IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 May 2022, accepted 23 June 2022, date of publication 29 June 2022, date of current version 11 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187032

RESEARCH ARTICLE

An Integrated Visual Odometry System With Stereo Camera for Unmanned Underwater Vehicles

ZHIZUN XU[1], MARYAM HAROUTUNIAN[2], ALAN J. MURPHY[2], JEFF NEASHAM[2], AND ROSE NORMAN[2], (Senior Member, IEEE)

[1]Maritime College, Guangdong Ocean University, Zhanjiang 524091, China
[2]School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Corresponding author: Zhizun Xu (zhizun@gdou.edu.cn)

ABSTRACT Navigation is a challenging problem in the area of underwater unmanned vehicles, due to the significant electronmagnetic wave attenuation and the uncertainties in underwater environments. The conventional methods, mainly implemented by acoustic devices, suffer limitations such as high cost, terrain effects and low refresh rate. In this paper, a novel low-cost underwater visual navigation method, named Integrated Visual Odometry with a Stereo Camera (IVO-S), has been investigated. Unlike pure visual odometry, the proposed method fuses the information from inertial sensors and a sonar so that it is able to work in context-sparse environments. In practical experiments, the vehicle was operated to follow specific closed-loop shapes. The Integrated Visual Odoemtry with Monocular Camera (IVO-M) method and other popular open source Visual SLAMs (Simultaneous Localisation and Mappings), such as ORB-SLAM2 and VINS-Mono, have been used to provide comparative results. The cumulative error ratio is used as the quantitative evaluation method to analyse the practical test results. It is shown that the IVO-S method is able to work in underwater sparse-feature environments with high accuracy, whilst also being a low cost solution.

INDEX TERMS Underwater navigation, underwater vehicles, visual-inertial odometry, sensor fusion.

I. INTRODUCTION

Unmanned Underwater vehicles (UUVs), specifically Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs), are widely used in offshore engineering, ocean research, and marine mining. ROVs have also been used since the 1980s for pipeline inspection missions [1], [2]. Modern ROVs are utilised broadly in the development of subsea oil and gas facilities [3]. For example, in the research project *MarMine*, a work class ROV was deployed with a drill mounted for testing in basalt rock at 2700m depth [4]. In [5], a swarm of AUVs was utilised to explore an unknown environment and the Autosub3 AUV has been used for geoscience beneath the Pine Island Glacier Ice Shelf [6]. Meanwhile, in April 2010, the Arctic Explorer completed over 1000 km of under-ice survey, mapping the sea floor during 10 days of continuous underwater operation [7], [8].

However, underwater navigation is still challenging. Because electromagnetic waves are attenuated in water, inertial measurements and acoustic positioning methods are commonly implemented for localising UUVs. Nevertheless, such conventional navigation systems suffer many drawbacks. The Inertial Navigation System (INS) suffers from white noise and bias noise in acceleration measurement, which cause unbounded position error after double integration. The Doppler Velocity Log (DVL) requires vehicles to be close to the seabed, since it needs to receive the reflected sound waves to deduce relative linear velocities. For the same reason, it is not sensitive for estimating rotational motions [9]. In acoustic navigation systems (Ultra Short Baseline (USBL) or Long Baseline (LBL)), much time and money are consumed by

The associate editor coordinating the review of this manuscript and approving it for publication was Ehsan Asadi.

the deployment and recovery of equipment. In addition, environmental factors, such as tide level, water current, and the speed of sound in water, may disturb the accuracy of acoustic transponders [10]. Hence, the integration of INSs, DVLs and acoustic transponders is usually utilised to provide a reliable navigation solution in practical implementations [8]. The arrangement however makes the cost of underwater localisation methods extremely high.

At the same time, Visual Odometry (VO) and Visual Simultaneous Localisation and Mapping (SLAM) Algorithms have been successfully applied in mobile robotics and aerial robotics [11]. One well-known visual odometry application has been on NASA's Mars exploration rovers [12]. A VO system with a stereo camera was used to detect and compensate for any accidental slip of the rovers. In 2005, Nister [13] developed visual odometry techniques with a monocular camera and a stereo camera for navigation of ground vehicles. In the results, error rates were generally less than 2.0%. Scaramuzza [14] developed a special visual odometry with an omnidirectional camera to guide ground cars; its cumulative error after 400m travelling was about 6.5m. Forster [15] reported a fast monocular visual odometry for micro drones and the drift in metres per second was about 0.0051m/s.

In comparison with the wide range of applications of visual odometry in mobile robots and air drones, the development of visual odometry in underwater environments is slow. Apart from the fact that the illumination and visibility are uncertain in the underwater environment, the main reason is that the seabed may not offer as many feature points as the ground environment can offer [10]; and the cost of underwater data collection is much higher relative to ground data collection. Recently, the LiDAR camera is popular in VO applications of mobile cars and drones [16], which can generate the point cloud image directly by using laser scanning. However, in water where the speed of light becomes different and unstable, the LiDAR system needs to be calibrated carefully to retain its accuracy. In addition to that, since the light ray in water is subject to the scattering and the attenuation, the measurement range of LiDAR systems is very limited. These shortcomings have prevented the development of applications of VOs on underwater vehicles.

In this paper, an Integrated Visual Odometry with Stereo Camera (IVO-S) has been investigated to address underwater visual navigation challenges in sparse-feature environments. Unlike general inertial-visual navigation methods which were developed based on tightly coupled algorithms and feature-based methods, the characteristics of the proposed method are as follows:

- After the stereo camera 3D reconstruction, unreliable feature points are filtered out according to depth information from the sonar;
- Prior to processing nonlinear optimisation, a robust linear estimation algorithm and a fusion algorithm are implemented to derive and correct translation vectors.

In the investigated method, the robust linear estimation algorithm plays an important role. More specifically, 3D positions of feature points are reconstructed using data from the stereo camera, and are tracked by the Lucas-Kanade Optical Flow (OF) algorithm. An IMU development kit and a gyroscope measure the acceleration and orientation of the vehicle, and a ping sonar measures the distance between the vehicle and the seabed. Subsequently, the linear estimation and the fusion algorithm are processed to calculate a translation vector. Finally, the Maximum A Posterior (MAP) method is employed to optimise the pose of vehicle by integrating the information from results of the linear estimation and the multiple sensors. The method is verified by practical experiments by means of quantitative evaluation methods, compared with other popular visual SLAMs and odometries. The main novelty of the method is that it is able to navigate underwater vehicles in sparse-feature environments with high accuracy, while other visual navigation methods perform unsuccessfully.

The outline of the paper is: Previous work on VO and underwater navigation systems are introduced first; Secondly, the ROV and related modifications for underwater data collection are presented. Thirdly, the geometry transformation matrices among the coordinates of various sensors are derived; Then, the IVO-S method is derived in detail; Next, the implementation of the proposed method, physical experiments, and discussion about the results are presented. Finally, the conclusions are drawn from the work.

II. RECENT WORKS OF VISUAL ODOMETRY AND UNDERWATER NAVIGATION

In recent years, a number of different visual odometry techniques have been reported. These approaches are grouped into two different methods: one is feature-based methods such as PTAM; the other is direct methods such as LSD-SLAM. The indirect methods need a pre-computation step, where the feature points are extracted and tracked. Conversely, the direct methods operate directly on pixel intensities, which results in subpixel precision at high frame-rates.

The feature-based methods include Parallel Tracking And Mapping (PTAM) [17] and ORB-SLAM algorithms [18]. They first extract the feature points with distinguishing descriptors from the initial image. Subsequently, these feature points are tracked by their own descriptors. Through these corresponding feature points in a series of images, the motion can be estimated by multiple-view geometry technology. The ORB-SLAM algorithm, developed from PTAM, utilised a robust feature detection algorithm named ORB [19] to extract and track these points with ORB descriptors. During the motion estimation process, the ORB-SLAM method implements the Levenberg-Marquardt algorithm to solve for optimal camera pose.

The direct methods include Fast Semi-direct monocular Visual Odometry (SVO), Large-Scale Direct Monocular SLAM (LSD-SLAM), and Direct Sparse Odometry (DSO). The SVO, developed by Forster [15], requires a flat plane parallel to the camera in the bootstrap. Hence, that method is usually applied for air drones. The LSD-SLAM and DSO

were developed by Engel [20], [21]. As mentioned before, they both skip the pre-process step, and operate on the pixels directly. In these two methods, the geometric error function is minimised by the Levenberg-Marquardt method to solve for the optimal camera pose.

While the pure visual SLAMs are accurate, low cost, and have bounded position error, the performances of these SLAM methods are dependent on the density of feature points. Therefore, these methods require the usage of global shuttering cameras with long focal-length and wide angle lenses to grab more reliable feature points. Pure SLAMs usually require at least 50 feature points to recover the cameras' motion. In the SVO method, at least 100 feature points are required in the bootstrap. Because of that, inertial-visual odometries or SLAMs have been developed to improve performances of pure Visual SLAMs. Leutenegger [22] developed an Open Keyframe-based Visual-Inertial SLAM (OKVIS) with a stereo camera and an inertial sensor. A Robust and Versatile Monocular Visual-Inertial State Estimator (VINS-Mono) using a monocular camera and an inertial sensor was developed by Qin [23]. The method is based on tightly coupled algorithms which minimise a summation of reprojection errors and inertial sensor errors by nonlinear regression algorithms, and performed well in dense-sparse environments.

In underwater navigation systems, visual odometry technology is mainly applied to aid other navigation methods. Caccia [24] investigated a horizontal motion estimation based on an optical laser camera. In this method, the depth is measured by a laser, and the optical algorithm only estimates the linear motion. Eustice [25] used a visual odometry method to aid DVL for underwater vehicles. He called it Visually Augumented Navigation (VAN). The DVL and VO are combined by a variant Extended Kalman Filter (EKF), where the attitude is corrected by visual odometry. In this case, visual odometry is applied to bound errors from inertial methods. Eustice [26] then improved the method by replacing the EKF filter with an information filter. The improved approach was applied to underwater exploration surveying of RMS Titanic, and the results showed an improvement in accuracy. Kim [27] applied the VAN method to inspect ship-hulls for the U.S. Navy. The VAN method was extended by Kaess [28] and Mahon [29]. In their work, the optimal trajectory estimation is considered as a smoothing and mapping problem formulation and it is solved by efficient matrix factorisations. The VAN algorithm with image sonar was also utilised to estimate pose and landmarks using pose graph methods in [30]. Kim [31] made an improvement on iSLAM to estimate the trajectory of a vehicle by fusing the information from DVL and a visual odometry system.

Croke [32] made a comparison for UUV navigation systems between the VO and the acoustic beacons methods. In this paper, the visual odometry with stereo camera gives 5% along-track error. Hildebrandt [33] reported an IMU-aided stereo visual odometry method to predict the trajectory of UUVs. However, the algorithm was tested in a synthetic underwater environment rendered by the 3DS Max program. Two open source visual odometries (libviso2 and fovis) were applied for underwater navigation by Wirth [34]. The tank floor was covered by a digital poster, in order to create a rich feature environment. Image matching technology was then used to obtain the ground truth. Based on Wirth's work, Carrasco [35] applied graph-SLAM to improve the visual odometry performance and the mean error was 0.17m after 46m operating in a feature-dense environment. Bellavia [36] developed a stereo visual odometry system using only highly reliable data in the estimation process. The authors applied the KITTI benchmark to evaluate the methods. However, as the KITTI dataset is collected by a ground vehicle, the results of the evaluation were that it was not suitable for underwater navigation.

Recently, Nawa [37] designed a light visual odometry method with a stereo camera adapted to the underwater context and a novel stereo matching approach. Zhang [38] reported a robust visual odometry with a stereo camera; the method is developed based on ORB-SLAM2, and updated by use of the Quad-Screening matching algorithm. A real time Monocular Visual Odometry System for the underwater environment has been developed by Ferrera [39]. In the work, the Optical Flow (OF) algorithm is used to track feature points. The depths of feature points are derived by triangulation calculation. Teixeira [40] investigated implementation of a deep learning algorithm on underwater visual odometry estimation. However, such an algorithm requires a large volume of training data and it is not robust. Dabove [41] reported a monocular visual odometry using low cost sensors. Their method is based on the Kalman filter. Since the aforementioned visual odometries only utilise visual sensors, these methods may not perform well in context-sparse environments.

Creuze [42] developed a monocular odometry aided by inertial and pressure measurements. It can estimate the absolute scale of observed objects. Rahman [43] developed an Inertial-Visual SLAMs with a profiling sonar based on the OKVIS. The information from multiple sensors was fused by a tightly-coupled nonlinear optimisation method. More particularly, camera pose estimations and underwater environment mapping were conducted by minimising the cost function. Later, Rahman [44] updated the method by introducing a loop-closure method, a different initialisation algorithm, and image pre-process technologies. However, quantitative evaluation methods were not applied to verify these methods in underwater environments. Hence, the accuracy of these methods are unknown in underwater navigation applications.

In [45], an Integrated Visual Odometry with a Monocular Camera (IVO-M) method has been investigated, which utilises a sonar, an IMU development kit and a gyroscope. In that method, the 3D feature points are reconstructed by the depth information from a sonar and the assumption of a partially flat seabed. The linear transformation is processed to derive the translation vectors. Although that method can localise the vehicle in underwater sparse-feature

FIGURE 1. VideoRay pro 3 with additional tube.

environments, it is limited by the assumption and suffers from noise on feature point detection and tracking directly. The proposed method applies the depth calculation algorithm of the stereo camera to reconstruct 3D feature points initially. Subsequently, the sonar filter is used to remove unreliable feature points. The trusted points are processed by a robust linear algorithm to derive the translation vectors. Before starting the nonlinear regression, the translation vectors are corrected by the sensor fusion algorithm. Since the depths of feature points are obtained through the stereo camera and the sonar works as a filter, the IVO-S method can have higher precision without requiring any assumptions to be made.

III. HARDWARE CONFIGURATION

The 3D feature points on the seabed are constructed from a pair of images captured by the stereo camera and the depth from the ping sonar. The gyroscope and IMU sensors are utilised to aid in estimating the rotational motions. These sensors were installed on a modified ROV (VideoRay Pro 3). In this section, the ROV and hardware configuration of the novel navigation method are discussed in detail.

Multiple precise sensors are fixed on a tray inside a waterproof tube, which is installed on the bottom of the original vehicle. The main sensors included are: an IMU development kit with signal processing capabilities, a gyroscope, a ping sonar, and an Intel RealSense T265 Tracking camera. The IMU sensor is expected to provide acceleration data, and the gyroscope is used to measure the heading angle precisely. The ping sonar detects the distance between the seabed and the vehicle. The Intel RealSense T265 Tracking Camera integrates a stereo camera, an internal IMU sensor and an Intel Visual Processing Unit (VPU). Therefore, the T265 tracking camera can supply stereo images for the IVO-S.

The underwater vehicle, with the additional tube installed, is shown in Fig. 1. The layout of sensors on the tray is shown in Fig. 2 and Fig. 3. The costs of the various sensors are listed in Table 1, illustrating that the novel IVO-S navigation algorithm can be implemented on low-cost hardware.

IV. GEOMETRY TRANSFORMATION

As mentioned previously, the sensors located on the tray include: an IMU, a gyroscope, a ping sonar and T265 tracking

FIGURE 2. Top view: sensors on electronics tray.

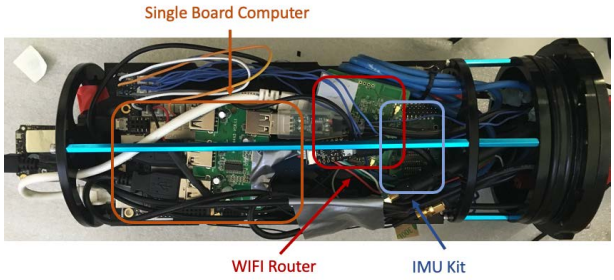


FIGURE 3. Back view: sensors on electronics tray.

TABLE 1. Sensors list.

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

camera. Each of the sensors has their own coordinates. In order to unify data from various sensors on the same coordinate reference frame, transformation matrices ($\boldsymbol{T}$) are derived in this section. The value measured in the sensors' coordinates can be transferred to the vehicle body's coordinate system. In linear algebra theory, the transformation matrix is in the Special Euclidean Group, and has three dimensions. Hence, $\boldsymbol{T} \in \boldsymbol{SE(3)}$, and $\mathbb{R}^3 \times \boldsymbol{SO(3)} \rightarrow \boldsymbol{T}$. The coordinates of the electronics tray are called the body reference. The relationship between body reference and global reference is presented in Fig.4.

The locations and orientations of the sensors relative to each other is arranged beforehand. The sensors are mounted following the blueprint. After that, the locations and orientations are measured manually again.

A. IMU DEVELOPMENT KIT

The IMU sensor can measure acceleration vectors and angular velocities of the device. In this work, the LPMS-ME1, a low cost IMU development kit with integrating signal processing, can output the attitude information based on magnetic field, and acceleration vectors on xyz axes. It uses the quaternions to describe the attitude of the IMU relative to the global coordinates. In computation of the coordinate transformation, the quaternion is converted to the rotation matrix $\boldsymbol{R}$. Unlike $\boldsymbol{T}$, the rotation matrix $\boldsymbol{R}$ is in the Special Orthogonal Group with three dimensions. Hence, it can be written as $\boldsymbol{R} \in \boldsymbol{SO(3)}$. Fig.5 shows the position of the IMU.

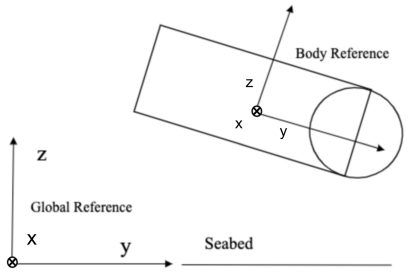


FIGURE 4. Body reference and camera reference.

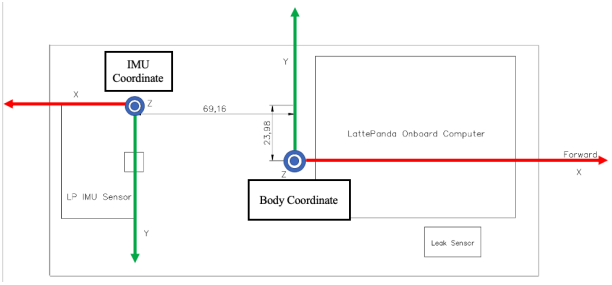


FIGURE 5. Top tray (Unit:mm).

The transformation matrix from IMU reference to body reference is denoted as ${}^{body}\boldsymbol{T}_{imu}$, and it is derived as,

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

$$ {}^{body}\boldsymbol{R}_{imu} = \boldsymbol{R}_z(\pi). \tag{2} $$

$$ {}^{body}\boldsymbol{t}_{imu} = [-0.069\ 0.024\ 0.003] \tag{3} $$

In (3), the units are meters; $\boldsymbol{R}_z(\pi)$ is a basic rotation matrix, which rotates vectors by an angle π about the z-axis in the anticlockwise direction. Now assuming $\boldsymbol{R}_{measure}$ was the measured rotation matrix from the IMU sensor, then ${}^{global}\boldsymbol{R}_{body}$ transferring the body reference relative to the global reference is,

$$ {}^{global}\boldsymbol{R}_{body} = \boldsymbol{R}_{measure}\ {}^{body}\boldsymbol{R}_{imu}^{-1}. \tag{4} $$

B. GYROSCOPE

The LPMS-NAV2 is an accurate one-axis gyroscope and integrates a 3-axis accelerometer [46]. It can measure the heading angle (yaw) with ultra-low drift error. The position of the LPMS-NAV2 is illustrated in Fig.6.

The ${}^{body}\boldsymbol{R}_{nav}$ denotes the transformation matrix from the NAV2 reference to the body reference, and it is derived as,

$$ {}^{body}\boldsymbol{T}_{nav} = \begin{bmatrix} {}^{body}\boldsymbol{R}_{nav} & {}^{body}\boldsymbol{t}_{nav} \\ 0 & 1 \end{bmatrix}. \tag{5} $$

$$ {}^{body}\boldsymbol{R}_{nav} = \boldsymbol{R}_z(-\frac{\pi}{2})\boldsymbol{R}_x(\pi). \tag{6} $$

$$ {}^{body}\boldsymbol{t}_{nav} = [-0.05733\ -0.00309\ -0.015] \tag{7} $$

However, the NAV2 only provides the heading angle (yaw). The yaw from the NAV2 will be mapped into the body coordinates, and used to correct the yaw from the IMU. In this way, the navigation information from the IMU and gyroscope is fused. Considering that $\pm\frac{\pi}{2}$ is the pitch range for the ROV, so the Gimbal lock problem is ignored.

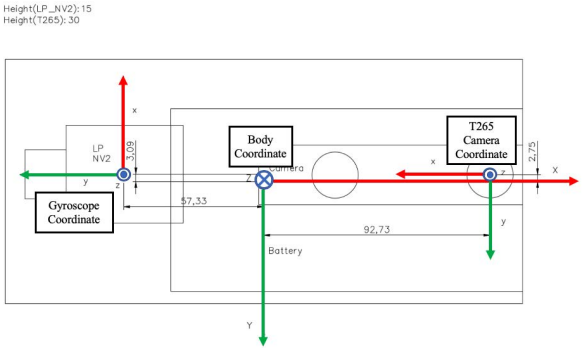


FIGURE 6. Bottom tray (Unit:mm).

C. T265 CAMERA

The Intel RealSense T265 Tracking Camera consists of a fisheye lens stereo camera, a low-cost IMU and VPU. It can output the camera pose based on the V-SLAM algorithm running on the VPU. In the official website, the T265 is stated as having less than 1% closed loop drift error under intended conditions [47]. Furthermore, there are practical projects and drone competitions applying the T265 tracking camera [48], [49] [50]. Fig.6 presents the location of T265 on the tray.

The ${}^{body}\boldsymbol{T}_{t265}$ denotes the transformation matrix from the T265 reference to the body reference, which is derived as,

$$ {}^{body}\boldsymbol{T}_{t265} = \begin{bmatrix} {}^{body}\boldsymbol{R}_{t265} & {}^{body}\boldsymbol{t}_{t265} \\ 0 & 1 \end{bmatrix}; \tag{8} $$

$$ {}^{body}\boldsymbol{R}_{t265} = \boldsymbol{R}_z(\pi)\boldsymbol{R}_x(\pi); \tag{9} $$

$$ {}^{body}\boldsymbol{t}_{t265} = [0.09273\ -0.00275\ -0.03]. \tag{10} $$

With ${}^{body}\boldsymbol{R}_{t265}$, the rotation of the camera can be obtained from the inertial sensors. It is also used to transfer depth from the sonar reference to camera reference.

D. SONAR

The sonar is a single-beam echosounder. It has multiple purposes: working as an altimeter for ROVs and AUVs, bathymetry work aboard a USV (Unmanned Surface Vehicle), and obstacle detection. The sonar can measure the depth from 0.5m to 30m, and the angle of beamwidth is about 30°.

The sonar is mounted on the hull of the ROV, and the position is illustrated in Fig. 7.

The ${}^{body}\boldsymbol{T}_{sonar}$ denotes the transformation matrix from the sonar reference to body reference. The rotation matrix is an identity matrix because the two references are parallel. Hence, only translation bias is considered.

$$ {}^{body}\boldsymbol{T}_{sonar} = \begin{bmatrix} {}^{body}\boldsymbol{R}_{sonar} & {}^{body}\boldsymbol{t}_{sonar} \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{11} $$

$$ {}^{body}\boldsymbol{R}_{sonar} = \boldsymbol{I}. \tag{12} $$

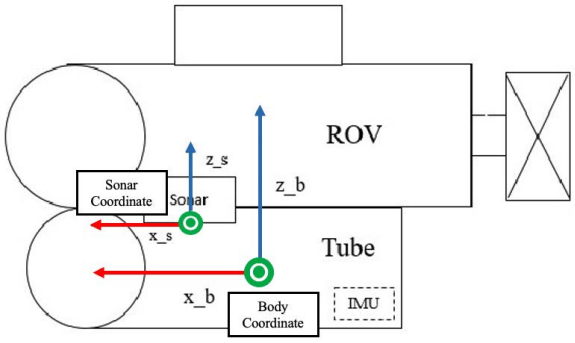


FIGURE 7. Sonar location.

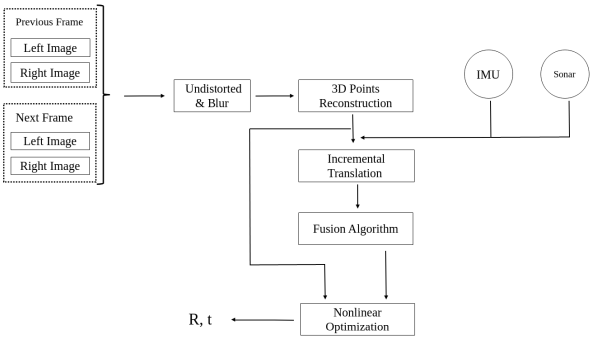


FIGURE 8. Flow diagram of IVO-S algorithm.

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

$$^{body}\boldsymbol{t}_{sonar} = [0.060,\ 0.070,\ 0.065]^T. \tag{13}$$

With the $^{body}\boldsymbol{T}_{sonar}$ transformation, the distance from the body to the ground can be obtained.

V. METHODOLOGY

In this section, the mathematical derivations are introduced. Two pairs of stereo images from previous and next frames are undistorted and blurred firstly. The blur process is utilised to reduce noise by smoothing. In this way, the reliable 3-space feature points can be detected and tracked. Then, the incremental translation between two frames is derived using a linear method. After the fusion algorithm, a nonlinear optimisation method is used to solve for the pose of the camera. The flow diagram of the proposed method is shown in Fig.8.

A. UNDERWATER CALIBRATION AND BLUR

The stereo image sequences are collected by the Intel RealSense Tracking camera T265. The camera installed on the bottom of the vehicle captures the calibration patterns in the underwater environment. While the T262 camera is provided with precise calibration information when it left the production line, it is calibrated in air. However, the refraction can not be ignored in water. It occurs when the light ray from the object in the water propagates through the water-tight container of the camera, causing nonlinear distortion on the imaging plane. The underwater calibration processed by [51] is expected to reduce the effects of the refraction by using the distortion polynomial and increasing the focal length [52], [53]. More specially, the parameters of the stereo camera are computed by OpenCV tools. The Kanalla-Brandt distortion model is applied to undistort these stereo images [54]. The collection of samples for underwater calibration is shown in Fig.9.

After the calibration, the intrinsic matrix $\boldsymbol{K}$, extrinsic parameters, and undistortion polynomial can be obtained. The extrinsic parameters, forming the homogeneous transformation between the left and right camera, are used to rectify the stereo camera. The calibration results are presented in the Appendix section.

FIGURE 9. Underwater pattern.

When the calibration and rectification are completed, the smoothing or blurring of images is processed. It is intended to remove unreliable and easily mismatched features which contribute to the drift error significantly. The smoothing kernel is chosen carefully to reduce the noise, but retain the robust features. In the proposed method, a Gaussian kernel is used to smooth the image. The size of the kernel was set as (3,5), and the covariance is calculated by the *getGaussianKernel()* in the OpenCV library.

B. 3D POINTS RECONSTRUCTION USING SONAR

There are many 3D reconstruction methods that use monocular video sequences, or a combination of monocular cameras with either a laser range scanner or a structured light [55]. However, in the IVO-S, stereo reconstruction is applied to recover the depth of each feature point.

1) FAST ALGORITHM AND OPTICAL FLOW TRACKING

The FAST algorithm (Features Accelerated Segment Test) developed by Rosten [56] is utilised to extract the feature points from the previous left-side image. Compared with other feature point detectors, such as SIFT (The scale-invariant feature transform) [57], the FAST algorithm can detect feature points on the image efficiently. In order to track the feature points in the right-side image and the next pair of images, the Optical Flow (OF) algorithm is implemented. It assumes brightness constancy, so that the intensity of the pixel remains the same despite small changes of position and time period [58].

In the IVO-S, OF is used to track the feature points between dual stereo frames. There are four associated feature points generated per set: F_l indicates 2-space feature points in the previous left-side frame, F_r indicates 2-space feature points in the previous right-side frame. Similarly, F_l' is a set of feature points in the next left-side frame, and F_r' is a set of feature points in the next right-side frame. Using the calibration parameters, the stereo rectification is processed to make sure that their epipolar lines coincide and become parallel to the x-axis of the image.

Given $\boldsymbol{p}_{nl} \in F_l$, and its associated feature point in right-side frame $\boldsymbol{p}_{nr} \in F_r$ matched along the epipolar line, the triangulation method is adopted to obtain the depth of the feature. With the perspective camera model, the 3D position of the feature $\boldsymbol{p}_n$, can be calculated. The the process of the 3D reconstruction of the stereo camera can be regarded as a map $H : \boldsymbol{p}_{nl} \times \boldsymbol{p}_{np} \rightarrow \boldsymbol{p}_n$, where $\boldsymbol{p}_{nl}, \boldsymbol{p}_{np} \in \boldsymbol{R}^2$, and $\boldsymbol{p}_n \in \boldsymbol{R}^3$.

2) GAUSSIAN FILTER WITH SONAR INFORMATION

Both the FAST feature points detection algorithm and the optical flow matching method suffer from noise, which indicates that the errors on the 2-space feature points are introduced into 3-space feature points via a 3D reconstruction algorithm. In the method, the unreliable feature points are removed by a Gaussian filter. The sonar provides the depth (d_{sonar}) from seafloor to vehicle and the depths of most 3-space feature points are expected to distribute near d_{sonar}. After translating d_{sonar} into camera coordinates, given an arbitrary feature point $\boldsymbol{p}_n$, the filter G_s is presented as,

$$G_s(\boldsymbol{p}_n^{[3]}) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(\boldsymbol{p}_n^{[3]} - d_{cam})^2}{2\sigma^2}). \tag{14}$$

$$\boldsymbol{d}_{cam} = {}^{body}\boldsymbol{T}_{cam}^{-1}\, {}^{body}\boldsymbol{T}_{sonar}\boldsymbol{d}_{sonar} \tag{15}$$

where, the $\boldsymbol{d}_{sonar}$ is the vector measured by the sonar. The d_{cam} is the third element of the $\boldsymbol{d}_{cam}$ vector. The $\boldsymbol{p}_n^{[3]}$ is the third element in the 3-space feature point.

If the output of G_s is over a specific threshold value, this 3D feature point is considered as being reliable. The selection of the threshold value is a trade-off issue: The small threshold can allow a large volume of data contributing to the localisation method, but meanwhile too much uncorrected data may increase the localisation error; The large threshold only lets a small partition of data be utilised in the computation, but most unreliable points may be removed, which can finally improve the positioning accuracy. In the method, the threshold value is set to 0.52.

C. INCREMENTAL TRANSLATION RECOVER

Before calculation of the incremental translation vector, the derivation of rotation matrix ${}^2\boldsymbol{R}_1$ from the previous frame to the next frame is required. In this case, the $\boldsymbol{R}_1$ and the $\boldsymbol{R}_2$ should be identified firstly. The $\boldsymbol{R}_1$ can be calculated as,

$$\boldsymbol{R}_1 = {}^{global}\boldsymbol{R}_{imu}\, {}^{imu}\boldsymbol{R}_{body}\, {}^{body}\boldsymbol{R}_{t265} \tag{16}$$

where the ${}^{global}\boldsymbol{R}_{imu}$ is the rotation matrix obtained from the IMU kit and gyroscope sensors for the previous frame time point. The ${}^{imu}\boldsymbol{R}_{body}$ denotes the rotation matrix from the body reference to the IMU coordinate, and the similar meaning for ${}^{body}\boldsymbol{R}_{t265}$.

The $\boldsymbol{R}_2$ matrix is obtained in similar way. Based on two rotation matrices $\boldsymbol{R}_1$ of previous frame and $\boldsymbol{R}_2$ of the next frame, the rotation matrix ${}^2\boldsymbol{R}_1$ can be obtained by,

$${}^2\boldsymbol{R}_1 = \boldsymbol{R}_2^{-1}\boldsymbol{R}_1. \tag{17}$$

Subsequently, the incremental translation ${}^2\boldsymbol{t}_1$ between two frames can be computed with the rotation matrix ${}^2\boldsymbol{R}_1$. Given a reliable feature point $\boldsymbol{p}_n$ in the previous frame and a corresponding point $\boldsymbol{p}_n'$ in the next frame, the translation vector ${}^2\boldsymbol{t}_1$ is derived as,

$${}^2\boldsymbol{t}_1 = \boldsymbol{p}_n' - {}^2\boldsymbol{R}_1\boldsymbol{p}_n. \tag{18}$$

Then, the ${}^2\boldsymbol{t}_1$ will be added into the Γ set. Assuming there are N pairs of feature points, the size of the Γ is N. That means there are N number of ${}^2\boldsymbol{t}_1$ according to different pairs of feature points. Since the feature points are corrupted by the imaging noise and reconstruction error, the N number of ${}^2\boldsymbol{t}_1$ is not identical. Hence, in order to select the reliable translation vector ${}^2\bar{\boldsymbol{t}}_1$ in the set Γ, a robust selection is conducted by calculating distance d_p as,

$$d_p(\boldsymbol{p}_n,\ \boldsymbol{p}_n',\ {}^2\boldsymbol{R}_1,\ {}^2\boldsymbol{t}_1) = \sum_{n=1}^{N} |\boldsymbol{p}_n' - {}^2\boldsymbol{T}_1\boldsymbol{p}_n|, \tag{19}$$

where the transformation matrix ${}^2\boldsymbol{T}_1$ is,

$${}^2\boldsymbol{T}_1 = \begin{bmatrix} {}^2\boldsymbol{R}_1 & {}^2\boldsymbol{t}_1 \\ 0 & 1 \end{bmatrix}. \tag{20}$$

As presented by the equation (19), every ${}^2\boldsymbol{t}_1$ has its own distance d_p, which sums the transformation deviation of each pair of feature points. Eventually, the translation vector ${}^2\boldsymbol{t}_1$ with the smallest distance is selected as ${}^2\bar{\boldsymbol{t}}_1$ from the Γ set. The algorithm of 3D reconstruction and translation recovery is presented as Algorithm 1. Since there is a small number of trusted feature points, such robust estimation does not increase the computational time dramatically.

D. FUSION ALGORITHM

Since the ${}^2\bar{\boldsymbol{t}}_1$ calculated in previous section is corrupted by imaging noise, it is to be updated with the measurement of the sonar by the Maximum A Posterior(MAP) algorithm. In order to accomplish the goal, the conditional posterior probability $p({}^2\boldsymbol{t}_1|\boldsymbol{P}, \boldsymbol{P}', \boldsymbol{R}_1, \boldsymbol{R}_2, d_2, \boldsymbol{t}_1)$ should be maximised with respect to the ${}^2\boldsymbol{t}_1$. The $\boldsymbol{P} = \{\boldsymbol{p}_1\ \boldsymbol{p}_2\ \ldots\ \boldsymbol{p}_n\}$ and $\boldsymbol{P}' = \{\boldsymbol{p}_1'\ \boldsymbol{p}_2'\ \ldots\ \boldsymbol{p}_n'\}$ are two observed feature point sets in the previous frame and next frame respectively. The $\boldsymbol{t}_1$ is the translation vector of the previous frame relative to the original reference and d_2 is the depth from the vehicle to seabed measured by the sonar, and already transferred to the camera reference.

Algorithm 1: 3D Reconstruction and Incremental Translation Recover

Result: ${}^2\bar{t}_1$
2D Feature Point Set on Left Image of previous frame $F_l \leftarrow$ Fast-Detection Algorithm;
Taking the corresponding points on right image of previous frame, left image on next frame, right image on next frame, generating the points sets F_r, F'_l, and F'_r respectively;
$d_0 \leftarrow \infty$;
$n \leftarrow 0$;
Set $A \leftarrow$ Null;
Set Γ (The set of translation vectors) $\leftarrow$ Null;
while $n < Sizeof(F_l)$ **do**
  $p_{nl} \in F_l, p_{nr} \in F_r\ p'_{nl} \in F'_l, p'_{nr} \in F'_r$;
  $p_n \leftarrow H(p_{nl}, p_{nr}), p'_n \leftarrow H(p'_{nr}, p'_{nl})$;
  if $Gs(p_n) > Threshold$ **then**
    ${}^2t_1 \leftarrow p'_n - {}^2R_1 p_n$;
    Add p_n and p'_n into A;
    Add ${}^2t_1$ into Γ;
  else
    Continue;
  end
  n++;
end
$n \leftarrow 0$;
while $n < Sizeof(\Gamma)$ **do**
  $p_n \in A, p'_n \in A$;
  $t_n \in \Gamma$;
  $d_n \leftarrow d_p(t_n, {}^2R_1, p_n, p'_n)$;
  if $d_n < d_0$ **then**
    ${}^2\bar{t}_1 \leftarrow t_n$;
    $d_0 \leftarrow d_n$;
  else
    Continue;
  end
  n++;
end

The posterior probability can be decomposed into the likelihood and the prior terms by the Bayesian formula, which is presented as,

$$p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_1,\boldsymbol{R}_2,d_2,\boldsymbol{t}_1) = \eta\, p(d_2|{}^2\boldsymbol{t}_1,\boldsymbol{R}_1,\boldsymbol{R}_2,\boldsymbol{t}_1)p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_2,\boldsymbol{R}_1). \quad (21)$$

where the $p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_2,\boldsymbol{R}_1)$ is the conditional probability of translation vector ${}^2\boldsymbol{t}_1$ corrupted by imaging noise. The $p(d_2|{}^2\boldsymbol{t}_1,\boldsymbol{R}_1,\boldsymbol{R}_2,\boldsymbol{t}_1)$ is the conditional probability of the measurement of the sonar d_2.

In order to maximise the posterior probability with respect to ${}^2\boldsymbol{t}_1$, the exact expressions of the two conditional probabilities are derived as follows.

As the noise models of the sonar and camera can fit the Gaussian distribution well [59]–[61], the conditional probability of ${}^2\boldsymbol{t}_1$ can be written as,

$$p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_1) \sim N({}^2\boldsymbol{t}_1|\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t). \quad (22)$$

where the term $\boldsymbol{\mu}_t$ is the result of Algorithm 1, $\boldsymbol{\mu}_t = {}^2\bar{\boldsymbol{t}}_1$.

The conditional probability of d_2 can be written as,

$$p(d_2|{}^2\boldsymbol{t}_1,\boldsymbol{R_2},\boldsymbol{R_1},\boldsymbol{t}_1) \sim N(d_2|\mu_d,\sigma_d^2). \quad (23)$$

The μ_d can be derived by,

$$\mu_d = (\boldsymbol{T}_1\ {}^2\boldsymbol{T}_1^{-1})_{[3,4]}. \quad (24)$$

where the $\boldsymbol{T}_1\ {}^2\boldsymbol{T}_1^{-1}$ is calculated as,

$$\boldsymbol{T}_1\ {}^2\boldsymbol{T}_1^{-1} = \begin{bmatrix} {}^2\boldsymbol{R}_1\boldsymbol{R}_1^T & \boldsymbol{t}_1 - {}^2\boldsymbol{R}_1\boldsymbol{R}_1^T\ {}^2\boldsymbol{t}_1 \\ 0 & 1 \end{bmatrix}. \quad (25)$$

With these information, the exact expressions of the likelihood and the prior can be obtained as,

$$p(d_2|{}^2\boldsymbol{t}_1,\boldsymbol{R_2},\boldsymbol{R_1},\boldsymbol{t}_1) = \frac{1}{2\pi\sigma} exp(-\frac{\hat{d}_2^2}{2\sigma^2}) \quad (26)$$

$$p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_1) = \frac{1}{|2\pi\boldsymbol{\Sigma}_t|^{1/2}} exp(-\frac{1}{2}\ {}^2\hat{\boldsymbol{t}}_1^T\ \boldsymbol{\Sigma}_t^{-1}\ {}^2\hat{\boldsymbol{t}}_1) \quad (27)$$

where the ${}^2\hat{\boldsymbol{t}}_1 = {}^2\boldsymbol{t}_1 - \boldsymbol{\mu}_t$ and $\hat{d}_2 = T(d_2) - \mu_d$. The $T(d_2)$ is a transform function which is used to transfer the d_2 from sonar coordinates to the original reference. In the IVO method, $T(d_2) = d_2 - d_{t0}$, d_{t0} is the depth information from the first frame.

Until now, the exact expression of the posterior probability $p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_1,\boldsymbol{R}_2,d_2,\boldsymbol{t}_1)$ has been known. Subsequently, the posterior will be maximised with respect to the ${}^2\boldsymbol{t}_1$. It is equivalent to minimise the negative log of posterior,

$${}^2\boldsymbol{t}_1 = \arg\min_{{}^2\boldsymbol{t}_1}\{-log(p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_1,\boldsymbol{R}_2,d_2,\boldsymbol{t}_1))\} \quad (28)$$

Since the equation (28) is convex, the ${}^2\boldsymbol{t}_1$ can be solved by making its differential equal to zero. Hence, there is,

$$\begin{aligned} &\frac{\partial\ -log(p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_1,\boldsymbol{R}_2,d_2,\boldsymbol{t}_1))}{\partial\,{}^2\boldsymbol{t}_t} \\ &\propto \frac{\partial\ -log(p(d_2|{}^2\boldsymbol{t}_1,\boldsymbol{R}_2,\boldsymbol{R}_1,\boldsymbol{t}_1))}{\partial\,{}^2\boldsymbol{t}_1} \\ &+ \frac{\partial\ -log(p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_2,\boldsymbol{R}_1))}{\partial\,{}^2\boldsymbol{t}_1} = 0 \end{aligned} \quad (29)$$

Based on equation (21), the equation (29) can be decomposed into two terms. For the first term, there is

$$\frac{\partial\ -log(p(d_2|{}^2\boldsymbol{t}_1,\boldsymbol{R}_2,\boldsymbol{R}_2,\boldsymbol{t}_1))}{\partial\ {}^2\boldsymbol{t}_1} = \frac{\hat{d}_2}{\sigma^2}\frac{\partial\ \hat{d}_2}{\partial\ {}^2\boldsymbol{t}_1} \quad (30)$$

In the equation (30), the $\frac{\partial\ \hat{d}_2}{\partial\ {}^2\boldsymbol{t}_1}$ is

$$\begin{aligned} \frac{\partial\ \hat{d}_2}{\partial\ {}^2\boldsymbol{t}_1} &= \frac{\partial\ T(d_2) - (\boldsymbol{T}_1\ {}^2\boldsymbol{T}_1^{-1})_{[3,4]}}{\partial\ {}^2\boldsymbol{t}_1} \\ &= \frac{\partial T(d_2) - (\boldsymbol{t}_1 - \boldsymbol{R}_1\ {}^2\boldsymbol{R}_1^T\ {}^2\boldsymbol{t}_1)_{[3]}}{\partial\,{}^2\boldsymbol{t}_1} \\ &= \frac{\partial T(d_2) - (\boldsymbol{t}_1 - \boldsymbol{R}^*\ {}^2\boldsymbol{t}_1)_{[3]}}{\partial\,{}^2\boldsymbol{t}_1} \\ &= \frac{\partial\ -\boldsymbol{t}_1^{[3]} + \boldsymbol{r}_3^T\ {}^2\boldsymbol{t}_1}{\partial\ {}^2\boldsymbol{t}_1} \\ &= \frac{\partial\ \boldsymbol{r}_3^T\ {}^2\boldsymbol{t}_1}{\partial\ {}^2\boldsymbol{t}_1} \\ &= \boldsymbol{r}_3, \end{aligned} \quad (31)$$

where the $\boldsymbol{r}_3$ is a column vector of the third row of $\boldsymbol{R}^*$. Substituting the equation (31) into the equation (30), it becomes,

$$\begin{aligned}\frac{\hat{d}_2}{\sigma^2}\frac{\partial\,\hat{d}_2}{\partial\,{}^2\boldsymbol{t}_1} &= \frac{\hat{d}_2}{\sigma^2}\boldsymbol{r}_3 \\ &= \frac{T(d_2)-(\boldsymbol{t}_1^{[3]}-\boldsymbol{r}_3^T\,{}^2\boldsymbol{t}_1)}{\sigma^2}\boldsymbol{r}_3 \\ &= \frac{T(d_2)\boldsymbol{r}_3-(\boldsymbol{t}_1^{[3]}-\boldsymbol{r}_3^T\,{}^2\boldsymbol{t}_1)\boldsymbol{r}_3}{\sigma^2}. \end{aligned} \tag{32}$$

The second term in the equation (29) is,

$$\begin{aligned}\frac{\partial - log(p({}^2\boldsymbol{t}_1|\boldsymbol{P},\boldsymbol{P}',\boldsymbol{R}_2,\boldsymbol{R}_1))}{\partial\,{}^2\boldsymbol{t}_1} &= \boldsymbol{\Sigma}_t^{-1}\,{}^2\hat{\boldsymbol{t}}_1\frac{\partial\,{}^2\hat{\boldsymbol{t}}_1}{\partial\,{}^2\boldsymbol{t}_1} \\ &= \boldsymbol{\Sigma}_t^{-1}\,{}^2\hat{\boldsymbol{t}}_1 \end{aligned} \tag{33}$$

Substituting the equation (32) and the equation (33) in the equation (29), there is,

$$\boldsymbol{\Sigma}_t^{-1}\,({}^2\boldsymbol{t}_1-{}^2\bar{\boldsymbol{t}}_1)+\frac{T(d_2)\boldsymbol{r}_3-(\boldsymbol{t}_1^{[3]}-\boldsymbol{r}_3^T\,{}^2\boldsymbol{t}_1)\boldsymbol{r}_3}{\sigma^2}=0 \tag{34}$$

The ${}^2\boldsymbol{t}_1$ is solved as,

$${}^2\boldsymbol{t}_1=(\boldsymbol{\Sigma}_t\boldsymbol{r}_3\boldsymbol{r}_3^T+\sigma^2\boldsymbol{I})^{-1}\{\sigma^2\,{}^2\bar{\boldsymbol{t}}_1-\boldsymbol{\Sigma}_t(T(d_2)-\boldsymbol{t}_1^{[3]})\boldsymbol{r}_3\}. \tag{35}$$

As discussed before, the ${}^2\boldsymbol{t}_1$ in equation (35) is the maximum point of the equation (21). It indicates that the ${}^2\boldsymbol{t}_1$ has been updated by the measurement of the sonar.

E. NONLINEAR OPTIMISATION

In the nonlinear iterative method, the associated Lie algebra $\boldsymbol{\xi}$, which comprises two separate 3-element vectors, is to describe the pose of the vehicle relative to the origin rather than $\boldsymbol{T}$ [62]. This is because it is much more numerically stable in the iterative process. The first 3-element vector in $\boldsymbol{\xi}$ is the rotation vector and the second 3-element vector is the translation vector. According to Lie algebra theory, the exponential map from $\boldsymbol{\xi}$ to $\boldsymbol{T}$ is [62],

$$\boldsymbol{T}=exp(\hat{\boldsymbol{\xi}}). \tag{36}$$

The $\hat{\boldsymbol{\xi}}$ denotes the skew symmetric matrix of the vector $\boldsymbol{\xi}$. It is written in the probability Maximum Likelihood form as,

$$p(\boldsymbol{P},\boldsymbol{P}'|\boldsymbol{\xi}). \tag{37}$$

where, $\boldsymbol{P}$ and $\boldsymbol{P}'$ are regarded as the observed feature points in body coordinates. There is,

$$\begin{aligned}p(\boldsymbol{P},\boldsymbol{P}'|\boldsymbol{\xi})\propto exp(-\frac{1}{2}\sum_{n=1}^{N}\{(\boldsymbol{p}_n'-exp(\hat{\boldsymbol{\xi}})\boldsymbol{p}_n)^T\boldsymbol{\Sigma}_n^{-1} \\ \times(\boldsymbol{p}_n'-exp(\hat{\boldsymbol{\xi}})\boldsymbol{p}_n)\}). \end{aligned} \tag{38}$$

The prior of $\boldsymbol{\xi}$ is obtained from the inertial sensors. The equation is,

$$p(\boldsymbol{\xi})\sim N(\bar{\boldsymbol{\xi}}_{imu},\boldsymbol{\Sigma}_{imu}). \tag{39}$$

The posterior of $\boldsymbol{\xi}$ is,

$$p(\boldsymbol{\xi}|\boldsymbol{P},\boldsymbol{P}')=p(\boldsymbol{P},\boldsymbol{P}'|\boldsymbol{\xi})p(\boldsymbol{\xi}). \tag{40}$$

The optimisation solution is equivalent to minimising the negative log-posterior with respect to the Lie algebra vector,

$$\boldsymbol{\xi}_{min}=\arg\min_{\boldsymbol{\xi}}\{-log(p(\boldsymbol{\xi}|\boldsymbol{P},\boldsymbol{P}'))\}. \tag{41}$$

It can be written as a cost function as,

$$f=\{\frac{1}{2}\sum_{n=1}^{N}\boldsymbol{e}_{pn}^T\boldsymbol{\Sigma}_n^{-1}\boldsymbol{e}_{pn}+\frac{1}{2}\boldsymbol{e}_{imu}^T\boldsymbol{\Sigma}_{imu}^{-1}\boldsymbol{e}_{imu}\} \tag{42}$$

$$\boldsymbol{e}_{pn}=\boldsymbol{p}_n'-exp(\hat{\boldsymbol{\xi}})\boldsymbol{p}_n \tag{43}$$

$$\boldsymbol{e}_{imu}=\boldsymbol{\xi}-\boldsymbol{\xi}_{imu}. \tag{44}$$

Hence, the f is,

$$f=\frac{1}{2}\boldsymbol{E}_f^T\boldsymbol{\Sigma}_f^{-1}\boldsymbol{E}_f. \tag{45}$$

$$\boldsymbol{E}_f=diag(\boldsymbol{e}_{p1},\ \boldsymbol{e}_{p2},\ \ldots,\ \boldsymbol{e}_{pn},\ \boldsymbol{e}_{imu}) \tag{46}$$

$$\boldsymbol{\Sigma}_f=diag(\boldsymbol{\Sigma}_1,\ \boldsymbol{\Sigma}_2,\ \ldots,\ \boldsymbol{\Sigma}_n,\ \boldsymbol{\Sigma}_{imu}). \tag{47}$$

The Jacobian matrix may be written as,

$$\boldsymbol{J}=\frac{\partial\boldsymbol{E}_f}{\partial\boldsymbol{\xi}}=diag(\frac{\partial\boldsymbol{e}_{p1}}{\partial\boldsymbol{\xi}},\ \ldots,\ \frac{\partial\boldsymbol{e}_{pn}}{\partial\boldsymbol{\xi}},\ \frac{\partial\boldsymbol{e}_{imu}}{\partial\boldsymbol{\xi}}) \tag{48}$$

$$\begin{aligned}\frac{\partial\boldsymbol{e}_{pn}}{\partial\boldsymbol{\xi}} &= \frac{\partial(\boldsymbol{p}_n'-\exp(\hat{\boldsymbol{\xi}})\boldsymbol{p}_n)}{\partial\boldsymbol{\xi}} \\ &= \frac{\partial-exp(\hat{\boldsymbol{\xi}})\boldsymbol{p}_n}{\partial\boldsymbol{\xi}} \\ &= -\begin{bmatrix}\boldsymbol{I}_{3\times3} & -[(exp(\hat{\boldsymbol{\xi}})\boldsymbol{p}_n)]_\times \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3}\end{bmatrix}. \end{aligned} \tag{49}$$

The partial derivative of the regularisation term ($\boldsymbol{e}_{imu}$) is,

$$\begin{aligned}\frac{\partial\boldsymbol{e}_{imu}}{\partial\boldsymbol{\xi}} &= \frac{\partial(\boldsymbol{\xi}-\boldsymbol{\xi}_{imu})}{\partial\boldsymbol{\xi}} \\ &= \boldsymbol{I}_{6\times6}. \end{aligned} \tag{50}$$

Hence, with the Jacobian matrix, the iterative algorithm for $\boldsymbol{\xi}$ can be expressed as,

$$\boldsymbol{\Delta}_\xi=-(\boldsymbol{J}^T\boldsymbol{\Sigma}_f\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{\Sigma}_f^{-1}\boldsymbol{E}_f; \tag{51}$$

$$\boldsymbol{\xi}_{new}=\boldsymbol{\xi}_{old}+\boldsymbol{\Delta}_\xi. \tag{52}$$

Compared with the nonlinear optimisation algorithms in other VOs, the IMU's measurement is used to constrain the rotation vector and the translation vector estimated by the pure visual odometry. In equation (40), the IMU constraint is introduced by the probabilistic model. In this case, the effects of the imaging noise can be decreased.

VI. IMPLEMENTATION AND EXPERIMENTS

The IVO-S algorithm was implemented in C++ language. The OpenCV libraries were utilised to process the images, including undistortion of the image, FAST corner detection and image smoothing. The Levenberg Marquardt (LM) algorithm, used to solve the nonlinear least squares problem, was implemented by the Ceres Solver. This is an open source C++ library for solving optimisation problems [63]. Pangolin is a light-weight portable library for managing OpenGL.

It was applied to draw the 3D trajectory and provide a graphical user interface. The hardware platform running the IVO-S includes a 3.2GHz 6-core Intel Core i7 processor with 8GB memory. The computational time is about 0.02 seconds for processing each dual frame with only one thread. Since the computational complexity of the IVO-S is $O(N)$, where the N is number of the feature points, it should be run online on the onboard computer.

The underwater data collection was conducted using the modified ROV. As mentioned previously, there are multiple precise sensors in the watertight tube. These collected data were stored in the onboard computer temporarily. Once the test was finished, the data was copied to a hard disk via WIFI. The different sensors have various sampling rates. During the data collection, the timestamps of different sensors are recorded as well. Since the data from different sensors will have consistent timestamps, the multi-sensors data can be associated by matching the nearest timestamps.

The modification changed the centres of gravity and buoyancy of the vehicle, and had a negative impact on the manoeuvrability of the vehicle. Hence, an operator held the vehicle with two half meter sticks, and moved it along the reference line of the test. Simultaneously, the sensors in the watertight tube kept recording the data.

The experiments were conducted in the towing tank at Newcastle University [64]. A square made of straps was set out, providing the reference path. The scale and 3D pictures of the square are illustrated in Fig.10. The vehicle was controlled by the operator within the square. Hence, the vehicle is operated to travel along the path drawn by the straps, as shown in Fig.11. The operator holding the vehicle tried to follow the reference straps however, it was difficult to follow the reference path exactly, particularly where there were sharp turns.

The towing tank is a sparse environment but has random features due to the deterioration of the coating of the bottom, which are not regular patterns. In addition, the wall of the tank and the bases of the reference straps mean that the bottom is not totally flat. Hence, the towing tank is similar to some real seabed environments, but it is more harsh for visual odometries because of the sparse feature points.

VII. EXPERIMENTAL RESULTS

The tests were conducted by operating the vehicle to travel along different reference paths. Meanwhile, the IVO-S method would estimate the trajectory of the vehicle. Actually, the navigation method localises the onboard camera, rather than vehicle. Because the camera is fixed on the vehicle, the position of vehicle can be obtained by multiplying by the transformation matrix. The shapes of the reference paths were: a square, a triangle and a figure-8 shape. In each test, the vehicle was expected to return to the starting point at the end. In other words, each estimated trajectory is supposed to be closed. However due to accumulated errors, the final pose of the ROV in the estimated trajectory has an offset relative to the starting point. The error ratio can be calculated by dividing the position error by the length of whole trajectory,

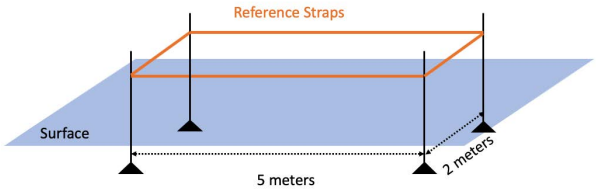


FIGURE 10. Square strap 3D in towing tank.

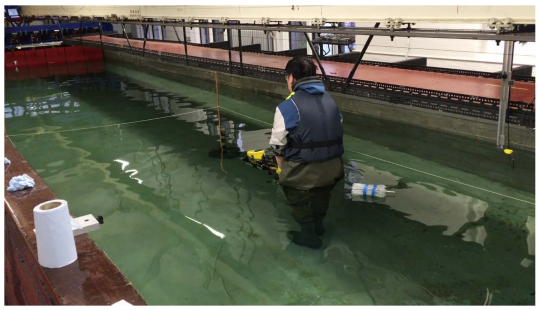


FIGURE 11. Square strap and operator.

which is,

$$e_r = \frac{offset}{trajectory\ length}. \tag{53}$$

Such an evaluation method is popular in verification of navigation system performance [65]. The IVO-S estimated 3D trajectories originally. In order to present results clearly, they have been converted into 2D figures. Each path shape was tested multiple times to prove the repeatability of the proposed method. The error measured in 3D space and length of estimated trajectory in each test is presented in Table 2. The comparative results presented in the following sections are from the IVO-M [45] methods and other popular visual SLAMs or odometries, such as ORB-SLAM2, SVO, and OKVIS, are provided as well.

TABLE 2. Error ratio and distance of each test using IVO stereo method.

Shape	Distance(meter)	Error Ratio
Triangular 1st	14.3150	0.0080
Triangular 2nd	12.9853	0.0183
Triangular 3rd	12.1408	0.0115
Square 1st	15.6826	0.0189
Square 2nd	15.6336	0.0139
Square 3rd	16.5278	0.0118
Figure-8 1st	18.3097	0.0161
Figure-8 2nd	20.3615	0.0054

A. ESTIMATED TRAJECTORIES FROM THE IVO-S AND THE IVO-M METHODS

The estimated trajectories from the IVO-S method are plotted in 2D figures, shown from Fig.12 to Fig.19. The comparisons between results from IVO-S and IVO-M are presented in Fig.12, Fig.15, and Fig.18. It should be noted that the data for IVO-S method and the IVO-M method are collected simultaneously. Since the data for the IVO-M method is collected by

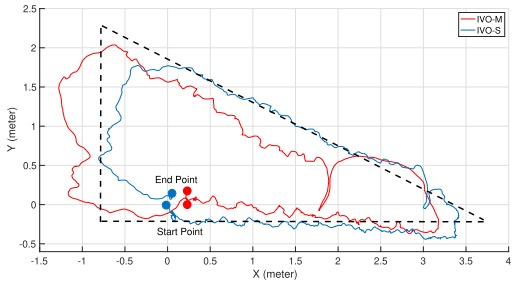


FIGURE 12. Triangular 1st, comparison with IVO-S and IVO-M. Error ratio of IVO-S is 0.0080, error ratio of IVO-M is 0.00510.

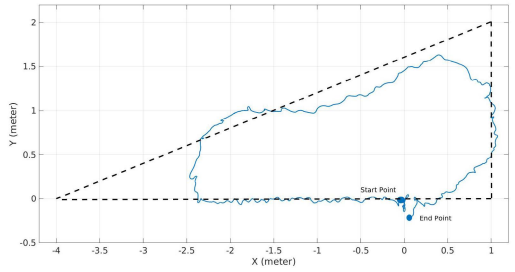


FIGURE 13. IVO-S Triangular 2nd, error ratio: 0.0183.

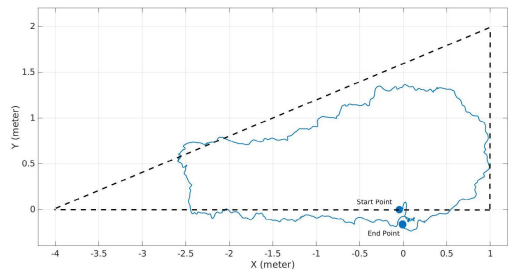


FIGURE 14. IVO-S Triangular 3rd, error ratio: 0.0115.

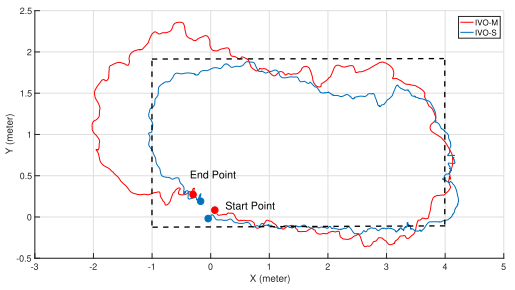


FIGURE 15. Square 1st, comparison between IVO-S and IVO-M. Error ratio of IVO-S is 0.0189, error ratio of IVO-M is 0.0560.

the monocular camera which locates at the front vehicle, the start points of the IVO-M method is ahead about 0.2 meter in the x-axis relative to those of the IVO-S method. The reference path is drawn in black dotted lines in these results. The experiments were conducted eight times in total: the triangular shape was tested three times, the square shape was tested three times, and the figure-8 shape was tested twice.

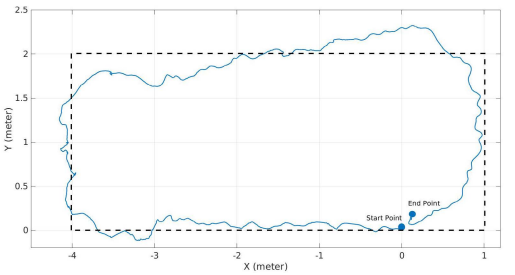


FIGURE 16. IVO-S Square 2nd, error ratio: 0.0139.

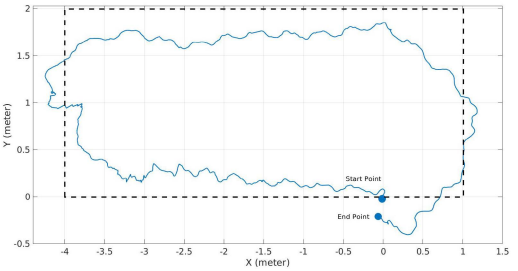


FIGURE 17. IVO-S Square 3rd, error ratio: 0.0118.

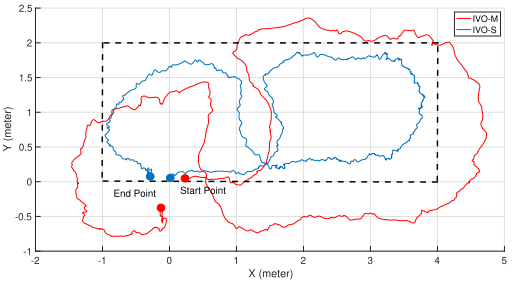


FIGURE 18. Figure-8 1st, comparison between IVO-S and IVO-M. Error ratio of IVO-S is 0.0161, error ratio of IVO-M is 0.0435.

1) DISCUSSION OF RESULTS FROM THE IVO-S AND THE IVO-M METHODS

According to the experimental results, the IVO-S method estimated the trajectories accurately. However, there are larger drifts on the trajectories estimated by the IVO-M algorithm. The reason is that the IVO-M algorithm does not remove the unreliable feature points, and lacks the sonar fusion algorithm to correct the translation vectors. In particular, in the IVO-M method, the noise from the feature point extraction algorithm is imposed on the linear calculation directly. In contrast, in the IVO-S method, the 3D feature points are reconstructed, and are processed by the sonar filter. In this case, the points with large disturbances can be moved out.

In Fig.12 to Fig.14, the vehicle was forced to follow a triangular path by the operator. Because of the narrow space in each corner, the operator could not follow the reference path exactly. In this way, the vehicle had to make a turn in advance rather than moving along the sharp angles. In the triangular-shape tests, the estimated position almost returned to the origin point, and the error ratios are about 1% by the IVO-S method. The results from the IVO-S and IVO-M

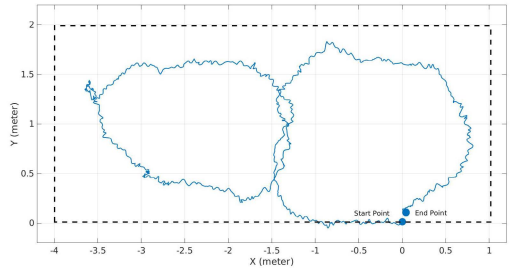


FIGURE 19. IVO-S Figure-8 2nd, error ratio: 0.0054.

methods are compared in Fig.12. Although the IVO-M method has a lower error ratio, the trajectory estimated by the IVO-S method is more rational, because it almost follows the reference path exactly, and is inside the triangle strap.

From Fig.15 to Fig.17, the IVO-S was evaluated over square reference paths. It is obvious that the estimated trajectories approximated a square shape. As mentioned before, because the vehicle was held by an operator, it could not follow the reference path exactly, and it was particularly hard at the corners for the operator to turn the vehicle exactly along the vertical angles. Along some segments of the estimated trajectories, the vehicle is estimated to go across the reference path. Since the operator moved close to the reference square holding the vehicle with half meter sticks, the vehicle may have crossed the reference line sometimes. In Fig.16 and Fig.17, the IVO-S has about 1% error ratio. The comparison between the IVO-S and the IVO-M methods is shown in Fig.15. Similar to the triangle-shape path, the trajectory estimated by the IVO-S method almost follows the reference path exactly, and is inside the square strap. Furthermore, it suffers from less drift error.

The vehicle was operated to follow a figure-8 shape within the reference rectangle in Fig.18 and Fig.19. The IVO-S method drew approximately figure-8 shape trajectories. In Fig.19, the error ratio is only 0.54%. In the Fig.18, there is a comparison between results from the IVO-M and IVO-S methods. The trajectory generated by the IVO-S method, not only suffers from less drift error, but also stays inside the square strap.

B. COMPARISONS BETWEEN IVO-S AND OTHER VISUAL NAVIGATION METHODS

The four different Visual SLAM algorithms and the IVO-S algorithm were compared on the same dataset (a triangular shape). The calibration parameters and coordinate transformation matrices in these Visual SLAMs were corrected according to the customised dataset. These open source algorithms were: ORB-SLAM2, SVO, VINS-Mono, and OKVIS. The ORB-SLAM2 method has been used for localising AUVs and mapping an underwater cave [65], [66]. The VINS-Mono and OKVIS algorithms with integrated inertial information have been implemented to localise cameras in underwater environments [43], [44]. These algorithms performed well in context-dense environments, however they performed less successfully in the underwater sparse-feature and complex

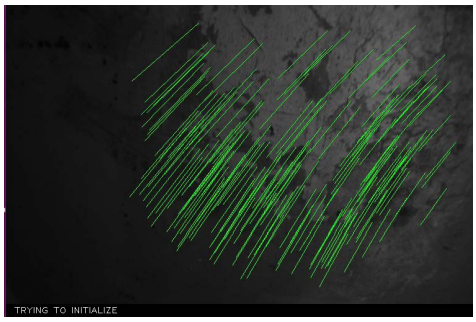


FIGURE 20. ORB-SLAM2 trying to initialise with the monocular dataset.

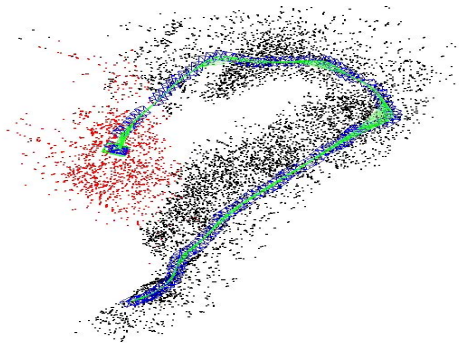


FIGURE 21. The trajectory estimated by ORB-SLAM2 Monocular (The green line is the estimated trajectory, black dots are measured feature points, red dots are active feature points.)

magnetic environment. The SVO developed from the direct method tracks invariant intensity values in images to recover the camera motion. However, at least 100 feature points are required to boot. Because of that, the SVO was stuck in the bootstrap when it was evaluated with the customised dataset. The other methods were developed from the feature-based method. Therefore, the quality and quantity of feature points determine the performances of these methods. In underwater sparse-feature environments, having few high-quality feature points may result in failures of these methods. More explicitly, the ORB-SLAM2 monocular algorithm was hardly able to identify enough reliable feature points to finish the initialisation process as shown in Fig.20, and lost the matched feature points frequently in the sparse-feature underwater environment. Hence, the trajectory estimated by ORB-SLAM2 is not complete, as shown in Fig.21. The ORB-SLAM2 stereo algorithm could not extract any feature points from the customised stereo dataset, as shown in Fig.22.

The performances of VINS-Mono and OKVIS algorithms on the dataset were unsatisfactory. The complex magnetic field in the towing tank may have increased the drift error of the low-cost IMU development kit. More seriously, it was difficult for these feature-based methods to constrain IMU noise using the few reliable feature points. In this case, unbounded IMU errors caused a large drift error. The feature detection processes of the two SLAMs are presented in Fig.23 and Fig.24. There is a common problem with the VINS-Mono, the OKVIS, and V-SLAM: unconstrained drift error, as shown in Fig. 23 and Fig.25.

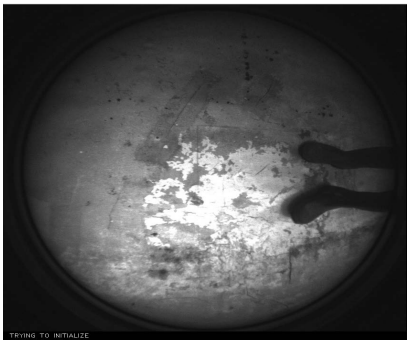


FIGURE 22. ORB-SLAM2 Stereo Feature Detection failed with the stereo dataset.

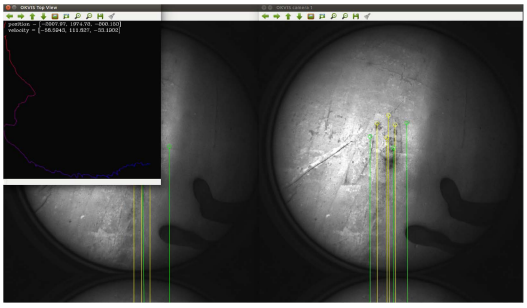


FIGURE 23. OKVIS with few matched feature points and huge drift error.

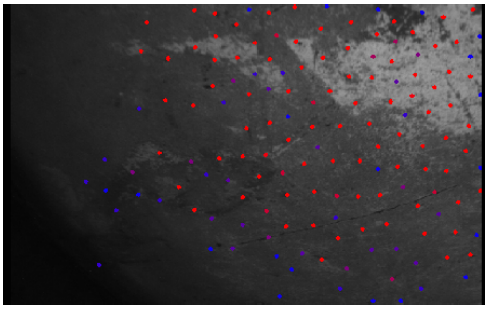


FIGURE 24. VINE-Mono trying to extract feature points.

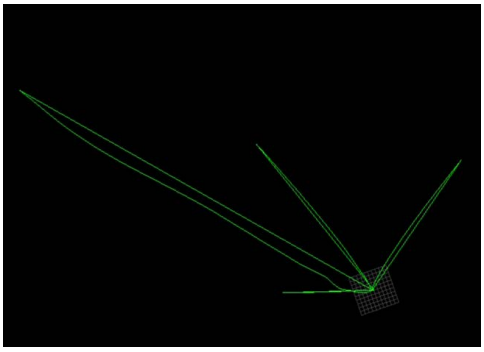


FIGURE 25. Huge drift error from the VINS-Mono (The green line is the estimated trajectory in 3D space.)

Conversely, the IVO-S method implements the FAST algorithm for extracting feature points and the Optical Flow algorithm for tracking these feature points, rather than utilising relatively complex descriptors to match corresponding points.

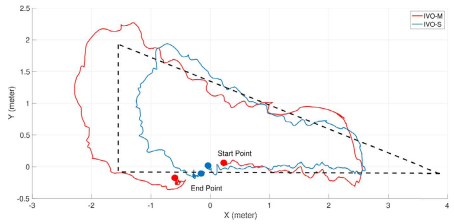


FIGURE 26. The trajectory estimated following the triangle pattern by the IVO-S (error ratio: 0.0183) and IVO-M (error ratio: 0.0559).

In addition, it integrates the sonar and inertial sensor information to skip the bootstrap. Unlike other inertial visual SLAMs based on tightly coupled algorithms, the linear estimations and the fusion algorithm are processed prior to nonlinear optimisation. Therefore, these unique characteristics mean that the IVO-S method performs well in the underwater sparse-feature environment. Compared with the IVO-M method, the IVO-S has the higher precision, since it can reconstruct depths of feature points directly and remove unreliable feature points. The trajectories estimated by the IVO-S and the IVO-M methods are shown in Fig.26.

VIII. CONCLUSION

The novel underwater navigation method IVO-S has been investigated in this paper. Compared with typical underwater navigation methods, the IVO-S is designed to offer a high precision underwater navigation solution at low cost. In addition, the proposed method can perform well in underwater sparse-feature environments with high precision, while other visual slams or odometries such as ORB-SLAM2 and OKVIS perform unsuccessfully in the same situations. In the future, the IVO-S method will be extended by introducing closure loop detection and map reconstruction approaches.

APPENDIX
CALIBRATION PARAMETER

A. CAMERA MATRICES AND DISTORTION COEFFICIENTS

See Tables 3 and 4.

TABLE 3. Intrinsic parameters of the left camera.

Camera Sensor ID	1
Width	848
Height	800
p_x	426.271
p_y	406.796
f_x	283.636
f_y	284.597
$Distortion$	0.00024637 0.0346615 -0.032878 0.00487357 0

TABLE 4. Intrinsic parameters of the right camera.

Camera Sensor ID	2
Width	848
Height	800
p_x	430.920
p_y	404.833
f_x	283.714
f_y	284.799
$Distortion$	-0.000605872 0.0392776 -0.0380382 0.00672605 0

B. EXTRINSIC PARAMETERS

The extrinsic parameters build the homogeneous transformation $\boldsymbol{T}_e$ from the left camera to the right camera.

$$\boldsymbol{T}_e = \begin{bmatrix} \boldsymbol{R}_e & \boldsymbol{t}_e \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}$$

where,

$$\boldsymbol{R}_e = \begin{bmatrix} 0.999965 & 0.00281616 & -0.00787895 \\ -0.00278412 & 0.999988 & 0.00407375 \\ 0.00789033 & -0.00405167 & 0.0644494 \end{bmatrix}$$

and,

$$\boldsymbol{t}_e = [0.0644494 \ -0.000108581 \ -0.000456553]^T.$$

Applying the *stereoRectify()* in OpenCV library with $\boldsymbol{T}_e$, the stereo rectification can be achieved.

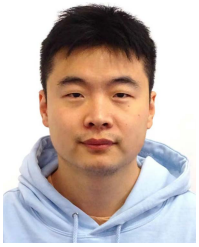
ACKNOWLEDGMENT

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

REFERENCES

[1] N. Braathen and A. Sandford, "Pipeline inspection by ROV," in *Submersible Technology*. Dordrecht, The Netherlands: Springer, 1986, pp. 313–318.
[2] D. MacGregor, "Fibre optics improving deepwater ROV pipeline inspection," presented at the SPE Offshore Eur., Aberdeen, U.K., Sep. 1983.
[3] R. Bogue, "Underwater robots: A review of technologies and applications," *Ind. Robot, Int. J.*, vol. 42, no. 3, pp. 186–191, May 2015.
[4] M. Ludvigsen, F. Soreide, K. Aasly, S. Ellefmo, M. Zylstra, and M. Pardey, "ROV based drilling for deep sea mining exploration," in *Proc. OCEANS*, Jun. 2017, pp. 1–6.
[5] S. Chatzichristofis, A. Kapoutsis, E. B. Kosmatopoulos, L. Doitsidis, D. Rovas, and J. B. de Sousa, "The NOPTILUS project: Autonomous multi-AUV navigation for exploration of unknown environments," *IFAC Proc. Volumes*, vol. 45, no. 5, pp. 373–380, 2012.
[6] S. D. McPhail, M. E. Furlong, M. Pebody, J. R. Perrett, P. Stevenson, A. Webb, and D. White, "Exploring beneath the PIG ice shelf with the Autosub3 AUV," in *Proc. OCEANS*, May 2009, pp. 1–8.
[7] D. R. Blidberg. (2020). *The Arctic Explorer AUV Spec Sheet*. Accessed: Apr. 4, 2020. [Online]. Available: http://https://auvac.org/configurations/view/206
[8] C. Kaminski, T. Crees, J. Ferguson, A. Forrest, J. Williams, D. Hopkin, and G. Heard, "12 days under ice—An historic AUV deployment in the Canadian high Arctic," in *Proc. IEEE/OES Auton. Underwater Vehicles*, Sep. 2010, pp. 1–11.
[9] I. Tena, "Automating rov operations in aid of the oil & gas offshore industry," SeeByte, Edinburgh, U.K., White Paper, 2011, pp. 1–9. [Online]. Available: https://www.unmannedsystemstechnology.com/wp-content/uploads/2013/10/White-Paper-Automating-ROV-Operations.pdf
[10] 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. 2013.
[11] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.
[12] 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.
[13] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robot.*, vol. 23, no. 1, pp. 3–20, 2006.
[14] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1015–1026, Oct. 2008.
[15] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 15–22.
[16] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, Apr. 2020.
[17] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.
[18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[20] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 834–849.
[21] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.
[22] 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.
[23] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3662–3669.
[24] M. Caccia, "Vision-based ROV horizontal motion control: Near-seafloor experimental results," *Control Eng. Pract.*, vol. 15, no. 6, pp. 703–714, Jun. 2007.
[25] 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.
[26] 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, Dec. 2006.
[27] 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. Robots Syst.*, Oct. 2009, pp. 1559–1565.
[28] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.
[29] 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.
[30] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Lenoard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, Oct. 2012.
[31] 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.
[32] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with underwater robot localization and tracking," in *Proc. IEEE Int. Conf. Robot. Automat.*, Apr. 2007, pp. 4556–4561.
[33] M. Hildebrandt and F. Kirchner, "IMU-aided stereo visual odometry for ground-tracking AUV applications," in *Proc. OCEANS*, May 2010, pp. 1–8.
[34] S. Wirth, P. L. Negre Carrasco, and G. O. Codina, "Visual odometry for autonomous underwater vehicles," in *Proc. MTS/IEEE OCEANS*, Jun. 2013, pp. 1–6.
[35] P. L. N. Carrasco, F. Bonin-Font, and G. O. Codina, "Stereo graph-SLAM for autonomous underwater vehicles," in *Intelligent Autonomous Systems 13*. Cham, Switzerland: Springer, 2016, pp. 351–360.
[36] F. Bellavia, M. Fanfani, and C. Colombo, "Selective visual odometry for accurate AUV localization," *Auton. Robots*, vol. 41, no. 1, pp. 133–143, 2017.
[37] M. Nawaf, D. Merad, J.-P. Royer, J.-M. Boï, M. Saccone, M. B. Ellefi, and P. Drap, "Fast visual odometry for a low-cost underwater embedded stereo system," *Sensors*, vol. 18, no. 7, p. 2313, Jul. 2018.
[38] J. Zhang, V. Ila, and L. Kneip, "Robust visual odometry in underwater environment," in *Proc. MTS/IEEE Kobe Techno-Oceans (OTO)*, May 2018, pp. 1–9.
[39] 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, p. 687, Feb. 2019.
[40] B. Teixeira, H. Silva, A. Matos, and E. Silva, "Deep learning for underwater visual odometry estimation," *IEEE Access*, vol. 8, pp. 44687–44701, 2020.

[41] P. Dabove, V. Di Pietra, and M. Piras, "Monocular visual odometry with unmanned underwater vehicle using low cost sensors," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2020, pp. 810–816.
[42] V. Creuze, "Monocular odometry for underwater vehicles with online estimation of the scale factor," in *Proc. IFAC World Congr.*, 2017, pp. 1–6.
[43] S. Rahman, A. Q. Li, and I. Rekleitis, "Sonar visual inertial SLAM of underwater structures," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–7.
[44] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: An underwater SLAM system using sonar, visual, inertial, and depth sensor," 2018, *arXiv:1810.03200*.
[45] Z. Xu, M. Haroutunian, A. J. Murphy, J. Neasham, and R. Norman, "An integrated visual odometry system for underwater vehicles," *IEEE J. Ocean. Eng.*, vol. 46, no. 3, pp. 844–863, Dec. 2020.
[46] Life Performance Research. (2018). *LPMS NA2 Introduction*. Accessed: Jan. 11, 2020. [Online]. Available: https://lp-research.com/lpms-nav2/
[47] Intel Corporation. (2018). *Intel Realsense T265 Tracking Camera*. Accessed: Jan. 10, 2020. [Online]. Available: https://www.intelrealsense.com/tracking-camera-t265/
[48] 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 Robots (ECMR)*, Sep. 2019, pp. 1–6.
[49] 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. [Online]. Available: http://www.aerialroboticscompetition.org/assets/downloads/2019SymposiumPapers/UniversityofMichigan.pdf
[50] M. Pelka, K. Majek, and J. Bedkowski, "Testing the affordable system for digitizing USAR scenes," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Sep. 2019, pp. 1–2.
[51] Y. Kwon and J. B. Casebolt, "Effects of light refraction on the accuracy of camera calibration and reconstruction in underwater motion analysis," *Sports Biomech.*, vol. 5, no. 2, pp. 315–340, Jul. 2006.
[52] J.-M. Lavest, G. Rives, and J.-T. Lapresté, "Underwater camera calibration," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2000, pp. 654–668.
[53] L. Huang, X. Zhao, X. Huang, and Y. Liu, "Underwater camera model and its use in calibration," in *Proc. IEEE Int. Conf. Inf. Autom.*, Aug. 2015, pp. 1519–1523.
[54] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.
[55] M. Parchami and G.-L. Mariottini, "A comparative study on 3-D stereo reconstruction from endoscopic images," in *Proc. 7th Int. Conf. Pervasive Technol. Rel. Assistive Environments*, May 2014, pp. 1–8.
[56] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 430–443.
[57] K. G. Derpanis, "The Harris corner detector," York Univ., Toronto, ON, Canada, Tech. Rep., 2004, pp. 1–2.
[58] 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.
[59] X. Jin and K. Hirakawa, "Approximations to camera sensor noise," *Proc. SPIE*, vol. 8655, Feb. 2013, Art. no. 86550H.
[60] R. D. Gow, D. Renshaw, K. Findlater, L. Grant, S. J. McLeod, J. Hart, and R. L. Nicol, "A comprehensive tool for modeling CMOS image-sensor-noise performance," *IEEE Trans. Electron Devices*, vol. 54, no. 6, pp. 1321–1329, Jun. 2007.
[61] T. Jia, X. Shen, and H. Wang, "Multistatic sonar localization with a transmitter," *IEEE Access*, vol. 7, pp. 111192–111203, 2019.
[62] J.-L. Blanco, "A tutorial on SE (3) transformation parameterizations and on-manifold optimization," Univ. Malaga, Málaga, Spain, Tech. Rep. 012010, 2010, vol. 3.
[63] S. Agarwal and K. Mierle. (2019). *Ceres Solver*. [Online]. Available: http://ceres-solver.org
[64] Newcastle University. (2010). *Hydrodynamic Lab in Newcastle University*. Accessed: Jan. 19, 2020. [Online]. Available: https://www.ncl.ac.U.K./engineering/about/facilities/marineoffshoresubseatechnology/hydrodynamics/#towingtank
[65] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2174–2181.
[66] N. Weidner, S. Rahman, A. Q. Li, and I. Rekleitis, "Underwater cave mapping using stereo vision," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5709–5715.



ZHIZUN XU received the B.Eng. degree from Shanghai Maritime University, in 2014, the M.Eng. degree from Mokpo National Maritime University, South Korea, in 2016, and the Ph.D. degree in marine technologies from Newcastle University, U.K., in 2021. He is currently a Lecturer with Marine College, Guangdong Ocean University. His research interests include 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, the M.Sc. degree in naval architecture from Newcastle University, U.K., in 2009, and the Ph.D. degree from Newcastle University for her research on bio-inspiration to improve the performance of unmanned underwater vehicles which was sponsored by the EPSRC-UK, in 2014. She became a Lecturer of marine technology at Newcastle University, in 2015, where she is currently a Naval Architect. Her research interests include marine hydrodynamics, manoeuvring and seakeeping simulation of marine vehicles, and 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 an Officer at the U.K. Merchant Navy, in 1996. He joined Newcastle University, in 2007, where he is currently a Reader in maritime engineering with the School of Engineering. His research interests include sustainable shipping, 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 a Research Associate working on the research and commercial product development in underwater acoustic communication, sonar imaging, and wireless sensor networks at Newcastle University, in 2007, where he is currently a Senior Lecturer of communications and signal processing with the School of Electrical and Electronic Engineering. 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 include 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 the Principle Engineer at Switched Reluctance Drives Ltd., in 2004, where she is currently 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.