Real-Time Monocular Visual Odometry for Turbid and Dynamic Underwater Environments

Maxime Ferrera, Julien Moras, Pauline Trouvé-Peloux, Vincent Creuze

sensors

MDPI

Article

Real-Time Monocular Visual Odometry for Turbid and Dynamic Underwater Environments

Maxime Ferrera [1,2,*], Julien Moras [1], Pauline Trouvé-Peloux [1] and Vincent Creuze [2]

[1] DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France; julien.moras@onera.fr (J.M.); pauline.trouve@onera.fr (P.T.)
[2] LIRMM, University of Montpellier, CNRS, 34080 Montpellier, France; vincent.creuze@lirmm.fr (V.C.)
* Correspondence: maxime.ferrera@onera.fr

Received: 26 December 2018; Accepted: 6 February 2019; Published: 8 February 2019

Abstract: In the context of underwater robotics, the visual degradation induced by the medium properties make difficult the exclusive use of cameras for localization purpose. Hence, many underwater localization methods are based on expensive navigation sensors associated with acoustic positioning. On the other hand, pure visual localization methods have shown great potential in underwater localization but the challenging conditions, such as the presence of turbidity and dynamism, remain complex to tackle. In this paper, we propose a new visual odometry method designed to be robust to these visual perturbations. The proposed algorithm has been assessed on both simulated and real underwater datasets and outperforms state-of-the-art terrestrial visual SLAM methods under many of the most challenging conditions. The main application of this work is the localization of Remotely Operated Vehicles used for underwater archaeological missions, but the developed system can be used in any other applications as long as visual information is available.

Keywords: underwater robotics; underwater visual localization; monocular visual odometry; SLAM

1. Introduction

Accurate localization is critical for most robotic underwater operations, especially when navigating in areas with obstacles such as rocks, shipwrecks or Oil & Gas structures. In underwater archaeology, Remotely Operated Vehicles (ROVs) are used to explore and survey sites in deep waters. Even if these robots are remotely operated by a pilot, systems providing their real time localization are valuable to efficiently use them. For example, this information can be used to ensure the completeness of photogrammetric surveys or as a feedback for autonomous navigation.

As radio signals are absorbed by sea water, it is not possible to use GPS systems to localize underwater vehicles. Acoustic positioning systems, such as Ultra Short Baseline (USBL), Short Baseline or Long Baseline (LBL) can be used as GPS alternatives. However these systems are expensive and require precise calibration to get a positioning accuracy in the order of one meter. In order to obtain submetric accuracy, the use of high-end Inertial Navigation Systems (INS) and of Doppler Velocity Logs (DVL) is often necessary. Some of the existing approaches for underwater localization complement these sensors with sonars to limit the drift due to the integration of measurements errors [1]. Such setups are mandatory if one seeks to localize underwater vehicles in the middle of the water column. However, when navigating close to the seabed, visual information becomes available given that the vehicle embeds a lightning system. In this scenario, cameras have also been used as a complementary sensor to limit the drift by matching temporally spaced images [2–7]. If the aforementioned approaches have shown good results on very large trajectories, they require the use of expensive high-end navigational sensors as the cameras or the acoustic positioning systems are only used to constrain the drift. robots motion as the cameras or the acoustic positioning systems are only used to constrain the drift.

In contrast, in this work we are interested in the development of a submeter grade localization system from a minimal set of low-cost sensors for lightweight ROVs used for deep archaeological operations (Figure 1). As an ROV always embeds a camera for remote control purpose, we decided to develop a visual localization framework based solely on a monocular camera to estimate in real time the ego-motion of the robot.

Visual Odometry (VO) and Visual Simultaneous Localization And Mapping (VSLAM) have been a great topic of research over recent decades [8]. VSLAM differs from VO by maintaining a reusable global map, allowing the detection of loop closures when seeing again already mapped scenes. In underwater environment, localization from vision is more complex than in aerial and terrestrial environments and state-of-the-art open-source VO or VSLAM algorithms fail when the operating conditions become too harsh [9,10]. This is mainly due to the visual degradation caused by the medium specific properties. Indeed, the strong light absorption of the medium shortens the visual perception to a few meters and makes the presence of an artificial lightning system mandatory when operating in deep waters. Besides, the propagation of light is backscattered by floating particles, causing turbidity effects on the captured images. In the darkness of deep waters, the fauna is also a cause of visual degradation as animals are attracted by the artificial light and tend to get in the field of view of the camera, leading to dynamism and occlusions in the images. In front of these difficulties, many works tackle the underwater localization problem using sonar systems [11–13], as they do not suffer from these visual degradation. Nevertheless, the information delivered by a sonar is not as rich as optical images [14] and remains very challenging to analyze. Furthermore, at close range, acoustic systems do not provide accurate enough localization information whereas visual sensing can be highly effective [15].

Figure 1. Remotely Operated Vehicle Dumbo performing archaeological sampling on an antic shipwreck in the Mediterranean Sea (460 m deep). This archaeological operation has been performed under the supervision of the French Department of Underwater Archaeology (DRASSM), in accordance with the UNESCO Convention on the Protection of the Underwater Cultural Heritage. Credit: F. Osada, DRASSM - Images Explorations.

In this paper, we propose UW-VO (UnderWater-Visual Odometry), a new monocular VO algorithm dedicated to the underwater environment that overcomes the aforementioned visual degradation. Inspired by state-of-the-art VO and VSLAM methods from aerial and terrestrial robotics, UW-VO is a keyframe-based method. The front-end of UW-VO has been made robust to turbidity and short occlusions by employing optical flow to track features and a retracking mechanism to retrieve temporarily lost features. The back-end relies on Bundle Adjustment [16] to ensure minimal drift and consistency in the reconstructed trajectory.

The paper contributions are the following:

- A thorough evaluation of visual features tracking methods on underwater images.
- The development of UW-VO: a monocular VO method robust to turbidity and to short occlusions caused by the environment dynamism (animals, algae...).
- An evaluation of state-of-the-art open-source monocular VO and VSLAM algorithms on underwater datasets and a comparison to UW-VO, highlighting its robustness to underwater visual degradation.
- The release of a dataset consisting of video sequences taken on an underwater archaeological site (https://seafile.lirmm.fr/d/aa84057dc29a4af8ae4a/.)

The paper is then organized as follows. First, in Section 2, we review works related to our contributions. In Section 3, we present an evaluation of features tracking methods on two underwater sets of images and we show that optical flow-based tracking performs better than methods based on the matching of descriptors. In Section 4, our monocular visual localization system UW-VO is described. Finally, in Section 5, we compare UW-VO to state-of-the-art open-source monocular VO and VSLAM on both a synthetic dataset and on a real one that we made publicly available for the community. We show that UW-VO competes with these methods in terms of accuracy and surpasses them in terms of robustness.

2. Related Work

The localization of robots from the output of a camera system has been a great topic of research for several decades. Here we review the major works in the field of underwater visual localization and we compare it to the aerial and terrestrial robotics field.

2.1. Underwater Features Tracking

The problem of features tracking is one of the cornerstones of visual localization. In the underwater context, [17] evaluate the use of SURF [18] features in their SLAM algorithms and assess improvements in the localization but their method do not run online. The authors of [19] evaluate different combinations of feature detectors and descriptors and they show that SURF features and Shi-Tomasi corners [20] matched by Zero Mean Normalized Cross-Correlation (ZNCC) are good combinations for visual state estimation. None of these works included the evaluation of binary features such as BRIEF [21] or ORB [22] nor of optical-flow, which are widely used methods for features tracking in VO and VSLAM.In [23,24], many feature detectors are evaluated on underwater images with a focus on their robustness to turbidity. They follow the evaluation protocol of [25] by using the detectors repeatability as their metric. Robustness to turbidity is essential for underwater visual localization but only evaluating repeatability of the detectors does not ensure good features tracking capacity as ambiguity can still arise when trying to match these features between two images. Our features tracking evaluation differs from these previous works by directly evaluating the features tracking efficiency of a wide range of feature detectors and tracking methods in the context of VO and VSLAM.

2.2. Underwater Visual Localization

In [2], the authors present a successful use of visual information as a complementary sensor for underwater robots localization. They used an Extended Information Filter (EIF) to process dead-reckoning sensors and insert visual constraints based on the 2D-2D matching of points coming from overlapping monocular images. Here, only the relative pose between cameras is computed so the visual motion is estimated up to scale and do not provide the full 6 degrees of freedom of the robot motions. Following their work, many stereo-vision-based systems were proposed [3,4,26]. The advantage of using stereo cameras lies in the availability of the metric scale in opposition to the scale ambiguity inherent to pure monocular system. The scale factor can indeed be resolved from the known baseline between both sensors (assuming the stereo system calibrated).

The authors of [5–7] later integrated nonlinear optimization steps to further process the visual data through bundle adjustment [16]. The authors of [27] extended [2] by keeping a monocular approach but adding loop-closure capability to their methods through the computation of a visual saliency metric using SIFT features. However, in all these methods the visual information is only used to bound the localization drift using low-overlap imagery systems (1–2 HZ), but their systems mainly rely on expensive navigational sensors.

Closer to our work, some stereo VO approaches use higher frame rate videos (10-20 hz) to estimate underwater vehicles ego-motion [28–30]. In [29], features are matched within stereo pairs to compute 3D point clouds and the camera poses are estimated by aligning these successive point clouds, making it a pure stereo vision method. In parallel, the authors of [30] use a keyframe-based approach but their features tracking is done by matching descriptors both spatially (between stereo images pair) and temporally. Moreover, they do not perform bundle adjustment to optimize the estimated trajectory.

Despite the advantage of stereo-vision systems over monocular cameras, embedding a single camera is materially more practical, as classical camera housings can be used and cameras synchronization issues are avoided. Furthermore, developing a monocular VO algorithm makes it portable to any kind of underwater vehicles, as long as it is equipped with a camera. Even if there is a projective ambiguity with monocular systems, it is possible to retrieve the scale factor from any complementary sensor capable of measuring a metric quantity.

The early works of [31,32] studied the use of a monocular camera as a mean of motion estimations for underwater vehicles navigating near the seabed. In [31], low-overlap monocular images are used to estimate the robot motions but the processing is performed offline. Gracias *et al.* [32] proposed a real time mosaic-based visual localization method, estimating the robot motions through the computation of homographies with the limiting assumptions of purely planar scenes and 4 degrees of freedom motions (x, y, z, yaw). Negahdaripour *et al.* [33] extended these by computing the 6 degrees of freedom of a camera equipped with inclinometers. In [34], an offline Structure From Motion framework was proposed to compute high quality 3D mosaicing of underwater environment from monocular images.

Underwater monocular-based methods using cameras at high frame rate (10–20 hz) were studied by [35,36]. In their approaches, they fuse visual motion estimation in an Extended Kalman Filter (EKF) along with an IMU and a pressure sensor. By using an EKF, they suffer from the integration of linearization errors which are limited in our system thanks to the iterative structure of bundle adjustment. The authors of [37] make use of a camera to detect known patterns in a structured underwater environment and use it to improve the localization estimated by navigation sensors integrated into an EKF. However, such methods are limited to known and controlled environment. More recently, Creuze [38] presented a monocular underwater localization method that does not rely on an EKF framework but iteratively estimates ego-motion by integration of optical flow measurements corrected by an IMU and a pressure sensor. This latter is used to compute the scale factor of the observed scene.

2.3. Aerial and Terrestrial Visual Localization

While most of the underwater odometry or SLAM systems rely on the use of an EKF, or its alternative EIF version, in aerial-terrestrial SLAM, filtering methods have been put aside to the profit of more accurate keyframe-based approaches using bundle adjustment [39]. PTAM [40] was one of the first approach able to use bundle adjustment in real time along with [41]. The work of Strasdat *et al.* [42] and ORB-SLAM from Mur-Artal *et al.* [43] are built on PTAM and improve it by adding a loop-closure feature highly reducing the localization drift by detecting loops in the trajectories. Whereas all these methods match extracted features between successive frames, SVO [44] and LSD-SLAM [45] are two direct approaches directly tracking photometric image patches to estimate the camera motions. Following these pure visual systems, tightly coupled visual-inertial systems have been recently presented [46–48] with higher accuracy and robustness than standard visual systems.

These visual-inertial systems are all built on very accurate pure visual SLAM or VO methods, as they use low-cost Micro Electro Mechanical Systems (MEMS) IMU, highly prone to drift.

Before considering the coupling of such complementary low-cost sensors for localization, the first step is to be able to rely on an accurate VO method. Hence, contrarily to most of the approaches in underwater localization, we propose here a keyframe-based VO method, solely based on visual data coming from a high frame monocular camera. Inspired by aerial-terrestrial SLAM, we choose to rely on bundle adjustment to optimize the estimated trajectories, thus avoiding the integration of linearization errors of filtered approaches. Furthermore, our method do not use any environment specific assumption and can hence run in any kind of environment (planar or not). We show that our method outperforms state-of-the-art visual SLAM algorithms on underwater datasets.

3. Features Tracking Methods Evaluation

As discussed in the introduction, underwater images are mainly degraded by turbidity. Moreover, underwater scenes do not provide many discriminant features and often show repetitive patterns, such as coral branches, holes made by animals in the sand or simply algae or sand ripples in shallow waters. In order to develop a VO system robust to these visual degradation, we have evaluated the performance of different combinations of detectors and descriptors along with the optical flow-based *Kanade-Lucas-Tomasi* (KLT) method [49] on two sets of underwater images.

3.1. Underwater Sets of Images

Two different sets of images are used here (Figure 2). The first one is the TURBID dataset [24], which consists of series of static pictures of a printed seabed taken in a pool. Turbidity was simulated on these images by adding a controlled quantity of milk between two shots. The second one consists of a sequence of images extracted from a video sequence recorded by a moving camera close to the seabed. This sequence exhibits the typical characteristics of underwater images: low texture and repetitive patterns. As this set is a moving one, we will refer to it as the VO set. On both sets, all the images used are resized to 640×480 and gray-scaled to fit the input format of classical VO methods.

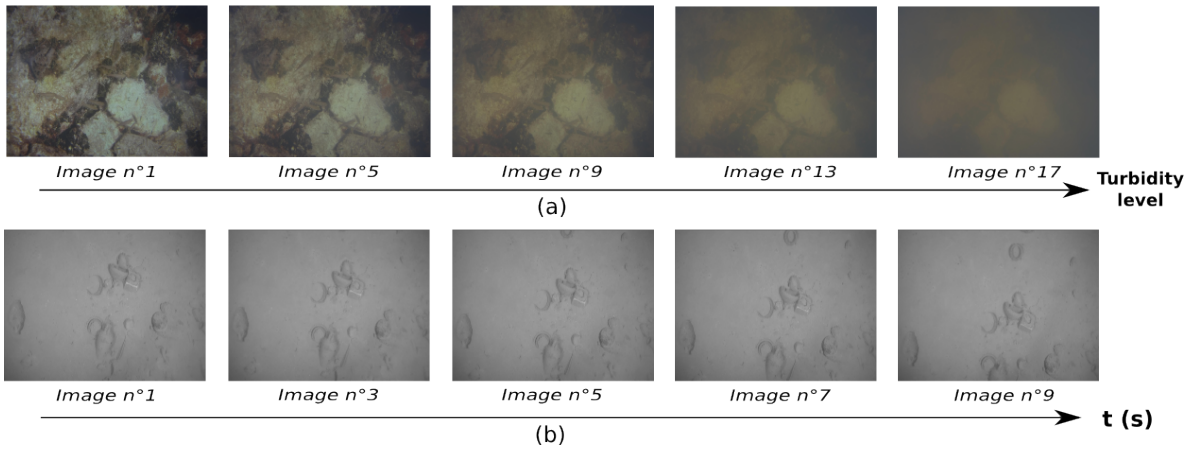

Figure 2. Images used to evaluate the tracking of features. (**a**) Images from the TURBID dataset [24]. (**b**) Images acquired on a deep antic shipwreck (depth: 500 meters, Corsica, France) - Credit: DRASSM (French Department of Underwater Archaeological Research).

3.2. Features Tracking Methods

We evaluate the performance of different features tracking methods using handcrafted features only as we seek real time capacity on CPU, which is not really possible yet with deep learning based methods. The following combination of features detector and descriptor are compared: ORB

[22], BRISK [50], FAST [51] + BRIEF [21], FAST + FREAK [52], SIFT [53] and SURF [18]. SIFT and SURF are used as a baseline here as their computational complexity make them unsuitable for real time applications on CPU. In addition to these descriptor-based methods, we also evaluate the performance of the KLT, an optical flow-based method. The KLT works by extracting Harris corners using the algorithm of Shi and Tomasi [20] and tracking these corners through optical flow with the Lucas-Kanade implementation [54]. As we will show that the KLT performs best, we also evaluate all the previous descriptors in conjunction with this Harris corner detector for a fair comparison.

3.3. Evaluation Protocol

The TURBID dataset is used to evaluate robustness to turbidity. The employed evaluation protocol on this set is the following:

- we divide each image into 500 cells and try to extract one feature per cell
- we track the features extracted in one image into the following one (i.e., the image shot right after an adding of milk)
- before each tracking, the second image is virtually translated of 10 pixels to avoid initializing the KLT at the right spot

Please note that the KLT method uses a local window, limiting the search space around the previous feature. As the tracking is here performed between images slightly translated one from the other, the search space for matching descriptors is limited to a 40x40 pixels window around the previous features. Therefore none of the tested methods is advantaged in front of another. Please note that, as translations are the predominant motion in frame-to-frame tracking in the context of VO, we did not apply rotation or scale change to the images.

The VO set is used to evaluate each method on a real VO scenario; that is we evaluate the efficiency of each method in tracking features over a sequence of images. For the methods relying on descriptors, we proceed as follows:

- we divide each image into 500 cells and try to extract one feature per cell
- we try to match the features extracted in the first image in all the following ones

For the KLT method, as it is a local method, we proceed slightly differently:

- we divide the first image into 500 cells and try to extract one feature per cell
- we try to track these features sequentially (image-to-image) by computing optical flow in a forward-backward fashion and remove features whose deviation is more than 2 pixels

For both methods we removed the outliers by checking the epipolar consistency of matched features in a RANSAC [55] scheme. On both sets, evaluation metric is the number of correctly tracked features. Please note that, depending on the feature detector, not all methods are be able to detect 500 features in the images.

3.4. Results

Figure 3 illustrates the results obtained on the TURBID set of images. Figure 3 displays (a) the number of features detected in each image for every method and (b) the number of tracked features between consecutive pictures. The resulting graphs clearly show that the KLT method is able to track the highest number of features. Indeed, more than 80% of the detected features are successfully tracked in the first fifteen images, whereas for the other methods this number is way below 50%. However, we can see that the Harris detector is the only one able to extract almost 500 features in each image (Figure 3a). We have therefore run another evaluation using only this detector. Please note that the requirements of some descriptors discard non suited detections, which is the reason of the difference in the number of detected features in Figure 3c. The results in Figure 3d show that the Harris detector

increases the performance of all the descriptors evaluated but none of them matches the performance of the KLT method.

Figure 4 illustrates the results obtained on the VO set of images. Figure 4a,b show that the KLT also tracks the highest number of features across this sequence. Around 60% of the features detected in the first image are still successfully tracked in the last image with the KLT. For the other methods, the ratio of features correctly tracked between the first and last image barely reaches 20%. Once again, using the Harris detector improves the results for most of the descriptors, increasing the tracking ratio up to about 35%, but the KLT remains the most efficient tracking method (Figure 4c,d).

In front of these results, it appears that the KLT method is more robust to the low quality of underwater images. The reason is that the low texture of the images leads to the extraction of ambiguous descriptors which cannot be matched robustly whereas the KLT, by looking for the minimization of a photometric cost, is less subject to these ambiguities.

Therefore, we choose to build our VO algorithm on the tracking of Harris corners detected with the Shi-Tomasi detector and tracked through optical flow with the method of Lucas-Kanade. Furthermore, an advantage of using the KLT over using descriptors resides in its low computation cost as it does not require the extraction of new features in each image.

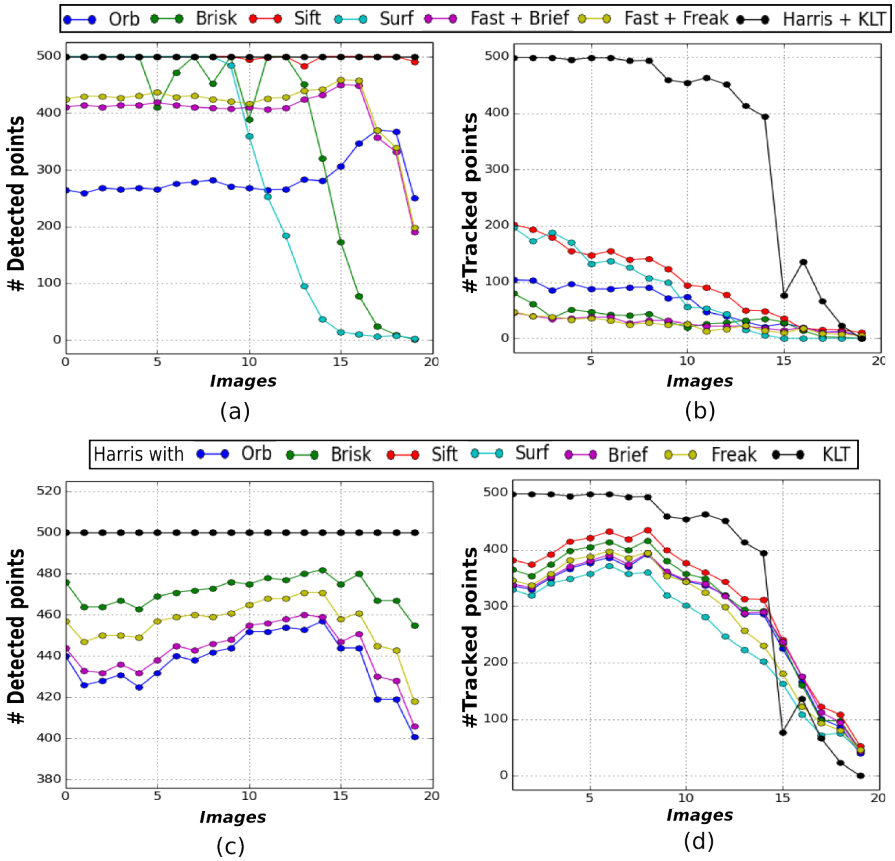

Figure 3. Features tracking methods evaluation on the TURBID dataset [24] (presented in Figure 2a). Graphs (**a**) and (**b**) illustrates number of features respectively detected and tracked with different detectors while (**c**) and (**d**) illustrates number of features respectively detected with the Harris corner detector and tracked as before (the SURF and SIFT curves coinciding with the Harris-KLT one in (c)).

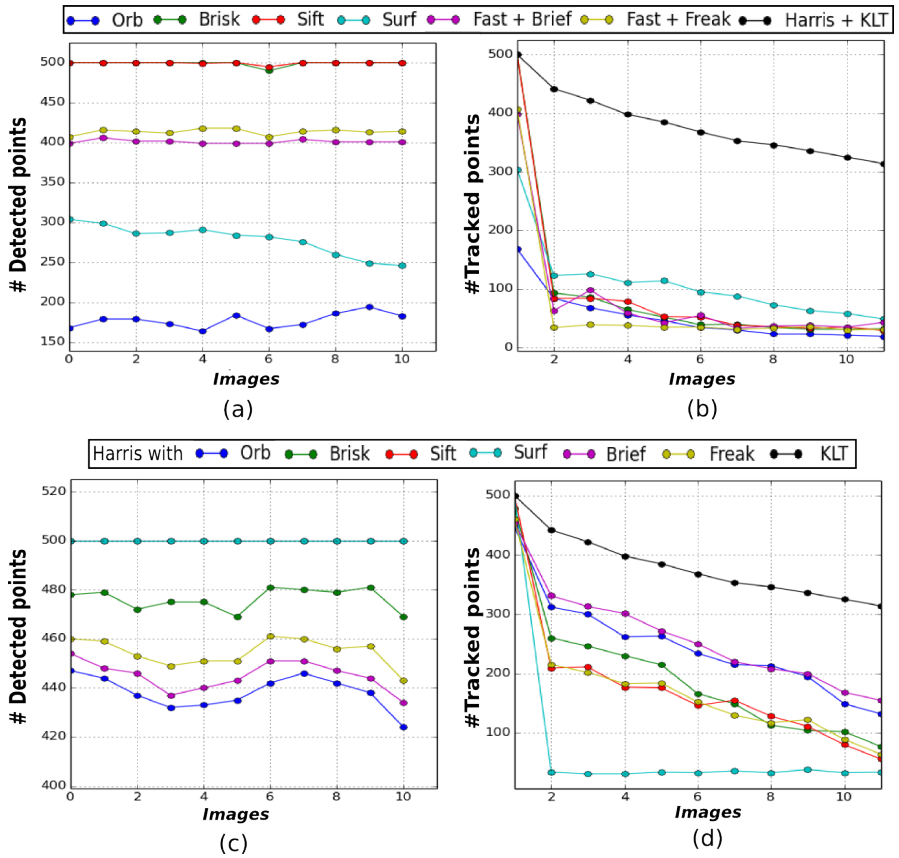

Figure 4. Evaluation of features tracking methods on a real underwater sequence (presented in Figure 2b). Graphs (**a**) and (**b**) illustrate the number of features respectively detected and tracked with different detectors, while (**c**) and (**d**) illustrate the number of features respectively detected with the Harris corner detector and tracked as before (the SIFT curve coinciding with the SURF one in (c)).

4. The Visual Odometry Framework

The pipeline of UW-VO is summarized in Figure 5. The system is based on the tracking of 2D features over successive frames in order to estimate their 3D positions in the world referential. The 2D observations of these 3D landmarks are then used to estimate the motion of the camera. Frames used for the triangulation of the 3D map points are considered as keyframes and the most recent ones are stored in order to optimize the estimated trajectory along with the structure of the 3D map through bundle adjustment. The method follows the approach of [40,42,43]. However, in opposition to these methods, we do not build the tracking on the matching of descriptors. Instead we use the KLT method, more adapted to the underwater environment as demonstrated in Section 3. The drawback of the KLT in opposition to descriptors is that it is only meant for tracking features between successive images. This is a problem when dealing with a dynamic environment as good features might be lost because of short occlusions. To make the KLT robust to such events, a retracking mechanism is added to the tracking part of the VO framework. This mechanism will be described in Section 4.2. Please note that, as UW-VO is a fully monocular system, scale is not observable and the camera's pose is hence estimated up to an unknown scale factor. The recovery of this scale factor could be done by integrating an additional sensor such as an altimeter or an IMU. This is not considered in this paper but will be the subject of future work.

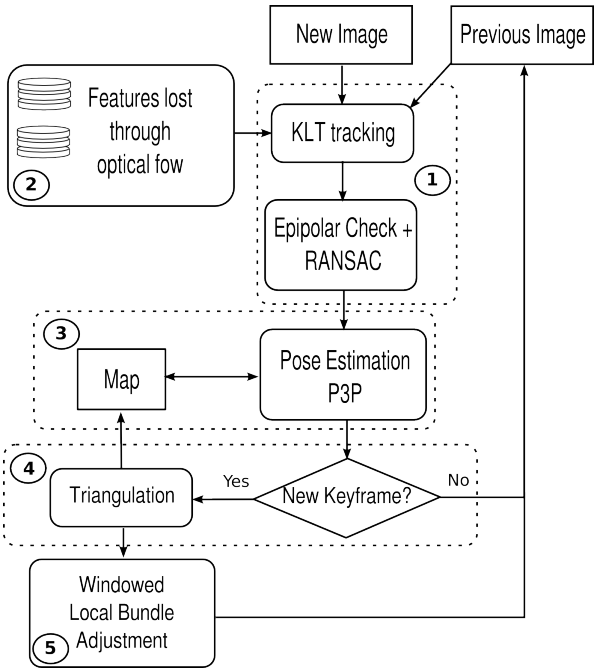

Figure 5. Pipeline of the proposed visual odometry algorithm.

More formally, the problem of VO can be defined as follows. At each frame i, we want to estimate the state of the system through the pose χ_j of the camera, defined as:

$$\chi_j = \begin{bmatrix} p_j & q_j \end{bmatrix}^T \quad , \; with \quad p_j \in \mathbb{R}^3, \; q_j \in SO(3) \tag{1}$$

where p_j is the position of the camera in the 3D world and q_j is its orientation. Furthermore, for each newly added keyframe k, we want to estimate new landmarks λ_i ($\lambda_i \in \mathbb{R}^3$) and then optimize a subset of keyframes pose with the respective observed landmarks. This set is denoted ζ:

$$\zeta = \{\chi_k, \chi_{k-1}, ..., \chi_{k-n}, \lambda_i, ..., \lambda_{i-m}\} \tag{2}$$

In the following, we assume that the monocular camera is calibrated and that distortion effects are compensated. The geometrical camera model considered in this work is the pinhole model and its mathematical expression of world points projection is:

$$\mathbf{proj}(T_j, X_i) = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K.T_j.X_i \tag{3}$$

$$\mathbf{proj}(T_j, X_i) = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} R_j & t_j \\ 0_{1\times 3} & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4}$$

with u and v the pixel coordinates, K the intrinsic calibration parameters, T_j the projective matrix computed from the state $\chi_j - T_j \in SE(3)$, $R_j \in SO(3)$, $t_j \in \mathbb{R}^3$ − and $X_i \in \mathbb{P}^3$ the homogeneous representation of the landmark λ_i. The coefficients f_x,f_y and c_x,c_y represent respectively the focal lengths along the x and y axes and the position of the optical center, expressed in pixels.

4.1. Frame-to-Frame Features Tracking

Features are extracted on every new keyframe using the Shi-Tomasi method to compute Harris corners. The motion of the 2D features is then estimated using the KLT. After each optical flow estimation, we thoroughly remove outliers from the tracked features: first, we compute the disparity between the forward and backward optical flow estimations and remove features whose disparity is higher than a certain threshold. Then, from the intrinsic calibration parameters, we compute the essential matrix using the 5-points method of Nister [56] between the previous keyframe and the current frame. This essential matrix is computed within a RANSAC process in order to remove the features not consistent with the estimated epipolar geometry.

Once enough motion is detected, the tracked 2D features are triangulated from their observations in the current frame and in the previous keyframe. The current frame used here is converted into a keyframe and new 2D corners are extracted in order to reach a specified maximum number of features. All these 2D features are then tracked in the same way as described above.

4.2. Features Retracking

The main drawback of optical flow-based tracking is that lost features are usually permanently lost. In opposition, the use of descriptors allows the matching of features with strong view-point change. In the underwater context, the powerful lights embedded by ROVs often attract bench of fishes in the camera field of view. The occlusions due to fishes can lead to strong photometric shifts and consequently to a quick loss of features. However, fishes are moving very fast and their position changes very quickly between successive frames. We take advantage of this fact to increase the robustness of our tracking method over short occlusions. The employed strategy is too keep a small window of the most recent frames (five frames is enough in practice) with the list of features lost through optical flow in it. At each tracking iteration, we try to retrack the lost features contained in the retracking window. Finally, retracked features are added to the set of currently tracked features.

This features tracking implementation is used to track both pure 2D features, for future triangulation, and 2D observations of already mapped points, for pose estimations.

4.3. Pose Estimation

The estimation of the 6 degrees of freedom of the pose of every frame uses their respective 2D-3D correspondences. The pose is computed with the Perpective-from-3-Points (P3P) formula, using the method of Kneip *et al.* [57]. This operation is done within a RANSAC loop to remove inaccurate correspondences. The pose is computed from the combination of points giving the most likely estimation for the set of features. The pose is then further refined by minimization of the global reprojection error using the set I of inliers:

$$\arg\min_{T_j} \sum_{i \in I} (x_i - proj(T_j, X_i))^2 \tag{5}$$

with x_i the 2D observation of the world point X_i and $proj(T_j, X_i)$ the reprojection of X_i in the frame j with its related projection matrix T_j.

This minimization is done through a nonlinear least-squares optimization using the Levenberg-Marquardt algorithm. The computed poses are then used to estimate the 3D positions of the tracked features.

4.4. Keyframe Selection and Mapping

The mapping process is triggered by the need for a new keyframe. Several criteria have been set as requirements for the creation of a keyframe. The first criterion is the parallax. If an important parallax from the last keyframe has been measured (30 pixels in practice), a new keyframe is inserted as it will allow the computation of accurate 3D points. The parallax is estimated by computing the

median disparity of every tracked pure 2D features from the previous keyframe. To ensure that we do not try to estimate 3D points from false parallax due to rotational motions, we unrotate the currently tracked features before computing the median disparity. The second criterion is based on the number of 2D-3D correspondences. We verify that we are tracking enough 2D observations of map points and trigger the creation of a keyframe if this number drops below a threshold defined as less than 50% of the number of observations in the last keyframe.

For further optimization, a window of the most recent keyframes along with their 2D-3D correspondences is stored. This optimization operation known as bundle adjustment is performed in parallel after the creation of every keyframe and is described next. Finally, new Harris corners are detected and the tracking loop is run again.

4.5. Windowed Local Bundle Adjustment

As stated above, a window of the most recent N keyframes is stored and optimized with bundle adjustment at the creation of new keyframes. To ensure a reasonable computational cost, only a limited number of the most recent keyframes are optimized along with their tracked map points. The remaining keyframes are fixed in order to constrain this nonlinear least-squares optimization problem. The size of the window is set adaptively by including every keyframe sharing a map point observation with one of the optimized keyframes. This adaptive configuration sets high constraints on the problem and helps in reducing the unavoidable scale drift inherent to monocular odometry systems. The Levenberg-Marquardt algorithm is used to perform this optimization. The problem is solved by minimizing the map points reprojection errors. As least-squares estimators do not make any difference between high and low error terms, the result would be highly influenced by the presence of outliers with high residuals. To prevent this, we use the robust M-Estimator Huber cost function [58] in order to reduce the impact of the highest error terms on the found solution.

We define the reprojection errors e_{ij} for every map point i observed in a keyframe j as:

$$e_{ij} = x_{ij} - proj(T_j, X_i) \tag{6}$$

We then define the set of parameters ζ^* to optimize as:

$$\zeta^* = \left\{ \chi_{j-2}, \chi_{j-1}, \chi_j, \lambda_i, ..., \lambda_{i+M} \right\} \tag{7}$$

with M the number of landmarks observed by the three most recent keyframes. In addition, we minimize (6) over the optimization window of N keyframes:

$$\arg\min_{\zeta^*} \sum_{j \in N} \sum_{l \in L_j} \rho(e_{ij}^T \Sigma_{ij}^{-1} e_{ij}) \tag{8}$$

with L_j the set of landmarks observed by the keyframe j, ρ the Huber robust cost function and Σ_{ij} the covariance matrix associated with the measures x_{ij}.

After convergence of the Levenberg-Marquardt algorithm, we remove the map points with a resulting reprojection error higher than a threshold. This optimization step ensures that after the insertion of every keyframe both the trajectory and the 3D structure of the map are statistically optimal.

4.6. Initialization

Monocular systems are subject to a "Chicken and Egg" problem at the beginning. Indeed, the motion of the camera is estimated through the observations of known 3D world points, but the depth of the imaged world points is not observable from a single image. The depth of these world points can be estimated using two images with a sufficient baseline. However, this baseline needs to be known to compute the depth and vice-versa. This is why monocular VO requires an initialization step to bootstrap the algorithm in opposition with stereo systems. In UW-VO, initialization is done

here by computing the relative pose between two frames through the estimation of an essential matrix with the 5-points methods of [56]. The norm of the estimated translation vector is then arbitrarily fixed to one, as scale is not observable with monocular setups. We assessed that this simple method is able to initialize accurately the VO framework in any configuration (planar or not), making it non-environment dependent.

5. Experimental Results

Implementation: The proposed system has been developed in C++ and uses the ROS middleware [59]. The tracking of features is done with the OpenCV implementation of the Kanade-Lucas algorithm [49]. Epipolar geometry and P3P pose estimations are computed using the OpenGV library [60]. Bundle Adjustment is performed using the graph optimization framework g2o [61] and runs in a parallel thread.

The average run time is of 25 ms per frame with the tracking limited to 250 features per frame and bundle adjustment is performed on the five most recent keyframes. The run time goes up to 35 ms when a new keyframe is required because of the features detection and triangulation overload. Thus our system can run in real time for video sequences with a frame rate up to 30 Hz. The experiments have been carried with an Intel Core i5-5200 CPU-2.20GHz-8 Gb RAM.

To the best of our knowledge there is no underwater method able to estimate localization from monocular images available open-source. Furthermore, no publicly available datasets were released with these methods, so we cannot compare with them. Hence, UW-VO has been evaluated along with ORB-SLAM (https://github.com/raulmur/ORB_SLAM2), LSD-SLAM (https://github.com/tum-vision/lsd_slam) and SVO (http://rpg.ifi.uzh.ch/svo2.html) on different datasets which are all available online, allowing future methods to compare to our results.

All algorithms are evaluated on real underwater datasets. UW-VO and ORB-SLAM are also evaluated on a simulated dataset, whose frame rate (10 Hz) is too low for SVO and LSD-SLAM to work. Indeed, SVO and LSD-SLAM are direct methods which require very high overlap between two successive images in order to work. Please note that ORB-SLAM and SVO have been fine-tuned in order to work properly. For ORB-SLAM, the features detection threshold was set at the lowest possible value and the number of points was set to 2000. For SVO, the features detection threshold was also set at the lowest possible value and the number of tracked features required for initialization was lowered to 50. For each method, every results presented are the averaged results over five runs.

5.1. Results on a Simulated Underwater Dataset

A simulated dataset created from real underwater pictures has been made available to the community by Duarte *et al.* [62]. Four monocular videos of a triangle-shaped trajectory are provided with four different levels of noise in order to synthetically degrade the images with turbidity-like noise (Figure 6). The images resolution of these videos is 320 $\times$ 240 pixels. In each sequence, the triangle-shaped trajectory is performed twice and it starts and ends at the same place. These four sequences have been used to evaluate the robustness against turbidity of UW-VO with respect to ORB-SLAM. For fair comparison, ORB-SLAM has been run with and without its loop-closing feature. We will refer this version of ORB-SLAM as V.O. ORB-SLAM in the following.

Table 1 presents the final drift at the end of the trajectory for each method. On the first three sequences, ORB-SLAM is able to close the loops and therefore has the lowest drift values, as the detection of the loop closures allows to reduce the drift accumulated in-between. On the same sequences, V.O. ORB-SLAM has the highest level of drifts. Please note that ORB-SLAM and its V.O. alternative fail half the time on the third level of noise sequence and have been run many times before getting five good trajectories. It is worth noting that the localization drift increases significantly for V.O. ORB-SLAM when the turbidity level gets higher. This is mostly due to the increased inaccuracy in its tracking of ORB features. On the last sequence, the turbidity level is such that ORB descriptors get too ambiguous and leads to failure in ORB-SLAM tracking. These results highlight the deficiency

of ORB-SLAM tracking method on turbid images. In comparison, UW-VO is able to run on all the sequences, including the ones with the highest levels of noise (Figure 7). The computed trajectories are more accurate than V.O. ORB-SLAM and we can note that it is barely affected by the noise level (Figure 8). These results confirm the efficiency of UW-VO as a robust odometry system in turbid environments.

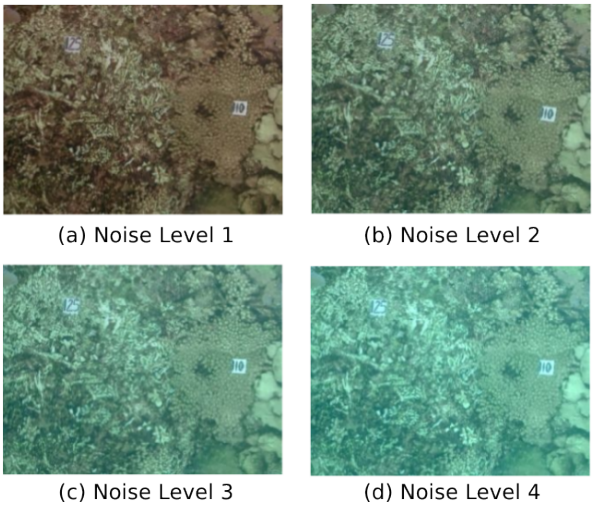

Figure 6. The four different turbidity levels of the simulated dataset.

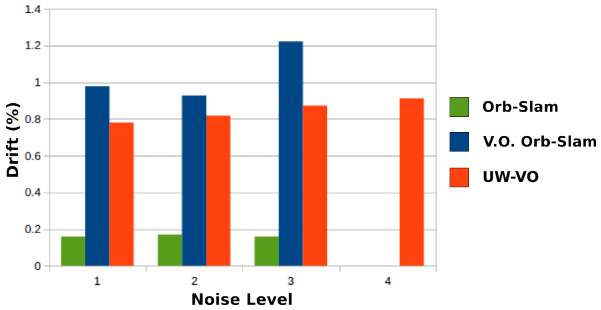

Figure 7. Drift of ORB-SLAM (green), V.O. ORB-SLAM (blue) and UW-VO (red) on the simulated underwater dataset.

Table 1. Translation drift (in %) on the simulated underwater video sequence with different level of noise simulating turbidity effects. Results are given averaging over five runs for each algorithm. V.O. ORB-SLAM designates ORB-SLAM without the loop closing feature enabled, i.e., performing only Visual Odometry. ORB-SLAM results are given for information. The (*) denotes very frequent failure of the algorithm.

		Drift (in %)		
Seq. Noise Level	Turbidity	*ORB-SLAM*	V.O. ORB-SLAM	UW-VO
1	None	*0.18*	0.97	**0.78**
2	Low	*0.18*	0.93	**0.81**
3	Medium	*0.17**	1.21*	**0.85**
4	High	*X*	X	**0.89**

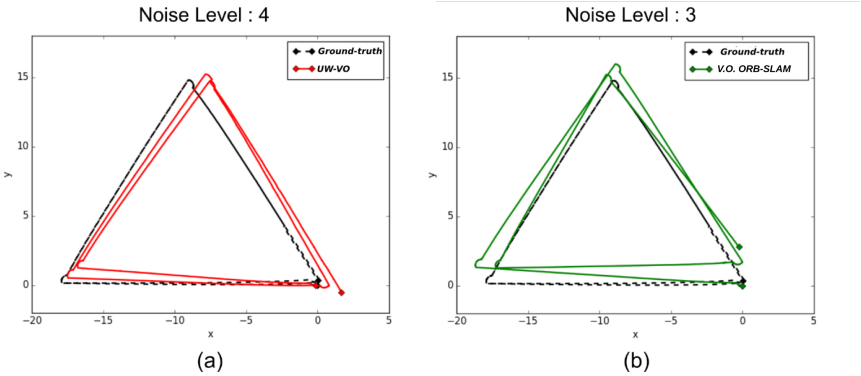

Figure 8. Trajectories estimated with (**a**) our method on the sequence with the highest level of noise and with (**b**) V.O. ORB-SLAM on the sequence with the noise level of 3.

5.2. Results on a Real Underwater Video Sequence

We now present experiments conducted on five real underwater video sequences. These sequences were gathered 500 meters deep in the Mediterranean Sea (Corsica), in 2016, during an archaeological mission conducted by the French Department of Underwater Archaeological Research (DRASSM). The videos were recorded from a camera embedded on an ROV and gray-scale 640x480 images were captured at 16 Hz. The calibration of the camera has been done with the Kalibr [63] library. Calibration was done in situ in order to estimate the intrinsic parameters and the distortion coefficients of the whole optical system. If the calibration is performed in the air, the water and camera's housing effects on the produced images would not be estimated without simulating their effects [64] and this would lead to a bad estimate of the camera's parameters. The camera recording the videos was placed inside an underwater housing equipped with a spherical dome and we obtained good results using the pinhole-radtan model (assessed by a reprojection error < 0.2 px).
These five sequences can be classified as follows:

- Sequence 1: low level of turbidity and almost no fishes.
- Sequence 2: medium level of turbidity and some fishes.
- Sequence 3: high level of turbidity and many fishes.
- Sequence 4: low level of turbidity and many fishes.
- Sequence 5: medium level of turbidity and many fishes.

For each of these sequences, a ground truth was computed using the state-of-the-art Structure-from-Motion software Colmap [65]. Colmap computes trajectories offline by exhaustively trying to match all the images of a given sequence, thus finding many loops and creating very reliable trajectories. We could assess the accuracy of the reconstructed trajectories both visually and by checking the correctness of the matched images.

Here, we compare ORB-SLAM, LSD-SLAM and SVO to UW-VO. We evaluate the results of each algorithm against the trajectories computed offline by Colmap by first aligning the estimated trajectories with a similarity transformation using the method of [66] and then computing the absolute trajectory error [67] (Figure 9). The results are displayed in Table 2. To observe the effect of the retracking mechanism (described in Section 4.2), we have run the UW-VO algorithm with and without enabling this feature, respectively referring to it as UW-VO and UW-VO* (Videos of the results for each method on the five sequences are available online (https://www.youtube.com/playlist?list=PL7F6c8YEyil-RuC7YptNMAM88gfBfn0u4)).

Table 2. Absolute trajectory errors (RMSE in %) for five underwater sequences with different visual degradation. Results are given averaging over five runs for each algorithm. UW-VO* designates our method without the retracking step, while UW-VO designates our method with the retracking step.

				Absolute Trajectory Error RMSE(in %)				
Seq. #	Duration	Turbidity Level	Short Occlusions	LSD-SLAM	ORB-SLAM	SVO	UW-VO*	UW-VO
1	4′	Low	Few	X	1.67	**1.63**	1.78	1.76
2	2′30″	Medium	Some	X	1.91	2.45	1.78	**1.73**
3	22″	High	Many	X	X	1.57	1.10	**1.04**
4	4′30″	Low	Many	X	**1.13**	X	1.61	1.58
5	3′15″	Medium	Many	X	1.94	X	2.08	**1.88**

As we can see, LSD-SLAM fails on all the sequences. This is most likely due to its semi-dense approach based on the tracking of edges with strong gradients, which are not frequent on sea-floor images. SVO is able to compute quite accurate trajectories on the sequences that are not too much affected by dynamism from moving fishes. The tracking of SVO, which is similar to optical flow, seems to work well even on turbid images, but its direct pose estimation method is not robust to bad tracked photometric patches like the one created by moving fishes (seq. 3,4,5). ORB-SLAM on the other hand performs well on highly dynamic sequences, but loses in accuracy when turbidity is present (seq. 2,3,5). Its pose estimation method based on the observations of independent features is hence robust to short occlusions and dynamic objects, but its tracking method fails on images degraded by turbidity. Furthermore, we can note that despite loop closures in the trajectories (see Figure 9), ORB-SLAM is not able to detect them. The failure to detect the loop closures indicates that the Bag of Words approaches [68] might not be suited to the underwater environment, which does not provide many discriminant features.

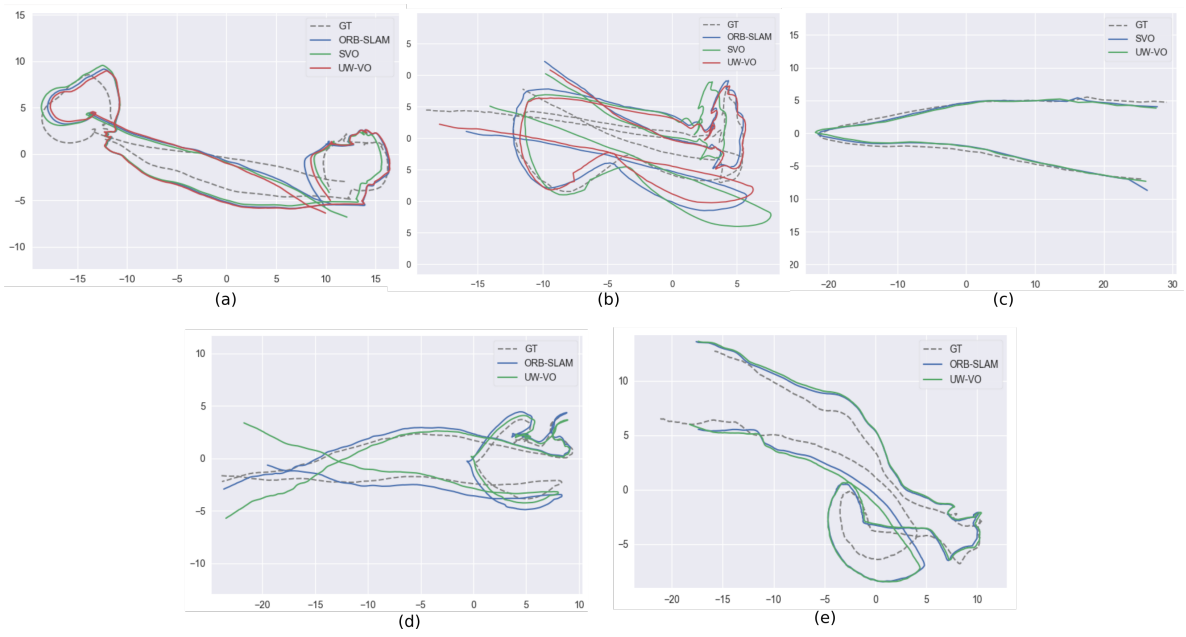

Figure 9. Trajectories of ORB-SLAM, SVO and UW-VO over the five underwater sequences. (**a**) Sequence 1, (**b**) Sequence 2, (**c**) Sequence 3, (**d**) Sequence 4, (**e**) Sequence 5. Ground-truths (GT) are extracted from Colmap trajectories.

UW-VO is the only method able to run on all the sequences. While the estimated trajectory is slightly less accurate on the easiest sequence (seq. 1), UW-VO performs better than ORB-SLAM and

SVO on the hardest sequences (seq. 2,3,5, with turbidity and dynamism, which is very common during archaeological operations). We can see the benefit of the developed retracking mechanism on most of the sequences. Nonetheless, this optical flow retracking step is not as efficient as the use of descriptors when the number of short occlusions is very large (seq. 4). Studying the effect of combining optical flow tracking with the use of descriptors could result in an interesting hybrid method for future work.

6. Conclusion

In this paper we have presented UW-VO, a new vision-based underwater localization method. While most of the existing approaches rely on expensive navigational sensors to estimate the motions of underwater vehicles, we have chosen to investigate the use of a simple monocular camera as a mean of localization. We propose a new keyframe-based monocular visual odometry method robust to the underwater environment. Different features tracking methods have been evaluated in this context and we have shown that optical flow performs better than the classical methods based on the matching of descriptors. We further enhanced this optical flow tracking by adding a retracking mechanism, making it robust to short occlusions due to the environment dynamism. We have shown that the proposed method outperforms the state-of-the-art visual SLAM algorithms ORB-SLAM, LSD-SLAM and SVO in underwater environments. We publicly released the underwater datasets used in this paper along with the camera calibration parameters and the trajectories computed with Colmap to allow future methods to compare to our results. The good results obtained on these sequences highlight the effectiveness of the developed method for localizing ROVs navigating in deep underwater archaeological sites. The computed localization could be used by the pilot as a driving assistance and could further serves as a feedback information for navigation if scale is recovered. Future work will study the implementation of this localization algorithm on an embedded computing unit in order to fulfill these tasks. The development of a monocular visual odometer was a first step towards a robust underwater localization method from low-cost sensors. One perspective is to enhance it by adding a loop-closure mechanism, turning into a visual SLAM method. We have observed that loop-closing approaches based on classical Bag of Words [68] do not work as expected results in our tests and alternative methods in the lead of [9] need to be investigated. Finally, in the same idea as visual-inertial SLAM algorithms, we will next study the tight fusion of a low-cost IMU and of a pressure sensor with this visual method to improve the localization accuracy and retrieve the scale factor.

Author Contributions: M.F. conducted this research during his doctoral work. M.F. designed and implemented the UW-VO algorithm. M.F., J.M. and P.T. analyzed the results of UW-VO. V.C. did the acquisition of the video sequences used. All the authors participated in the redaction of the article.

Acknowledgments: The authors acknowledge support of the CNRS (Mission pour l'interdisciplinarité - Instrumentation aux limites 2018 - Aqualoc project) and support of Région Occitanie (ARPE Pilotplus project). The authors are grateful to the DRASSM for its logistical support and for providing the underwater video sequences.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: A review. *IEEE J. Oceanic Eng.* **2014**, *39*, 131–149.
2. Eustice, R.M.; Pizarro, O.; Singh, H. Visually Augmented Navigation for Autonomous Underwater Vehicles. *IEEE J. Oceanic Eng.* **2008**, *33*, 103–122.
3. Johnson-Roberson, M.; Pizarro, O.; Williams, S.B.; Mahon, I. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *J. Field Rob.* **2010**, *27*, 21–51.
4. Mahon, I.; Williams, S.B.; Pizarro, O.; Johnson-Roberson, M. Efficient View-Based SLAM Using Visual Loop Closures. *IEEE Trans. Rob.* **2008**, *24*, 1002–1014.
5. Beall, C.; Lawrence, B.J.; Ila, V.; Dellaert, F. 3D reconstruction of underwater structures. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 18–22 October 2010; pp. 4418–4423.

6. Warren, M.; Corke, P.; Pizarro, O.; Williams, S.; Upcroft, B. Visual sea-floor mapping from low overlap imagery using bi-objective bundle adjustment and constrained motion. In Proceedings of the Australasian Conference on Robotics and Automation, Wellington, New Zealand, 3–5 December 2012.
7. Carrasco, P.L.N.; Bonin-Font, F.; Campos, M.M.; Codina, G.O. Stereo-Vision Graph-SLAM for Robust Navigation of the AUV SPARUS II. *IFAC-PapersOnLine* **2015**, *48*, 200 – 205.
8. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Rob.* **2016**, *32*, 1309–1332.
9. Carrasco, P.L.N.; Bonin-Font, F.; Oliver, G. Cluster-based loop closing detection for underwater slam in feature-poor regions. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2589–2595.
10. Weidner, N.; Rahman, S.; Li, A.Q.; Rekleitis, I. Underwater cave mapping using stereo vision. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5709–5715.
11. Ribas, D.; Ridao, P.; Tardos, J.D.; Neira, J. Underwater SLAM in a marina environment. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 29 October–2 November 2007; pp. 1455–1460.
12. White, C.; Hiranandani, D.; Olstad, C.S.; Buhagiar, K.; Gambin, T.; Clark, C.M. The Malta cistern mapping project: Underwater robot mapping and localization within ancient tunnel systems. *J. Field Rob.* **2010**, *27*, 399–411.
13. Yuan, X.; Martínez-Ortega, J.; Fernández, J.A.S.; Eckert, M. AEKF-SLAM: A New Algorithm for Robotic Underwater Navigation. *Sensors* **2017**, *17*, 1174.
14. Bonin-Font, F.; Oliver, G.; Wirth, S.; Massot, M.; Negre, P.L.; Beltran, J.P. Visual sensing for autonomous underwater exploration and intervention tasks. *Ocean Eng.* **2015**, *93*, 25 – 44.
15. Palomeras, N.; Vallicrosa, G.; Mallios, A.; Bosch, J.; Vidal, E.; Hurtos, N.; Carreras, M.; Ridao, P. AUV homing and docking for remote operations. *Ocean Eng.* **2018**, *154*, 106 – 120.
16. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle Adjustment — A Modern Synthesis. In *Vision Algorithms: Theory and Practice*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2000; pp. 298–372.
17. Aulinas, J.; Carreras, M.; Llado, X.; Salvi, J.; Garcia, R.; Prados, R.; Petillot, Y.R. Feature extraction for underwater visual SLAM. In Proceedings of the OCEANS 2011, Cantabria, Spain, 6–9 June 2011.
18. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In Proceedings of the European Conference on Computer Vision (ECCV), Graz, Austria, 7–13 May 2006; pp. 404–417.
19. Shkurti, F.; Rekleitis, I.; Dudek, G. Feature Tracking Evaluation for Pose Estimation in Underwater Environments. In Proceedings of the 2011 Canadian Conference on Computer and Robot Vision, St Johns, NF, Canada, 25–27 May 2011; pp. 160–167.
20. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 21-23 June 1994; pp. 593–600.
21. Calonder, M.; Lepetit, V.; Ozuysal, M.; Trzcinski, T.; Strecha, C.; Fua, P. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1281–1298.
22. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
23. Garcia, R.; Gracias, N. Detection of interest points in turbid underwater images. In Proceedings of the OCEANS 2011, Cantabria, Spain, 6–9 June 2011.
24. Codevilla, F.; Gaya, J.D.O.; Filho, N.D.; Botelho, S.S.C.C. Achieving Turbidity Robustness on Underwater Images Local Feature Detection. In Proceedings of the British Machine Vision Conference (BMVC), Swansea, UK, 7–10 September 2015.
25. Mikolajczyk, K.; Tuytelaars, T.; Schmid, C.; Zisserman, A.; Matas, J.; Schaffalitzky, F.; Kadir, T.; Gool, L.V. A Comparison of Affine Region Detectors. *Int. J. Comput. Vision* **2005**, *65*, 43–72.
26. Pfingsthorn, M.; Rathnam, R.; Luczynski, T.; Birk, A. Full 3D navigation correction using low frequency visual tracking with a stereo camera. In Proceedings of the OCEANS 2016, Shanghai, China, 10–13 April 2016.

27. Kim, A.; Eustice, R.M. Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency. *IEEE Trans. Rob.* **2013**, *29*, 719–733.
28. Corke, P.; Detweiler, C.; Dunbabin, M.; Hamilton, M.; Rus, D.; Vasilescu, I. Experiments with Underwater Robot Localization and Tracking. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 10-14 April 2007; pp. 4556–4561.
29. Drap, P.; Merad, D.; Hijazi, B.; Gaoua, L.; Nawaf, M.M.; Saccone, M.; Chemisky, B.; Seinturier, J.; Sourisseau, J.C.; Gambin, T.; Castro, F. Underwater Photogrammetry and Object Modeling: A Case Study of Xlendi Wreck in Malta. *Sensors* **2015**, *15*, 30351–30384.
30. Bellavia, F.; Fanfani, M.; Colombo, C. Selective visual odometry for accurate AUV localization. *Autom.Rob.* **2017**, *41*, 133–143.
31. Garcia, R.; Cufi, X.; Carreras, M. Estimating the motion of an underwater robot from a monocular image sequence. In Proceedings of the 2001 IEEE/RSJ Intelligent Robots and Systems (IROS), Maui, HI, USA, 29 October–3 November 2001; pp. 1682–1687.
32. Gracias, N.R.; van der Zwaan, S.; Bernardino, A.; Santos-Victor, J. Mosaic-based navigation for autonomous underwater vehicles. *IEEE J. Oceanic Eng.* **2003**, *28*, 609–624.
33. Negahdaripour, S.; Barufaldi, C.; Khamene, A. Integrated System for Robust 6-DOF Positioning Utilizing New Closed-Form Visual Motion Estimation Methods in Planar Terrains. *IEEE J. Oceanic Eng.* **2006**, *31*, 533–550.
34. Nicosevici, T.; Gracias, N.; Negahdaripour, S.; Garcia, R. Efficient three-dimensional scene modeling and mosaicing. *J. Field Rob.* **2009**, *26*, 759–788.
35. Shkurti, F.; Rekleitis, I.; Scaccia, M.; Dudek, G. State estimation of an underwater robot using visual and inertial information. In Proceedings of the 2011 IEEE/RSJ Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25-30 September 2011.
36. Burguera, A.; Bonin-Font, F.; Oliver, G. Trajectory-Based Visual Localization in Underwater Surveying Missions. *Sensors* **2015**, *15*, 1708–1735.
37. Palomeras, N.; Nagappa, S.; Ribas, D.; Gracias, N.; Carreras, M. Vision-based localization and mapping system for AUV intervention. In Proceedings of the 2013 MTS/IEEE OCEANS, San Diego, CA, USA, 10–14 June 2013.
38. Creuze, V. Monocular Odometry for Underwater Vehicles with Online Estimation of the Scale Factor. In Proceedings of the IFAC 2017 World Congress, Toulouse, France, 9–14 July 2017.
39. Strasdat, H.; Montiel, J.; Davison, A.J. Visual SLAM: Why filter? *Image Vision Comput.* **2012**, *30*, 65–77.
40. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), Nara, Japan, 13-16 November 2007; pp. 225–234.
41. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. Real Time Localization and 3D Reconstruction. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17-22 June 2006; pp. 363–370.
42. Strasdat, H.; Davison, A.J.; Montiel, J.M.M.; Konolige, K. Double window optimisation for constant time visual SLAM. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2352–2359.
43. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Rob.* **2015**, *31*, 1147–1163.
44. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Rob.* **2017**, *33*, 249–265.
45. Engel, J.; Schops, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 834–849.
46. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-Based Visual-Inertial Odometry using Nonlinear Optimization. *Int. J. Rob. Res.* **2015**, *34*, 314–334.
47. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust visual inertial odometry using a direct EKF-based approach. In Proceedings of the 2015 IEEE/RSJ Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.

48. Lin, Y.; Gao, F.; Qin, T.; Gao, W.; Liu, T.; Wu, W.; Yang, Z.; Shen, S. Autonomous aerial navigation using monocular visual-inertial fusion. *J. Field Rob.* **2018**, *35*, 23–51.
49. Bouguet, J.Y. *Pyramidal Implementation of the Lucas Kanade Feature Tracker*. Intel: Santa Clara, CA, USA, 2000.
50. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6-13 November 2011; pp. 2548–2555.
51. Rosten, E.; Porter, R.; Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32* 105-119.
52. Alahi, A.; Ortiz, R.; Vandergheynst, P. FREAK: Fast Retina Keypoint. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, 16-21 June 2012; pp. 510–517.
53. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision* **2004**, *60*, 91–110.
54. Baker, S.; Matthews, I. Lucas-Kanade 20 Years On: A Unifying Framework. *Int. J. Comput. Vision* **2004**, *56*, 221–255.
55. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395.
56. Nister, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770.
57. Kneip, L.; Scaramuzza, D.; Siegwart, R. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 2969–2976.
58. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
59. Quigley, M.; C., K.; Gerkey, B.P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: an open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
60. Kneip, L.; Furgale, P. OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1–8.
61. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g2o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3607–3613.
62. Duarte, A.C.; Zaffari, G.B.; da Rosa, R.T.S.; Longaray, L.M.; Drews, P.; Botelho, S.S.C. Towards comparison of underwater SLAM methods: An open dataset collection. In Proceedings of the 2016 MTS/IEEE OCEANS, Monterey, CA, USA, 19–23 September 2016.
63. Furgale, P.; Rehder, J.; Siegwart, R. Unified temporal and spatial calibration for multi-sensor systems. In Proceedings of the 2013 IEEE/RSJ Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 1280–1286.
64. Łuczyński, T.; Pfingsthorn, M.; Birk, A. The Pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings. *Ocean Eng.* **2017**, *133*, 9 – 22.
65. Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
66. Umeyama, S. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380.
67. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the 2012 IEEE/RSJ Intelligent Robots and Systems (IROS), Taipei, Taiwan, 7–12 Oct. 2012; pp. 573–580.

68. Galvez-Lopez, D.; Tardos, J.D. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Rob.* **2012**, *28*, 1188–1197.