Robust Visual Odometry in Underwater Environment

Jun Zhang
Australian Centre for Robotic Vision
Australian National University
Canberra, Australia
jun.zhang2@anu.edu.au

Viorela Ila
Australian Centre for Robotic Vision
Australian National University
Canberra, Australia
viorela.ila@anu.edu.au

Laurent Kneip
Mobile Perception Lab
SIST, ShanghaiTech
Shanghai, China
lkneip@shanghaitech.edu.cn

***Abstract*—The accurate estimation of pose and velocity of an autonomous underwater vehicle (AUV) is critical to ensure the repeatability and validity of scientific data that is captured using sensors onboard the AUV. A low-cost and effective way is by using stereo camera sensors to perform visual odometry (VO). However, this is a difficult problem in underwater due to poor imaging condition and inconsistent motion caused by water flow. This paper proposes a robust and effective stereo underwater VO system that can overcome aforementioned difficulties and accurately localize the AUV. Experimental results demonstrate that the proposed pipeline outperforms existing VO systems in underwater environment, as well as obtains a comparative performance on the KITTI benchmark dataset.**

I. INTRODUCTION

In recent years, the demand for exploration in underwater environment using autonomous underwater vehicle (AUV) is constantly increasing. Positioning sensors such as Doppler Velocity Logs (DVL) or acoustic transponders like long baseline system (LBL) and ultra short baseline system (USBL) are sometimes used in localizing AUVs . Although these methods can provide accurate pose estimation, they are very expensive and not easy to integrated or deployed in many cases.

A cost effective alternative is to use visual sensors and perform visual odometry (VO). This technique is becoming popular in computer vision and robotics [1]–[3], and provides a low-cost and effective solution to estimate the robot trajectory. Nevertheless, this becomes comparatively challenging in underwater environment due to the following issues: a) As is shown in Fig.3, the imaging conditions in water are poor due to light attenuation, poor/artificial illumination, haze and scattering. When the AUV operates in shallow waters, scattering of the sun light is highly problematic. b) Motion blur can also be present and is due to motion of the robot while the camera shutter is open. c) The vehicle/camera's motion is inconsistent with oscillation, especially in shallow underwater area, due to the water waves. All these problems greatly increases the difficulty in estimating the robot location.

In this paper, we intensively test and evaluate possible solutions of the mentioned problems, and propose a stereo underwater VO system that is able to robustly and accurately localize the AUV. Our system is built upon a popular visual-based localization system in robotics called ORB-SLAM2 [1]. We carefully modified this system to accommodate the above mentioned challenging conditions and we demonstrate that the proposed pipeline outperforms existing VO systems in underwater environment.

II. RELATED WORK

Using a set of image sequence from camera sensor, visual odometry seeks to incrementally estimate the motion of the vehicle from visual information of the environment. A VO system mainly refers to an open loop estimation of the robot location, in comparison to simultaneous localization and mapping (SLAM) which integrates loop-closure. According to different formulations, VO methods can be divided into two categories: feature-based methods [1], [4] based on detecting salient features in the images and, direct [3] or semi-direct [2] methods, which directly estimate motion using all or patches of the pixels in the image. The standard pipeline of feature-based methods consist of extracting sparse salient features in each image, matching them in successive frames using invariant feature descriptors, and then recovering the structure and camera motion using Epipolar geometry. Feature-based methods are robust to brightness inconsistencies and large view-point changes among consecutive frames. Nevertheless, feature extraction and matching bring considerable computational cost. Besides, few features can be extracted and matched in low-texture and blurring scenes, which may result in losing track easily. Direct methods, on the other hand, estimate structure and motion directly from intensity values in the image. One significant advantage of direct methods is that they can perform relatively dense 3D reconstruction, because more image information is used in tracking. However, they become unstable in rapid light changing environment, as the brightness constancy assumption does not always hold.

Current state-of-the-art VO methods work effectively under certain conditions such as, smooth motion, static scene, good illumination and rich texture, etc. However, most of these conditions can not be met in the underwater case, which results in either large drift or complete failure in trajectory estimation when using off-the-shelf VO systems. In [5], the authors compare different open-source VO solutions in different environments, including underwater reefs and shipwrecks. From their results, we can see that direct VO methods do not work in underwater, while feature-based methods are relatively

more consistent on equal conditions. The results make sense because direct methods estimate motion via minimizing the photometric error which is significantly influenced by poor imaging conditions and the scattering phenomena in underwater, while feature-based method use feature descriptors that are robust to the change in perspective and illumination. In our work, therefore, we choose to develop a feature-based VO system.

A standard feature-based visual odometry pipeline consists of three main steps: feature extraction, feature association (matching) and motion estimation. In the feature extraction step, salient keypoints are detected in the image, and for each detected keypoint, a compact descriptor is extracted, which can be used to match against others. In feature-based VO task, features like Scale-Invariant Feature Transform (SIFT) [6], Speed Up Robust Feature (SURF) [7] and Oriented FAST and Rotated BRIEF (ORB) [8] are widely used. Extensive comparison experiments have been conducted on these features literarily [9], while few work has evaluated them in underwater environment with visual odometry task [10]–[12]. In this paper, therefore, we experimentally compare these features to see which solution is best for performing underwater visual odometry. During the feature association step, corresponding features between consecutive image frames are searched by comparing their feature descriptors. Brute-force search is a straightforward method to find the correspondences with high accuracy. However, it requires high time-cost if the candidate number is high. Another widely-used approach is FLANN [13], which is fast and relatively accurate, but it is not robust enough if there is high similarity in the detected features. In visual odometry application, an effective way to do feature matching is making use of the previous motion model to predict an approximate feature projecting location and narrow the search area. However, it is prone to fail in underwater environment due to the inconsistent motion. In this paper, we make use of all the four stereo images: current left, right and previous left, right frames and perform a loop search to find the matchings. The proposed matching algorithm is not only fast and accurate, but also robust against inconsistent motion. Motion estimation is the core step in a VO system. In this step, the camera motion between the current and the previous frame is computed. This is normally treated as the Perspective-n-Point (PnP) problem that retrieves the camera motion with respect to a scene object from n 2D-3D point correspondences [14], [15]. In order to robustly estimate the motion, a standard approach consists of first using P3P in a RANSAC scheme [16] to remove the outliers, and then PnP on all remaining inliers. Similarly, in our VO pipeline, for each new coming frame, an initial motion is acquired using P3P [17] with RANSAC, then a non-linear optimization for all the inliers is applied to refine the final solution.

III. ALGORITHM

The state-of-the-art ORB-SLAM2 VO pipeline failed to work with our underwater datasets due to the challenging problems discussed in Section I, see Fig.5(a). We thereby carefully evaluate and modified it as follows. Fig.1 shows our visual odometry pipeline in comparison with the VO part of ORB-SLAM2. In the feature extraction, we experimentally compare three state-of-the-art feature techniques, and find the optimal solution that is most robust against challenging image condition to conduct underwater visual odometry. Inconsistent motion is handled by two techniques in feature association and motion estimation, respectively: a) Instead of using constant motion model (in frame-to-frame tracking) to match features, we propose an effective circular search between current left, right and previous left, right frames (Quad matching) to find the matchings. Our Quad matching approach is simple, fast and not affected by unpredictable motion. b) Rather than directly optimizing (motion-only Bundle Adjustment) with motion model as initial value, we perform a simple pose estimation from 2D to 3D correspondences (P3P) [17] with RANSAC to get an initial pose for each new camera frame.

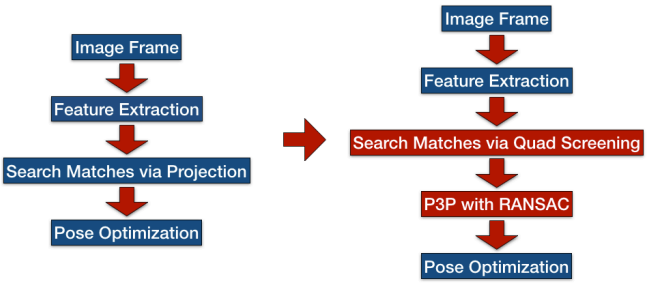


Fig. 1: The visual odometry pipeline of ORB-SLAM2 (left) and ours (right).

Algorithm.1 presents the pseudo code of our proposed VO pipeline. The algorithm inputs the stereo images sequence $\mathcal{I}$ and outputs the camera poses $\mathcal{C}$ in the world coordinate system. For each incoming stereo frame $\{{}^{l}\boldsymbol{I}_i, {}^{r}\boldsymbol{I}_i\}$, feature extraction (denoted as Φ) is performed with keypoints $\{{}^{l}\boldsymbol{P}_i, {}^{r}\boldsymbol{P}_i\}$ and their descriptors $\{{}^{l}\boldsymbol{D}_i, {}^{r}\boldsymbol{D}_i\}$ as outcome, which are then used to perform stereo matching ($\mathcal{M}_s$) via Epipolar search. The first frame is initialized as the world coordinate frame and an initial map $\boldsymbol{L}_i = \boldsymbol{L}$ is created with all the stereo matchings $\boldsymbol{M}_i^s$ via back-projection (Π^{-1}). The map $\boldsymbol{L}$ is maintained and updated in our pipeline for the purpose of performing Local Bundle Adjustment (LBA). Furthermore, the LBA process is used in outlier rejection process by removing the points with high reprojection error.

Except for the initialization, as shown in step 9-10, we do not use all the stereo matchings $\boldsymbol{M}_i^s$ to find the temporal matchings $\boldsymbol{M}_{i-1,i}^t$. Instead, only the union set of 1) $\boldsymbol{L}_{i-1} \in \boldsymbol{L}$, the 3D landmarks that can be found in frame $\boldsymbol{I}_{i-1}$, and 2) a set of new back-projected 3D landmarks $\boldsymbol{L}_{temp}$ from $\boldsymbol{I}_{i-1}$, whose depth uncertainty are lower than a preset depth threshold d_{th}. In this way, only reliable 3D landmarks are used and the motion estimation is more robust and accurate.

The selected landmarks and their corresponding features in $\boldsymbol{I}_{i-1}$ are used to find their matchings in $\boldsymbol{I}_i$ via Quad matching ($\mathcal{M}_q$). Fig.2 and Algorithm.2 illustrates the details

Algorithm 1 Propose Visual Odometry Pipeline

Require: $\mathcal{I} = \{{}^{l}\boldsymbol{I}_1, {}^{r}\boldsymbol{I}_1, ..., {}^{l}\boldsymbol{I}_n, {}^{r}\boldsymbol{I}_n\}$
Ensure: $\mathcal{C} = \{\boldsymbol{C}_1^w, ..., \boldsymbol{C}_n^w\}$
1: **for** $i = 1$; $i <= n$; $i++$ **do**
2: $\{{}^{l}\boldsymbol{P}_i, {}^{l}\boldsymbol{D}_i\} = \Phi\{{}^{l}\boldsymbol{I}_i\}, \{{}^{r}\boldsymbol{P}_i, {}^{r}\boldsymbol{D}_i\} = \Phi\{{}^{r}\boldsymbol{I}_i\}$;
3: $\boldsymbol{M}_i^s = \mathcal{M}_s\{{}^{l}\boldsymbol{P}_i, {}^{l}\boldsymbol{D}_i, {}^{r}\boldsymbol{P}_i, {}^{r}\boldsymbol{D}_i\}$;
4: **if** $i = 1$ **then**
5: $\boldsymbol{C}_i^w = eye(4,4)$;
6: $\boldsymbol{L}_i = \Pi^{-1}\{\boldsymbol{M}_i^s, \boldsymbol{K}, \boldsymbol{C}_i^w\}$;
7: continue;
8: **else**
9: $\boldsymbol{L}_{temp} = \Pi^{-1}\{\boldsymbol{M}_{i-1}^s, d_{th}, \boldsymbol{K}, \boldsymbol{C}_{i-1}^w\}$;
10: $\boldsymbol{L}_{i-1} = \boldsymbol{L}_{i-1} \cup \boldsymbol{L}_{temp}$;
11: **end if**
12: $\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i\} = \mathcal{M}_q\{\boldsymbol{L}_{i-1}, \boldsymbol{D}_{i-1}, \boldsymbol{D}_i, \boldsymbol{M}_{i-1}^s, \boldsymbol{M}_i^s\}$;
13: $\{\boldsymbol{C}_i^w, \boldsymbol{L}_i\} = \Theta\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i, \boldsymbol{P}_i\}$;
14: $\{\boldsymbol{C}_i^w, \boldsymbol{L}_i\} = \Psi\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i, \boldsymbol{P}_i\}$;
15: **end for**
16: **return** $\mathcal{C}$ **;**

Algorithm 2 Quad Matching

Require: :
1: $\boldsymbol{M}_{i-1}^s = \{\boldsymbol{m}_{i-1,j}^s \mid \boldsymbol{m}_{i-1,j}^s = \{{}^{l}\boldsymbol{p}_{i-1,j}, {}^{r}\boldsymbol{p}_{i-1,j}\}\}$ Stereo matchings in $\boldsymbol{I}_{i-1}$;
2: $\boldsymbol{M}_i^s = \{\boldsymbol{m}_{i,j}^s \mid \boldsymbol{m}_{i,j}^s = \{{}^{l}\boldsymbol{p}_{i,j}, {}^{r}\boldsymbol{p}_{i,j}\}\}$ Stereo matchings in $\boldsymbol{I}_i$;
Ensure: $\boldsymbol{M}_{i-1,i}^t$: Temporal matchings between $\boldsymbol{I}_{i-1}$ and $\boldsymbol{I}_i$ frames;
3: **for** each $\boldsymbol{m}_{i-1,j}^s = \{{}^{l}\boldsymbol{p}_{i-1,j}, {}^{r}\boldsymbol{p}_{i-1,j}\} \in \boldsymbol{M}_{i-1}^s$ **do**
4: $\boldsymbol{S}_l = \mathcal{S}_w\{{}^{l}\boldsymbol{I}_i\}, \boldsymbol{S}_r = \mathcal{S}_w\{{}^{r}\boldsymbol{I}_i\}$;
5: ${}^{l}\boldsymbol{p}_{i,j}^*$=$\{{}^{l}\boldsymbol{p}_{i,j} \in \boldsymbol{S}_l \mid \boldsymbol{D}_{dist}({}^{l}\boldsymbol{p}_{i,j}, {}^{l}\boldsymbol{p}_{i-1,j})$ is min$\}$
6: ${}^{r}\boldsymbol{p}_{i,j}^*$=$\{{}^{r}\boldsymbol{p}_{i,j} \in \boldsymbol{S}_r \mid \boldsymbol{D}_{dist}({}^{r}\boldsymbol{p}_{i,j}, {}^{r}\boldsymbol{p}_{i-1,j})$ is min$\}$;
7: **if** ${}^{*}\boldsymbol{m}_{i,j}^s = ({}^{l}\boldsymbol{p}_{i,j}^*, {}^{r}\boldsymbol{p}_{i,j}^*) \in \boldsymbol{M}_i^s$ **then**
8: $\boldsymbol{M}_{i-1,i}^t \leftarrow \{\boldsymbol{m}_{i-1,j}^s, {}^{*}\boldsymbol{m}_{i,j}^s\}$;
9: **end if**
10: **end for**
11: **return** $\boldsymbol{M}_{i-1,i}^t$ **;**

of Quad matching, which is the implementation of line 12 in Algorithm.1. For each frame step, stereo matchings are computed by Epipolar line search. Then each keypoint of stereo matchings in previous left and right frame perform a window search ($\mathcal{S}_w$) in current left and right frame to get their optimal matchings by comparing their descriptor distances ($\boldsymbol{D}_{dist}$), respectively. If their optimal match happens to be the stereo match in the current frame, these four keypoints are accepted as a set of quad matchings.

At the end of the loop (lines 13 and 14 in Algorithm.1), a P3P estimation with RANSAC (Θ) is performed to get an initial pose and inliers, and then a non-linear optimization (Ψ) is applied to refine the pose with all the inliers, which is based on Levenberg−Marquardt method implemented in g2o [18].

In comparison, the VO pipeline in ORB-SLAM2 is demonstrated in Algorithm.3, where the differences can be summarized as follows: 1) During initialization, as no motion has been recovered yet, Bag-of-Words (BoW) matching [19] ($\mathcal{M}_b$) is utilized in the first two frame to accomplish feature association in ORB-SLAM2 (Note that the details of BoW construction are omitted for simplification), see step 12-14 of Algorithm.2, however, it is unnecessary in our proposed pipeline. 2) In the temporal matching step, a coarse current camera position is calculated using motion model $\boldsymbol{T}$, then the temporal matchings are found by projecting the selected landmarks $\boldsymbol{L}_{i-1}$ into current frame $\boldsymbol{I}_i$ and search locally ($\mathcal{M}_p$). Nevertheless, this becomes invalid in underwater environment because of the motion inconsistency. Instead of this, we propose an effective Quad matching approach to find the correspondences, which is robust to unpredictable motion. 3) Only non-linear optimization is performed to recover the camera position. In this case, the result is easily affected by the initial value, i.e., coarse result computed from constant motion model. If the motion model is not reliable, the optimization may not get the optimal solution or even end up with failure. This has been observed from our experiment that only by optimization the system loses track easily.

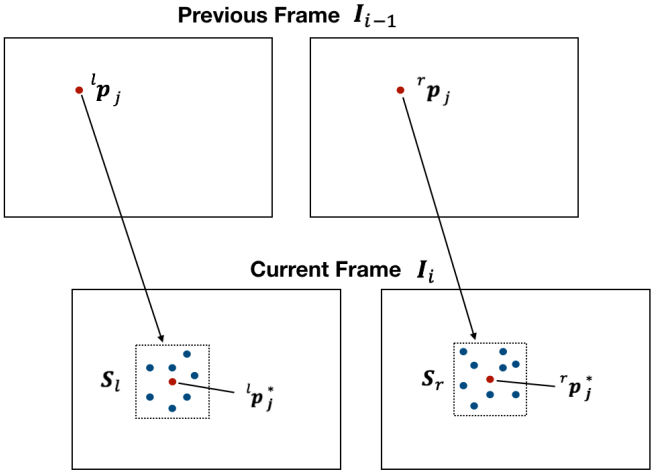


Fig. 2: Sketch map of our Quad matching method.

IV. DATASET

In this paper, two underwater datasets will be tested and evaluated. One of them we call it as Underwater Coral Dataset, which is a stereo video sequence dataset that is captured manually over the undersea coral reef using a stereo Gopro rig. The video contains 2500 frames of image size 1920x1080. It starts at one marked place, and heads forward for around 60 meters to another marked spot, then turns back to its starting point. As is shown in Fig.3(a), this dataset is affected by certain level of hazing, and most of the images are half invisible because of the camera viewing angle. The most challenging part of this dataset is the inconsistent motion in most of the frames due to the shallow water waves, which not only results in motion blur that affects the feature extraction

Algorithm 3 Visual Odometry pipeline in ORB-SLAM2

Require: $\mathcal{I} = \{{}^{l}\boldsymbol{I}_1, {}^{r}\boldsymbol{I}_1, ..., {}^{l}\boldsymbol{I}_n, {}^{r}\boldsymbol{I}_n\}$
Ensure: $\mathcal{C} = \{\boldsymbol{C}_1^w, ..., \boldsymbol{C}_n^w\}$
for $i = 1;\ i <= n;\ i++$ **do**
  $\{{}^{l}\boldsymbol{P}_i, {}^{l}\boldsymbol{D}_i\} = \Phi\{{}^{l}\boldsymbol{I}_i\}, \{{}^{r}\boldsymbol{P}_i, {}^{r}\boldsymbol{D}_i\} = \Phi\{{}^{r}\boldsymbol{I}_i\};$
  $\boldsymbol{M}_i^s = \mathcal{M}_s\{{}^{l}\boldsymbol{P}_i, {}^{l}\boldsymbol{D}_i, {}^{r}\boldsymbol{P}_i, {}^{r}\boldsymbol{D}_i\};$
  if $i = 1$ **then**
    $\boldsymbol{C}_i^w = eye(4,4);$
    $\boldsymbol{L}_i = \Pi^{-1}\{\boldsymbol{M}_i^s, \boldsymbol{K}, \boldsymbol{C}_i^w\};$
    continue;
  else
    $\boldsymbol{L}_{temp} = \Pi^{-1}\{\boldsymbol{M}_{i-1}^s, d_{th}, \boldsymbol{K}, \boldsymbol{C}_{i-1}^w\};$
    $\boldsymbol{L}_{i-1} = \boldsymbol{L}_{i-1} \cup \boldsymbol{L}_{temp};$
  end if
  if $i = 2$ **then**
    $\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i\} = \mathcal{M}_b\{\boldsymbol{L}_{i-1}, \boldsymbol{D}_{i-1}, \boldsymbol{D}_i, \boldsymbol{M}_{i-1}^s, \boldsymbol{M}_i^s\};$
    $\{\boldsymbol{C}_i^w, \boldsymbol{T}, \boldsymbol{L}_i\} = \Psi\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i, \boldsymbol{P}_i\}$
  else
    $\boldsymbol{C}_i^w = \boldsymbol{T} \cdot \boldsymbol{C}_{i-1}^w;$
    $\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i\} = \mathcal{M}_p\{\boldsymbol{L}_{i-1}, \boldsymbol{C}_i^w, \boldsymbol{D}_{i-1}, \boldsymbol{D}_i, \boldsymbol{M}_{i-1}^s, \boldsymbol{M}_i^s\};$
    $\{\boldsymbol{C}_i^w, \boldsymbol{T}, \boldsymbol{L}_i\} = \Psi\{\boldsymbol{M}_{i-1,i}^t, \boldsymbol{L}_i, \boldsymbol{P}_i\};$
  end if
end for
return $\mathcal{C}$ **;**

and matching, but also makes any motion model assumption fail.

The other dataset is the Underwater Shipwreck Dataset [5] that is also a stereo dataset of capturing around an underwater shipwreck by a stereo Gopro rig. This video (totally 1800 frames of image size 1920x1080) starts at the front of the shipwreck, and circles around it for a lap. The camera motion in this dataset is relatively smooth comparing to Underwater Coral Dataset and only in some parts of the image sequence are encountered with severe shaking. Nevertheless, the big obstacle in this dataset is the poor imaging condition that is caused by the turbid water and light attenuation, leading to only a limited number of valid features being detected and tracked in every frame.

V. FEATURE EVALUATION

In the underwater environment, feature should be carefully selected to be robust against poor illumination, haze and scattering. SIFT [6] feature is one of the highest quality feature descriptors due to its strong invariance to scale, rotation, illumination change and noise. However, it requires a large computational complexity which is a major drawback for real-time applications such as visual odometry. SURF [7] feature, which is an approximation of SIFT, performs faster than SIFT without reducing much quality of the detected points. Alternatively, ORB [8] is another efficient choice, which is a binary descriptor requiring less complexity but is still highly distinctive.

(a)

(b)

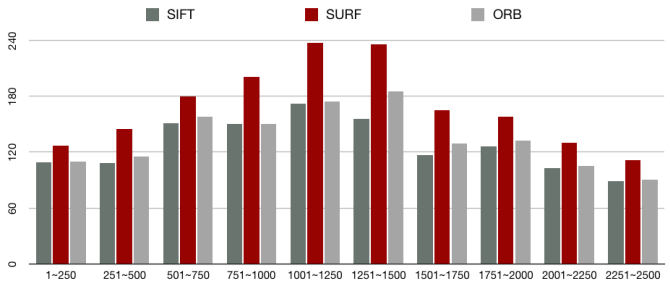
Fig. 3: Selected typical sample images of (a) the Underwater Coral Dataset and (b) the Underwater Shipwreck Dataset.

TABLE I: COMPARISON OF AVERAGE INLIER NUMBER OBTAINED BY THREE FEATURE TECHNIQUES.

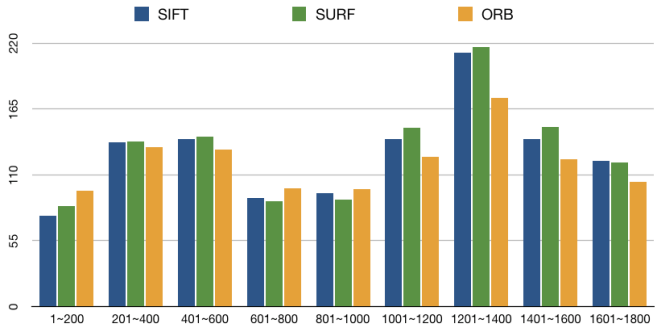
Feature Techniques	SIFT	SURF	ORB	Mean Value
Coral Dataset	128	**169**	134	144
Shipwreck Dataset	126	**129**	119	125

To see their performance on underwater images, we tested the three feature techniques on our visual odometry pipeline, separately. To be more precise, in our pipeline, FAST [20] corners at 8 scale levels are detected uniformly distributed on the image. To ensure that enough features are obtained to track the camera, we set the number of detected features to 6000 per image (in size 1920x1080) in both underwater datasets due to the bad imaging quality. At the same time, SIFT, SURF and ORB descriptors are extracted from these corners for comparison. In addition, subpixel correlation, orientation and scale consistency are also considered during the matching step.

Their performance is evaluated by comparing the number of matched inliers, which are obtained after the camera pose between subsequent frames is estimated using the matchings found by each type of descriptor. This is reasonable because, by getting more inliers, the accuracy of pose estimation is higher. Table.I presents the average inlier number acquired



(a)



(b)

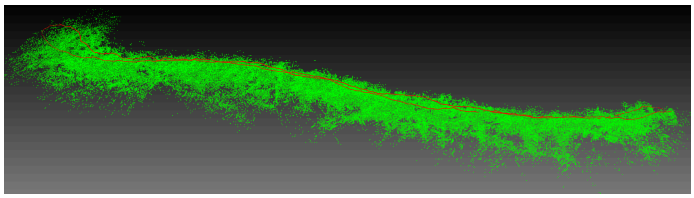
Fig. 4: Comparison of inliers number among the three feature techniques in (a) the Underwater Coral Dataset and (b) the Underwater Shipwreck Dataset, where the vertical axis refers to inlier number, and the horizontal axis refers to frame index.

by the tested features on both underwater datasets, which intuitively show that SURF techniques outperforms the other two, especially in the Underwater Coral Dataset. We thereby choose SURF technique to extract feature descriptor. From the mean values between datasets, we can see that the Coral Dataset obtains approximately 15% more inliers than that of the Shipwreck Dataset. This accounts for the truth that the imaging condition in Shipwreck Dataset is poorer and less valid features can be used to track.

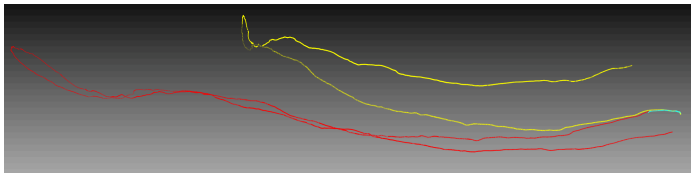
A more detailed histogram distribution of the inlier number is demonstrated in Fig.4, which reflects the characters of both datasets. Specifically, for instance, in (a) there are less inliers at the beginning and towards the end, because the camera is heading forward and most of the scene during that time is half visible, while in the middle frames (1000-1500) the inliers increase, as the camera is down-looking at the ground and taking a turn, so more features can be detected and tracked. In (b), for another example, the frames between 600 and 1000 gets lowest inliers among the whole sequence, when the camera is approaching the stern and turning back. There is little structure in the scene to be tracked in this area. That is why the drift grows quickly during that time, as shown in Fig.6(b).

VI. RESULTS

We first show the result of our system performing visual odometry task on the Underwater Coral Dataset. As is illustrated in Fig.5(a), our method can successfully recover the whole camera trajectory, which is very close to the real trajectory, as well as the 3D structure. The tracking time is around 2.8 seconds per frame when run on an i7 quad-core 2.5Ghz laptop. This is mainly because of the high amount of feature extraction and matching. This can be improved by employing a GPU-based parallel implementation and achieve real-time performance.



(a)



(b)

Fig. 5: (a) Camera trajectory (red) and the 3D structure (green) produced by our proposed method. (b) Comparison of trajectories generated by our proposed method (red), LIBVISO2 (yellow) and ORB-SLAM2 (cyan).

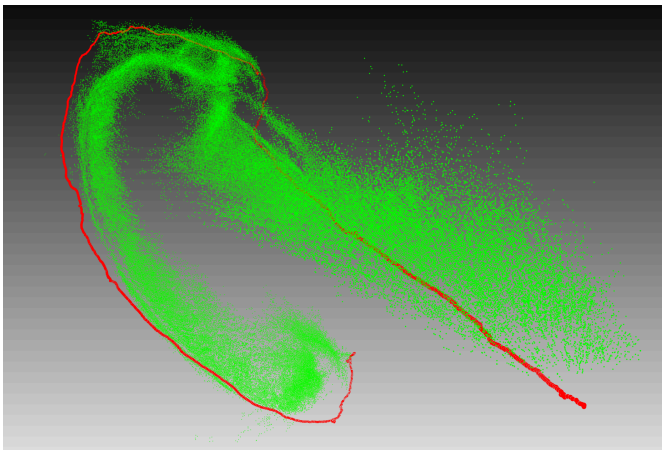
We also compared our method with two state of the art VO systems: LIBVISO2 [4] and ORB-SLAM2 [1]. For comparison, We remove the loop closure module (including Global Bundle Adjustment) in ORB-SLAM2 to make it a pure visual odometry system. Besides, we set all the shared same parameters in our method and ORB-SLAM2 as the same, and the parameters in LIBVISO2 are kept as the default setup. Fig.5(b) shows that LIBVISO2 succeeds to run the whole dataset, but it introduces large drift. ORB-SLAM2 can only survive the first 60 frames (Fig.5(b), cyan color), and even with reset, it loses track quickly.

In order to evaluate the proposed VO, we also tested the three methods on the KITTI benchmark dataset [21], which has ground truth trajectories. Table.II demonstrates the quantity results (Relative Pose Error [22]) of the three methods, and Fig.7 shows the absolute trajectories compared with the ground truth trajectory. It can be seen that our proposed method has comparable performance with original version of ORB-SLAM2 and the ground truth.

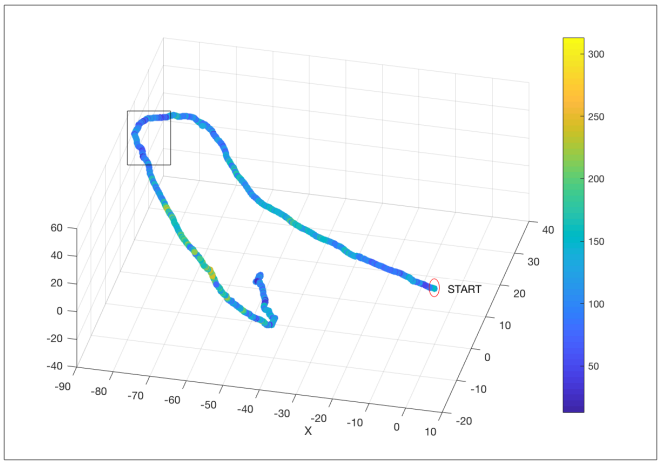
In addition, Fig.6(a) shows the whole trajectory and the 3D structure generated from the challenging Underwater Shipwreck Dataset using our proposed method. We can see that drift accumulates all along the way, especially when the camera takes a turn and comes back from the stern. Fig.6(b) reveals the inlier distribution along the trajectory. This result intuitively shows the the drift correlates closely with the inlier number. As an example, at the end of the turn (see the black

TABLE II: COMPARISON OF TRANSLATION (meter) AND ROTATION (degree) RMSE IN KITTI DATASET.

Method	Sequence	00	01	02	03	04	05	06	07	08	09	10
Proposed	R	**0.6195**	0.2330	**0.2019**	**0.0846**	0.0704	**0.1881**	**0.1267**	**0.1485**	0.1413	**0.1259**	**0.1316**
	t	0.2083	8.8563	0.1908	**0.1047**	0.1395	**0.0923**	0.0995	0.0927	**0.3410**	**0.1534**	0.1179
ORB-SLAM2	R	0.6196	**0.1318**	0.2048	0.0891	**0.0696**	0.1987	0.1303	0.1539	**0.1410**	0.1318	0.1355
	t	0.2032	**0.4546**	0.1933	0.1106	**0.1234**	0.0947	**0.0938**	0.0909	0.3424	0.1656	0.1134
LIBVISO2	R	0.6471	0.2883	0.2384	0.1404	0.1596	0.2512	0.2251	0.2513	0.2149	0.2023	0.1831
	t	**0.1941**	3.2146	**0.1810**	0.1185	0.1715	0.1024	0.1199	**0.0830**	0.3415	0.1694	**0.1076**



(a)



(b)

Fig. 6: (a) Camera trajectory (red) and the 3D structure (green) for the shipwreck dataset, produced by our proposed method. (b) Color bar showing the distribution of inlier number on the whole trajectory.

bounding box), where the inlier numbers are lower, the drift of the rotation becomes larger.

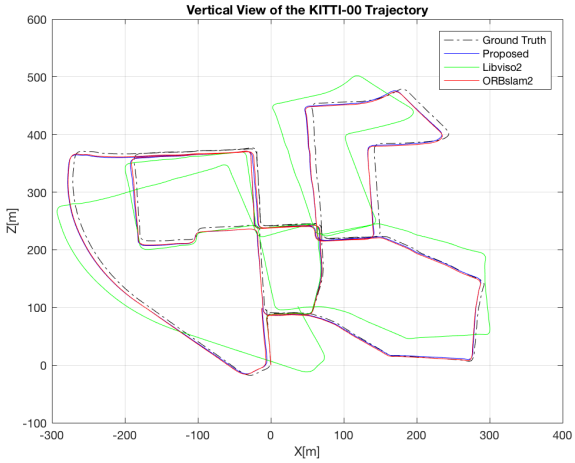
VII. EXTRA EXPERIMENT

In our pipeline, we implement an extra experiment that try to improve the turbid image quality before feature extraction, and expect that more reliable features can be extracted to estimate motion. Many image enhancement or dehazing methods have been proposed to tackle this problem. Here, we select three state-of-the-art methods to compare and evaluate how enhancing or dehazing techniques would affect the visual odometry task. Concretely, Contrast-limited adaptive histogram equalization (CLAHE) [23] is a widely-used image contrast enhancement algorithm, which divides the images into regions and performs local histogram equalization (HE) and reduces noise by partially reducing the local HE. Underwater images and videos enhancement by fusion (FUSION) [24] is a fusion-based framework that blends different filters to enhance underwater images. Dark Channel Prior Dehazing (DCPD) [25] is based on a kind of statistics of the haze-free outdoor images called dark channel prior. Together with haze imaging model, the thickness of the haze can be directly calculated and finally a high quality haze-free image can be recovered.

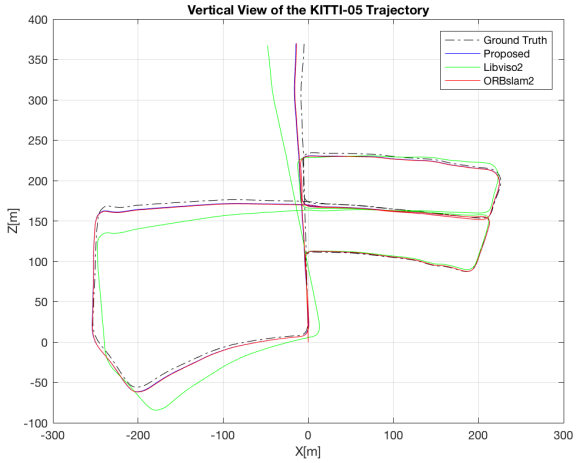
Fig.8 shows a comparison of enhancement results on both underwater datasets. Qualitatively, all the tested algorithms have improved the visibility at different levels, compared with the original image. More precisely, results of CLAHE and FUSION are brighter and have larger visible regions, i.e., far away unclear structure becomes more clear. On the contrary, the luminance becomes lower when applying DCPD, and not much visibility improvement is obtained, but the structure details become finer.

To evaluate their performance on visual odometry, they are all applied to test our VO pipeline on both underwater datasets, separately. Note that FUSION and DCPD fails to work on the Underwater Shipwreck Dataset, so only the CLAHE is compared with the default in this dataset. Similar to feature techniques comparison, the inlier number obtained in each frame are used as a comparable index. Fig.9 illustrates the comparison of inlier number distribution. It can be observed from (a) that, in some parts of the dataset (for instance, 1-250, 251-500, etc.), CLHAE and FUSION have increased the average inlier number, but in some other parts, they got even much less inliers than the original (501-750,etc.). Similar trend can be observed in (b) as well. This suggests that certain level of noise has been introduced when the image is enhanced using the surveyed methods.

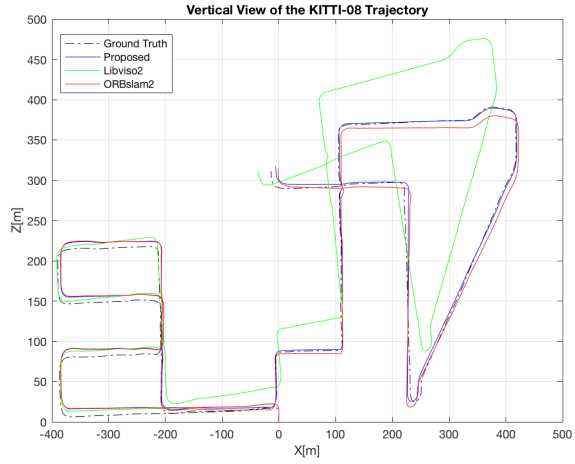
Table.III demonstrates the average values of the whole sequence. In particular, the average match number drops slightly after enhancement. This means that less extracted features are qualified to be chosen in our matching procedure despite the fact that the scene visibility has been improved by enhance-



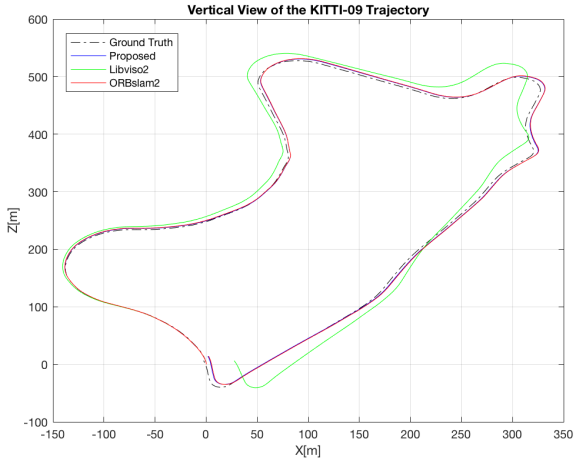
(a) KITTI-00



(b) KITTI-05



(c) KITTI-08



(d) KITTI-09

Fig. 7: Trajectory results on part of the selected sequences due to limited space.

ment. So enhancement measure should have introduced certain level of noise to the image and brought negative effect on the completeness and uniqueness in feature descriptor. Overall, the results reveal the fact that none of the approaches has significant contribution to the visual odometry task, though they do increase the image visibility in accordance with the human's observing experience.

VIII. CONCLUSIONS

In this paper, we introduced a robust and effective stereo underwater VO system that can accurately recover the camera motion. We carefully analyze each part of our visual odometry system, including image restoration, feature extraction and matching, motion estimation, to explore possibility of improvement on the system in underwater environment. Experimental results indicate that our system helps to achieve excellent performance in localizing the camera in underwater and obtains satisfactory results in the KITTI benchmark

TABLE III: COMPARISON OF QUANTITY RESULTS TOWARDS DIFFERENT ENHANCEMENT METHODS.

Underwater Coral Dataset				
Enhancement Techniques	Oiginal	CLAHE	FUSION	DCPD
Average Match Number	**242**	237	224	219
Average Inlier Number	**135**	134	129	123
Average Inlier Rate	0.5425	0.5538	**0.5613**	0.5429

Underwater Shipwreck Dataset		
Enhancement Techniques	Original	CLAHE
Average Match Number	**257**	237
Average Inlier Number	**135**	131
Average Inlier Rate	0.5253	**0.5527**

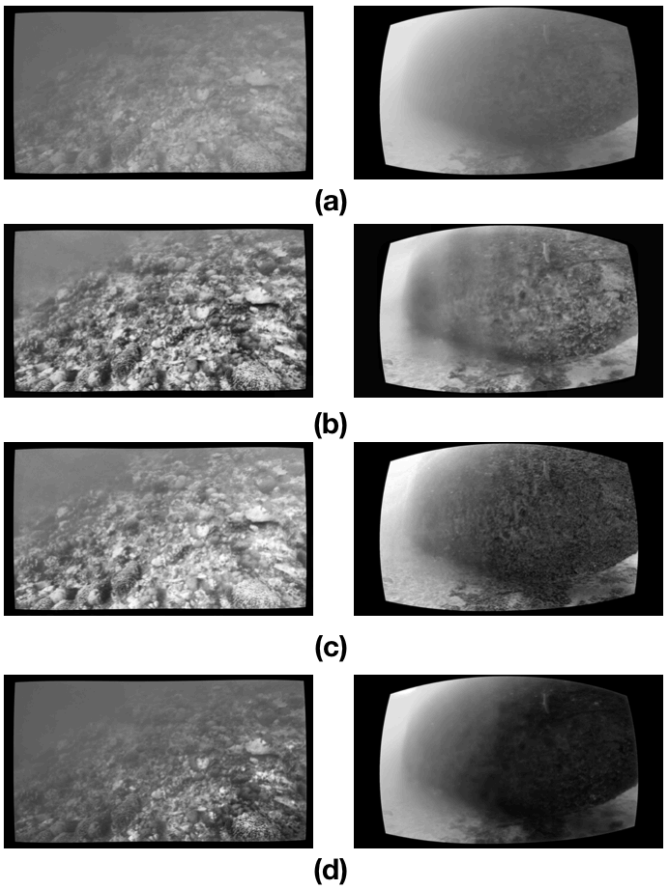


Fig. 8: Comparison of different image enhancement algorithms. (a) Original image. (b) Enhanced by CLAHE. (c) enhanced by FUSION. (d) Enhanced by DCPD.

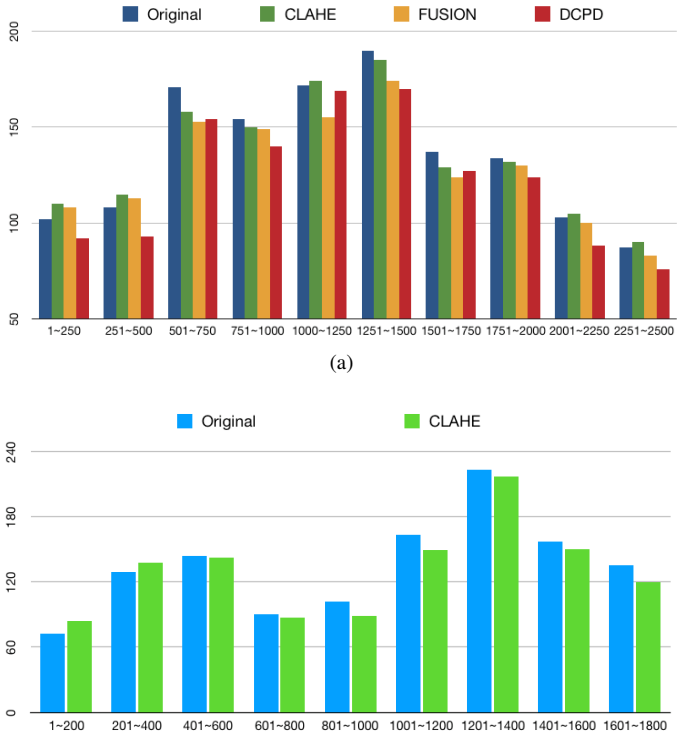


Fig. 9: Comparison of inliers number among the enhancement methods in (a) the Underwater Coral Dataset and (b) the Underwater Shipwreck Dataset, where the vertical axis refers to inlier number, and the horizontal axis refers to frame index.

dataset by comparing to the state-of-the-art and the ground truth.

Nevertheless, there are problems that remain to be solved. One of them is how to improve the underwater image quality for underwater visual odometry. The goal should be to increase the number of valid corners to be detected, simultaneously preserve the completeness of their features and avoid the increase of noise. Many image enhancement or dehazing methods has been proposed and some are special designed for underwater environment. But our experiment results has shown that most of them are not able to help effectively improving the VO task. In the future we plan to look into this specific problem and explore for an efficient solution.

ACKNOWLEDGMENT

This research is supported by the Australian Research Council through the Australian Centre of Excellence for Robotic Vision (CE140100016), and the Discovery Early Career Researcher Award (DECRA) Program (DE150101365). The authors would like to thank Dr. Feras Dayoub and Dr. Matthew Dunbabin for providing the Underwater Coral Dataset, and the group of Dr. Jason O'Kane for providing the Underwater Shipwreck Dataset.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[3] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[4] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[5] A. Q. Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane, and I. Rekleitis, "Experimental comparison of open source vision based state estimation algorithms," in *Proc. International Symposium on Experimental Robotics*, 2016.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to sift or surf," *2011 International Conference on Computer Vision*, pp. 2564–2571, 2011.

[9] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: Performance comparison for distorted images," *arXiv preprint arXiv:1710.02726*, 2017.

[10] F. Shkurti, I. Rekleitis, and G. Dudek, "Feature tracking evaluation for pose estimation in underwater environments," in *computer and robot*

vision (CRV), 2011 Canadian conference on. IEEE, 2011, pp. 160–167.

[11] S. Wirth, P. L. N. Carrasco, and G. O. Codina, "Visual odometry for autonomous underwater vehicles," in *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, 2013, pp. 1–6.

[12] A. Burguera, F. Bonin-Font, and G. Oliver, "Towards robust image registration for underwater visual slam," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 3. IEEE, 2014, pp. 539–544.

[13] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *In VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.

[14] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, pp. 155–166, 2009.

[15] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2969–2976.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[17] L. Kneip, P. Furgale, and R. Siegwart, "Using multi-camera systems in robotics: Efficient solutions to the npnp problem," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3770–3776.

[18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[19] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[20] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 573–580.

[23] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics gems IV*. Academic Press Professional, Inc., 1994, pp. 474–485.

[24] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing underwater images and videos by fusion," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 81–88.

[25] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2011.