IMU-Aided Stereo Visual Odometry for Ground-Tracking AUV Applications

Marc Hildebrandt* and Frank Kirchner*
*Underwater Robotics Department
DFKI RIC Bremen
Germany, 28359 Bremen
Email: marc.hildebrandt@dfki.de

***Abstract*— This paper addresses the problem of AUV navigation by showing the feasibility of a stereo visual-inertial approach to odometry retrieval for an AUV. This information is intended as input for a complete SLAM system. After its classification among many other similar approaches in recent work is shown, the algorithm is described in detail. A number of experiments conducted on synthetic data show the performance in respect to precision and computational cost. As a conclusion, future extensions and applications are briefly discussed.**

I. Introduction

Navigation for AUVs is an ongoing research topic. The problem of providing good on-board localization for autonomous underwater vehicles is of vital importance for its ability to perform complex missions. Current state of the art systems providing high-accuracy localization require external infrastructure (e.g. long/short baseline (LBL/SBL) based systems), power-intensive hardware such as Doppler velocity log (DVL) systems or even both. Development of a self-contained solution for the localization problem which is both robust and versatile is of key importance for the advance in AUV systems. This paper introduces the efforts to create a visual-inertial based solution for the motion estimation problem which is the first step towards a complete localization system. Using the inherent advantages of an on-board-only system without the need for external infrastructure it has to cope with typical problems associated with such approaches: temporal drift due to the accumulation of bias error and reliance on scene structure.

II. Problem Analysis

A number of different visual odometry algorithms were developed over the last decade. Since most AUV systems incorporate a camera, usage of such visual algorithms for the underwater domain was intuitive. In addition to the known issues of this approach from ground-based robotics, a number of problems special to the underwater domain aggravate the situation: full 6DOF, rough, dynamic terrain, poor lighting conditions and limited on-board computational power. It is the authors' impression, that none of the previously conceptualized systems meets all requirements mentioned above. A selection of recent work which has influenced this paper will briefly be presented in the following paragraphs.

Corke, Lobo and Dias [1] make a comprehensive overview of visual-inertial systems for motion estimation. Based on the initial observation that inertial and visual sensors have complementary advantages, their combined usage is strongly advised. Two types of systems are discerned: loosely and tightly coupled systems which differ by the degree of coupling between the inertial and visual systems. This nomenclature will be used in this paper to characterize the different approaches described here.

Moreno et al. [2] describe a visual odometry framework based on a stereo camera system. Their work does not incorporate any other sensors, yet the basic idea is similar to the approach of the visual part presented in this work: selecting features (Lowe's scale invariant feature transform (SIFT) [3] in this case) in a stereo pair, reprojecting them into 3D space using epipolar constraints, tracking the features over a number of frames and using the gathered information in a probabilistic approach for pose change estimation. They propose their work as input for a SLAM approach, since no global reference map or loop closing approaches are employed and thus their algorithm's performance degrades strongly with respect to driven distance.

The Perceptual Robotics Laboratory (PeRL) at the University of Michigan is actively researching underwater AUV-SLAM approaches. Basing their work on Ryan Eustice's visually aided navigation (VAN) approach [4], [5] and extending it with stereo vision capabilities [6] they achieve impressive results. Yet, as the name VAN implies, their vision system is only meant to improve the performance of a fully instrumented vehicle (including a DVL) and the information of navigation sensors is not utilized in their visual motion estimation, but later integrated into the final estimation. Using the nomenclature introduced in [1], this approach can be called loosely coupled visual-inertial system.

Corke et al. [7] describe a series of experiments where visual odometry based navigation on a small Starbug AUV is compared with the results from a LBL tracking system. While the visual odometry algorithm is fairly basic (described in more detail in [8]), the idea of using a system yielding dual measurements which they then compare in a later stage is very promising. Their 'ground truth' measurement is done by a drift-free system (LBL), so the accuracy of their visual odometry can be quantified quite well. For their experiments it lay in the order of 5m after 100m travel or 5% of the traveled distance. Their approach does not incorporate any measure to

978-1-4244-5222-4/10/$26.00 ©2010 IEEE

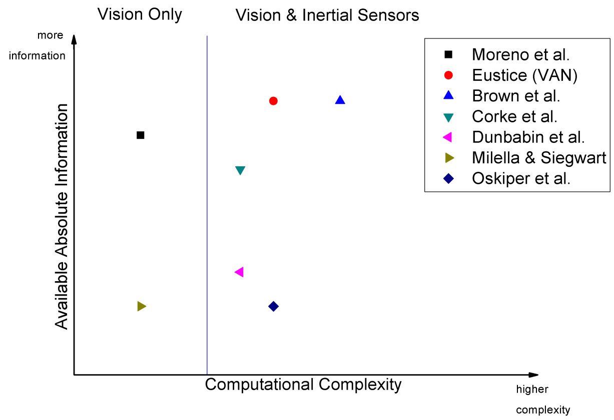

Fig. 1. Comparison of the described algorithms.

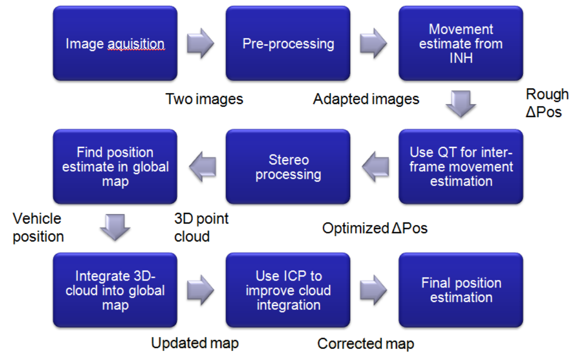

Fig. 2. Outline of the proposed algorithm.

create an upper bound for this error.

Milella and Siegwart [9] describe a basic framework for stereo camera motion estimation using iterative closest point (ICP) approaches. Using a feature descriptor to select salient points in a stereo image pair, tracking them across consecutive image pairs and using ICP for inter-frame matching they achieve reasonable results for land based robots. Even though teir application scenario was land based robotics the idea of combining stereo cameras with ICP approaches strongly influenced this work.

Oskiper et al. [10] present a stereo camera-IMU based system for motion estimation. Their system uses two stereo camera heads mounted in opposite directions to improve robustness in most situations. However, before integration with IMU data, only one of the two stereo camera datasets is selected using a back-projection based metric. After that this system is a typical loosely coupled visual-inertial system.

A key factor seems to be the balance between purely vision-based systems on the one hand, and combinations of vision systems with fully instrumented AUVs (including LBL/DVL systems) on the other hand. The Starbug AUV from CSIRO is one of the few examples for stereo vision enabled AUVs which are used for ground-tracking vision-based odometry. Their design however is a vision only system: it does not incorporate any other sensors. An example for the other extreme is PeRL's modification of the IVER2 AUV with the stereo vision-SLAM-head. As described above, they base their work on the VAN-algorithm by Eustice et al. which requires extensive vehicle sensors systems besides the vision system, and thus only uses visual information to improve the (already very good) position estimates from the other sensors, making the DVL a very important sensor system in their approach. An overview of the different algorithm's attributes in respect to usage of camera/intertial information, available global information and computational complexity is given in figure 1.

III. Proposed Solution

The proposed solution tries to find a good equilibrium between vision-only systems and algorithms expecting a fully instrumented vehicle. It combines visual data from a calibrated stereo camera system with data from an IMU system. The outline is shown in figure 2. As a first step the images are acquired and preprocessed. This preprocessing consists of undistortion and rectification using the calibration data from the cameras. The data from the IMU is doubly integrated to receive a dead-reckoned position estimate. This estimate is prone to drift, magnetic bias and integration errors, but nevertheless serves as initial guess for vehicle movement. After feature extraction in the stereo images this information is used to prune the search space in inter-frame feature matching using the quadrifocal tensor for the stereo camera images. This refined movement estimate already strongly reduces drift errors from the IMU. In the next step the features in the image pairs from the stereo camera system are spatially matched using epipolar constraints. After back-projection into 3D space this yields a 3D-point cloud for every frame. Using the movement estimate from the IMU/quadrifocal tensor, each point cloud can be positioned into a global reference frame. Using a simple distance-based metric a number of spatially close frames from the current estimated position in the global reference frame are selected. Using an ICP (iterative closest point) approach, the new frame is integrated into this local map, yielding the final movement estimate. The result is a globally consistent on-board localization without use of external navigation aids. The following sections will focus on the individual steps.

A. *Dead Reckoning*

Dead Reckoning is the term used for navigation using IMU/DVL sensors in marine systems. Originating in naval systems in the age of sailboats, it was originally consisting of a magnetic compass, a speed measurement device (called chip log) and a time measurement device. By maintaining a selected heading and speed for a fixed time, the traveled path could be plotted onto a map. This technique was used for centuries of marine navigation. The modern version uses an IMU as replacement for the compass, and a DVL instead of the chip log. Time measurement is provided by both devices directly.

The method remains the same: integration of the two measurements over time, where usually the integration interval is much shorter (in the order of 10Hz instead of six times per hour). Additionally a probalistic filter (e.g. Kalman Filter) is usually used to improve the quality of the calculated information. The basic principle is described accurately in [11]. In case of a missing DVL the vehicle speed can be inferred by integrating the accelerometers of the IMU, thus yielding a speed estimate. This estimate usually is of poor quality though, and makes the final position estimation a double integration, which is prone to error. The biggest problems with systems lacking a DVL are drift and error accumulation. The drift is caused by sensor imperfections, and causes the sensors to pick up movement even if the IMU is sitting perfectly still. Error accumulation results from the high integration frequency, which causes small errors to cumulatively bias the result. Taking all these points into account it becomes clear that for an IMU-only system only short term movement estimations are feasible. For purpose of this algorithm the IMU is only used to roughly estimate vehicle movement between two frames, to prune the search space for the quadrifocal tensor estimation.

B. Feature Extraction

A key step in the visual odometry approach is the extraction of spatially and temporal stable salient features from the stereo images in each frame. For this purpose Bay's SURF (speeded-up robust features) feature detector/descriptor is used [12]. Based on Lowe's SIFT (scale invariant feature transform) [3] SURF features combine the robustness of SIFT features with strongly improved runtime characteristics.

The OpenCV implementation used in this algorithm yields the spatial keypoints together with a 128 dimensional descriptor for each keypoint. The parameters for the SURF feature extraction are adjusted according to the number of found features in the last image using a simple proportional controller. This assures a relatively constant number of features even with changing scene structure and works very well.

C. Quadrifocal Tensor Estimation

The quadrifocal tensor describes the geometric relations between two stereo pairs [13]. Its computation requires information about the projection of a number of points in the scene (at least 6, as stated in [14]) onto all four images, meaning that the feature correspondences of these point projections have to be known. Using this information the tensor can be calculated. Since in real data noise is to be expected (either wrong correspondences or non-accurate points among the images), a probabilistic refinement of the results is advisable (e.g. the RANSAC-algorithm, see [15]). Once the quadrifocal tensor is computed the camera matrices can be directly recovered (details see [14]). For the purpose of the algorithm described here the number of points used as input for the computation lies in the order of 100 points. The information from the IMU dead-reckoning are used to prune the search space for the establishment of feature correspondences, yielding very fast and good results. The feature correspondences in two image pairs are shown in figure 3.

D. Frame Selection

The task of the frame selection routine is to find a rough global position estimate of the vehicle in respect to the starting point of the visual odometry. This history of the visual odometry can be treated as a rough global map. Using this position estimate, frames in close proximity are selected from the map. As measure for frame proximity the euclidean distance between two frames was selected. This is efficiently computable, even for big sets of candidate frames. The 3D data from the resulting set of frames is combined into the 'model' point cloud for the ICP. The threshold distance has to be selected carefully, since it will have a strong impact on the overall algorithm's performance. Too high values will result in large 'model' datasets, and long computation times of the ICP. In case the value is selected too small, the benefits of a global re-integration will not be applicable, in the worst case resulting in the previous frame as the only matching candidate.

E. ICP Optimization

The ICP algorithm introduced by Besl [16] can be used to align two sets of 3D points of the same object. It is widely used to align laser-scans made from different perspectives, e.g. in robot navigation or geometric reconstruction [17]. It takes two coarsely aligned 3D point clouds and calculates the 4x4 transformation matrix which minimizes the distance between neighboring points. This is done iteratively, while usually a threshold for near point detection is lowered during the process. Typical for laser-scans no further descriptive information is available for the individual points in the cloud. This is different in the data available for the algorithm described in this paper: each 3D point is associated with a feature descriptor which has already been matched to the previous frame. Additionally utilizing this information the first iteration of the ICP can use these correspondences as initial guess instead of just relying on Euclidean distances. This has the strong advantage that a good initial match of the two point clouds is established in this phase. The downside is that the feature descriptor is illumination dependent. This means, that in case of re-visiting of a vista the same keypoint may have a different descriptor - resulting in a poor matching score. To avoid this the information from the descriptor is only used in the first iteration, all further iterations use the euclidean distance as sole minimization criteria.

IV. Experiments

The proposed algorithm was tested using several synthetic datasets with ground-truth. The synthetic datasets were created using a state of the art modeling, animation and rendering program (3ds MAX) providing realistic data and ground-truth position information. The scene used as basis for all input data is shown in figure 5. In a first step the impact of different lighting conditions was evaluated and a configuration with four discrete lights illuminating the scene relatively homogeneously

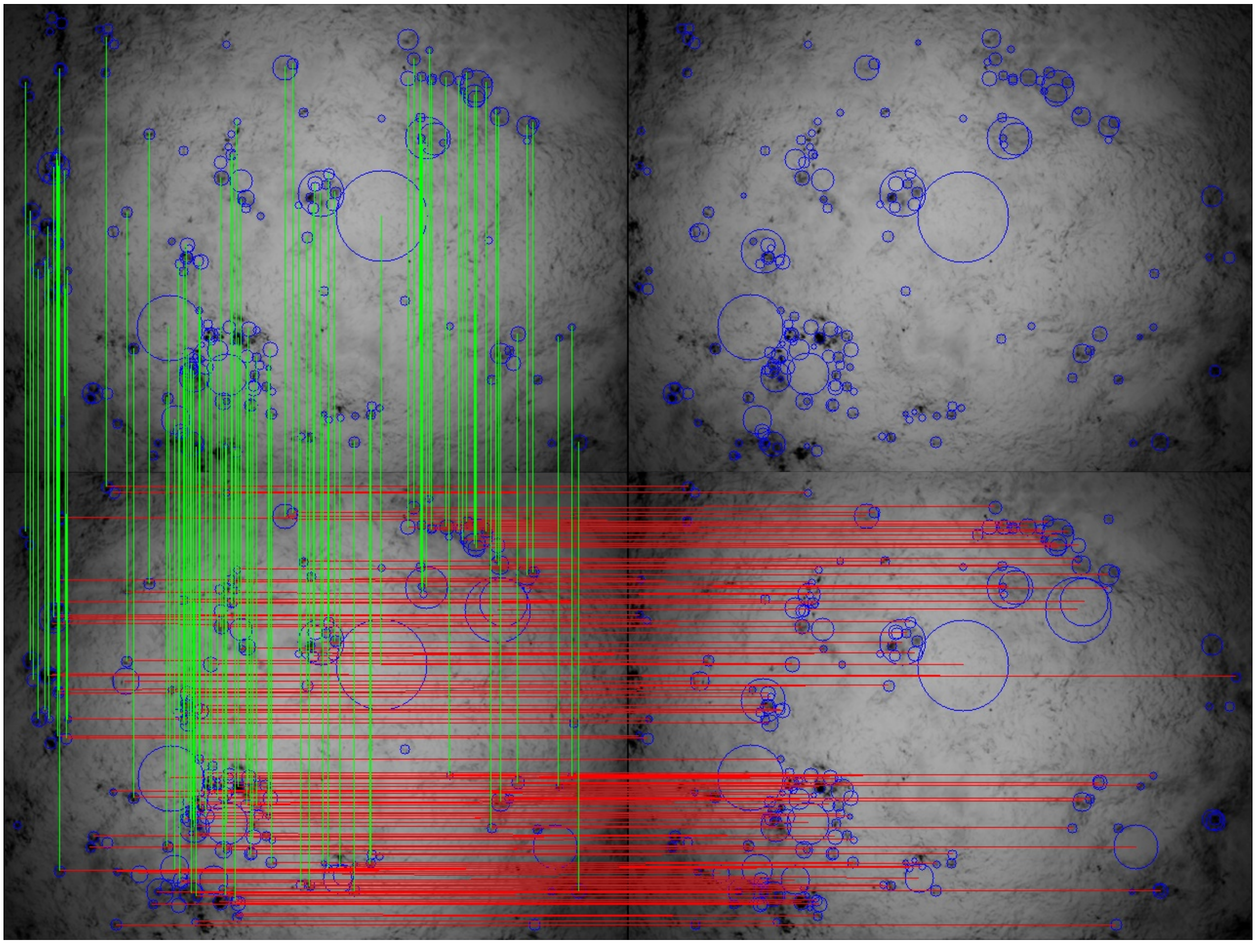

Fig. 3. The relations inside to subsequent image pairs. The extracted features are shown as blue circles. The stereo correspondences are shown as red lines (only for the lower pair), which are horizontal since the images are rectified. The green lines show the inter-frame correspondences (only shown for the left image pair), which are vertical since the cameras moved in a straight line between the two frames.

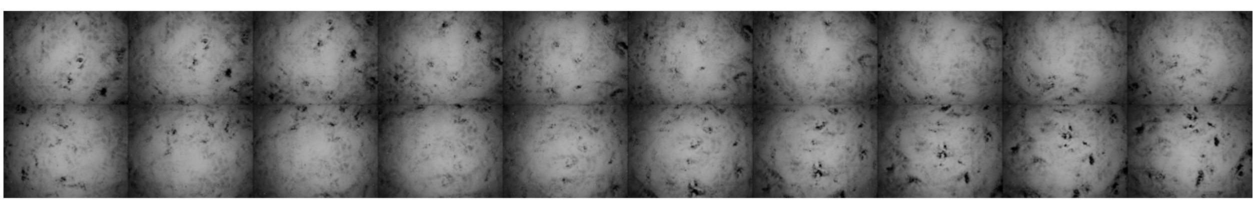

Fig. 4. A sequence of images from the test dataset showing one of the u-turns. It shows 20 frames from the left camera, using every 10th frame.

(especially at the border areas) was chosen. Two datasets were produced: one 'plain' dataset with no visual noise (from now on referred to as 'plain' dataset) in the images and a second dataset with artificial marine snow (referred to as 'snow' dataset). The marine snow is simulated by five layers of randomly moving particles of varying size and speed, each layer consisting of 100.000 particles of which approximately 1200 are visible in each image. This fairly closely resembles real marine snow effects under calm sea conditions. The rendering of the datasets took about a month on a 2.67 GHz Intel i7 PC. Each dataset consists of 2x3000 images in the resolution 640x480 pixels taken from two virtual cameras with a field of view of 60 (a short sequence of these images is shown in figure 4). The baseline of this virtual stereo camera set is 32cm, yielding an approximately 90% image overlap at working distances. The virtual cameras/lights were traveling in a survey pattern, meandering in five consecutive s-shapes over an area of 400x250 meters, covering a distance of approximately 4.5 kilometers. The mean distance from the floor was 4 meters. The resulting inter-frame overlap was about 96%. On a real vehicle this would reflect image processing rates in comparison to vehicle speed, and thus represents an important practical factor. On the algorithm side the amount of information used for ICP optimization was varied. Either a section of the global map was used (as described in III-D) or simply the previous frame. For the rest of this paper these

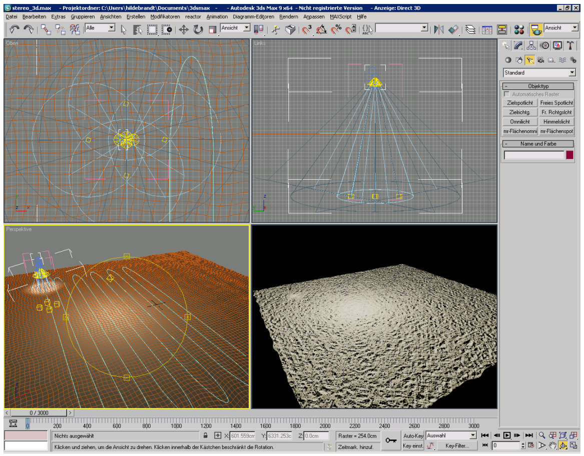

Fig. 5. The scene used to create the synthetic dataset. Note that an additional omni directional light was added to create this view of the scene.

two variants are referred to as 'local' and 'global' algorithms, respectively.

A. Performance Analysis

The overall results are very promising. The absolute error after traveling 4.5 kilometers lies in the order of 5 meters (see figure 7 for a graphical representation). Most of this error originates from poor orientation estimation, especially at the turning points of the trajectory. One reason for this behavior can be found in the generation of the synthetic dataset: While the translational speed of the vehicle is kept constant, the rotational speed is not limited. This results in very high rates of turn at the turning points, and as final consequence in poor estimation. Future datasets will take care of this effect by limiting rate of turn. This fact will also impact the control of the real AUV. Unfortunately the fact, that all rotation is captured only by 20 images (at each turning point) made a variation in image density impossible - the intended usage of only every 10th image would have resulted in only 3 images used for a 180 rotation - which is not feasible.

Typical for a visual odometry approach is the unbound cumulative error (see figure 6). The error/distance quotient of 10^{-3} still is very low, about 50 times lower as in [7], [9] and [10]. Since they were operating in a real environment higher error rates are to be expected, but an improvement of this magnitude is still a good starting point.

One of the key properties of the algorithm became apparent when comparing the results of the 'plain' dataset with the results from the 'snow' dataset. The noise has virtually no effect, the accuracy stays coherent in all measured quantities (see table I for details). This behavior is explained first by the feature detector/descriptor, which rejects small features. The second explanation is the ICP's ability to iteratively reject points which do not fit the overall transformation estimate by decreasing the threshold value for the local search.

TABLE I

PERFORMANCE COMPARISON FOR THE ALGORITHM WITH DIFFERENT PARAMETERS.

Algorithm	Dataset	Mean Error	Standard Deviation	Max. Error
Local	Plain	4.51114m	3.31759m	13.87669m
Local	Snow	5.07538m	3.28339m	13.98115m
Global	Plain	2.93658m	1.94536m	8.66041m
Global	Snow	3.4725m	2.30304m	9.41607m

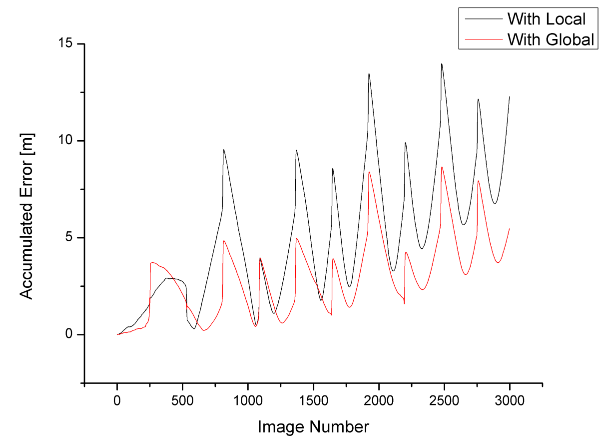

Fig. 6. Distance error between visual odometry and ground truth for local and global algorithm variants.

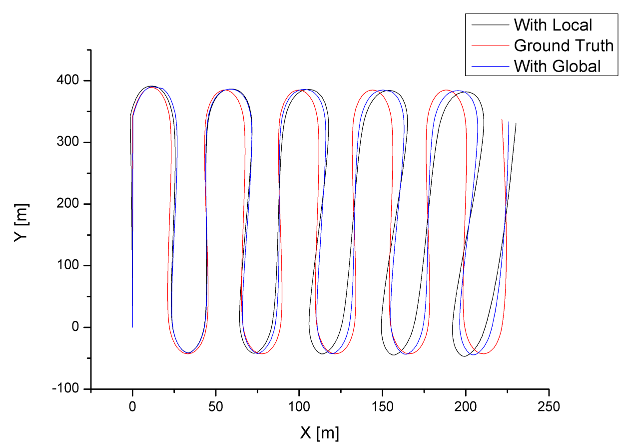

Fig. 7. Comparison between the performance of visual odometry with and without a global map.

TABLE II

AVERAGE COMPUTATION TIMES FOR THE MAJOR STEPS OF THE ALGORITHM FOR LOCAL-ONLY (APPROX. 133 POINTS IN THE ICP) AND GLOBAL OPTIMIZATION (APPROX. 2647 POINTS IN THE ICP).

Algorithm	Feature extraction	QT estimation	ICP	Overall
Local	224ms	5ms	37ms	266ms
Global	221ms	5ms	3165ms	3391ms

B. Runtime Analysis

The current version is not optimized for real-time performance. It is a simple single-threaded c++ application. Nevertheless all algorithms were chosen with the future aim of real time-capability in mind. Table II shows the current computation times on a 2.67 GHz Intel i7 Processor. In order to achieve a performance of at least 15 images per second, the maximum overall duration per calculation is allowed to be 66ms, so an optimization in the order of one magnitude is necessary. The most work has obviously to be done for the feature extraction and ICP steps. Image acquisition and preprocessing is believed to be optimizable by normal means, parallelization and pre-calculation. The current implementation uses the OpenCV 2.0 SURF implementation [18], which can be improved by utilizing a graphics processing unit (GPU) for this calculation. Cornelis and Van Gool have created a version of the SURF descriptor on a GPU which can exceed frame rates of 100 frames per second (FPS) for images of 640x480 pixels [19]. This would reduce the time required for computation below the 10ms mark and at the same time offload the CPU. The target vehicle for the algorithm will incorporate a GPU (see section V-B). The current implementation of the ICP algorithm uses two arrays to store the required information for matching. This is sub-optimal, since any search operation on an array has the complexity O(n) when the size of the array is n. Tree structures, such as k-dimensional trees [20] have significantly better search complexity (in the order of O(n*log(n))). This together with multi-threading approaches is believed to improve the runtime far enough for the real-time requirements described above.

V. CONCLUSION

This paper described a complete stereo visual odometry algorithm which makes use of inertial data from an IMU. The benefits of this approach were described and experimental data using two synthetic datasets was shown. The high accuracy encourages further applications of this idea. The runtime analysis shows potential for tweaking which should allow real-time application.

A. Target Vehicle

The algorithm described in this paper is intended for a vehicle equipped with a stereo camera and an inertial navigation system (typically an IMU). In order to verify the results from the visual motion estimation algorithm, the vehicle which is built in the DFKI's CUSLAM project (for details see www.dfki.de/robotics) will have a complete AUV sensor system, including a DVL, FOG extension for the IMU and LBL tracking capabilities. The idea is to use established algorithms together with the aforementioned sensors to create ground-truth measurements in real environments while in-parallel calculating the vehicle's motion on basis of the stereo camera system and IMU. The results of both approaches can then be compared, and a meaningful and accurate measure for the feasibility of visual-inertial approaches can be given. The AUV will have strong processing capacities, sporting three dual-core CPUs in addition to a GPU. This processing power ensures its use as verification and experimentation testbed. The camera system will consist of two high-definition cameras with extremely sensitive image sensors. An early concept of the AUV is shown in figure 8.

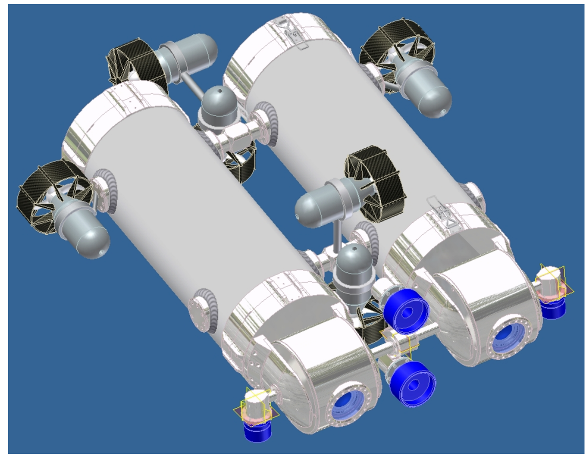

Fig. 8. CAD drawing of the target vehicle.

B. Future Work

The algorithm described in this paper will be tested on a real vehicle (see section V-A) under various environmental conditions by the end of 2010. The final aim is to extend the algorithm to a fully-fledged SLAM system for small AUVs, tested and verified in the field. In order to achieve this aim the visual odometry approach described in this paper will be used as input for a SLAM system, similar to Eustice's approach but with the stronger focus on visual-inertial data.

For the approach and any other algorithms extending it to work in real-time, a number of optimizations have to be conducted. The processing power of the vehicle's GPU will be tapped for feature extraction, and a second CPU may be used for the SLAM part if deemed necessary.

The cameras are currently modeled as pointing straight down, facing the sea floor. Experiments will be conducted changing the camera's angle to face forward to a certain degree, since this is expected to improve overall performance by extending the information gained from motion to another dimension.

The SURF features used so fare have yielded good results on the synthetic data. Since it has been mentioned by Nicosevici

[21], that other detectors/descriptors may perform better in underwater environments, these will be evaluated as well. Especially the maximally stable extremal regions (MSER) approach by Matas et al. [22] seems well applicable for environments with sparse salient features.

Acknowledgment

The authors would like to thank all colleagues at the DFKI Bremen for their support and feedback. The work presented here is part of the CUSLAM project, funded by the German Ministry of Economics (BMWI), grant No. 03SX290.

References

[1] P. Corke, J. Lobo, and J. Dias, "An Introduction to Inertial and Visual Sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, June 2007. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364907079279

[2] F. A. Moreno and J. L. Blanco, "An Efficient Closed-Form Solution to Probabilistic 6D Visual Odometry for a Stereo Camera," pp. 932–942, 2007.

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. vol, pp. 60no2pp91–110, 2004.

[4] R. M. Eustice, H. Singh, and O. Pizarro, "Visually Augmented Navigation for Autonomous Underwater Vehicles," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, April 2005. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4625213

[5] R. M. Eustice, "Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles," PhD, MIT, April 2008.

[6] H. C. Brown, A. Kim, and R. M. Eustice, "An Overview of Autonomous Underwater Vehicle Research and Testbed at PeRL," *Marine Technology Society Journal*, vol. 43, no. 2, pp. 33–47, May 2009. [Online]. Available: http://openurl.ingenta.com/content/xref?genre=article&issn=0025-3324&volume=43&issue=2&spage=33

[7] P. Corke, C. Detweiler, M. W. Dunnigan, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with Underwater Robot Localization and Tracking," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 4556–4561, April 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4209799

[8] M. Dunbabin, P. Corke, and G. Buskey, "Low-cost vision-based AUV guidance system for reef navigation," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, no. April. Ieee, 2004, pp. 7–12. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1307121

[9] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," *In IEEE International Conference on Computer Vision Systems*, 2006.

[10] T. Oskiper, Z. Zhu, S. Samarasekera, R. Kumar, and S. Corporation, "Visual Odometry System Using Multiple Stereo Cameras and Inertial Measurement Unit," 2007.

[11] G. Zhen and S. Feng, "Research on integrated navigation method for AUV," *Marine Science*, vol. 4, no. 2, pp. 34–38, 2005.

[12] H. Bay, a. Ess, T. Tuytelaars, and L. Vangool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314207001555

[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, 2nd ed*, 2004, vol. 0521540518.

[14] R. I. Hartley, *Computation of the Quadrifocal Tensor*. London: Springer-Verlag, 1998, vol. 1406, pp. 20–35.

[15] M. A. Fischler and R. C. Bolles, "Paradigm for Model," *Communications of the ACM*, vol. 24, no. 6, 1981.

[16] P. J. Besl, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[17] B. A. Shahrabi, "Automatic Recognition and 3D Reconstruction of Buildings through Computer Vision and Digital Photogrammetry," Dissertation, University of Stuttgart, 2000.

[18] Intel and WillowGarage, "OpenCV Library." [Online]. Available: http://sourceforge.net/projects/opencvlibrary/

[19] N. Cornelis and L. Van Gool, "Fast scale invariant feature detection and matching on programmable graphics hardware," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Ieee, June 2008, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4563087

[20] D. Lee and C. Wong, "Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees," *Acta Informatica*, vol. 9, no. 1, 1977. [Online]. Available: http://www.springerlink.com/index/10.1007/BF00263763

[21] T. Nicosevici and R. Garcia, "Online Robust 3D Mapping Using Structure from Motion Cues," in *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*. Kobe: IEEE, April 2008, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4531022

[22] J. Matas, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0262885604000435

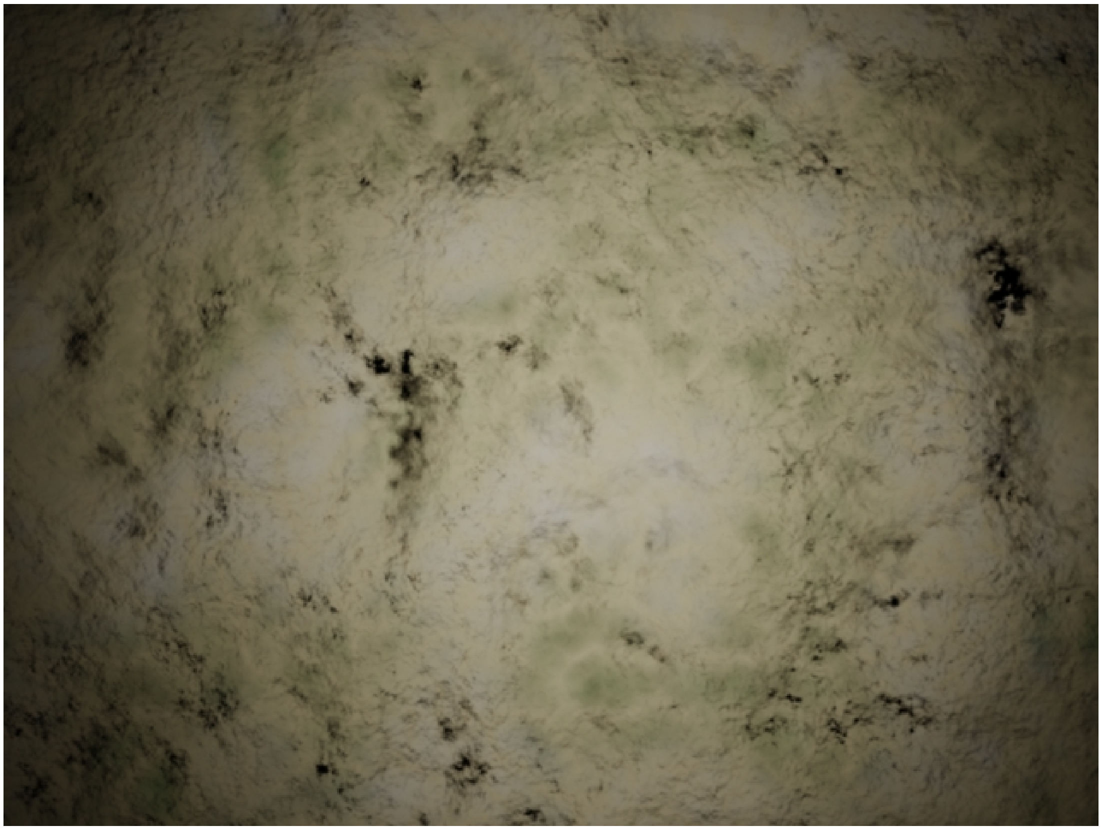

Fig. 9. An image from the sequence.

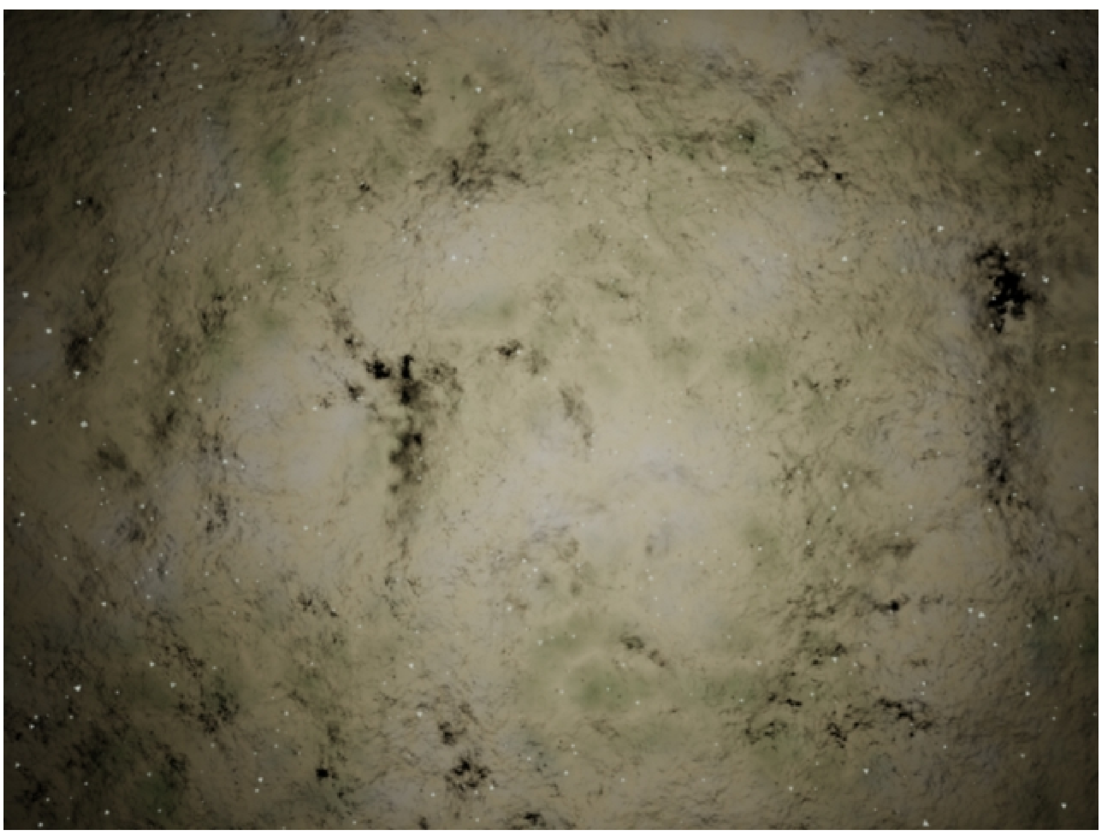

Fig. 10. The same image as above, but with the marine snow simulation visible.