



## **Autonomous GPU-based UAS for inspection of confined spaces**

Application to marine vessel classification

**Brogaard, Rune Y.; Boukas, Evangelos**

*Published in:*  
Robotics and Autonomous Systems

*Link to article, DOI:*  
[10.1016/j.robot.2023.104590](https://doi.org/10.1016/j.robot.2023.104590)

*Publication date:*  
2024

*Document Version*  
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

*Citation (APA):*  
Brogaard, R. Y., & Boukas, E. (2024). Autonomous GPU-based UAS for inspection of confined spaces: Application to marine vessel classification. *Robotics and Autonomous Systems*, 172, Article 104590. <https://doi.org/10.1016/j.robot.2023.104590>

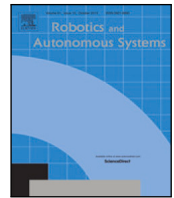
---

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



# Autonomous GPU-based UAS for inspection of confined spaces: Application to marine vessel classification<sup>☆</sup>

Rune Y. Brogaard<sup>a,b,\*</sup>, Evangelos Boukas<sup>a</sup>

<sup>a</sup> Technical University of Denmark, Kgs. Lyngby, Denmark

<sup>b</sup> FORCE Technology, Brøndby, Denmark

## ARTICLE INFO

### Keywords:

Inspection  
Absolute localization  
Machine learning  
UAV

## ABSTRACT

Inspection of confined spaces poses a series of health risks to human surveyors, and therefore a need for robotic solutions arises. In this paper, we design and demonstrate a real-time system for collecting 3D structural and visual data from a series of inspection points within a prior map of a confined space. The system consists of a GPU accelerated 3D point cloud registration and a visual inertial odometry estimate fused in an Unscented Kalman Filter. Using the state-of-the-art deep learning-based feature descriptors, FCGF –and the robust Teaser++ 3D registration algorithm– point clouds from a narrow field of view, time-of-flight, camera can be registered to a prior map of the environment, to provide accurate cm-level absolute pose estimates. The uncertainty of the system is furthermore estimated on the basis of the novel GPU-based Stein ICP algorithm. Visual defects, represented by augmented reality fiducial markers, are automatically detected during inspection, and their positions are estimated in the map frame of the confined space. The performance of the system has been evaluated in realtime onboard a small UAV, within a mock-up model of a water ballast tank from a marine vessel, where the UAV was able to navigate and inspect the ambiguous and featureless environment. All defects were estimated within  $\pm 10$  cm of their actual position.

## 1. Introduction

Inspection of dark and confined spaces can be a challenging task for both human and robotic applications [1]. Inspections of these spaces are performed mainly by humans due to the often complex structures and limitations of current robotic capabilities. The environment can often be considered dangerous due to the variability in atmospheric conditions, the risk of falling, and the often unknown integrity of the structure. Likewise, this line of work can also be tiresome and dirty, further increasing the risk of human error, such as missing critical defects. Therefore, it is important to minimize human involvement during inspection of confined and structurally complex spaces, to eliminate human risk, and to maximize repeatability of inspections. The repeatability of an automated inspection solution can also allow analysis of the trend of defects in critical areas.

For the inspection of the water ballast tanks, three modes of operation are of interest, namely (1) Mapping and exploration of the environment, (2) Visual close-up inspections, and (3) Contact-based thickness measurement of the metal structure. The authors of [2] have shown how water ballast tanks can be mapped using an exploration-based approach where unknown spaces are used as an element in the RRT

cost function. Therefore, the focus of this paper is on autonomously performing visual inspections in an already known environment. The environment can be known from either the CAD drawings of the ship or from the construction of a map as shown in [2].

The goal of this paper is to autonomously fly an inspection mission in a water ballast tank, using an unmanned aerial system as shown in Fig. 1, while collecting inspection data in the process. The proposed inspection system to solve this problem can be seen in Fig. 2, where this paper covers the entire pipeline except the data evaluation part. In-depth research on deep learning-based detection and evaluation of images collected from the UAV is covered by [3,4].

Most autonomous systems require a good localization module, and this will therefore be the backbone of our system. A large part of our proposed inspection system is dedicated to both localizing the UAV itself, but also localizing detected defects. Localizing within structured, confined, and ambiguous spaces poses a range of challenges. These spaces often contain low lighting conditions and poor visual features, therefore limiting visual tracking or recognition. Likewise, the nature of industrial man-made spaces is often built with a certain repeatable pattern in mind, challenging classical 3D geometric pattern recognition

<sup>☆</sup> This research was supported by the Inspectrone project at DTU, funded by Innovation Fund Denmark.

\* Corresponding author at: Technical University of Denmark, Kgs. Lyngby, Denmark.

E-mail addresses: [rybro@dtu.dk](mailto:rybro@dtu.dk) (R.Y. Brogaard), [evanb@dtu.dk](mailto:evanb@dtu.dk) (E. Boukas).

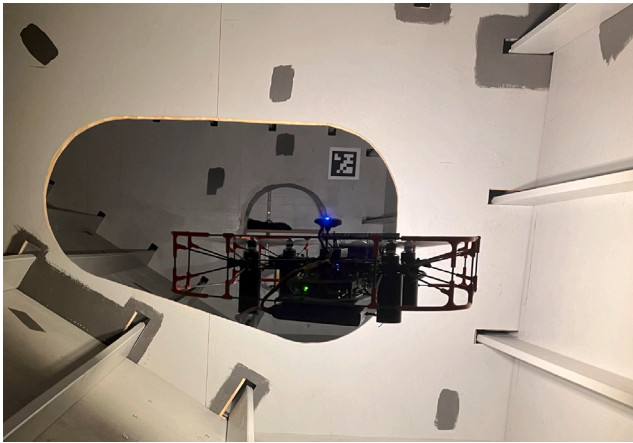


Fig. 1. The Unmanned Aerial Vehicle used in the experiments of the autonomous inspection system. A simulated defect is depicted as an AR tag in the next ballast tank compartment.

and subsequent data association. The inspection pipeline handles a range of these challenges by utilizing prior information from the CAD drawing of the industrial space, which in this case is the water ballast tanks of marine vessels. The pipeline, furthermore, uses a Kalman fused combination of Visual Inertial Odometry and 3D deep learning-based geometric features, collected, respectively, from an Intel T265 stereo camera [5] and L515 ToF camera [6].

It is important to note that the designed inspection system has the potential to be utilized on various kind of robotic platforms, where in this paper we have tested the pipeline on a small Unmanned Aerial Vehicle (UAV). Therefore, the purpose of the UAV is to visit planned inspection points inside the ballast tank and collect image data of defects. In this paper, the defects are marked using wall-mounted ArTags, at typical positions of structural stress concentration.

Part of the localization system has been tested offline in [7], whereas this paper focus on real time execution of a whole automatic inspection pipeline. This work, therefore, does not require a human in command during inspection, compared to the previous work where a human was still in control of the robot. Due to less requirements for real-time performance as well as the classification societies requirements for final decision by humans, the last part of the pipeline –i.e. the automatic evaluation of the inspection data– should still be performed offline using machine learning-based methods as described in [3,4].

The main contributions of this work are as follows.

1. The tangible deployment of Graphics Processing Units (GPUs) in conjunction with deep learning techniques for precise localization, capitalizing on 3D geometric data, that enables the autonomous operation of inspection robots within environments which lack distinctive features.
2. Proving that a faster but inferior deep geometric feature descriptor is sufficient for the accurate *online* localization of aerial inspection robots under the condition that the system includes a robust registration algorithm capable of providing accurate transformations even in the presence of high percentage of outliers.
3. The employment, for the first time, of a state-of-the art 3D registration uncertainty estimation algorithm –Stein ICP, originally designed for 3D point cloud matching– in an online GPU-based robot localization system.

### 1.1. Related work

Entering confined spaces is considered a high-risk operation for humans, and therefore the need to use robots has emerged. The most common approaches are submerged Remotely Operated Vehicles (ROV),

Magnetic Crawlers, or Unmanned Wheeled Robots, and to some extent, Aerial Systems.

Research on ROV applications, such as [8–10], has focused mainly on inspecting the exterior of the hull. This has been done by submerging the vehicle near the vessel and either manually controlling the ROV or by using wheel encoder data and to some degree visual odometry. For the confined spaces in the ballast tanks, ROVs are, however, not considered as a viable solution. The ballast tanks are rarely filled with water, and visibility of the water in the tanks is often limited due to impurities in the water such as mud and marine growth.

The authors in [11,12] have investigated the usage of crawlers and small legged robots for inspection of marine vessels. The authors of [11] have designed a magnetic track wheel robot, which is capable of autonomously building a mosaic image of a vertical inspection run on a planar surface of the cargo bay. This design is limited by a simple odometry localization system, but also by its ability to traverse obstructions in the ballast tanks such as ladders, beams, and bulkheads. Instead, a small legged robot was designed in [12], which uses electromagnets on its feet, and is able to traverse tight areas and obstacles, as long as they are magnetic. To our knowledge, they do, however, not provide a localization or navigation system for autonomous control of the robot. The authors of [13] build on a similar magnetic approach as [11], but uses an external pose estimate for reference, which is not considered possible for the complex and confined space of the water ballast tanks.

The challenges of traversing confined spaces using aerial systems have been described in [3,14], where they mention reliable absolute localization and lack of remote control as some of the major difficulties. It is therefore clear that some level of autonomy is required for the UAV. The authors of [15,16] solve the localization issue by implementing Ultra Wide Band (UWB) satellites inside the confined space to give them an absolute global coordinate system for the UAV. This proves to provide accurate results, but installing UWB electronic hardware in the water ballast tanks is in our case not considered a viable solution. Other aerial systems such as [17,18] use cameras and touch-based sensors to demonstrate the ability to traverse small man-way-sized ventilation shafts, but have not focused on collecting inspection data during their flight. The authors of [7] used point clouds from time-of-flight(ToF) cameras combined with the well-known Fast Point Feature Histograms (FPFH) [19] to perform scan to map registrations. However the FPFH descriptors performed poorly in these structured environments, compared to deep learning-based geometric feature descriptors, due to the high level of structural similarity. The authors of [20] provide a path planning solution for autonomous inspection of the outside of marine hulls, using multiple UAVs to first perform a general inspection and secondly a close-up inspection of detected defects. The system was not tested on real vessels or in more constricted environments such as confined spaces but, rather, in simulated environments. Other research [21] has focused on exploration and mapping of the environment using a combination of Simultaneous Localization and Mapping (SLAM) and Convolutional Neural Networks (CNN). This research was conducted using a similar sized UAV as ours, but was however only tested in a well lit office environment.

The authors of [22] used LIDARs, cameras, and IMU data from the onboard AscTec flight controller to navigate around the cargo hull of a large marine vessel. The UAV used a 2D lidar to estimate the distance to the surrounding walls and a 1D lidar to estimate the altitude inside the cargo hull. The system was tested within both the cargo hull and the top-side ballast tanks, but a user was still required to give high-level commands to the UAV through a base station using a joystick. The system was further expanded in [23] to use multi-threaded Binary descriptor-based Image MOSAicing (BIMOS) to create a single overview image of the inspection surface, using ORB features and Keyframes.

The research carried out in [24] implements an autonomous Unmanned Aerial Fire Detection System, for marine vessels. It can navigate inside the main areas of the ship, such as corridors and doorways, by utilizing a prior map of the environment. A particle-filter-based

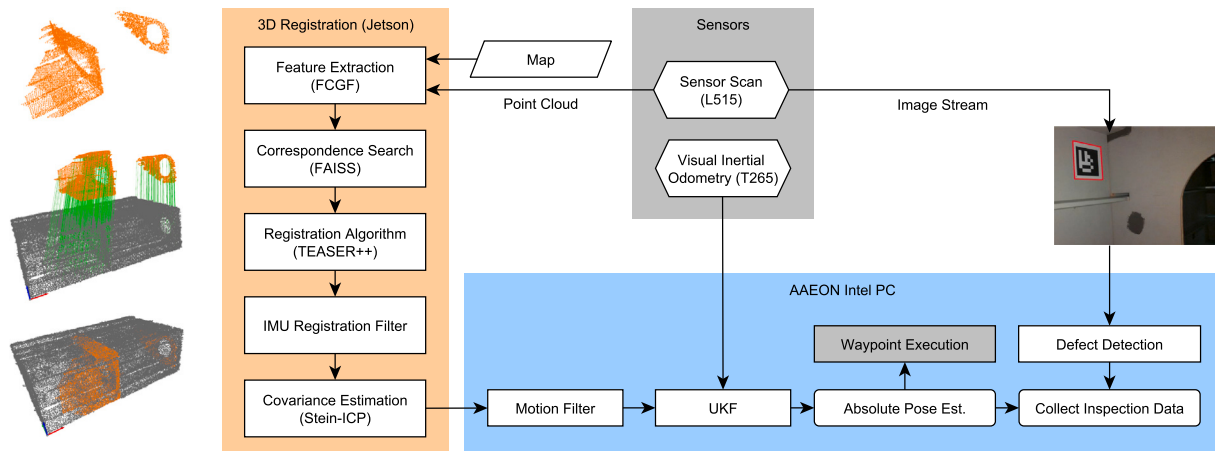


Fig. 2. Overview of the inspection system running onboard the UAV. 3D point cloud registrations between a known map and scan from the L515 sensor are fused with a VIO velocity estimate in an UKF to give an absolute pose estimate in the map frame. A defect detection system is continuously evaluating image data from the sensor to provide an estimate of defects within the map.

localization algorithm was used to localize the UAV in a generated 3D octomap. They are able to localize the fire using a thermal camera, but point to some failures and limitations due to drifting odometry and poor 3D registrations between the sensor-scan and the prior map.

In [25] the authors used a UAV to explore an underground mine environment, to locate objects of interest, which in their case were humans in need of rescue. The solution is based on the fusion of LiDAR data with thermal vision frames and inertial measurements. The system proved capable of autonomously navigating the mine environment, however, the repeatable structure and ambiguity of the ballast tanks do not allow for a simple transfer of the system to new domains.

In summary, existing research is still needed for autonomous confined space inspection of water ballast tanks, which present unique challenges that motivate our work.

## 2. System description

### 2.1. Unmanned Aerial Vehicle

To test the entire inspection navigation pipeline, an Unmanned Aerial Vehicle (UAV) is custom built to be able to enter through the man-way in the water ballast tanks. The UAV shown in Fig. 1 is based on the specifications in Table 1. It consist of a Lynxmotion Crazy2Fly Drone frame equipped with a Pixhawk 4 Mini Flight Controller Unit (FCU), an Aeeon PICO-WHU4 I5 PC, and a Nvidia Jetson Xavier NX that functions as the GPU of the system. A protection cage is installed around the UAV to protect the propellers from potential impacts with the surrounding environment. The cage of the UAV increases the footprint of the UAV to  $500 \times 500$  mm, with a height of 140 mm.

The UAV is also equipped with an Intel L515 Time-of-Flight camera and two Intel T265 VIO cameras. The front facing T265 is acting as the VIO input to the localization system, where as the down-facing camera is completely decoupled from the localization system, to only provide the ground truth estimate from a series of AR tags installed on the bottom of the tank. The UAV uses the two onboard computers to process sensor data and communicate with the onboard FCU. The main computer is the AAEON PICO-WHU4, and the second processing unit is the Nvidia Jetson NX development board. The onboard PICO PC handles the communication to the FCU and drives the data collection of the T265 and L515 cameras. The L515 point cloud is down-sampled using a voxel grid of 0.05 m, and the point cloud is then transmitted to the Nvidia Jetson over a cabled Ethernet connection. The Nvidia Jetson handles the majority of the localization pipeline by utilizing the Graphical Processing Unit. The latter system also runs the 3D registration part of the Localization pipeline, where especially the

**Table 1**  
Specifications of the UAV used throughout the experiments.

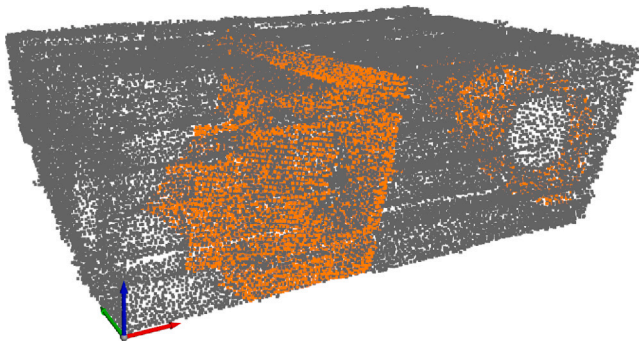
Aerial Platform	
Base frame	Lynxmotion Crazy2Fly
Protection cage	Custom design
Propellers	Quad, 9", (HQPROP 9 × 4.5)
Dimensions	H: 140 mm, W: 500 mm, L: 500 mm
Weight	2.5 kg including a 4 s battery
FCU	Pixhawk 4 Mini, Px4 v.1.10 Firmware
Onboard Sensors	
VIO	Intel T265 camera (Front-facing)
Point cloud	Intel L515 ToF camera
Inspection cam.	Intel L515 color camera
Ground truth	Intel T265 camera (Downward-facing)
Onboard Compute units	
Onboard PC	AAEON PICO-WHU4 (Intel I5, 8th gen)
GPU	Nvidia Jetson NX (16 Gb memory)

correspondence search and feature descriptors benefit from the parallel computation power. The low-level control of the UAV is done by the FCU Pixhawk 4 Mini [26], running the PX4 1.10 software stack, where the communication to the AAEON PC is carried out by wired RS232 connection. The PC sends 20 Hz pose estimates and waypoints to the FCU, which then ensures the robot is on the commanded position with an acceptance threshold set to  $\pm 10$  cm. The PICO and the Jetson run Ubuntu 18.04 with ROS Melodic [27], where the PICO PC is running as the ROS master and the Nvidia Jetson as a secondary device, with a wired Ethernet connection. The defects, represented as April Tags, are detected using the ROS package AprilTag 3 visual fiducial detector [28] on the PICO PC using the RGB image stream from the L515. Using the camera matrix, the 2D April tag is transformed into the 3D position of the camera frame and then transformed into the map frame using the absolute pose estimate from the localization pipeline. It is important to note that no April tags are used for the actual localization part of our system.

## 2.2. Environment

The proposed system has the potential to operate in a wide range of confined spaces, but the focus of this paper has been to verify the system in a water ballast tank of a marine vessel. To enable rapid development and testing –and due to temporary access restrictions to vessels– a mock-up model of a topside ballast tank has been used for the experiments. A map of the ballast tank is represented as the gray point cloud shown in Fig. 3, where as a photo of the inside of the tank

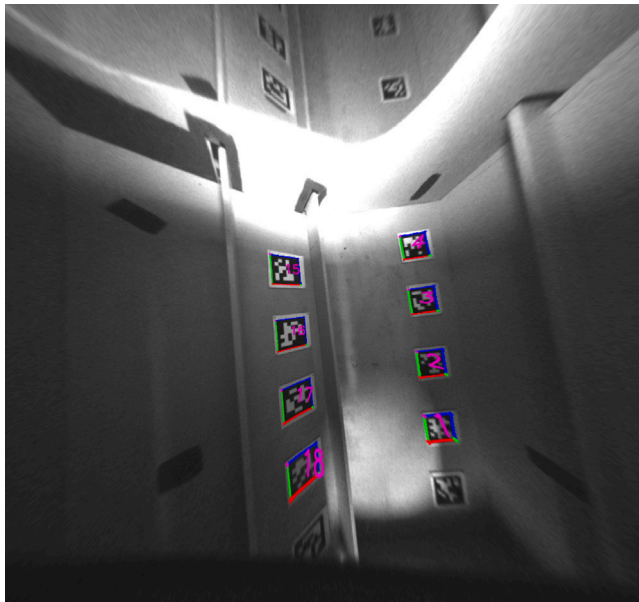




**Fig. 3.** Illustration of the mock-up model of the water ballast tank, shown as the gray point cloud. An example of an aligned sensor scan is depicted with orange points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

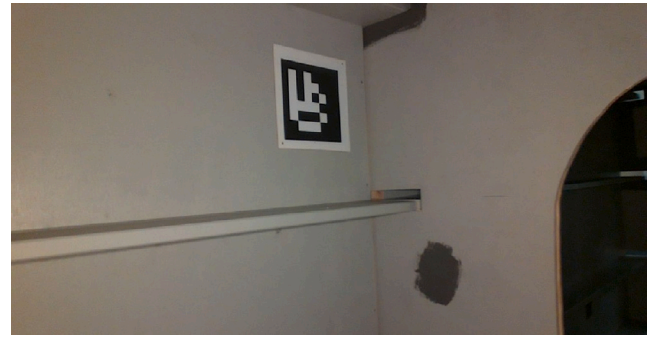


**Fig. 4.** Photo of the interior of the mock-up model of the water ballast tank, without the mounted ground truth tags and defects.



**Fig. 5.** Undistorted image with the floor mounted ground truth tags visible, captured from the down-facing T265 camera. The tags are only used for the ground truth estimation and not for the actual localization pipeline.

is captured in Fig. 4. The ballast tank consists of two compartments with a bulkhead/webframe as the divider. A registered scan from the



**Fig. 6.** Example of an April Tag defect detected by the Camera.

L515 is shown as the orange point cloud in the figure. A simplified top-down view of the ballast tank is also depicted with black bold lines in Fig. 10. The bounding box of the ballast tank is  $5.2 \times 3.0 \times 1.8$  meter. To provide a reasonable ground truth estimate, a series of April Tags are mounted on the floor of the ballast tank, as shown in Fig. 5. The positions of these tags were first manually mapped and then used in the localization part of the TagSLAM package [29], to provide the ground truth pose of the UAV.

The defects the UAV needs to detect and localize are represented by 6 April Tags mounted on the walls within the tank, and in the field of view of the camera at the specified inspection points, as shown in Fig. 6. Inspection points are defined based on known areas of the ship that must be inspected according to the shipping regulations. These inspection points are often areas that are prone to deterioration or failure, due to the high material stresses within the structure of the ship.

### 3. Inspection pipeline

To autonomously perform the inspection, the system is designed as shown in Fig. 2. During startup of the UAV, a list of inspection waypoints is loaded onto the onboard PC and a point cloud map of the environment, with its extracted geometric features, is loaded into the GPU memory of the Nvidia Jetson.

#### 3.1. Inspection waypoint execution

The map of the environment is provided based on the CAD drawings of the ship and an inspection path can be generated for the UAV based on inspection points. For older vessels where the CAD drawings might not be readily available, a map of the environment could be obtained manually from the 2D drawings or by either using 3D surveying (large 3D scanners) or post processing (Full Bundle Adjustment and Dense 3D reconstruction) from currently manually operated UAVs. A lower-resolution map could also furthermore be obtained from UAVs equipped with 3D spinning LiDARs combined with methods such as [30–32]. Inspection points, or areas, are defined in the inspection specifications of the classification societies [33], and are often based on areas of high stress concentration or exposure to buckling, resulting in an increased risk of corrosion. The generated route from the inspection points is, thereafter, executed by the UAV using the Pixhawk FCU using the localization pipeline of Section 3.2 as its absolute pose estimate. During the inspection execution, the UAV will arrive at a series of inspection points, where along its path it will look for defects and collect visual data in the form of RGB images and surface point clouds for any detected defects. The RGB images and surface point clouds will later serve as the evaluation and decision base for onshore inspection personnel. While not included in this paper, for the sake of conciseness, we have in the past automated the evaluation of corrosion or defects using different machine learning methods as investigated by [3,4].

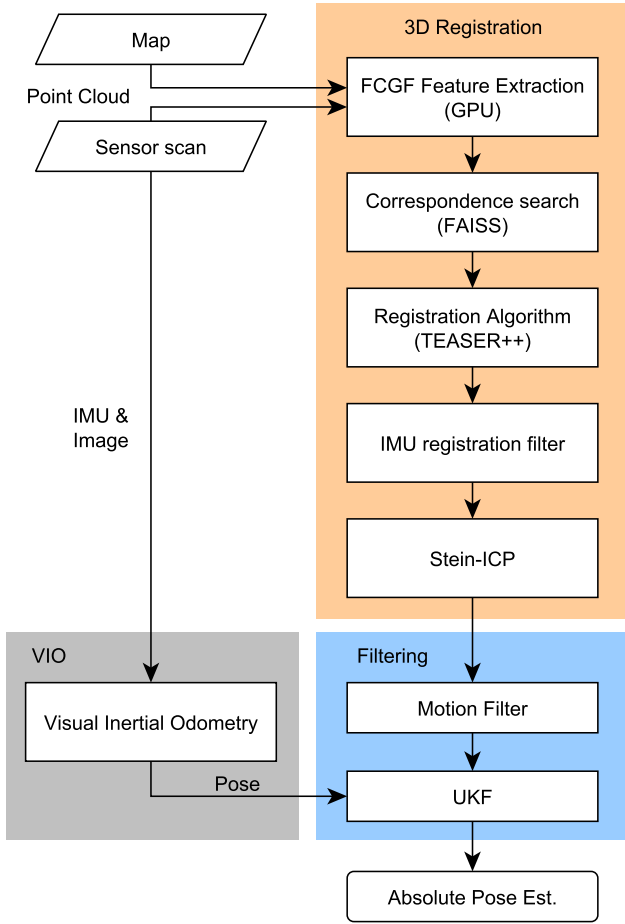


Fig. 7. Localization pipeline.

### 3.2. Localization pipeline

A major part of an autonomous inspection system is its localization pipeline, as illustrated in Fig. 7. The pipeline can be divided in three sections; a 3D registration module, Visual Inertial Odometry (VIO), and lastly a filtering step. The 3D registration module generates an absolute pose estimate by registering the sensor scan (point cloud) to the map of the environment. The VIO provides a relative odometry estimate, where in our case the velocity is used as a state update in our filtering step. The filtering module of the pipeline first removes poor absolute pose estimates from the 3D registration based on a motion filtering threshold, and then fuses the inlier pose estimates with the velocity estimate from the VIO, in an Unscented Kalman Filter (UKF).

#### 3.2.1. Localization based on 3D registration

Due to the fact that the intended environments (i.e. ballast tanks of marine vessels) for this system are visually featureless, the localization pipeline in this work relies on 3D registration of –live acquired– 3D scans with pre-existing 3D models. While pair-wise registration is usually employed for 3D point cloud registration tasks [34], the algorithms proposed in the literature, mainly variants of ICP, are not suitable for implementation on an inspection UAV, due to the ICP’s requirement of approximate prior knowledge of the model/target pose, the problem of local minima and due to its inability of providing uncertainty estimates (see Section 3.2.4) which is a must have for appropriate state estimation. Therefore, we follow the approach of feature based 3D registration and specifically we employ deep learning feature extraction. Then we utilize a robust registration system, which is designed to accurately provide correspondences even in the presence of significantly more

outliers than inliers. Finally, since the output of the 3D registration is used for state estimation, we integrated a state of the art method for uncertainty estimation of 3D pointcloud registrations, allowing us to overcome a common problem of the 3D registration, i.e. uncertainty underestimation [35].

**3.2.1.1 Feature extraction and data association** The first step in the 3D registration module of the localization pipeline is to extract geometric features both in the map and in the sensor scan. Hand-crafted features have been very successful in providing some specific qualities, e.g. rotational invariance. A typical example is the FPFH [19] algorithm (and its predecessor PFH) which use a histogram of surrounding 3D points to calculate correspondences invariant to changes in orientation. We have tested such approaches in [7] and it was concluded that they do not perform well and are therefore not included in this investigation.

To overcome the tradeoff between these different handcrafted features, we can employ learned features which, despite being less explainable, incorporate multiple qualities present in several custom designed features. Two such approaches are presented in this work, SmoothNet [36] which can achieve high accuracy but is computationally expensive –therefore used as a baseline– and FCGF [37] which is fit for online execution, albeit with less accuracy. We will briefly present both of the approaches for completeness.

SmoothNet is a deep learning-based descriptor with a focus on robustness as well as rotational and isometry invariance. Rather than implementing an end to end trainable network it has two distinct parts; (a) the “preprocessing” of a pointcloud using Local Reference frames (LRF) and smoothed density value (SDV) representation and (b) the fully convolutional part, which encodes a descriptor. To achieve rotational invariance, the neighborhoods of randomly selected points in the pointclouds are structured in SDVs with LRFs defined by the neighborhoods themselves. Finally, the compact representation (3D SDV voxel grid) is fed into a siamese network to produce a descriptor.

In terms of the specific implementation of Smoothnet as a baseline for our system, the interest points were selected randomly — approximately 50% of all points in the map and 25% of the sensor point cloud. By randomly selecting points in both clouds, we limit the computational load while maintaining a high probability of acquiring correspondences.

As proven in our previous work on pre-recorded datasets [7,38], Smoothnet is able to provide superior results in our application domain (WBT of large vessels) which however comes at a significant computational cost. The cost is such that the approach could not be used for the online localization of our robot; a fully autonomous, self sufficient, power and time limited inspection robot in real-life. While there is a fair argument to be made about the usage of Smoothnet on larger robots (e.g. a tethered aerial robot), we found that the method used in our robot, i.e. FCGF, is an order of magnitude more computationally efficient and that with the appropriate outlier rejection and robust estimation the system can provide adequate results.

The computational efficiency of FCGF stems from its structure, which is a one-pass 3D fully convolutional network, relying on sparse tensors and sparse convolutions. Additionally, FCGF does not rely on low-level pre-processing of the input point clouds. Technically, the FCGF is a ResUNet (or Deep Residual UNET) architecture that uses skipped connections and residual blocks to extract fully convolutional descriptor features. By replacing the bridge of the UNET with residual blocks, the network is minimal in terms of parameters.

The most important contribution of the FCGF is the usage of “hardest-contrastive” losses. The contrastive loss is defined as follows: Similar features –or positive– should be as close as possible in the output feature dimension, and dissimilar features –or negative– must be at least a margin away. Instead of accumulating pairs of features (either positive or negative), the idea of hardest contrastive loss is to structure quadruplets using a positive pair with their “hardest” (or closer) negatives. This procedure is called hard negative mining. In the

case of triplets (positive–positive–negative) rather than pairs, “hardest-triplet” losses can be structured. In our paper the network is setup to use standard metric learning losses based on “hardest-contrastive” losses.

The extracted features (whether Smoothnet or FCGF) are then searched for correspondences to features in the map using the GPU based KNN algorithm *Facebook AI Similarity Search* (FAISS) [39].

### 3.2.1.2. Registration

The extracted and associated geometric features from the scan and the map, are prone to a notable number of outliers due to the ambiguity of the environment. We, therefore, employ the robust 3D registration algorithm TEASER++ [40,41], which allow for outlier rejection. TEASER++ is able to provide an accurate transformation between two corresponding point sets, even in the presence of a high percentage of outliers. The following brief description of the registration algorithm was also described in [7], where the equations are largely based on the work from the authors of TEASER++ [40].

For ideal cases, where the 3D-point correspondence list contains zero outliers, the 3D registration can be defined as the following nonlinear least square solution:

$$\min_{s>0, \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \frac{1}{\sigma_i^2} \|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2, \quad (1)$$

where minimization is performed over the scale  $s$ , the rotation  $\mathbf{R}$ , and the translation  $\mathbf{t}$ .

For our case with the ballast tanks, we can assume a metric environment with no significant scale changes between the scan and the map, and the scale factor  $s$  can therefore be fixed to 1. The total number of correspondence points are denoted as  $N$ , and  $\mathbf{a}$  and  $\mathbf{b}$  represent two vectors of the corresponding pairs between the map and the sensor point cloud.

To include robustness of measurement noise in point clouds, a Gaussian noise with isotropic covariance described by  $\sigma^2$  is included. So far we have assumed correct corresponding points, however, for most real-world cases, correspondences with zero outliers cannot be safely assumed [40] and therefore robust registration can be performed using a Truncated Least Squares function as stated by Eq. (2):

$$\min_{s>0, \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3} \sum_{i=1}^N \min \left( \frac{1}{\beta_i^2} \|\mathbf{b}_i - s\mathbf{R}\mathbf{a}_i - \mathbf{t}\|^2, \bar{c}^2 \right) \quad (2)$$

Eq. (2) provides a least squares solution to measurements with minimal residuals (no greater than  $\bar{c}^2$ ), where  $\beta_i$  represents a given limit for the noise. This noise limit is either set as the maximum error allowed for an inlier or 3 standard deviations. Any measurements with large residuals (more than  $\bar{c}^2$ ) are disregarded. In the experimental setup,  $\bar{c}$  is set to 1. In order to simplify the solution, TEASER++ separates rotation and translation, as shown by Eqs. (3) and (4) respectively.

$$\hat{\mathbf{R}} = \arg \min_{\mathbf{R} \in SO(3)} \sum_{i,j=1}^N \min \left( \frac{\|\mathbf{b}_{ij} - \mathbf{R}\mathbf{a}_{ij}\|}{\delta_{ij}^2}, \bar{c}^2 \right) \quad (3)$$

To determine the estimated rotation  $\hat{\mathbf{R}}$ , the first step is to minimize the distance between corresponding points, expressed as  $\mathbf{b}_{ij}$  and  $\mathbf{a}_{ij}$ , with a defined noise limit of  $\delta_{ij}$ . Once the rotation is estimated, the translation can be calculated using Eq. (4). The translation is obtained on a component-wise basis, meaning the entries  $t_1, t_2, t_3$  of  $\mathbf{t}$  are calculated individually. Similar to the previous equations, any measurements with large residuals (more than  $\bar{c}^2$ ) are discarded.

$$\hat{t}_j = \arg \min_{t_j} \sum_{i=1}^N \min \left( \frac{(t_j - [\mathbf{b}_i - \hat{\mathbf{R}}\mathbf{a}_i]_j)^2}{\beta_i^2}, \bar{c}^2 \right) \quad (4)$$

An analytic derivation of the aforementioned formulation can be found in [40].

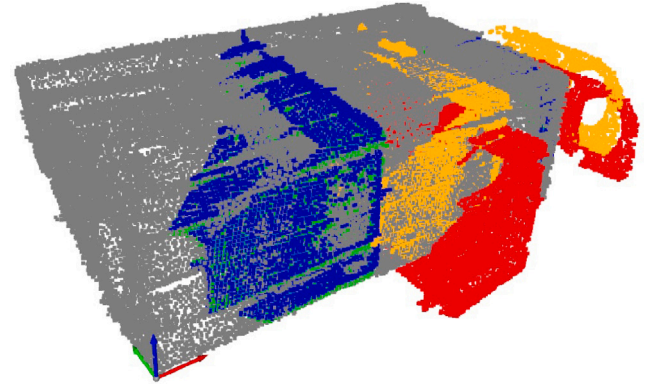


Fig. 8. Stein vs ICP point cloud registration. Red is the raw scan point cloud, green is the ground truth, yellow is p2p-ICP and blue is the Stein ICP pipeline. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The transformation from the global TEASER registration is then compared with the roll and pitch attitude of the UAV. If the registration is offset by a given threshold, the registration is rejected, and the 3D registration pipeline is rerun with a new point-cloud scan from the sensor. If the registration is within a  $\pm 5$  degree limit of the measured roll and pitch angle of the UAV, the uncertainty of the registration can then be calculated, along with a refinement step of the registration.

### 3.2.1.3. Uncertainty estimation

To estimate the uncertainty, different methods are available with different properties. The uncertainty of the final 3D registration can be estimated using efficient closed form solutions, such as CELLO [42,43], which provides a computationally fast solution, but sensor noise is not adequately captured, and the method has been shown in experimental trials by [44] to underestimate the covariance. The authors of [44] improved the work of [43], but the results still showed an underestimate of the covariance. Stein-ICP [35] can provide a better covariance estimation than previous methods, using a sampling-based approach but, therefore, also has a lower computational efficiency than closed form solutions. However, Stein-ICP can be parallelized to run on a GPU for optimal time efficiency. Stein-ICP proved to provide good covariance estimations and is therefore used as the method of choice for this paper.

Stein-ICP initializes a set of  $K$  randomly generated particles within some set of 6D pose boundaries. For the use case with the UAV we can limit the initial distribution of the points on the roll and pitch axis, since we know these from the IMU of the onboard FCU. Using an accurate IMU, one could also exclude these two degrees of freedom; however, due to some uncertainty and noise in the IMU measurements on the UAV, we still include the roll and pitch degrees of freedom in our experiments. Each particle represents the transformation of sampled points from a source point cloud, which in our case is the sensor scan. For all points in each transformed batch, the corresponding closest point in a reference cloud (map) is determined based on the nearest neighboring points in the 3D space. Mean gradients are then estimated for all matching pairs belonging to each  $K$  particles. Next, the Stein variational gradients [35,45] are obtained independently for translations and rotations, which, in turn, are then used to update each particle. This estimation is repeated for  $T$  iterations, producing an adjusted set of  $K$  particles that represent the posterior distribution. The distribution of the  $K$  particles can be used as a discrete estimation of the uncertainty and ambiguity of the environment for each degree of freedom in the registration. Using the Kernel Density Estimation (KDE) method on the distribution, the best registration can be selected. An inherent feature of Stein-ICP is that it can also apply a small refinement of the registration, which is utilized to obtain a more accurate local registration than the one provided by the global TEASER registration algorithm.



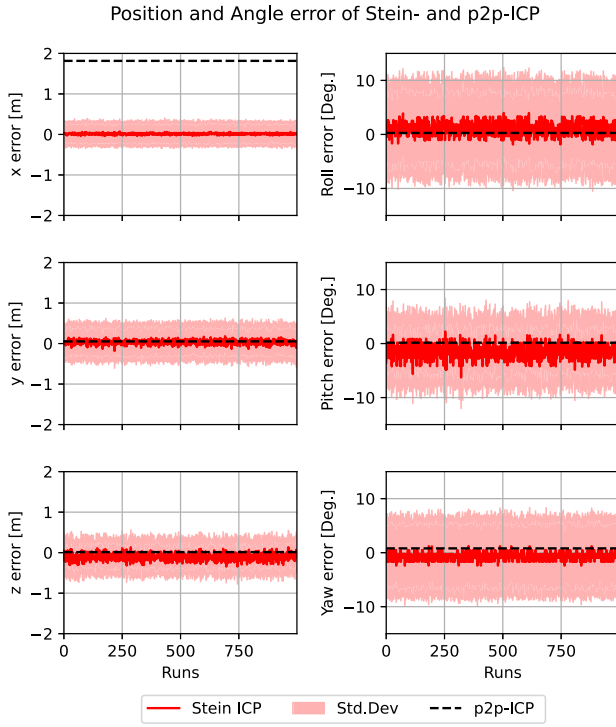


Fig. 9. The position and angle error of p2p-ICP registration (deterministic) compared to statistics (mean and STD) of stein ICP (1000 runs).

#### 3.2.1.4. Registration quality

After the uncertainty and refinement step have been performed, the quality of the registration is calculated. The quality estimation is based on the overlapping points between the map and the transformed sensor scan, where overlaps are considered if points are within the resolution of the voxel down sampling of the point clouds. The threshold for the experiments is set to reject registrations below 75% of overlapping points. This allows for some inconsistency between the map and the scan point cloud, which could be due to areas with severe buckling or structural revisions not applied to the CAD drawings of the vessel.

#### 3.2.2. Visual inertial odometry

To aid the localization system with pose updates between absolute pose estimates, a relative pose system is introduced, namely Visual Inertial Odometry. The system in the experiments uses the front facing Intel T265 camera, with a proprietary VIO algorithm, which provides a pose and velocity message. For our absolute localization system we use the velocity message as the input to our Extended Kalman Filter.

#### 3.2.3. Filtering

Outliers are removed from the absolute pose estimate using a motion filter that is based on a maximum velocity threshold of the pose estimate compared to the previous pose update. The filtered pose and covariance matrix is then used in an UKF [46,47] along with the VIO pose and covariance matrix. The UKF is a non-linear filtering algorithm, and is well described in [46], and thus only a high-level explanation will be given here. It is based on the unscented transform (UT) technique for propagating a mean and its covariance through a nonlinear transformation, and the UKF is proposed as an improvement to the well-known linear EKF. Instead of the linearization required by the EKF, the UT approximate method is used in the UKF. A set of weighted sigma points are chosen based on the mean and covariance of a prior state. Each of these points are transformed to a new state using the non linear function in the UKF. The predicted mean and covariance are then calculated on the basis of the newly transformed

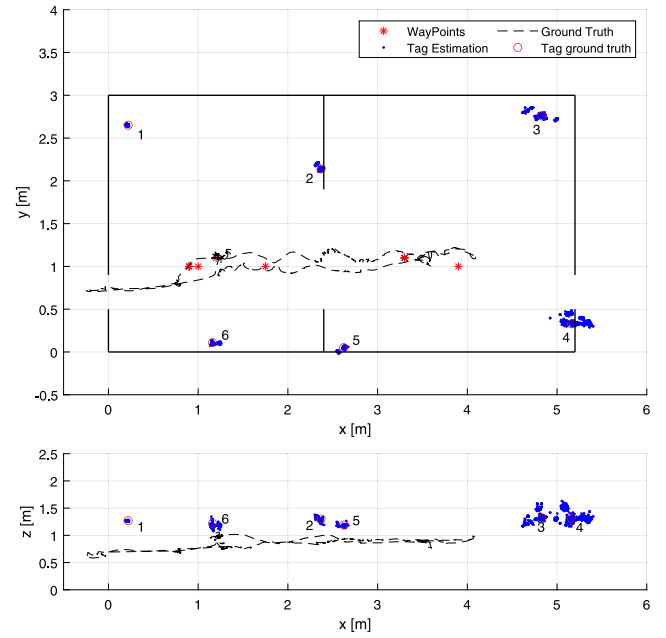


Fig. 10. Top down view of the trajectory of a single inspection run in the mock up model of the water ballast tank. The thick black lines indicate the walls of the tank. The dotted black line is the actual trajectory of the UAV. The blue points indicate the estimated position of defects and the red circles their actual position. The red stars mark the way-points the UAV has to visit on its inspection run. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

points. The added value of UKF compared to EKF is that it better represents non-Gaussian noise in the system [48].

#### 3.2.4. Stein-ICP vs P2P-ICP

Fig. 8 illustrates a point cloud registration for a single scan using standard Point-2-Point (P2P) ICP [49,50] and our 3D registration part of the pipeline. P2P-ICP settles at a local minima, with the y- and z-axis being correct, but with an incorrect registration of 1.8 meters on the x-axis, as can be seen in Fig. 9. With the current settings of Stein-ICP, p2p-ICP is more than 100 times faster than Stein-ICP, however the structural ambiguity of our environment results in p2p-icp settling into the incorrect local minima. Besides ICP's large incorrect registration on the x-axis, ICP also fails to provide a covariance or uncertainty estimate of the registration, which the main benefit of using Stein-ICP in this paper. Based on our experience, another benefit of Stein ICP vs ICP is its ability to additionally provide refinement in slightly erroneous registrations provided by TEASER++.

## 4. Experimental results

Due to limited access to real ships during the COVID-19 pandemic, experiments were carried out in a mock-up model of a water ballast tank. A series of flights with different inspection points and paths within the mock-up model were carried out. The average flight time of each inspection mission was 2 min, 15 s with a standard deviation of 9 s. For ease of reading, a single flight is described in this section.

The path of an inspection flight can be seen as a top view and side view in Fig. 10, where the UAV enters the ballast tank through the man-way in the wall on the left. Hereafter, the inspection mission begins, with its first waypoint being (1,1,1). The UAV then visits the predefined interest points, marked as red stars, looking for defects at the inspection point itself and along its path, shown as the dotted black line. Fig. 10 also shows the estimation of the tags/defects illustrated as the blue points, where the equivalent ground truth is shown as a red



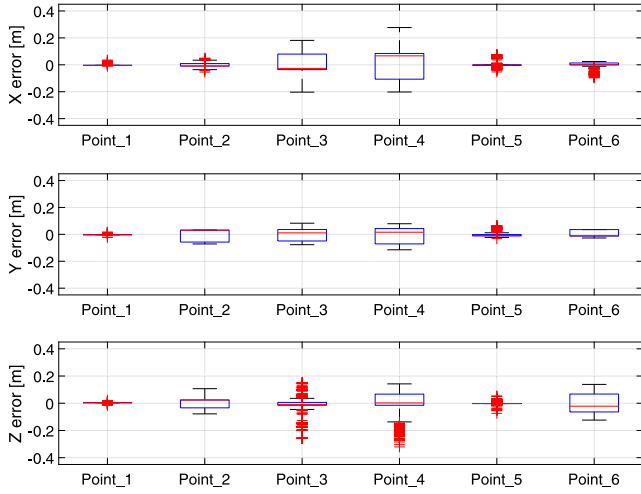


Fig. 11. Box plot of the position error for each of the 6 defects located in the tank.

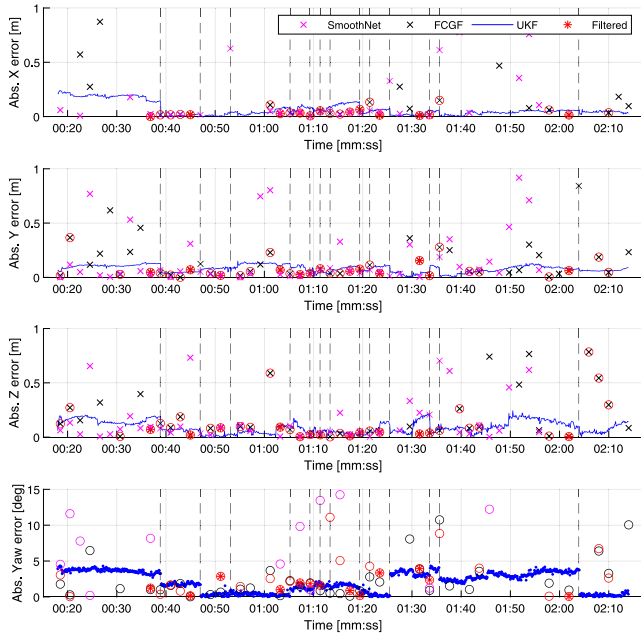


Fig. 12. Error of localization pipeline. Vertical dotted lines indicate an absolute pose update with a correction to the position of the robot.

circle. As can be seen in the figure, some position error in the estimate of the defect location is still present, where this error is shown as a box plot for each of the 6 defects in Fig. 11. All the defects were on average estimated within  $\pm 10$  cm of their actual 3D position. From Fig. 11 it can also be seen that the errors are generally low, but points 3 and 4 have a larger variation on the  $x$ -axis than the rest. This is caused by both an imprecise and varying timestamp provided by the L515, and the points being in the FOV even at large distances, which in turn exaggerate small camera calibration errors. An increasing error also arises during long time intervals between absolute updates from the localization system. The position of the specific points can be seen in Fig. 10.

The error of the absolute localization pipeline can be seen in Fig. 12, where the blue line indicates the output of the entire pipeline. The 3D registration estimate using FCGF, Teaser++ and Stein-ICP is shown as red stars, and depicts the absolute error of the pure 3D registration element in the pipeline. All 3D registrations are estimated every 2 s

Table 2

Mean and standard deviation of absolute pose estimations using different feature descriptors. Filtered depicts the results from the UKF fusion of FCGF+Stein and VIO.

Mean [m]	3DSmoothNet	FCGF	FCGF+Stein	Filtered
x	-0.67	-1.09	-0.60	-0.24
y	-0.08	0.13	0.13	0.04
z	-0.08	-0.01	-0.06	-0.02
Std. Dev [m]				
x	1.43	1.86	1.72	0.80
y	0.47	0.69	0.51	0.04
z	0.70	1.03	0.21	0.04

(0.5 Hz), whereas the UKF, which fuses the VIO with the 3D registration pose estimate, runs at 30 Hz. The intermittent output from the Teaser++ registration algorithm, in Fig. 7, using either SmoothNet or FCGF as its feature descriptors, are shown as the purple and black crosses, respectively. The SmoothNet feature descriptor was unable to run reliably in real time on the Jetson NX, and was therefore subsequently added using a ROS-bag file on a desktop PC. The FCGF feature descriptor was used in the pipeline during the flight as shown in the figure. The results from [7,38] however indicate that SmoothNet would be slower than FCGF by more than a factor of 10 if running on the Jetson. The computational time for each absolute pose update is on average 1.85 s. The mean accuracy and standard deviation of each feature descriptor, together with the output of the filtered output pose, can be seen in Table 2. From the table it can also be seen that Stein-ICP corrects some of the  $x$ -position error from the raw FCGF-Teaser++ output, as was also shown in Section 3.2.4.

The vertical dashed lines indicate an absolute pose update of the UKF from the 3D registration part of the pipeline. The varying intervals between the lines is due to the rejection of low-quality 3D registrations based on the overlapping points between the map and scan.

To estimate the ground truth of the UAV, a series of closely spaced April Tags are mounted at known locations on the floor of the ballast tank. The ground-truth tags are only mounted on the floor in order to avoid providing artificial visual feature points for the forward looking VIO camera. Using the image stream from the downward-facing T265 camera, the localization part of the TagSLAM package [29], can be used to provide a ground-truth estimate. The downward-facing camera data and the derived ground-truth estimate are not used in the localization pipeline but only serve as a method for comparing the performance of our absolute localization estimate. Due to the limited space inside the ballast tanks, a motion camera system such as VICON was not deemed a viable solution as a ground-truth estimate.

To represent the defects within the tank, six AR tags are mounted on the walls. Tags are recognized using the image stream from the L515 color camera, which serves as the inspection camera of the UAV. When a tag/defect is detected, the inspection system will save a point cloud and image of the defect and estimate the defects absolute position in the map frame.

## 5. Conclusions

In this paper an autonomous system was designed to inspect known confined spaces and tested within a mock-up model of a water ballast tank. Our method supplemented a standard Visual Inertial Odometry (VIO) with state-of-the-art deep learning-based feature descriptors on 3D point cloud scans acquired using a time-of-flight camera. By registering scans from the camera to a prior map of the environment, our system was able to accurately localize itself, as well as simulated defects within the confined and dark environment, with an accuracy of  $\pm 10$  cm. Specifically, the system achieves this level of accuracy by utilizing the cutting-edge GPU-accelerated FCGF feature descriptor, as well as the GPU-based correspondence search (FAISS) and the fast Teaser++ registration algorithm. This combination proves capable

of ensuring accurate online localization for aerial inspection robots, even when confronted with a high percentage of outliers. A UKF was furthermore implemented by fusing VIO with the 3D registration to provide a fast and accurate absolute pose in the map frame. Another crucial aspect of our work was the use of the GPU-accelerated Stein ICP algorithm, which was able to provide a UKF with the required 6-dof uncertainty estimate for the highly ambiguous and structured environment. A simple inspection path was created, based on structural interest points within the structure, as the foundation for the waypoint execution system. The entire system was tested in real time on a small UAV platform without the need for external human input during the inspection execution. Using the 3D registration and Visual Inertial Odometry, the UAV was able to accurately navigate and inspect the ambiguous and featureless environment of the water ballast tanks and, moreover, locate and estimate the position of the installed defects, within  $\pm 10$  cm. The position of the defects were accurately reported within the map frame to be used for further evaluation by the human surveyor. Additionally, this work proves via experimental results that an inferior but significantly more computationally efficient 3D description algorithm (FCGF) when paired with robust outlier rejection (Teaser++) and accurate uncertainty estimation (Stein-ICP) can replace a better performing 3D description algorithm (3DSmoothnet), allowing for online execution on light-weight UAVs.

Future work on the system includes the retraining of the feature descriptor on large real-world datasets that are domain specific. Additional work could include the employment of the 3D processing presented in this work as a 3D deformation detection to further assist the classification of vessels.

#### CRediT authorship contribution statement

**Rune Y. Brogaard:** Conceptualization, Methodology, Software.  
**Evangelos Boukas:** Data curation, Writing – original draft.

#### Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rune Y. Brogaard reports financial support was provided by Force Technology.

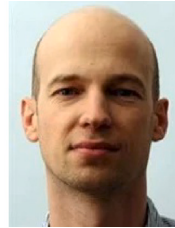
#### Data availability

The authors do not have permission to share data.

#### References

- [1] R. Andersen, R. Brogaard, E. Boukas, Autonomous robotic inspection for remote inspection technique systems: A review, *Field Robotics* 3 (1) (2023) 69–96.
- [2] R.Y. Brogaard, R.E. Andersen, L. Kovac, M. Zajackowski, E. Boukas, Towards an autonomous, visual inspection-aware 3D exploration and mapping system for water ballast tanks of marine vessels, in: *International Conference on Imaging Systems and Techniques, IST, IEEE, 2021*, pp. 1–6, <http://dx.doi.org/10.1109/IST50367.2021.9651476>.
- [3] R. Andersen, L. Nalpanitidis, O. Ravn, E. Boukas, Investigating deep learning architectures towards autonomous inspection for marine classification, in: *International Symposium on Safety, Security, and Rescue Robotics, SSR, IEEE, 2020*, pp. 197–204.
- [4] R.E. Andersen, L. Nalpanitidis, E. Boukas, Vessel classification using a regression neural network approach, in: *International Conference on Intelligent Robots and Systems, IROS, IEEE/RSJ, 2021*, pp. 4480–4486.
- [5] P. Schmidt, J. Scaife, M. Harville, S. Liman, A. Ahmed, Intel® RealSense™ tracking camera T265 and intel® RealSense™ depth camera D435-tracking and depth, *Real Sense* (2019).
- [6] A. Breitbarth, C. Hake, G. Notni, Measurement accuracy and practical assessment of the lidar camera intel RealSense L515, in: *Optical Measurement Systems for Industrial Inspection XII, Vol. 11782, SPIE, 2021*, pp. 218–229.
- [7] R.Y. Brogaard, O. Ravn, E. Boukas, GPU-accelerated localization in confined spaces using deep geometric features, in: *International Conference on Imaging Systems and Techniques, IST, IEEE, 2021*, pp. 1–6, <http://dx.doi.org/10.1109/IST50367.2021.9651425>.
- [8] F.S. Hover, R. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, J. Leonard, Advanced perception, navigation and planning for autonomous in-water ship hull inspection, *Int. J. Robot. Res.* 31 (12) (2012) 1445–1464.
- [9] C. Juan, C. William, M. David, A. George, Delgado, G. Delgado, D. Vladimir, Automatic ship hull inspection using fuzzy logic, in: *Applied Imagery Pattern Recognition Workshop, AIPR, IEEE, 2012*, pp. 1–5.
- [10] V. Kostenko, A.Y. Bykanova, A.Y. Tolstogonov, Underwater robotics complex for inspection and laser cleaning of ships from biofouling, in: *IOP Conference Series: Earth and Environmental Science, No. 2, IOP Publishing, 2019*, 022103.
- [11] A. Milella, R. Maglietta, M. Caccia, G. Bruzzone, Robotic inspection of ship hull surfaces using a magnetic crawler and a monocular camera, *Sensor Rev.* (2017).
- [12] T. Bandyopadhyay, R. Steindl, F. Talbot, N. Kottege, R. Dungavell, B. Wood, J. Barker, K. Hoehn, A. Elfes, Magneto: A versatile multi-limbed inspection robot, in: *International Conference on Intelligent Robots and Systems, IROS, IEEE/RSJ, 2018*, pp. 2253–2260.
- [13] M. Eich, T. Vögele, Design and control of a lightweight magnetic climbing robot for vessel inspection, in: *19th Mediterranean Conference on Control Automation, MED, 2011*, pp. 1200–1205.
- [14] D. Floreano, R.J. Wood, Science, technology and the future of small autonomous drones, *Nature* 521 (7553) (2015) 460–466.
- [15] B. Yang, E. Yang, L. Yu, A. Loeliger, High-precision UWB-based localisation for UAV in extremely confined environments, *IEEE Sens. J.* 22 (1) (2021) 1020–1029.
- [16] J. Tiemann, A. Ramsey, C. Wietfeld, Enhanced UAV indoor navigation through SLAM-augmented UWB localization, in: *International Conference on Communications Workshops, ICC Workshops, IEEE, 2018*, pp. 1–6.
- [17] P. De Petris, H. Nguyen, T. Dang, F. Mascarich, K. Alexis, Collision-tolerant autonomous navigation through manhole-sized confined environments, in: *International Symposium on Safety, Security, and Rescue Robotics, SSR, IEEE, 2020*, pp. 84–89, <http://dx.doi.org/10.1109/SSRR50563.2020.9292583>.
- [18] P. De Petris, H. Nguyen, M. Kulkarni, F. Mascarich, K. Alexis, Resilient collision-tolerant navigation in confined environments, in: *International Conference on Robotics and Automation, ICRA, IEEE, 2021*, pp. 2286–2292.
- [19] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, in: *International Conference on Robotics and Automation, ICRA, IEEE, 2009*, pp. 3212–3217.
- [20] O. Idir, A. Renzaglia, Multi-robot weighted coverage path planning: a solution based on the DARP algorithm, in: *2022 17th International Conference on Control, Automation, Robotics and Vision, ICARCV, IEEE, 2022*, pp. 98–104.
- [21] A. Steenbeek, F. Nex, CNN-based dense monocular visual SLAM for real-time UAV exploration in emergency conditions, *Drones* 6 (3) (2022) 79.
- [22] A. Ortiz, F. Bonnin-Pascual, E. García-Fidalgo, J.P. Company, Visual inspection of vessels by means of a micro-aerial vehicle: An artificial neural network approach for corrosion detection, *Adv. Intell. Syst. Comput.* 418 (2016) 223–234.
- [23] A. Ortiz, F. Bonnin-Pascual, E. García-Fidalgo, J.P. Company-Corcoles, F. Bonnin-Pascual, E. García-Fidalgo, J. Pep Company-Córcoles, The INCASS Project Approach towards Automated Visual Inspection of Vessels, in: *Jornadas Nacionales de Robótica (Spanish Robotics Workshop), Vol. 2017, 2017*, pp. 1–6.
- [24] Z. Fang, S. Yang, S. Jain, G. Dubey, S. Roth, S. Maeta, S. Nuske, Y. Zhang, S. Scherer, Robust autonomous flight in constrained and visually degraded shipboard environments, *J. Field Robotics* 34 (1) (2017) 25–52.
- [25] T. Dang, F. Mascarich, S. Khattak, H. Nguyen, H. Nguyen, S. Hirsh, R. Reinhart, C. Papachristos, K. Alexis, Autonomous search for underground mine rescue using aerial robots, in: *IEEE Aerospace Conference Proceedings, IEEE Computer Society, 2020*, pp. 1–8, <http://dx.doi.org/10.1109/AERO47225.2020.9172804>.
- [26] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, PIXHAWK: A system for autonomous flight using onboard computer vision, in: *International Conference on Robotics and Automation, IEEE, 2011*, pp. 2992–2997, <http://dx.doi.org/10.1109/ICRA.2011.5980229>.
- [27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, et al., ROS: an open-source robot operating system, in: *ICRA Workshop on Open Source Software, No. 3.2, Kobe, Japan, 2009*, p. 5.
- [28] J. Wang, E. Olson, AprilTag 2: Efficient and robust fiducial detection, in: *International Conference on Intelligent Robots and Systems, IROS, IEEE/RSJ, 2016*, pp. 4193–4198, <http://dx.doi.org/10.1109/IROS.2016.7759617>.
- [29] B. Pfrommer, K. Daniilidis, Tagslam: Robust slam with fiducial markers, 2019, arXiv preprint, arXiv:1910.00679.
- [30] T. Shan, B. Englot, Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain, in: *International Conference on Intelligent Robots and Systems, IROS, IEEE/RSJ, 2018*, pp. 4758–4765.
- [31] H. Wang, C. Wang, C.-L. Chen, L. Xie, F-loam: Fast lidar odometry and mapping, in: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021*, pp. 4390–4396.
- [32] J. Lin, F. Zhang, Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV, in: *2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020*, pp. 3126–3131.
- [33] Lloyd's Register, Thickness Measurement and Close-Up Survey Guidance, Lloyd's Register, 2019.

- [34] D. Holz, A.E. Ichim, F. Tombari, R.B. Rusu, S. Behnke, Registration with the point cloud library: A modular framework for aligning in 3-D, *IEEE Robot. Autom. Mag.* 22 (4) (2015) 110–124.
- [35] F.A. Maken, F. Ramos, L. Ott, Stein ICP for uncertainty estimation in point cloud matching, *IEEE Robot. Autom. Lett.* 7 (2) (2022) 1063–1070.
- [36] Z. Gojcic, C. Zhou, J.D. Wegner, A. Wieser, The perfect match: 3d point cloud matching with smoothed densities, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5545–5554.
- [37] C. Choy, J. Park, V. Koltun, Fully convolutional geometric features, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8958–8966.
- [38] R.Y. Brogaard, O. Ravn, E. Boukas, Absolute localisation in confined spaces using deep geometric features, *Electron. Lett.* (2021).
- [39] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, 2017, arXiv preprint [arXiv:1702.08734](https://arxiv.org/abs/1702.08734).
- [40] H. Yang, J. Shi, L. Carlone, Teaser: Fast and certifiable point cloud registration, *Trans. Robotics* (2020).
- [41] H. Yang, P. Antonante, V. Tzoumas, L. Carlone, Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection, *IEEE Robot. Autom. Lett.* 5 (2) (2020) 1127–1134.
- [42] D. Landry, F. Pomerleau, P. Giguere, CELLO-3D: Estimating the covariance of ICP in the real world, in: *International Conference on Robotics and Automation, ICRA, IEEE*, 2019, pp. 8190–8196.
- [43] A. Censi, An accurate closed-form estimate of icp's covariance, in: *Proceedings International Conference on Robotics and Automation, IEEE*, 2007, pp. 3167–3172.
- [44] M. Brossard, S. Bonnabel, A. Barrau, A new approach to 3D ICP covariance estimation, *Robotics Autom. Lett.* 5 (2) (2020) 744–751.
- [45] Q. Liu, D. Wang, Stein variational gradient descent: A general purpose bayesian inference algorithm, *Adv. Neural Inf. Process. Syst.* 29 (2016).
- [46] E.A. Wan, R. Van Der Merwe, The unscented Kalman filter for nonlinear estimation, in: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, IEEE, 2000, pp. 153–158.
- [47] T. Moore, D. Stouch, A generalized extended kalman filter implementation for the robot operating system, in: *Intelligent Autonomous Systems 13*, Springer, 2016, pp. 335–348.
- [48] S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation, *Proc. IEEE* 92 (3) (2004) 401–422.
- [49] P. Besl, N.D. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (2) (1992) 239–256.
- [50] Q.-Y. Zhou, J. Park, V. Koltun, Open3D: A modern library for 3D data processing, 2018, arXiv preprint [arXiv:1801.09847](https://arxiv.org/abs/1801.09847).



**Rune Yding Brogaard** earned his M.Sc. from Technical University of Denmark (DTU) in Electrical engineering in 2012 and is currently a Ph.D. student at DTU. Rune is also employed as a specialist within robotics at FORCE Technology, where his focus has been on aerial platforms for autonomous inspection. His interests include Unmanned Aerial Vehicles, Computer Vision, State estimation and Deep Learning.



**Evangelos Boukas** is an Associate Professor of Perception and Machine Learning for Perception in the Department of Electrical Engineering, Technical University of Denmark (DTU). Following his Ph.D. studies associated with the Automation and Robotics Section (TEC-MMA) of the European Space Agency, he served as an assistant/associate professor of robotics at Aalborg University. Evangelos has published more than 40 journal and conference papers in the fields of Robotics Vision and Machine Learning.