Underwater Photogrammetry and Visual Odometry

Mohamad Motasem Nawaf, Jean-Philip Royer, Jerôme Pasquet, Djamal Merad and Pierre Drap

Aix-Marseille Université, CNRS, ENSAM, Université De Toulon, LSIS UMR 7296, Domaine Universitaire de Saint-Jérôme, Bâtiment Polytech, Avenue Escadrille Normandie-Niemen, 13397 Marseille, France

Abstract: We propose an improved visual odometry approach that is adapted to low computational resources systems in an underwater environment. The aim is to guide underwater photogrammetry surveys in real time. The visual odometry relies on stereo image stream that is captured by an embedded system. An improved pose estimation procedure underlying fast stereo matching approach is followed by a semi-global bundle adjustment. Computed trajectory is maintained stochastically and a divergence measure is used for more realistic optimization zone selection. In particular, we propose a new approach to find an approximation of the uncertainty for each estimated relative pose based on machine learning manifesting on simulated data. This allows the user to find potential overlaps in the estimated trajectory for better drifts handling and loop closure. The evaluation of the proposed method demonstrates the gain in terms of computation time w.r.t. other approaches. The built system opens promising areas for further development and integration of embedded vision techniques.

Keywords: photogrammetry; visual odometry; underwater imaging; embedded systems; probabilistic modelling

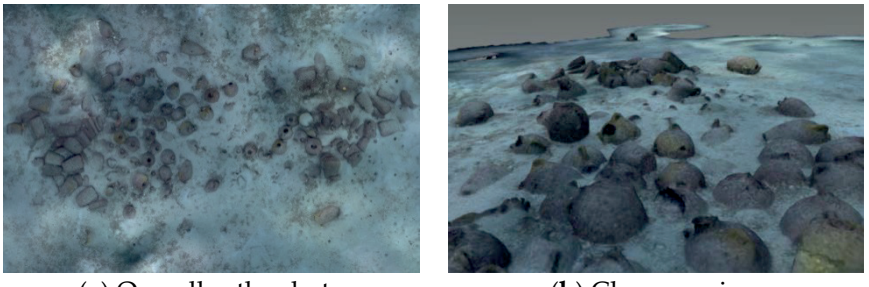
---

1. Introduction

Mobile systems nowadays undergo a growing need for self-localization to accurately determine its absolute/relative position over time. Despite the existence of very efficient technologies that can be used on-ground (indoor/outdoor) and in-air, such as Global Positioning System (GPS), optical, radio beacons, etc. However, in the underwater context most of these signals are jammed so that the corresponding techniques cannot be used. On the other side, solutions based on active acoustics, such as imaging sonars and Doppler Velocity Logs (DVL) devices remain expensive and require high technical skills for their deployment and operation. Moreover, their size specifications prevent their integration within small mobile systems or even being hand held. The research for an alternative is

ongoing, notably, the recent advances in embedded systems outcome relatively small, powerful, and cheap devices. This opens interesting perspectives to adapt a light visual odometry approach that provides relative path in real-time, this describes our main research direction. The developed solution is integrated within underwater archaeological site survey where it plays an important role to facilitate image acquisition. An example of targeted sites is shown in Figure 1.

In underwater survey tasks, mobile underwater vehicles (or divers) navigate over the target site to capture images. The obtained images are treated in a later phase to obtain various information and to also form a realistic three-dimensional (3D) model using photogrammetry techniques Drap (2012). In such a situation, the main problem is to totally cover the underwater site before ending the mission. Otherwise, we may obtain incomplete 3D models and the mission cost will raise significantly as further exploitation is needed. However, the absence of an overall view of the site especially under bad lighting conditions makes the scanning operation blind. In practice, this yields to over-scanning the site, which is a waste of time and cost. Moreover, the quality of the taken images may go below an acceptable limit. This mainly happens in terms of lightness and sharpness, which is often hard to quantify visually on the fly. In this work, we propose solutions for the aforementioned problems. Most importantly, we propose to guide the survey based on a visual odometry approach that runs on a distributed embedded system in real-time. The output ego-motion helps to guide the site scanning task by showing approximate scanned areas. Moreover, an overall subjective lightness and sharpness indicators are computed for each image to help the operator to control the image quality. Overall, we provide a complete hardware and software solution for the problem.



(**a**) Overall orthophoto (**b**) Close-up view

Figure 1. Example of a three-dimensional (3D) model of an underwater site; a Phoenician shipwreck located near Malta.

In common approaches of visual odometry, a significant part of the overall processing time is spent on feature points detection, description, and matching. In the tested baseline algorithm, the aforementioned operations represent ~65% of

processing time in case of local/relative bundle adjustment (BA) approach, which occupies in return the majority of the time left. In our proposed method, we rely on low level Harris based detection and template matching procedure, which significantly speeds up the feature matching speed. Further, whereas in traditional stereo matching the search for correspondence is done along the epipolar line within certain fixed range, in our method we proceed first by computing *a priori* rough depth belief based on image lightness and following the law of light divergence over distance. This is only valid for a configuration where the light source is fixed to the system, which is the case here. Hence, our first contribution is that we benefit from the rough depth estimation to limit points correspondence search zone to reduce processing time.

From another side, traditional visual odometry methods based on local BA suffers from rotation and translation drifts that grows with time Mouragnon et al. (2009). In contrary, the solutions based on using features from the entire image set, such as global BA Triggs et al. (2000), require more computational resources that are very limited in our case. Similarly, the simultaneous localization and mapping (SLAM) approaches, Thrun et al. (2005), which are known to perform good loop closure, are computationally intensive, especially when complex particle filters are used Montemerlo and Thrun (2007), and they can only operate in moderate size environments if real-time processing is needed. In our method, we adopt a semi-global approach Nawaf et al. (2016), which proceed in the same way as local method in optimizing a subset of image frames. However, it differs in the way of selecting the frames subset, as local methods use Euclidean distance and deterministic pose representation to select frames, ours represents the poses in a probabilistic manner, and uses a divergence measure to select such sub set.

The rest of the paper is organized as follows: We survey related works in Section 2. In Section 3 we describe the designed hardware platform that we used to implement our solution. Our proposed visual odometry method is explained in Section 4. The analytical results are verified through simulation experiments presented in Section 5. Finally, we present a summary and conclusions. We note that parts of this work have been presented in Nawaf et al. (2016) and Nawaf et al. (2017).

2. Related Works

2.1. Ego-Motion Estimation

Estimating the ego-motion of a mobile system is an old problem in computer vision. Two main categories of methods are developed in parallel, namely; simultaneous localization and mapping (SLAM) Davison (2003), and visual odometry Nistér et al. (2004). In the following, we highlight the main characteristics for both of the approaches.

SLAM family of methods uses probabilistic model to handle vehicle pose, although this kind of methods is developed to handle motion sensors and map landmarks, they work efficiently with visual information solely. In this case, a map of the environment is built, and at the same time it is used to deduce the relative pose, which is represented using probabilistic models. Several solutions to SLAM involve finding an appropriate representation for the observation model and motion model, while preserving efficient and consistent computation time. Most methods use additive Gaussian noise to handle the uncertainty which imposes using extended Kalman Filter (EKF) to solve the SLAM problem Davison (2003). In case of using visual features, computation time and used resources grows significantly for large environments. A remarkable improvement of SLAM is the FastSLAM approach Montemerlo and Thrun (2007), which improves largely the scalability, it uses recursive Monte Carlo sampling to directly represent the non-linear process model, although the state-space dimensions are reduce using Rao-Blackwellisation approach Blanco et al. (2008), the method remains not scalable to long autonomy. In the context of long trajectories, several solutions are proposed to handle relative map representations, such as Eade and Drummond (2008), Davison et al. (2007), Piniés and Tardós (2007). In particular, by breaking the estimation into smaller mapping regions, called sub-maps, then computing individual solutions for each sub-map. The issues with this kind of approaches arise in sub-mapping creation, overlapping, fusion of sub-maps, and map size selection, especially in our context where the S-shape scanning causes very frequent sub-maps switches, which is time consuming.

In all of the reviewed SLAM methods, the measurement noise is modeled by diagonal covariance matrix with equal values that are set empirically for the case of using pure visual information. This modeling leads to produce spherical measurement uncertainty (though estimated pose has an associated full degrees of freedom (DOF) uncertainty) in 3D when using only visual features. This does not approve with practical cases where uncertainty is not spherical. Although there exist several works in literature that studied the uncertainty of 3D reconstructed points based on their distance from the camera and the baseline distance between frames, such as in Eade and Drummond (2006) and Montiel et al. (2006), the effect of the relative motion parameters on the uncertainty of the pose estimation have not been taken into account. For a complete review for SLAM methods, we refer the reader to Bailey and Durrant-Whyte (2006).

From another side, visual odometry methods uses structure from motion methodology to estimate the relative motion Nistér et al. (2004). Based on multiple view geometry fundamentals, Hartley and Zisserman (2004), approximate relative pose can be estimated, this is followed by a BA procedure to minimize re-projection errors, which yields in improving the estimated structure. Fast and efficient BA approaches are proposed simultaneously to handle larger number of

images Lourakis and Argyros (2009). However, in case of long time navigation, the number of images increases dramatically and prevents applying global BA if real time performance is needed. Hence, several local BA approaches have been proposed to handle this problem. In local BA, a sliding window copes with motion and select a fixed number of frames to be considered for BA Mouragnon et al. (2009). This approach does not suit S-Type motion since the last n frames to the current frame are not necessarily the closest. Another local approach is the relative BA proposed in Sibley et al. (2009). Here, the map is represented as Riemannian manifold based graph with edges representing the potential connections between frames. The method selects the part of the graph where the BA will be applied by forming two regions, an active region that contains the frames with an average re-projection error changes by more than a threshold, and a static region that contains the frames that have common measurements with frames in active region. When performing BA, the static region frames are fixed, whereas active region frames are optimized. The main problem with this method is that distances between frames are metric, whereas the uncertainty is not considered when computing inter-frames distances.

Recently, a novel relative BA method is proposed by Nawaf et al. (2016). Particularly, an approximation of the uncertainty for each estimated relative pose is estimated using a machine learning approach manifesting on simulated data. Neighboring observations that are used for the semi-global optimization are established based on a probabilistic distance in the estimated trajectory map. This helps to find the frames with potential overlaps with the current frame, while being robust to estimation drifts. We found this method most adapted to our context.

2.2. Feature Points Matching

Common ego-motion estimation methods rely on feature points that are matching between several poses Nistér et al. (2004). The choice of the used approach for matching feature points depends on the context. For instance, features matching between freely taken images (six degrees of freedom), must be invariant to scale and rotation changes. Scale invariant feature descriptors (SIFT) Lowe (2004) and the Speeded Up Robust Features (SURF) Bay et al. (2006) are well used in this context Nawaf and Tremeau (2014). In this case, the search for a point's correspondence is done w.r.t. all of the points in the destination image.

In certain situations, some constraints can be imposed to facilitate the matching procedure. In particular, limiting the correspondence search zone. For instance, in case of pure forward motion, the focus of expansion (FOE), being a single point in the image, the search for the correspondence for a given point is limited to the epipolar line Yamaguchi et al. (2013). Similarly, in case of sparse stereo matching, the correspondence point lies on the same horizontal line in the

case of rectified stereo or on the epipolar line otherwise. This speeds up the matching procedure first by having less comparisons to perform, and second low-level features can be used Geiger et al. (2011). According to our knowledge there is no method that proposes an adaptive search range following a rough depth estimation from lightness in underwater imaging.

3. Hardware Platform

As mentioned earlier, we use an embedded system platform for our implementation. Being increasingly available and cheap, we choose the popular Raspberry Pi © (RPi) [1] as the main processing unit of our platform. This allows to run smoothly most of image processing and computer vision techniques. A description of the built system is shown in Figure 2, which is composed of two RPi's computers, where each is connected to one camera module to form a stereo pair. The cameras are synchronized using a hardware trigger. Both computers are connected to one more powerful computer that can be either within the same enclosure or on-board in our case. Using this configuration, the embedded computers are responsible for image acquisition. The captured stereo images are first partially treated on the fly to provide image quality information, as will be details in Section 4.1. Images are then transferred to the main computer, which handles the ego-motion computation that the system undergoes. For visualization purposes, we use two monitors that are connected to the embedded computers to show live navigation and image quality information (See Figure 2).

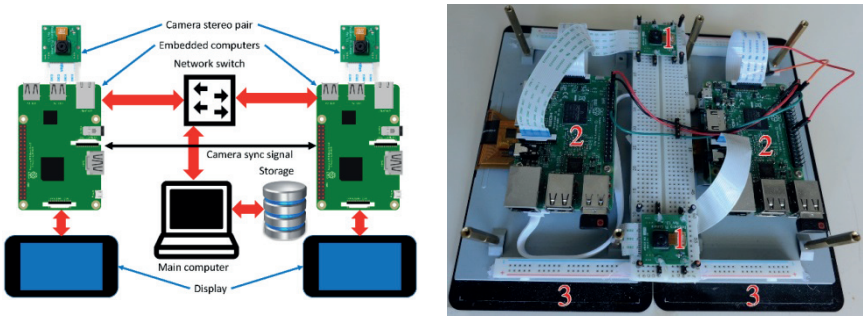


Figure 2. The hardware platform used for image acquisition and real-time navigation; it is composed mainly of (1) stereo camera pair, (2) Raspberry Pi © computers, and (3) monitors.

[1] A credit-card size ARM architecture based computer with 1.2 GHz 64-bit quad-core CPU and 1GB of memory, running Rasbian ©, a Linux based operating system.

4. Visual Odometry

Starting by computing and displaying image quality measures, the images are transferred over the network to a third computer as shown in Figure 2. This computer is responsible for hosting the visual odometry process, which will be explained in this section. We start first by introducing the used feature matching approach, and then we present the ego-motion estimation, finally, we explain the semi-global BA approach.

4.1. Image Quality Estimation

Real-time image quality estimation provides two benefits, first, it can alert the visual odometry process of having bad image quality, two reactions can be taken in this case, either pausing the process until taken image quality is recovered, or predicting position estimation based on previous poses and speed. We go for the first case while leaving the second for further development in future. Second, image quality indicators provide direct information to the operator to avoid going too fast in case of blur, or changing the distance to the captured scene when going under or over-exposed.

The first indicator is the image sharpness, we rely on image gradient measure that detects high frequencies that are often associated with sharp images, hence, we use a Sobel kernel based filtering, which computes the gradient with smoothing effect. This removes the effect of dust that is commonly present in underwater imaging. We consider the sharpness measure to be the mean value of the computed gradient magnitude image. The threshold can be easily learned from images by fixing a minimum number of matched feature points that are needed to correctly estimate the ego-motion. Similarly, an image lightness indicator is estimate as the average of L channel in CIE-LAB color space.

4.2. Sparse Stereo Matching

Matching feature points between stereo images is essential to estimate the ego-motion. As the two cameras alignment is not perfect, we start by calibrating the camera pair. Hence, for a given point on the right image, we can compute the epipolar line containing the corresponding point in the left image. However, based on the known fixed geometry, the corresponding point position is constrained by a positive disparity. Moreover, given that at deep water, the only light source is the one used in our system, the furthest distance that feature points that can be detected is limited, see Figure 3 for illustration.

This means that there is a minimum disparity value that is greater than zero. Furthermore, when going too close to the scene, parts of the image will become overexposed, similar to the previous case, this imposes a limited maximum disparity. Figure 4 illustrates the aforementioned constraints by dividing the

epipolar line into four zones, in which only one is an acceptable disparity range. This range can be straightforward identified by learning from a set of captured images (oriented at 30 degrees for better coverage).

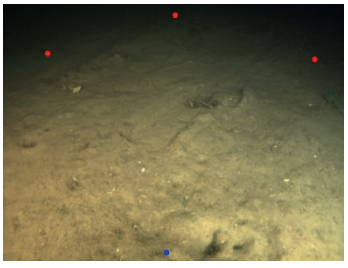


Figure 3. An example of underwater image showing minimum disparity (red dots, ~140 pixels) and maximum disparity (blue dot, ~430 pixels).

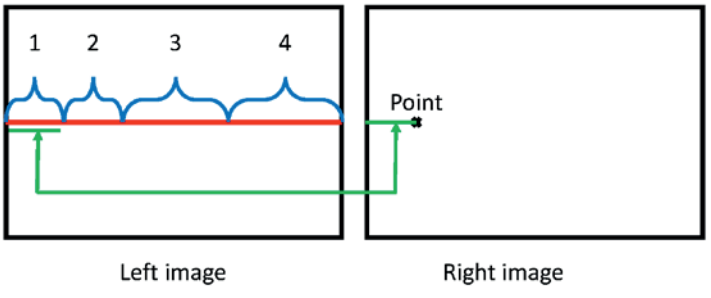


Figure 4. Illustration of stereo matching search ranges. (1) Impossible (2) Impossible in deep underwater imaging due to light's fading at far distances (3) Possible disparity (4) The point is very close so it becomes overexposed and undetectable.

In our approach, we propose to constraint the so-called acceptable disparity range further, which corresponds to the third range in Figure 4. Given the used lighting system, we can assume a light diffuse reflection model where the light reflects equally in all directions. Based on inverse-square law that relates light intensity over distance, image pixels' intensities are roughly proportional to their squared disparities. Based on such an assumption, we could use pixels' intensity to constraint the disparity and hence limiting the range of searching for a correspondence. To do so, we are based on a dataset of stereo images. For each pair, we perform feature points matches. Each point match (x_i, y_i) and (x_i', y_i'), x being the coordinate in the horizontal axis, we compute the squared disparity $d_i^2 = (x_i - x_i')^2$. Next, we associate each d_i^2 to the mean lightness value of a

window centered at the given point computed from L channel in CIE-LAB color space.

We assign a large window size (≈ 12) to compensate for using Harris operator that promotes local minimum intensity pixels as salient feature points. The computed $(\bar{l}_{x_i,y_i}, d_i^2)$ pair shows the linear relationship between the squared disparity and the average lightness. A subset of such pairs is plotted in Figure 5.

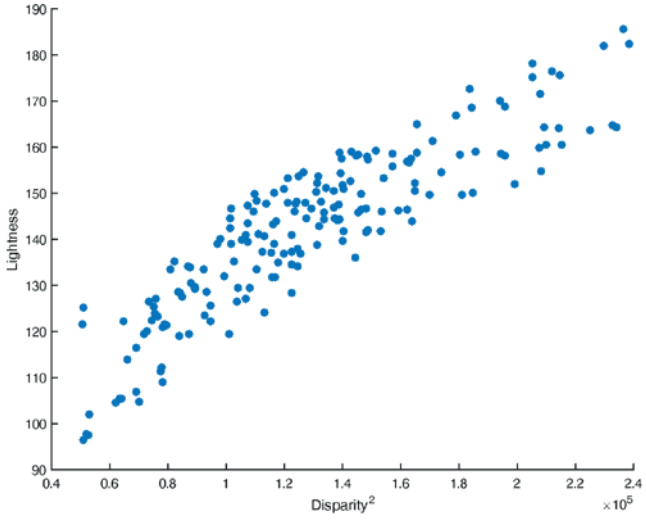


Figure 5. A subset of matched points squared disparity plotted against average pixel lightness.

In addition to finding the linear relationship between both variables, it is also necessary to capture the covariance that represents how rough is our approximation. More specifically, given the diagram shown in Figure 6, we aim at defining a tolerance t that is associated to each disparity as a function of lightness l. In our method, we rely on the Principal Component Analysis (PCA) technique to obtain this information. In details, for a given lightness l_i , we first compute the corresponding squared disparity d_i^2 using a linear regression approach as follows:

$$d_i^2 = -\alpha l_i - \beta \tag{1}$$

$$\alpha = \frac{Cov(L, D^2)}{Var(L)} \tag{2}$$

$$\beta = \bar{l} - \alpha \bar{d}^2 \tag{3}$$

where D and L are the disparity and lightness training set, d and l are their respective means.

Second, let $\boldsymbol{V}_2 = (v_{2,x}, v_{2,y})$ be the computed eigenvector that correspondences to the smallest eigenvalue λ_2. Based on the illustration shown in Figure 6, the tolerance t associated to d_i^2 can be written as:

$$t = \sqrt{\lambda_2^2 \left(\frac{{v_{2,x}}^2}{{v_{2,y}}^2} + 1 \right)} \tag{4}$$

By considering a normal error distribution of the estimated rough depth, and based on the fact that t is equal to one variance of D^2, we define the effective disparity range as:

$$d_i \pm \gamma \sqrt[4]{t} \tag{5}$$

where γ represents the number of standard deviations. It is trivial that γ is a trade-off between computation time and the probability of having points correspondences within the chosen tolerance range. We set $\gamma = 2$, which means that there is 95% probability to cover the data.

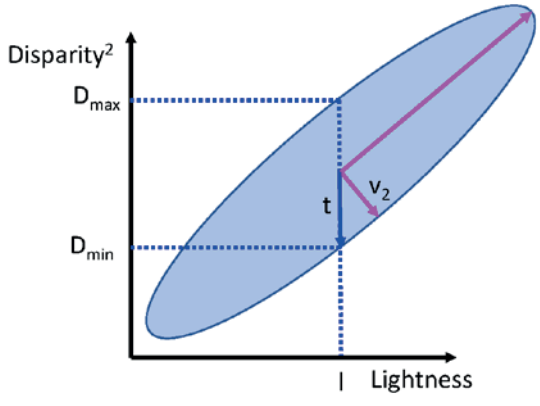


Figure 6. Illustration of disparity tolerance t given a lightness value l.

4.3. Initial Ego-Motion Estimation

Given left and right frames at time t (we call them previous frames), our visual odometry pipeline consists of four stages (an illustration is shown in Figure 7):

- Feature points matching for every new stereo pair $t + 1$. As described in Subsection 4.2.
- 3D reconstruction of the matched feature points using triangulation as described in Hartley and Zisserman (2004). Two displaced point clouds are obtained at this step.

- Relative motion computation using adaptation between the point clouds for the frames at t and $t+1$. Semi-Global BA procedure Nawaf et al. (2016) is applied to minimize re-projection errors; to be explained in the following subsections.

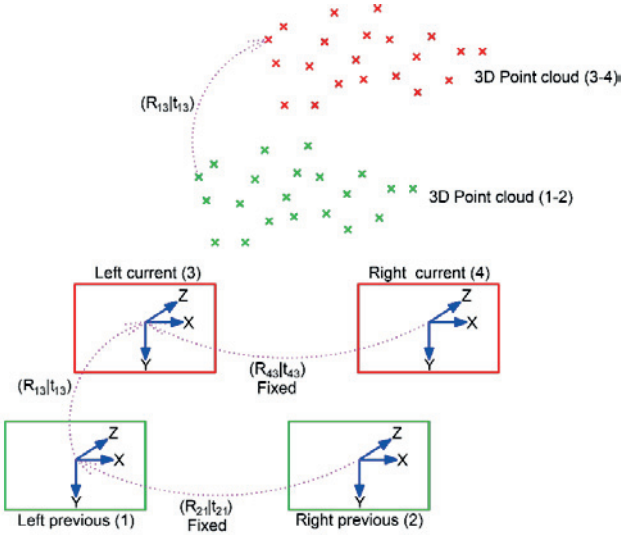


Figure 7. Image quadruplet, current (left and right) and previous (left and right) frames are used to compute two 3D point clouds. The transformation between the two points clouds is equal to the relative motion between the two camera positions.

In details, let (f_1, f_2, f_3, f_4) denote the previous left, previous right, current left and current right frames, respectively. For each new captured image pair, we compute a 3D point cloud using triangulation, as described in Hartley and Zisserman (2004) for the matched feature points that are obtained using the method proposed in the previous subsection.

The rigid transformation $[R|T]$ that is required for expressing the frames at time $t+1$ in the reference frame at time t is the rigid transformation that is required to move the 3D point cloud at time t to the one obtained at time $t+1$. Hence, the problem of calculating the orientation of the cameras at time $t+1$ in relation to time t leads back to the calculation of the transformation used to move from one point cloud to the other. This is possible under our configuration, with small rotation. We note here that there is no scale problem between both point clouds, which is specific to stereo systems. We consider here the left previous to left current frames $f_1 \rightarrow f_3$ positions to represent the system relative motion, and their relative transformation denoted $[R_{13}|T_{13}]$.

Below, we present the method to compute the transformation for passing from the point cloud calculated at time $t+1$, denoted P, to the one calculated at time t, denoted P'. So, we have two sets of n homologous points $P = P_i$ and $P' = P_i'$ where $1 \leq i \leq n$. We have:

$$P_i' = R_{13}P_i + T_{13} \tag{6}$$

The best transformation the minimizes the error r, the sum of the squares of the residuals:

$$r = \sum_{i=1}^{n} \|R_{13}P_i + T_{13}P_i'\|^2 \tag{7}$$

To solve this problem, we use the singular value decomposition (SVD) of the covariance matrix C:

$$C = \sum_{i=1}^{n} (P_i - \bar{P})(P_i' - \bar{P}') \tag{8}$$

where $\bar{P}$ and $\bar{P}'$ are the centers of mass of the 3D points sets P and P', respectively. Given the SVD of C as: $[U, S, V] = SVD(C)$, the final transformation is computed as:

$$R_{13} = VU^T \tag{9}$$

$$T_{13} = -R_{13}\bar{P} + \bar{P}' \tag{10}$$

Once the image pair $t+1$ is expressed in the reference system of the image pair t, the 3D points can be recalculated using the four observations that we have for each point. A set of verifications are then performed to minimize the pairing errors (verification of the epipolar line, the consistency of the y-parallax, and re-projection residues). Once validated, the approximated camera position at time $t+1$ are used as input values for the BA, as described earlier.

4.4. Uncertainty in Visual Odometery

Like any visual odometry estimation, the estimated trajectory using the method mentioned in the previous section is exposed to a computational error, which translates to some uncertainty that grows in time. A global BA may handle this error accumulation, however it is time consuming. From another side, a local BA is a trade-off for precision and computational time. The selection of n closest frames is done using standard Euclidean distance. Loop closure may occur when overlapping with already visited areas, which in turn enhances the precision. This approach remains valid as soon as the uncertainty is equal in all directions. However, as uncertainty varies across dimensions, the selection of the closest frames based on Euclidean distance is not suitable. In the following, we are going to prove that it is the case in any visual odometry method. Also, we will provide a more formal definition of the uncertainty.

Most visual odometry and 3D reconstruction methods rely on matched feature points to estimate relative motion between two frames. The error of matched features is resulting from several accumulated errors. These errors are due, non-exclusively, to the following reasons; the discretization of 3D points projection to image pixels, image distortion, the camera internal noise, salient points detection, and matching. By performing image un-distortion, and constraining the points that are matching with the fundamental matrix, the aforementioned errors are considered to follow a Gaussian distribution, so as their accumulation. This is actually implicitly considered in most computer vision fundamentals. Based on this assumption, we can prove that the error distribution of the estimated relative pose is unequal among dimensions. Indeed, it can be fitted to a multivariate Gaussian whose covariance matrix has non-equal Eigen values as we will see later. Formally, given a pair of matched points between two frames $\boldsymbol{m} \leftrightarrow \boldsymbol{m}'$. Based on our assumption, each matched point can be represented by a multivariate Gaussian distribution:

$$\mathcal{N}(m, \Sigma) \leftrightarrow \mathcal{N}(m', \Sigma) \tag{11}$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \tag{12}$$

The pose estimation procedure relies on the fundamental matrix that satisfies $m'Fm = 0$. Writing $m = [x\ y\ 1]^T$ and $m' = [x'\ y'\ 1]^T$. The fundamental matrix constraint for one matching pair of points can be written as:

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{13}$$

To show the variance of error distribution of estimated pose, without the loss of generality, we consider one example of configuration; identity camera intrinsic matrix $K = diag(1\ 1\ 1)$. Let us now take the case of pure translational motion between the two camera frames, $T = \begin{bmatrix} T_x\ T_y\ T_z \end{bmatrix}^T$, and $\theta = \begin{bmatrix} \theta_x\ \theta_y\ \theta_z \end{bmatrix}^T = [0\ 0\ 0\ ]$, the fundamental matrix in this case is given as:

$$F = {K^{-1}}^T E K^{-1} = {K^{-1}}^T [T]_x R\ K^{-1} = \begin{bmatrix} 0 & -T_Z & T_Y \\ T_Z & 0 & -T_X \\ -T_Y & T_X & 0 \end{bmatrix} \tag{14}$$

where $[T]_x$is the skew-symmetric cross-product matrix of T, and R is the rotation matrix, which is the identity in this case. Hence, equation 14 simplifies to:

$$-x'yT_Z + x'T_Y + y'xT_Z - y'T_X - xT_Y + yT_X = 0 \tag{15}$$

By using enough matched points (seven points in this case), we can recover the translation vector T by solving a linear system. However, the Gaussian noise whose covariance matrix is expressed by equation 12 will propagate to the variables T_X and T_Y, whereas for T_Z the error distribution is different due to the

product of two variables, where each is a Gaussian distribution. So the covariance is equal to $\Sigma/2$. Moreover, the recovered translation variables are correlated even though the observations are un-correlated. This is due to the usage of least square approach through SVD Strutz (2010). This leads to have the estimated pose follow a Gaussian distribution (proved experimentally in the following) with a full DOF covariance matrix (within the positive semi-definite constraint).

4.5. Pose Uncertainty Modeling

Pose uncertainty is difficult to estimate straightforward. This is due to the complexity of the pose estimation procedure and the number of variables. In particular, noise propagation through two consecutive SVDs (used for Fundamental matrix computation and Essential matrix decomposition). Instead, inspired by the unscented Kalman filter approach as proposed in Wan and Van Der Merwe (2000), we proceed similarly by simulating noisy input and trying to characterize the output error distribution in this case. This process is illustrated in Figure 8. In our work, we propose to learn the error distribution based on finite pose samples. This is done using a Neural Network approach which fits well to our problem as it produces soft output.

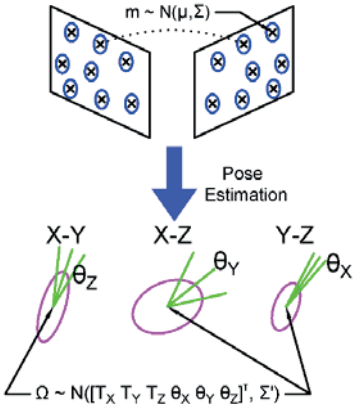


Figure 8. Illustration of error propagation through the pose estimation procedure. Estimated pose uncertainty is shown for each of the six degrees of freedom DOF. Full covariance matrix can result from diagonal error distribution of matched two-dimensional (2D) feature points.

There are two factors that plays role in the estimated pose uncertainty. First, the motion $\Omega = [T\ \theta]^\intercal$ between the two frames expressed by a translation T and a rotation θ, which is explained in the previous section. Second, the 3D location of the matched feature points. Although their location is not computed explicitly in our method, their distance from the camera affects the computation precision. In

particular, the further the points are from the camera, the less precise is the estimated pose. This is due to the fact that close points yield larger 2D projection disparity which is more accurate to estimate after the discretization. For instance, in pure translation motion, if all of the matched points are within the blind zone of the vision system (yield zero-pixels disparity after discretization), the estimated motion would be equal to zero. In the contrary, it will be more accurate when points are closer. Both mentioned factors are correlated to some point. For instance, given some points in 3D ($n > 7$), the estimated pose precision is a function of their depth, but also to the baseline distance. Hence, considering one factor is sufficient. In our work, we consider the motion as a base to predict the uncertainty.

Formally, given a motion vector $\Omega_j = \left[T_j\ \theta_j\right]^\intercal$, ideally, we want to find the covariance matrix that expresses the associated error distribution. Being a positive semi-definitive (PSD), such $n \times n$ covariance matrix has unique $(n^2 + n)/2$ entries, where $n = 6$ in our case, this yields 21 DOF, in which six are the variances. However, learning this number of parameters freely violates the PSD constraint. Whereas finding the nearest PSD in this case distorts largely the diagonal elements (being much fewer). At the same time, we found experimentally that the covariance between T and θ variables is relatively small when compared to such of inter T and inter θ. Thus, we propose to consider two covariance matrices Σ_T and Σ_θ. So, in total, we have 12 parameters to learn, in which six are the variances.

For the aim of learning Σ_T and Σ_θ, we have created a simulation of the pose estimation procedure. For a fixed well distributed 3D points $\{X_i \in \boldsymbol{R}^3 : i = 1..8\}$, we simulate two cameras with known relative rotation and translation. The points are projected according to both cameras to 2D image points, let us say $\{x_i \in \boldsymbol{R}^2\}$ and $\{x_i' \in \boldsymbol{R}^2\}$. These points are disturbed with random Gaussian noise as given by the equations 11 and 12. Next, the 3D relative pose is estimated based on the disturbed points. Let $\widetilde{\Omega}_j = \left[\tilde{T}_j\ \tilde{\theta}_j\right]^\intercal$ be the estimated relative motion. Repeating the same procedure (with the same motion Ω_j) produce a motion cloud around the real one. Now, we compute the covariance matrices [2] Σ_T and Σ_θ of the resulting motion cloud in order to obtain the uncertainty associated to the given motion Ω_j. Further, we repeat this procedure for a wide range of motion values[3]. Now, having the output covariance matrices (two for each motion vector Ω_j), we proceed to build a system that learns the established correspondences (motion ↔ uncertainty). So, that in case of new motion, we will be able to estimate the uncertainty. This soft output is offered by Neural Networks by nature, which is the reason that we adopt

---

[2] We increase the number of simulation runs until the output mean is close enough to the input real motion Ω_j, in our case we run the simulation 10000 times for each pose.

[3] In the performed simulation, we use the range [0-1] with 0.25 step size for each of the 6 dimensions, these values are in radians in case of rotation. This raises up to 15625 test case.

this learning method. In our experiments, we found that a simple Neural with single hidden layer Bishop (1995) was sufficient to fit well the data. The input layer has six nodes that correspond to motion vector. The output layer has 12 nodes, which corresponds to the unique entries in Σ_T and Σ_θ, hence, we form our output vector as:

$$O = [\Sigma_T^{11}\ \Sigma_T^{22}\ \Sigma_T^{33}\ \Sigma_T^{12}\ \Sigma_T^{13}\ \Sigma_T^{23}\ \Sigma_\theta^{11}\ \Sigma_\theta^{22}\ \Sigma_\theta^{33}\ \Sigma_\theta^{12}\ \Sigma_\theta^{13}\ \Sigma_\theta^{23}]^\intercal \tag{16}$$

where $\Sigma_{\cdot}^{ij}$ is the element of row i and column j of the covariance matrix $\Sigma_{\cdot}$.

In the learning phase, we use a gradient-descent based approach Levenberg-Marquardt backpropagation, which is described in Hagan et al. (1996). Further, by using the mean-squared error as a cost function we could achieve around 3% error rate. The obtained parameters are rearranged in symmetric matrices. In practice, the obtained matrix is not necessarily PSD, although this is rare to happen in the case of small variances. We proceed to find the closest PSD as $Q\Lambda_+Q^{-1}$, where Q is the eigenvector matrix of the estimated covariance, and Λ_+ the diagonal matrix of Eigen values in which negative values are set to zero.

4.6. Semi-Global Bundle Adjustment

After initiating the visual odometry, the relative pose estimation at each frame is maintained within a table that contains all pose related information (18 parameters per pose, in which six for the position, and 12 for two covariance matrices). At any time, it is possible to get the observations in the neighborhood of the current pose being estimated in order find potential overlaps to consider while performing BA. Since we are dealing with statistical representations of the observations, a divergence measure has to be considered. Here, we choose Bhattacharyya distance (Modified metric version can also be used Comaniciu et al. (2003)) for being reliable and relevant to our problem. In our case, the distance between two observations $\{\Omega^1\ , \Sigma_T^1, \Sigma_\theta^1\}$ and $\{\Omega^2\ , \Sigma_T^2, \Sigma_\theta^2\}$ is given as:

$$D = \frac{1}{8}\left(\Omega^1\ - \Omega^2\ \right)^{\mathrm{T}}\Sigma^{-1}\left(\Omega^1\ - \Omega^2\ \right) + \frac{1}{2}\ln\left(\frac{det\ \Sigma}{\sqrt{det\ \Sigma_1 + det\ \Sigma_2}}\right) \tag{17}$$

where

$$\Sigma_{\cdot} = \begin{pmatrix} \Sigma_T & \mathbf{0} \\ \mathbf{0} & \Sigma_\theta \end{pmatrix}, \Sigma = \frac{\Sigma_1 + \Sigma_2}{2} \tag{18}$$

Having selected the set of frames F in the neighborhood of the current pose statistically, we perform BA as follows; First, we divide F into two subsets, similar to Sibley et al. (2009), the first subset F_d contains the current and previous frames in time, whereas the other sub-set F_s contains the remaining frames, mostly resulting from overlapping with an already scanned area. Second, BA is performed on both subsets, however, although F_s parameters are included in the optimization,

they are masked as static so that they are not optimized in contrary to F_d . This strategy is necessary in order to keep past trajectories consistent.

After determining the error distribution arising with a new pose, it has to be compounded with propagated error from the previous pose. Similar to the SLAM approach, we propose to use a "Kalman filter" like gain, which allows controllable error fusion and propagation. Given an accumulated previous pose estimation defined by $\{\Omega^{\mathrm{p}}, \Sigma_T^{\mathrm{p}}, \Sigma_\theta^{\mathrm{p}}\}$ and a current one $\{\Omega^{\mathrm{c}}, \Sigma_T^{\mathrm{c}}, \Sigma_\theta^{\mathrm{c}}\}$, the updated current pose is calculated as:

$$\Omega^u = \Omega^c \tag{19}$$

$$\Sigma_T^{\mathrm{u}} = \left(\mathrm{I} - \Sigma_T^{\mathrm{p}}\left(\Sigma_T^{\mathrm{p}} + \Sigma_T^{\mathrm{c}}\right)^{-1}\right)\Sigma_T^{\mathrm{p}} \tag{20}$$

$$\Sigma_\theta^{\mathrm{u}} = \left(\mathrm{I} - \Sigma_\theta^{\mathrm{p}}\left(\Sigma_\theta^{\mathrm{p}} + \Sigma_\theta^{\mathrm{c}}\right)^{-1}\right)\Sigma_\theta^{\mathrm{p}} \tag{21}$$

5. Evaluation

The proposed method is desired to represent a trade-off between precision and computation time, the maximum precision being the case of global BA, whereas the fastest computation time is pure visual odometry. Moreover, a performance improvement is expected w.r.t local method due for better selection of neighboring observations. Therefore, we analyze the performance of our method from two points of view; computation time and precision.

5.1. Computation Time

We tested and compared the computation speed of our method as compared to using high level feature descriptors, specifically SIFT and SURF. At the same time, we monitor the precision for each test. The evaluation is done using the same set of images.

We run our experiments using the speed optimized BA toolbox as proposed in Lourakis and Argyros (2009). In the obtained results, the computation time when using the reduced matching search range, as proposed in this work is ~72% when compared[4] to the method using the whole search range (range 3 in Figure 4). Concerning SIFT and SURF, the computation time is 342% and 221%, respectively, as compared to the proposed method. The precision of the obtained odometry is reasonable which is within the limit of 3% for the average translational error and 0.02 [deg/m] for the average rotational error.

[4] The time evaluation is shown in percentage because the evaluation is carried out on three platforms with different computational power, in which one is an embedded unit. The minimum computation time being 220 ms.

5.2. Simulation Using Orthophoto

Our work falls within a preliminary preparation for a real mission. All of the experiments are tested within a simulated environment which uses images from previously reconstructed orthophoto in Drap et al. (2015) which is illustrated in Figure 9. The area covered is approximately $60m^2$ with very high resolution ~330 megapixels. The advantage of using simulated environment is that we can define precisely the trajectory, and then, after running the visual odometry method we can evaluate the performance and tune different components. Especially, with the lack of real sequences provided with odometry ground truth. Hence, we created a dataset of images based on simulating stereo camera motion which is shown superimposed on the orthophoto in Figure 9. The motion has an S-shape type scaled in one direction. The reason is to test the visual odometry method in two cases; when there is an overlap with previously scanned area and another case when there is not. Our method is more adapted to the first case scenario.

We evaluate the proposed semi-global BA as compared to three cases, using global BA, local BA, and without using BA. As expected, the method that uses global BA performs the best in this context. The translation error is 1.2%, while the rotation error 0.009 [deg/m]. Followed by our method, with 2.44% of translation and 0.011 [deg/m] of rotation errors. This is fairly ahead of the local BA method that achieved 3.68% of translation and 0.012 [deg/m] of rotation errors. The optimization free visual odometry showed the largest divergence with a translation error of 6.8% and rotation error of 0.08 [deg/m]. Figures 10 and 11 show the obtained trajectories for our method and the mentioned methods, respectively.

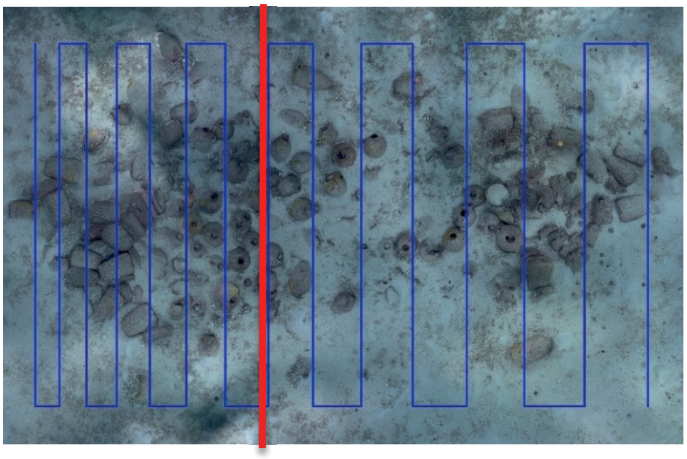


Figure 9. Simulation scenario with modified S-shape scanning profile which covers two situations; neighboring observations. Red border divides the map in overlapping/non overlapping path.

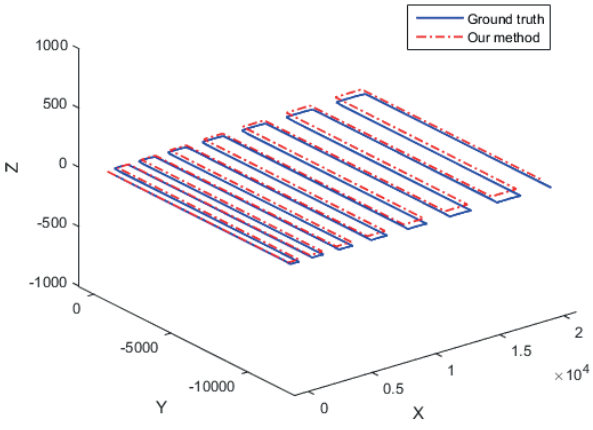


Figure 10. Estimated 3D trajectory using our method compared to ground truth.

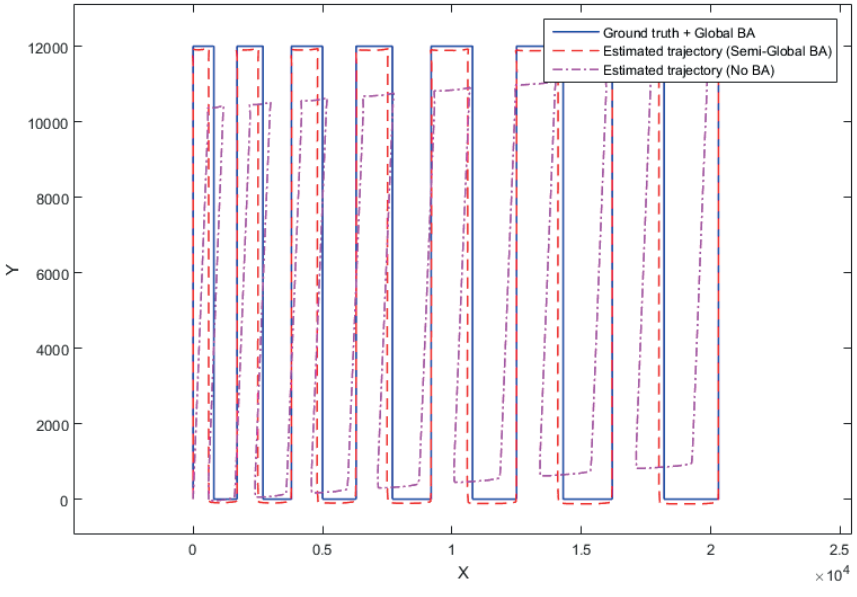


Figure 11. Comparison between several cases of visual odometry in terms of using BA. Note that the trajectory produced by the method without BA is scaled by ~1.8 for visualization purpose.

6. Conclusions and Perspectives

In this work, we introduced several improvements to the current traditional visual odometry approach in order to serve in the context of underwater surveys. The goal is to be adapted to embedded systems that are known for their lower resources. The sparse feature points matching guided with a rough depth

estimation using lightness information is the main factor beyond most of the gain in computation time when compared to sophisticated feature descriptors combined with brute-force matching. Also, using stochastic representation and selection of frames in the semi-global BA improved the precision as compared to local BA methods, while remaining within real-time limits.

Our future perspectives are mainly centered on reducing the overall system size, for instance, replacing the main computer in our architecture with a third embedded unit, which in turn does not keep evolving. This also allows to the user to reduce the power consumption, which increases the navigation time. On the other hand, dealing with visual odometry failure is an important challenge specially in the context of underwater imaging, which is mainly due to bad image quality. The ideas of failing scenarios discussed in this paper can be extended to deal with the problem of interruptions in the obtained trajectory.

Acknowledgments: This work has been partially supported by both a public grant overseen by the French National Research Agency (ANR) as part of the program *Contenus numériques et interactions* (CONTINT) 2013 (reference: ANR-13-CORD-0014), GROPLAN project (*Ontology and Photogrammetry; Generalizing Surveys in Underwater and Nautical Archaeology*)[5], and by the French Armaments Procurement Agency (DGA), DGA RAPID LORI project (*LOcalisation et Reconnaissance d'objets Immerges*). Logistic support for underwater missions is provided by COMEX [6].

References

1. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117.
2. Bay, H.;Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
3. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
4. Blanco, J.-L.; Fernandez-Madrigal, J.-A.; González, J. A novel measure of uncertainty for mobile robot slam with rao—Blackwellized particle filters. *Int. J. Robot. Res.* **2008**, *27*, 73–89.
5. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-Based Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–575.
6. Davison, A.J. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In Proceedings of the Ninth IEEE International Conference on Computer Vision—Volume 2, Washington, DC, USA, 13–16 October 2003.

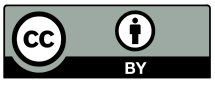
[5] http://www.groplan.eu

[6] http://www.comex.fr/

7. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067.
8. Drap, P. *Underwater Photogrammetry for Archaeology*; InTech Open: Rijeka, Croatia, 2012.
9. Drap, P.; Merad, D.; Hijazi, B.; Gaoua, L.; Nawaf, M.; Saccone, M.; Chemisky, B.; Seinturier, J.; Sourisseau, J.-C.; Gambin, T.; et al. Underwater Photogrammetry and Object Modeling: A Case Study of Xlendi Wreck in Malta. *Sensors* **2015**, *15*, 29802.
10. Eade, E.; Drummond, T. Scalable Monocular SLAM. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 469–476.
11. Eade, E.; Drummond, T. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In Proceedings of the British Machine Vision Conference 2008, Leeds, UK, 1–4 September 2008.
12. Geiger, A.; Ziegler, J.; Stiller, C. Stereoscan: Dense 3d reconstruction in real-time. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 963–968.
13. Hagan, M.T.; Demuth, H.B.; Beale, M.H.; De Jesús, O. *Neural Network Design*; PWS publishing company Boston: 1996.
14. Hartley, R.I.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2004.
15. Lourakis, M.I.; Argyros, A.A. SBA: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Softw.* **2009**, *36*, 2.
16. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
17. Montemerlo, M.; Thrun, S. FastSLAM 2.0. In *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*; Springer: Berlin, Heidelberg, Germany, 2007; pp. 63–90.
18. Montiel, J.; Civera, J.; Davison, A.J. Unified inverse depth parametrization for monocular SLAM. *Analysis* **2006**, *9*, 1.
19. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. Generic and real-time structure from motion using local bundle adjustment. *Image Vis. Comput.* **2009**, *27*, 1178–1193.
20. Nawaf, M.M.; Drap, P.; Royer, J.P.; Merad, D.; Saccone, M. Towards Guided Underwater Survey Using Light Visual Odometry. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-2/W3*, 527–533.
21. Nawaf, M.M.; Hijazi, B.; Merad, D.; Drap, P. Guided Underwater Survey Using Semi-Global Visual Odometry. In Proceedings of the 15th International Conference on Computer Applications and Information Technology in the Maritime Industries, Lecce, Italy, 9–11 May 2016; pp. 288–301.
22. Nawaf, M.M.; Tremeau, A. Monocular 3D Structure Estimation for Urban Scenes. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 526–535.

23. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; Volume 1, pp. I-652–I-659.
24. Piniés, P.; Tardós, J.D. Scalable SLAM building conditionally independent local maps. In Proceedings of the IROS 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 3466–3471.
25. Sibley, D.; Mei, C.; Reid, I.; Newman, P. Adaptive relative bundle adjustment. In Proceedings of Robotics: Science and Systems V, Seattle, WA, USA, 28 June–1 July 2009.
26. Strutz, T. *Data Fitting and Uncertainty: A Practical Introduction to Weighted Least Squares and Beyond*; Vieweg + Teubner: Wiesbaden, Germany, 2010.
27. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
28. Triggs, B.; Mclauchlan, P.; Hartley, R.; Fitzgibbon, A. Bundle adjustment: A modern synthesis. In *Vision Algorithms: Theory and Practice*; 2000; pp. 153–177.
29. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the Adaptive Systems for Signal. Processing, Communications, and Control. Symposium 2000, Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
30. Yamaguchi, K.; Mcallester, D.; Urtasun, R. Robust Monocular Epipolar Flow Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1862–1869.

Nawaf, M.M.; Royer, J.-P.; Pasquet, J.; *et al.* Underwater Photogrammetry and Visual Odometry. In *Latest Developments in Reality-Based 3D Surveying and Modelling*; Remondino, F., Georgopoulos, A., González-Aguilera, D., Agrafiotis, P., Eds.; MDPI: Basel, Switzerland, 2018; pp. 257–278.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).