Underwater Visual SLAM using a Bottom Looking Camera

Francisco Bonin-Font, Antoni Burguera, Gabriel Oliver
{francisco.bonin, antoni.burguera, goliver}@uib.es
Department of Mathematics and Computer Science
Systems, Robotics and Vision Group
University of the Balearic Islands
Carretera de Valldemossa km 7.5, Palma de Mallorca, Spain.

March 4, 2014

Abstract

This paper proposes a straightforward but effective approach to perform visual SLAM, especially suitable for underwater vehicles. One of the most important steps in this procedure is the image registration method, since it reinforces the data association and thus makes it possible to close loops reliably. Since the traditional EKF-SLAM approaches are usually costly in terms of running time, the approach presented in this paper strengthens this method by adopting a trajectory-based schema that reduces the computational requirements. The pose of the vehicle is estimated using an *Extended Kalman Filter* (EKF), which predicts the vehicle motion by means of a visual odometer and corrects these predictions using the data associations (loop closures) between the current frame and the previous ones. Since the use of standard EKFs entail linearization errors that can distort the vehicle pose estimations, the approach was also tested using an *Iterated Kalman Filter* (IEKF) instead. The approach has been tested on real underwater vehicles, in controlled scenarios and in shallow sea waters. The approach has shown an excellent performance in diverse experiments, with very limited errors of the estimated trajectory.

1 INTRODUCTION

1.1 Problem Statement

Thanks to recent technological advances, the sub-aquatic world is more accessible for exploration, scientific research and industrial activity. At present, *Remotely Operated Vehicles* (ROVs) are commonly used in a variety of applications, such as surveying, scientific sampling, rescue operations or industrial infrastructure inspection and maintenance.

Trying to overcome some of the intrinsic limitations of ROVs, such as their limited operative range or the need for a support vessel, *Autonomous Underwater Vehicles* (AUVs) are progressively being introduced, especially in highly

repetitive, long or hazardous missions. Because they are untethered and self-powered, AUVs offer a significant independence from support ships and weather conditions. This can reduce notably the operational costs and the complexity of human and material resources, comparing to operations conducted with tethered ROVs.

Localization, which consists in determining and keeping track of the robot location in the environment, becomes a crucial issue in AUVs. The mission success depends, to a great extent, on the precision of the estimated vehicle pose. Errors in orientation generate important drifts on the computed robot trajectory thus hindering the accomplishment of the programmed mission.

There are several ways to estimate the robot motion in underwater vehicles, for instance, (a) using inertial sensors, such as gyroscopes and accelerometers, (b) with odometry computed via acoustic sensors (sonars or DVLs) or cameras, and, (c) combining inertial sensors and odometers, fusing all the sensorial data by means of navigation filters, such as EKFs or particle filters, to smooth trajectories and errors ([15], [16], [13]).

Nevertheless, all these measurements are, to a greater or lesser extent, prone to drift, being necessary to adjust periodically the pose of the vehicle to reduce, as far as possible, the accumulated error. To this end, the so called *Simultaneous Localization And Mapping* (SLAM) [7] techniques constitute the most common and successful approach to perform precise localization. The principal aim of SLAM is the reduction of errors present in odometry by localizing the robot with respect to landmarks or significant points of the environment. This localization process is reinforced by recognizing regions previously visited by the robot in a process known as loop closing. Landmarks are incorporated to an incremental map and their location is refined simultaneously with the vehicle pose.

In most of the sub-aquatic environments, the process of sensing the environment becomes particularly complex. When light propagates in water, it interacts with molecules and dissolved particulate matter. As a consequence, the light traveling distance underwater is dramatically reduced when compared to air. Contrarily, sound propagates faster and it is able to travel larger distances in water than in air. Consequently, acoustic sensors have been traditionally considered the best choice for AUVs [18, 24, 19, 5]. However, acoustic sensors have low spatial and temporal resolutions compared to optical sensors. This means that, in general, they capture less details and scan at lower frequencies than modern cameras with high resolutions and fast frame rates. Thus, although the quality of images in sub-aquatic environments is strongly limited by the water and by the illumination conditions, in certain situations optical cameras offer more advantages than acoustic sensors [4]. Visual platforms are not really appropriate in the water column where it can be difficult to see the seabed or other reference points. However, for surveying or intervention applications, where the vehicle has to navigate relatively close to the sea bottom or it has to locate itself near the object to be manipulated, the use of cameras can certainly be a convenient option.

Lately, researchers are focusing their efforts on the enhancement of visual SLAM techniques (the use of cameras to perform SLAM) to be applicable in sub-aquatic environments and to be operative online, in missions conducted by real underwater vehicles.

1.2 Related Work

Visual SLAM in natural sub-aquatic scenarios has several inarguably difficulties not present in land: the light attenuation, flickering, scattering, the special nature of underwater environments with no man made structured frameworks, and the subsequent difficulty to define, find and track, reliable features or natural landmarks that can be used to match scenes visualized from different viewpoints and time instants.

The key of a successful underwater visual SLAM lies in the data association procedure to detect loop closings. This data association has to be robust under different viewpoints and illumination conditions. In the context of visual SLAM, the data association is also known as image registration. The image registration is in charge of recognizing scenes visualized by the robot from different viewpoints, in frames that have certain overlapping, and to compute the camera relative displacement between both views.

The literature is scarce in efficient visual SLAM solutions specially addressed to underwater and tested in field robotic systems. Most of these solutions particularize the approach commonly known as EKF-SLAM [7], correcting the dead-reckoning data with the results of an image registration process in a *Extended Kalman Filter* (EKF) context. These systems normally incorporate newly observed visual landmarks in a state vector that contains also the vehicle pose and velocity. One of the positive issues of this approach is the continuous correction of the vehicle and all the landmark poses contained in the filter at every iteration, which involves a simultaneous refinement of the vehicle trajectory and of the whole map. As a consequence, the filter running time increases with the size of the map, making the system not applicable on-line for long routes.

The same idea is used to locate an AUV equipped with a stereo camera with respect to a ship hull in [21]. In this work, 3D landmarks corresponding to points on the hull are computed from the stereo images. Similarly to [7], the filter state contains the vehicle pose and the observed landmarks.

Salvi et al [20] proposed a new method for underwater SLAM where the vector state is composed of the pose and the velocity of the vehicle given by a DVL, and the 3D pose of the successive detected landmarks computed with a stereo camera. The image registration process is used for the filter update and it is obtained by comparing new 3D landmarks with all those that are stored in the filter state. Previously to the filter execution, images are pre-processed to enhance their contrast and increase their brightness. proposed a new method for underwater SLAM where the vector state is composed of the pose and the velocity of the vehicle given by a DVL, and the 3D pose of the successive detected landmarks computed with a stereo camera. The image registration process is used for the filter update and it is obtained by comparing new 3D landmarks with all those that are stored in the filter state. Previously to the filter execution, images are pre-processed to enhance their contrast and increase their brightness.

Another concern for researchers has been how to make their approaches robust or immune to linearization errors inherent to EKF-based methods. To solve this problem, Aulinas *et al* implemented a submapping EKF-SLAM approach and tested it on an AUV, with highly convincing results [1].

A different alternative was proposed by Eustice *et al* with the *Delayed State Filtering* [10] approach: the state vector only contains the current vehicle pose,

its linear velocity, acceleration and the angular rate. The successive poses of the vehicle are predicted by dead reckoning and incorporated to the filter state. Images are taken at every position. Imagery overlapping provides pose constraints and image registration, which are used to define the observation function of the update stage. Since landmarks are not included in the filter state vector, the computational resources needed for every iteration are minor than in other approaches. However, the image registration process is still costly in time.

Other authors have focused the underwater visual SLAM problem from the graph-optimization or bundle adjustment point of view. Using these methods, the successive odometric poses of the vehicle, and, in some cases, the position of landmarks constitute the subsequent nodes of a graph linked by edges, which usually represent the distance from node to node. When a loop is closed, the complete graph is optimized, which means a complete graph adjustment entailing nodes (their labels) and distances between them [3]. This approach eludes the linearization errors, but graphs grow hugely with the amount of landmarks incorporated to the map thus increasing the computational resources needed.

Accordingly, this study presents a vision-based approach to perform underwater SLAM to accurately estimate the pose of an AUV. The proposal of this work is to integrate information coming from a monocular bottom-looking visual system, an altimeter, and a dead reckoning sensor.

While our research is close to that presented in [10], several distinctions with this inspiring work should be emphasised, mainly addressed to decrease the running time and the errors in the EKF results.

To reduce the error associated to the Kalman filtering process, we use a trajectory-based schema that includes in the filter state the successive vehicle displacements and rotations, instead of referring them to the global frame [6, 5].

Furthermore, the optimization of time and resources in the image registration process is tackled twofold. First, a unique and simple RANSAC-based algorithm filters out outlier correspondences and simultaneously computes the relative roto-translation between the evaluated frames. Second, we execute this image registration process only between images corresponding to locations that are within a fixed search radius, skipping the overlapping verification process proposed in [10]. Results exposed in forthcoming sections are aimed to validate the benefits of these design decisions.

The main advantages and contributions of our proposal are summarized next:

1. It is simple and fast, and requires less computational resources than previous solutions. The state vector does not include the landmarks, so the complexity of the image registration process is reduced without loosing robustness and accuracy in the loop closing determination procedure.

2. Detecting loop closings properly is extremely important as they provide valuable information to the SLAM process. Since the proposal presented here uses external altitude information, it is not constrained to constant altitude missions. In this way, the proposed image registration method is able to deal with translation, rotation and scale changes.

3. Our approach to SLAM adopts a *Trajectory Based* schema [6], in order to reduce estimation errors and computational complexity.

4. Finally, the approach has been assessed with an EKF and with an *Iterated Extended Kalman Filter* (IEKF) [2] to evaluate the convenience of using IEKFs instead of EKFs to reduce the linearization errors.

The system implementation has been tested on real underwater robots in aquatic environments, giving conclusive results.

The paper is structured as follows: section 2 explains the data association and image registration procedure used to detect loop closings; section 3 details the design and the structure of the EKF used to perform the visual SLAM; section 4 shows extensive experimentation that validates our approach and, finally, section 5 concludes the paper and outlines some forthcoming work.

2 IMAGE REGISTRATION

In SLAM, data association refers to the registration of current sensory input to previously gathered data. This process permits to identify parts of the environment already visited by the robot. Registering successfully such pieces of information is essential to perform loop closures, which impose several pose constraints that increase accuracy in the incremental localization process.

When using vision sensors, data association is tightly related to image registration. Image registration consists in overlaying several images of the same scene or part of the scene, taken at different times, and from different view points. The goal of the image registration process is to verify if there exists total or partial frame coincidence, and in case there is, to measure the relative motion of the camera between the two points at which both frames were taken.

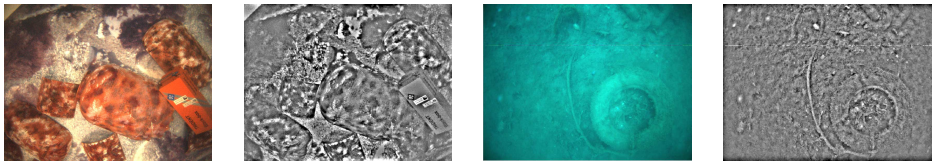
Image registration usually relies on the detection and matching of image features. If two frames represent, totally or partially, the same scene, features corresponding to coincident parts of that scene, should present, to a certain extend, similar descriptors. This statement depends on multiple conditions, mainly: changes on view point, scale or position, illumination conditions, brightness or contrast.

In consequence, applying special attention to the image registration process is fundamental to get accurate pose estimates underwater.

Given two images, our proposal to data association starts by searching their features and descriptors according to *Scale Invariant Feature Transform* (SIFT) [17]. Although other feature detectors and matchers can also be used, SIFT has been choosen for the first set of experiments because its invariance to changes on translation, rotation, scale and to illumination conditions. Furthermore, they provide sufficient number of putative correspondences for loop closing, increasing the robustness of the registration process [10].

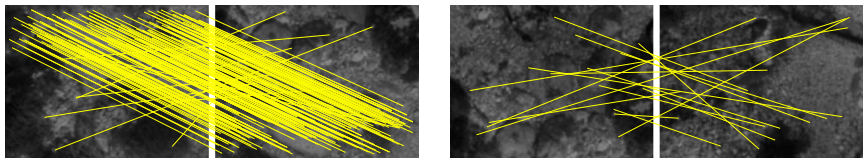
Due to the nature of the aquatic environments where our robots have to operate, an image preprocessing algorithm is recommended to enhance contrast and thus to improve the feature detection and matching. Here, images are filtered in the frequency domain using a Butterworth low pass filter. See in figure 1 two examples of underwater scenes, unfiltered ((a),(c)) and filtered with the low pass filter ((b),(d)). The image of figure 1-(a) was taken in a pool and the one on figure 1-(c) was taken in the sea. Section 4 shows a comparison of the SLAM results with and without filtering.

Feature coordinates, which are found in pixels, are then converted to meters, assuming a locally flat floor and that the distance to the bottom and the camera



(a) (b) (c) (d)

Figure 1: Image processing previous to the image registration. (a) and (c) original images, (b) and (d) filtered images



(a) (b)

Figure 2: Feature matching using underwater images. Yellow lines represent correspondences between features. (a) Overlapping images (b) Non overlapping images.

focal length are known. The former can be measured with an altimeter and the latter through a camera calibration process. Thanks to this, changes on the vehicle altitude which are responsible for scale changes between images, can be properly taken into account.

The next step towards the image registration is to compute feature matchings between the two images currently involved. In underwater scenarios it is very likely to obtain wrong correspondences due to problems inherent to this media, for instance: bad illumination, blur, scatter, untextured seafloor or excessive texture in the bottom, or the fact that, in certain scenes most of the gathered images look similar.

Figure 2-a exemplifies a common situation where there exists overlapping between two images. The majority of the features are matched correctly, but there are still some wrong associations. Although they represent a small percentage of the total number of matchings, these outliers distort the registration result. Also, SIFT, as well as many other feature matchers, are likely to detect matchings even between images corresponding to non overlapping areas, as illustrated in Figure 2-b. Wrong image associations can cause wrong loop closings and, as a consequence, unrecoverable errors in the SLAM process.

In order to find a model where inliers fit and outliers are discarded, a method based on RANSAC([11]) has been used for image registration. The key aspect of the data association is to determine whether two images overlap or not and, if they do, compute the roto-translation that better explains the correct overlay between them. Our proposal is based on the following premise: correct matchings tend to propose a single roto-translation whilst incorrect matchings do not and thus can be considered outliers.

Algorithm 1 shows the proposed procedure to compute the roto-translation between two underwater images using RANSAC. The symbol $\oplus$ denotes the compounding operator, as described in [22]. Roughly speaking, this algorithm randomly selects a subset C of feature matchings M and then computes the

Algorithm 1: RANSAC Image Registration

Input:
F_{ref}: Features $\{p_1, p_2, \cdots, p_m\}$ in the first image
F_{cur}: Features $\{q_1, q_2, \cdots, q_n\}$ in the second image
M: Matchings $M = \{(i,j) | visual\_matching(p_i, q_j)\}$
$nIter$: Number of iterations to perform
N: Number of matchings to be randomly selected
α: Maximum allowable error per matching
β: Min. number of selected matches to consider a model

Output:
X_{best}: The estimated roto-translation
ε_{best}: The error of the estimated roto-translation
$found$: Boolean stating if reliable matching found

Algorithm:

```
begin
    k ←0 ; ε_best ← ∞ ; found ← false;
    while k < nIter do
        C ← random selection of N items from M;
        (X, ε) ← find_motion(F_ref, F_cur, C);
        foreach (i, j) ∈ (M − C) do
            if ‖p_i − X ⊕ q_j‖ < α then
                C ← C ∪ {(i, j)};
        if |C| > β then
            (X, ε) ← find_motion(F_ref, F_cur, C);
            if ε < ε_best then
                ε_best ← ε ; X_best ← X ; found ← true;
        k ← k + 1;
```

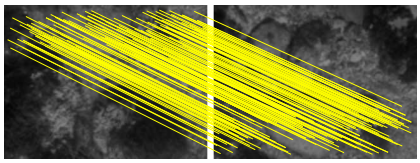


Figure 3: RANSAC underwater image registration

roto-translation $X = [x, y, \theta]^T$ that better explains them, under the assumption of a local planar motion. This assumption is perfectly acceptable in many common surveying missions where AUVs have to move parallel to the seabed which is formed by sand, rocks, algae and with no relevant relief. In this case, image points will not correspond exactly to coplanar points on the scene, but in practice they can be considered to do, if the lens axis is perpendicular to the bottom and height of the camera is much greater than the height of the elements lying on the seabed.

Next, each of the non-selected matchings is tested to check if it fits X with an acceptable error level. If so, it is selected too. Finally, if the number of selected matchings $|C|$ exceeds a certain threshold, the roto-translation that better explains all the selected matchings is computed. After a fixed number of iterations, the best of the computed roto-translations constitutes the output of the algorithm, and those correspondences that can not be related through this transformation with an acceptable error are considered to be outliers. If not enough matchings have been selected in any of the iterations, the algorithm assumes that the two images do not overlap.

Compared to other methods where the camera motion between two images is computed after the filtering of outliers, this algorithm is able to discriminate outliers from inliers while it computes the camera transformation, simplifying and speeding up the whole process

The algorithm relies on the so called $find\_motion$ function, which takes a set of feature matchings C and their coordinates in the first (F_{ref}) and in the second image (F_{cur}) as inputs. This function provides the roto-translation X that better explains the overlap between the images by searching the values of x, y, and θ that minimize the sum of squared distances between the matchings in C. More specifically, the roto-translation X and the associated error ε are computed as follows:

$$X = \underset{x}{\operatorname{argmin}}\, f(x) \tag{1}$$

$$\varepsilon = f(X) \tag{2}$$

being

$$f(x) = \sum_{\forall (i,j) \in C} ||p_i - x \oplus q_j||^2 \tag{3}$$

where p_i and q_j are feature coordinates in F_{ref} and F_{cur} respectively.

As an example, figure 3 shows the feature correspondences after applying our proposal to the images previously shown in figure 2-a. It can be seen how the wrong correspondences have been rejected and only those explaining the

true motion remain. Our proposal has also been applied to the images in figure 2-b, determining correctly the lack of overlap.

3 VISUAL SLAM

Being based on EKF-SLAM, our approach performs three main steps: prediction, state augmentation and update. During the prediction, the robot pose is estimated by means of dead reckoning. The state augmentation is in charge of storing the newly acquired information. Finally, the measurement step updates the prediction by associating the current image to previously stored data using the data association algorithm described in section 2.

Our proposal is to perform the measurement update using only one every N frames and thus reducing the computational cost. Henceforth, the used frame will be called a *keyframe* and N will be referred to as the *keyframe separation*.

In this study, similarly to the Trajectory-Based schema, the state vector X_k is defined as follows:

$$X_k = [x_1^0, x_2^1, x_3^2, \cdots, x_k^{k-1}]^T \tag{4}$$

where each $x_i^{i-1} (2 \leq i \leq k)$ denotes a roto-translation from keyframe F_{i-1} to keyframe F_i and x_1^0 represents the initial robot pose relative to a world fixed coordinate frame. Let us assume, without loss of generality, that $x_1^0 = [0, 0, 0]^T$. Thus, contrarily to other EKF Visual SLAM methods where the visual features themselves are stored in the state vector, our proposal requires much less computational resources because it stores only the motion estimates between keyframes.

The pose of the most recent keyframe with respect to the world fixed coordinate frame can be computed as $x_k^0 = x_1^0 \oplus x_2^1 \oplus x_3^2 \oplus \cdots \oplus x_k^{k-1}$. Also, the current robot pose can be computed by composing the last keyframe pose estimate and the dead reckoning information.

3.1 Prediction and state augmentation

Under the assumption of static environment, the state vector does not change during the EKF prediction step. However, it has to be augmented as follows when a new keyframe is available.

$$X_k^- = [X_{k-1}^-, x_k^{k-1}]^T \tag{5}$$

,where X_k^- is the predicted state vector and x_k^{k-1} is the motion estimate provided by the dead reckoning sensors. From a practical point of view and in order to take advantage of the cameras, a visual odometer was used in the experiments conducted with the robot. Details are given further in section 4.

Keyframes are also stored outside the state vector.

3.2 The update step

3.2.1 Image Overlapping

In order to detect loop closings, every time a new keyframe is gathered, it could be compared with all the previous ones using the image registration algorithm

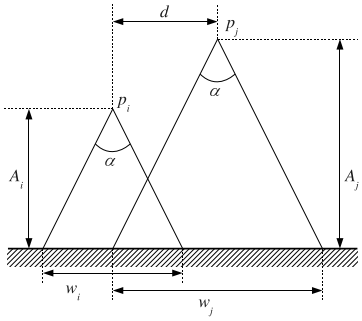


Figure 4: Simple camera model to determine whether two images overlap or not. Given two images gathered at times t_i and t_j and heights A_i and A_j using a camera with an angle of vision of α degrees, the observed regions have a diameter of w_i and w_j respectively. The term d denotes the distance between the image acquisition points.

proposed in section 2. However, performing such exhaustive test at every filter iteration can be extremely time consuming.

Therefore, computing the image registration process only on images that really present an acceptable overlap and discarding those that do not, would save time and resources and increase the accuracy in the matching process. Different approaches can be found in the literature concerning this issue [8].

One way to evaluate the degree of overlapping between two images is considering only pure geometrical issues. Similarly to [10], the camera field of view can be modeled as a cone. Under this assumption, the region of the sea bottom observed by the camera is a circle whose radius depends on the lens field of view and the height at which the image is gathered.

Being the field of view constant, the observed region basically depends on the camera's height when the image is obtained. Accordingly, it can be decided whether two images overlap or not using the height information and the position at which they were gathered. This idea is illustrated in Figure 4.

It is easy to see that the diameter of the observed region is as follows:

$$w_k = 2 \cdot A_k \cdot \tan(\frac{\alpha}{2}) \quad (k = i, j) \tag{6}$$

Two images gathered at times t_i and t_j can overlap if the following condition is satisfied:

$$||p_i - p_j|| \leq d_{max} = \frac{w_i}{2} + \frac{w_j}{2} \tag{7}$$

where p_i and p_j denote the camera position at times t_i and t_j respectively, and can be taken from the state vector.

In consequence, the image registration process between the current image and all the previous ones can be done only in case the condition of equation 7 is fulfilled.

As Equation 7 only depends on the positions and does not involve any image analysis, it is fast to compute.

Notice that d_{max} should be modified depending on the position uncertainties. Although doing so will lead to a more accurate search radius, it would increase the computation time while the corrective effects would be almost negligible.

If the robot is moving at a constant height A_n, then d_{max} is constant: $d_{max} = A_n \cdot \tan(\frac{\alpha}{2})$, and equation 7 can be reformulated as:

$$||p_i - p_j|| = d_f \leq d_{max} \tag{8}$$

where d_f denotes a distance threshold, always smaller than d_{max}, used to decide whether there is image overlapping or not.

Missions like ours, where robots have to survey an area for mapping, object detection or intervention are quite common to be performed at constant height. Although, in practice, controllers do not keep the vehicle at exactly the same altitude, for all practical purposes, it can be considered that they do, if the mean altitude at which the camera is working is much higher than the altitude oscillations and the mean height of the seabed relief. This is valid in all the environments where our system has to operate, that is in enclosed environments, shallow waters or coastal areas where the sea bottom is formed by small rocks, algae and sand.

Using a threshold for $||p_i - p_j||$ simplifies the approach since, it permits to do the registration of the current image directly with all the rest of images that are closer than d_f, thus avoiding the process of evaluating unlikely overlaps. The challenging point now is to determine the optimum value for d_f to get the maximum number of loop closings with the minimum quantity of outliers in the majority of the compared image pairs. As this value depends on the height and on the lens field of view, it has to be adjusted in every mission and at every different environment. Section 4 details the experimental process followed to find the optimum value of d_f and a quantification of the saved computational resources.

3.2.2 Data Associations as Measurement Vector

The data association procedure is in charge of evaluating if two images contain elements of the same scene, although they have been taken from different points of view. Scene coincidence normally entails coincidence in some set of features. If two images overlap, the data association procedure provides an estimate of the roto-translation between them.

This information is used to build our measurement vector Z_k:

$$Z_k = [(z_k^{C1})^T, (z_k^{C2})^T, \cdots, (z_k^{Cn})^T]^T \tag{9}$$

where $C1, C2, \cdots, Cn$ denote the keyframes that match the current one and z_k^{Ci} represents the motion estimated by our RANSAC based approach from the keyframe C_i to the most recent one.

In EKF-SLAM, the observation function h_i is in charge of telling how z_k^{Ci} is expected to be according to the state vector X_k^-. Because of the state vector format, this can be computed as follows:

$$h_i(X_k^-) = x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_k^{k-1} \tag{10}$$

Figure 5 illustrates the idea of a measurement z_k^{Ci} and the associated observation function h_i.

The observation matrix H_i is as follows:

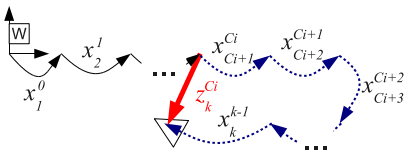


Figure 5: Illustration of a measurement (thick red arrow) and the corresponding observation function (dashed blue arrows)

$$H_i = \left.\frac{\partial h_i}{\partial X_k}\right|_{X_k^-} = \left[\left.\frac{\partial h_i}{\partial x_1^0}\right|_{X_k^-} \left.\frac{\partial h_i}{\partial x_2^1}\right|_{X_k^-} ... \left.\frac{\partial h_i}{\partial x_k^{k-1}}\right|_{X_k^-} \right] \tag{11}$$

It is straightforward to see that

$$H_i = \left[\underbrace{\begin{matrix} 000 \\ 000 \\ 000 \end{matrix}}_{\times Ci} \left.\frac{\partial h_i}{\partial x_{Ci+1}^{Ci}}\right|_{X_k^-} \left.\frac{\partial h_i}{\partial x_{Ci+2}^{Ci+1}}\right|_{X_k^-} ... \left.\frac{\partial h_i}{\partial x_k^{k-1}}\right|_{X_k^-} \right] \tag{12}$$

By applying the chain rule, the non-zero terms of this Equation are as follows:

$$\begin{aligned} \left.\frac{\partial h_i}{\partial x_j^{j-1}}\right|_{X_k^-} = & \left.\frac{\partial h_i}{\partial x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_j^{j-1}}\right|_{X_k^-} \cdot \\ & \cdot \left.\frac{\partial x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_j^{j-1}}{\partial x_j^{j-1}}\right|_{X_k^-} \end{aligned} \tag{13}$$

According to [9] this can be computed as follows:

$$\left.\frac{\partial h_i}{\partial x_j^{j-1}}\right|_{X_k^-} = J_{1\oplus}\{g_j, \ominus g_j \oplus h_i\}|_{X_k^-} \cdot J_{2\oplus}\{g_j \ominus x_j, x_j\}|_{X_k^-} \tag{14}$$

where $J_{1\oplus}$ and $J_{2\oplus}$ are the Jacobians of the composition of transformations [22] and

$$g_j = x_{Ci+1}^{Ci} \oplus x_{Ci+2}^{Ci+1} \oplus ... \oplus x_j^{j-1} \tag{15}$$

At this point, the full observation function h and the full observation matrix H considering all the matched keyframes are as follows:

$$h(X_k^-) = \begin{bmatrix} h_1 \\ h_2 \\ ... \\ h_n \end{bmatrix} \quad H = \begin{bmatrix} H_1 \\ H_2 \\ ... \\ H_n \end{bmatrix} \tag{16}$$

In few words, the observation function estimates the relative position between two overlapping frames composing all the intermediate displacements stored in the state vector in successive iterations. Also, the measurement vector stores the relative position between the same overlapped frames directly

obtained from the image registration algorithm. The difference between both values, which is the so called filter innovation, is the measure used by the Kalman filter to improve the trajectory.

It is worth to emphasize that, for each pair of registered images, the whole portion of the trajectory that connects them is explicitly corrected, contrarily to traditional methods that only explicitly correct the endpoints. For example, all the robot motions depicted as dashed blue arrows in figure 5 will be corrected by the single measurement $z_k^{C_i}$.

At this point, the standard EKF update equations, which basically depend on the observation function and the measurement vector, can be used.

In order to reduce the linearization errors an IEKF [6] [2] can be used instead a classic EKF. Roughly speaking, the IEKF consists on iterating an EKF and relinearizing the system at each iteration until convergence is achieved. When the IEKF achieves convergence, the state vector in the last iteration constitutes the updated state X_k^+.

Section 4 shows and analyzes the results obtained by an implementation of this SLAM approach using an EKF and an IEKF.

4 EXPERIMENTAL RESULTS

In order to show the validity of our proposal, some image sequences were recorded in diverse conditions using a simulated and a real robot. Later our algorithms were run off-line on these recordings.

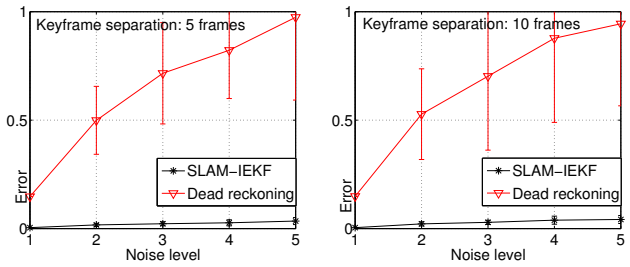
4.1 Experiments with a Simulated Environment

For the simulated experiments the underwater robot simulator UWSim [23] was used. The environment where the simulated robot was deployed consisted of a mosaic of a real sub-sea environment. Pictures shown in Figure 2 are examples of the imagery gathered by the simulated underwater camera.

The simulated mission consisted in performing a sweeping task. During the mission execution, images obtained from a monocular bottom looking camera were gathered. The robot pose was also recorded but solely used as ground truth. Altitude was constant in these simulations. The visual odometry was computed in 2D through the homography that transforms image features inter frames.

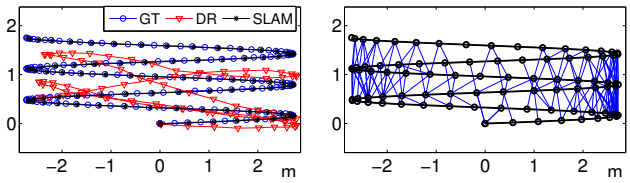
Tests were performed with two different keyframe separations, 5 and 10 and using an IEKF instead of an EKF, to minimize linearization errors. With the configuration of the simulated environment particularly set for these tests, running the algorithm with a separation of 5 frames means, in the straight parts of the trajectory, an overlap between consecutive keyframes of 55% of the image. A separation of 10 frames leads to an overlap close to a 10%.

In order to test the robustness of our approach in front of the drift accumulated in the visual odometry estimations, we added synthetic noise to the odometry data. Five noise levels were tested for each keyframe separation. The noise used is additive zero mean Gaussian and the covariance ranges from a $[\Sigma_x, \Sigma_y, \Sigma_\theta] = [0, 0, 0]$ (noise level 1) to $[\Sigma_x, \Sigma_y, \Sigma_\theta] = [4 \cdot 10^{-5}, 4 \cdot 10^{-5}, 5 \cdot 10^{-4}]$ (noise level 5). The random noise was added to each visual odometry estimate. For each configuration (5 or 10 frames of separation between keyframes) and



(a) (b)

Figure 6: Errors in meters and 2σ bound. (a) Using keyframe separation of 5. (b) Using keyframe separation of 10.



(a) (b)

Figure 7: Example of the results obtained with noise level 2 and keyframe separation 10. GT and DR denote Ground Truth and Dead Reckoning. (a) Trajectories (b) Registered images

noise level, 100 trials have been performed in order to obtain significant statistical results. The resulting SLAM trajectories have been compared to the ground truth in order to quantitatively measure their error. The error of a SLAM trajectory is computed as the mean distance between each of the SLAM estimates and the corresponding ground truth pose.

The results obtained when using a keyframe separation of 5 are shown in Figure 6-a and those obtained using a keyframe separation of 10 are depicted in Figure 6-b. It can be observed that the SLAM error is significantly below the error in dead reckoning. It is clear that the differences due to the keyframe separation and the noise level are very small. Thus, these experiments suggest that our proposal leads to pose estimates whose quality is nearly unrelated to the dead reckoning noise and to the keyframe separation, as long as the overlap between consecutive keyframes is sufficient.

Also, it is remarkable that the error covariances, which are shown as 2σ bounds in Figure 6, are small and significantly lower than those of dead reckoning. That is, even if very different dead reckoning trajectories are used, the SLAM results are very close to the ground truth.

Figure 7-a shows an example of the results obtained with noise level 2 and a keyframe separation of 10. The figure shows the resulting SLAM trajectory, which is almost identical to the ground truth. This is especially remarkable taking into account that the starting dead reckoning data, as it can be seen, is strongly disturbed by noise. Figure 7-b depicts the data associations that have been performed during the SLAM operation.

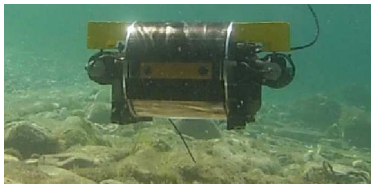


Figure 8: The Fugu-C

4.2 Experiments in a Water Tank

4.2.1 Experimental Setup

Experiments in aquatic environments were conducted with the Fugu-C platform (Figure 8). Fugu-C is a low-cost mini-AUV developed at the University of the Balearic Islands. The sensor suit for this vehicle includes two stereo rigs, one looking forward and another one looking downward, a MEMS-based Inertial Measurement Unit and a pressure sensor. Even so, only the information gathered by the down-looking camera was used for the experiments described below.

In order to feed our SLAM approach with odometric visual information, and considering that the robot moves in 3D and its visual equipment, two off-the-shelf stereo visual odometers, LibViso2 [12] and Fovis [14] were assessed and compared to be used in our experiments with the robot.

These two approaches were initially selected because of three main reasons:

1. Both systems are based on similar principles and are perfectly suitable for real-time stereo vision-based applications. They simplify the feature detection and tracking process, accelerating the overall procedure and minimizing the number of failures. Both algorithms have been tested in real platforms with high dynamics, such as cars and aerial vehicles.

2. A pure stereo-3D process is used to estimate motion in 6DOF.

3. The large amount of feature matchings makes it possible to deal with high resolution images, which is especially important for an stereo odometer.

By experimentally evaluating both odometers in undersea conditions, we observed that LibViso2 translation errors were smaller than those of Fovis. Also, both odometers provided rotation errors below $0.008^o/m$ [25]. As a consequence of these assessment, LibViso2 was used as the visual odometer in these experiments. The LibViso2 motion estimates in the x-y plane constitute our 2D odometric data and the z position estimates provide the height information. Furthermore, the pressure sensor was used to correct the drift in z caused by the odometry. Both odometric data and corrected height were provided at 10Hz.

It is worth to emphasize that we use stereo odometry due to the limited sensor suit of our robot. Of course, the described methodology can be reproduced using other odometers such as a DVL, if available.

The first experiments with the robot were conducted in a pool 7 meters long, 4 meters wide and 1.5 meters depth, whose bottom was covered with a printed digital image of a real seabed. In order to obtain a ground truth in this

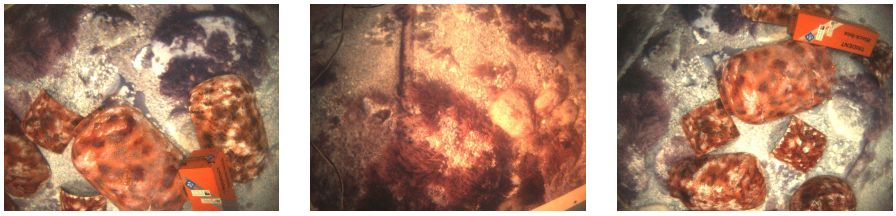
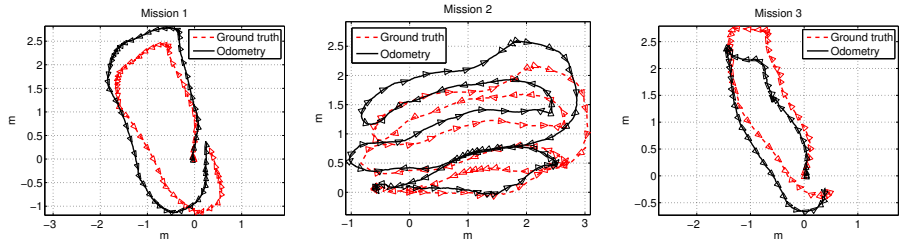


Figure 9: Examples of images obtained during the experiments



(a) (b) (c)

Figure 10: Ground truth and odometry corresponding to (a) first mission, (b) second mission and (c) third mission.

environment, each gathered image was registered to the whole printed digital image, which was previously known.

In this environment, three missions were executed. The first mission consisted in a single loop, the second mission was a sweeping trajectory and the third one was also a single loop. However, prior to the execution of the third mission, several objects such as amphoras and rock replicas were deployed inside the pool in order to simulate a realistic, non flat, sea floor. Figure 9 shows some examples of the imagery gathered during the third mission. Figures 10-a, 10-b and 10-c show the ground truth and the visual odometry corresponding to the first, second and third missions, respectively. It can be observed that, although visual odometry properly approximates the overall trajectory, there is also a significant drift error.

4.2.2 Tuning the Search Radius

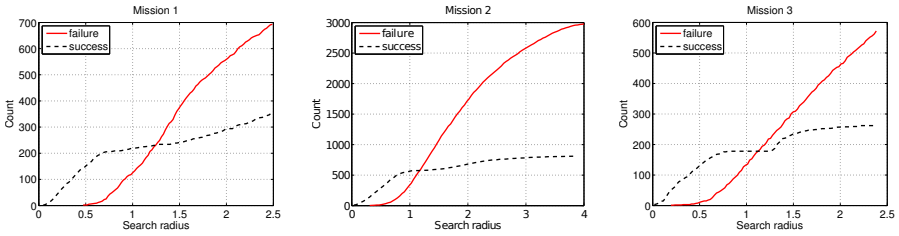
As stated in Section 3.2.1, deciding which of the gathered images may overlap with the current one is a crucial issue to save execution time. Although RANSAC would reject two non-overlapping images, such rejection is time consuming. Thus, it is important to feed RANSAC only with images that are likely to overlap and avoid unnecessary computation.

According to Equation 8, the selection of candidate overlapping images can be performed using a fixed search radius d_f. In this way, given the current image, only those whose estimated position is within the search radius are subsequently tested using RANSAC. As it was explained in Section 3.2.1, using a constant value is reasonable in surveying missions as they tend to be executed at a constant altitude.

In order to tune d_f for our experiments we computed the theoretical radius

Mission	min	max	mean	std
1	2.33	2.64	2.51	0.03
2	4.02	4.47	4.27	0.05
3	2.06	2.62	2.36	0.1

Table 1: Minimum, maximum, mean and standard deviation of d_{max} for each of the three missions. Data are expressed in meters.



(a) (b) (c)

Figure 11: Count of RANSAC failures and successes depending on the search radius for (a) mission 1, (b) mission 2 and (c) mission 3.

d_{max} of Equation 7 for every image pair that could be matched during the SLAM execution. This data was recorded for each of the three aforementioned missions. To do so the height information provided by the visual odometer and the pressure sensor was used.

Table 1 summarizes the results by showing the minimum and maximum values of d_{max}, as well as the mean and the standard deviation. Since in those experiments the robot navigated at an altitude with low variations, d_{max} values computed for all the possible image pairs are always in a very narrow range, reflected in a small standard deviation. The differences between missions are due to the differences in altitude between them.

According to this data, a good criteria to select a fixed search radius is to use the mean values. Thus, a first approximation is to set $d_f = 2.51m$ in the first mission, $d_f = 4.27m$ in the second mission and $d_f = 2.36m$ in the third one. However, this criteria tends to be too optimistic. In particular, even if two images actually overlap, the overlapping region may be too small or produce too few features for RANSAC to match them properly.

In order to obtain a more adequate value for d_f, we proceeded as follows. First, the three missions were performed using our SLAM proposal with the obtained mean values as d_f. Thus, the mean values constitute our initial guess. Every time RANSAC was executed, the estimated distance between the two compared images was recorded and labeled as a *success* or a *failure* depending on the RANSAC output: if RANSAC was able to find a roto-translation between the images we considered it a success whilst those cases in which RANSAC could not find such roto-translation were considered a failure. Failures are, precisely, the situations we want to avoid as they correspond to non overlapping images rejected by RANSAC, which is time consuming.

Using this information, the amount of successes and failures can be computed as a function of the distance between images. Also, it is clear that the number of failures and successes that will appear if a certain search radius is selected is the

Mission	True positive	False positive
1	64.03%	14.89%
2	77.13%	9.76%
3	56.49%	18%

Table 2: True and false positives for the three mission when using the proposed search radius d_f.

Mission	d_f	d_{max}	Improvement
1	186.58 s	726.84 s	74.33%
2	416.35 s	2243.74 s	81.44%
3	143.54 s	479.57 s	70.07%

Table 3: Execution times using the proposed search radius (d_f) and the theoretical one (d_{max}).

sum of failures and successes corresponding to all the distances lower or equal to the selected radius. Figure 11-a, 11-b and 11-c summarize this information for the first, second and third mission respectively. It can be observed how the number of failures increases with the search radius whilst the number of successes seems to stabilize from a certain search radius onward.

Our goal is to select a search radius for each mission so that the number of failures is reduced whilst the number of successes is as large as possible. According to the obtained data, the optimal search radius is 0.6m during the first mission and 0.7m during the second and third missions. Figure 11 shows clearly how these values correspond to the region where the number of failures is very low and the number of successes has reached a highly acceptable local maximum. Thus, henceforth during the first mission only those images whose distance to the current one is below $d_f = 0.6m$ will be analyzed by RANSAC. The same criteria will be applied during the second and third missions using $d_f = 0.7m$ in both cases.

In order to evaluate the selected values for d_f, we measured the number of true and false positives they produce. In this context, a true positive appears when a couple of images that RANSAC would not be able to match is discarded because of the search radius prior to the RANSAC execution. A false positive corresponds to the situation in which two images that RANSAC actually could match are discarded because of the search radius. In other words, a true positive appears when discarding a RANSAC fail and a false positive appears when discarding a RANSAC success. Table 2 summarizes the results.

For example, in the second mission the number of RANSAC executions is reduced a 86.89% (77.13 + 9.76). That is, for both, true and false positives, RANSAC is not executed, and only a 9.76% correspond to discarded images that should have been registered.

Table 3 shows the improvements in the execution time when performing SLAM using the proposed search radius d_f compared with the execution time when using the purely geometrical criteria d_{max}. The time was computed executing a Matlab implementation on an Intel Centrino 2 at 2.4GHz, with only one CPU kernel used, and running Ubuntu 10.04. The separation between keyframes was 30 frames.

It can be seen that the reduction in the running time is related to the percentage of skipped RANSAC executions shown in table 2.

It should be noticed that, although the process has been tested using a non optimized codification running on a regular computer, the execution time obtained for each mission, when d_f is used, is close (slightly above) to the real mission duration. For instance, the navigation time for mission 1 was 169 seconds while the whole SLAM process took 186.58 seconds. Thus, obtaining an on-line version is straightforward.

Although the experimental method to tune the search radius requires a training work each time the robot is deployed in an unknown environment, it is worthwhile performing the proposed approach taking into account the huge reduction in computation time.

Further experiments will analyze the accuracy of the pose estimates based on the d_f values obtained here.

4.2.3 Quantitative evaluation

As stated in section 4.2.1, three different missions have been conducted in a water tank. The first mission consisted in a single loop, the second one consisted in a sweeping trajectory and the third mission consisted in a single loop over a non-flat terrain. Both ground truth and visual odometry have been shown in Figure 10. In the three cases, a significant odometric error appears.

In order to provide a complete evaluation, the goal was to compare the quality of every main component of the present approach.

In our implementation, all the following combinations were easily configurable and interchangeable, allowing the achievement of the different results exposed later.

First, for each of the three missions, our approach has been tested using both, IEKF and EKF, in the update step. For each filter update method, the system has been tested using both, the images as they are provided by the camera and filtering them using a Butterworth low pass filter as suggested in section 2. For each of these configurations, three different keyframe separations have been tested: 20 and 30 frames to show the SLAM behavior in a realistic operation and 90 frames to push the system to its limits.

In addition, for each filter update, image treatment and keyframe separation, the visual odometry was corrupted with 5 different levels of additive zero mean Gaussian error. The covariance of this noise ranged from $[\Sigma_x, \Sigma_y, \Sigma_\theta] = [0, 0, 0]$ in noise level 1 to $[\Sigma_x, \Sigma_y, \Sigma_\theta] = [4 \cdot 10^{-5}, 4 \cdot 10^{-5}, 4 \cdot 10^{-4}]$ in noise level 5. For each of these cases, 50 trials were executed. This leads to a total of 9000 trials.

The error of each SLAM estimate in each trial was computed by comparing it to the corresponding ground truth pose. The error of each trial is defined as the mean error of the corresponding SLAM estimates. This error was finally divided by the true trajectory length of the corresponding mission, provided by the ground truth. In this way, the error units are meters of error per traveled meter. Thanks to this, the errors obtained for each of the three missions can be compared and also joined in order to obtain an overall measure of quality.

The first relevant observed results is that, in all cases, the statistical differences between keyframe separations of 20 and 30 are barely appreciable. This leads to a similar conclusion to the one obtained under simulation: as long as

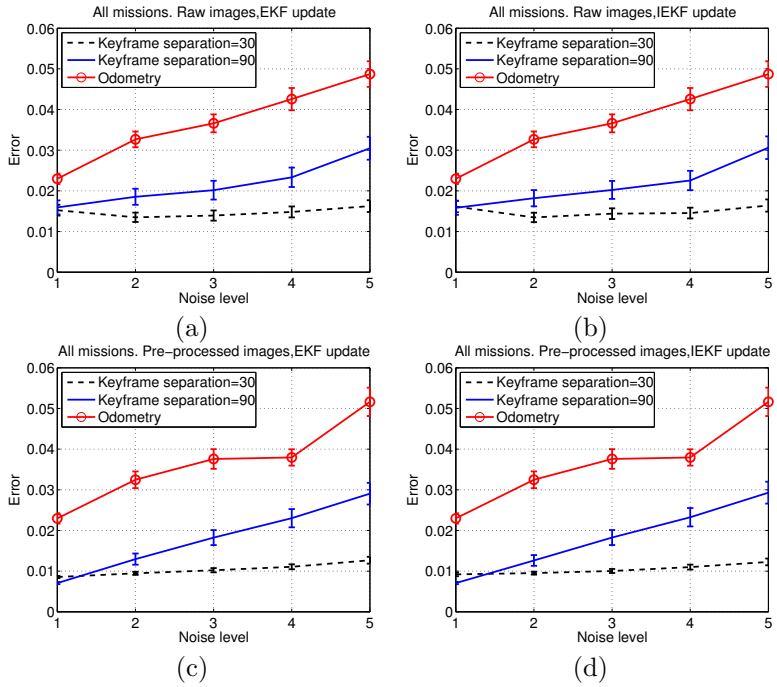
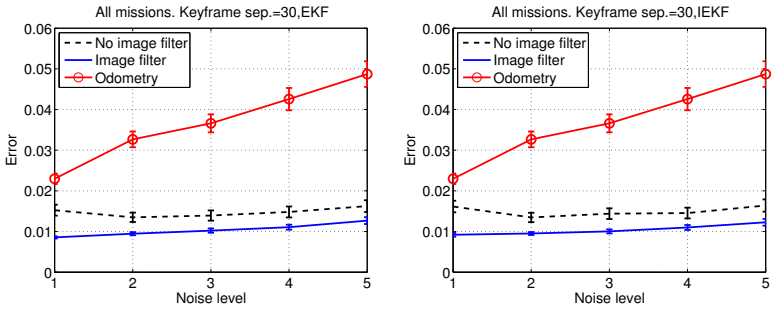


Figure 12: Mean and standard deviation of the errors corresponding to 30 and 90 keyframe separations for (a) raw images and EKF update, (b) raw images and IEKF update, (c) filtered images and EKF update and (d) filtered images and IEKF update. The standard deviation is depicted as 0.1σ to provide a clear representation.

sufficient overlap between consecutive images is provided, the quality of our proposal is scarcely influenced by the keyframe separation.

The results comparing keyframe separations of 30 and 90 are shown in Figure 12. All the aforedescribed test cases are shown. In all four cases it can be seen a significant improvement when using 30 frames instead of 90. Also, as the noise level increases, the error when using a separation of 30 frames barely increases, whilst using 90 frames leads to a growing error. Moreover, the standard deviation of the error remains almost constant when using 30 frames between SLAM executions, suggesting that even large differences between initial estimates, reflected by the large odometric covariance, lead to SLAM results close to the ground truth. Thus, using 30 frames instead of 90 provides a significant improvement in the pose estimates. Accordingly, henceforth the keyframe separation used during this quantitative evaluation will be 30 frames. However, either using 30 or 90 frames, the SLAM estimates provide an important improvement with respect to the stereo visual odometer.

Figure 12 also provides some insights regarding the other proposed SLAM components. For example, it can be observed in Figures 12-a and 12-b how the IEKF update and the EKF update provide similar results. The same can be observed when comparing Figures 12-c and 12-d. This suggests that, at least in these missions, the reduction of linearization errors thank's to the use of IEKF is nearly unobservable. Additionally, when comparing the results corresponding to filtered and non filtered images it becomes clear that image filtering actually



(a) (b)

Figure 13: Comparison between pose errors using raw images and filtered images, combined with an (a) EKF and (b) IEKF. The standard deviation is depicted as 0.1σ to provide a clear representation.

Noise level	1	2	3	4	5
Visual odometry	0.023	0.033	0.037	0.043	0.049
SLAM	0.008	0.009	0.01	0.011	0.013
Improvement	62.8%	71.0%	72.1%	74.0%	74.0%

Table 4: Comparison of errors in visual odometry and SLAM using a keyframe separation of 30, EKF update and filtered images. Errors are expressed in meters of error per travelled meter.

leads to an appreciable improvement in the accuracy of the pose estimation.

Figure 13 compares explicitly the errors obtained using raw images and filtered images combined with both, an EKF and an IEKF. It can be observed that filtering the images actually provides a significant improvement in terms of error reduction with respect to the results obtained using raw images. Comparing Figures 13-a and 13-b confirms that the use of an IEKF barely changes the results. Also, the error standard deviation corresponding to tests conducted with filtered images are smaller than those resulting from the use of non-filtered images.

In summary, the option that combines important reductions in running time with smallest errors in the pose estimates is using a keyframe separation of 30 frames, an EKF for the update step and a previous image filtering to enhance image contrast.

Table 4 summarizes the results by comparing the initial guess provided by the visual odometer and the SLAM output. The percentage of improvement is also shown.

4.2.4 Qualitative evaluation

Figures 14, 15 and 16 show some representative examples of the SLAM operation under different conditions for the three missions. In all cases, EKF update and filtered images were used.

Each figure shows, for its particular mission, the robot trajectory, estimated composing the odometry and the SLAM pose estimates of executions with 30 and 90 keyframes of separation with noise levels 1, 3 and 5. All plots show the

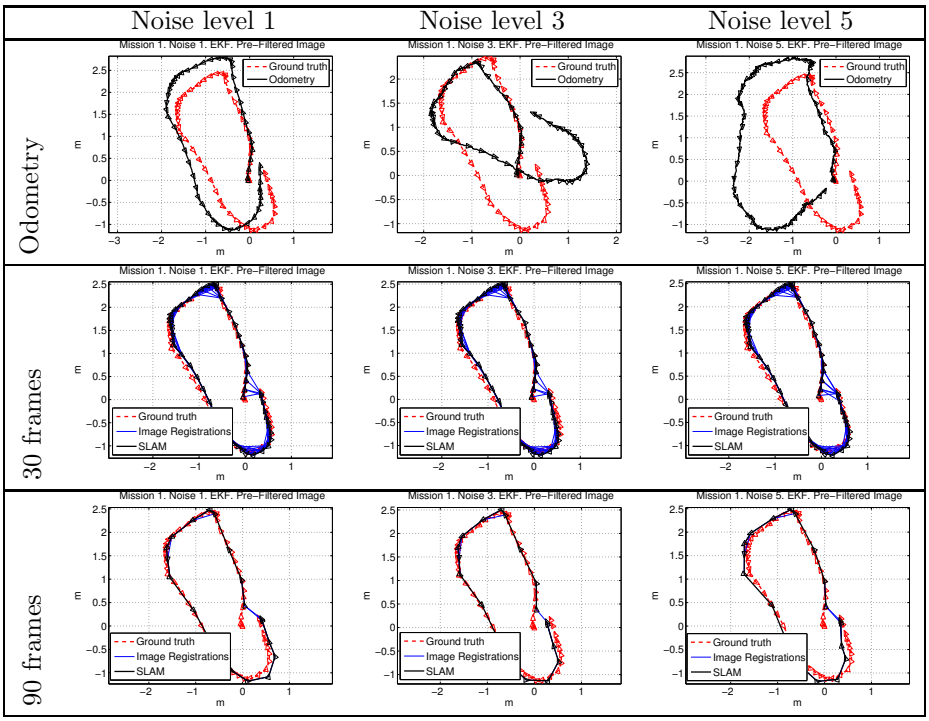


Figure 14: Example results corresponding to the first mission. The first row compares visual odometry and ground truth. Next rows correspond to different keyframe separations. First, second and third columns are related to noise levels of 1, 3 and 5 , respectively.

positive image registrations in blue and also incorporate the ground truth to facilitate its comparison with the resulting path. The robot is included in the representation as a triangle pointing towards the direction of motion.

It can be observed that, in the three missions, the final results are scarcely influenced by the initial conditions (i.e. the noise level).

4.3 Subsea Experiments

A final experiment was conducted in real undersea conditions, in Port de Valldemossa (Mallorca, Spain). Being a real environment, the floor was non flat, fully covered by stones and algae, and the robot motion was influenced by small currents and waves. Also, due to the small waves and the sun light, some minor flickering and shadows appeared in the images. Figure 17 shows some examples of the imagery gathered during this experiment.

Ground truth was not available. However, the desired mission was to perform an approximately eight shaped trajectory with the second loop larger than the first one, and ending at the same starting point. One artificial marker was placed on the seabed to assure that the endpoint of the trajectory corresponded with the initial point. The search radius was experimentally tuned to 1.4m.

Figure 18-a shows the obtained results using a keyframe separation of 20 frames. All plots show the positive image registrations in blue, the trajectory

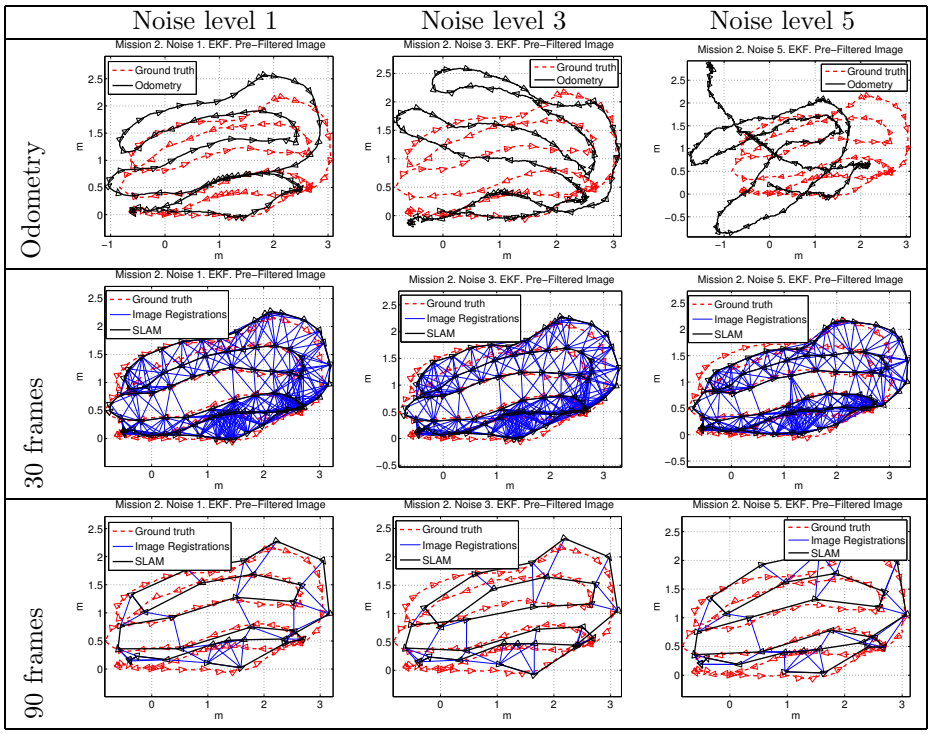


Figure 15: Example results corresponding to the second mission. The first row compares visual odometry and ground truth. Next rows correspond to different keyframe separations. First, second and third columns are related to noise levels of 1, 3 and 5, respectively.

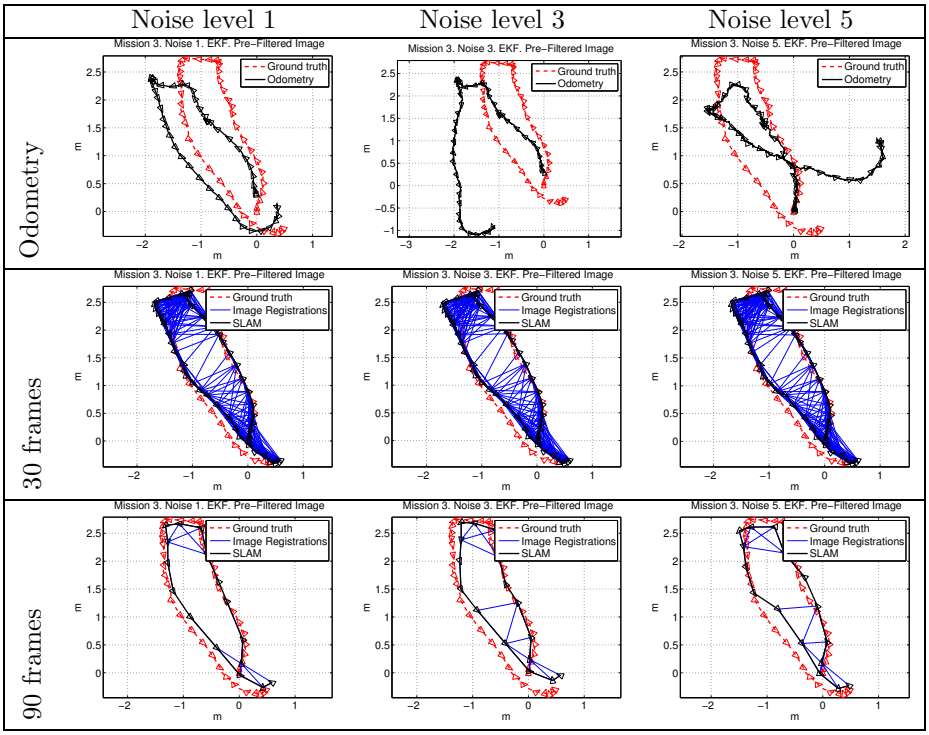


Figure 16: Example results corresponding to the third mission. The first row compares visual odometry and ground truth. Next rows correspond to different keyframe separations. First, second and third columns are related to noise levels of 1, 3 and 5, respectively.

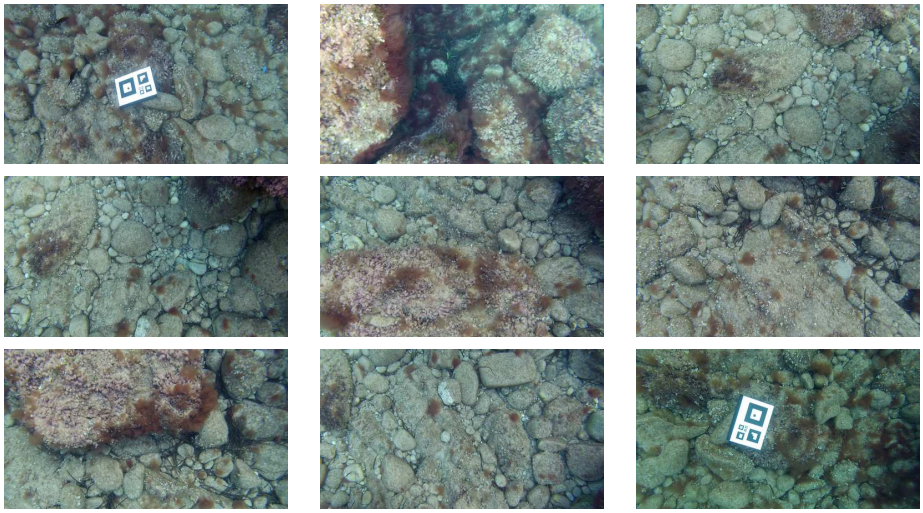
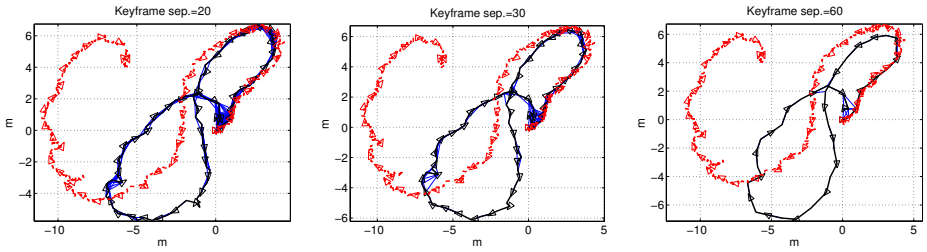


Figure 17: Some images gathered during the experiment in the sea, in Port de Valldemossa. The image on the first row-first column corresponds to the start of the trajectory and the image on the third row-third column corresponds to the end. The trajectory was performed at a constant depth.



(a) (b) (c)

Figure 18: Visual odometry (dashed red line) and SLAM (continuous black like) pose estimates in Port de Valldemossa using keyframe separations of (a) 20 frames, (b) 30 frames and (c) 60 frames

computed from the visual odometry in red and the SLAM trajectory in black. Notice how loop closings are found not only in the origin-end of the trajectory but also along it. Again, the robot is represented as a triangle, with one of its vertex pointing towards the direction of motion.

It can be observed how visual odometry presents an important drift in this scenario. To the contrary, the SLAM estimates are much closer to the real eight shaped trajectory and, thanks to the several loop closings established during the mission execution, the trajectory is considerably correct, ending at the same point where it started.

The same applies to Figure 18-b and Figure 18-c, where the results for separations of 30 frames and 60 frames are shown.

5 CONCLUSION AND FUTURE WORK

This paper proposes a simple and practical approach to perform underwater visual SLAM, which improves the traditional EKF-SLAM by reducing both the computational requirements and the linearization errors. Moreover, the focus of this paper is the image registration, which is used in the SLAM data association step, making it possible to close loops robustly. Thanks to that, as shown in the experiments, the presented approach provides accurate pose estimates both using a simulated robot and a real one, in controlled and in real underwater scenarios.

Nonetheless, the presented approach makes two assumptions that limit the environments where the robot can be deployed. On the one hand, it is assumed that the camera is always pointing downwards. Although this may seem a hard requirement, the experiments with the real robot show that the small oscillations in roll and pitch inherent to the robot motion are not significantly influencing the results of our approach. However, avoiding this requirement is one of our future research lines. The simplest way to solve this problem is to use the roll and pitch provided by the gyroscopes in the IMU and use this information to re-project the feature coordinates. On the other hand, the proposal presented here assumes a locally flat floor. Some experiments included in this paper show that real oceanic floors with no significant relief are well tolerated by the proposal presented here. However, incoming work is currently focused on using stereo data to overcome this restriction and to perform 3D SLAM.

Acknowledgments

This work is partially supported by the Spanish Ministry of Economy and Competitiveness under contracts PTA2011-05077 and DPI2011-27977-C03-02, FEDER Funding and by Govern Balear (Ref 71/2011).

References

[1] J. Aulinas, Y. Petillot, X. Llado, J. Salvi, and R. Garcia. Vision-based underwater slam for the sparus auv. In *Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries*, pages 171–180, 2011.

[2] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley and Sons Inc., 2001.

[3] C. Beall, F. Dellaert, I. Mahon, and S.B. Williams. Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys. In *Proceedings of Oceans*, Santander, Spain, June 2011.

[4] F. Bonin, A. Burguera, and G. Oliver. Imaging systems for advanced underwater vehicles. *Journal of Maritime Research*, 8(1):65–86, April 2011.

[5] A. Burguera, Y. González, and G. Oliver. Underwater slam with robocentric trajectory using a mechanically scanned imaging sonar. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, October 2011.

[6] A. Burguera, G. Oliver, and Y. González. Scan-based slam with trajectory correction in underwater environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, Taipei (Taiwan), 2010.

[7] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): part I. *IEEE Robotics and Automation Magazine*, 13(2):99–110, June 2006.

[8] A. Elibol, N. Gracias, and R. Garcia. Augmented state extended kalman filter combined framework for topology estimation in large-area underwater mapping. *Journal of Field Robotics*, 27(5):656–674, 2010.

[9] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, August 2005.

[10] R.M. Eustice, O. Pizarro, and H. Singh. Visually augmented navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 33(2):103–122, April 2008.

[11] 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*, 24(6):381–395, 1981.

[12] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[13] M. Hildebrandt and F. Kirchner. Imu-aided stereo visual odometry for ground-tracking auv applications. In *Proceedings of Oceans*, Sydney, Australia, May 2010.

[14] A.S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Proceedings of the International Symposium on Robotics Research (ISRR)*, Flagstaff (Arizona), USA), August 2011.

[15] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, Lisbon, Portugal, September 2006.

[16] P.M. Lee, S.M Kim, B.H. Jeon, H.T. Choi, and Ch.M. Lee. Improvement on an inertial-doppler navigation system of underwater vehicles using a complementary range sonar. In *Proceedings of the IEEE/RSJ International Symposium on Underwater Technology*, pages 133–138, 2004.

[17] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[18] P.M. Newman, J.J. Leonard, and R.J. Rikoski. Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In *Proceedings of the International Symposium on Robotics Research*, October 2003.

[19] D. Ribas, P. Ridao, and J. Neira. *Underwater SLAM for Structured Environments Using an Imaging Sonar*, volume 65 of *Springer Tracts in Advanced Robotics*. Springer, 2010.

[20] J. Salvi, Y. Petillot, and E. Batlle. Visual slam for 3d large-scale seabed acquisition employing underwater vehicles. In *Proceedings of the International Conference on Intelligent Robots and Sytems*, pages 1011–1016, 2008.

[21] R. Schattschneider, G. Maurino, and W. Wang. Towards stereo vision slam based pose estimation for ship hull inspection. In *Proceedings of Oceans*, pages 1–8, Waikoloa, Hawaii, June 2011.

[22] R. Smith, P. Cheeseman, and M. Self. A stochastic map for uncertain spatial relationships. In *Proceedings of International Symposium on Robotic Research, MIT Press*, pages 467–474, 1987.

[23] UWSim. UWSim: The underwater simulator. Web. Accessed: 20-June-2013, 2013.

[24] S. Williams and I. Mahon. Simultaneous localisation and mapping on the great barrier reef. *Proceedings IEEE International Conference on Robotics and Automation*, 2:1771–1776 Vol.2, 26-May 1, 2004.

[25] S. Wirth, P.L. Negre Carrasco, and G. Oliver. Visual odometry for autonomous underwater vehicles. In *Proceedings of the IEEE Oceans*, Bergen, Norway, 2013.