2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Maximum Likelihood Mapping with Spectral Image Registration

Max Pfingsthorn Andreas Birk Sören Schwertfeger Heiko Bülow Kaustubh Pathak

Abstract— **A core challenge in probabilistic mapping is to extract meaningful uncertainty information from data registration methods. While this has been investigated in ICP-based scan matching methods, other registration methods have not been analyzed. In this paper, an uncertainty analysis of a Fourier Mellin based image registration algorithm is introduced, which to our knowledge is the first of its kind involving spectral registration. A covariance matrix is extracted from the result of a Phase-Only Matched Filter, which is interpreted as a probability mass function. The method is embedded in a pose graph implementation for Simultaneous Localization and Mapping (SLAM) and validated with experiments in the underwater domain.**

I. INTRODUCTION

Research interest in maximum likelihood mapping has increased significantly in the last years. Here, maps are usually represented as graphs where vertices represent poses of the robot and edges contain pairwise localization information, generated for example by odometry or scan matching.

Several optimization algorithms for graph-based maps exist [1], [2], [3], [4], [5], [6]. The general idea is also used in a Visual SLAM technique called Bundle Adjustment [7], which not only optimizes camera poses, but also their calibration parameters at the same time.

The main challenge for meaningful probabilistic mapping is extracting uncertainty information from measurements. This is generally not a problem when dealing with a vehicle motion model and odometry. However, extracting uncertainty information from sensor data registration methods is not that simple.

There has been some investigation into ICP-based methods in order to compute a covariance matrix for the result [8], [9], [10]. Other registration methods have not been investigated in this fashion.

In the following, an uncertainty analysis of the iFMI image registration algorithm [11], [12] developed at Jacobs University Bremen is presented. To our knowledge, this constitutes the first uncertainty analysis of any spectral registration method. The process is then integrated into a maximum likelihood mapping framework and used to build several maps from different underwater image sequences.

II. IMAGE REGISTRATION AND UNCERTAINTY EXTRACTION

A. The iFMI image registration algorithm

The iFMI registration algorithm [11], [12] uses a Phase-Only Match Filter (POMF) to determine rotation/scaling and translation in two steps. This correlation approach makes use of the fact that two shifted signals having the same spectrum magnitude are carrying the shift information within its phase.

$$f(t-a) \circ\!\!-\!\!\bullet\, F(\omega)e^{i\omega a} \tag{1}$$

When both signals are periodically shifted the resulting inverse Fourier transformation of the phase difference of both spectra is an ideal Dirac pulse. This Dirac pulse indicates the underlying shift of both signals which have to be registered.

$$\delta(t-a) \circ\!\!-\!\!\bullet\, e^{i\omega a} \tag{2}$$

The resulting shifted Dirac pulse deteriorates with changing signal content of both signals. As long as the inverse transformation yields a clear detectable maximum this method can be used for matching two signals (see figure 1 for an example).

This relation of the two signal phases is used for calculating the Fourier Mellin Invariant Descriptor (FMI). The next step for calculating the desired rotation parameter exploits the fact that the 2D spectrum (eq. 4) rotates exactly the same way as the signal in the time domain itself (eq. 3):

$$s(x,y) = s_R\left[R(\alpha)\,(x,y)^T\right] \tag{3}$$

$$|S(u,v)| = \left|S_R\left[R(\alpha)\,(u,v)^T\right]\right| \tag{4}$$

where α is the rotation angle, and $R(\alpha)$ is the corresponding 2D rotation matrix. This fact is used to decouple rotation and translation, since the spectrum magnitudes of two shifted (i.e. translated) signals are identical. Two consecutive camera images do not completely fulfill this relation, but the image contents (i.e. the signals) are expected to be similar enough such that this assumption still holds. The following steps to recover the rotation and scaling parameters are applied to the spectrum magnitudes of the input images, not the images themselves.

For turning the rotation into a signal shift the magnitude of the signal spectrum is simply re-sampled into polar coordinates. For turning a signal scaling into a signal shift several steps are necessary. The following Fourier theorem shows the relations between a signal scaling and its spectrum.

$$f(\frac{t}{a}) \circ\!\!-\!\!\bullet\, |a|F(a\omega) \tag{5}$$

This relation can be utilized in combination with another transform called Mellin transform which is generally used for the calculation of moments:

$$V^M(f) = \int_0^\infty v(z)z^{i2\pi f-1}dz \tag{6}$$

All authors are with the School of Engineering and Science, Jacobs University Bremen, Campus Ring 1, 28759 Bremen, Germany a.birk@jacobs-university.de

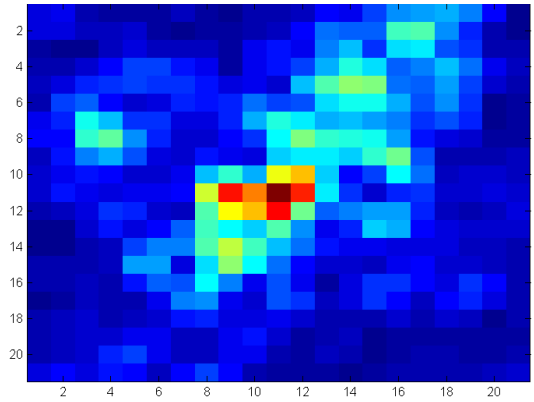

Fig. 1. An example parameter space showing a clear maximum corresponding to the registration result. The peak is not very pronounced because of ambiguous structures in the registered data.

With two functions $v1(z)$ and $v2(z) = v1(az)$ differing only by a dilation, the resulting Mellin transform with substitution $az = \tau$ is:

$$\begin{aligned} V_2^M(f) &= \int_0^\infty v1(az) z^{i2\pi f-1} dz \\ &= \int_0^\infty v1(\tau)(\frac{\tau}{a})^{i2\pi f-1} \frac{1}{a} d\tau \\ &= a^{-i2\pi f} V_1^M(f) \end{aligned} \quad (7)$$

The factor $a^{-i2\pi f} = e^{-i2\pi f ln(a)}$ is complex which means that with the following substitutions

$$\begin{aligned} z = e^{-t}, ln(z) = -t, dz = -e^{-t}dt, \\ z \to 0 \Rightarrow t \to \infty, z \to \infty \Rightarrow t \to -\infty \end{aligned} \quad (8)$$

the Mellin transform can be calculated by the Fourier transform with logarithmically deformed time axis:

$$\begin{aligned} V^M(f) &= \int_\infty^{-\infty} v(e^{-t}) e^{-t(i2\pi f-1)}(-e^{-t})dt \\ &= \int_{-\infty}^{\infty} v(e^{-t}) e^{-i2\pi f t} dt \end{aligned} \quad (9)$$

Now the scaling of a signal using a logarithmically deformed axis can be transfered into a shift of its spectrum. Finally, the spectrum's magnitude is logarithmically re-sampled on its radial axis and concurrently the spectrum is arranged in polar coordinates exploiting the rotational properties of a 2D Fourier transform as described before. Scaling and rotation of an image frame are then transformed into a 2D signal shift where the 2D signal is actually the corresponding spectrum magnitude of the image frame. This intermediate step is called the FMI descriptor.

The following is a sketch of the overall algorithm. The POMF is calculated as follows:

1) calculate the spectra of two corresponding image frames
2) calculate the phase difference of both spectra
3) apply an inverse Fourier transform of this phase difference

The following steps are taken for a full determination of the rotation, scaling and translation parameters:

1) calculate the spectra of two corresponding image frames
2) calculate the magnitude of the complex spectral data
3) resample the spectra to polar coordinates
4) resample the radial axes of the spectra logarithmically
5) calculate a POMF on the resampled magnitude spectra
6) determine the corresponding rotation/scaling parameters from the Dirac pulse
7) re-size and re-rotate the corresponding image frame to its reference counterpart
8) calculate a POMF between the reference and re-rotated/scaled replica image
9) determine the corresponding x,y translation parameters from the Dirac pulse

B. Uncertainty information from registration results

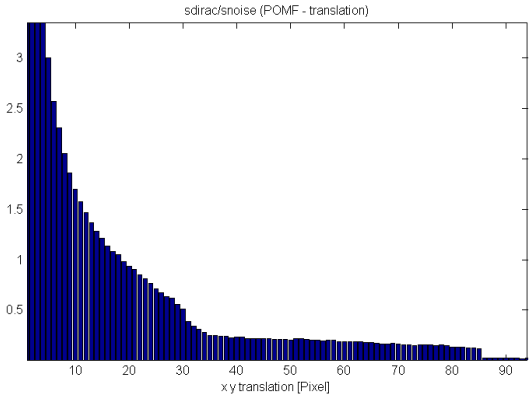

Fig. 2. Values of *PNR* (eq. 13) as translation difference between two registered images increases.

Ambiguities within matched structures will have corresponding indications in the parameter space. The result can be a smeared peak, up to multiple peaks for certain repeating structures (see figure 1). In this work, it is assumed that the globally maximum peak is the correct one. While other peaks may exist, they are considered irrelevant. More complex multi-modal distributions are not usually employed in SLAM research, and are thus neglected here.

To express the matching uncertainty, a $2x2$ covariance matrix is fitted to a neighborhood of size K around the registration result (x^*, y^*).

$$C_{xy} = \sum_{x=x^*-K/2}^{x^*+K/2} \sum_{y=y^*-K/2}^{y^*+K/2} dirac(x,y)[x\ y]^T[x\ y] \quad (10)$$

where $dirac(x,y)$ is the intensity in the parameter space for the translation (x,y).

It is assumed that the parameter space has been normalized and thus can be treated as a discrete probability mass function. Since covariance is defined as the expected value of the squared distance to the mean, a heuristic in eq. 10 weights the squared distance in the parameter space by the normalized intensity.

A covariance for the scaling and rotation parameters is computed analogously. The only difference is that this covariance has to be scaled according to the rotation and scaling resolution of the iFMI descriptor.

An important piece of information (e.g. for loop detection) is whether the registration was successful or not. Here the ratio between the intensity of the maximum, which is supposed to be the dirac peak indicating the correct parameter, and the surrounding range is taken as a measurement whether the registration was successful. After a normalization (eq. 11), the maximum search yields the intensity s_{dirac} in the parameter space signifying the registration result (eq. 12). A peak to noise ratio *PNR* (eq. 13) is defined, which is similar to the signal to noise ratio.

$$dirac_{norm}(x,y) = \frac{dirac(x,y)}{\sum_{x=0}^{X}\sum_{y=0}^{Y} dirac(x,y)} \tag{11}$$

$$s_{dirac} = \max_{x,y}(dirac_{norm}(x,y)) \tag{12}$$

$$PNR = \frac{s_{dirac}}{1 - s_{dirac}} \tag{13}$$

Figure 2 shows the ratio for different translations of two image frames. For two identical images this ratio would be infinity since the peak (eq. 12) would be one. As the image pair drifts apart from each other, interfering structures generate noise outside the peak and the peak itself decreases in amplitude. The specific example presented in the graph shows a continuous exponential decrease of this ratio up to a translation of 86 pixels. At this point the registration failed, which results in a significant drop in the ratio. This behavior is typical and leads to a general threshold for *PNR* of 0.2. Using this threshold, the ratio *PNR* yields a clear indication that the registration was successful or not.

III. POSE GRAPH MAPPING

A. Data Structure

The pose graph is a very popular map representation in maximum likelihood mapping. Such a graph based map has been used extensively in the recent literature [13], [14], [15], [1], [2], [3], [4], [5], [6], [16].

The basic idea behind the pose graph is that it is possible to represent a map as localized sensor data. Each vertex in the graph represents a pose where the robot collected mapping data, and consequently contains data which needs to be situated in a global coordinate system in order to build a map. This can be either laser scans (2D [16], [13] or 3D [15]), other range data such as stereo camera images, landmark observations, surface models (e.g. planes [17]), metric sub-maps in a hierarchical mapping approach (e.g. occupancy grids [18]), or in the case of this paper, overhead monocular camera images.

Each edge contains constraints that relate two (or possibly more in case of a hypergraph) poses. In the literature, these constraints are usually full pose constraints, even though more general constraints are feasible as well. Full pose constraints usually originate from odometry or data registration techniques, such as scan matching or, as in this paper, image registration. It is important that each constraint contains confidence information in the form of a covariance matrix. Other forms of confidence information are also possible, however optimization algorithms for them do not exist yet.

Most pose graph optimization algorithms use the Mahalanobis distance as the objective function [3], [15], [16].

$$\sum_c f_c(X)^T \; C_c^{-1} \; f_c(X)$$

where X is the vector of global poses, $f_c(X)$ is the function for constraint c, and C_c is its covariance matrix. In the following discussion of mapping results, this metric will be used to show that the error in the graph decreased.

B. Framework and Integration

The maps in the following section were generated with a mapping framework developed at Jacobs University Bremen, which implements the general pose graph map representation.

This framework was designed to be sensor data agnostic in a way that abstracts how data is registered, stored within a serializable data structure, and finally rendered into a composite map. It is therefore straightforward to reuse several graph map optimization and generic loop closing algorithms with a new kind of sensor data. The framework only requires interface implementations that can reliably register and render that new kind of data. By exactly this extension paradigm, it was possible to generate complete maps from overhead camera data quickly.

The optimization of a pose graph depends only on the constraints present in the graph and is thus almost universally applicable, no matter what algorithm or sensor generated the constraints or what kind of data is present at the vertices. However, loop closing is another matter, it heavily depends on the data used for mapping. In terms of mapping framework integration of the iFMI registration described above, the $\frac{s_{dirac}}{n_{dirac}}$ ratio is used to this end. This ratio can be computed for the rotation/scaling parameters for no significant additional cost from the pre-computed fourier transforms and a suitable threshold is easily found as shown above. Thus, it is possible to check potential registration pairs that lie within a neighborhood of each other quickly.

In this paper, we use both the inspection of the specific registration result as well as manual loop detection using a special purpose GUI. This application also allows real time visualization of the mapping result during video capture. The pose graph maps in the following section are optimized with the open source TORO optimizer [2].

IV. EXPERIMENTS AND RESULTS

A. Cold Corals

The first data set discussed here was recorded off the coast of Sweden. It was collected by a remote operated vehicle (ROV) and contains a video stream showing cold water corals.

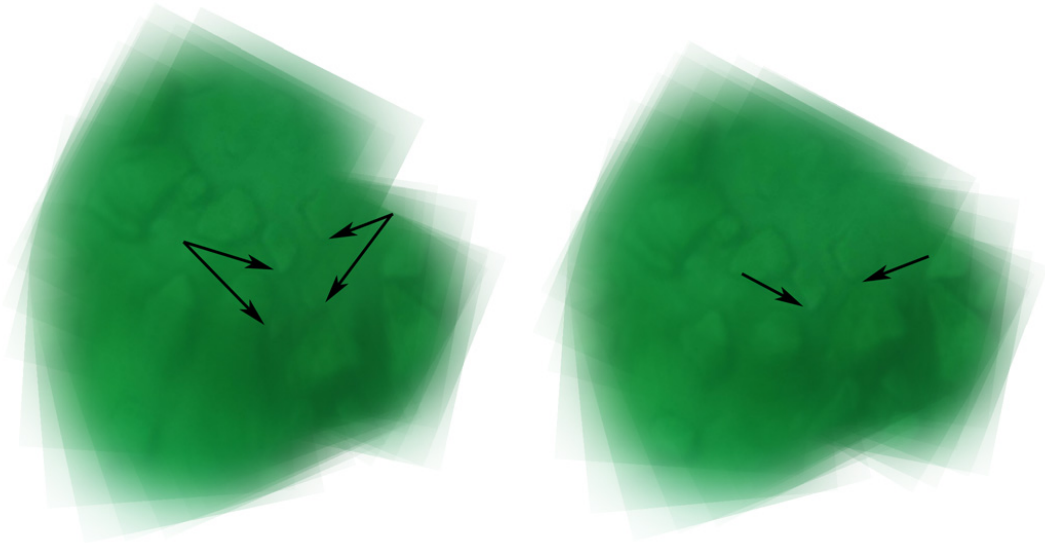

Fig. 3. Complete image map generated from the testing pool data set, before (left) and after (right) graph optimization. Note the blurred lower left part of the map before optimization and the misalignment of two rocks (arrows). After optimization, the map is visibly improved and the rocks are aligned properly.

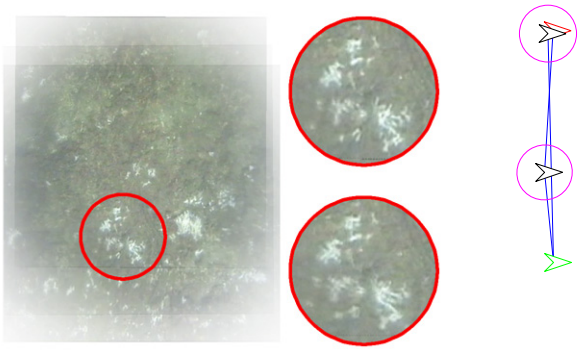

Fig. 4. Short sequence of three images from the cold water corals data set. The detail on the bottom shows the merged image before optimization, and the one above shows the much clearer structures of the optimized image map. The corresponding pose graph with a slightly misaligned top is shown on the right. Note the global covariance matrices plotted as ellipses in magenta.

The data is quite challenging for an image registration algorithm as fish, plankton, and the ground itself all give rise to different flow patterns. Our results show that the iFMI registration algorithm is both robust against such noise and generates covariance matrices usable for maximum likelihood mapping with the pose graph.

Figure 4 shows a set of three images which were mutually registered. All three images are rendered transparently on top of each other using a gaussian kernel. This is clearly shown in the resulting pose graph shown on the right. Each of the three vertices is connected to the two other vertices. The global covariances of the poses are plotted in magenta. Note that the first node (here shown in green) does not have any positional uncertainty as it is defined as the origin of the global map coordinate frame.

The final result after pose graph optimization is shown on the left of figure 4. Even before optimization, the error is only minimal, as seen in the close up (bottom center) and in the pose graph on the right. The top vertex in the graph is actually projected to two different poses, depending on the path taken to accumulate the registration results. Such conflicts are visualized in red.

The detail view in the middle of figure 4 shows how the slight error that is accumulated even over two steps can be mitigated by maximum likelihood mapping. Before optimization, distinguishing features are blurred and smudged in the composite image, shown in the bottom center image. Once the underlying pose graph is optimized, the highlighted features are much crisper since the images are aligned better.

In addition, the significant drop in the objective function used during optimization shows quantitively that the map improved. The squared Mahalanobis distance over all edges before optimization is 0.611929. After optimization, it is reduced to $3.05721 \cdot 10^{-32}$. This demonstrates that all three constraints are satisfied simultaneously. The computation time of the pose graph optimization with three vertices and three edges is negligible, less than 0.2 ms on an Intel Core i7 2.67 GHz without parallelization.

Figure 7 shows a full map constructed from the complete data set, consisting of more than 2000 images overall. Unfortunately, the ROV did not loop back to the beginning of the path, and no larger loops could be closed.

B. Rocky Testing Pool

The following data has been recorded in a small pool used for testing landers and crawlers at Jacobs University Bremen's Ocean Lab. The bottom of the pool was covered with sand and several differently sized rocks. Conditions in

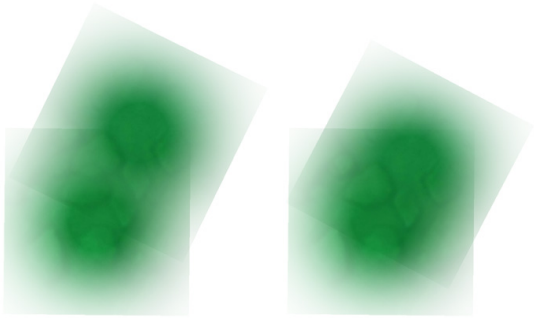

Fig. 5. Detail of the first and last frame of the testing pool image map, before and after pose graph optimization.

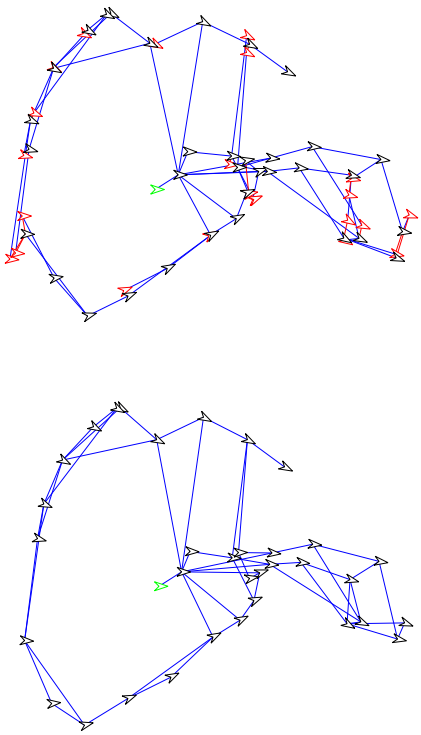

Fig. 6. Pose graph that represents the map shown in Figure 3, before (top) and after (bottom) optimization. It consists of 35 vertices and 57 edges. The starting pose is green, constraint edges from registration are blue, vertices that are projected to multiple poses because of accumulated errors in a loop are shown in red. Note that these conflicts are resolved after optimization.

the pool were not favorable as it was mostly taken over by algae and the ground was barely visible through the green tinted water. However, since it was a very controlled environment, we were able to collect data that contains many loops and is thus very suitable for mapping.

Figure 3 shows a map of the pool ground and rocks before and after optimization. Before optimization, as shown on the left of the figure, the map uses only the cumulative transformations calculated by sequential image registration. It is quite obvious that the lower part of the map is blurred beyond recognition. Two rocks, which seem to be properly localized in the upper portion of the map are actually visible twice, as pointed out by the arrows. Due to translational errors in the right part of the map, two rocks (the triangular one and its left neighbor) seem properly sized, but are localized incorrectly relative to each other.

After optimization, the map shows much crisper and more visible features and is shown on the right of figure 3. The bottom left of the map is clearer. The two rocks indicated by the arrows are localized properly.

The optimization of the pose graph was finished after 5.25 ms on an Intel Core i7 2.67 GHz. No parallelization of the optimization algorithm was performed. The small computation time is due to the relatively small size of the graph. This map contains only 35 vertices and 57 edges, as shown in figure 6. Please refer to [1], [2] for a detailed investigation and performance analysis of the optimization algorithm.

An interesting detail is revealed by looking only at the first and last picture of the sequence. Figure 5 shows how far apart the two pictures are localized before optimization. This specific artifact is the same as indicated by arrows in the complete map (figure 3). After optimization, the two images overlap properly. This is especially visible when looking at the diamond shaped rock. In the picture before optimization, it is present twice, afterwards, both images show the rock at the same location.

Figure 6 shows the underlying pose graph data structure. The arrow heads represent the potential global poses of the vertices. Blue lines show edges between vertices. Conflicts in the pose graph where the same vertex is projected to two or more different poses due to loops are highlighted in red. A similar distribution of errors is visible in this visualization. Vertices on the lower left loop of the graph are not very well localized before optimization. One of the vertices projects to significantly different poses, shown here in red with lines connecting them to the most likely pose drawn in black. After optimization, all constraints were satisfied and no conflicts remain.

This fact is also demonstrated by the significant decrease in the Mahalanobis distance. Before optimization, the value is 367.478. Afterwards the value drops to $9.527 \cdot 10^{-4}$. These values quantify the error in the map described above and show that the optimization algorithm was able to use the generated uncertainty information in a meaningful way.

C. Performance

The performance of the registration is dominated by the underlying FFT implementation. In the following experiments, the FFT implementation of the Gnu Scientific Library is used. Preliminary tests have shown that more optimized FFT implementations may increase performance by 30%.

On an Intel Core i7 2.67 GHz with 6GB RAM, registration of two images takes on average 0.055 seconds ($\sigma = 0.0071$). Please note that the program was not multi-threaded and thus did not use the processor to the fullest extend. Performance can be improved if the images are sequential, which means that some of the previous FFT computations can be reused. The registration then only takes 0.028 seconds on average

Fig. 7. Complete map generated from the Cold Corals data set, contains more than 2000 images.

(σ = 0.0058) for each sequential registration after the first one. These numbers include the time needed to extract the described uncertainty information.

Generating the map shown in figure 3 took 1.9 seconds, figure 7 took 133.7 seconds, both excluding file IO to read the images from the disk.

V. CONCLUSION

The improved Fourier Mellin Invariant (iFMI) method is a spectral approach to image registration, which is suited to generate robot maps by estimating the transformations between consecutively acquired image frames. For proper Simultaneous Localization and Mapping (SLAM) uncertainty measures are required, which have not received any attention in the context of spectral registration approaches. Here, iFMI registration is augmented by an uncertainty analysis. Concretely, it was shown how the dirac domain of the spectral image registration algorithm can be analyzed to generate uncertainty information about the registration result. This uncertainty is then used in a maximum likelihood mapping algorithm. The approach is tested with data from the underwater domain where qualitative as well as quantitative improvements can be observed by using SLAM with the uncertainty measure.

Acknowledgments

The research leading to the results presented here has received funding from the European Community's Seventh Framework Programme (EU FP7) under grant agreement n. 231378 "Cooperative Cognitive Control for Autonomous Underwater Vehicles (Co3-AUVs)", http://www.Co3-AUVs.eu.

References

[1] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, J. Leonard, Ed., 2006, pp. 2262–2269.

[2] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent," in *Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

[3] E. Takeuchi and T. Tsubouchi, "Multi sensor map building based on sparse linear equations solver," *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 2511–2518, Sept. 2008.

[4] M. Golfarelli, D. Maio, and S. Rizzi, "Correction of dead-reckoning errors in map building for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 1, pp. 37–47, Feb 2001.

[5] U. Frese, "Treemap: An o(log n) algorithm for simultaneous localization and mapping," in *Spatial Cognition IV. Reasoning, Action, and Interaction*, ser. Lecture Notes in Computer Science, J. G. Carbonell and J. Siekmann, Eds. Springer Berlin / Heidelberg, 2005, vol. 3343/2005.

[6] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 196–207, April 2005.

[7] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, 2000, vol. 1883, pp. 153–177.

[8] O. Bengtsson and A.-J. Baerveldt, "Location in changing environments – estimation of a covariance matrix for the idc algorithm," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct. 2001, pp. 1931–1937.

[9] S. T. Pfister, K. L. Kriechbaum, S. I. Roumeliotis, and J. W. Burdick, "Weighted range sensor matching algorithms for mobile robot displacement estimation," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 2, 2002, pp. 1667–1674.

[10] A. Censi, "An accurate closed-form estimate of icp's covariance," in *IEEE Int. Conf. on Robotics and Automation*, April 2007, pp. 3167–3172.

[11] H. Buelow, A. Birk, and V. Unnithan, "Online Generation of an Underwater Photo Map with Improved Fourier Mellin based Registration," in *International OCEANS Conference*. IEEE Press, 2009.

[12] H. Buelow and A. Birk, "Fast and Robust Photomapping with an Unmanned Aerial Vehicle (UAV)," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 2009.

[13] M. Pfingsthorn and A. Birk, "Efficiently Communicating Map Updates with the Pose Graph," in *International Conference on Intelligent Robots and Systems (IROS)*. Nice, France: IEEE Press, 2008.

[14] M. Pfingsthorn, Y. Nevatia, T. Stoyanov, R. Rathnam, S. Markov, and A. Birk, "Towards Collaborative and Decentralized Mapping in the Jacobs Virtual Rescue Team," in *RoboCup 2008: Robot WorldCup XII, Lecture Notes in Artificial Intelligence (LNAI)*, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Springer, 2009.

[15] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3d mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130–142, 2008.

[16] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[17] K. Pathak, N. Vaskevicius, J. Poppinga, M. Pfingsthorn, S. Schwertfeger, and A. Birk, "Fast 3D Mapping by Matching Planes Extracted from Range Sensor Point-Clouds," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 2009.

[18] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 2, Sept. 2003, pp. 1899–1906 vol.2.