International Journal of Advanced Robotic Systems
March-April 2019: 1–14
© The Author(s) 2019
DOI: 10.1177/1729881419838967
journals.sagepub.com/home/arx
SAGE

Underwater navigation using visual markers in the context of intervention missions

Arturo Gomez Chavez, Christian A Mueller, Tobias Doernbach and Andreas Birk

Abstract

Intervention missions, that is, underwater manipulation tasks, for example, in the context of oil-&-gas production, require a high amount of precise, robust navigation. In this article, we describe the use of an advanced vision system suited for deep-sea operations, which in combination with artificial markers on target structures like oil-&-gas production-Christmas-trees significantly boosts navigation performance. The system is validated in two intensive field tests running off the shore of Marseille, France. In the experiments, a commercial remotely operated vehicle equipped with the system and a mock-up structure with an oil-&-gas production panel is used to evaluate the navigation performance.

Keywords

Navigation, intervention, oil-&-gas production, visual odometry, augmented reality markers, deep-sea operations

Date received: 5 December 2018; accepted: 25 January 2019

Topic: Field Robotics
Topic Editor: Yangquan Chen
Associate Editor: Pengfei Zhang

Introduction

In recent years, there has been an increasing interest in autonomous behaviors for intervention missions, that is, missions with autonomous underwater vehicles (AUVs) or at least semiautonomous remotely operated vehicles (ROVs), which include automated, machine-controlled manipulation tasks.[1–9] Surveillance and inspection missions are usually not so critical in terms of navigation performance. For these missions, the vehicle observes the environment only from a distance and the localization accordingly needs to be only reasonably accurate. Also, post-processing of the data with, for example, simultaneous localization and mapping or even with manual correction of outliers by an end user is a common practice. But intervention missions in contrast have higher demands on the navigation. Especially, the vehicle is by definition closer to objects of interest that need to be manipulated—precise and robust localization in real time is hence of high interest.

We present here work on the use of artificial markers to improve navigation in the context of intervention missions. Concretely, ArUco markers[10] are used. This is motivated by the fact that intervention typically takes place in environment settings that involve man-made structures, for example, Christmas-tree installations in the context of oil-&-gas production (OGP), where the markers can be easily added before deployment of the structures. The idea

Department of Computer Science and Electrical Engineering, Jacobs University Bremen, Bremen, Germany

Corresponding author:
Andreas Birk, Jacobs University Bremen, Campus Ring 1, 28759 Bremen, Germany.
Email: a.birk@jacobs-university.de

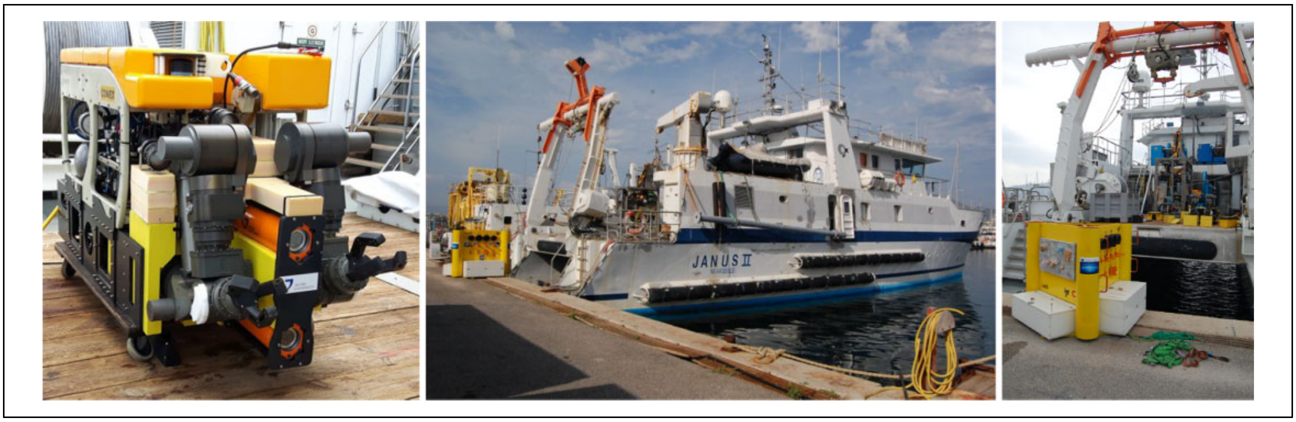

Figure 1. Our navigation system has been employed at a high TRL of 6. It was tested, for example, in two extensive field trials off the shore in Marseille. In addition to the vision system to aid the navigation (left), the setup in these trials consists of an Apache ROV extended a dual arm setup, which is deployed from the COMEX Janus II vessel (center). A mock-up panel structure is used to test different application scenarios (right). TRL: technology readiness level; ROV: remotely operated vehicle.

to exploit man-made structures for navigation in the context of intervention can also be found, for example, by Evans et al.,[9] where edges extracted by computer vision are matched against a priori known 3-D CAD models of the structures. Visual markers have also been used before in underwater applications, for example, in a form of active light beacons for docking.[11] In the work presented here, augmented reality (AR) markers are used, which are in general designed to provide good identification and localization capabilities. AR markers have among others been used underwater for their original purpose, that is, for AR by displaying virtual objects and virtual scenes into a diver's view of the real world at the location of the markers,[12] for example, to assist divers in commercial operations[13] or to enable underwater games for divers.[14] AR markers have also been used to enable a diver to communicate with an AUV[15] and for the identification of nodes in an underwater sensor network.[16] Last but not least, AR markers have also been used in the context of navigation, for example, for the visual servoing of an ROV on a moored target[17] and the detection and localization of panel elements for intervention.[18]

The work presented here is done in the context of the EU project "Effective Dexterous ROV Operations in Presence of Communications Latencies (DexROV)".[19] DexROV deals with the problem that the state of the art for underwater manipulation is dominated by costly ROV operations, which require an offshore crew. This crew typically consists of at least an intendant, an operator, and a navigator. And it often has to be duplicated or even tripled due to work shifts enabling 24/7 operations for, for example, missions in OGP. Furthermore, intervention is still dominated by low-level, manual control of the manipulator(s) and of the vehicle itself. The core idea of DexROV is to enable operations from an onshore control center. This includes among others a reduction of the gap between low-level teleoperation and full autonomy. The user in an onshore control center (e.g. in Brussels, Belgium in our field trials) interacts with a real-time simulation environment, and a cognitive engine analyzes the user's control requests and turns them into movement primitives that the ROV needs to autonomously execute in the real environment (e.g. in the waters off the shore of Marseille, France, in our trials). One challenge for this operation scheme is the communication latencies of the satellite link between the control center and the vessel.

The contributions of our work on navigation aided by AR markers presented here are among others (a) the use of a novel underwater calibration method and of image enhancement methods to improve marker detection and localization, (b) additional measures with respect to the view angle and distance range to increase the robustness, and most importantly (c) a significant amount of system development and integration leading to a high technology readiness level (TRL) of six suited for field trials in realistic application conditions (Figure 1), especially with respect to a wide range of challenging visibility conditions.

The system components

The core navigation system

Our core navigation system is based on a standard, state-of-the-art approach, namely, the use of a Doppler velocity log (DVL) and an inertial measurement unit (IMU). Concretely, a Navquest 600P micro DVL and a Xsens MTi-300 IMU are used. The DVL is rated for up to 6000 m depth, that is, it is suited for deep-sea operations. The DVL is mounted on the bottom of the DexROV-skid, which is added to the Apache ROV in the field trials. The IMU is integrated into the compute bottle of the vision system, which is also integrated on the skid. The DVL is directly connected to the compute bottle.

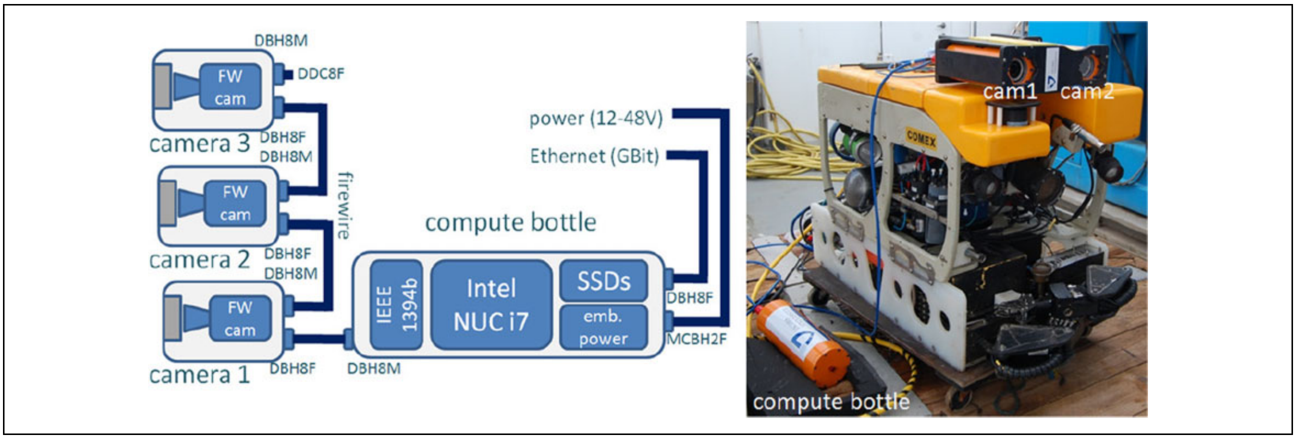

Figure 2. The camera system consists of a computer bottle with significant online processing power to which multiple cameras in pressure housings can be daisy-chained (left), for example, as a stereo setup with two cameras (right).

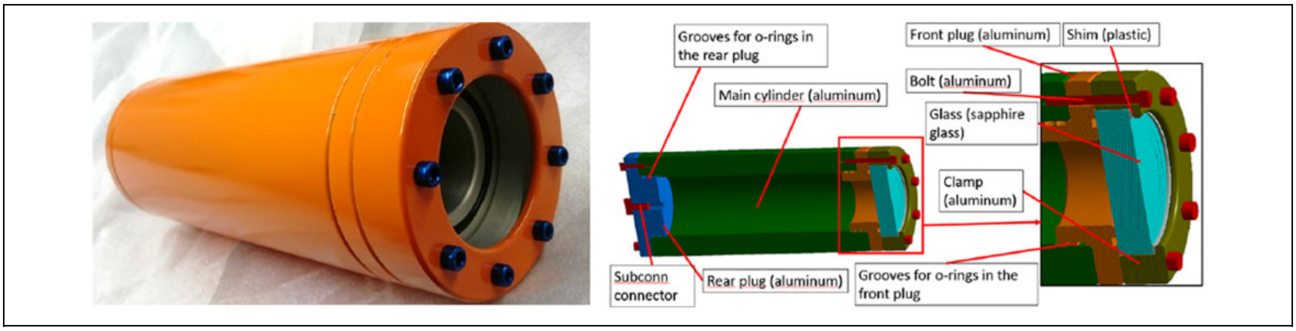

Figure 3. The camera pressure housing of one camera of the intelligent underwater camera system.

The DVL provides altitude as well as velocities in X, Y, and Z (speed over ground), which is used for the computation of the translation. The IMU is used to track the ROV orientation. As described in more detail later on, a standard approach is used for the processing of the data for the core navigation system, namely, an extended Kalman filter (EKF). This processing is handled by the vision computer in the onboard compute bottle, which is described in the next section.

Vision system components

State-of-the-art ROVs like the Apache used in DexROV often rely on analog camera systems. However, the image quality is strongly influenced by the connection to the support vessel. Due to long cables used in deep-sea operations, images are noisy and there can be interrupts in the streams, which makes processing of the data by computer vision quite challenging. The use of digital cameras has multiple advantages, especially when they are combined with computation power on the vehicle to generate an intelligent vision system. First of all, an intelligent underwater vision system (Figure 2, 3 and 4) can be used to minimize the traffic over the umbilical cable from the ROV to the vessel. Especially, it allows to online adapt the image resolution, the compression factor, and the frame rate to optimally use the available bandwidth for the task at hand. Furthermore, computer vision can be used directly onboard of the ROV to assist core capabilities, like navigation in the work presented here, up to the provision of autonomous functions. This processing onboard of the ROV minimizes latencies and increases robustness compared to processing on the vessel, which requires the transmission of sensor data over the limited data connection of a tether up to the vessel and also the sending of commands down to the ROV again.

Our intelligent vision system is based on high-resolution firewire (IEEE 1934b) cameras in pressure housings. Concretely, Point Grey Grasshoppers2 cameras are used. They are based on Sony ICX285 CCD sensors, which are known for good performance in lowlight conditions. The firewire bus signals are relayed over high-frequency underwater cables between the bottles to allow the daisy-chaining of multiple cameras connected to an embedded computer, which is used for vision processing and for adaptive video compression onboard of the ROV. For the onboard vision computer, which also services the core navigation based on the DVL and the IMU, an Intel

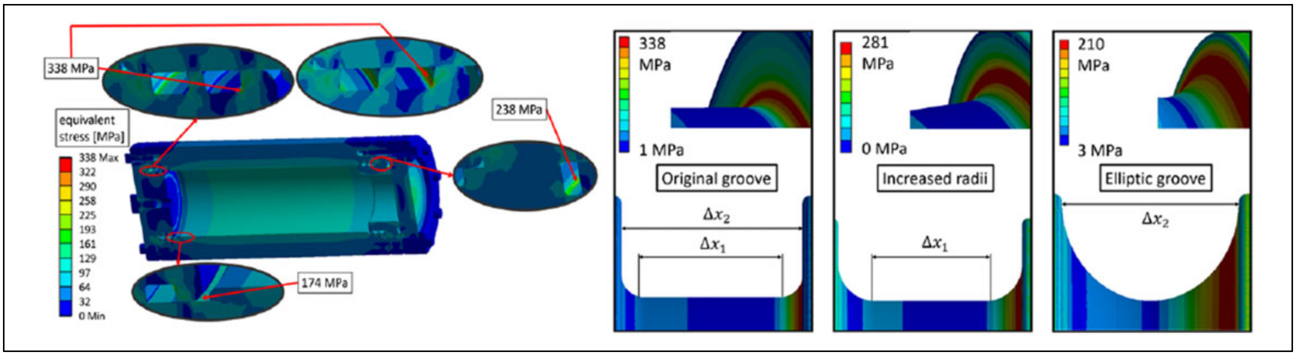

Figure 4. The design of the pressure housings for the cameras and the compute bottle are optimized by numerical simulations.

NUC with a 4th-Gen Intel Core i5-4250U is used. The firewire bus supports among others the synchronization of the cameras. They can hence be used for stereo, respectively, multi-camera setups to generate depth information from different views with a known relative geometry. The option of more than two synchronized cameras allows implementing different baselines to cover different range/resolution trade-offs in one system. Due to payload constraints of the Apache ROV, a stereo setup with two cameras is used in all field trials.

The camera bottles are equipped with flat Sapphire glass windows. All bottles designs are optimized by numerical simulations for 4000 msw and pressure tested in the real world for deep-sea operations up to 2000 msw. The cameras need to be calibrated intrinsically and extrinsically with respect to the ROV platform. For intrinsic calibration, own work is used,[20] which allows calibrating the system in air, prior to underwater deployment with no need of further in-water calibration. The predicted camera parameters take salinity, temperature, and pressure into account, so the varying depth of the application can easily be taken care of. This is facilitated by a new camera model dubbed PinAx as it combines an axial and a pinhole camera model.[20] Based on the calibrated camera model, the images from both cameras are rectified to remove the distortions, which stem from the refraction caused by the water and the protective glass panel in front of the cameras. This contributes to the robust recognition and localization of the AR markers. Furthermore, several pre-processing steps are executed to enhance the image quality. Especially, own methods to reduce haze are used in this context.[21,22]

System operation

For the onboard vision system, reliability and robustness are of high interest. Furthermore, the network connection through the tether between the ROV and the vessel is in general subject to delays and dropouts, so any processing involving the vessel computing facilities should be avoided. Therefore, an approach as automated as possible is chosen for the operation of the onboard vision computer. This includes a possibility to (re-)start the sensor drivers and the data recording via asynchronous commands which, in case of network failures, are transmitted after the communication has been restored and do not fail in such cases. This is realized with a finite state machine on the onboard vision computer, which gets triggered by asynchronous network commands from the vessel computers, or otherwise boots and runs autonomously once the ROV is switched on without the need of any operator interaction. It is also possible to do any computation processes including, for example, data collection completely autonomously by launching the system a certain time after the ROV is powered on or once a certain altitude above sea ground has been reached.

Approach and methods

Definitions and notations

A core idea for the work presented here is as mentioned to exploit the fact that intervention typically is done in environments featuring man-made structures. Here, this is a mock-up structure with several panels for testing different application scenarios. Given the panel as a landmark, a kinematic model is used, which describes the spatial relationships among the ROV and the panel (see Figure 5). Therein, the following transformations are defined:

- robot in camera frame: ${}^{C}_{R}\mathbf{T}$;
- camera in marker frame: ${}^{M}_{C}\mathbf{T}$;
- marker in panel frame: ${}^{P}_{M}\mathbf{T}$;
- panel in odom frame: ${}^{O}_{P}\mathbf{T}$;
- robot in marker frame: ${}^{M}_{R}\mathbf{T} = {}^{M}_{C}\mathbf{T}{}^{C}_{R}\mathbf{T}$;
- marker in odom frame: ${}^{O}_{M}\mathbf{T} = {}^{O}_{P}\mathbf{T}{}^{P}_{M}\mathbf{T}$; and
- robot in odom frame: ${}^{O}_{R}\mathbf{T} = {}^{O}_{M}\mathbf{T}{}^{M}_{R}\mathbf{T}$.

Our approach focuses on the estimation of position and orientation of the panel with respect to the origin of the vehicle trajectory. Our approach incorporates a priori knowledge, especially the CAD model of the panel and the placement of the visual markers at known locations on the panel. Based on this, the panel pose in the odometry frame ${}^{O}_{P}\mathbf{T}$ can be reliably estimated using the detected marker pose with

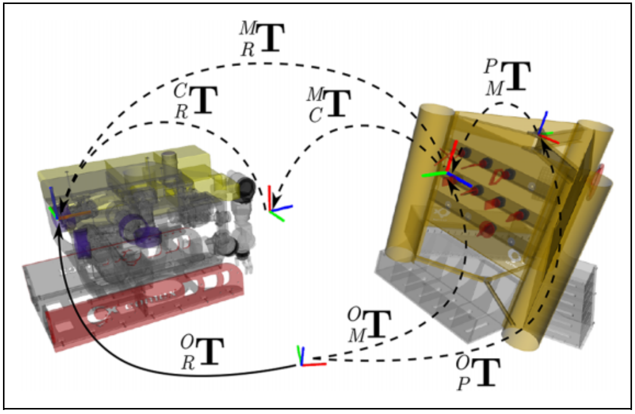

Figure 5. An illustration of the transformations between the vehicle and the panel. ROV: remotely operated vehicle.

respect to the camera frame ${}^{C}_{M}\mathbf{T}$, the camera pose on the robot frame ${}^{R}_{C}\mathbf{T}$, the panel pose in the marker frame ${}^{M}_{P}\mathbf{T}$, and the current robot pose in the odometry frame ${}^{O}_{R}\mathbf{T}$ (Figure 5).

$$ {}^{O}_{P}\mathbf{T} = {}^{O}_{R}\mathbf{T}{}^{R}_{C}\mathbf{T}{}^{C}_{M}\mathbf{T}{}^{M}_{P}\mathbf{T} \tag{1} $$

Consequently, n marker observations lead to n panel pose estimates ${}^{O}_{P}\mathbf{T}$ that eventually allow to compute the pose mean, which includes the mean position and the orientation, determined by spherical linear interpolation (Slerp).[23]

Vehicle localization

Localization is a challenging task, especially in underwater conditions, due to noisy sensor readings typically based on acoustic devices like ultra-short baseline (USBL) systems, single-beam or multi-beam sonars, DVL, or readings provided by inertial navigation systems. Consequently, localization methods rely on multiple modalities to increase reliability.[24,25] A typical and well-established approach to deal with sensor fusion is the EKF,[26] which allows incorporating the different modalities on the basis of their different uncertainties.

In order to increase the pose accuracy, we exploit the panel as a visual landmark due to its static pose on the seafloor and its visual augmentation with multiple markers. Once the panel pose is estimated, the robot pose can be inferred and used as an additional EKF input modality. In the following, we describe our EKF-based localization system incorporating standard sensor readings and the visual landmarks.

Visual landmark-based localization

Figure 6 shows a sample pose estimate of a visual marker, which is used to infer the panel pose through the space transformations, shown in Figure 5,—note that the panel is only partially observed. The panel is taken as a fixed landmark and the robot pose ${}^{O}_{R}\mathbf{T}$ can be estimated as follows

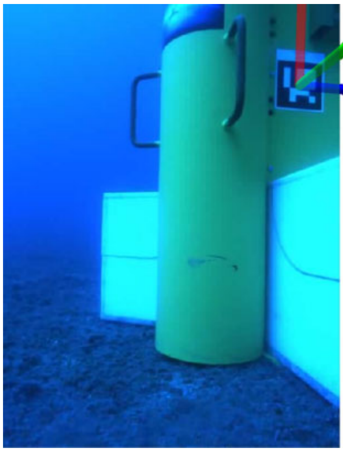

Figure 6. An example of a marker detection under sea trial conditions.

$$ {}^{O}_{R}\mathbf{T} = {}^{O}_{P}\mathbf{T}{}^{P}_{M}\mathbf{T}{}^{M}_{C}\mathbf{T}{}^{C}_{R}\mathbf{T} \tag{2} $$

where ${}^{O}_{P}\mathbf{T}$ is the panel pose in the odometry frame, ${}^{P}_{M}\mathbf{T}$ is one marker pose in the panel frame, ${}^{M}_{C}\mathbf{T}$ is the camera pose with respect to the marker, and ${}^{C}_{R}\mathbf{T}$ is the robot fixed pose with respect to the camera. Further on, the means of the robot position ${}^{O}_{R}\bar{\mathbf{p}}$ and its orientation ${}^{O}_{R}\bar{\mathbf{q}}$ with respect to the odometry frame are estimated from multiple marker detections (for better readability, ${}^{O}_{R}\bar{\mathbf{p}} = \bar{\mathbf{p}}$, ${}^{O}_{R}\bar{\mathbf{q}} = \bar{\mathbf{q}}$)

$$ \bar{\mathbf{p}} = \frac{1}{n}\left(\sum_{i=1}^{n}\mathbf{p}_i\right) \tag{3} $$

where $\mathbf{p}_i = {}^{O}\mathbf{p}_i$ is the position of the robot in the odometry frame O estimated from detecting marker i; n is the number of detected markers.

For calculating the mean $\bar{\mathbf{q}}$ of rotations $\mathbf{q}$, we use the Slerp algorithm

$$ \bar{\mathbf{q}} = \mathrm{s}_n \tag{4} $$

with

$$ s_i = \mathrm{Slerp}(s_{i-1}, \mathbf{q}_i, t), s_1 = \mathrm{Slerp}(\mathbf{q}_1, \mathbf{q}_2, t) \tag{5} $$

$$ \mathrm{Slerp}(\mathbf{q}_i, \mathbf{q}_{i+1}, t) = \frac{\sin(1-t)\theta}{\sin\theta}\mathbf{q}_i + \frac{\sin t\theta}{\sin\theta}\mathbf{q}_{i+1} \tag{6} $$

$$ \theta = \arccos(\mathbf{q}_i \cdot \mathbf{q}_{i+1}), t = \frac{1}{n} \tag{7} $$

where $\mathbf{q}_i = {}^{O}_{R}\mathbf{q}_i$ is the rotation quaternion between the odometry frame O and the robot frame R estimated from detecting marker i.

In order to use this robot pose estimate in the localization filter, a covariance matrix ${}^{O}_{R}\mathbf{C}$ has to be determined. This requires the element-wise variances $\sigma^2_{\mathbf{p}}$ of the marker positions $\mathbf{p}$, which can be calculated as

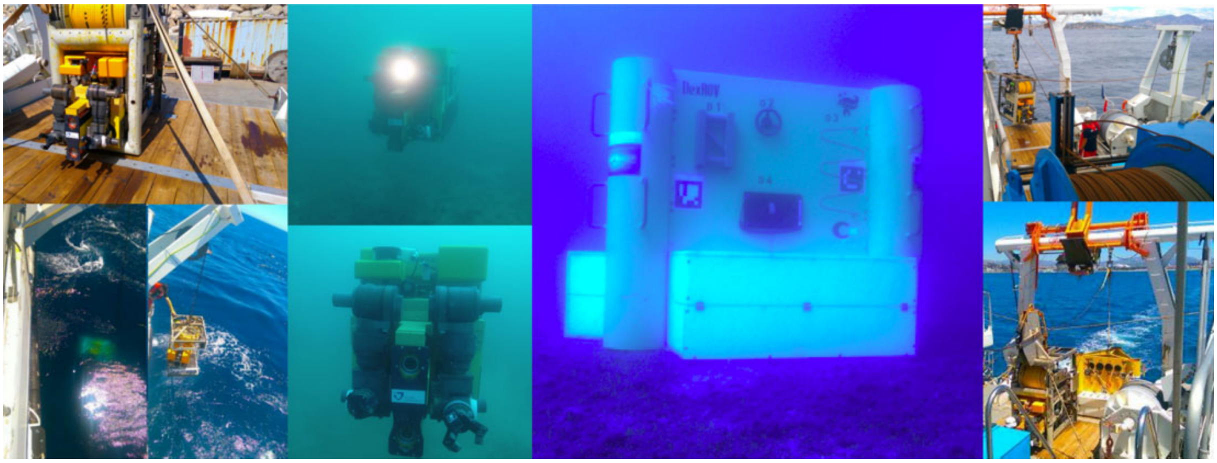

Figure 7. Impressions of sea trials in Marseille.

$$\sigma_{\mathbf{p}}^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{p}_i - \bar{\mathbf{p}})^2 \tag{8}$$

Additionally, the element-wise orientation variances $\sigma_{\mathbf{q}}^2$ have to be computed using the circular variance[27] due to the angle wraparound on the unit circle

$$\sigma_{\mathbf{q}}^2 = 2\pi\left(1 - \frac{r}{n}\right) \tag{9}$$

with

$$r = \sqrt{\left(\sum_{i=1}^{n}\sin\mathbf{q}_i\right)^2 + \left(\sum_{i=1}^{n}\cos\mathbf{q}_i\right)^2} \tag{10}$$

Finally, the covariance matrix ${}_R^O\mathbf{C}$ for the transformation between the odometry frame O and the robot frame R can be assembled as

$${}_R^O\mathbf{C} = \begin{pmatrix} \sigma_{\mathbf{p}_x}^2 & & & & & \\ & \sigma_{\mathbf{p}_y}^2 & & & 0 & \\ & & \sigma_{\mathbf{p}_z}^2 & & & \\ & & & \sigma_{\mathbf{q}_\phi}^2 & & \\ & 0 & & & \sigma_{\mathbf{q}_\theta}^2 & \\ & & & & & \sigma_{\mathbf{q}_\psi}^2 \end{pmatrix} \tag{11}$$

Eventually, a covariance matrix ${}_R^O\mathbf{C}$ for the robot pose can be defined in the form of

$${}_R^O\mathbf{C} = \mathrm{diag}(\sigma_{\mathbf{p}_x}^2, \sigma_{\mathbf{p}_y}^2, \sigma_{\mathbf{p}_z}^2, \sigma_{\mathbf{q}_\phi}^2, \sigma_{\mathbf{q}_\theta}^2, \sigma_{\mathbf{q}_\psi}^2) \tag{12}$$

The full robot pose estimate ${}_R^O\mathbf{T} = \langle {}_R^O\bar{\mathbf{p}}, {}_R^O\bar{\mathbf{q}} \rangle$ along with the respective covariance matrix ${}_R^O\mathbf{C}$ is then taken as an input for the localization filter.

Extended Kalman filter

As mentioned before, we use a standard EKF[26] to estimate the robot pose over time with a state space that consists of position x, y, z, orientation ϕ, θ, ψ, translational $\dot{x}, \dot{y}, \dot{z}$, and angular velocities $\dot{\phi}, \dot{\theta}, \dot{\psi}$ as well as translational accelerations $\ddot{x}, \ddot{y}, \ddot{z}$. We only incorporate direct sensor measurements to the EKF, that is, no integrated or differentiated values are used. An inertial navigation system measures angular and linear accelerations, a DVL provides position outputs in a form of altitude readings and linear velocities, and the information from the marker landmarks is incorporated as pose readings. To increase the localization filter robustness, obvious outliers from sensor readings are rejected heuristically, and the pose inputs inferred from visual markers are tuned based on our experimental results.

Experiments and results

Data collection at sea trials

The results presented here include extensive field trials of 2 weeks each time in the Mediterranean Sea off the shore of Marseille in June/July 2017 and in June/July 2018 (Figure 7). A test panel was developed for validation by the DexROV project partner "Compagnie maritime d'expertises (COMEX)". COMEX also provided the Apache ROV and the Janus-II vessel for the sea trials. The panel served as target for the trials to emulate different scenarios, for example, offshore oil-&-gas facilities or the handling of archeological artifacts (Figures 6, 1 and 8). The panel consists of three sides, which are equipped with mock-up elements. One side is used to test components in offshore oil-&-gas interfaces based on the ISO 13628 standard including, for example, valves and wet-mate connectors. Furthermore, a biologic panel including mock-up corals and an archeological box including mock-up ceramics

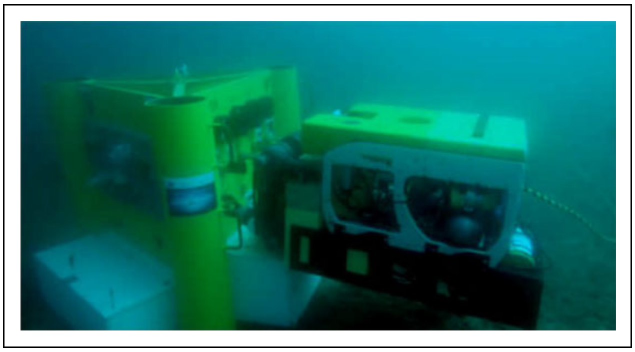

Figure 8. The ROV and the test panel during the field trials. ROV: remotely operated vehicle.

are included. The panel is augmented with ArUco AR markers as reference points to aid navigation.

The panel was submerged at different depths under different weather conditions with accordingly different visibility conditions. In order to provide a reliable estimate for the alignment, only marker pose estimates with respect to the ROV camera are used, that is, other cues are neglected due to high noise level of the other available sensor feeds (e.g. DVL, USBL, IMU). The experimental setup facilitates experiments ranging from pure algorithm performance experiments considering noise-free ground truth sensor feeds over increasing noise levels under real-world sea trial conditions. Especially, a high-fidelity simulator in the loop (SIL) based on Gazebo can be used to replace components for testing purposes. This is also of interest during the development of the system components and their integration. It allows, for example, the investigation of bottlenecks, constraints, and expected performance under certain environmental conditions or for specific configurations.

Panel pose estimation

The first benchmarking test $\mathcal{T}_P$ evaluates the accuracy of the panel pose estimation, as this pose estimation is the starting point for other tasks like handle pose estimation and especially also navigation. This consequently validates the robustness of the visual markers under different real-world conditions. In order to replicate a realistic trajectory for the robot to navigate, the robot poses given by the detected markers in recorded real-world data are extracted and first used as waypoints in the SIL for comparison. In this way, the same visual perspectives as in the field trials are obtained in the SIL, which represent a routine trajectory commonly executed by the robot operators. Thus, we can determine the expected error as the difference between the ground-truth panel pose in the SIL ${}^{O}_{P}\mathbf{T}_S$ and the panel pose determined from marker detection ${}^{O}_{P}\mathbf{T}_M$

$$\begin{aligned} m(\mathcal{T}_P, \mathcal{E}) &= d({}^{O}_{P}\mathbf{T}_S, {}^{O}_{P}\mathbf{T}_M) \\ &= \langle d({}^{O}_{P}\bar{\mathbf{p}}_S, {}^{O}_{P}\bar{\mathbf{p}}_M), d({}^{O}_{P}\bar{\mathbf{q}}_S, {}^{O}_{P}\bar{\mathbf{q}}_M) \rangle \end{aligned} \tag{13}$$

where $d({}^{O}_{P}\bar{\mathbf{p}}_S, {}^{O}_{P}\bar{\mathbf{p}}_M)$ is the Euclidean distance between positions and $d({}^{O}_{P}\bar{\mathbf{q}}_S, {}^{O}_{P}\bar{\mathbf{q}}_M)$ is the minimal geodesic distance between orientations[28] under conditions $\mathcal{E}$.

Figure 9 shows the mean $\overline{M}(\mathcal{T}_P, \mathcal{E})$ and standard deviation $\sigma(M(\mathcal{T}_P, \mathcal{E}))$ for all panel observations under noise-free conditions $\mathcal{E}^0$, as used initially in the development stage, and underwater conditions $\mathcal{E}^*$, as used for validation.

As expected, our approach features very high accuracy in the noise-free environment $\mathcal{E}^0$ with a translation and orientation error of only 0.02 m and 1.2°, respectively. As to be expected, underwater conditions decrease the accuracy and the number of detections. Nevertheless, there is still only a reasonable translation and orientation error of 0.118 m and 4.2°, respectively, under the $\mathcal{E}^*$ conditions. For our navigation requirement, this error is small enough to reach the desired performance. Furthermore, it can be improved by image registration and the variance $\sigma^2(M(\mathcal{T}_P, \mathcal{E}^*))$ can be used to fine-tune the robot pose covariance matrix ${}^{O}_{R}C$ to improve the localization. Tests $\mathcal{T}_H$ and $\mathcal{T}_L$ show this in the next experiments.

Localization

The next test $\mathcal{T}_L$ benchmarks the localization method. First, we validate the use of visual landmarks in the localization filter through SIL under the found $\mathcal{E}^*$ conditions for comparison. Then, real-world data are used and we compare the task performance with and without the use of visual markers. Finally, results with EKF parameters, which are tuned based on the results from tests $\mathcal{T}_{L1}$ and $\mathcal{T}_P$, are presented.

Evaluation measures. In order to provide meaningful numerical results, we introduce the following error measures:

Robot pose estimate error (simulated data):

The pose estimate error of ${}^{O}_{R}\mathbf{T}_M$ and ${}^{O}_{R}\mathbf{T}_F$ with respect to simulation ground truth is used in the first benchmarking test; it is denoted by $m_{S,M}(\mathcal{T}_{L1}) = d({}^{O}_{R}\mathbf{T}_S, {}^{O}_{R}\mathbf{T}_M)$, respectively. $m_{S,F}(\mathcal{T}_{L1}) = d({}^{O}_{R}\mathbf{T}_S, {}^{O}_{R}\mathbf{T}_F)$ as shown in equation (13).

Robot pose estimate error (real-world data):

To evaluate benchmarking tests on real-world data, we use the robot pose estimate given by the marker ${}^{O}_{R}\mathbf{T}_M$ as reference ground truth and compute the mean and standard deviation of the measure $m_{M,F}(\mathcal{T}_{Li}) = d({}^{O}_{R}\mathbf{T}_M, {}^{O}_{R}\mathbf{T}_F)$ plus the *lag-one autocorrelation* $m_A(\mathcal{T}_{Li}) = \sum_t {}^{O}_{R}\mathbf{T}_F(t) {}^{O}_{R}\mathbf{T}_F(t-1)$ on the EKF-predicted poses. $m_A(\mathcal{T}_{Li})$ is a measure of the trajectory smoothness, which is of interest to prevent sudden jumps in the navigation estimate that can interfere with manipulation tasks.

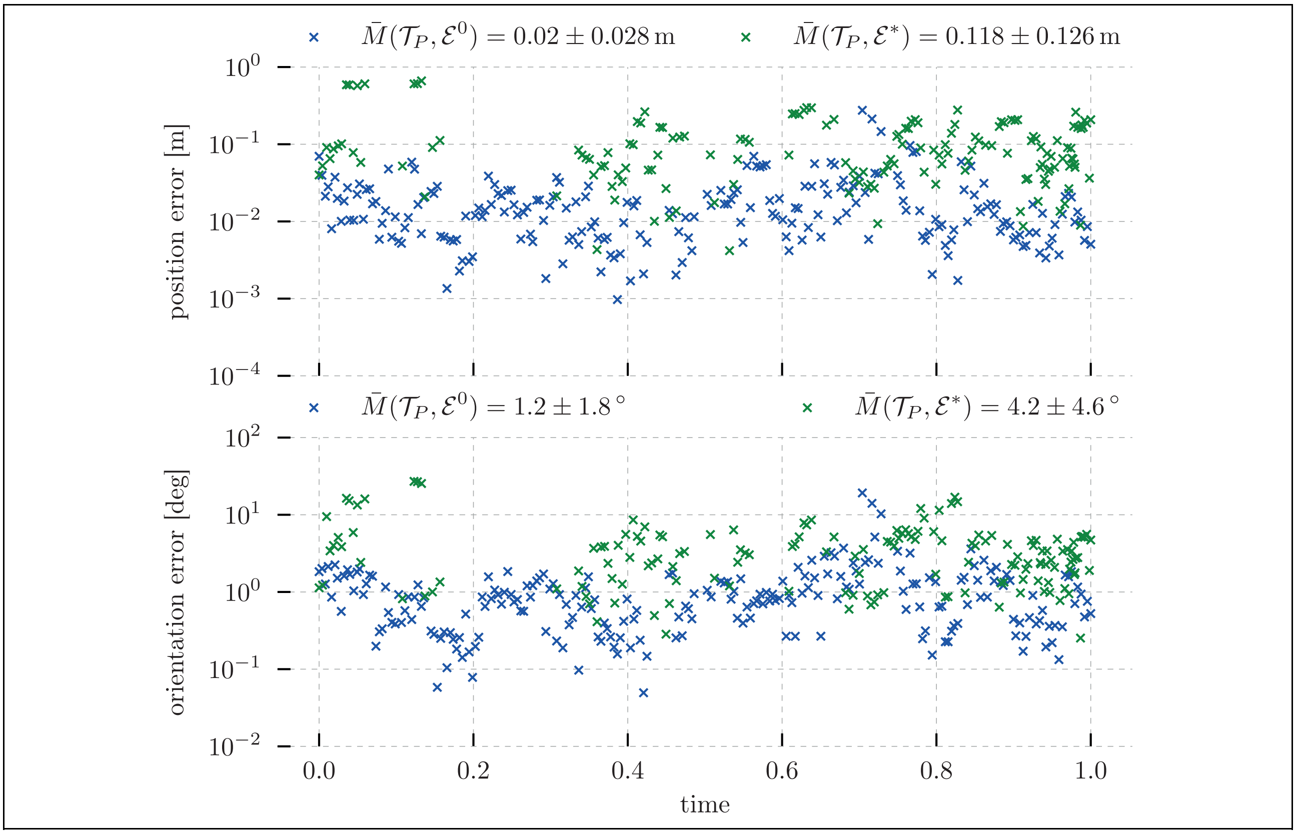

Figure 9. $\mathcal{T}_P$ results: panel detection errors $M(\mathcal{T}_P, \mathcal{E}^0)$ and $M(\mathcal{T}_P, \mathcal{E}^*)$ for noise-free $\mathcal{E}^0$ and underwater $\mathcal{E}^*$ conditions.

Relative image quality:

Furthermore, we use several measures for all images where the ground-truth landmark can be detected. This comprises of the NIQMC metric,[29] a model of visual image quality with respect to contrast distortion, as well as the number of correspondences between two consecutive images of several established feature descriptors, which are commonly used for registration, recognition, and mapping. All of the measures are normalized over all collected real-world data, that is, the resulting quality measures are relative numbers with the help of which different images can be qualitatively compared. From all measures, we compute the mean $\overline{m_I}(t)$ and the standard deviation $\tilde{m}_I(t)$ for each timestamp t

$$\bar{m}_I(t) = \frac{1}{|M_I|}\sum_m \hat{m}_{I,m}(t) \tag{14}$$

$$\tilde{m}_I(t) = \sqrt{\frac{1}{|M_I|}\sum_m \left(\hat{m}_{I,m}(t) - \overline{m}_I(t)\right)^2} \tag{15}$$

where

$$\hat{m}_{I,m}(t) = \frac{m_{I,m}(t) - \min\limits_t(m_{I,m}(t))}{\max\limits_t(m_{I,m}(t)) - \min\limits_t(m_{I,m}(t))} \tag{16}$$

and

$$\begin{aligned} &m_{I,m} \in M_I \\ &M_I = \{\mathrm{NIQMC}, \mathrm{SIFT}, \mathrm{SURF}, \mathrm{BRISK}, \mathrm{KAZE}, \mathrm{AKAZE}\} \end{aligned} \tag{17}$$

is one individual image quality measure.

$\mathcal{T}_{L1}$—localization in simulation. For real-world underwater localization, no accurate ground-truth data are usually available. For this reason, the performance of the proposed localization filter that integrates visual landmarks into the EKF is tested in high-fidelity simulation first.

In this test $\mathcal{T}_{L1}$, the simulated robot follows a trajectory around the panel for which the visual input has been recorded in the 2017 field trials. This video stream has been used to determine the robot pose with respect to the panel in order to recreate a similar trajectory in simulation. Note, however, that it is not necessary in this first experiment to determine accurate ground truth, but an approximately similar trajectory to the one used in the field trial is sufficient.

During this movement, the ground-truth robot pose in simulation ${}^{O}_{R}\mathbf{T}_S$ is recorded alongside the robot pose determined through the detected marker pose ${}^{O}_{R}\mathbf{T}_M$ in simulation and the localization filter ${}^{O}_{R}\mathbf{T}_F$. Note that the EKF receives only the visual landmark-based pose estimates to prove that it converges to ground truth. Like with all our experiments, this test is repeated several times to

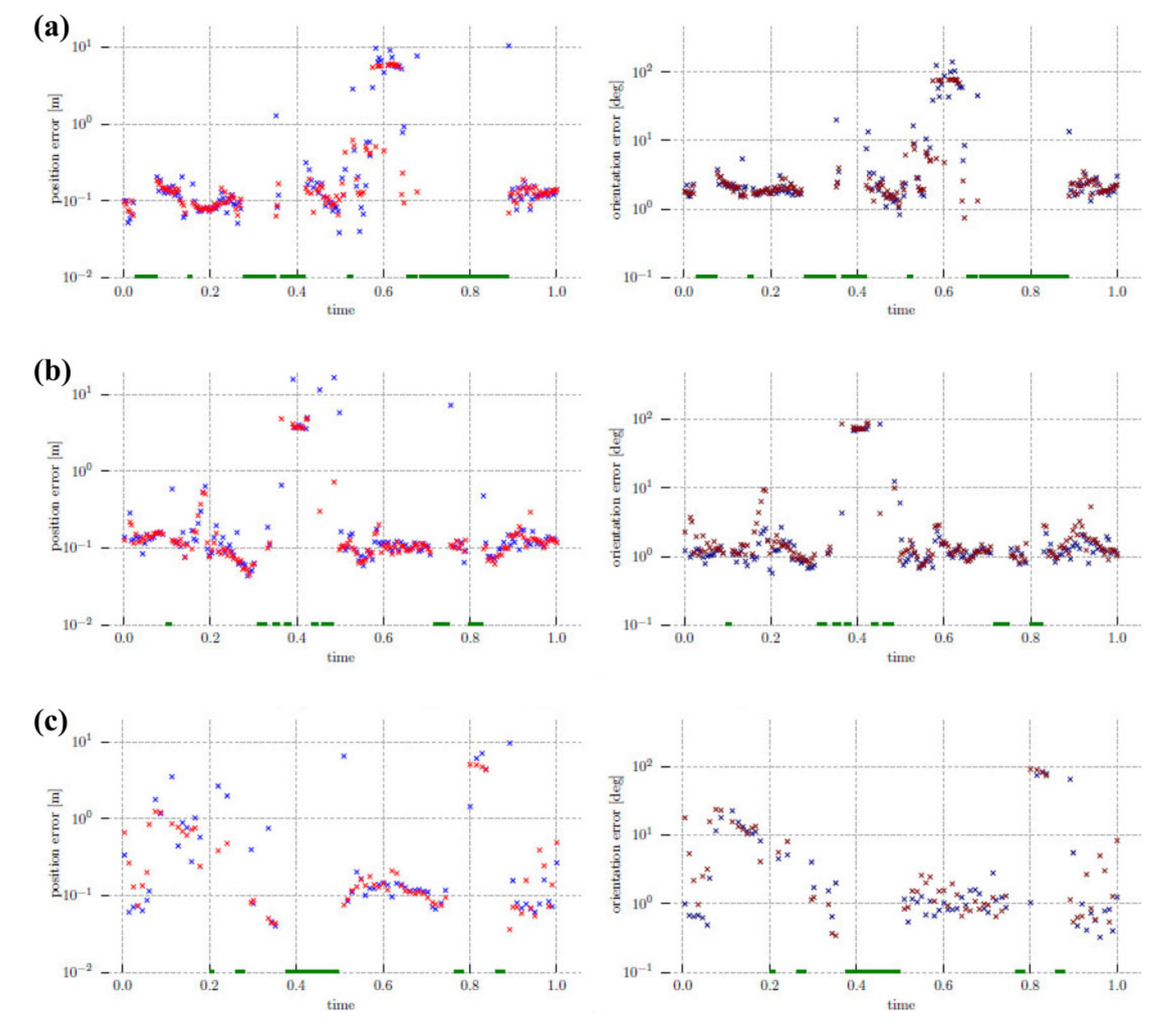

Figure 10. $\mathcal{T}_{L1}$ results: position errors $d({}^O_R\bar{\mathbf{p}}_S, {}^O_R\bar{\mathbf{p}}_M)$ (red)/$d({}^O_R\bar{\mathbf{p}}_S, {}^O_R\bar{\mathbf{p}}_F)$ (blue) and orientation errors $d({}^O_R\bar{\mathbf{q}}_S, {}^O_R\bar{\mathbf{q}}_M)$ (dark-red)/$d({}^O_R\bar{\mathbf{q}}_S, {}^O_R\bar{\mathbf{q}}_F)$ (dark-blue) between ground-truth robot pose and marker-based/localization filter-based robot pose estimates, while the robot moves around the panel on a trajectory recorded in field trials; no marker detected for sampling times is marked green. (a) $\mathcal{T}_{L1a}$—integration stage; before localization parameter tuning. (b) $\mathcal{T}_{L1b}$—integration stage; after localization parameter tuning. (c) $\mathcal{T}_{L1c}$—validation stage.

prove the feasibility of test $\mathcal{T}_{L1}$ through both the development and the validation stage.

Integration stage:$\mathcal{T}_{L1\{a-b\}}$

During development, we initially execute $\mathcal{T}_{L1}$ on the non-optimized localization settings in the beginning of the continuous system integration cycle (denoting this first run as $\mathcal{T}_{L1a}$). The resulting pose estimate error of ${}^O_R\mathbf{T}_M$ and ${}^O_R\mathbf{T}_F$ with respect to simulation ground truth, $m_{S,M}(\mathcal{T}_{L1})$ and $m_{S,F}(\mathcal{T}_{L1})$, is shown in Figure 10(b).

Figure 11(a) shows the driven-off trajectory with the resulting pose errors; this and the detailed error breakdown in Figure 10(a) show that whenever no marker is detected for a while, the EKF error increases significantly on the next reading, but then quickly reconverges toward ground truth. On parts of the trajectory where markers are constantly visible, the localization error decreases substantially below 0.3 m/3°, for example, between time marks 0.1 and 0.25.

Further down the system integration path, $\mathcal{T}_{L1}$ is repeated as $\mathcal{T}_{L1b}$ with improved parameters of the localization method for validation. The results are shown in Figures 10(a) and 11(b). Prominently, the trajectory, which is estimated by the localization method, is a lot denser than in $\mathcal{T}_{L1a}$. The reason for this is the fact that visual markers are available only on the flat panel surfaces, but not on the cylindrical edges. Hence, our optimized localization filter is able to estimate the expected robot pose accurately from the previous visual input even though this is stalled for some time.

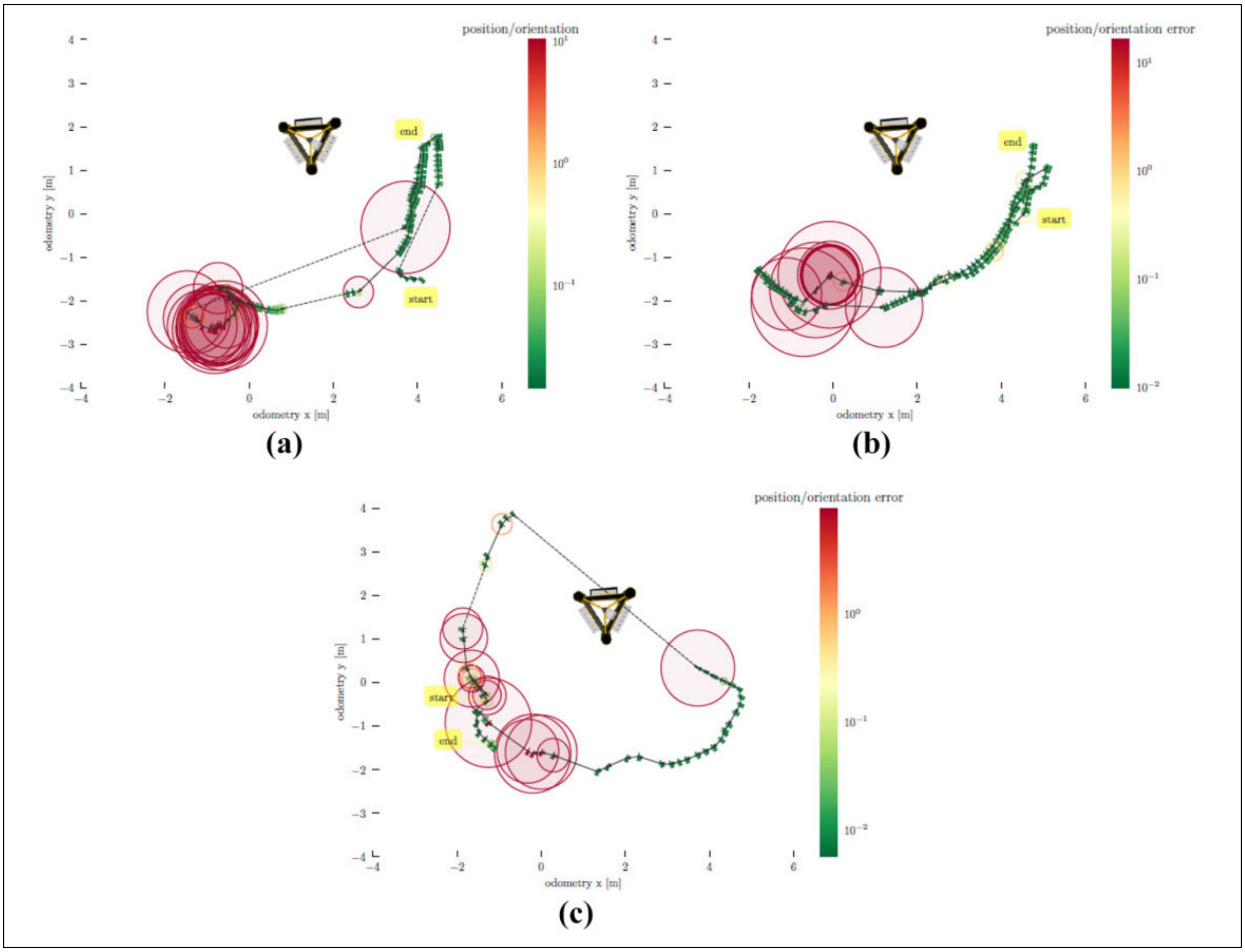

Figure 11. $\mathcal{T}_{L1a-b}$ results: robot poses (triangles) with orientation error $d({}^{O}_{R}\bar{\mathbf{q}}_S, {}^{O}_{R}\bar{\mathbf{q}}_F)$ (triangle color) and position error $d({}^{O}_{R}\bar{\mathbf{p}}_S, {}^{O}_{R}\bar{\mathbf{p}}_F)$ (circle color, log-scaled circle radius) while the robot moves back and forth on a half-circular trajectory around the panel recorded in field trials. (a) $\mathcal{T}_{L1a}$—integration stage; before localization parameter tuning. (b) $\mathcal{T}_{L1b}$—integration stage; after localization parameter tuning. (c) $\mathcal{T}_{L1c}$—validation stage.

Validation stage: $\mathcal{T}_{L1c}$

Within the validation stage, the method was tested again on the 2018 field trials data where a circular trajectory has been recorded (see Figure11(c)). In contrary to the integration stage, a full round was performed around the panel with no markers recognized for a while (top-right part of the figure), but then the localization is quickly on track again as soon as the next marker is perceived. Together with the position and orientation errors shown in Figure 10(c), the localization on simulated sensor data is shown to be similar to the results from the integration stage. This means that the localization method itself works sufficiently well to be deployed with purely real-world data as in the next tests $\mathcal{T}_{L2-5}$.

$\mathcal{T}_{L2}$—*Real-world localization using only core navigation.* In order to get a baseline to compare the performance of the localization filter when integrating visual landmarks, only core navigation, that is, a standard approach with an EKF on DVL and IMU measurements, is used in this subtest. This is also done for tuning the use of visual markers in the EKF, because navigation sensors are not integrated in the

Table 1. Description of localization tests $\mathcal{T}_{L\{2..5\}}$.

Test	Description
$\mathcal{T}_{L2}$	EKF with real-world data and only navigation sensors
$\mathcal{T}_{L3}$	EKF with real-world data, using navigation sensors and visual markers (default parameters)
$\mathcal{T}_{L4}$	$\mathcal{T}_{L3}$, plus covariance ${}^{O}_{R}C$ of the robot pose estimates from marker detections adjustment with results from test $\mathcal{T}_{P}$, that is, using $(0.126\text{m})^2$ and $(4.6^{\circ})^2$ (see Figure 9) as diagonal values for single marker detections
$\mathcal{T}_{L5}$	$\mathcal{T}_{L4}$, plus rejection of marker pose estimates whose distance $d({}^{O}_{R}\mathbf{T}_M, {}^{O}_{R}\mathbf{T}_F)$ to the current prediction are greater than 1 m and 12°; determined from $\mathcal{T}_{L1}$ and the results are in Figure 10(a) and (b)

EKF: extended Kalman filter.

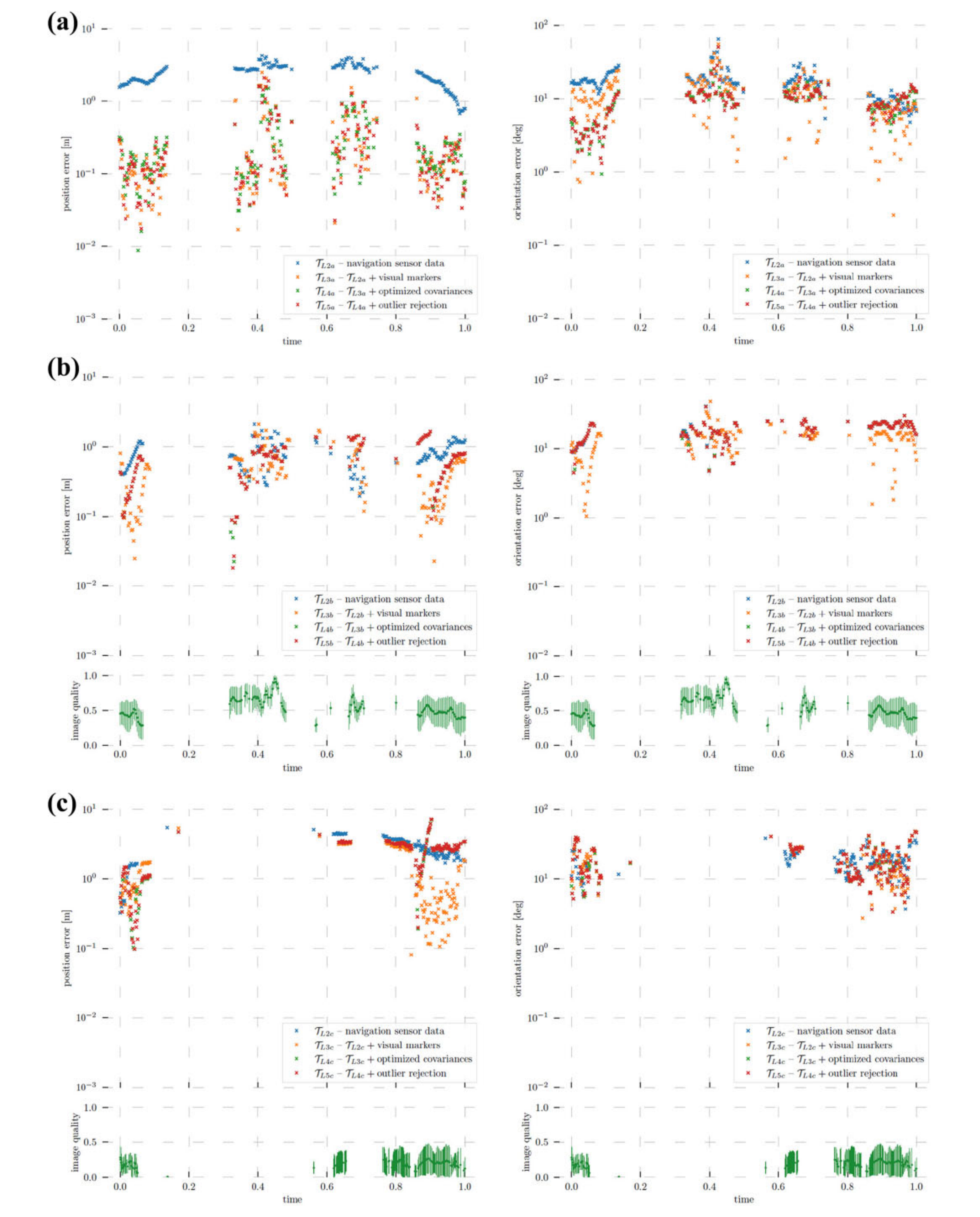

Figure 12. $\mathcal{T}_{L\{2-5\}}$ results: position error $d({}^{O}_{R}\bar{p}_M, {}^{O}_{R}\bar{p}_F)$ (left column) and orientation error $d({}^{O}_{R}\bar{q}_M, {}^{O}_{R}\bar{q}_F)$ (right column) with the relative image quality mean $\bar{m}_I(t)$ (dark-green) and standard deviation $\tilde{m}_I(t)$ in the bottom row; where the results for a test $\mathcal{T}_{Lx}$ are not visible; they coincide with $\mathcal{T}_{Ly}$ for $y > x$. (a) $\mathcal{T}_{L\{2-5\}a}$—integration stage; before localization parameter tuning. (b) $\mathcal{T}_{L\{2-5\}b}$—integration stage; after localization parameter tuning. (c) $\mathcal{T}_{L\{2-5\}c}$—validation stage.

Table 2. Tests $\mathcal{T}_{L(2-5)}$ measure results.

(a) $\mathcal{T}_{L(2-5)a}$—Integration stage, before localization parameter tuning

	$\mathcal{T}_{L2a}$	$\mathcal{T}_{L3a}$	$\mathcal{T}_{L4a}$	$\mathcal{T}_{L5a}$
$\bar{m}_{M,F}(\mathcal{T}_{Li}\langle\bar{p}\rangle)$ (m)	2.10 ± 0.95	0.26 ± 0.40	0.29 ± 0.32	0.28 ± 0.36
$\bar{m}_{M,F}(\mathcal{T}_{Li}\langle\bar{q}\rangle)$ (°)	15.59 ± 7.33	10.24 ± 7.57	8.82 ± 5.16	8.86 ± 5.19
$m_A(\mathcal{T}_{Li})$	0.95	0.72	0.91	0.94

(b) $\mathcal{T}_{L(2-5)b}$—Integration stage, before localization parameter tuning

	$\mathcal{T}_{L2b}$	$\mathcal{T}_{L3b}$	$\mathcal{T}_{L4b}$	$\mathcal{T}_{L5b}$
$\bar{m}_{M,F}(\mathcal{T}_{Li}\langle\bar{p}\rangle)$ (m)	0.81 ± 0.36	0.54 ± 0.44	0.70 ± 0.42	0.70 ± 0.42
$\bar{m}_{M,F}(\mathcal{T}_{Li}\langle\bar{q}\rangle)$ (°)	18.83 ± 5.63	13.78 ± 6.56	18.65 ± 5.77	18.66 ± 5.79
$m_A(\mathcal{T}_{Li})$	0.92	0.91	0.93	0.93

(c) $\mathcal{T}_{L(2-5)c}$—Validation stage

	$\mathcal{T}_{L2c}$	$\mathcal{T}_{L3c}$	$\mathcal{T}_{L4c}$	$\mathcal{T}_{L5c}$
$\bar{m}_{M,F}(\mathcal{T}_{Li}\langle\bar{p}\rangle)$ (m)	2.82 ± 1.10	1.43 ± 1.20	2.44 ± 1.32	2.44 ± 1.33
$\bar{m}_{M,F}(\mathcal{T}_{Li}\langle\bar{q}\rangle)$ (°)	17.95 ± 6.32	17.65 ± 9.29	17.85 ± 9.43	17.81 ± 9.48
$m_A(\mathcal{T}_{Li})$	0.9	0.9	0.9	0.91

simulation at this development stage. The results of this subtest are depicted and described together with the following subtests in the next subsection.

$\mathcal{T}_{L3-5}$—***Real-world localization with visual markers.*** In these tests, we show the localization results using all sensor data recorded in field trials along with visual marker-based pose estimates. A description of the respective tests is given in Table 1. Figure 12 shows experimental results. Since no ground truth is available in this real-world underwater scenario and all errors are given with respect to the ROV pose as recognized from the visual markers, during parts of the trajectory where no markers have been perceived, no error measures can be computed.

As expected, the EKF instance with only core navigation sensors as an input ($\mathcal{T}_{L2}$—blue in Figures 12(a) and (b)) bears the largest error. Integrating the visual markers ($\mathcal{T}_{L3}$—orange) significantly reduces the error. But it increases the number of jerky motion estimates while navigating; this is due to the fact that marker input is not always available. This hence leads to the worst $m_A(\mathcal{T}_{Li})$. Finally, we show that, based on results from the previous tests $\mathcal{T}_P$ and $\mathcal{T}_{L1}$, the localization performance can be optimized by adjusting the pose estimates covariances ($\mathcal{T}_{L4}$—green) and by rejecting outliers (red). $\mathcal{T}_{L4}$ and $\mathcal{T}_{L5}$ yield only very slightly lower accuracies, but the latter achieves by far the smoothest navigation trajectory, measurable by the higher autocorrelation $m_A(\mathcal{T}_{Li})$.

This means that the vehicle localization does not get lost as often as with a lower $m_A(\mathcal{T}_{Li})$. This is desirable as, during autonomous navigation for intervention, loss of localization should be avoided under all circumstances in order to not crash into any obstacles. The slightly decreasing pose accuracy in the later benchmarking tests is due to the lower relative image quality in the far distances from the visual landmark as visible in the bottom row of Figure 12(b). As can be compared with the trajectory in Figure 11(b), the regions where the ROV is located far away from the panel lead to the largest pose errors due to the low relative image quality ($\bar{m}_I(t) \leq 0.5$) in the real-world data. However, in near-distance regions like for $0.3 < t < 0.5$, the relative image quality is much more favorable. This overall performance is taken into account in this use case in favor of smooth trajectories.

The trajectory recorded in the validation stage (see Figure 11(c)), in contrary to the one of the integration stage, leads around the whole testing panel, covering also the parts where less visual markers are placed. In addition, the visibility and hence marker detection percentage with respect to the trajectory length was lower on the days when this trial happened. This is easily visible in the relative image quality mean ($\bar{m}_I(t) < 0.3$ for the whole regarded data, Figure 12(c) bottom) which is significantly lower than in the data of the integration stage (Figure 12(b) bottom). Keeping this in mind, the numerical results in Table 2 and the position and orientation errors in Figure 12(c) are competitive with respect to the ones obtained during the integration stage.

Conclusion

We presented an advanced navigation system for applications in the context of intervention missions, that is, underwater manipulation tasks, which require a high amount of precision and robustness in real time from the navigation system. For this purpose, a core navigation system in form of an EKF with input from a DVL and an inertial navigation system is extended by visual odometry using artificial markers. Concretely, AR markers on a target structure are used. The presented system has a high TRL, featuring an intelligent vision system suited for deep-sea operations on a commercial off-the-shelf ROV. The system was validated among others in two intensive field tests running off the shore of Marseille, France, where a substantially increased navigation performance due to the use of the vision system was demonstrated.

Acknowledgement

The presented research was carried out in the project "Effective Dexterous ROV Operations in Presence of Communications Latencies (DexROV)".

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The project "Effective Dexterous ROV Operations in Presence of Communications Latencies (DexROV)" is supported by the European Commissions Horizon 2020 Framework Program for Research and Innovation, under the topic "Blue Growth: Unlocking the Potential of Seas and Oceans", BG6-2014 "Delivering the subsea technologies for new services at sea".

ORCID iD

Tobias Doernbach https://orcid.org/0000-0001-6488-8211
Andreas Birk https://orcid.org/0000-0003-4577-2525

References

1. Lodi Rizzini D, Kallasi F, Aleotti J, et al. Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks. *Computers and Electrical Engineering* 2017; 58: 560–571.
2. Bonin-Font F, Oliver G, Wirth S, et al. Visual sensing for autonomous underwater exploration and intervention tasks. *Ocean Eng* 2015; 93: 25–44.
3. Ridao P, Carreras M, Ribas D, et al. Intervention AUVs: the next challenge. In: *IFAC world conference*, vol. 47, pp. 12146–12159.
4. Prats M, Ribas D, Palomeras N, et al. Reconfigurable AUV for intervention missions: a case study on underwater object recovery. *Intel Serv Robot* 2012; 5(1): 19–31.
5. Garcia JC, Javier Fernandez J, Sanz RMPJ, et al. Towards specification, planning and sensor-based control of autonomous underwater intervention. In: *IFAC world congress*, vol. 44, IFAC, pp. 10361–10366.
6. Garcia JC, Fernandez JJ, Sanz PJ, et al. Increasing autonomy within underwater intervention scenarios: the user interface approach. In: *2010 IEEE international systems conference*, pp. 71–75.
7. Marani G, Choi SK, and Yuh J. Underwater autonomous manipulation for intervention missions AUVs. *Ocean Engineering, Special Issue on Autonomous Underwater Vehicles* 2009; 36(1): 15–23.
8. Krupinski S, Maurelli F, Grenon G, et al. Investigation of autonomous docking strategies for robotic operation on intervention panels. In: *OCEANS* 2008, pp. 1–10. IEEE.
9. Evans J, Redmond P, Plakas C, et al. Autonomous docking for intervention- AUVs using sonar and video-based real-time 3D pose estimation. In: *Oceans* 2003, vol. 4. pp. 2201–2210. IEEE.
10. Garrido-Jurado S, Munoz-Salinas R, Madrid-Cuevas F, et al. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* 2014; 47(6): 2280–2292.
11. Palomeras N, Vallicrosa G, Mallios A, et al. AUV homing and docking for remote operations. *Ocean Engineering* 2018; 154: 106–120.
12. Blum L, Broll W, and Muller S. Augmented reality under water. *Science News*, March 24, 2009.
13. Morales R, Keitler P, Maier P, et al. An underwater augmented reality system for commercial diving operations. In: *OCEANS* 2009, pp. 1–8. IEEE.
14. Oppermann L, Blum L, Jun-Yeong L, et al. Areef multi-player underwater augmented reality experience. In: *IEEE international games innovation conference (IGIC)*, pp. 199–202.
15. Dudek G, Sattar J, and Xu A. A visual language for robot control and programming: a human-interface study. In: *IEEE international conference on robotics and automation (ICRA)*, pp. 2507–2513.
16. Speers A, Topol A, Zacher J, et al. Monitoring underwater sensors with an amphibious robot. In: *2011 Canadian conference on computer and robot vision*, pp. 153–159. IEEE.
17. Plotnik AM, and Rock SM. Visual servoing of an rov for servicing of tethered ocean moorings. In: *OCEANS* 2006, pp. 1–6. IEEE.
18. Palomeras N, Penalver A, Massot-Campos M, et al. I-AUV docking and panel intervention at sea. *Sensors* 2016; 16(10): 1673.
19. Birk A, Doernbach T, Mueller C, et al. Dexterous underwater manipulation from distant onshore locations. *IEEE Robotics and Automation Magazine (RAM)* 2018; 25(4): 24–33.
20. Luczynski T, Pfingsthorn M, and Birk A. The Pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings. *Ocean Eng* 2017; 133: 9–22.
21. Luczynski T, and Birk A. Underwater image haze removal with an underwater-ready dark channel prior. In: *Oceans (Anchorage)*.
22. Doernbach T, Gomez Chavez A, Mueller CA, et al. High-fidelity deep-sea perception using simulation in the loop. In: *IFAC conference on control applications in marine systems*.
23. Shoemake K. Animating rotation with quaternion curves. In: *Conference on computer graphics and interactive techniques. SIGGRAPH '85*, pp. 245–254.
24. Chen L, Wang S, McDonald-Maier K, et al. Towards autonomous localization and mapping of AUVs: a survey. *Int J of Intelligent Unmanned Systems* 2013; 1(2): 97–120.

25. Li D, Ji D, Liu J, et al. A multi-model EKF integrated navigation algorithm for deep water AUV. *Int J Adv Robot Syst* 2016; 13(1): 3.
26. Moore T and Stouch D. A generalized extended kalman filter implementation for the robot operating system. In: *International conference on intelligent autonomous systems*.
27. Fisher NI. *Statistical analysis of circular data*. Cambridge: Cambridge University Press, 1995.
28. Huynh D. Metrics for 3D rotations: comparison and analysis. *J Math Imaging Vis* 2009; 35(2): 155–164.
29. Gu K, Lin W, Zhai G, et al. No-reference quality metric of contrast-distorted images based on information maximization. *IEEE Trans on Cybern* 2017; 47(12): 4559–4565.