An Integrated Visual Odometry System for Underwater Vehicles

Zhizun Xu, Maryam Haroutunian, Alan J. Murphy, Jeff Neasham, and Rose Norman, *Senior Member, IEEE*

***Abstract*—Underwater navigation is always a challenging problem because of electromagnetic attenuation. The traditional methods involve beacons, inertial sensors, and Doppler velocity log, but they have many shortcomings, such as high cost and lengthy setup time. In order to solve underwater navigation problems at low cost, an integrated visual odometry system has been developed and discussed in this article. In this method, two inertial sensors provide acceleration and attitude of the vehicle, and an underwater sonar is used to provide the distance between the vehicle and the seabed, whilst in the visual odometry section, an optical flow algorithm has been applied for tracking feature points. With the depth provided by the sonar, 3-D position of feature points can be calculated. Linear motion of the vehicle is then predicted through these feature points in dual frames. Finally, nonlinear optimization is used to correct the attitude of the vehicle using visual information. In the proposed algorithm, the vehicle trajectory can be estimated in absolute scale by using a single camera; computational complexity is reduced dramatically compared to other visual odometry methodologies; and this algorithm allows the approach to work in sparse texture conditions. The results from practical experiments demonstrate that the method is effective and it is also a low-cost solution.**

***Index Terms*—Sensor fusion, underwater vehicles, visual-inertial odometry.**

I. Introduction

THE oceans cover most of the earth's surface and are critical sources of food and other resources such as oil and gas. Conversely, the underwater environment can threaten the safety of human beings engaged in underwater operations. Hence, remotely operated vehicles (ROVs) are usually employed to conduct offshore oil and gas installations, and autonomous underwater vehicles (AUVs) are currently used for scientific survey tasks, oceanographic sampling, underwater archeology, and under-ice survey work [1]–[3].

Accurate localization and navigation is essential to ensure that underwater vehicles conduct these operations successfully. However, due to the rapid attenuation of electromagnetic waves in the underwater environment, navigation and localization for underwater vehicles are challenging problems. The conventional methods to solve underwater localization problems are using inertial sensors such as inertial measurement unit (IMU) sensors, acoustic beacons installed in the region of interest, and the Doppler velocity log (DVL) [4]. Some underwater vehicles are also required to rise up to the surface periodically in order to receive satellite signals. The main disadvantages of traditional navigation approaches are that they either suffer from unbounded drift, or they require external infrastructure that needs to be set up and calibrated [5].

Inertial sensors, involving accelerometers, gyroscopes, and DVL, suffer from unbounded drift errors. The performances of an inertial unit are mainly determined by the quality of its components [6]; in general, a more expensive unit has better performance. The most precise DVL device can achieve a drift of 0.1% of the distance traveled, however, a general DVL usually has a drift of about 5% of the distance traveled. Even so, the cost of most DVLs is over 20k USD [6].

Acoustic devices, such as long beacons and ultrashort baseline, require predeployed and localized infrastructure [7]. However, low bandwidth, low data rate, and variable sound speed restrict their application.

Compared with conventional methodologies, visual odometry (VO) can provide position and attitude of vehicles with extremely low cost. It can also bound position error by using simultaneous localization and mapping (SLAM) algorithms [1]. Visual navigation approaches have been applied in mobile robotics and drones [8], [9]. One well-known VO application has been on NASA's Mars exploration rovers [10].

VO algorithms try to track feature points in continuous images captured by stereo cameras or a monocular camera, and the camera pose can be determined by the motion of these tracked feature points. In this case, the VO requires as many feature points as possible to detect, so that the algorithms are able to reduce the errors introduced by falsely matched feature points in different images. Hence, most VO approaches are used for work in dense-texture environments where VO can provide accurate trajectory estimates, with relative position error ranging from 0.1% to 2% [11], [12]. However, in sparse environments or poor illumination conditions, the performance of VO is unreliable.

Most VO algorithms based on monocular cameras only provide the estimated pose at a relative scale built in the initialization. That indicates that the absolute scale is unknown if the relative scale is uncertain. Furthermore, they all require a dense-feature environment to get acceptable predictions. However, there are few feature points on the seabed under low illumination conditions. In order to overcome these limitations, a novel

Manuscript received April 28, 2020; revised July 27, 2020 and October 23, 2020; accepted November 2, 2020. Date of publication December 30, 2020; date of current version July 14, 2021. *(Corresponding author: Zhizun Xu.)*

Associate Editor: R. Diamant.

The authors are with the School of Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, U.K. (e-mail: z.xu21@newcastle.ac.uk; maryam.haroutunian@newcastle.ac.uk; a.j.murphy@newcastle.ac.uk; jeff.neasham@newcastle.ac.uk; rose.norman@newcastle.ac.uk).

Digital Object Identifier 10.1109/JOE.2020.3036710

integrated visual odometry (IVO) system has been developed in this article.

The proposed IVO method with a monocular camera is expected to be capable of replacing the high-cost traditional navigation systems. The Lucas–Kanade optical flow (OF) algorithm has been applied for tracking feature points between dual-frame images captured by an optical sensor [13]. Meanwhile an IMU development kit, with integrated signal processing, can output the acceleration and orientation of the vehicle. Linear and nonlinear methods have been utilized to collect information from the multiple sensors and then predict the pose and the trajectory of the vehicle. Such a methodology is able to work in sparse texture environments, it operates with low computational complexity, and it estimates trajectory in absolute scale even though it only uses a monocular camera as the optical sensor. Practical experiments are reported to verify the methodology.

This article is organized as follows. Section II reviews previously reported recent research on VO and visual SLAM. Section III provides details of the underwater vehicle used to capture the data, and an important assumption made in the work. Section IV reports the geometry transformation between the coordinates of inertial sensors, sonar, and the monocular camera. Section V introduces the details of the proposed method. Section VI describes the implementation of the proposed method, experiments, results, and discussion. Section VII presents the conclusion of the work.

II. Review of Related Work in Underwater Visual Odometries

VO and visual SLAM algorithms have been successfully applied in mobile robotics and aerial robotics [12]. The well-known visual SLAM methods include parallel tracking and mapping (PTAM) [14], dense tracking and mapping [15], and large-scale direct (LSD-SLAM) [16]. Fast semi-direct monocular visual odometry (SVO) has been developed for air drones equipped with downward-looking cameras [17]. It operates directly on pixel intensities, which results in subpixel precision at high frame-rates. In this system, a probabilistic mapping method that explicitly models outlier measurements is used to estimate 3-D points.

The ORB-SLAM algorithm is a feature-based method and was developed based on PTAM. It is a reliable and complete solution for monocular SLAM [18]. It uses the same feature points for all tasks including the tracking, mapping, relocalization, and loop closing.

Direct sparse odometry is developed based on LSD-SLAM. It is a VO method developed from a novel, highly accurate, sparse and direct structure, and motion formulation [19]. It combines a fully direct probabilistic model (minimizing the photometric error) with consistent, joint optimization of all model parameters.

Engel's research has shown that visual SLAM can provide accuracy, low cost, and bounded position error navigation methods. However, all of these methods are pure visual SLAM. The performances of these SLAM methods depend on the density of feature points. They all require the use of global shuttering cameras with long focal-length, wide angle lenses, to provide a wider view. Such lenses can enable the camera to catch more feature points; SLAMs usually require at least 50 feature points to estimate the cameras' motion. In fact, in the SVO method, at least 100 feature points are required in the initialization section.

In addition, these SLAM methods are suitable for slow vehicle motion. High frame-rate global shuttering cameras may give enough images, but the computing time of the SLAM algorithm may lag the response of the navigation systems due to onboard computers having limited computation abilities. For monocular SLAMs, the 3-D positions of feature points are obtained from triangular calculations. Hence, if a single camera is subject to pure rotational motion, the triangular calculation lacks translation information, and the monocular SLAM algorithm will fail. In order to improve the performance of pure visual SLAMs, Leutenegger, and Qin developed inertial-visual odometries (OKVIS and VINE-Mono) for air drone navigation [20], [21]. In their work, a cost function, constructed by summing reprojection error and inertial sensor error, is minimized to solve for camera pose.

In [22], Eustice reports a visual navigation system for underwater vehicles, called visually augmented navigation (VAN). The multisensor fusion filter integrates the benefits of optical and inertial navigation methods and is robust to low overlap of imagery [6]. The filter is developed in a version-based form, based on the extended Kalman filter (EKF), where pose is estimated by VO. Actually, VO provides constraints for estimation of inertial methods. Eustice also applies an information filter to replace the EKF filter, and the results show an improvement in accuracy. This approach was applied to underwater exploration in the surveying of RMS Titanic [23]. In [24], the VAN method was used to inspect ship-hulls for the U.S. Navy. Both [25] and [26] were extensions of the VAN method, a smoothing and mapping problem formulation and efficient matrix factorizations are proposed to be able to efficiently recover the mean and covariance values. The VAN algorithm has also been introduced into image sonar with pose graph methods applied to predict pose and landmarks by Hover [27]. In [28], Kim proposed a novel approach to estimate the trajectory of a vehicle using DVL and a VO in a poor environment context. Based on Eustice's idea, Li developed a pose-graph SLAM using forward-looking sonar to estimate trajectories [29].

Some stereo VO approaches using higher frame rate videos (10–20 Hz) to estimate underwater vehicle pose have been presented recently. In [30], features are matched within stereo pairs to compute 3-D point clouds and the camera poses are estimated by aligning these successive point clouds, making it a pure stereo vision method. In parallel, the work of Bellavia uses a keyframe-based approach but their feature tracking is carried out by matching descriptors both spatially (between stereo image pairs) and temporally [31].

Recent parallel work has been done by Maxime [32] and Sharmin [33], [34]. Maxime developed a real time Monocular VO system for the underwater environment. It uses the OF algorithm to track feature points and the depth information is obtained through triangulation calculations. Sharmin extended the open keyframe-based visual-inertial SLAM (OKVIS) [20]

with underwater profiling sonar. The method fuses multiple information from a stereo camera, a profiling sonar, an IMU and a pressure sensor by using a tightly coupled nonlinear optimization. More specifically, Sharmin derived a cost function by summing the reprojection error, the IMU error, and the sonar error. Because of that, camera pose estimation and mapping of underwater structures are processed simultaneously by minimizing the cost function. Sharmin improved the method by applying a robust initialization method, an image enhanced technique, and a loop-closure technique in [34]. However, these methods were not tested by quantitative evaluation methods in the underwater environment. This means that the accuracy or drift errors of these methods are not clearly understood in underwater navigation applications.

Compared with conventional underwater navigation, the current research is a relatively low-cost navigation solution with acceptable accuracy. Unlike other visual or inertial-visual navigation methods, the proposed method reconstructs 3-D positions of feature points by using a monocular camera and a ping sonar. The work is based on a main assumption: the seabed is locally flat. First, the 3-D positions of feature points in camera coordinates are identified efficiently with the rotation matrix obtained from a low cost inertial unit and depth from an underwater sonar. Second, the translation vector is bounded from the OF algorithm. Finally, a nonlinear optimization solver is used to correct the attitude from inertial sensors and give the optimal incremental motion between two frames. Therefore, the computational complexity of the methodology is relatively light. The novelty of the method is that it is able to localise the vehicle in underwater feature-sparse environments with 3% to 4% drift error, while other algorithms perform unsuccessfully.

III. Assumption and Data Collection

A. Assumption of the Proposed Method

The novel IVO method is based on one main assumption.

Assumption 1 (The Seabed is Locally Flat): In the IVO method, the optical sensor is used to identify feature points on the seabed. In actual situations, underwater vehicles could maintain constant distance from the seabed, and peaks and troughs caused by small objects on the seabed can be ignored.

B. Data Collection Vehicle

The data collection vehicle is a modified VideoRay Pro 3. A waterproof tube with multiple precise sensors inside has been built and installed on the bottom of the original vehicle. The main sensors involved are an IMU development kit with integrated signal processing, which is expected to provide orientation data, an optical sensor (global shutter camera), which is able to capture grayscale images when the vehicle is in motion, a gyroscope, a sonar, and an Intel RealSense T265 Tracking Camera. The gyroscope can offer the yaw angle, which is not affected by magnetic fields. The sonar is to detect the distance between seabed and the vehicle. The T265 Tracking Camera is installed on the electronics tray in order to provide comparative results. The T265 consists of a stereo camera, an IMU senor and a processor, which can provide the trajectory of the camera directly using its own V-SLAM algorithm. Hence, the results from proposed IVO, the T265 and the other open source visual SLAMs mentioned in Section II are compared.

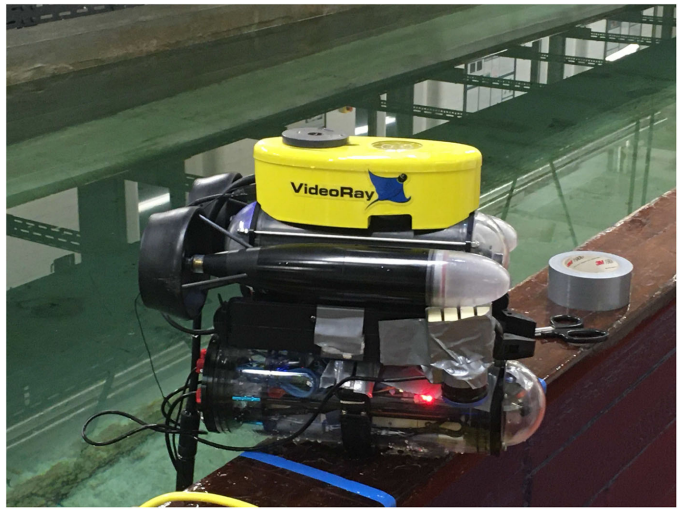


Fig. 1. VideoRay Pro 3 with additional tube.

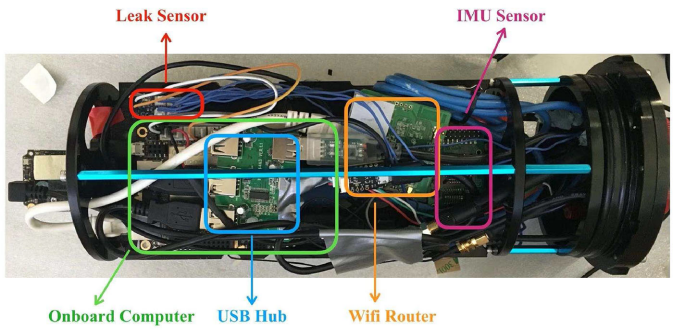


Fig. 2. Top view: Sensors on electronics tray.

Fig. 3. Back view: Sensors on electronics tray.

The underwater vehicle with the additional tube is shown in Fig. 1. The sensors on the tray inside the tube are shown in Figs. 2 and 3. The main measurement sensors and their costs are listed in Table I, illustrating that the proposed IVO method is implemented on low-cost hardware.

TABLE I
SENSORS LIST

Sensors	Model	Cost
Global Camera	mvBlueFOX-MLC 200w	£200
Underwater Sonar	BlueRobotics Ping Sonar	£100
Inertial Measure Unit	LPMS-ME1	£50
Gyroscope	LPMS-NV2	£100
Visual Odometry(Stereo Camera)	Intel RealSense T265	£179

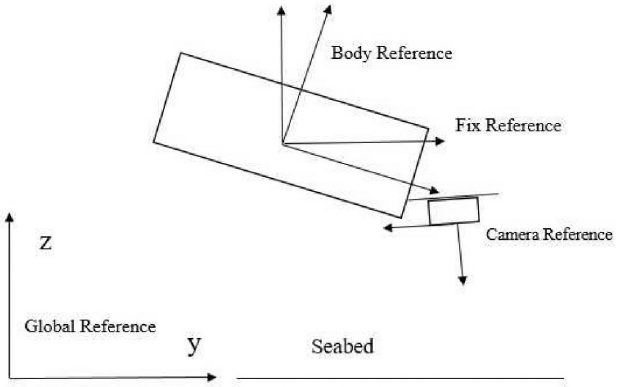


Fig. 4. Fixed reference and body reference.

IV. GEOMETRY TRANSFORMATION

As detailed in the previous section, there are multiple sensors located on the electronics tray. The sensors, including the IMU, gyroscope, sonar, and camera, each have their own coordinates. Hence, transformation matrices ($\boldsymbol{T}$) are needed to transfer the value measured in the sensors' coordinates to the vehicle body's reference. The transformation matrix belongs to the special Euclidean group, and has three dimensions. It can be written as, $\boldsymbol{T} \in \boldsymbol{SE(3)}$, and $\mathbb{R}^3 \times \boldsymbol{SO(3)} \to \boldsymbol{T}$. In this section, the transformation matrices for the IMU, gyroscope, sonar, and camera are discussed in order to make sure that the information is combined in the same reference frame. Let $\boldsymbol{R}$ be the rotation matrix and $\boldsymbol{t}$ be the translation vector. The coordinates of the electronics tray are called the body reference.

In order to compute the depth of the image pixel using depth information from the sonar, a fixed reference has been created. The rotation in the reference is the same as the rotation relative to the global reference, but the translational vector is zero related to the body reference, as shown in Fig. 4.

The transformation matrix from the body reference to the fixed reference is

$$^{\text{fix}}\boldsymbol{T}_{\text{body}} = \begin{bmatrix} ^{\text{global}}\boldsymbol{R}_{\text{body}} & \boldsymbol{0} \\ 0 & 1 \end{bmatrix}. \tag{1}$$

The transformation matrix will be used in the methodology section to calculate the linear translation. The other coordinates transformation are discussed in the following sections.

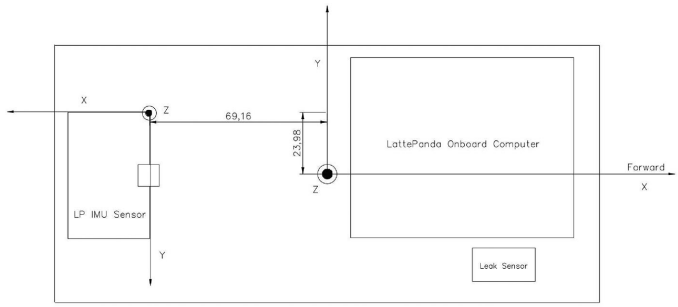


Fig. 5. Top tray (unit:mm).

A. IMU Development Kit

LPMS-ME1 is a low-cost IMU development kit with integrated signal processing, which can provide a quaternion and accelerations on *xyz* axes. The quaternion is used to describe the attitude of the IMU relative to the global reference. For computational convenience, the quaternion will be converted into a rotation matrix $\boldsymbol{R}$. Similar to $\boldsymbol{T}$, the rotation matrix $\boldsymbol{R}$ is of a special orthogonal group. Because it is in three dimensions, it is written as $\boldsymbol{R} \in \boldsymbol{SO(3)}$. The location of the IMU kit is shown in Fig. 5.

The transformation matrix from IMU reference to body reference is presented as

$$^{\text{body}}\boldsymbol{T}_{\text{imu}} = \begin{bmatrix} ^{\text{body}}\boldsymbol{R}_{\text{imu}} & ^{\text{body}}\boldsymbol{t}_{\text{imu}} \\ 0 & 1 \end{bmatrix} \tag{2}$$

$$^{\text{body}}\boldsymbol{R}_{\text{imu}} = \boldsymbol{R}_z(\pi) \tag{3}$$

$$^{\text{body}}\boldsymbol{t}_{\text{imu}} = [-0.069 \;\; 0.024 \;\; 0.003]. \tag{4}$$

The units are meters; $\boldsymbol{R}_z(\pi)$ indicates that the reference is rotated by 180° in the anticlockwise direction along the z-axis. Therefore, assuming $\boldsymbol{R}_{\text{measure}}$ is the rotation matrix related to the world reference, $\boldsymbol{R}_{\text{body}}$, the body reference relative to the global reference is

$$^{\text{global}}\boldsymbol{R}_{\text{body}} = {}^{\text{global}}\boldsymbol{R}_{\text{measure}} \, {}^{\text{body}}\boldsymbol{R}_{\text{imu}}^{-1}. \tag{5}$$

B. Gyroscope

LPMS-NAV2 is an inertial sensor for navigation applications, which is composed of a high accuracy one-axis gyroscope and a three-axis accelerometer [35]. It can achieve precise heading information with ultralow drift error. The location of the LPMS-NAV2 is presented in Fig. 6.

The transformation matrix from the NAV2 reference to the body reference is presented as

$$^{\text{body}}\boldsymbol{T}_{\text{nav}} = \begin{bmatrix} ^{\text{body}}\boldsymbol{R}_{\text{nav}} & ^{\text{body}}\boldsymbol{t}_{\text{nav}} \\ 0 & 1 \end{bmatrix} \tag{6}$$

$$^{\text{body}}\boldsymbol{R}_{\text{nav}} = \boldsymbol{R}_z\left(-\frac{\pi}{2}\right)\boldsymbol{R}_x(\pi) \tag{7}$$

$$^{\text{body}}\boldsymbol{t}_{\text{nav}} = [-0.05733 \;\; -0.00309 \;\; -0.015]. \tag{8}$$

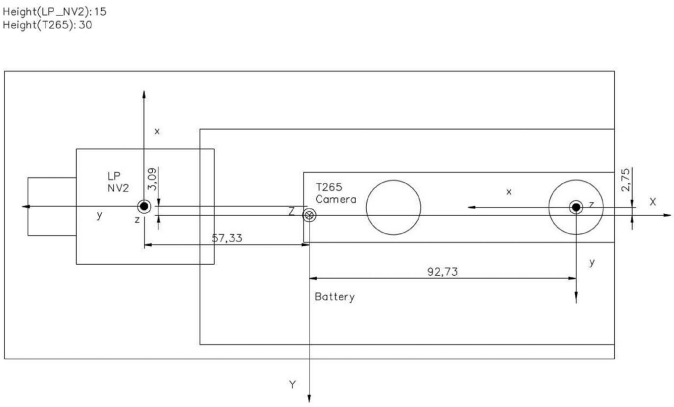


Fig. 6. Bottom tray (unit:mm).

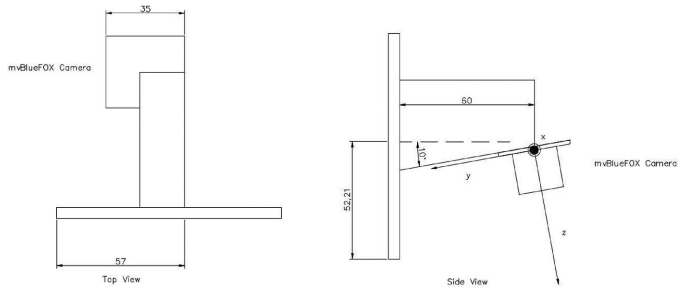


Fig. 7. Camera location (unit:mm).

Because the NAV2 only provides the heading angle related to the global reference, the rotation matrix from the IMU is decomposed into Euler angles (yaw θ, roll ϕ, and pitch ψ). Then, the yaw from the NAV2 is transferred to the body reference and will replace the yaw from the IMU to generate a new rotation matrix combining the IMU kit and Gyroscope information. Because the range of pitch of the vehicle in motion is $\pm\pi/2$, the Gimbal lock problem is not considered.

C. T265 Camera

The Intel RealSense T265 Tracking Camera is designed to give tracking performance using two fisheye lense sensors, an IMU, and an Intel VPU (visual processing unit). All of the V-SLAM algorithms run directly on the VPU, which means the estimated trajectory can be obtained directly from T265. It is stated that the T265 can provide less than 1% closed loop drift under intended use conditions [36]. The T265 has been implemented in mobile robotics projects and drone competitions to provide the navigation information [37]–[39]. In [40], the T265 camera was used in conjunction with a depth camera to build an environment map, and in the RoboSub 2019 Competition a T265 camera was used to localize an underwater vehicle [41]. However, Bekawi did not provide experimental results for underwater navigation by T265 camera. The calibration of the T265 camera is conducted on the production line and the intrinsic and extrinsic parameters of the camera can be obtained manually through RealSense SDK. The V-SLAM program running on the Intel VPU of the T265 camera can access these parameters directly. However, although underwater calibration was conducted, the resulting camera parameters cannot be used by the T265 V-SLAM program.

The installation of T265 is shown in Fig. 6. The transformation matrix from T265 reference to body reference is presented as

$$^{\text{body}}\boldsymbol{T}_{t265} = \begin{bmatrix} ^{\text{body}}\boldsymbol{R}_{t265} & ^{\text{body}}\boldsymbol{t}_{t265} \\ 0 & 1 \end{bmatrix} \tag{9}$$

$$^{\text{body}}\boldsymbol{R}_{t265} = \boldsymbol{R}_z(\pi)\boldsymbol{R}_x(\pi) \tag{10}$$

$$^{\text{body}}\boldsymbol{t}_{t265} = [0.09273 \quad -0.00275 \quad -0.03]. \tag{11}$$

The estimated trajectory of the T265 can be transferred to the body reference using $\boldsymbol{T}_{t265}$, hence, it can provide comparative results relative to the results from the proposed IVO method.

D. Monocular Camera

The mvBlueFOX-MLC 200w is a global shutter grayscale camera, located on the tray as shown in Fig. 7. The camera can give 752×480 resolution images with 60 fps (frames per second). A wide-angle lense is used on the camera, and a perspective model is adopted to calibrate the camera.

The transformation matrix from monocular camera reference to body reference is presented as

$$^{\text{body}}\boldsymbol{T}_{\text{cam}} = \begin{bmatrix} ^{\text{body}}\boldsymbol{R}_{\text{cam}} & ^{\text{body}}\boldsymbol{t}_{\text{cam}} \\ 0 & 1 \end{bmatrix} \tag{12}$$

$$^{\text{body}}\boldsymbol{R}_{\text{cam}} = \boldsymbol{R}_z\left(-\frac{\pi}{2}\right)\boldsymbol{R}_x\left(\pi + \frac{\pi}{18}\right) \tag{13}$$

$$^{\text{body}}\boldsymbol{t}_{\text{cam}} = [0.166 \quad 0.024 \quad 0.003]. \tag{14}$$

Therefore, feature points in camera coordinates can be transferred to the body coordinates by using $^{\text{body}}\boldsymbol{T}_{\text{cam}}$.

E. Underwater Sonar

The Ping sonar is a multipurpose single-beam echosounder. It can be used as an altimeter for ROVs and AUVs, for bathymetry work aboard a USV, as an obstacle avoidance sonar, and for other underwater distance measurement applications. The range of the sonar is from 0.5 to 30 m, and the beamwidth is 30°.

The sonar has been installed on the outside of the watertight tube. The location is shown in Fig. 8.

Compared to the other sensors' transformation matrices, the transformation matrix for the sonar is much simpler because the two references, body and sonar, share identical rotation vectors and only have translation bias. Hence

$$^{\text{body}}\boldsymbol{T}_{\text{sonar}} = \begin{bmatrix} ^{\text{body}}\boldsymbol{R}_{\text{sonar}} & ^{\text{body}}\boldsymbol{t}_{\text{sonar}} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{15}$$

$$^{\text{body}}\boldsymbol{R}_{\text{sonar}} = \boldsymbol{I}. \tag{16}$$

The rotation matrix is the identity matrix and the translation vector is

$$^{\text{body}}\boldsymbol{t}_{\text{sonar}} = [0.060, 0.070, 0.065]^T. \tag{17}$$

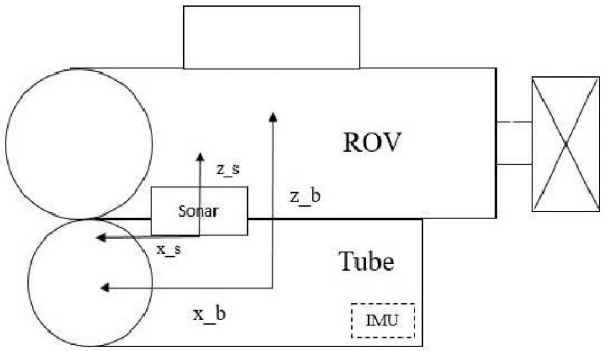


Fig. 8. Sonar location.

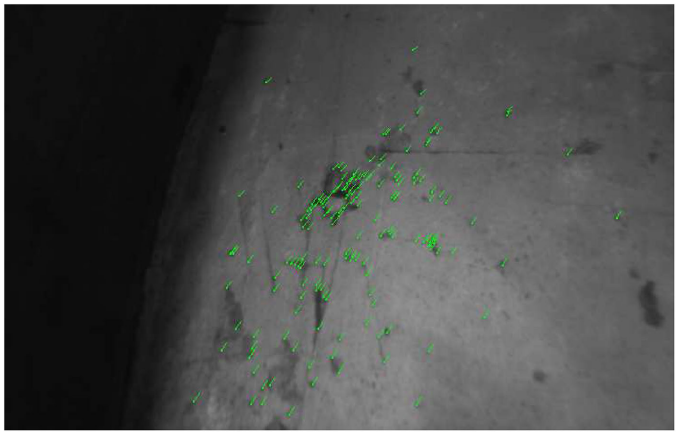


Fig. 10. Feature points extracted from underwater image.

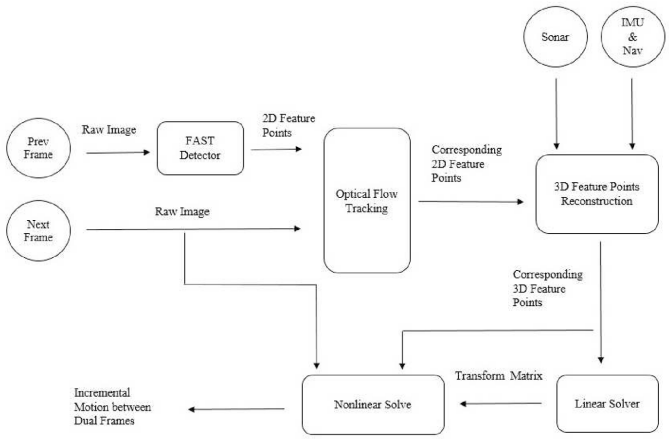


Fig. 9. IVO approach.

Through the $\boldsymbol{T}_{\text{sonar}}$ transformation, the depth from sonar to seabed becomes the depth from the body to seabed.

V. Methodology

In this section, the methodology of the proposed IVO is presented in detail. The IVO is based on a dual-frame VO algorithm. The incremental motion between two frames is predicted and the incremental motions are accumulated to make up a whole trajectory. In the first step, the FAST Corner detector (see subsection A) is applied to extract the feature points from the previous frame. Second, the OF algorithm is employed to find the corresponding feature points in the next frame. Third, the 3-D positions of the feature points in the two different frames are estimated using depth measured by the sonar. Linear motion is then predicted using attitude from the inertial sensors, i.e., the IMU kit and compass. Finally, linear motion and attitude are selected as initial values to minimize the projected error by the Levenberg Marquardt (LM) algorithm. The process of the IVO method is presented in Fig. 9.

In ideal situations, the principal axis of the camera is strictly vertical to the seabed. Hence, the relationship between altitude and velocity is approximately given by $h = v/q$, where h is the height, v denotes the horizontal forward velocity, and q indicates the magnitude of the OF vector in response to translational movement [42]. In practice, the principal axis of the camera will not be perfectly vertical to the seabed, due to camera installation, and roll and pitch motions of the vehicle.

Through tracked feature points, the orientation of the current frame relative to the previous frame can be solved. In the conventional algorithm, at least eight points are needed to obtain the fundamental matrix. However, in sparse texture environments, the result becomes unreliable. In the proposed method, the orientation of the vehicle can be determined from the IMU sensors, and height of the vehicle above the seabed/tank bottom can also be obtained by the sonar. With these two constraints, the translational vector can be calculated from a single feature point.

A. FAST Corner Detector

The FAST corner detector is applied for extracting feature points from the image [43]. There are different kinds of feature detectors such as SIFT [44], Harris [45], and ORB [46] which can provide high quality feature information, but they are too computationally intensive for use in real-time applications [43]. According to [12], the FAST algorithm is much more computationally efficient and has acceptable robustness. Another key point is that it can detect many more feature points in sparse environments compared to other algorithms.

Because the FAST algorithm is not robust to high levels of noise, the images captured from the camera are processed by a Gaussian blur filter in order to remove the noise in the image. The feature points extracted from an example image are shown in Fig. 10.

B. OF Tracking

The OF algorithm, used to track the apparent motions of brightness across a series of images, relies on the approximation of motion fields computed from a set of image sequences. The algorithm is based on the brightness constancy assumption that the intensity of the pixel remains the same despite small changes of position over time [47], which is described by

$$\nabla I(\boldsymbol{x}, t)\boldsymbol{u} + I_t(\boldsymbol{x}, t) = 0. \tag{18}$$

Here, ∇I and I_t denote spatial and temporal partial derivatives of the image I, and $\boldsymbol{u}$ denotes the 2-D velocity [48].

After seabed images are captured by an optical sensor, feature points are extracted from these images and then the OF algorithm is applied for tracking these points on the next frame. Hence, the correspondence of feature points is determined.

C. Positions of Feature Points

The 3-D positions of feature points are reconstructed using the depth from the sonar. Assuming that the seabed is flat planar, the one parameter λ equation is

$$\boldsymbol{X}_{\text{fix}}(\lambda) = {}^{\text{fix}}\boldsymbol{T}_{\text{body}}\ {}^{\text{body}}\boldsymbol{T}_{\text{camera}}\boldsymbol{X}_c(\lambda). \tag{19}$$

Here, λ is the depth of the corresponding pixel, $\boldsymbol{X}_c(\lambda)$ is a feature point position in camera coordinates, and $\boldsymbol{X}_{\text{fix}}(\lambda)$ is the feature point position in fixed coordinates. The definition of ${}^{\text{fix}}\boldsymbol{T}_{\text{camera}}$ is

$${}^{\text{fix}}\boldsymbol{T}_{\text{camera}} = {}^{\text{fix}}\boldsymbol{T}_{\text{body}}\ {}^{\text{body}}\boldsymbol{T}_{\text{camera}}. \tag{20}$$

The depth can be described in sonar coordinates by

$$\boldsymbol{d}_{\text{sonar}} = [0, 0, z_{\text{sonar}}, 1]^T. \tag{21}$$

In the fixed coordinates, $\boldsymbol{d}_{\text{sonar}}$ becomes

$$\boldsymbol{d}_{\text{fix}} = {}^{\text{fix}}\boldsymbol{T}_{\text{body}}\ {}^{\text{body}}\boldsymbol{T}_{\text{sonar}}\boldsymbol{d}_{\text{sonar}} \tag{22}$$

Therefore, because the seabed is assumed to be flat planar and the distance to the seabed and feature points are both in fixed coordinates, they share the same depth from the x-plane to the seabed. This should be satisfied by the equation

$$\boldsymbol{X}_{\text{fix}}(\lambda)^3 = \boldsymbol{d}_{\text{fix}}^3. \tag{23}$$

Here, $\boldsymbol{X}_{\text{fix}}(\lambda)^3$ and $\boldsymbol{d}_{\text{fix}}^3$ mean the third element in $\boldsymbol{X}_{\text{fix}}$ and $\boldsymbol{d}_{\text{fix}}$, respectively. Using this condition, λ is solved according to (22) and the perspective camera model

$$\boldsymbol{d}_{\text{fix}}^3 = \lambda[r_{31}, r_{32}, r_{33}]\boldsymbol{K}^{-1}\boldsymbol{X}_p + t_{34}. \tag{24}$$

Here, r_{31}, r_{32}, and r_{33} are the 1st, 2nd, and 3rd elements in the third row of ${}^{\text{fix}}\boldsymbol{T}_{\text{camera}}$, t_{34} is the 4th element in the third row of the matrix and $\boldsymbol{K}$ is the camera matrix.

Finally, according to (24) the scalar λ should be

$$\text{den} = r_{31}\frac{x_p - p_x}{f_x} + r_{32}\frac{y_p - p_y}{f_y} + r_{33} \tag{25}$$

$$\lambda = \frac{\boldsymbol{d}_{\text{fix}}^3 - t_{34}}{\text{den}}. \tag{26}$$

According to (19), the position of a feature point in camera coordinates can be derived with known λ. Hence, the positions for 3-D feature points in camera coordinates are gained through one image from the camera and depth information from the sonar.

D. Linear Estimation

As mentioned before, the proposed IVO is based on a dual-frame algorithm. The incremental linear motion is predicted between one frame and the next frame. Feature points between the two frames are extracted by the FAST algorithm and reconstructed in three dimensions.

Assuming $\boldsymbol{P}$ is the 3-D position of a feature point in one frame, and $\boldsymbol{P}'$ is the same feature point in the next frame, the rotation matrix $\boldsymbol{R}_\Delta$ and translation $\boldsymbol{t}$ satisfy

$$\boldsymbol{P}' = \boldsymbol{R}_\Delta\boldsymbol{P} + \boldsymbol{t}. \tag{27}$$

The rotation matrix $\boldsymbol{R}_\Delta = {}^{\text{next}}\boldsymbol{R}_{\text{prev}}$ indicates the rotation from one frame to the next frame caught by the camera. The ${}^{\text{global}}\boldsymbol{R}_{\text{imu}}$ is obtained from the IMU kit directly. Because the points' positions are described in camera coordinates, the pose change should be mapped from the IMU to the camera

$${}^{\text{imu}}\boldsymbol{R}_{\text{camera}} = {}^{\text{body}}\boldsymbol{R}_{\text{imu}}^{-1}\ {}^{\text{body}}\boldsymbol{R}_{\text{camera}} \tag{28}$$

$${}^{\text{global}}\boldsymbol{R}_{\text{camera}} = {}^{\text{global}}\boldsymbol{R}_{\text{imu}}\ {}^{\text{imu}}\boldsymbol{R}_{\text{camera}} \tag{29}$$

$${}^{\text{next}}\boldsymbol{R}_{\text{prev}} = {}^{\text{global}}\boldsymbol{R}_{\text{next}}^{-1}\ {}^{\text{global}}\boldsymbol{R}_{\text{prev}}. \tag{30}$$

The translation vector can be described by

$$\boldsymbol{t}_\Delta = \boldsymbol{P}' - \boldsymbol{R}_\Delta\boldsymbol{P} \tag{31}$$

$$\boldsymbol{T}_\Delta = {}^{\text{next}}\boldsymbol{T}_{\text{prev}} = \begin{bmatrix}\boldsymbol{R}_\Delta & \boldsymbol{t}_\Delta \\ \boldsymbol{0} & 1\end{bmatrix}. \tag{32}$$

Assuming that there is a series of images from time t_1 to t_n, ${}^{t_n}\boldsymbol{T}_{t_1}$ can be derived as

$${}^{t_n}\boldsymbol{T}_{t_1} = {}^{t_n}\boldsymbol{T}_{t_{n-1}}\ {}^{t_{n-1}}\boldsymbol{T}_{n-2}\ldots{}^{t_2}\boldsymbol{T}_{t_1}. \tag{33}$$

Generally, the pose of the camera at t_1 is set as an origin of the trajectory. In this case, the pose of camera at t_n in the global coordinates is

$${}^{\text{global}}\boldsymbol{T}_{t_n} = {}^{t_n}\boldsymbol{R}_{t_1}^{-1}. \tag{34}$$

1) Robust Estimation: Assuming there are N feature points, the error function can be written as

$$E_p = \sum_{n=1}^{N} ||\boldsymbol{t} - (\boldsymbol{P}'_n - \boldsymbol{R}\boldsymbol{P}_n)||_2^2. \tag{35}$$

The minimum E_p with respect to the translation vector $\boldsymbol{t}$ will be

$$\boldsymbol{t}_{\text{min}} = \frac{\sum_{n=1}^{N}(\boldsymbol{P}'_n - \boldsymbol{R}\boldsymbol{P}_n)}{N}. \tag{36}$$

2) Error Analysis: Assuming there is white noise on the pixel coordinates and depth measured by the sonar

$$\hat{x}_p = x_p + \epsilon_{xp} \tag{37}$$

$$\hat{y}_p = y_p + \epsilon_{yp} \tag{38}$$

$$\hat{f}_{\text{fix}}^3 = f_{\text{fix}}^3 + \epsilon_f. \tag{39}$$

where ϵ_{xp}, ϵ_{yp}, and ϵ_f are white noise. The variances are σ_{xp}, σ_{yp}, and σ_f, and the means are all zero.

The λ matrix was obtained through (22) and (24). The Jacobian matrix for λ with respect to pixel coordinates and depth, is

derived as follows:

$$\boldsymbol{J}_\lambda = \left[\frac{\partial\lambda}{\partial x_p} \quad \frac{\partial\lambda}{\partial y_p} \quad \frac{\partial\lambda}{\partial f^3_{\text{fix}}}\right] \tag{40}$$

$$\boldsymbol{C}_\lambda = \begin{bmatrix} \sigma_{xp} & 0 & 0 \\ 0 & \sigma_{yp} & 0 \\ 0 & 0 & \sigma_{f^3_{\text{fix}}} \end{bmatrix}. \tag{41}$$

Therefore, σ_λ is

$$\sigma_\lambda = \boldsymbol{J}_\lambda \boldsymbol{C}_\lambda \boldsymbol{J}_\lambda^T \tag{42}$$

and λ is

$$\lambda \sim N(\overline{\lambda}, \sigma_\lambda). \tag{43}$$

The derivation of error for $\boldsymbol{t}$ is similar to that for λ. The matched error is also called tracked error because the feature tracking algorithm cannot match all feature points correctly. Hence, this matched error is described by the white noise on the corresponding feature points in the next frame pixel coordinates: x'_p and y'_p. The Jacobian matrix of $\boldsymbol{t}$ with x'_p, y'_p , λ, and λ' is derived as

$$\boldsymbol{J}_t = \left[\frac{\partial(\boldsymbol{P}' - \boldsymbol{RP})}{\partial x'_p} \quad \frac{\partial(\boldsymbol{P}' - \boldsymbol{RP})}{\partial y'_p} \quad \frac{\partial(\boldsymbol{P}' - \boldsymbol{RP})}{\partial\lambda} \times \frac{\partial(\boldsymbol{P}' - \boldsymbol{RP})}{\partial\lambda'}\right]. \tag{44}$$

According to (24) and the camera model, the $\boldsymbol{J}_t$ can be described in detail as

$$\boldsymbol{J}_t = \begin{bmatrix} \frac{\lambda}{f_x} & 0 & \frac{r_{11}(x_p - p_x)}{f_x} + \frac{r_{12}(y_p - p_y)}{f_y} + r_{13} & \frac{x'_p - p_x}{f_x} \\ 0 & \frac{\lambda}{f_y} & \frac{r_{21}(x_p - p_x)}{f_x} + \frac{r_{22}(y_p - p_y)}{f_y} + r_{23} & \frac{y'_p - p_y}{f_y} \\ 0 & 0 & \frac{r_{31}(x_p - p_x)}{f_x} + \frac{r_{32}(y_p - p_y)}{f_y} + r_{33} & 1 \end{bmatrix} \tag{45}$$

$$\boldsymbol{C}_t = \begin{bmatrix} \sigma_{xp'} & 0 & 0 & 0 \\ 0 & \sigma_{yp'} & 0 & 0 \\ 0 & 0 & \sigma_\lambda & 0 \\ 0 & 0 & 0 & \sigma_{\lambda'} \end{bmatrix} \tag{46}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{J}_t \boldsymbol{C}_t \boldsymbol{J}_t^T. \tag{47}$$

For $\boldsymbol{t}$, the error can be described by

$$\boldsymbol{t} \sim N(\overline{\boldsymbol{t}}, \boldsymbol{\Sigma}). \tag{48}$$

Therefore, drift error in the linear approach is introduced by noise on the sonar, pixel coordinate error, and match error. According to (45) and (46), smaller pixel motion indicates smaller covariance in its distribution. In order to reduce these errors, the Gaussian Blur filter has been applied to remove the pixel noise and a high-frame-rate monocular camera is used to reduce the pixel motion between dual frames. Noise on the IMU is not considered here because the attitude of vehicle is corrected by the nonlinear optimal method, which is discussed in the next section.

E. Nonlinear Iteration Methods

Due to its benefits in iteration methods [49], the pose of vehicle $\boldsymbol{T} \in \boldsymbol{SE(3)}$ is described by the associated Lie algebra, $\boldsymbol{\xi} \in \mathfrak{se}(3)$ and $\boldsymbol{\xi} \in \mathbb{R}^6$. The first three elements in $\boldsymbol{\xi}$ represent the rotation and the latter three elements represent the translation. According to Lie algebra, there is an exponential map from $\boldsymbol{\xi}$ to $\boldsymbol{T}$ [49]

$$\boldsymbol{T} = \exp(\hat{\boldsymbol{\xi}}). \tag{49}$$

The $\hat{\boldsymbol{\xi}}$ is a skew symmetric matrix of $\boldsymbol{\xi}$. Using the Direct method, it is described by probability maximum likelihood as

$$p(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots \boldsymbol{P}_n, I', I|\boldsymbol{\xi}). \tag{50}$$

Here, $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots \boldsymbol{P}_n$ are 3-D feature points in previous frame, I' and I are the intensity maps of next frame and previous frame, they are denoted by $\boldsymbol{Z}$

$$p(\boldsymbol{Z}|\boldsymbol{\xi}) \propto \exp\left(-\frac{1}{2}\sum_{n=1}^{N}\{(I(u(\boldsymbol{P}_n)) - I'(u(\exp(\hat{\xi})\boldsymbol{P}_n)))^T \times \boldsymbol{\Sigma}_n^{-1}(I(u(\boldsymbol{P}_n)) - I'(u(\exp(\hat{\xi})\boldsymbol{P}_n)))\}\right). \tag{51}$$

Where u is a map from 3-D feature points to 2-D pixel points at image plane, $\boldsymbol{P} \in \mathbb{R}^3, \boldsymbol{v} \in \mathbb{R}^2, u(\boldsymbol{P}) \to \boldsymbol{v}$. According to the perspective camera model, u is described as

$$(X, Y, Z)^T \to (fX/Z + p_x, fY/Z + p_y)^T. \tag{52}$$

The prior of $\boldsymbol{\xi}$ is predicted from the IMU sensor. The function is

$$p(\boldsymbol{\xi}_{\text{imu}}) \sim N(\overline{\boldsymbol{\xi}}_{\text{imu}}, \boldsymbol{\Sigma}_{\text{imu}}). \tag{53}$$

The posterior of $\boldsymbol{\xi}$ becomes

$$p(\boldsymbol{\xi}|\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{\xi})p(\boldsymbol{\xi}). \tag{54}$$

The optimal pose is obtained by minimizing the negative log-posterior

$$\boldsymbol{\xi}_{\min} = \arg\min_{\boldsymbol{\xi}}\{-\log(p(\boldsymbol{\xi}|\boldsymbol{Z}))\}. \tag{55}$$

It is equivalent to minimizing the cost function

$$f = \left\{\frac{1}{2}\sum_{n=1}^{N}\boldsymbol{e}_{pn}^T\boldsymbol{\Sigma}_n^{-1}\boldsymbol{e}_{pn} + \frac{1}{2}\boldsymbol{e}_{\text{imu}}^T\boldsymbol{\Sigma}_{\text{imu}}^{-1}\boldsymbol{e}_{\text{imu}}\right\} \tag{56}$$

$$\boldsymbol{e}_{pn} = \boldsymbol{P}'_n - \exp(\hat{\xi})\boldsymbol{P}_n \tag{57}$$

$$\boldsymbol{e}_{\text{imu}} = \boldsymbol{\xi} - \boldsymbol{\xi}_{\text{imu}}. \tag{58}$$

f can be written as

$$f = \frac{1}{2}\boldsymbol{E}_f^T\boldsymbol{\Sigma}_f^{-1}\boldsymbol{E}_f \tag{59}$$

$$\boldsymbol{E}_f = \text{diag}(\boldsymbol{e}_{p1}, \boldsymbol{e}_{p2}, \ldots, \boldsymbol{e}_{pn}, \boldsymbol{e}_{\text{imu}}) \tag{60}$$

$$\boldsymbol{\Sigma}_f = \text{diag}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_n, \boldsymbol{\Sigma}_{\text{imu}}). \tag{61}$$

The Jacobian matrix is

$$\boldsymbol{J} = \frac{\partial \boldsymbol{E}_f}{\partial \boldsymbol{\xi}} = \text{diag}\left(\frac{\partial \boldsymbol{e}_{p1}}{\partial \boldsymbol{\xi}}, \ldots, \frac{\partial \boldsymbol{e}_{pn}}{\partial \boldsymbol{\xi}}, \frac{\partial \boldsymbol{e}_{\text{imu}}}{\partial \boldsymbol{\xi}}\right) \tag{62}$$

$$\begin{aligned}\frac{\partial \boldsymbol{e}_{pn}}{\partial \boldsymbol{\xi}} &= \frac{\partial (I(u(\boldsymbol{P}_n)) - I'(u(\exp(\hat{\boldsymbol{\xi}})\boldsymbol{P}_n)))}{\partial \boldsymbol{\xi}} \\ &= \frac{\partial - I'(u(\exp(\hat{\boldsymbol{\xi}})\boldsymbol{P}_n))}{\partial \boldsymbol{\xi}} \\ &= -\frac{\partial I'}{\partial u}\frac{\partial u}{\partial \exp(\hat{\xi})\boldsymbol{P}_n}\frac{\partial \exp(\hat{\xi})\boldsymbol{P}_n}{\partial \boldsymbol{\xi}}\end{aligned} \tag{63}$$

$$\frac{\partial \exp(\hat{\xi})\boldsymbol{P}_n}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{I}_{3\times 3} & -[(\exp(\hat{\boldsymbol{\xi}})\boldsymbol{P}_n)]_\times \\ \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 3} \end{bmatrix}. \tag{64}$$

The $\partial I'/\partial u$ and $(\partial u(\boldsymbol{P})/\partial \boldsymbol{P})$ may be derived from (18) and (52). The derivative of the regularization term ($\boldsymbol{e}_{\text{imu}}$) is

$$\begin{aligned}\frac{\partial \boldsymbol{e}_{\text{imu}}}{\partial \boldsymbol{\xi}} &= \frac{\partial (\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{imu}})}{\partial \boldsymbol{\xi}} \\ &= \boldsymbol{I}_{6\times 6}.\end{aligned} \tag{65}$$

Hence, with the Jacobian matrix, different iterative approaches can be used to obtain the optimal results. For instance, applying the Gauss–Newton iteration algorithm, the update for $\boldsymbol{\xi}$ is

$$\boldsymbol{\Delta}_\xi = -(\boldsymbol{J}^T \boldsymbol{\Sigma}_f \boldsymbol{J})^{-1} \boldsymbol{J}^T \boldsymbol{\Sigma}_f^{-1} \boldsymbol{E}_f \tag{66}$$

$$\boldsymbol{\xi}_{\text{new}} = \boldsymbol{\xi}_{\text{old}} + \boldsymbol{\Delta}_\xi. \tag{67}$$

The Levenberg–Marquardt iteration approach can also be applied to minimize the cost function with respect to $\boldsymbol{\xi}$.

1) Error Analysis: Assuming that $\overline{\boldsymbol{\xi}}$ is obtained from minimization of the cost function, the covariance matrix of $\boldsymbol{\xi}$ is [50]

$$\boldsymbol{\Sigma}_\xi = (\boldsymbol{J}_f^T \boldsymbol{\Sigma}_f^{-1} \boldsymbol{J}_f)^{-1}|_{\boldsymbol{\xi}=\overline{\boldsymbol{\xi}}} \tag{68}$$

where $\boldsymbol{\xi}$ is expected to obey

$$\boldsymbol{\xi} \sim N(\overline{\boldsymbol{\xi}}, \boldsymbol{\Sigma}_\xi). \tag{69}$$

VI. Implementation and Experiments

In the implementation of the algorithm, all codes were written in C++ language. Image undistortion, the FAST corner detector, and Gaussian Blur filter were implemented using OpenCV. In the nonlinear section, the LM approach was applied to solve the optimization problem, the Ceres solver developed by Google [51] was used for the LM. The 3-D trajectory was reconstructed by Pangolin which is a light-weight portable rapid development library for managing OpenGL.

The modified ROV collected the data including depth from the sonar, IMU information, compass information, images from the monocular camera, images from the T265 stereo camera, and position from the T265 chip. This information was stored in the onboard computer temporarily. Once the test finished, the data was transferred to a hard disk via WIFI.

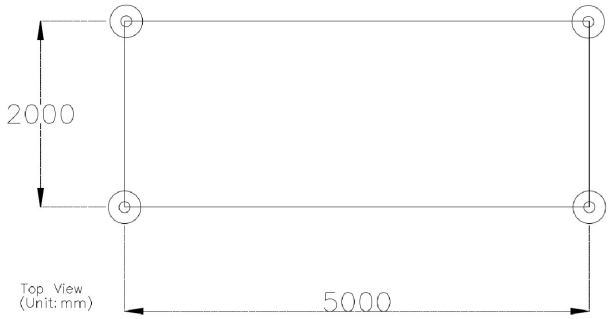


Fig. 11. Square strap scale in towing tank.

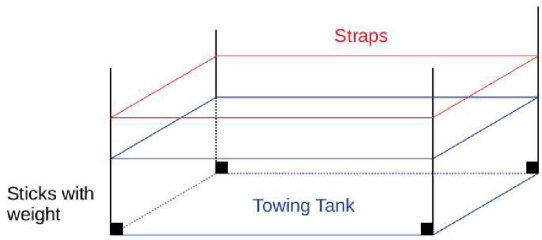


Fig. 12. Square strap 3-D in towing tank.

Fig. 13. Square strap and operator.

Due to the modification, the centers of gravity and buoyancy of the vehicle changed significantly affecting the hydrodynamic characteristics. To resolve this problem, an operator moved the modified ROV along the reference line of the test. Meanwhile, the sensors in the ROV kept recording the data. The operator tried to avoid being in the view of the cameras.

A series of practical tests was conducted in the towing tank in Newcastle University [52]. In order to verify the effectiveness of the method, a square was set out with straps in the towing tank. The scale and 3-D pictures are presented in Figs. 11 and 12. The operator controlled the vehicle manually in the square to ensure that the vehicle travelled along the path marked out with the straps, as shown in Fig. 13. In addition to the comparison

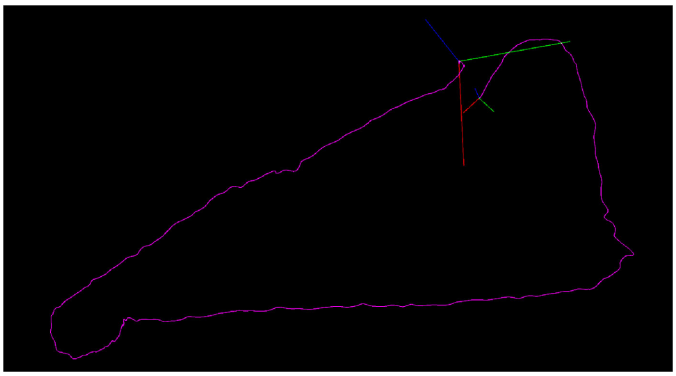


Fig. 14. Estimated trajectory in 3-D.

with T256 camera, the performance of the method is compared with that of other open source visual SLAMs.

The towing tank can be considered as an underwater feature-sparse environment. In realistic subsea environments, objects lying on the seabed can contribute feature points to the method. The uncertain environmental factors, such as illumination, and transparency of sea water, have a negative effect on the performance of the method in terms of preventing the camera catching a number of feature points. Because of that, if the method can work well in underwater feature-sparse environment, it is expected to be able to work adequately at sea.

A. Results of IVO and V-SLAM of T265 Camera

Three tests were carried out with different reference paths. The results presented are from the proposed IVO method and the T265 position data. The vehicle was operated to travel along reference paths in the form of a square, a triangle, and a figure-8 shape. In each case, the vehicle returned to the starting point at the end of the test, hence each estimated trajectory is expected to be closed. However, due to manual and accumulated errors, the final pose of the ROV in the estimated trajectory has an offset relative to the starting point. The error ratio used to analyse the performances is defined as

$$e_r = \frac{\text{offset}}{\text{trajectory length}}. \tag{70}$$

The estimated trajectory was drawn in 3-D, as shown in Fig. 14. In order to evaluate the performance effectively, the 3-D estimated trajectory has been plotted in 2-D as a top view. The 2-D results with reference path are presented in Figs. 15 to 26. In order to prove the repeatability of the method, each shape with reference is tested twice. In the results from T265, the reference is not shown, because the trajectories estimated by T265 have a huge drift error relative to the size of the reference. The distance and error ratio of each test are listed at Table II.

The results without reference path are presented in Figs. 27 to 30. The vehicle tries to follow the triangle and circle shapes and return the start point at the end. The error ratio is shown in the figure caption for these results.

1) Discussion: From the results, it can be seen that the performance of the proposed IVO method is significantly better

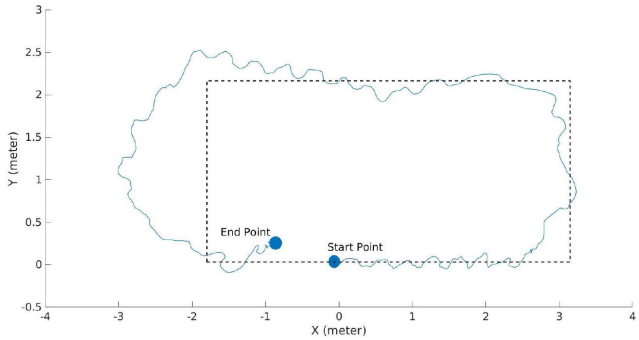


Fig. 15. IVO square shape (1st test) with reference, $e_r = 5.6\%$.

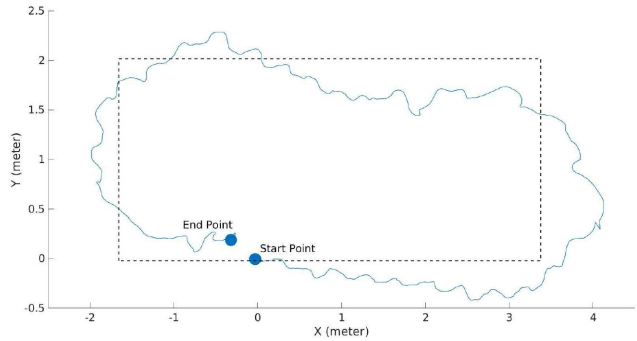


Fig. 16. IVO square shape (2nd test) with reference, $e_r = 2.3\%$.

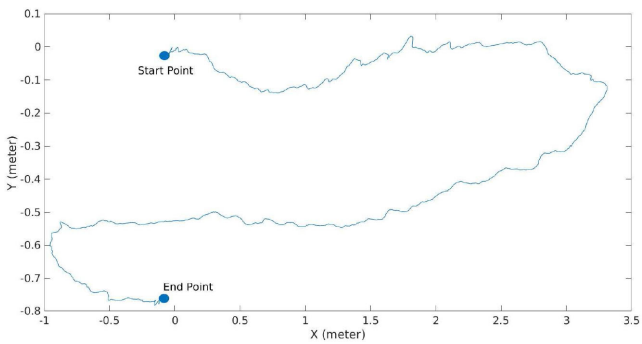


Fig. 17. T265 square shape (1st test).

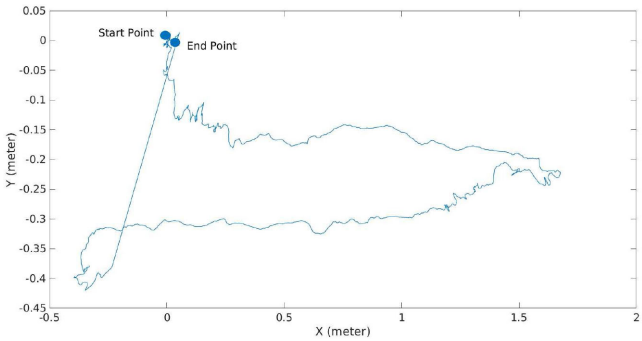


Fig. 18. T265 square shape (2nd test).

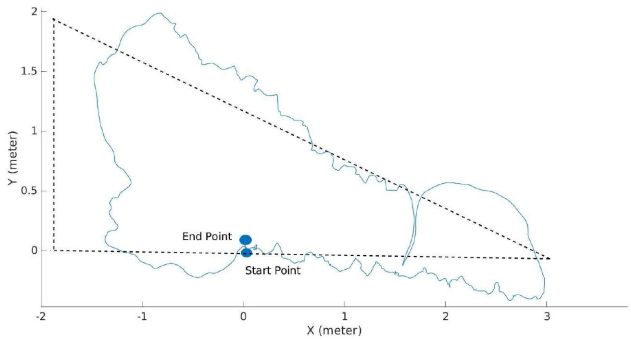


Fig. 19. IVO triangle shape (1st test) with reference, $e_r = 0.51\%$.

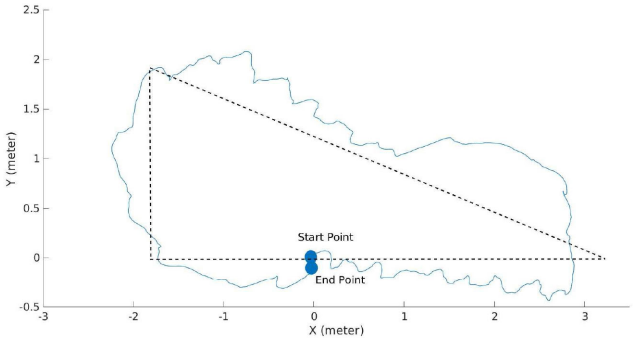


Fig. 20. IVO triangle shape (2nd test) with reference, $e_r = 0.90\%$.

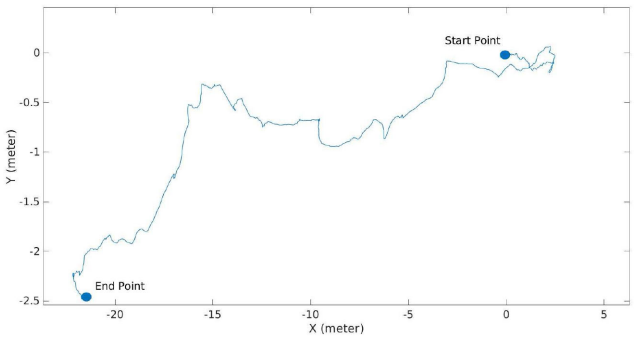


Fig. 21. T265 triangle shape (1st test).

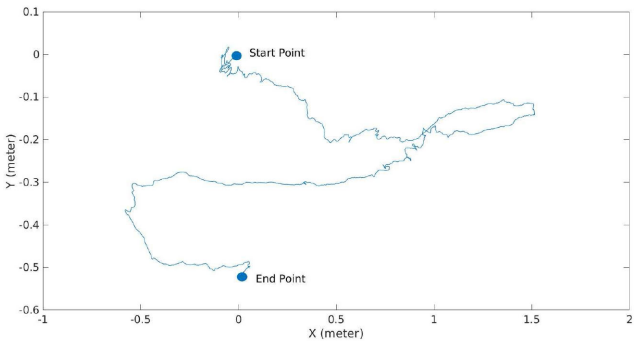


Fig. 22. T265 triangle shape (2nd test).

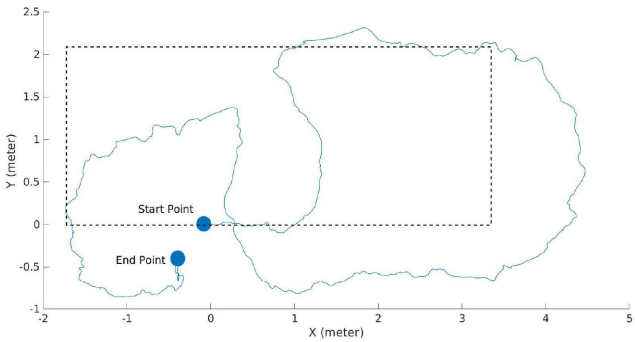


Fig. 23. IVO figure-8 shape (1st test) with reference, $e_r = 3.15\%$.

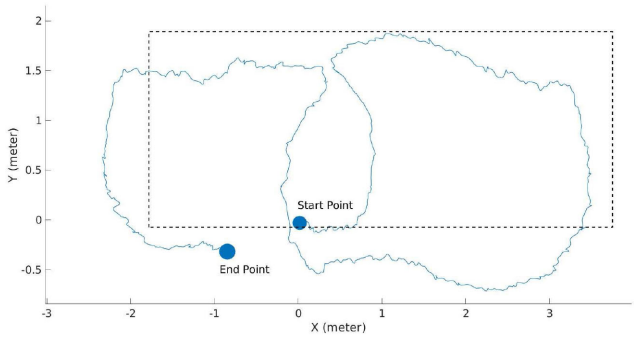


Fig. 24. IVO figure-8 shape (2nd test) with reference, $e_r = 4.35$.

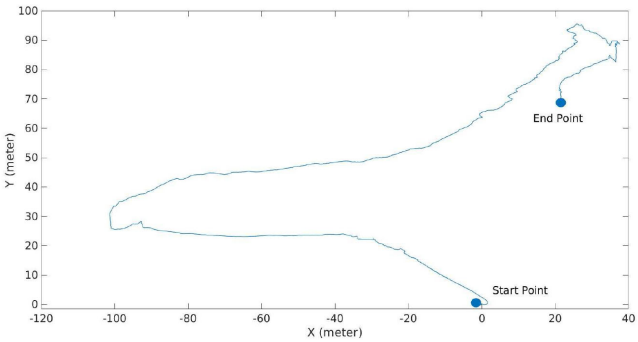


Fig. 25. T265 figure-8 shape (1st test).

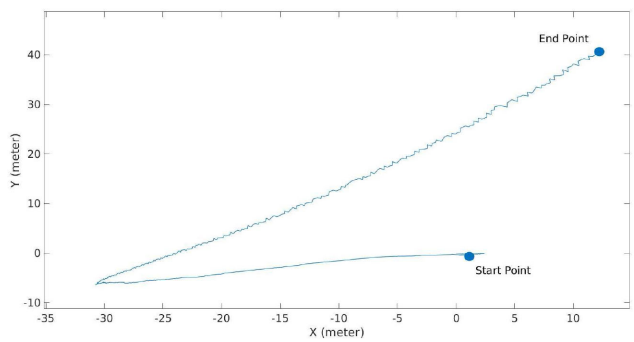


Fig. 26. T265 figure-8 shape (2nd test).

TABLE II
ERROR RATIO AND DISTANCE OF EACH TEST USING IVO-M METHOD

Shape	Distance(metre)	Error Ratio
With Reference Path		
Square 1st	17.5219	0.0560
Square 2nd	16.8559	0.0230
Triangular 1st	15.7125	0.0051
Triangular 2nd	16.1831	0.0090
Figure-8 1st	22.6775	0.0315
Figure-8 2nd	22.7320	0.0435
Without Reference Path		
Triangular	14.9377	0.0270
Circle	17.8133	0.0391

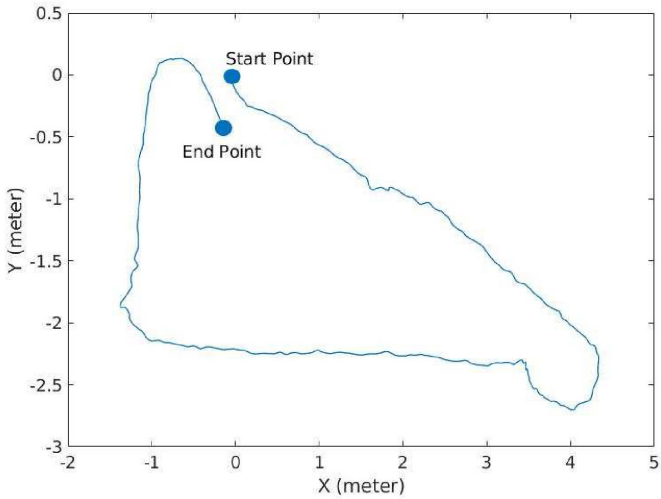


Fig. 27. IVO triangle without reference, $e_r = 2.7\%$.

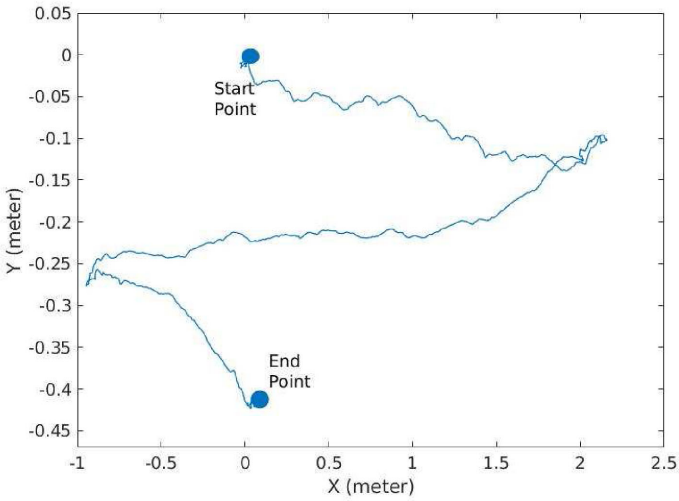


Fig. 28. T265 triangle without reference.

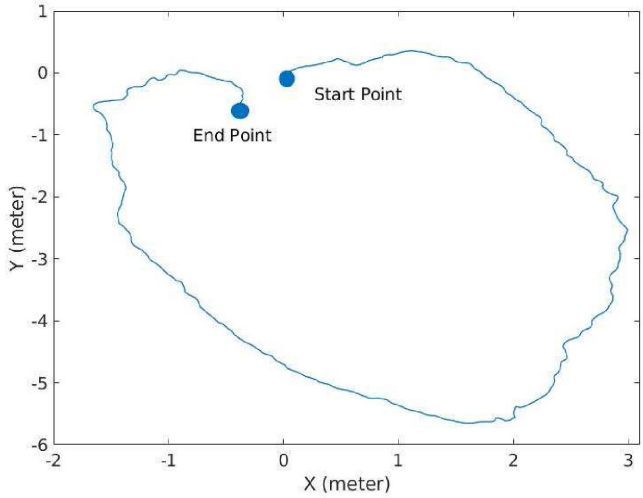


Fig. 29. IVO circle without reference, $e_r = 3.9\%$.

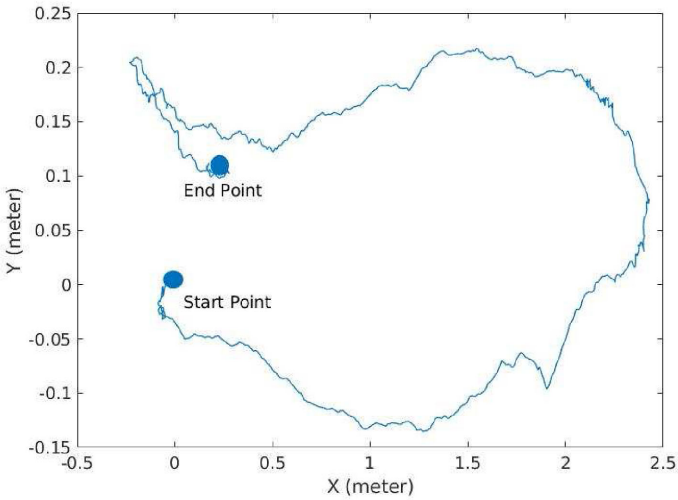


Fig. 30. T265 circle without reference.

than that of the T265. Actually, the T265 failed totally in terms of navigation. The reasons for this are: first, the navigation tests were in a sparse environment and the V-SLAM algorithm used on the T265 requires a dense texture environment, which can provide enough feature points; second, the experiments were conducted in a laboratory with a complex electromagnetic environment, and the inertial sensors in the modified ROV were more robust than the ones in the T265. Hence, in most of the T265 results, the estimated trajectory lost the correct orientation and the linear estimation from the T265 was poor.

Comparing the results with the reference in Figs. 15 and 16, the vehicle was expected to travel along a square reference path, and the estimated trajectories also show a square shape. Because the vehicle was held by an operator, it could not follow the reference path exactly; it was particularly difficult at the corners for the operator to turn the vehicle exactly along the marked path. In Fig. 15, at the start of the test, the estimated trajectory followed the reference line exactly but by the time it had passed

the third corner, the estimated trajectory had a relatively large offset. This indicates that there is an accumulated error resulting in the offset and the corresponding error ratio is 5.6% in the first test and 2.3% in the second test.

In Figs. 19 and 20, the vehicle followed a triangular path. In these tests, the estimated position almost returned to the origin point and the estimated trajectory followed the reference line closely. At each corner, the vehicle had to make the turn in advance, nevertheless the error ratios are small at 0.51% and 0.90%.

In Figs. 23 and 24, the vehicle followed a path in a figure-8 shape within the rectangle set out by the straps. The estimated trajectories show an approximately figure-8 shape within the rectangle, although there is some offset. The error ratio is 3.15% in the first test and 4.15% in the second test. The tests without the reference shape used paths in the form of triangle and circle shapes. In Fig. 27, the estimated trajectory at each corner has a semicircle shape because the operator held the vehicle at a distance to avoid the monocular camera, hence when making a turn, the camera in the vehicle travelled along a semicircle. In Figs. 27 and 29, the estimated trajectories show a triangle and a circle, respectively. The trajectories are almost closed, however, due to the accumulated error, there are acceptable offsets of 2.7% and 3.9%.

Compared with the T265, for which the documentation declares the drift error is less than 1% [36], the novel IVO method has a much better performance in texture-sparse and complex electromagnetic environments. Compared with other inertial sensors, the novel IVO method has low error drift and is low cost. Compared with other VO systems and visual SLAMs, the proposed IVO is designed to work underwater, and is robust in sparse environments. Because the visual methods integrated the information from inertial sensors and sonar, the computational cost has been reduced dramatically.

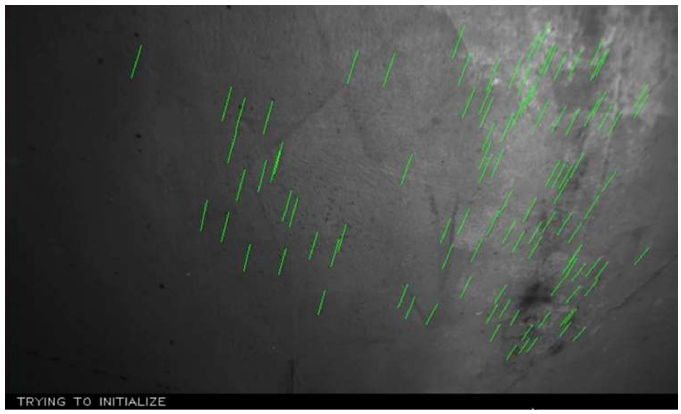


Fig. 31. ORB-SLAM2 trying to initialize in sparse-feature environment.

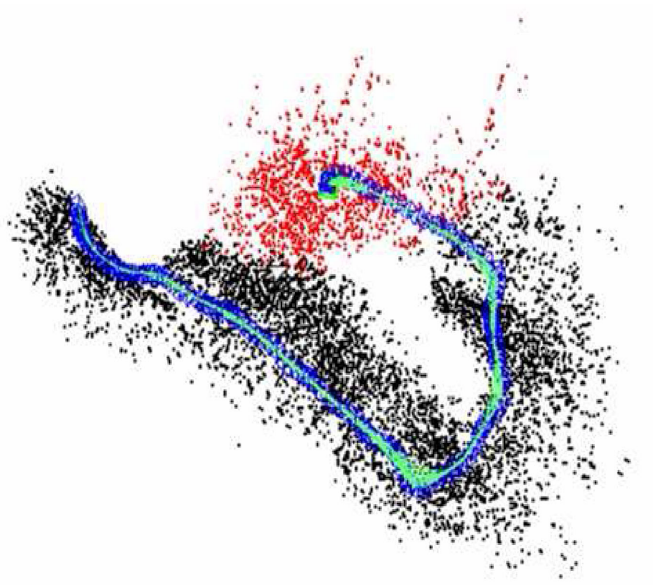


Fig. 32. Trajectory estimated by ORB-SLAM2 (the green line is the estimated trajectory, black dots are measured feature points, red dots are active feature points.).

B. Comparisons Between IVO and Other Visual Navigation Methods

The proposed IVO method was compared with four different Visual SLAM algorithms: ORB-SLAM2, SVO, VINS-Mono, and OKVIS. These algorithms were evaluated on the same dataset (a triangular shape) with the underwater calibrated camera parameters and coordinate transformation matrices. The ORB-SLAM2 method has been implemented for navigating AUVs in [53], and for mapping an underwater cave in [54]. The VINS-Mono and OKVIS algorithms, integrating inertial sensors, have been used to track the pose of a camera in an underwater environment [33], [34]. These methods can work well in feature-dense environments, but they work less successfully in sparse-feature and complex magnetic environments. The SVO is a direct VO method using invariant intensity values in images to recover the camera motion, but it requires at least 100 points to initialize. Therefore, when the SVO was tested using the customized dataset, it became stuck in the initialization stage and failed. The other methods are feature-based methods, so their performances rely on the quality of feature points. In these methods, descriptors are derived to match corresponding points. The sparse-feature environment, with few high-quality feature points, resulted in the failure of these methods. More specifically the ORB-SLAM2 monocular algorithm was hardly able to complete the initialization process, as shown in Fig. 31, and lost feature points easily in the sparse-feature underwater environment. Hence, the trajectory estimated by the ORB-SLAM2 was not complete, as shown in Fig. 32. Meanwhile, the trajectory estimated by the IVO is shown in Fig. 33.

Running the dataset on VINS-Mono and OKVIS algorithms was unsuccessful. The complex magnetic field in the towing tank caused huge errors on the IMU data, because low-cost IMU kits usually use magnetic vectors to correct orientation. The main issue was that the limited feature points could not constrain the drift error from the IMU acceleration data, so that the final drift error was very large. Figs. 34 and 35 illustrate that they could hardly detect any reliable feature points in the sparse-feature

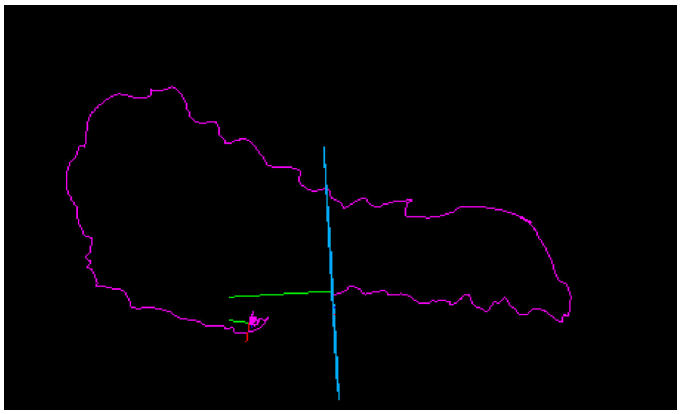


Fig. 33. Trajectory estimated by the IVO (the violet line is the estimated trajectory.).

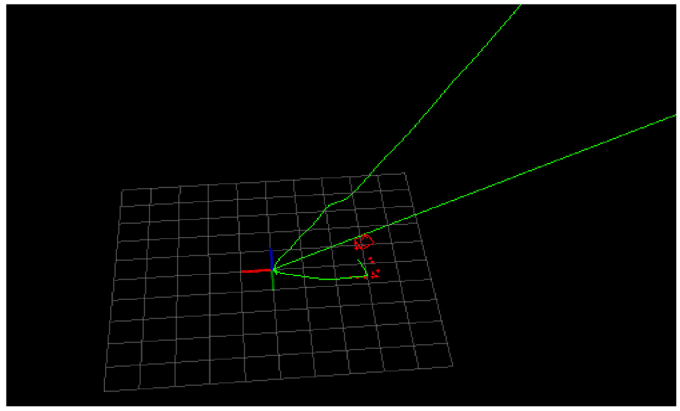


Fig. 36. VINS-mono with huge drift error.

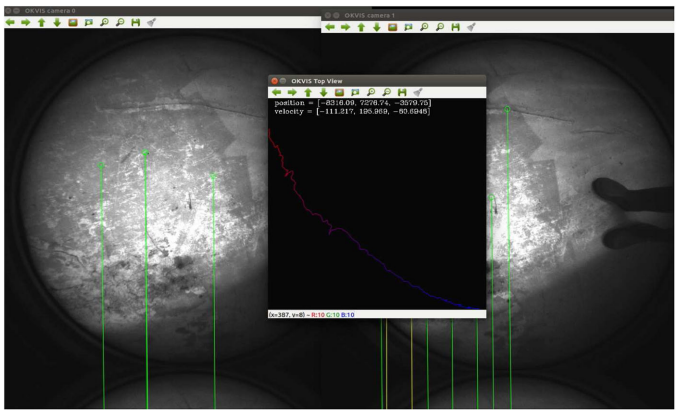


Fig. 34. OKVIS with few matched feature points and huge drift error.

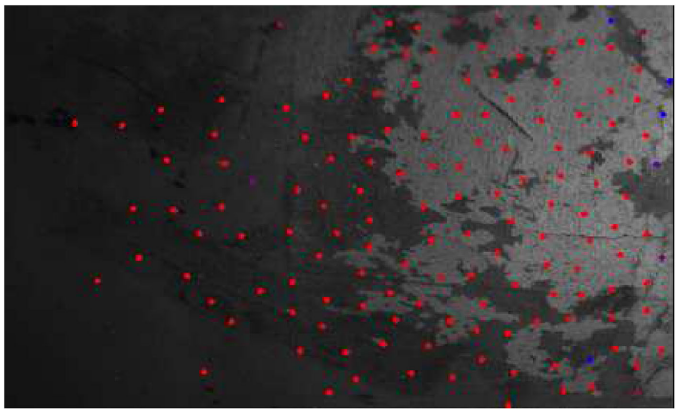


Fig. 35. VINE-mono trying to extract feature points.

underwater environment. Because the T265 V-SLAM is an IMU-Aided VO, it along with the VINS-Mono, OKVIS, and V-SLAM have a common problem: unconstrained drift error, as shown in Figs. 34 and 36.

In contrast to the above methods, the IVO method utilises the FAST algorithm to detect feature points and the OF algorithm to track these identified feature points, rather than using complicated descriptors to match feature points. Furthermore, integrating the sonar and inertial sensor information, the IVO method can skip the initialization stage, which is commonly applied in monocular visual navigation methods. Hence, these characteristics allow the IVO method to perform much more successfully than other techniques in underwater sparse-feature environments, as shown in Fig. 33.

VII. Conclusion

In this article, a novel IVO method has been developed. The navigation approach is designed for low cost underwater applications. It benefits from multiple sensor information and implementations of the FAST algorithm and the OF algorithm. Hence, this method is able to work in texture-sparse environments and provide the estimated trajectory at real scale with only a monocular camera. According to the experimental results, the T265 and other open source Visual SLAMs, which rely on dense feature points, fail to give correct navigation information in this challenging environment, but the IVO approach can provide the estimated trajectory with acceptable offsets. The main limitation of the approach is the assumption that requires the seabed to be locally flat. In future work, it is planned for a stereo camera to replace the monocular camera so that the assumption will not be needed.

Acknowledgment

The authors would like to thank the generous support of John and Vivien Prime in funding aspects of this work.

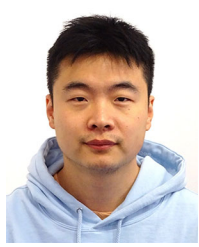
References

[1] G. Antonelli, T. I. Fossen, and D. R. Yoerger, "Underwater robotics," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2008, pp. 987–1008.

[2] D. W. Caress *et al.*, "High-resolution multibeam, sidescan, and subbottom surveys using the MBARI AUV D. Allan B," in *Proc. Mar. Habitat Mapping Technol.*, 2008, pp. 47–69.

[3] R. B. Wynn *et al.*, "Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience," *Mar. Geol.*, vol. 352, pp. 451–468, 2014.

[4] K. Nicholls *et al.*, "Measurements beneath an Antarctic ice shelf using an autonomous underwater vehicle," *Geophys. Res. Lett.*, vol. 33, no. 8, pp. L08612.1–L08612.4, Apr. 2006.
[5] A. Plueddemann, A. Kukulya, R. Stokey, and L. Freitag, "Autonomous underwater vehicle operations beneath coastal sea ice," *IEEE/ASME Trans. Mechatron.*, vol. 17, no. 1, pp. 54–64, Feb. 2012.
[6] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014.
[7] P. Batista, C. Silvestre, and P. Oliveira, "A sensor-based controller for homing of underactuated AUVs," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 701–716, Jun. 2009.
[8] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robot.*, vol. 23, no. 1, pp. 3–20, 2006.
[9] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 2531–2538.
[10] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, 2007.
[11] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 78–90, Jun. 2012.
[12] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.
[13] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *Int. J. Comput. Vis.*, vol. 61, no. 3, pp. 211–231, 2005.
[14] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 1–10.
[15] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2320–2327.
[16] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.
[17] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 15–22.
[18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[19] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
[20] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.
[21] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2018, pp. 3662–3669.
[22] R. M. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE J. Ocean. Eng.*, vol. 33, no. 2, pp. 103–122, Apr. 2008.
[23] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the RMS Titanic: Conservative covariance estimates for slam information filters," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1223–1242, 2006.
[24] A. Kim and R. Eustice, "Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2009, pp. 1559–1565.
[25] M. Kaess, A. Ranganathan, and F. Dellaert, "ISAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.
[26] I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson, "Efficient view-based slam using visual loop closures," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1002–1014, Oct. 2008.
[27] F. S. Hover *et al.*, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, 2012.
[28] A. Kim and R. M. Eustice, "Real-time visual slam for autonomous underwater hull inspection using visual saliency," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 719–733, Jun. 2013.
[29] J. Li, M. Kaess, R. M. Eustice, and M. Johnson–Roberson, "Pose-graph slam using forward-looking sonar," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2330–2337, Jul. 2018.
[30] P. Drap *et al.*, "Underwater photogrammetry and object modeling: A case study of Xlendi Wreck in Malta," *Sensors*, vol. 15, no. 12, pp. 30351–30384, 2015.
[31] F. Bellavia, M. Fanfani, and C. Colombo, "Selective visual odometry for accurate AUV localization," *Auton. Robot.*, vol. 41, no. 1, pp. 133–143, 2017.
[32] M. Ferrera, J. Moras, P. Trouvé–Peloux, and V. Creuze, "Real-time monocular visual odometry for turbid and dynamic underwater environments," *Sensors*, vol. 19, no. 3, 2019, Art. no. 687.
[33] S. Rahman, A. Q. Li, and I. Rekleitis, "Sonar visual inertial slam of underwater structures," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1–7.
[34] S. Rahman, A. Q. Li, and I. Rekleitis, "An underwater slam system using sonar, visual, inertial, and depth sensor," 2018, *arXiv:1810.03200*.
[35] "LPMS na2 introduction," 2018, Accessed: Jan. 11, 2020. [Online]. Available: https://lp-research.com/lpms-nav2/
[36] "Intel realsense t265 tracking camera," 2018, Accessed: Jan. 10, 2020. [Online]. Available: https://www.intelrealsense.com/tracking-camera-t265/
[37] J. Bayer and J. Faigl, "On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense T265," in *Proc. Eur. Conf. Mobile Robot.*, 2019, pp. 1–6.
[38] M. Deegan, A. Dziedzic, C. Jiang, R. Moon, D. Pisarski, and R. Wunderly, "Autonomous quadrotors for the 2019 international aerial robotics competition," in *Proc. Int. Aerial Robot. Competition*, 2019, pp. 1–11.
[39] M. Pelka, K. Majek, and J. Bedkowski, "Testing the affordable system for digitizing USAR scenes," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, 2019, pp. 1–2.
[40] E. Tsykunov, V. Ilin, S. Perminov, A. Fedoseev, and E. Zainulina, "Coupling of localization and depth data for mapping using intel realsense T265 and D435I cameras," 2020, *arXiv:2004.00269*.
[41] A. R. Bekawi *et al.*, "EMU aquabotics: Development of an autonomous underwater vehicle (Caretta2)," in *Proc. RoboSub*, 2019, pp. 1–9.
[42] M. Garratt and J. Chahl, "An optic flow damped hover controller for an autonomous helicopter," in *Proc. 22nd Int. UAV Syst. Conf.*, 2007, pp. 16–18.
[43] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.
[44] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[45] C. G. Harris and J. Pike, "3D positional integration from image sequences," *Image Vis. Comput.*, vol. 6, no. 2, pp. 87–90, 1988.
[46] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[47] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "Computer vision for human–machine interaction," in *Computer Vision for Assistive Healthcare*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 127–145.
[48] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2432–2439.
[49] J.-L. Blanco, "A tutorial on SE(3) transformation parameterizations and on-manifold optimization," Univ. Malaga, Malaga, Spain, Tech. Rep. 012010, vol. 3, 2010.
[50] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York, NY, USA: Cambridge Univ. Press, 2003, pp. 23–150.
[51] S. Agarwal *et al.*, "Ceres solver," 2010, Accessed: Jan. 2, 2019. [Online]. Available: http://ceres-solver.org
[52] "Hydrodynamic lab in newcastle university," 2018, Accessed: Jan. 19, 2020. [Online]. Available: https://www.ncl.ac.uk/engineering/about/facilities/marineoffshoresubsea technology/hydrodynamics/#towingtank
[53] J. Zhang, V. Ila, and L. Kneip, "Robust visual odometry in underwater environment," in *Proc. MTS/IEEE Oceans Conf.*, 2018, pp. 1–9.
[54] N. Weidner, S. Rahman, A. Q. Li, and I. Rekleitis, "Underwater cave mapping using stereo vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5709–5715.



Zhizun Xu received the B.Eng. degree from Shanghai Maritime University in 2014 and the M.Eng. degree from Mokpo National Maritime University, South Korea, in 2016. Since 2017, he has been working toward the Ph.D. degree with Marine Technology Group, Newcastle University, U.K.

His research field is control systems and visual navigation systems for unmanned underwater vehicles.

Maryam Haroutunian received the B.Sc. degree in ship building engineering from the AmirKabir University of Technology, Tehran, Iran, in 2007 and the M.Sc. degree in naval architecture from Newcastle University, U.K., in 2009. She was awarded the Ph.D. degree in 2014 from Newcastle University for her research on bio-inspiration to improve the performance of unmanned underwater vehicles which was sponsored by the EPSRC-UK.

She became a lecturer in Marine Technology at Newcastle University in 2015. She is currently a Naval Architect with specific research interests in marine hydrodynamics including,manoeuvring & seakeeping simulation of marine vehicles as well as novel approaches to optimise the design, control and overall performance of underwater vehicles.

Alan J. Murphy received the B.Eng. degree in naval architecture from Newcastle University, U.K., in 2000 and the Ph.D. degree in experimental hydrodynamics from the University of Southampton, U.K., in 2005.

He was previously an Officer with the U.K. Merchant Navy before taking up academic studies in 1996. He joined Newcastle University in 2007 where he is now a Reader in maritime engineering with the School of Engineering. His research interests include sustainable shipping, including reduction in airborne pollution, marine robotics, and marine data analytics.

Jeff Neasham received the B.Eng. degree in electronic engineering from Newcastle University, U.K., in 1994.

He was at Newcastle University until 2007 as a Research Associate working on the research and commercial product development in underwater acoustic communication, sonar imaging, and wireless sensor networks, before taking up an academic post. He is currently a Senior Lecturer in communications and signal processing with the School of Electrical and Electronic Engineering, Newcastle University. He has published over 100 conference and journal publications and his work on underwater acoustic communication and positioning has been commercialised by three U.K. companies. His research interests are in underwater acoustic signal processing and device design, wireless communication networks, and biomedical instrumentation.

Rose Norman (Senior Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from Leeds University, U.K., in 1989 and the M.Sc. and Ph.D. degrees in electrical engineering from Bradford University, U.K., in 1990 and 1994, respectively.

She was a Principle Engineer at Switched Reluctance Drives Ltd. prior to joining Newcastle University, U.K., in 2004 where she is now a Senior Lecturer with the School of Engineering. Her research interests include underwater vehicles, marine robotics and automation, and marine applications of data analytics and machine learning.