American Journal of Mechanical Engineering, 2020, Vol. 8, No. 2, 76-87
Available online at http://pubs.sciepub.com/ajme/8/2/5
Published by Science and Education Publishing
DOI:10.12691/ajme-8-2-5

Monocular Visual SLAM for Underwater Navigation in Turbid and Dynamic Environments

Chinthaka Amarasinghe[1,*], Asanga Ratnaweera[2], Sanjeeva Maitripala[2]

[1]Department of Science & Technology, Uva Wellassa University, Badulla, Sri Lanka
[2]Department of Mechanical Engineering, University of Peradeniya, Peradeniya, Sri Lanka
*Corresponding author: chinthakaa@uwu.ac.lk

Received July 17, 2020; Revised August 19, 2020; Accepted August 28, 2020

Abstract Localization, navigation, and mapping using vision-based algorithms are an active topic in underwater robotic applications. Although many algorithms developed in recent years, especially in the ground and areal robotic communities, directly applying those methods in underwater navigation remain challenging due to the visual degradation induced by the medium. In this paper, we proposed UW-SLAM (Underwater SLAM), a new monocular visual SLAM algorithm focused on the underwater environment which addresses the turbidity and dynamism. The proposed method was evaluated with several underwater datasets with comparison to the state of the art monocular SLAM methods.

Keywords: *monocular visual navigation, underwater vision, visual SLAM*

Cite This Article: Chinthaka Amarasinghe, Asanga Ratnaweera, and Sanjeeva Maitripala, “Monocular Visual SLAM for Underwater Navigation in Turbid and Dynamic Environments.” *American Journal of Mechanical Engineering*, vol. 8, no. 2 (2020): 76-87. doi: 10.12691/ajme-8-2-5.

1. Introduction

Exploration of the oceans and shallow waters is attracting the interest of many industries and institutions all over the world, because of the valuable resources, the knowledge that it houses for scientists, and also for rescue purposes. For the past decades, remotely operated vehicles (ROV) are the widely used method for the exploration of the underwater environment. ROVs operated using a wired connection between the operator and the vehicle which limits usability and maneuverability. Due to these limitations, ROVs are now replaced by Autonomous Underwater Vehicles (AUVs). Although AUVs offers unique advantages over ROV and also present a uniquely challenging navigational problem as they operate autonomously in a highly unstructured environment where satellite-based navigation isn’t directly available [1]. Navigation plays a significant role in the operation of AUVs and consists of two fundamental aspects localization and mapping [2]. Currently used methods for AUV navigation can be grouped into three categories [1].

1. Inertial / Dead Reckoning
2. Acoustic transponders and modems techniques
3. Geophysical navigation

An inertial navigation system (INS) is navigation that uses a processing unit, motion sensors (accelerometers), rotational sensors (gyroscopes), and magnetic sensors (magnetometers) to continuously calculate the dead reckoning the velocity, orientation, and the position of the moving object without any need for external references. All of the methods in this category have position error growing with time and need to be corrected by an external reference. Acoustic transponders and modems are used to measure the time-of-flight (TOF) of the sound signals underwater. Different types of acoustic-based sensors such as Doppler Velocity Log (DVL), Mechanically Scanning Imaging Sonar (MSIS), Underwater Acoustic Positioning System (UAPS), bathymetric sonars, Side scan, etc have been developed. On the other hand, geophysical navigation techniques need external environmental information as references for navigation. This is achieved by detecting, identifying, and classifying of environmental features by using various kinds of sensors such as Cameras, Laser range finders, magnetic sensors, pressure/depth sensors, and processing those sensor data with effective fusion algorithms [1].

Sonar-based (Acoustic Navigation) methods are the most extended approach in the underwater robotics community, because of the good properties of sound propagation in the water. However, these are more suited for long distances underwater missions and not for short-range missions (below 1 m) [1]. The high cost of the acoustic sensors and the infrastructure required for acoustic-based sensing limit their usability in small-scale AUVs and bio-inspired vehicles such as Robotic fish [2]. Further, most of the underwater missions are used for inspection purposes that require sub metric accuracy, which is difficult and expensive to achieve with acoustic type sensors. However, when navigating close to the seabed or the inspection structures, visual information becomes available and cameras can be used as an inexpensive alternative to sonar based sensors.

Underwater vision navigation systems are already implemented in many applications including underwater infrastructure inspection and maintenance, power or gas line inspection, transmission or communications cable monitoring, marine life monitoring, military missions, deep underwater seabed reconstruction, inspection of sunken old ships, ship hull inspection, etc. Vision is essential for all these applications, either as a main navigation sensor or as a supplement for other navigation systems such as sonar. Therefore there is a high motivation to improve the vision-based underwater navigation techniques by expanding its independence, capabilities, and utility [3].

One of the main objectives of this research is to develop a vision based Simultaneous Localization and Mapping (SLAM) navigation algorithm as the main navigational method. Several recent investigation in to this can be found in [1,4,5,6,7]. Currently, vision based navigation is an Active research area specially in ground robotic communities. These systems can be categorized according to the type of sensors that they used as mono vision, stereo vision, RGB-D, and Lidar. Even though many solutions are available, most of them have failed in underwater. These failures occur due to many reasons such as decrease of operating range dramatically in muddy or turbid waters, light scattering, light wave attenuation in the medium, image poorly contrast and hazy, and image blur etc. Therefore, the images need extensive correction before using them in a navigational algorithm. There have been many researches published regarding the underwater image correction and enhancement. It has been observed though out the literature that most of the image correction and enhancement methods are targeted at improvement of the visible quality of the standalone images, whereas very few of them looked at the improvement for visual navigation, such as detection and tracking quality of the feature on the entire images [8,9,10].

It is noted that the mono SLAM system is more challenging compare to other systems and fairly easy and cost-effective to implement [11]. The camera used as a way of observation in most of the underwater vehicles can be used for navigation purposes. On the other hand, Stereo cameras are way more expensive compared to the mono cameras and stereo 3D observation fails when the scene of observation is far away from the baseline. In such a situation, stereo systems are meant to work as a mono system. RGB depth cameras are a good sensor to observe the depth information of the scene and a frequent choice for ground robotic navigation [12,13]. However, in underwater conditions RGB-D camera range decreases drastically as the IR wavelength absorbs by the water. It has experimented that RGB-D camera works effectively only up to 20cm in underwater [14,15]. To navigate using depth cameras, AUV needs to be very closer to the ocean bottom (lower than the 20 cm) and it is not recommended as the ocean floor is highly unstructured. Further, many other factors are affecting the quality of the observed data from depth sensors and need extensive corrective actions for effective use.

Several articles on the use of laser sensors, such as Light Detection and Ranging (LIDAR) in use for the underwater SLAM are reported in the literature. LIDAR uses a laser beam projection by emitting a very powerful laser beam that can hardly be weakened by water. Therefore, a vision with laser can recover more accurate localization than a single camera [2]. Despite the cost, LIDAR is a good choice for observing 3D metric measurement of the environment. They are a very popular choice on ground robotic navigation. Even though low-cost LIDAR are available, they are not meant to use underwater, as those LIDAR need extensive calibration to operate underwater. There are specifically designed underwater LIDARs which mainly used for underwater structure inspection purposes. As far as the cost is concerned, LIDAR solutions are expensive for small scale underwater vehicles. As such this research focuses on the development of a mono camera-based navigation system for the AUV.

Mono camera navigation is a well rich research topic in the ground and areal robotic communities. A few attempts of mono camera navigation in underwater robots are reported in the literature and will be discussed in the review, in section 2. There are three main types of mono camera-based navigation systems: Direct method, Feature-based method, and hybrid method. In the direct method the intensity values of each and every pixel match with the other image pixel intensity values on an epipolar line. In the feature-based method, a set of defined feature points, that are matched across the image frames. The third category is a hybrid method where a combination of both was taken in to action. In this research, more focus is given on feature point-based methods as almost all of the vision-based underwater navigation systems were designed using feature point-based methods and direct methods have a tendency to frequent failing in underwater conditions [16]. In underwater conditions, most of the scenes can be observed by a down-looking camera, and most of the cases, observe scene is flat. Due to low light conditions and the use of artificial light in the low texture underwater environment, the observed scene has nearly the same level of intensities in most of the pixels, which makes a lot of false positives in direct based methods. As a result, many direct based methods tend to fail in underwater.

In this research, a feature point-based method is used to build up the navigation algorithm. AUVs are slow-moving vehicles and it is obvious that the scene will not move drastically between each successive frame. This fact motivates us to use Kanade-Lucas-Tomasi (KLT) [17] point tracker for the frame to frame tracking. Corner points were detected using Harris detector [18]. This combination of detection and tracking resulted in a low computational load compared to the descriptor based tracking method. Also, it is observed that the descriptor-based tracking method fails in underwater conditions with increase turbidity levels as described by Maxime Ferrera et al [16]. Further, they showed Harris corner detector combined with KLT point tracker gives a better result in high turbidity environments. The feature detector evaluation for underwater images presented in [19] and [20] also stated the robustness of the

Harris corner detector and KTL for different turbidity images.

In this paper, we proposed UW-SLAM (Underwater SLAM), a new monocular visual SLAM algorithm with loop closing capabilities dedicated to the underwater environment with all the major components of a complete visual SLAM [21], which include visual initialization, data association, pose estimation, map generation, BA/PGO/map maintenance, failure recovery, and loop closure. In addition, an image preprocessing functionality is introduced to overcome the visual degradations, mentioned above. The proposed system consists of three threads and works on a key frame-based method with optimization to correct the non-linear error. A hybrid model using both descriptor and non-descriptor based feature points were proposed that effectively incorporate the advantages of both methods. Harris corner detector and KTL tracker are used as the frame to frame feature tracking purpose and SURF feature to detect the loop closer. The front end of our system consists of image acquisition, image preprocessing (enhancement), and feature point detection and tracking. The back end of the algorithm consists of Bundle adjustment Loop closer detection and pose graph optimization to ensure minimal drift.

This paper contributions are as follows:

- Development of hybrid tracking method based on descriptors and non-descriptor based feature points for underwater visual SLAM.
- Development of UW-SLAM: a monocular Visual SLAM with loop closer robust to turbidity and short occlusions.
- Modified bag of feature loop closer detection system for underwater navigation match between groups of keyframes with clustering.
- Large scale operation using submap with scale correction during the loop closer.
- Comparison of the Proposed UW-SLAM with state-of-the-art open-source monocular Visual odometry and Visual-SLAM algorithms on two underwater datasets.

The rest of the paper is organized as follows. A review of recent research work related to underwater visual navigation is reported in section 2. In Section 3, the development of the UW-SLAM is described and section 4 describes the modified loop closer method. Finally, the evaluation of the UW-SLAM with two publically available data sets, one with syntactically made and one with real data from an underwater mission is described in section 5.

2. Related Work

The localization of robots from the output of a single camera system has been an active topic of research for the last fifteen years. Generally, visual SLAM can be categorized into two sections: filter-based and non-filter based. Filter based solutions are more common before 2010 and non-filter based methods attracted attention thereafter [21]. Strasdat *et. al* stated that key frame-based techniques are more accurate than filtering based techniques for the same computational cost [22]. Several surveys for the general SLAM reported in the literature, but only a few of them address the monocular SLAM extensively. More recent surveys can be found in [21,23,24] where [23] describes a complete mono SLAM problem, [21] describes non-filter keyframe based SLAM and [24] describes filter-based visual SLAM extensively. Most of the SLAM problems were developed based on mobile robot and ground vehicles but later extended to underwater robot communities with extended difficulties.

The earliest system developed based on the most characteristic of a keyframes SLAM is probably PTAM by Klein and Murray [25]. They introduce the idea of splitting tracking and mapping in separate threads. This was the first attempt in using bundle adjustment in real-time. After that, many techniques were developed based on PTAM. Strasdat *et al.* [26] added a loop closer with pose graph optimization using similarity constraints (7DoF). 7DoF can correct scale drift in monocular SLAM. Pirker *et al.* [27] The proposed CD-SLAM is a complete system, including loop closure, repositioning, large-scale activation, and efforts to work on dynamic environment. Ra´ul Mur-Artal *et al.* [28] develop Orb SLAM, which uses Orb features for all the SLAM tasks. Further, recently published dense and semi-dense methods such as DTAM [29] and LSD-SLAM [30] are also getting attraction, especially in UAV navigation.

The use of visual odometry for underwater navigation goes back to 2003 when Gracias proposed an approach for vision-based navigation of underwater vehicles that relies on the use of large-scale video mosaics of the sea bottom as environmental representations for navigation [16]. Their work relies solely upon the vision to provide information for all the relevant degrees of freedom as heading, pitch, and yaw information. Authors have used Harris corner detection method to extract point features and the matching was conducted using KTL tracker. Authors have proved accurate navigation on large areas using previously acquired mosaics for large periods, without the use of any additional sensory information. Eustice, Pizarro, and Singh in 2008, developed a visual navigation method called, visually augmented navigation (VAN) to improve the precision of near-seafloor navigation. This is a multi-sensory based approach that combines the benefits of optical and inertial navigation using an extended Kalman filter. This uses a camera as an accurate and inexpensive perceptual sensor to collect near-seafloor images and to match directly the overlapping image pairs from a calibrated camera [23]. In 2012 the same authors proposed an algorithm to overcomes some of the specific challenges in feature-poor regions with underwater visual SLAM [24]. Kim and Eustice in 2009 use the same VAN method to ship hull inspection. They mount a calibrated monocular camera on a tilt actuator so that the camera approximately maintains a nadir view to the hull. A combination of scale-invariant feature transform (SIFT) and Harris features detectors are used within a pairwise image registration [25]. Inspired by the VAN several stereo camera navigations were developed [26,27,28] which can obtain a metric scale of the map.

More recently, mono camera based navigation , a work related to the work reported in this research, was presented by Maxima *et al.* [16]. This monocular navigation techniques developed using Harris detector and KTL tracker found to be robust to the short occasion and high turbidity water. Authors also evaluate different feature tracking methods for different turbidity levels. This system runs in real time up to the frame rate of 30Hz. However this system is a visual odometry and loop closer and failure recovery systems need to be implemented. Another similar work presented in [29] includes loop closer with SIFT features. This work also an extended version of VAN navigation with mono vision but runs only low frame rate (1-2Hz). Work presented in [30] and [31] are more suitable for high frame rate (10-20Hz). Both of these are stereo systems that used stereo mapped point clouds to generate camera poses. Bundle adjustment or any of the optimization methods have not been used in this study. Another Stereo SLAM is presented as Stereo Graph-SLAM [32] by Pep Lluis *el al.* works 10 Hz with pose graph optimization.

3. The Visual SLAM Framework

The proposed system consists of three threads and works on a key frame-based method with an optimization routine to accommodate nonliner error. A hybrid method is suggested for tracking by incorporating the strength of both descriptor and non-descriptor based feature points. The front end of the system consists of image acquisition, image enhancement, feature point detection, and tracking and camera pose estimation. Image enhancement is performed by the haze removal method proposed by Kaiming et al. [33], followed by a contrast stretching. Haze removing is based on Dark Channel Prior and can perform a real-time enhancement. This enhancement makes the proposed system robust to different turbidity levels. After enhancement, Harris corner detector is applied to detect feature points. Points were selected in the strongest order concerning an evenly distributed manner. 2000 points were selected in the detection algorithm. As the Harris corner is applied, that respective frame is made as a keyframe. Creation of keyframe discussed in a separate paragraph in the paper. After the acquisition of feature points, successive frames were tracked using the Kanade-Lucas-Tomasi (KLT) method in a pyramidal implementation. The tracking algorithm also equipped with a backtrack feature that calculates the forward and backward optical flow and keeps the points tracked accurately and forwards them to the next tracking cycle. A dynamic tracking window is used in this study to avoid tracking of fast-moving features such as aquatic lifeforms and suspended debris that comes into the field of view. This technique makes the navigation algorithm robust to short occlusions. The algorithm assumes a perspective pin-hole projection system. The mapping from the 3-D space to a 2-D image is given with respect to the first person's perspective is expressed in the projection equation (1).

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KX_i = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1)$$

Where $X_i = [x \quad y \quad z]^T$ be a scene point in the camera reference frame and $P = [u \quad v]^T$ its projection on the image plane measured in pixels, λ is the depth factor, α_u and α_v the focal lengths, and u_0, v_0 the image coordinates of the projection center. Visual odometry can be expressed in technical terms as follows: At each frame j, the state of the system is estimated through the pose of the camera as given in the equation (2).

$$\xi_j = \begin{bmatrix} p_j & q_j \end{bmatrix}^T \quad (2)$$

Where $p_j \in \mathbb{R}^3$ is the position of the camera in the 3D world coordinate frame and $q_j \in SO(3)$ is the orientation of the camera. Furthermore, for each newly added key frame k, we want to estimate new landmarks $\lambda_i \in \mathbb{R}^3$ are estimated and then a subset of keyframes pose with the respective observed landmarks is optimized. This set is denoted by equation (3)

$$\Phi = \left\{ \xi_k, \xi_{k-1}, \ldots, \xi_{k-n}, \lambda_i, \ldots, \lambda_{i-m} \right\}. \quad (3)$$

Equation no (1) stated above can be rewritten with respect to third-person perspective as follows. where T_j the projective matrix computed from the state ξ_j . $T_j \in SE(3), R_j \in SO(3), t_j \in \mathbb{R}^3$

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KX_i T_j = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} R_j & t_j \\ 0_{1x3} & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \end{bmatrix} . \quad (4)$$

3.1. Key Frame Creation

Keyframe creation is initiated with the triggering one of the three criteria. 1. number of active tracking points 2. the parallax, 3. the number of frames passes. First criteria trigger when the half of the defined number of points lost in the tracking process, Second criteria triggers when the parallax of the tracked point is larger than 35 pixels, and the final criteria trigger when the number of frames exceeds 45 between the current frame and the last known keyframe. The algorithm also calculates the parallax between each successive frame and if the parallax is less than a certain threshold, those frames were ignored. Parallex is computed after unrotating the images. These techniques avoid unnecessary computation of very slow motion and hovering situations of the UAV. In a new keyframe, the algorithm detects new feature points and the system continues.

3.2. Camera Pose Estimation

Algorithm is developed to track 2D feature points on successive frames and triangulate them to create 3D landmarks. Triangulation of 3D landmarks only takes place in keyframes. Landmarks and their correspondences of 2d feature points were used to estimate the camera poses. Algorithm follows the implementation from [16,34,35]. Outline of the system represented in Figure 1.

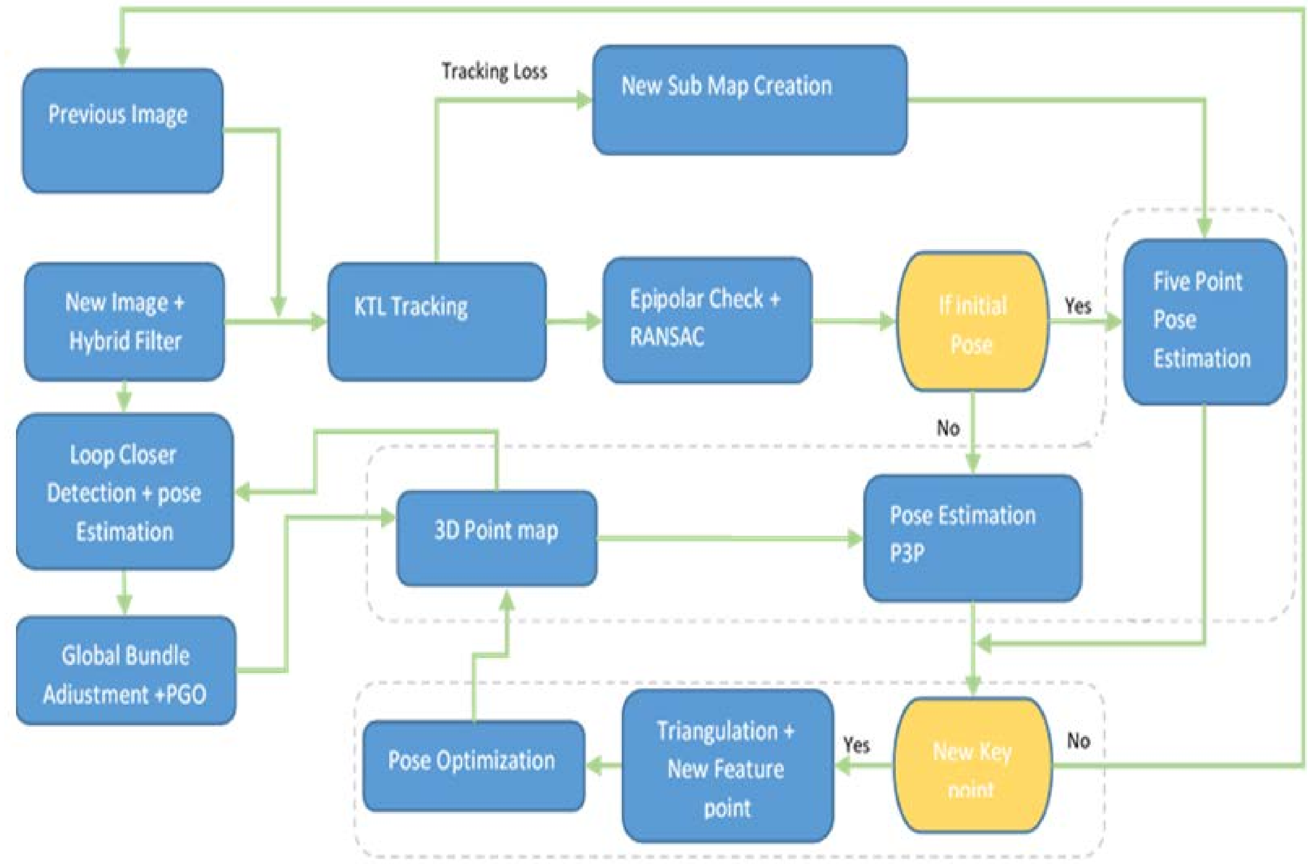

Figure 1. Outline of the proposed system

3.3. Initialization

In the beginning, the system creates the first keyframe with the very first image frame that can acquire a certain number of corner points using the Harris detector. Then the location and orientation of the first keyframe were defined and stored. This location is taken as the origin of the creation of the feature map. Then the successive frames were tracked by the KLT and when the keyframe creation triggers, the system creates the second keyframe attach to the current frame. Then the 2D-2D correspondences were extracted. Feature point correspondences for both keyframes were undergone a thorough outlier removal process by checking the epipolar consistency in a RANSAC scheme [36]. Then the Essential matrix (E) was estimated using the five-point algorithm expressed by Nister [37]. The essential matrix was used to estimate the relative pose of the second keyframe. The geometric relations between two images I_k and I_{k-1} of a calibrated camera are described by the essential matrix Ej. Essential matrix contains the camera motion parameters up to an unknown scale factor for the translation in the following form, where t_j represent a skew symmetric matrix of $t_j \in \mathbb{R}^3$.

$$E_j = t_j R_j. \tag{5}$$

It is noted that the pose of the second keyframe was estimated up to an unknown scale. Finding this unknown scale need to be done with a fusion of additional sensors such as DVL, inertial measurement unit(IMU), and Pressure sensors, and will be addressed in future works.

3.4. Pose from 2D-3D Correspondence

Once the two initial keyframes were established, 2D correspondence of the feature points were triangulated to generate 3D points. Those 3D points with 2D-3D correspondence were then used to estimate the next camera pose in the upcoming frames. The pose is estimated with the Perspective-from-3-Points (P3P) formula, using the method expressed by Gao et al [38]. This calculation also eliminates spurious correspondences using the M-estimator sample consensus (MSAC) algorithm [39]. The pose is then further refined by minimizing the global re-projection error using nonlinear least-squares optimization, a variant of the Levenberg-Marquardt algorithm [40]. It is worth noting that this algorithm is not designed to estimate the camera pose of each and every frame, but only in keyframes. Further, the estimating camera poses of each frame are not required as the UAVs are low-speed vehicles. Estimation of the poses between two keyframes can be carried out by the integration of an IMU with minimum computational resources, which will be the focus of future work. This technique saves computational resources which can be used with other tasks such as motion planning and UAV controlling. After refinement of the pose new 3D points were triangulated and used with the next iteration. Calculated camera pose with the 2D feature point and 3D structure points were then saved in a data structure for further refinement with the windowed bundle adjustment.

3.5. Re-tracking

Unlike the descriptor-based algorithm, UW-SLAM does not scan every frame but scans at keyframes. The rest of the frames are tracked using KTL. However, this causes an additional problem as the loss of a point will become permanent unless it scans in the next keyframe. To void this problem a re-tracking mechanism is added as per the method proposed by Maxime [16]. The system keeps a track of loss points in the last few consecutive frames and adds those points to the tracking of the next frames.

3.6. New Point Detection at A Key-Point

At every new keyframe, the strongest corner points are detected in a distributed manner across the image. Corner points that are identical or very closer to the previous successive tracked, are identified and rejected using scattered data interpolation [41]. Then the remaining new points with the previous successive tracked points were added to the tracking cycle in a distributed manner up to a certain threshold. This takes place on every keyframe creation and makes the successive tracked points remained in the tracking window until they out of the field of view.

3.7. Windowed Bundle Adjustment

After calculating the new camera pose, a windowed bundle adjustment is performed in a parallel thread. The size of the window kept as five keyframes to seed up the process. The first two poses of the window kept as freeze poses to minimize the scale drift of the mono system. Bundle adjustment runs in two stages as pose only bundle adjustment and full bundle adjustment. For full bundle adjustment, only the last two keyframes were used. Refined 3D structure points were filtered according to their re-projection errors and the map was updated accordingly.

4. Loop Closer

Loop closure detection is an crucial task for any SLAM problem. Monocular SLAM loop closer can divide in three broad categories:

I. map-to-map,
II. image- to-image and
III. image-to-map.

A comparison of the above loop closing systems can be found in [42,43,44]. In this research, loop closer was designed with an image to image mapping techniques using the Bag of Feature(BoF) method [45]. Surf features were used to create the visual vocabulary from images which randomly selected from the data set. Vocabulary creation reduces the number of features through the quantization of feature space using K-means clustering. At each keyframe, the keyframe vectors were stored in an inverted index data store. At the same time algorithm looks for similar vectors inside the data store. If a match is found and the matching score is above a certain threshold, the system triggers a loop closer and poses within the loop are recalculated and optimized.

In the classical visual SLAM systems, the loop closing detection is performed between two key frames [34]. In underwater scenarios, feature matching between pairs of keyframes may be insufficient due to low-quality features and a poor number of feature points [46]. The loop closing algorithm used in this research matches features between two groups of keyframes, inspired by the method proposed by Lluis Negre et al. [46]. However, we still use BoF in the loop closing algorithm. In practice, three consecutive keyframes are kept as a group. Feature points in a group undergo density-based clustering method DBSCAN and the algorithm removes feature points that are not included in any of the clusters [47]. This provides a larger area with filtered keyframes of very distingue scenes of the seabed to be recognized when revisited. In practice, this makes more efficient loop closure compared with keyframe to keyframe loop closing.

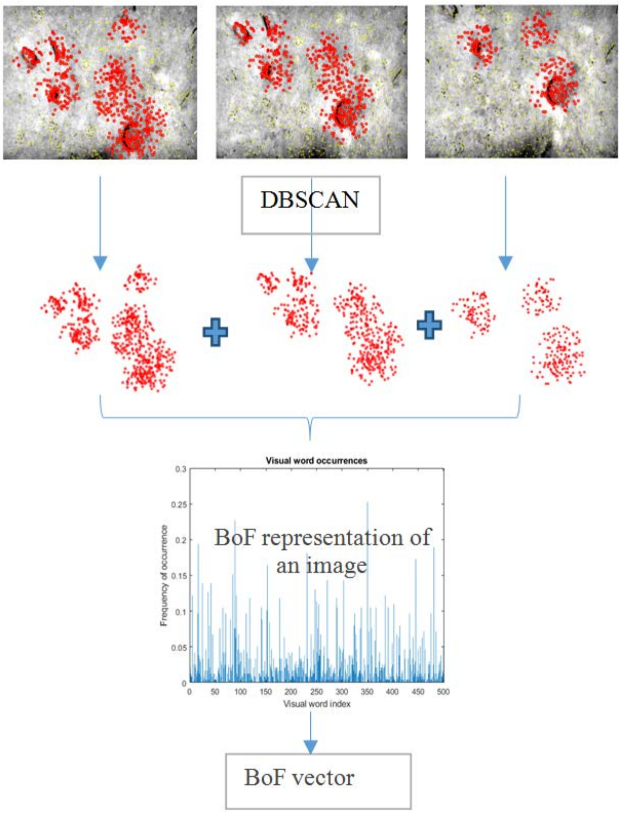

Figure 2. Outline of the Loop Closer

5. Pose Graph Optimization

When a Loop closer detected, camera poses and the 3D point correspondences to those particular poses were recalculated. Accumulated drift by the system is distributed with the detected loop using g2o optimization [48]. Then the algorithm runs a global bundle adjustment within the detected loop to further refine the poses re-estimate the 3D points. This pose optimization runs on a separate thread without affecting the main functions.

5.1. Sub Map and Failure Recovery System

It is obvious that mono SLAM tracking can fail due to many reasons, such as moving to a non-feature terrain, short occlusions, debris and livestock interference between the camera and the scene, etc. In order to re-initialized tracking, concept of sub mapping is introduced. This concept is named as the Hierarchical SLAM usually with large scale maps [49]. The submap techniques developed in this research initiates a sub map staring from the last known locations when the system unable to track the point in the previous submap. Sub maps generated by a mono SLAM system are of different scales and thus scaling is required to align with other submaps. Sub map scaling is an extended version of the loop closer where a single loop close position is not adequate to retrieve the difference in scale. One interesting work on submap scaling and loop closer was presented by Williams et al. [50]. Further, comparison of the different algorithms was presented in [42].

In this research, two types of scenarios were investigated as short term and long term solutions for tracking failures. Short term solution is proposed when the scene is being interfered with the livestock or suspended debris and loss of track is for a short time period. When the interference cleared, part of the last keyframe observations is still visible. In those scenarios, the algorithm tries to match the correspondence of the current feature points with the last known keyframe images. If enough correspondence found, pose and the scale is calculated by the P3P algorithm. If the tracking failure in a long term, such as moving to a non-feature area, the algorithm starts a new submap.

When a submap initiation triggers, the algorithm moves to a constant velocity motion model to predict the last pose of the current submap with the initiation of a keyframe. The algorithm continues to observe feature points from the upcoming images and ones enough points are found, it initiates the first pose of the new submap with the motion model predictions. Then the rest of the poses of the new submap are estimated, as usual using five-point algorithm and P3P algorithm. Sub map scaling was performed as an image to map comparison with the extended version suggested by Reid *et al.* and Williams *et al.* [42,50]. However, according to the Williams *et al.* scale was observed after finding two-loop closers within the same set of submap. Moreover in the proposed method scale is retrieved after making the first loop closer. Loop closer detection function which runs on a separate thread is responsible to find the already visited places across the different sub-maps. When a loop closer detected as described in the loop closer section, 2D to 2D feature points are matched using SURF features, and the corresponding 3D to 3D points are found accordingly. It is noted that these 3D correspondences are on a different scale and direct matching is not possible. 2D points already matched are then thoroughly checked for epipolar constrain under the RANSAC scheme. Outliers are removed and 2D to 3D correspondences were generated between the current frame and loop detected pose. The new pose is calculated with P3P algorithm. Then the relative distances between matched 3D points are used to calculate the relative scale between two submaps. It is worth noting that, theoretically ratio of any relative distance between matching 3D points of the two submaps should be equal. However, in practice due to non-linear error, relative scale consists of many values and the median value is obtained as the correct scale factor. The estimated scale factor is used to scale the poses and the 3D space coordinate points in the pose graph optimization thread. Theoretically, if the algorithm finds *n* correspondences, $\frac{n}{2}(n-1)$ sale factors can be derived and median gives reasonably accurate results. To demonstrate the submap creation, the data set was synthetically modified with blank frame to create three submaps. Figure 3a shows submap with a different scale. The scale between first and second sub maps is clearly visible and second and third is not visible. That is due to the reason that second and third submaps are on the same scale as the way the data set is made. (i.e. constant velocity movement on camera). Figure 3b shows the corrected pose after the scaling of the submaps with the loop closer.

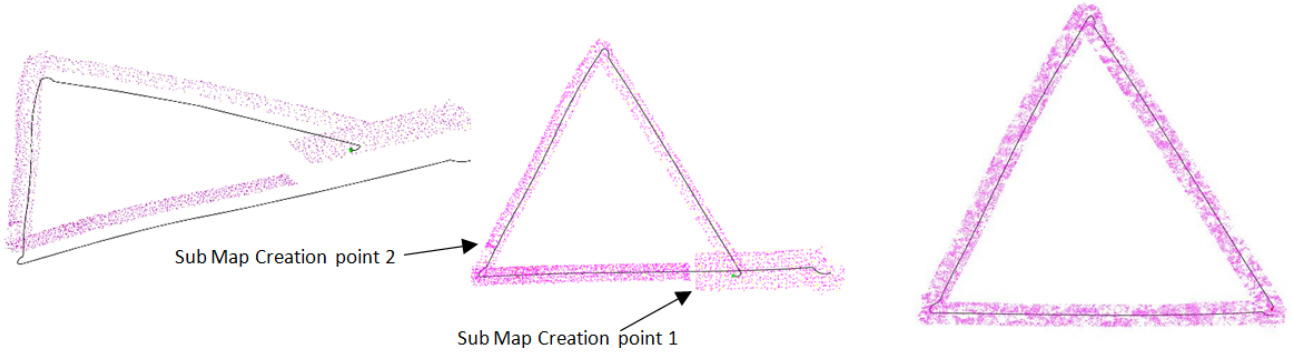

Figure 3a: Tracking failure and sub map creation

Figure 3b: Scale corrected map after loop close

Figure 3.

Table 1. Drift of The Last Pose Compare To First Pose (* means frequent failures)

Sequence	Turbidity Level	UW-SLAM-without Loop Closer	UW-SLAM-with Loop Closer	OpenVSLAM	OpenVSLAM with Loop Closer
1	**None**	**1.44**	**0.32**	**5.91**	**0.46**
2	**Low**	**1.46**	**0.38**	**5.99**	**0.36**
3	**Medium**	**1.54**	**0.4**	**6.04**	**0.46**
4	**High**	**1.64**	**0.45**	***6.02**	***0.55**

6. Experiment Results

6.1. Experiment Results in a Simulated Underwater Dataset

An open collection of simulated datasets produced by Duarte *et al*. using an underwater simulator was used to test the proposed development [51]. Datasets contain few shape of trajectories with four levels of turbidity. To test the developed system a triangular trajectory is selected with different levels of noises. The resolution of these sequences is 320 × 240 pixels. In each sequence, trajectory is formed twice and it starts and ends at the same place. These four sequences have been used to evaluate the robustness of the proposed underwater SLAM algorithm against different turbidity levels. However, as the initial stage, the lowest turbidly level is used to demonstrate the functionality and accuracy of the proposed underwater SLAM algorithm. Trajectories with and without loop closer for turbidity level 3 and 4 is presented in Figure 4. Latest visual SLAM method OpenVSLAM is used for comparison [34].

6.2. Experiment Results in a Real Underwater Dataset

Newly developed UW-SLAM is compared with the dataset presented by Maxime *et al* [16]. The latest visual SLAM methods OpenVSLAM and DSO are used in this experiment for comparison [34,52]. OpenVSLAM uses ORB features and works similarly to the ORB SLAM [34]. DSO is a direct SLAM method. Datasets presented by Maxime consist of five sequences with different turbidity levels and different short occlusion levels (short occlusion due to livestock interference).

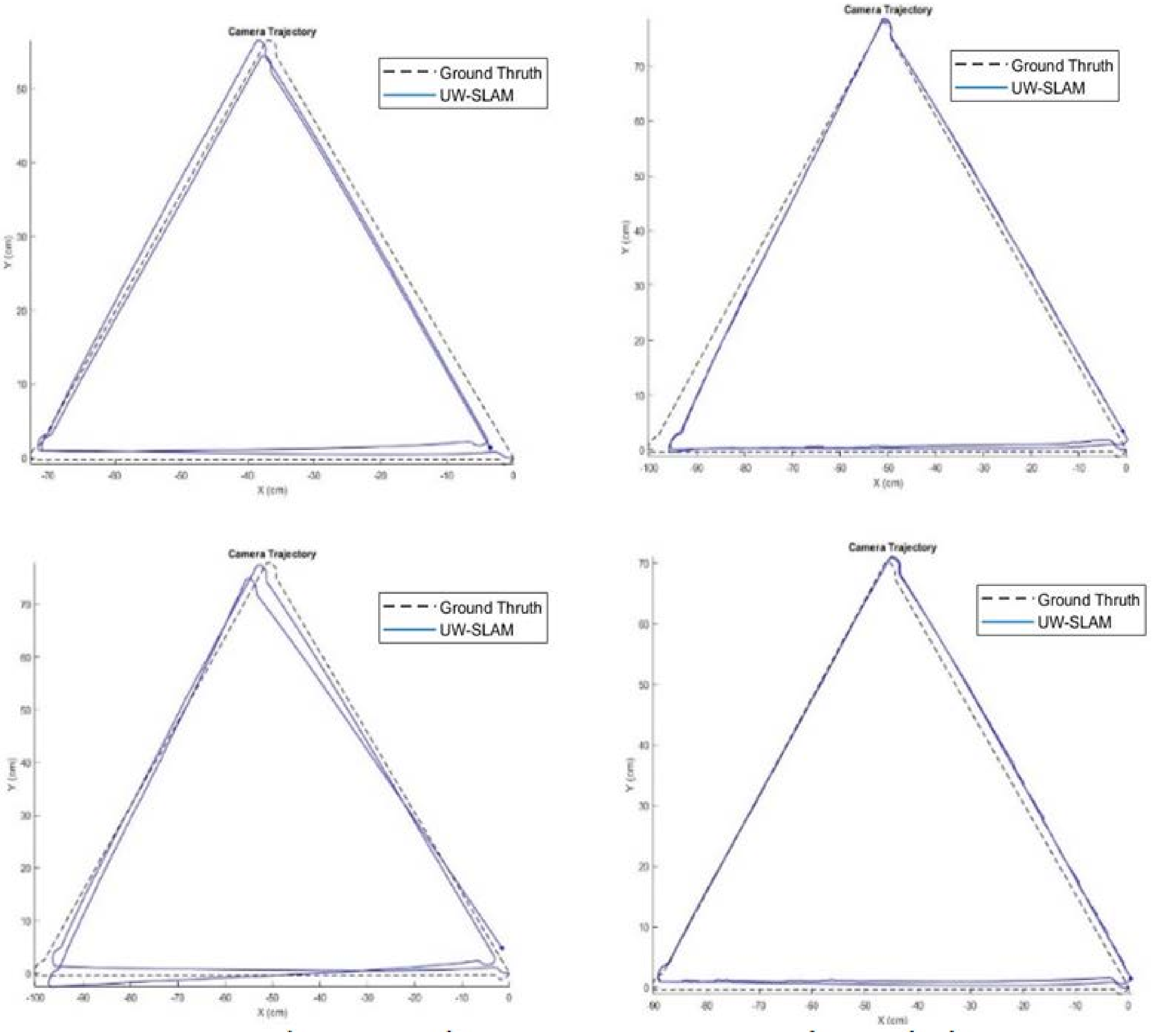

Figure 4a. Trajectory with UW-SLAM, row 1 and 2 Turbidity sequence 3 and 4, column 1 without loop closer, column 2 with loop closer

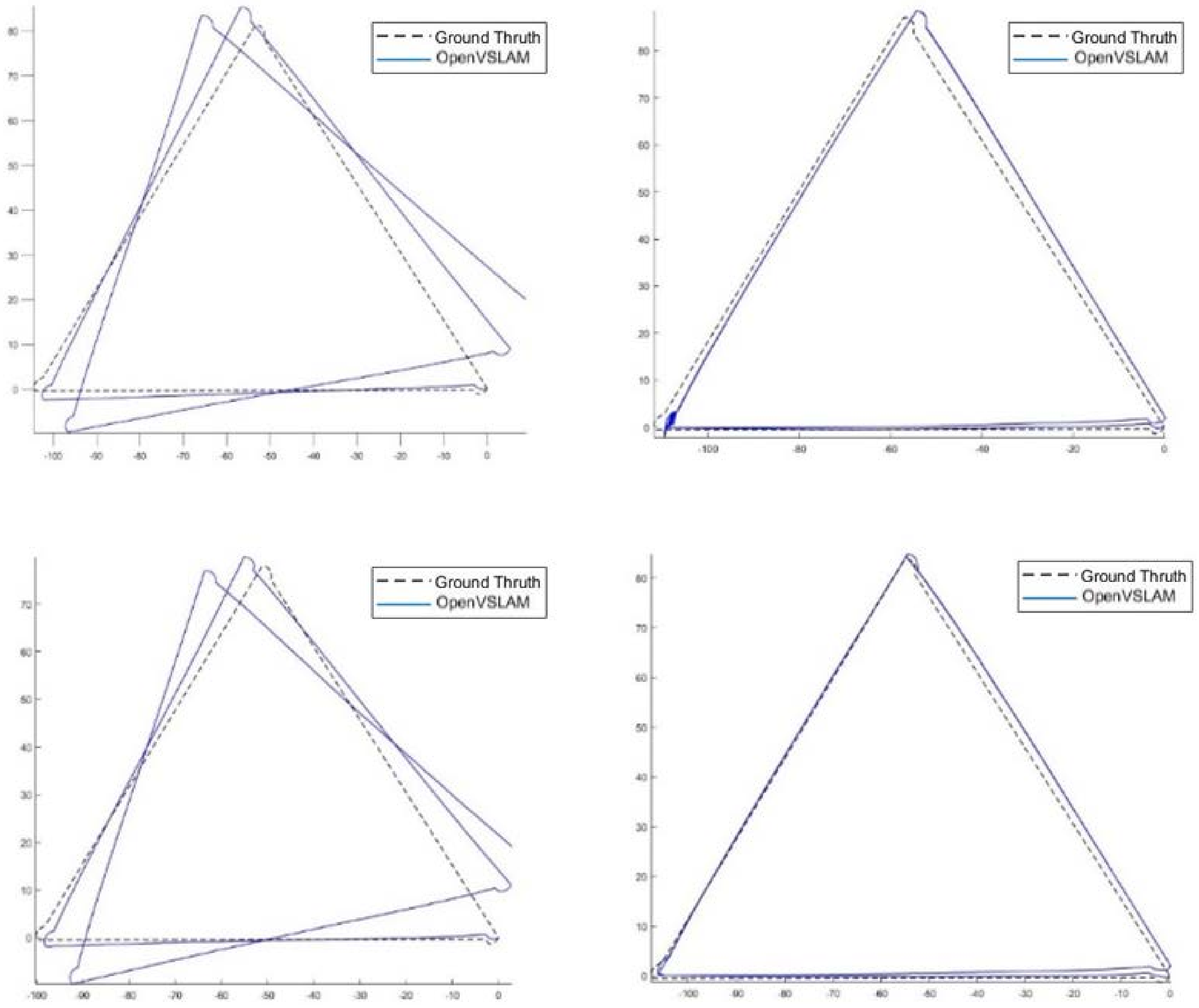

Figure 4b. Trajectory with OpenVSLAM, row 1 and 2 Turbidity sequence 3 and 4, column 1 without loop closer, column 2 with loop closer

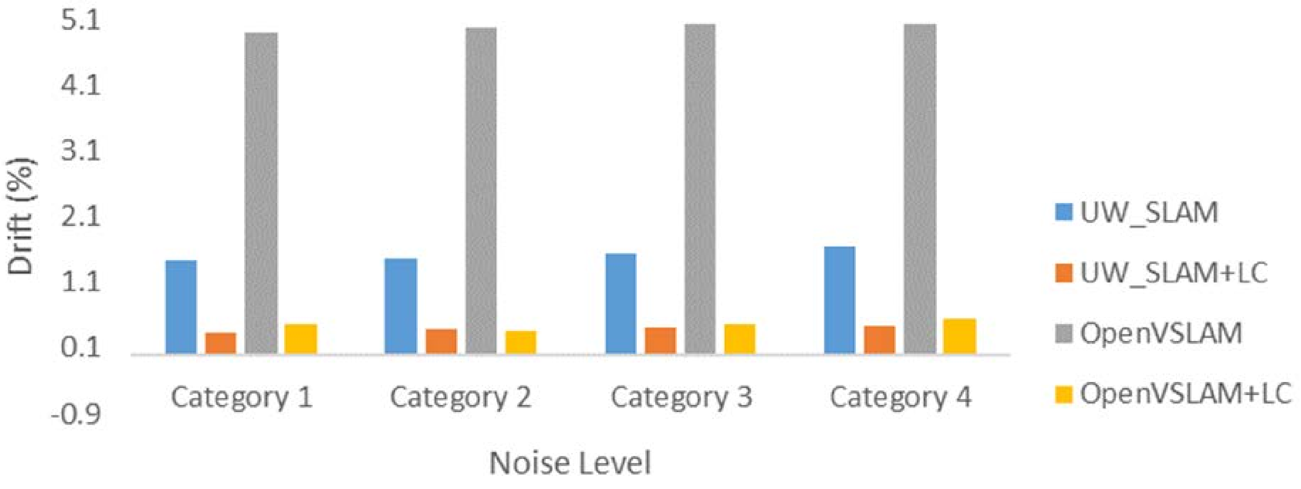

Figure 5. Drift of the last pose compare to first pose

Authors of the dataset, already tested the sequence with ORB SLAM, LSD SLAM [53], and SVO [16], and stated that ORB SLAM fails on sequence no 3, LSD SLAM fails on all sequence and SVO on sequences 4 and 5. No loop closers recorded in any of the trials [16]. In this research, the video sequences were executed with openVSLAM and DSO in comparison with UW-SLAM. The results were tabulated in Table 2. Results were averaged over five runs. UW_SLAM ran with and without a loop closer option and the number of loop closers were recorded. UW-SLAM+LC means UW-SLAM ran with a loop closer enabled.

OpenVSLAM fails to run sequence no 3 and 5 and DSO on all the sequences. UW-SLAM ran all the sequences and recorded successive loop closed for sequence no 1, 2, and 4. Sequence no. 3 is very short run and no potential loop closers can be found. Sequence no. 5 does have one or two potential loop closers but the algorithm fails to identify such cases. On the other hand, OpenVSLAM is not been able to detect any loop closer and either the Orb SLAM and SVO. In contrast, this dataset was tested with several visual SLAM and visual odometry proposed by the community such as ORB SLAM, LSD SLAM, SVO, OpenVSLAM, DSO and none of them were successful in completing all five sequence and none of them were able to make a loop closer. Only the Proposed UW-SLAM and the visual odometry proposed by Maxime *et al*. were able to complete all five sequences [16]. Moreover, the proposed method was able to loop close in three sequences. Trajectories are shown in Figure 6.

Table 2. Trajectory Evaluation With Different SLAM Methods

				Absolute Trajectory Error RMSE (%)				
Sequence no	Duration	Turbidity level	Disturbance	OpenVSLAM	DSO	UW-SLAM	UW-SLAM+LC	# of Loop Closers
1	4'	Low	Few	1	X	1.37	1.28	2
2	2' 22"	Medium	Some	1.34	X	1.24	1.12	2
3	22"	High	Many	X	X	1.10	1.10	0
4	4' 30"	Low	Many	1.37	X	1.21	1.08	2
5	3' 15"	Medium	Many	X	X	1.81	1.81	0

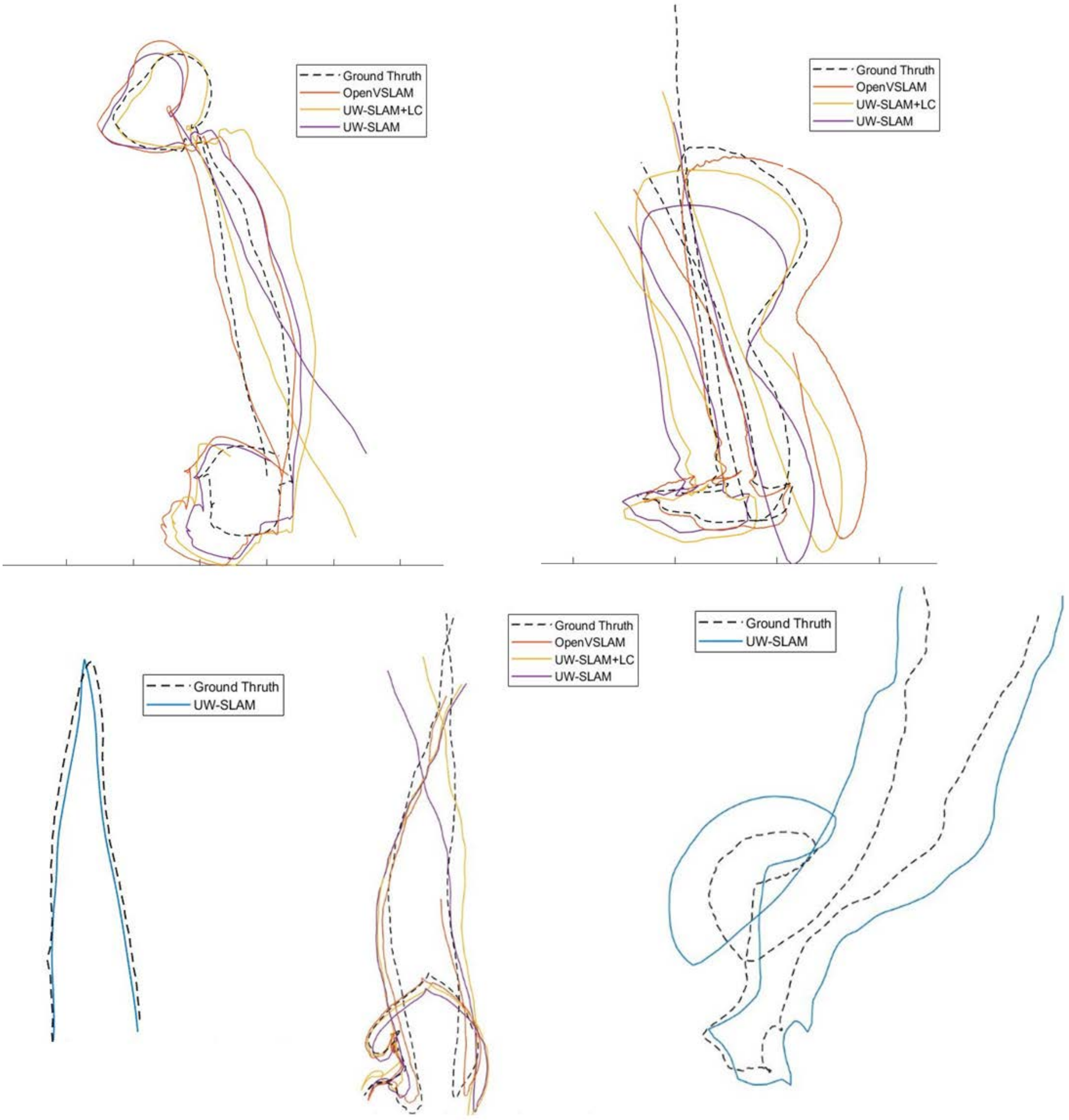

Figure 6. Trajectories of UW-SLAM, UW-SLAM+LC, OpenVSLAM

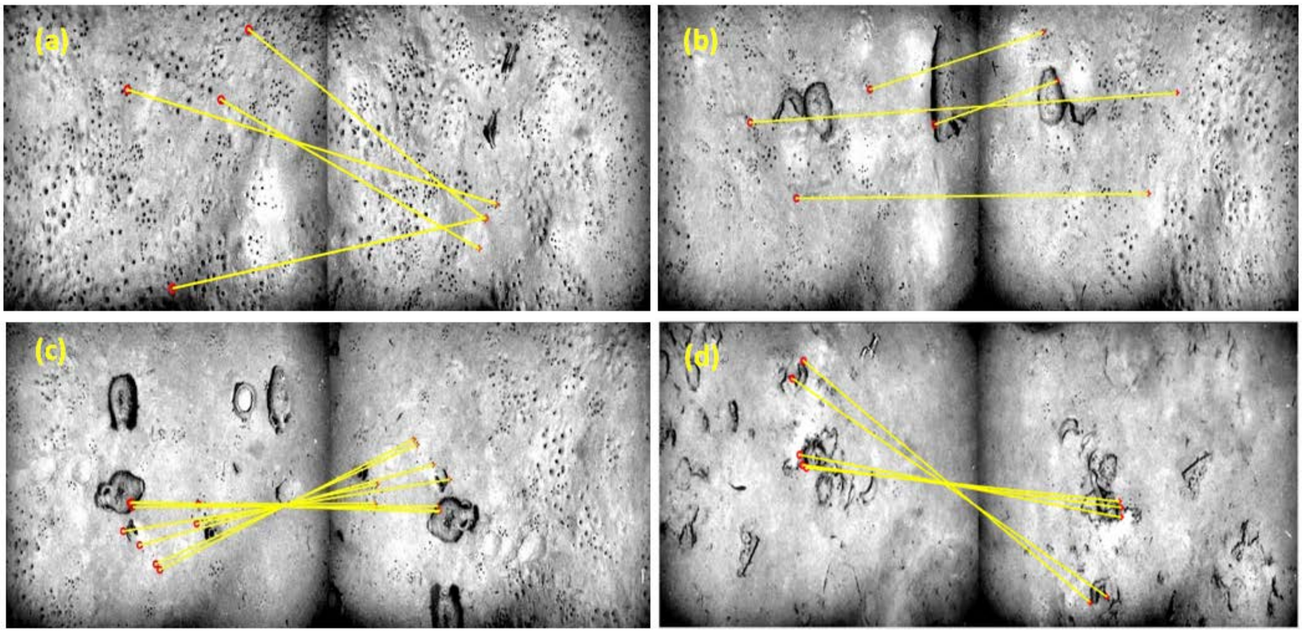

Figure 7. Trajectories of UW-SLAM, UW-SLAM+LC, OpenVSLAM

6.3. Result of Loop Closer

The proposed loop close method in the UW-SLAM is able to close the loop and estimate the pose successively in a few times in sequence no 1,2 and 4. However, these values seem to be low as there are more potential possibilities for loop closers. However, loop closing threads were able to detect many possibilities for loop closing which consists of both false positives and true positives. Figure 7(a) and 7(b) shows a true negative and a false positive match. True positives were filtered out by checking epipoler constraint in a RANSAC scheme and the poses were estimated using a P3P algorithm in the M-estimator scheme [38]. High constraints were set to estimate the pose and epipoler constraint as wrong pose will shift the trajectory unwantedly. It is often experienced that even a true positive match may fail to estimate a correct pose in the P3P algorithm. Also noted that for a successive loop closer, very distinguished features should be observed such as artificial objects or rocks. Figure 7 (c) and (d) illustrate two of the loop closer matches proposed by the algorithm. Even though the loop closer algorithm outperforms the existing bag of feature loop closer method, more improvements are needed. Those are being currently investigated.

7. Conclusion

This research is dedicated to development of a keyframe-based monocular SLAM for dynamic underwater environment. Descriptor and non-descriptor based feature points were studied for tracking and loop closing respectively. Proposed algorithm consists of following special functionalities that are required in underwater vision navigation and not included in popular vison based SLAM such as Orb SLAM.

- Image Preprocessing for high turbidity image enhancement.
- Tracking conducted by Harris detector and KTL tracker which robust to turbidity.
- Re-tracking and adaptive track window selection for the robustness of short occlusions.
- Cluster-based multi keyframe BoF loop closer system.
- Large scale operation with sub maps with failure recovery.

The new algorithm was tested using two different datasets and the obtained result shows proposed system outperformed the existing underwater monocular SLAM method. The proposed method improved the accuracy of the trajectory by 5-10%. The developed system runs in real time on average 5-8Hz on a commercial laptop of intel i7 8gb of ram tracking 2000 points on 640 x 480 images. Developmet of the algorithm in Mathlab environment is available on https://github.com/chintha/UW-SLAM

References

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1. pp. 131-149, 2014.

[2] L. Chen, S. Wang, K. Mcdonald-maier, and H. Hu, "Towards autonomous localization and mapping of AUVs: a survey," vol. 1, no. 2, pp. 97-120, 2013.

[3] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual Navigation for Mobile Robots: A Survey," *J. Intell. Robot. Syst.*, vol. 53, no. 3, pp. 263-296, 2008.

[4] D. Loebis, R. Sutton, and J. Chudley, "Review of multisensor data fusion techniques and their application to autonomous underwater vehicle navigation," *J. Mar. Eng. Technol.*, vol. 1, no. 1, pp. 3-14, 2002.

[5] J. C. Kinsey, R. Eustice, and L. L. Whitcomb, "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges," *7th Conf. Manoeuvring Control Mar. Cr.*, pp. 1-12, 2006.

[6] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges," *{IFAC} Conf. Manoeuvering Control Mar. Cr.*, 2006.

[7] J. Yuh, "Design and Control of Autonomous Underwater Robots: A Survey," vol. 24, pp. 7-8, 2000.

[8] H. Lu, Y. Li, Y. Zhang, M. Chen, S. Serikawa, and H. Kim, “Underwater Optical Image Processing: A Comprehensive Review,” 2017.

[9] S. Corchs and R. Schettini, “Underwater image processing: State of the art of restoration and image enhancement methods,” *EURASIP J. Adv. Signal Process.*, vol. 2010, 2010.

[10] J. Banerjee, R. Ray, S. R. K. Vadali, S. N. Shome, and S. Nandy, “Real-time underwater image enhancement: An improved approach for imaging with AUV-150,” *Sadhana - Acad. Proc. Eng. Sci.*, vol. 41, no. 2, pp. 225-238, 2016.

[11] R. Giubilato, M. Pertile, and S. Debei, “A comparison of monocular and stereo visual FastSLAM implementations,” *3rd IEEE Int. Work. Metrol. Aerospace, Metroaerosp. 2016 - Proc.*, pp. 227-232, 2016.

[12] J. Cunha, E. Pedrosa, C. Cruz, A. J. Neves, and N. Lau, “Using a depth camera for indoor robot localization and navigation,” *DETI/IEETA-University of Aveiro*, 2011.

[13] D. Maier, A. Hornung, and M. Bennewitz, “Real-time navigation in 3D environments based on depth camera data,” in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, 2012, pp. 692-697.

[14] S. T. Digumarti, R. Siegwart, A. Thomas, and P. Beardsley, “Underwater 3D Capture using a Low-Cost Commercial Depth Camera,” vol. 1.

[15] A. Dancu, M. Fourgeaud, Z. Franjcic, R. Avetisyan, and Q. Ab, “Underwater reconstruction using depth sensors,” pp. 1-4, 2014.

[16] J. M. Maxime Ferrera, “Real-Time Monocular Visual Odometry for Turbid and Dynamic Underwater Environments,” pp. 1-19, 2019.

[17] J. Bouguet, “Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm,” vol. 1, no. 2, pp. 1-9.

[18] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” pp. 23.1-23.6, 2013.

[19] R. Garcia and N. Gracias, “Detection of interest points in turbid underwater images,” *Ocean. 2011 IEEE - Spain*, 2011.

[20] F. Codevilla, J. D. O. Gaya, N. D. Filho, and S. S. C. C. Botelho, “Achieving Turbidity Robustness on Underwater Images Local Feature Detection,” pp. 154.1-154.13, 2015.

[21] G. Younes, D. Asmar, E. Shammas, and J. Zelek, “Keyframe-based monocular SLAM: design, survey, and future directions,” *Rob. Auton. Syst.*, vol. 98, pp. 67-88, 2017.

[22] H. Strasdat, J. M. M. Montiel, and A. J. Davison, “Visual SLAM: Why Filter?”

[23] 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, 2008.

[24] F. S. Hover *et al.*, “Advanced perception, navigation and planning for autonomous in-water ship hull inspection,” *Int. J. Rob. Res.*, vol. 31, no. 12, pp. 1445-1464, 2012.

[25] A. Kim and R. Eustice, “Pose-graph Visual SLAM with Geometric Model Selection for Autonomous Underwater Ship Hull Inspection.”

[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, 2008.

[27] A. Jalón-Monzón, C. G. R. De León, M. Alvarez-Múgica, S. Méndez-Ramírez, M. Á. Hevia-Suárez, and S. Escaf-Barmadah, “Utilidad de la biopsia frÍa ureteral durante la cistectomÍa radical como predictor de riesgo de recidiva: RevisiÓn de nuestra serie,” *Arch. Esp. Urol.*, vol. 71, no. 5, pp. 486-494, 2018.

[28] M. Pfingsthorn, R. Rathnam, T. Luczynski, and A. Birk, “Full 3D navigation correction using low frequency visual tracking with a stereo camera,” in *OCEANS 2016 - Shanghai*, 2016, pp. 1-6.

[29] 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, 2013.

[30] P. Drap, D. Merad, B. Hijazi, and L. Gaoua, “Underwater Photogrammetry and Object Modeling: A Case Study of Xlendi Underwater Photogrammetry and Object Modeling: A Case Study of Xlendi Wreck in Malta,” no. December, 2015.

[31] F. Bellavia, M. Fanfani, and C. Colombo, “Selective Visual Odometry for Accurate AUV Localization,” pp. 1-12, 2014.

[32] P. Lluis, N. Carrasco, F. Bonin-font, and G. O. Codina, “Stereo Graph-SLAM for Autonomous Underwater Vehicles⋆,” vol. 07122.

[33] H. Kaiming, S. Jian, and T. Xiaoou, “Single image haze removal using dark channel prior. Single image haze removal using dark channel prior.,” *Cvpr*, vol. 33, no. 12, pp. 2341-2353, 2009.

[34] J. M. M. Montiel, “ORB-SLAM: a Versatile and Accurate Monocular SLAM System,” pp. 1-17.

[35] Georg Klein and David W., “Parallel Tracking and Mapping for Small AR Workspaces,” *2007 6th IEEE ACM Int. Symp. Mix. Augment. Real.*, 2007.

[36] M. A. Fischler and R. C. Bolles, “Paradigm for Model,” vol. 24, no. 6, 1981.

[37] D. Nister, “An efficient solution to the five-point relative pose problem,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756-770, 2004.

[38] X.-R. H. Xiao-Shan Gao, “Complete Solution Classification for the Perspectieive-Three- Point-Problem.”

[39] P. H. S. Torr, “MLESAC: A New Robust Estimator with Application to Estimating Image Geometry,” vol. 156, pp. 138-156, 2000.

[40] M. I. A. Lourakis and A. A. Argyros, “SBA: A Software Package for Generic Sparse Bundle Adjustment,” vol. 36, no. 1, 2009.

[41] I. Amidror, “Scattered data interpolation methods for electronic imaging systems: a survey,” *J. Electron. Imaging*, vol. 11, no. 2, p. 157, 2002.

[42] I. D. Reid, B. Williams, M. Cummins, J. D. Tardós, J. Neira, and P. Newman, “A comparison of loop closing techniques in monocular SLAM,” *Rob. Auton. Syst.*, vol. 57, no. 12, pp. 1188-1197, 2009.

[43] “Review on Loop Closure Detection of Visual Slam,” no. 6, pp. 81-86, 2018.

[44] K. Granström and T. B. Schön, “Learning to close the loop from 3D point clouds,” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 2089-2095, 2010.

[45] Dorian, Galvez-Lopez, D.Juan, and Tardos, “Bags of Binary Words for Fast Place Recognition in Image Sequences,” *IEEE Trans. Robot. VOL. , NO. , Mon. YEAR. SHORT Pap.*, vol. 6, no. 3, 2012.

[46] P. L. Negre, F. Bonin-Font, and G. Oliver, “Cluster-based loop closing detection for underwater slam in feature-poor regions,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, pp. 2589-2595, 2016.

[47] M. Ester, H. Kriegel, X. Xu, and D.- Miinchen, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” 1996.

[48] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 3607-3613, 2011.

[49] C. Estrada, J. Neira, and J. D. Tardos, “Hierarchical SLAM: real-time accurate mapping of large environments,” *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 588-596, 2005.

[50] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, “An image-to-map loop closing method for monocular SLAM,” *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 2053-2059, 2008.

[51] A. C. Duarte, G. B. Zaffari, S. Rosa, L. M. Longaray, P. Drews-jr, and S. S. C. Botelho, “Towards Comparison of Underwater SLAM Methods: An Open Dataset Collection,” no. October, 2016.

[52] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” 2016.

[53] J. Engel, T. Sch, and D. Cremers, “Direct Monocular SLAM,” pp. 1-16.