Stereo Graph-SLAM for Autonomous Underwater Vehicles *

Pep Lluis Negre Carrasco, Francisco Bonin-Font, and Gabriel Oliver Codina

Systems, Robotics and Vision Group, University of the Balearic Islands (UIB), Palma de Mallorca, 07122 (Spain)

pl.negre@uib.cat, francisco.bonin@uib.es, goliver@uib.es, home page: http://srv.uib.es, Telephone:+34971171391

Abstract. The increasing use of Autonomous Underwater Vehicles (AUV) in industrial or scientific applications makes the vehicle localization one of the challenging questions to consider for the mission success. Graph-SLAM has emerged as a promising approach in land vehicles, however, due to the complexity of the aquatic media, these systems have been rarely applied in underwater vehicles. The few existing approaches are focused on very particular applications and require important amounts of computational resources, since they optimize the coordinates of the external landmarks and the vehicle trajectory, all together. This paper presents a simplified and fast general approach for stereo graph-SLAM, which optimizes the vehicle trajectory, treating the features out of the graph. Experiments with robots in aquatic environments show how the localization approach is effective underwater, online at 10fps, and with very limited errors. The implementation has been uploaded to a public repository, being available for the whole scientific community.

Keywords: Visual navigation, underwater robots, graph optimization.

1 Introduction and Related Work

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. AUVs are progressively being introduced especially in highly repetitive, long or hazardous missions, since they offer an important independence from support vessels and permit to reduce operational costs and resources.

^{*} 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). The authors are grateful to the members of the CIRS (University of Girona) for making available their facilities, including the AUV Girona500, used for some experiments.

An accurate robot localization becomes a crucial issue to assure the success of missions programed in AUVs, since, for example, errors in orientation generate large drifts on the estimated trajectory thus in the detection of the goal points.

The pose of an underwater vehicle can be estimated by means of, for instance, (a) inertial sensors, (b) acoustic or visual odometers, or (c) combining both, fusing all the sensorial data in navigation filters that smooth trajectories and errors ([1]). However, inertial and odometric methods are prone to drift, being necessary their periodical correction.

Simultaneous Localization And Mapping (SLAM) [2] techniques constitute the most common and successful approach to perform precise localization in unknown environments. SLAM approaches support on environmental landmarks, reinforcing the localization process by recognizing regions previously visited by the robot, in a process known as loop closing (image registration for visual systems).

Although the quality of images in sub-aquatic environments is strongly limited by the water and the illumination conditions, cameras present larger spatial and temporal resolutions than acoustic sensors, being more suitable for certain applications such as surveying, object identification or intervention [3].

Lately, researchers are focusing their efforts on the enhancement of visual SLAM techniques. Some of these solutions integrate, in an *Extended Kalman Filter* (EKF) context, the dead-reckoning data, the landmark locations and the loop-closings. These systems increase the running time with the size of the map, being too slow for long routes [4]. However, the running time can be reduced by either excluding landmarks from the filter state or performing the image registration in a limited portion of the surveyed environment [5].

EKF-based approaches require the definition of stochastic models for the vehicle motion and for the external measurements, implying errors inherent to their subsequent linearizations. Furthermore, the covariance matrices become dense over time, increasing the system uncertainty. To overcome these limitations, other authors approach the visual SLAM problem from the graph-optimization point of view: the vehicle odometry and the landmarks constitute the subsequent nodes of a graph. Nodes are linked by edges which represent the homogeneous transformation between nodes. When a loop is closed, the complete graph is optimized by applying, for instance, Gauss-Newton or Levenberg Marquardt techniques to minimize non-linear error functions based on least-squares or stochastic gradient descent [6] [7] [8] [9]. This last step means a complete graph adjustment entailing all nodes and transformations between them.

However, due to the inarguable difficulties that the aquatic environments present (light attenuation, flickering, scattering, lack of structured frameworks and the difficult to find and match image features), these techniques have been rarely applied to underwater field robotic systems. Very occasionally, some AUVs have used \sqrt{SAM} (*Smoothing and Mapping*) [6] for localization during the inspection of coral reefs [10] or in large vessel hull inspection applications [11].

This paper presents an evolved stereo-SLAM procedure especially adapted for underwater vehicle navigation and based on the g^2o [9] library. The main contribution of this work lies in the particular design and assembly of the main modules, its speeding up with respect previous solutions, and its application for AUV localization, obtaining noticeable results. Visual stereo odometry defines the initial guess for the graph nodes/edges and image registrations provide additional pose constraints needed to adjust and optimize the graph.

Section 2 outlines the theoretical background of the graph-optimization problem; section 3 explains the navigation software architecture; section 4 shows the experimental setup and some significant results, and section 5 concludes the paper.

2 Problem Formulation

Let $x = (x_1, x_2, ..., x_i, ..., x_n)$ be the absolute poses of the robot, associated to each graph node *i*, from the starting point (x_1) to the current point (x_n) .

Lets define $o_{i,j}$ as the observed pose constraint, calculated as the odometric displacement, between two consecutive nodes i and j, and let be $O_{i,j}$ its uncertainty matrix. In general, $o_{i,m}$ represents the odometric displacement between two non-consecutive nodes i and m, defined as the *composition* of the successive odometric displacements $o_{i,j}$ from i to m: $o_{i,m} = o_{i,i+1} \oplus o_{i+1,i+2} \oplus o_{m-2,m-1} \oplus o_{m-1,m}$. Lets define $f_{i,m}(x)$ as the function that measures the, in principle, noise-free transformation of node i to node m (the idea is illustrated in figure 1).



Fig. 1. A direct and indirect transformation from x_i to x_m .

Lets define the zero mean Gaussian error $e_{i,m}(x) = f_{i,m}(x) - o_{i,m}$ and lets define the cost function $F_{i,m}(x) = e_{i,m}^T O_{i,m} e_{i,m}$. Assuming also that the observations and measurements are independent, the global likelihood function can be defined as:

$$F(x) = \sum_{\forall (i,m) \in C} F_{i,m}(x), \tag{1}$$

where C represents the set of node pairs with an existing constraint $o_{i,m}$. The goal of the optimization problem is to find the value x^+ such as

$$x^{+} = \operatorname*{argmin}_{x} F(x). \tag{2}$$

To solve this non-linear least-squares problem, one can apply either Gauss-Newton or Levenberg Marquardt, approximating $e_{i,m}(x)$ by its first order Taylor expansion [9]. The result of the optimization process is x, which means that all the graph nodes can be recalculated simultaneously, anytime, during the mission.

3 Navigation Architecture

Consecutive stereo pairs are inserted into the navigation architecture. From each stereo pair, the system extracts image features, matches them reciprocally, and computes their 3D coordinates using the stereoscopy principle.

The features corresponding to each stereo pair are stored in a database, together with their corresponding 3D points, and a node identification number. The current pose associated to the vehicle is calculated composing the pose of the last graph node with the camera displacement computed by a stereo odometer. The system creates a new node with the computed raw pose and adds one edge between the last graph node and the new one, labeled with the odometric displacement. Afterwards, the algorithm searches for loop-closings between the image of the new node and the image of all previous nodes located inside a spherical *Region of Interest* (ROI), centered at the current camera pose. Limiting this search in a ROI saves running time and it is coherent with the fact that, although the robot passes through an already visited area, the overlap will be limited by the camera FOV (*Field of View*), being smaller as the distance between the current pose and the poses of the camera when imaged this area increases.

The Perspective N-Point (PNP) problem is adapted to perform the image registration. This method does not need any training data set, contrarily to systems based on, for instance, *Bag-of-Words* (BoW) [12], saving time and computational resources. This PNP problem refers to the process of computing the absolute camera pose given its intrinsic parameters and a set of 2D-3D point correspondences, and can be found in the literature formulated in a wide range of applications such as structure from motion, object recognition and localization [13]. This method was adapted as follows: (a) recover and match the 2D features of the current node and the node candidate for loop-closing; (b) if the number of matches is below a certain threshold, reject that node as a candidate; (c) elsewhere, extract the 3D points of the current node; (d) back-project the 3D points of the current node onto the 2D features of the candidate node, assuming the existence of a rotation and a translation, and applying RANSAC [14] to eliminate outliers:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fl_x & 0 & c_x \\ 0 & fl_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R|t] & \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$
(3)

where (u, v, 1) are the homogeneous image coordinates of the re-projected point, (X, Y, Z, 1) defines the 3D world point to be re-projected, (fl_x, fl_y) is the focal length, (c_x, c_y) is the principal point, and [R|t] turns out to be the matrix that describes the camera motion between both static scenes and minimizes the total re-projection error as: $[R|t] = argmin_{R,t} \sum_{i=1}^{N} ||p_i - s_i||^2$, being p_i the reprojected image point and s_i its corresponding original feature on the image of the candidate node. If [R|t] exists, both scenes are very likely to present strong similarities or to view common parts of the environment (although they can be rotated or/and translated), and the system assumes the presence of a loopclosing. Although [R|t] is not error-free, it is taken as the measurement $f_{i,m}$ defined in section 2.

The identification of a loop-closing entails a new direct edge between both nodes, labeled with [R|t], contrasting with the already existing indirect connection between the same nodes, calculated as the composition of the successive odometry displacements $(o_{i,m})$ between them.

At this point, the graph is optimized according to equation 2, recomposing simultaneously all nodes and edges with their respective labels in order to have a minimum quadratic error (see figure 2).



Fig. 2. Flow chart of the whole localization and mapping process.

Contrarily to the SAM-based [6] solutions, here, feature locations are not included in the graph, reducing its complexity and saving computation time and resources. Location of features can be improved later using their relative position with respect to the camera and the improved robot poses.

4 Experimental Results

Experiments were conducted with two different robots, Girona500 and Fugu-C, both equipped with one stereo rig looking downwards. Girona500 is a reconfigurable multipurpose AUV, designed by the CIRS (Centre for Research in Underwater Robotics-University of Girona) [15], and Fugu-C is a low-cost micro-AUV

developed at the University of the Balearic Islands, including also in its sensor suit an additional stereo rig looking forward, a MEMS-based Inertial Measurement Unit and a pressure sensor. The down-looking camera was used for the incoming localization experiments. Figure 3 shows an image of each robot.

Both robots use Linux as operating system and ROS [16] as middleware. The present approach, installed and executed in both robots, has been also implemented in ROS, and it has been uploaded and published in a public repository [17].

The General Graph Optimization library $(g^2 o)$ [9] was used as the framework for graph management, including, graph and node creation/addition, graph update and graph optimization. This approach deserves special attention since: a) it offers a ROS wrapper, b) it covers all different particular issues addressed in other approaches ([6] [18] [7] [8]), c) outperforms [6] [8] in running time and it achieves a similar performance than [7], and d) it is extensible to a wide range of optimization problems.



Fig. 3. (a): Fugu-C. (b) Girona500. (c) UIB pool bottom. (d) CIRS pool bottom.

All executions were run online with a frame rate of 10fps. The radius for loopclosing search was set to 7m-8m. LibViso2 [19] was used as a stereo odometer and SIFT features/descriptors were chosen to compute the image registrations. The sparse Cholesky decomposition (CHOLMOD) was used as the linear solver and Levemberg-Marquardt with 40 iterations as the algorithm for the optimization [9]. Odometry samples were queued and synchronized with the last image, and the SLAM samples were computed for each queued image/odometry pair and also synchronized with them. The user can trust the last robot pose given by the last graph optimization although this might not be the corresponding to the last gathered image.

As a sample of the obtained results, four representative experiments are shown next: three conducted with Fugu-C, two of them moving in a pool located at the UIB and another one in an artificial pond, and a fourth one corresponding to a survey conducted with Girona500 in a pool located at the CIRS. The bottom of both pools was covered by printed digital posters of real marine contexts. Both posters were used to compute the trajectory ground truth by registering each image grabbed online during the trajectory with the whole figure of the corresponding poster. In some of the experiments, several objects such as amphoras and rock mockups were deployed inside the pool in order to simulate a realistic, non flat, sea floor with 3D structures. Figures 3-(c) and 3-(d) show two images captured by the robots during the respective trials in the pools. See how in figure 3-(c) the poster of the bottom is occluded by the objects (amphoras and an orange box) deployed on it.

Plots 4-(a), 4-(c) and 4-(e) show the three trajectories in the (x,y) plane, done in the pools. The ground truth is depicted in green, the SLAM trajectory with the graph nodes in blue, and the odometry trajectory in red. Additionally, the plots on the left show, also in blue, all the detected loop closings as lines linking nodes. All plots show how the SLAM trajectory is close to the ground truth while the odometry drifts in all cases.



Fig. 4. Trajectory and errors: (a)-(b) CIRS pool with Girona500, (c)-(d) Fugu-C: UIB pool survey, (e)-(f) Fugu-C: UIB pool loop with amphoras and boxes on the bottom.

Plots 4-(b), 4-(d) and 4-(f) show the trajectory error for the three experiments, computed as the euclidean distance between each 3D ground truth point and its corresponding odometry or SLAM 3D point, according to the common timestamp. It is noticeable that the SLAM error, in all cases, is bounded between 0 and 30cm during the whole trajectory, meaning that, the localization of the vehicle will be close to the ground truth, regardless the length and duration of the robot mission. Conversely, the odometry error grows unbounded and it can reach peaks up to 2.7m. Although in the third experiment (4-(e)-(f)) the pool bottom is covered, in some points, by objects that occlude the poster, the SLAM trajectory error is still very low.

Table 1 shows, for each experiment, the trajectory length and the mean error for the odometry and for the SLAM estimates. Results show that the SLAM mean error is clearly much lower than the odometry mean error.

Experiment	odometry	SLAM
	mean $error(m)/traj. dist(m)/$	mean $\operatorname{error}(m)/\operatorname{traj.} \operatorname{dist}(m)$
CIRS pool Girona500	0.85/46.9	0.17/46.0
Fugu-C pool/survey	0.25/22.7	0.06/23.0
Fugu-C pool/amphoras	1.81/8.4	0.07/7.0

Table 1. Trajectory length and mean errors for the odometry and SLAM.

Fugu-C was also programmed to perform a zig-zag survey in an artificial pond, starting and finishing at the same point where a marker was placed for the unique purpose of identification. Although the presence of mud on the pond bottom and regardless the marker placed at this initial/final point, the system was able to find many loop-closings along the route. Figure 5 shows four loop closings corresponding to nodes quite distant in time. Features are shown in red and the inlier matchings in blue. In practice, the PnP problem was solved positively with a minimum of 7 inliers. See as image 5-(b) shows the correspondences between two frames that visualize part of the marker, one grabbed at the start (node 2) and the other at the end of the trajectory (node 89). Figure 6 shows the route followed by the robot according to the stereo odometry estimates (in red) and to the SLAM (blue bullets as the nodes and discontinuous lines to represent the loop-closing edges). Although the lack of ground truth impedes the computation and comparison of trajectory errors it is evident how the odometry drifts considerably and the SLAM estimates follow the zig-zag trajectory, ending at the same point where it started.

5 Conclusions

This paper has presented an evolved stereo graph-SLAM algorithm especially addressed to improve the localization of underwater robots equipped with a stereo rig. The graph management is based on the g^2o library and the image registration process is reinforced by minimizing the 3D-2D re-projection error



Fig. 5. Four loop closings emerged during the experiment in the pond

of environmental landmarks. This type of visual SLAM approaches is scarcely applied in underwater field robotic applications, and they normally implement variations of the SAM approach, which includes in the graph the robot pose and landmark coordinates. Contrarily, our process graphs only the robot displacements, which is conceptually simpler and faster. Experimental results obtained with underwater robots moving in aquatic scenarios show how the error of the SLAM estimates is bounded, meaning that the approach maintains the same range of errors, regardless the duration and length of the trajectory. Finally, the implementation is available to the scientific community in a public repository.



Fig. 6. Trajectory corresponding to the experiment in the pond: (a) in 2D, (b) in 3D.

References

1. Kinsey, J.C., Eustice, R.M., Whitcomb, L.L.: 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)

- Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping (SLAM): part I. IEEE Robotics and Automation Magazine 13(2) (June 2006) 99–110
- Bonin, F., Burguera, A., Oliver, G.: Imaging systems for advanced underwater vehicles. Journal of Maritime Research 8(1) (April 2011) 65–86
- Schattschneider, R., Maurino, G., Wang, W.: Towards stereo vision slam based pose estimation for ship hull inspection. In: Proceedings of Oceans, Waikoloa, Hawaii (June 2011) 1–8
- Eustice, R., Pizarro, O., Singh, H.: Visually augmented navigation for autonomous underwater vehicles. IEEE Journal of Oceanic Engineering 33(2) (April 2008) 103– 122
- Dellaert, F., Kaess, M.: Square root sam: Simultaneous localization and mapping via square root information smoothing. The International Journal of Robotics Research 25(12) (2006) 1181–1203
- Konolige, K., Grisetti, G., Kmmerle, R., Burgard, W., Limketkai, B., Vincent, R.: Efficient sparse pose adjustment for 2d mapping. In: Proceedings of the IEEE/RSJ IROS, Taipei, Taiwan (2010)
- Strasdat, H., Montiel, J., Davison, A.: Scale drift-aware large scale monocular slam. In: Proceedings of Robotics: Science and Systems (RSS). (2010)
- Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g²o: A general framework for graph optimization. In: Proceedings of the IEEE ICRA. (Shangai-China 2011) 3607–3613
- Beall, C., Dellaert, F., Mahon, I., Williams, S.: Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys. In: Proceedings of Oceans, Santander, Spain (June 2011)
- Hover, F.S., Eustice, R.M., Kim, A., Englot, B., Johannsson, H., Kaess, M., Leonard, J.J.: Advanced perception, navigation and planning for autonomous inwater ship hull inspection. Journal of Robotics Research 31(12) (2011) 1445–1464
- Kim, A., Eustice, R.M.: Combined visually and geometrically informative link hypothesis for pose-graph visual slam using bag-of-words. In: Proceedings of the IEEE/RSJ IROS, San Francisco, USA (2011) 1647–1654
- Bujnak, M., Kukelova, S., Pajdla, T.: New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. Lecture Notes in Computer Science (6492) (2011) 11–24
- Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6) (1981) 381–395
- Ribas, D., Palomeras, N., Ridao, P., Carreras, M., Mallios, A.: Girona 500 auv: From survey to intervention. IEEE/ASME Transactions on Mechatronics 17(1) (2012) 46–53
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.: ROS: an open source robot operating system. In: ICRA Workshop on Open Source Software. (2009)
- Negre, P.L., Massot, M.: ros.org. http://wiki.ros.org/stereo_slam (2013) [Online; published January 2014].
- Grisetti, G., Stachniss, C., Burgard, W.: Non-linear constraint network optimization for efficient map learning. IEEE Transactions on Intelligent Transportation Systems 10(3) (2009) 428–439
- Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3d reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany (June 2011)