ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sensing* **13**.5 (Mar. 2021), 988. ISSN: 2072-4292. DOI: 10.3390/rs13050988. URL: http://dx.doi.org/10.3390/rs13050988.

[22] B. Iancu et al. ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sensing* **13**.5 (2021), 988.

[23] IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros. *IEEE Std 952-1997* (1998), 1–84. DOI: 10.1109/IEEESTD.1998.86153.

[24] *Ships and marine technology — Marine gyro-compasses*. Standard. Geneva, CH: International Organization for Standardization, 2014.

[25] E. R. Jellum et al. "The syncline model-analyzing the impact of time synchronization in sensor fusion". *2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2022, pp. 1446–1453.

[26] V. Jindal. History and architecture of Wireless sensor networks for ubiquitous computing. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* **7**.2 (2018), 214–217.

[27] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* **32**.5 (1976), 922–923.

[28] N. Kaempchen and K. Dietmayer. "Data synchronization strategies for multi-sensor fusion". *Proceedings of the IEEE Conference on Intelligent Transportation Systems*. Vol. 85. 1. 2003, pp. 1–9.

[29] Y. Kang, H. Yin, and C. Berger. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. *IEEE Transactions on Intelligent Vehicles* **4**.2 (2019), 171–185.

[30] K. Kim, J. Kim, and J. Kim. Robust Data Association for Multi-Object Detection in Maritime Environments Using Camera and Radar Measurements. *IEEE Robotics and Automation Letters* **6**.3 (2021), 5865–5872. DOI: 10.1109/LRA.2021.3084891.

[31] M. Kristan et al. Fast image-based obstacle detection from unmanned surface vehicles. *IEEE transactions on cybernetics* **46**.3 (2015), 641–654.

[32] H. Landau. Sampling, data transmission, and the Nyquist rate. *Proceedings of the IEEE* **55**.10 (1967), 1701–1706. DOI: 10.1109/PROC.1967.5962.

[33] S. Leutenegger et al. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research* **34**.3 (2015), 314–334. DOI: 10.1177/0278364914554813.

[34] Y. Liao, J. Xie, and A. Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**.3 (2022), 3292–3310.

[35] B. D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision". *IJCAI'81: 7th international joint conference on Artificial intelligence*. Vol. 2. 1981, pp. 674–679.

[36] Z. Lum. The Measure of MASINT-MASINT, perhaps the most comprehensive of US intelligence disciplines, is also the least understood-more through a lack of understanding than a desire for secrecy. *Journal of Electronic Defense* **21**.8 (1998), 43–48.

[37] W. Maddern et al. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research* **36**.1 (2017), 3–15.

[38] M. Mandal, L. K. Kumar, and S. K. Vipparthi. "MOR-UAV: A benchmark dataset and baselines for moving object recognition in UAV videos". *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, pp. 2626–2635.

[39] S. Mandal et al. "Motion prediction for autonomous vehicles from lyft dataset using deep learning". *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. IEEE. 2020, pp. 768–773.

[40] G. Mandlburger and B. Jutzi. On the feasibility of water surface mapping with single photon LiDAR. *ISPRS International Journal of Geo-Information* **8**.4 (2019), 188.

[41] J. A. Marvel and R. Bostelman. A cross-domain survey of metrics for modelling and evaluating collisions. *International Journal of Advanced Robotic Systems* **11**.9 (2014), 142.

[42] P. Mittal, R. Singh, and A. Sharma. Deep learning-based object detection in low-altitude UAV datasets: A survey. *Image and Vision computing* **104** (2020), 104046.

[43] S. Moosbauer et al. "A Benchmark for Deep Learning Based Object Detection in Maritime Environments". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[44] M. Mueller, N. Smith, and B. Ghanem. "A benchmark and simulator for uav tracking". *European conference on computer vision*. Springer. 2016, pp. 445–461.

[45] A. Nanda et al. KOLOMVERSE: KRISO open large-scale image dataset for object detection in the maritime universe. *arXiv preprint arXiv:2206.09885* (2022).

[46] T. Neagoe, V. Cristea, and L. Banica. "NTP versus PTP in com puter networks clock synchronization". *2006 IEEE International Symposium on Industrial Electronics*. Vol. 1. IEEE. 2006, pp. 317–362.

[48] B. Nguyen et al. "Application and evaluation of direct sparse visual odometry in marine vessels". *Proceedings of the 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles, (CAMS 2022)*. 2022.

[49] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. `https://www.openstreetmap.org`. 2017.

[50] M. T. Orchard, C. A. Bouman, et al. Color quantization of images. *IEEE transactions on signal processing* **39**.12 (1991), 2677–2690.

[51] I. M. Organization and M. O. Morska. *Solas: Consolidated Text of the International Convention for the Safety of Life at Sea, 1974, and Its Protocol of 1988, Articles, Annexes and Certificates, Incorporating All Amendments in Effect from 1 January 2020*. IMO publication. International Maritime Organization, 2020. ISBN: 9789280116908.

[52] *P-1775 IMU Photonic Inertial Measurement Unit*. DS P1775 IMU 0421. KVH Industries, Inc. 2021.

[53] L. T. Parker IV et al. "mTITAN: multi-domain tactical intelligent teaming and autonomous navigation". *Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2023*. Vol. 12544. SPIE. 2023, pp. 55–64.

[54] M. A. Pravia et al. "Generation of a fundamental data set for hard/soft information fusion". *2008 11th International Conference on Information Fusion*. IEEE. 2008, pp. 1–8.

[55] A. Reuther et al. "Survey and Benchmarking of Machine Learning Accelerators". *2019 IEEE High Performance Extreme Computing Conference (HPEC)*. 2019, pp. 1–9. DOI: `10.1109/HPEC.2019.8916327`.

[56] M. Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics* **23**.3 (1952), 470–472.

[57] N. El-Sheimy, H. Hou, and X. Niu. Analysis and modeling of inertial sensors using Allan variance. *IEEE Transactions on instrumentation and measurement* **57**.1 (2007), 140–149.

[58] J.-P. Skeete. Level 5 autonomy: The new face of disruption in road transport. *Technological Forecasting and Social Change* **134** (2018), 22–34.

[59] P. Spagnolo et al. "A new annotated dataset for boat detection and re-identification". *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. 2019, pp. 1–7.

[60] J. Stuart. "High quality digital audio". *IEE Colloquium on Digital Audio Signal Processing*. IET. 1991, pp. 111–112.

[61] H. Taheri and Z. C. Xia. SLAM; definition and evolution. *Engineering Applications of Artificial Intelligence* **97** (2021), 104032.

[62] R. F. Tomlinson. The impact of the transition from analogue to digital cartographic representation. *The American Cartographer* **15**.3 (1988), 249–262.

[63] A. Tritsarolis, Y. Kontoulis, and Y. Theodoridis. The Piraeus AIS dataset for large-scale maritime data analytics. *Data in brief* **40** (2022), 107782.

[64] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**.4 (1991), 376–380. DOI: `10.1109/34.88573`.

[65] L. Wallace et al. Development of a UAV-LiDAR system with application to forest inventory. *Remote sensing* **4**.6 (2012), 1519–1543.

[66] K. Wolsing et al. Anomaly Detection in Maritime AIS Tracks: A Review of Recent Approaches. *Journal of Marine Science and Engineering* **10**.1 (Jan. 2022), 112. ISSN: 2077-1312. DOI: `10.3390/jmse10010112`. URL: `http://dx.doi.org/10.3390/jmse10010112`.

[67] B. D. Wood and E. Taghizadeh. A primer on information processing in upscaling. *Advances in Water Resources* **146** (2020), 103760.

[68] X. Xu et al. A review of multi-sensor fusion slam systems based on 3D LIDAR. *Remote Sensing* **14**.12 (2022), 2835.

[69] X. Yang et al. "Deep learning for practical image recognition: Case study on kaggle competitions". *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 923–931.

[70] B. Yu et al. "Building the computing system for autonomous micromobility vehicles: Design constraints and architectural optimizations". *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE. 2020, pp. 1067–1081.

[71] Z. Zeng et al. Review of hybrid aerial underwater vehicle: Cross-domain mobility and transitions control. *Ocean Engineering* **248** (2022), 110840.

[72] M. M. Zhang et al. "VAIS: A dataset for recognizing maritime imagery in the visible and infrared spectrums". *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015, pp. 10–16. DOI: `10.1109/CVPRW.2015.7301291`.