Visual Odometry for Autonomous Underwater Vehicles

Stephan Wirth, Pep Lluis Negre Carrasco, Gabriel Oliver Codina
Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears
Cra. Valldemossa, km. 7,5 07122 Palma de Mallorca (Spain)
Email: {stephan.wirth, pl.negre, goliver}@uib.es

***Abstract*—Vision based motion estimation algorithms are widely used in ground-based and aerial robotics. Combined with inertial measurement units, they have proven to be a precise and low-cost sensor for velocity and pose estimation. In this paper we show that stereo vision based odometry can be used by autonomous underwater vehicles (AUV) that navigate close to the seabed for velocity and incremental pose estimation in small areas. We present the integration of two different stereo visual odometry algorithms into an AUV and experiments carried out in laboratory and harbour conditions comparing vision based pose estimates with ground truth.**

I. Introduction

The latest advances in Autonomous Underwater Vehicles (AUVs) demand accurate estimations of the vehicle's pose and velocity. The tasks to be performed by such robots are increasingly complex and, in most cases it is needed to precisely position the underwater vehicle to a certain location in order to execute the desired task. Navigation systems based on wirelessly transmitted data, such as the Global Positioning System (GPS) or mobile network positioning, can not be used in underwater environments. One can overcome this issue by introducing artificial landmarks. However the cost and effort for this is high and for small missions it is preferred to rely on positioning using self-contained sensors. One of the most popular positioning sensors for underwater vehicles is the Doppler Velocity Log (DVL) which provides precise velocity and altitude updates. However, it is a big and expensive device that cannot be integrated in many underwater vehicles.

Visual odometry is the process of estimating a vehicle's 3D pose using only visual images. This technique is becoming popular in AUVs for navigation, station keeping and the provision of feedback information for manipulation. Visual odometry output is often fused with Inertial Measurement Units (IMU) to provide a cheaper alternative to DVLs [1], [2]. However, the limitations of underwater vision are widely known, and its performance depends on many factors such as visibility, lighting and distortion resulting from varying refractive indices.

In this paper we present the integration of two existing stereo visual odometry algorithms into an AUV, the experiments carried out in laboratory and harbour environments, as well as the evaluation method to compare the performance of both algorithms.

First, the paper explains related work on visual odometry. Section III describes the general concepts of visual odometry systems and presents a brief overview of the two analyzed algorithms, followed by the explanation of the evaluation method in Section IV and experiments in different evironments. Finally, the conclusion of the study and future perspectives are presented.

II. Related Work

The basic visual odometry algorithm pipeline [3] consists of the following steps: first, keypoints (landmarks) are identified in each camera frame and feature descriptors for these points are extracted. Then, the depth for every landmark is estimated using stereo, structure from motion or a separate depth camera. Subsequently, features are matched across time frames and the rigid-body transformation that best aligns the features between frames is estimated. The result of this process is an estimation of camera motion between frames and therefore it is necessary to integrate this data over time to obtain the vehicle's absolute position and orientation.

Many implementations of visual odometers for AUVs use this pipeline and adapt every stage (such us keypoint detector type and feature matching method) to their needs. Common feature detectors used for real-time visual odometry include Harris corners [4], [5], FAST [6], [7] or SIFT [8], [9] features. Wide range of approaches are involved in the process of matching feature across frames, [10] propose a circle match between stereo image pairs of two consecutive frames and uses non-minima/maxima-suppression techniques [11] to reduce the number of correspondences. RANSAC-based methods [12], [13] and graph-based consistency algorithms [14] have also been proven to robustly match features across frames.

Previous to the motion estimation stage, keypoint bucketing [15] has been shown to help reduce the inlier reprojection errors. Finally, the motion estimation process can be solved through different methods. Using a closed-form solution to the least-squares problem of absolute orientation [16], it is possible to directly minimize the Euclidean distance between matched features in order to compute the rigid-body transformation of the camera frame [7]. Several ground-based visual odometers do not use 3D distances but implement the method of minimizing the pixel reprojection error [7], [10], [14].

Other localization algorithms are for AUVs are based on mosaics or structured environments and therefore assume prior knowledge of the area to explore [17]–[19].

Many of the techniques and algorithms described in this section are primary focused on ground and aerial vehicles and most of them have not been tested in underwater environments.

978-1-4799-0002-2/13/$31.00 ©2013 IEEE

In this field, the complexity of the visual odometry task increases due to the dynamic light conditions and decreasing visibility with depth and turbidity. Our focus is primarily on integrating and validating two different stereo-based visual odometry approaches in such environments.

III. Visual Odometry Systems

A visual odometry system consists of one or more cameras, a processing unit, and the required algorithms that process incoming images. Algorithms for visual odometry - as opposed to full SLAM algorithms - focus on fast frame-to-frame motion estimates without keeping a large history for loop closing. The emphasis lies on accurate measurements at high frequencies. Systems that use just one camera need translational movement for 3D motion estimation and all measurements have to be scaled by an unknown factor to be on a metric scale. Using a calibrated stereo camera overcomes both of the aforementioned problems as 3D coordinates of matched points in a single left/right image pair can be computed by triangulation. We therefore focus our work on visual odometry systems that use stereo cameras.

The two publicly available algorithms we compare are *libviso2* [10] and *fovis* [7]. The former has been successfully used on cars, and the latter has proven to work for micro aerial vehicles.

We created wrappers for both libraries[1] to integrate them into the *Robot Operating System* ROS [20]. This simplifies the integration in different robotic vehicles as well as comparison with other integrated sensors.

Both algorithms have very similar processing pipelines containing feature detection, filtering, matching, and motion estimation. Table I summarizes the steps of the algorithms, outlining similarities and differences. To limit the extent of this paper, the interested reader is referred to the original papers for a deeper understanding and a more detailed description of the algorithms. In the following we give a brief overview of both algorithms.

A. libviso2

The features used by libviso2 are simple blob and corner detector masks, resulting in a large amount of interest points of four different classes. Non-minima-/non-maxima-suppression is used for initial feature reduction. In this step, two different thresholds for minima/maxima selection are used resulting in two sets of sparse strong features and dense less strong features. This allows for a multi-stage matching coarse to fine using matches of the strong features to limit the search range for the weaker ones. The used descriptor contains only 16 values of Sobel filter responses distributed inside an 11x11 pixel window. The similarity measure for two features is the sum of absolute differences of their descriptors. The computation of this measure is sped up using SIMD instructions.

The matching of features follows the following order: current left image to current right image, current right image to previous right image, previous right image to previous left image and previous left image to current right image. If the target feature of the last matching is the same as the source feature of the first matching the match is classified as valid. Apart from the aforementioned search range limitation, possibly wrong matches are discarded using a neighborhood support heuristic. From the obtained matches a subset is drawn that is distributed over the whole image by dividing the image in a grid of subimages and drawing a fixed maximum number of matches from each cell.

Subsequently, 3D coordinates of the features in the previous image pair are calculated through triangulation. Gauss-Newton minimization of the reprojection error of these 3D points onto the current left and right images leads to the rigid transformation from the previous image's camera pose to the current camera pose. The motion estimation procedure is wrapped in a RANSAC [21] scheme to get rid of outliers.

B. fovis

fovis uses the FAST feature detector [6], [22] with an adaptive threshold on three Gaussian pyramid levels. The number of features is reduced taking the best features of each cell of a regular grid. The feature descriptor is an intensity normalized 9x9 pixel window centered on the feature location. The bottom right pixel is left out to get better memory alignment for rapid similarity measure computation using SIMD instructions.

Features are matched from the current left to the left reference frame. The right frames are used for disparity lookup only. Inlier/outlier classification is done using the concept of a maximally consistent clique, i.e. matches are sequentially added while the relation of euclidean distance to other matched features does not change from the reference to the current frames. For motion estimation, the reprojection error from 3D positions of current features to the reference left frame as well as from 3D positions of the reference features to the current left frame is minimized using the Gauss-Newton method. If the reprojection error of a certain match exceeds a threshold, that match is discarded. The motion estimate is refined using only those matches that survived this process.

Apart from the used feature detectors, an important difference of the two algorithms is the selection of the reference frame for motion computation. libviso2 does not have automatic selection of the reference frame and replaces it in each iteration. In contrast, fovis only replaces the reference frame if the number of inlier matches drops below a given threshold. Its purpose is to reduce the drift that could arise from sensor noise or bad calibration.

Both algorithms use feature descriptors that are not invariant to rotation or scale changes. The frequency of both algorithms has therefore to be high to cope with these movements.

IV. Experiments

The experiments carried out use data gathered during the TRIDENT project[2] both in laboratory and harbour conditions using the Girona500 AUV [23] as vehicle (Figure 1). This vehicle has been designed by the Universitat de Girona and, without change, does not have stereo vision system, but allows the mounting of external systems on the bottom payload area.

[1] See http://www.ros.org/wiki/viso2_ros and http://www.ros.org/wiki/fovis_ros for source code and documentation.

[2] See http://www.irs.uji.es/trident/

Table I. COMPARISON OF THE TWO VISUAL ODOMETRY ALGORITHMS.

	libviso2	**fovis**
Feature Scales	1 (reduced) level	3 gaussian pyramid levels
Feature Detection	5x5 blob and corner masks	FAST with adaptive threshold
Initial Feature Reduction	non-minimum-/non-maximum-suppression	grid filter
Descriptor	16 Sobel filter responses	9x9 pixel window, intensity normalized
Subpixel Position Refinement	parabolic fitting	ESM
Matching Search Space Limitation	two-step candidate reduction	initial rotation estimate (Mei et al.) + search window
Matching Images	circlular match	left to right & left to previous left
Match Reduction	grid filter	none
Initial Outlier Rejection	2D neighborhood support	none
Outlier Classification	iteratively during motion estimation (RANSAC)	maximally consistent clique
Minimized Reprojection Error	previous 3D to current left and right image	previous 3D to current left and current 3D to previous left image
Motion Filtering	Kalman	none (external)
Keyframe Selection	none (external)	automatic, based on number of matches

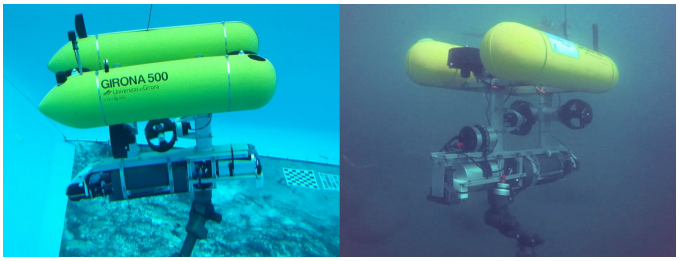


Figure 1. The Girona 500 AUV during laboratory and harbour experiments. The white box in the front of the lower part is the stereo camera. The cylinder right next to it contains the image processing unit.

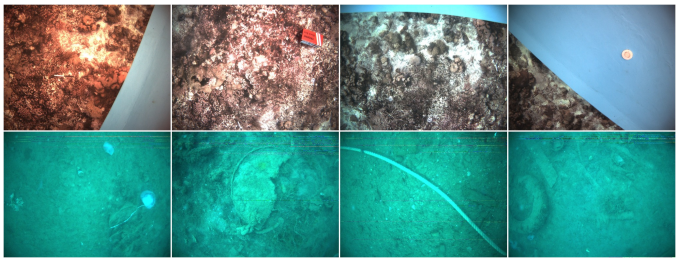


Figure 2. Sample images from the sequences during the laboratory (top row) and harbour (bottom row) experiments.

This feature is used to attach the stereo vision unit developed by our group in the Universitat de les Illes Balears. This unit is called Fugu-Flexible which is basically composed of two stereo rigs and a computer system. Each module, hardware and cameras, is placed in an independent sealed case rated for up to 100 meters depth. The main Fugu-Flexible hardware specifications are:

- Two stereo cameras, one with a focal distance of 3.8mm (66°HFOV), and the other with 2.5mm (97°HFOV).
- A motherboard with an Intel i5 processor at 2.33GHz with 4 cores.
- A PCI express card with two firewire ports to connect the cameras to the motherboard.

Figure 2 shows some example images from the laboratory and harbour environments. Ground truth for the laboratory experiment is extracted by matching each image that has been captured against the known image that is printed on the floor of the test pool. As the size of the print is known, 6 DOF camera poses can be computed minimizing reprojection errors of matched features.

For the harbour experiment, the determination of ground truth is more difficult, as no external sensors have been used and positions of natural landmarks are not known. One aim of the project was the offline construction of a consistent seabed mosaic (see [24]). The computation of this mosaic includes global optimization of all camera poses. The resulting trajectory does not suffer from drift and is therefore chosen as our ground truth reference.

Table II. EXPERIMENT DETAILS

	CIRS Lab	**Roses Harbour**
Total Trajectory Length (m)	47.54	90.48
Average Velocity (m/s)	0.15	0.12
Average Altitude (m)	1.59	1.34
Average Depth (m)	2.98	1.46

In the harbour experiment, a DVL and an AHRS have been used as navigation sensors to let the vehicle follow a previously defined trajectory autonomously. In the laboratory experiments, the vehicle has been remotely controlled by a human operator. Table II summarizes the characteristics of the experiments.

The AUV was configured to not control pitch, roll and sway. Motions in these degrees of freedom are therefore rather small and we cannot compute reasonable error measures for them.

A. *Evaluation Method*

As visual odometry suffers from drift, comparing whole trajectories, i.e. poses in time directly to ground truth is not very meaningful. Instead, we subdivide the paths into small pieces and compare velocities for each piece to the matching piece in our ground truth. This method is widely used in visual odometry evaluation [25] and depends clearly on the size of the sub-paths. However, since it is used to compare different visual odometers, the important thing here is to be coherent with the size of the pieces for all algorithms.

Table III. COMPARISON OF TRANSLATIONAL AND ROTATIONAL ERRORS FOR LABORATORY AND HARBOUR EXPERIMENTS.

Environment	Algorithm	Translation	Rotation
Laboratory	libviso2	0.978%	0.001739 [deg/m]
	fovis	0.601%	0.006311 [deg/m]
Harbour	libviso2	0.833%	0.007913 [deg/m]
	fovis	1.049%	0.003122 [deg/m]

Table IV. COMPARISON OF NUMBER OF INLIERS AND RUNTIME FOR LABORATORY AND HARBOUR EXPERIMENTS.

Environment	Algorithm	Num. inliers	Runtime
Laboratory	libviso2	366	0.106 [s]
	fovis	500	0.246 [s]
Harbour	libviso2	137	0.086 [s]
	fovis	64	0.058 [s]

B. Results

A comparison of linear velocity estimates to ground truth can be found in Figure 3 for libviso2 and 4 for fovis. Upper graphics refer to experiments in laboratory while the bottom ones refer to harbour tests. Linear velocity is shown on the left for both odometry estimation (blue) and ground truth (green), together with its error drawn in red. Finally, on right plot, a comparison of angular velocity for odometry estimation (blue) and ground truth (green) is also presented. No differences can be highlighted comparing angular velocity error for libviso2 and fovis in both the laboratory and harbour conditions. However, when linear velocities are examined, fovis presents better performance when executed in laboratory environment.

These results are summarized in Table III for laboratory and harbour experiments. The translation column indicates the mean average translation error that suffers the odometer in percent, which is the number of meters diverted for every 100 meters of trajectory. Furthermore, the rotation column indicates how many degrees the odometer is deviated per meter, relative to ground truth.

The results presented in Table III demonstrates that libviso2 has similar mean translation error for both environments, but differs significantly in rotation error. As can be seen in Table IV, the mean number of inliers per frame is larger in the laboratory than in the harbour (where environmental conditions are worse). It means libviso2 presents a strong performance in the linear motion estimation even when the number of inliers is small.

Results for the fovis odometer are substantially different in translation error comparing laboratory and harbour conditions. By examining the number of inliers of the two odometers shown in Table IV it is easy to see that fovis is significantly less robust in harbour environments where the quality of the images is lower and the seabed has a poor texture (see Figure 2). Furthermore, unlike libviso2, fovis presents a better performance when estimating the angular motion in harbour conditions. This improvement over libviso2 is due to the method of keyframe selection based on the number of matches introduced by fovis. This technique has a positive impact when the vehicle experiences pure rotational movements, as it can keep the reference frame for longer and therefore reduce the drift significantly.

The runtime column shown in Table IV indicates the mean algorithm execution time per frame. For both libviso2 and fovis odometers the runtime is clearly related to the number of inliers to be processed by the motion estimation algorithm, thus large number of inliers increases processing time.

V. CONCLUSION

Visual odometry is a method for motion estimation that gives good results using a low-cost sensor in reasonable conditions. We presented open source wrappers for two publicly available visual odometry algorithms to the community that ease the integration into existing robotic platforms. We have shown that these algorithms can be used in an underwater environment if both the visibility conditions and the appearance of the sea floor result in images with sufficient texture. Like all incremental motion estimation methods, visual odometry suffers from drift and has to be combined with other sensors to get precise long-term position estimates. Apart from that, failure might occur in situations with insufficient texture, motion blur, or bad visibility.

Despite the mentioned drawbacks we have shown that a visual odometer using a downward looking stereo camera can be a valuable sensor for underwater vehicles giving good estimates for linear movement and rotation about the vehicle's vertical axis. Comparing the performance of libviso2 and fovis in underwater environments, libviso2 is preferable when the vehicle operates in real marine waters with critical lighting conditions and slightly textured seabed. The robustness of this algorithm against poor visibility contrasts with the fact of having a worse rotation estimation when compared with fovis in harbour experiments.

A possible improvement for the angular motion estimation in libviso2 could be the implementation of the reference frame selection technique developed by fovis algorithm. This method could substantially reduce the drift in both linear and angular estimates.

VI. ACKNOWLEDGMENT

This work is partially supported by the Spanish Ministry of Economy and Competitiveness under contract DPI2011-27977-C03-02 (TRITON Project), by the Balear Goverment (ref 71/2011), FEDER Fundings and by the European Commissions FP7 under grant agreement 248497 (TRIDENT Project).

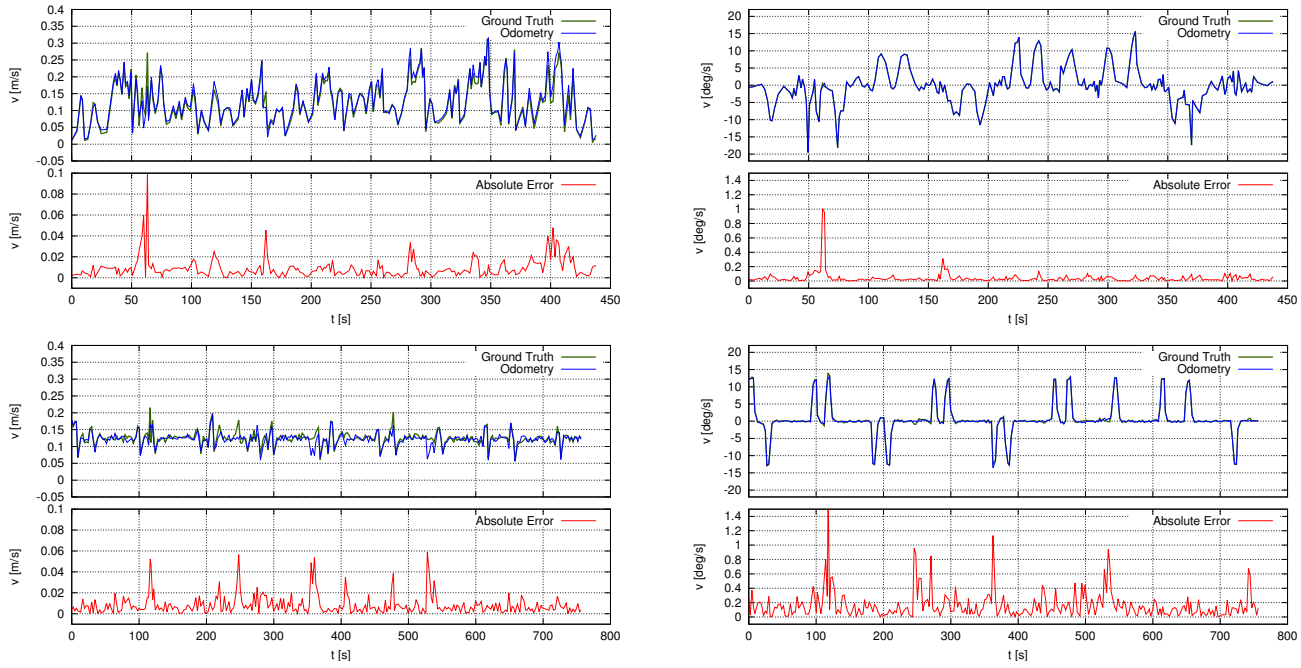
REFERENCES

[1] M. Hildebrandt and F. Kirchner, "Imu-aided stereo visual odometry for ground-tracking auv applications," in *OCEANS 2010 IEEE - Sydney*, May 2010, pp. 1–8.

[2] M. Dunbabin, P. Corke, and G. Buskey, "Low-cost vision-based auv guidance system for reef navigation," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 1, May 2004, pp. 7–12 Vol.1.

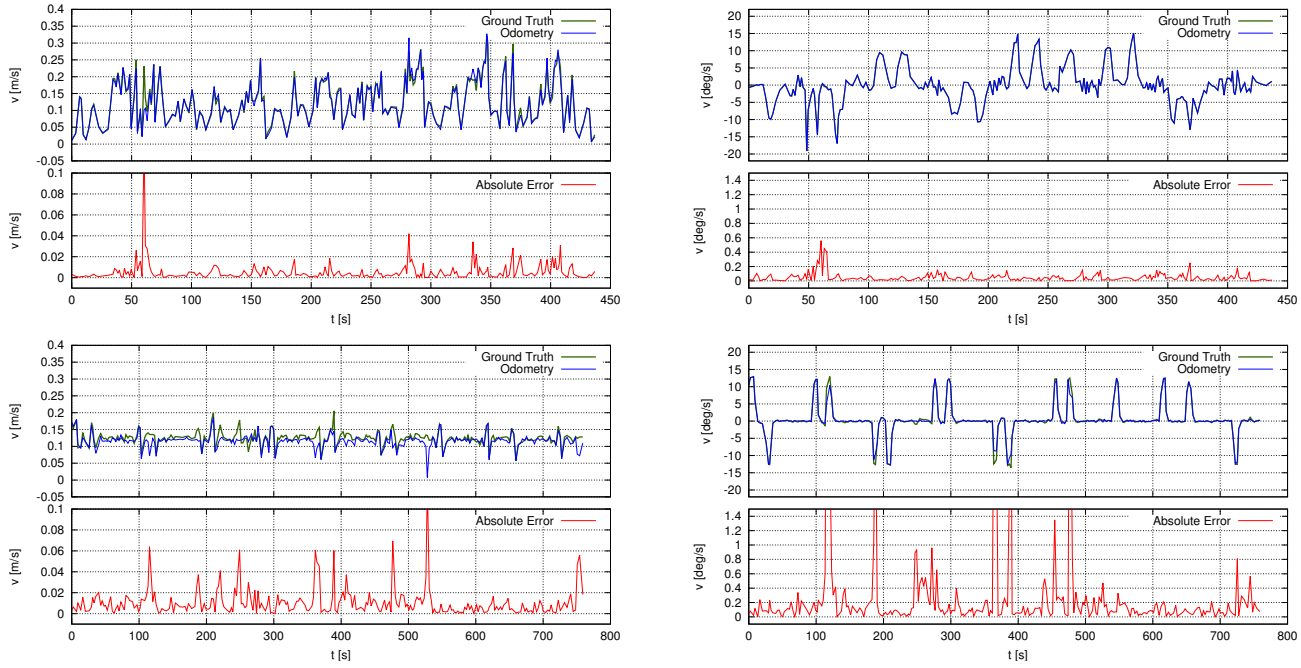
[3] H. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," Ph.D. dissertation, Stanford University, 1980.

[4] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.



(a) (b)

Figure 3. Linear (a) and angular (b) velocity estimates compared to ground truth for the laboratory (top) and the harbour (bottom) experiments for libviso2.



(a) (b)

Figure 4. Linear (a) and angular (b) velocity estimates compared to ground truth for the laboratory (top) and the harbour (bottom) experiments for fovis.

[5] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[6] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, vol. 1, 2006, pp. 430–443.

[7] 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 *International Symposium on Robotics Research (ISRR)*, Flagstaff, Arizona, USA, Aug. 2011, pp. 1–16.

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

2004.

[9] S. da Costa Botelho, P. Drews, G. Oliveira, and M. da Silva Figueiredo, "Visual odometry and mapping for underwater autonomous vehicles," in *Robotics Symposium (LARS), 2009 6th Latin American*, Oct 2009, pp. 1–6.

[10] A. Geiger and J. Ziegler, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, jun 2011.

[11] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," *Asian Conference on Computer Vision*, pp. 25–38, 2010.

[12] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–652.

[13] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies, "Robust and efficient stereo feature tracking for visual odometry," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 39–46.

[14] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3946–3952.

[15] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial intelligence*, vol. 78, no. 1, pp. 87–119, 1995.

[16] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[17] R. Garcia, X. Cufi, and M. Carreras, "Estimating the motion of an underwater robot from a monocular image sequence," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, 2001, pp. 1682–1687 vol.3.

[18] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor, "Mosaic-based navigation for autonomous underwater vehicles," *Oceanic Engineering, IEEE Journal of*, vol. 28, no. 4, pp. 609–624, Oct 2003.

[19] M. Carreras, P. Ridao, R. Garcia, and T. Nicosevici, "Vision-based localization of an underwater robot in a structured environment," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, Sept 2003, pp. 971–976 vol.1.

[20] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

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

[22] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking." in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1508–1511.

[23] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 1, pp. 46–53, 2012.

[24] J. Ferrer, A. Elibol, O. Delaunoy, N. Gracias, and R. Garcia, "Large-area photo-mosaics using global alignment and navigation data," in *Oceans 2007*. IEEE, 2007, pp. 1–9.

[25] A. Geiger. (2013, Mar.) The kitti vision benchmark suite - a project of karlsruhe institute of technology and toyota technological institute at chicago @ONLINE. [Online]. Available: http://www.cvlibs.net/datasets/kitti