The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

An Efficient Strategy for Data Exchange in Multi-Robot Mapping under Underwater Communication Constraints

Max Pfingsthorn Andreas Birk Heiko Bülow

Abstract— **The online generation of underwater image maps or mosaicking is of high interest for underwater robots, e.g., for autonomous navigation, exploration, or object detection. Here, a cooperative approach is presented that addresses the particular challenges of the severe constraints on communication bandwidth in the underwater domain. Concretely, a special update strategy for a cooperatively maintained pose graph as basis for Simultaneous Localization and Mapping (SLAM) is introduced. The strategy tries to transmit the most relevant information within the limits of the communication bandwidth to maximize the quality of the cooperative map. It is shown in experiments with simulations based on real world data that the strategy leads to near optimal results while obeying the severe bandwidth constraints of realistic underwater communication.**

I. INTRODUCTION

Underwater image mapping or mosaicking has a long history, starting in the late 1960s [1], [2], [3], [4], [5], [6]. It has been recognized as an important means to generate detailed underwater maps [7], [5] for example for surveying, i.e. as a mission deliverable, as well as for the robot control itself, e.g., for autonomous navigation, exploration, or object detection.

Different methods to combine image sequences into a mosaic have been proposed. In many cases, a feature-based approach is used as it allows registration of full 6-DoF pose changes independent of camera distortions [8], [6], [9], [7]. Often, well known image feature descriptors, such as SIFT [10] and SURF [11] are used, respectively variants of the Lucas-Kanade point tracker [7]. These methods are best suited for feature-rich sea floors (e.g. sponge beds, corals, archaeological sites), but do not work well with mostly homogeneous (e.g. sandy) ground without many salient features.

Other approaches for image registration that were successfully used in underwater settings are based on image correlation [5], [12], [3]. This kind of registration approaches is interesting as it does not require the computation of image features. Therefore it needs less parameters and is often faster and more robust than feature-based methods. It can successfully register image content that does not contain many discernible features as slight gradients are often sufficient. However, it is no longer possible to compute as many registration parameters. Usually, these methods are limited to two, respectively four motion parameters, namely 2D translation [5]), respectively 2D translation, rotation, and scale [3]). Please note that the latter allows ”typical” 3D-motions of Autonomous Underwater Vehicles (AUV), namely horizontal translations and rotations in the plane and the change of the depth where the AUV is operating, which is reflected in the scale parameter.

The execution of missions can in general profit a lot from multi-robot systems. And as unmanned underwater vehicles (UUV) in general and AUV in particular get more and more mature, there is an increasing interest to investigate cooperative systems. But previous research on underwater image mapping with multiple vehicles has mainly focused on the theoretical aspects [13], [14], [15] of applying Kalman and information filters in a multi-robot setting. These methods use landmark-based maps and do not usually construct an information-rich photo-mosaic map. Constrained underwater communication has not been investigated thoroughly, though first steps were taken in [16], [17], where the emphasis of the work is on encoding the data in small communication packets.

Naturally, depending on the specific landmark or feature type used, transmitting a landmark map can be significantly cheaper than transmitting a photomosaic map. Landmarks can describe anything, ranging from very high-level and mission-specific places such as bubble plumes [18] and mine locations [16], to terrain ridges [19], or local image features [8], [6], [7], [9], [4]. This is the main reason such very-low bandwidth and very mission-specific map transmission and update strategies as discussed in [16] work, even though they only transmit a very small number of bits. However, landmark-based maps do not retain the high level of detail found in photomosaics and can only be used for localization. The corresponding photomosaic maps produced in some studies are much more meaningful for a human.

To overcome the inherent bandwidth challenges in multi-robot underwater image mapping and combine it with the superior localization performance and small communication footprint of landmark-based algorithms, this paper presents a multi-robot map update strategy for pose graph SLAM.

Pose graph SLAM maintains a graph structure in which vertices denote poses where sensor data was recorded and added to the map, and edges denote pose differences generated by odometry, successful sensor data registration, or other sensors.

Specifically, the iFMI spectral image registration method developed at Jacobs University [20] is used in this context. It was recently extended to provide uncertainty estimates for the registration parameters [21]. Additionally, the pose

Robotics Group, School of Engineering and Science, Jacobs University Bremen, D-28725 Bremen, Germany. http://robotics.jacobs-university.de, a.birk@jacobs-university.de

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

graph map representation has previously been shown to require significantly less communication bandwidth than the occupancy grid map representation [22] for incremental updates in 2D ground robot mapping. This result is extended to photomosaic maps with 3D trajectories, combined with the multi-robot update strategy.

The rest of this paper is structured as follows. First, different off-the-shelf solutions for underwater modems are discussed in section II. An overview of pose graph SLAM and the proposed update strategy is given in section III. Experiments and results are discussed in section IV. The final section V describes the conclusions reached in this paper.

II. REALISTIC DATA RATES OF UNDERWATER MODEMS

Underwater communication is a very challenging topic for many reasons. It is a relatively new field, at least when compared to research on communication technologies in standard applications scenarios on the earth's surface and air. One way to address the problem is to use tethers even for AUV [23], but this has its obvious limitations and it contradicts the main reasons why AUV are used. So, wireless communication solutions are by far of most interest.

The predominant approach used in standard wireless technologies, namely to rely on radio frequency (RF), is impossible or at least tremendously difficult in water. There is some work on using optical means for wireless underwater communication [24], [25]. But this is still in a relatively early, experimental state and it remains unclear how feasible according approaches are in real world application scenarios.

The predominant carrier for underwater communication is hence sound, which propagates well in water. But it has - among many other challenges - the disadvantage that sound is relatively slow. The communication rates that can hence be achieved are very limited. There are also quite many other aspects that limit acoustic underwater communication like multi-path-effects, presence of noise in many application scenarios, and so on. For some general discussions of the challenges and approaches for acoustic underwater modems see for example [26], [27], [28].

A rough, non-exhaustive overview of commercial-off-the-shelf (COTS) solutions is given in this section, which is used to motivate reasonable estimates of communication bandwidths for multi-robot operations in the underwater domain. Please note that latency does not affect the work presented here, it is hence omitted in the following discussion. But latency can of course be relevant for other aspects of robot cooperation.

The Woods Hole Oceanographic Institution (WHOI) has a long tradition in underwater communication research. They offer their WHOI Micro-Modem [29] as a customizable kit to interested parties. The WHOI Micro-Modem provides up to 5400 bits/sec and up to 4 channels at different frequencies. The company Evologics [30] provides several high speed modems. The S2C R 48/78 acoustic modem has a bandwidth of up to 28,000 bit/sec. Its range is 1000 meters with a horizontal, omnidirectional spreading, therefore it is best suited for shallow water scenarios. The Evologics S2C R 40/80 has even a faster data rate of up to 33,000 bit/sec over a range of 2000 meters. This very high speed is bought at the cost of a directional beam with 70 degree opening angle. It hence would require cooperative robots to move in special patterns or to use actuators to align the transducers to keep a steady link at this very high rate. The company Linkquest [31] also offers several underwater modems with quite high speeds. The UWM1000 and the UWM2000 provide both data rates up to 19,200 bit/sec. They differ in their power consumption and range, namely up to 350 meters for the UWM1000 and up to 1500 meters for the UWM2000. Both can be used with directional or omni-directional transducers. The Linkquest UWM2200 offers even higher rates with up to 38,400 bit/sec but again at the cost of requiring a directional beam.

The amount of actual payload data depends of course on several additional factors. First of all, the maximum data rates are only achieved in quite favorable conditions. But as cooperative robots tend to operate relatively closely to each other, namely in the range of a few meters to a few dozen or at most hundreds meters distance, it is likely to have good conditions. Second, the data rates are so to say raw values that do not take the overhead for network management into account. But the network management can be made very efficiently, especially once the links are established and the devices are in burst modes. Last but not least, sound propagation in water is a shared medium. Directional devices are not obvious to use for moving multi-robot teams. Hence, either multiple channels, i.e. non-overlapping frequency bands, must be available or time division multiplexing must be used.

We assume here small robot teams with up to four robots. It is quite simple - at least from the perspective of the architecture and the necessary network protocol overhead - to establish a fully connected network via three links and one robot acting as hub in the case of four robots. As discussed above, the WHOI Micro-Modem provides 5,400 bits/sec and up to 4 channels. There are several solutions with 19,200 to 28,000 bits/sec and omnidirectional spreading, thus three time-multiplexed channels may achieve between 6,400 to 9,333 bit/sec. We hence chose 4,800 bit/sec as a realistic estimate for the *upper limit* in the experiments presented later on in section IV.

III. MULTI-ROBOT POSE GRAPH SLAM WITH IMAGES

A. Pose graph SLAM Overview

In this section, the basis of the Simultaneous Localization and Mapping (SLAM) used here is introduced in a formal way. It is based on a pose graph, which is a very popular map representation in maximum likelihood mapping. In general, graph based map as have been popular in the recent literature [22], [32], [33], [34], [35], [36], [37], [38], [39], [40].

A pose graph is an undirected graph $G = (V,E)$. All $v \in V$ are poses ($pose(v)$) where sensor observations ($sensorob(v)$) were made and inserted into the map. The edges $e \in E$ ($e = (v_{start}, v_{end})$) are constraints on the poses of the vertices

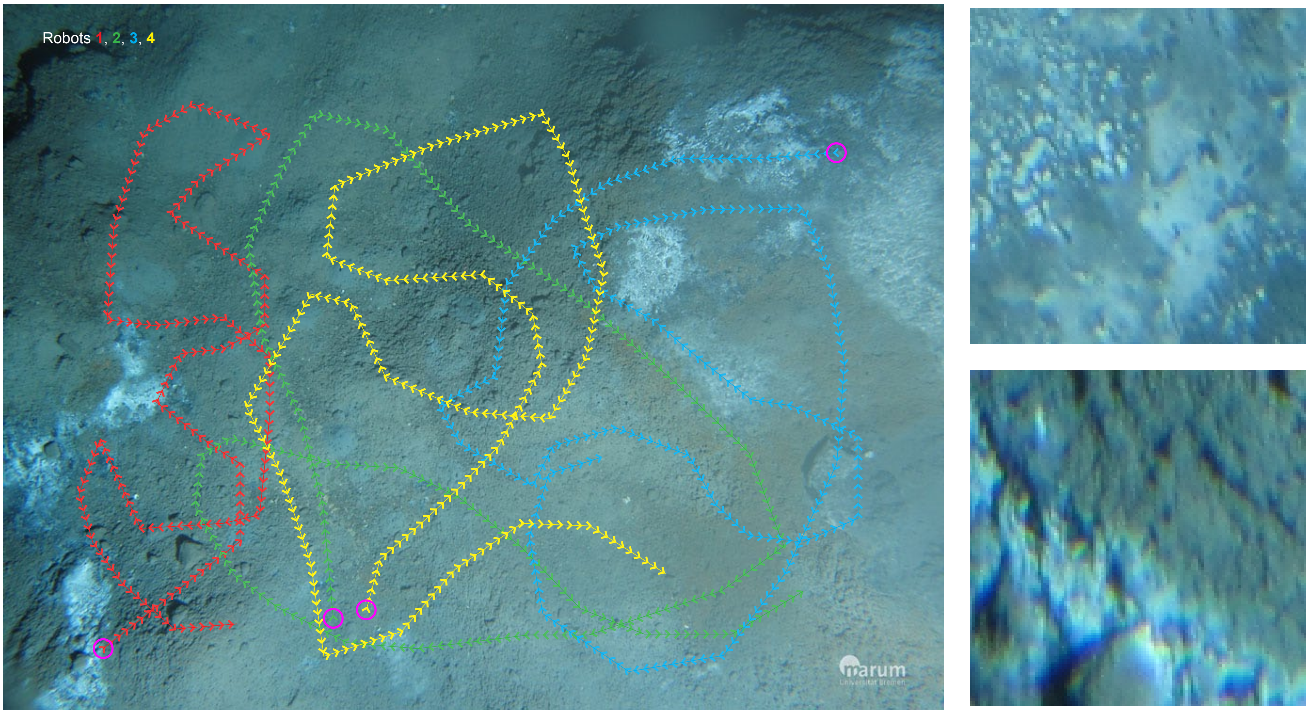

Fig. 1. Left: Sample paths on a high resolution image for a team of four robots. Start locations are circled. Right: Example images extracted for the start locations of robots 1 (bottom) and 3 (top).

they connect ($start(e) = v_{start}, end(e) = v_{end}$), usually in the form of a pose difference ($posediff(e)$) accompanied by a covariance matrix ($cov(e)$) to describe the uncertainty of the constraint (its elasticity). Vertices are marked as *start* or *end* vertices of the undirected edge because the pose difference depends on the traversal direction. The edges can be traversed in both directions.

Constraint types other than full pose differences, such as position-only (for landmarks), range-only, or bearing-only (for beacons) are also conceivable, but have not found much use in the literature.

A popular cost function is the sum of squared Mahalanobis distances

$$cost(G) = \sum_i (p^e_{e_i} \ominus p^s_{e_i} - p_{e_i})^T C^{-1}_{e_i} (p^e_{e_i} \ominus p^s_{e_i} - p_{e_i}) \qquad (1)$$

where $p^s_{e_i} = pose(start(e_i))$, $p^e_{e_i} = pose(end(e_i))$, $p_{e_i} = posediff(e_i)$, $C_{e_i} = cov(e_i)$, and $\ominus$ denotes the pose difference operator [41]. Various methods exist to optimize a pose graph map containing full pose difference constraints (3-DoF or 6-DoF) with covariance matrices [35], [34], [38], [37]. Most of these methods use the above mentioned cost function.

A multi-robot pose graph consists of one or more disconnected components, with vertices and edges supplied by each member of the team. Components can become connected by successfully registering sensor data collected by two different robots. Formally, the multi-robot pose graph

$$G^+ = (V^+, E^+)$$

consists of multiple disjoint sets of vertices

$$V^+ = \bigcup_{i=1}^{N} V_i$$

where N is the number of robots in the team. This set of edges E^+ can be expressed similarly as

$$E^+ = \left(\bigcup_{i=1}^{N} E_i \right) \cup \left(\bigcup_{i=1}^{N-1} \bigcup_{j=i+1}^{N} E_{ij} \right)$$

Here E_i is the set of edges just involving vertices of robot i (that is all incremental and loop closing edges of the single robot map of robot i), and E_{ij} is the set of edges connecting vertices of robot i with vertices of robot j.

The edges in $E_{ij} \cup E_j$ are significant as they allow robot i to close additional loops, for example when two robots start at one location and meet at another after having taken different routes. Robots i and j do not have to revisit places in their own trajectory, but can reuse their teammates' trajectories for the same purpose.

It is also important to note that the above mentioned optimization methods can be run anytime, which means that they will find the best configuration of the pose graph given the currently available information. This means that additional edges or sensor observations from other robots are not necessary for the optimization to succeed. Therefore, any latency or delay caused by exhaustion of the bandwidth in transmission of pose graph vertices, edges, and sensor observations from other robots is irrelevant. New information coming from other robots can easily be integrated as it is received, even if the mission is already over.

While the edges in E_i are readily available after local loop closing, the efficient retrieval of edges in $E_{ij}, j = 1 \dots N, j \neq i$ is the main objective of this paper.

B. Multi-Robot Map Update Strategy

The main idea behind the strategy proposed here is that only selected data needed to form the edges in E_{ij} must be transmitted between robots i and j. Other robots do not need to know specific sensor observations used to generate these inter-robot edges as their vertices are not involved in these edges. Usually, these edges occur in overlapping regions of the two robots' trajectories. As long as all robots know the current estimates of their teammates trajectories, which are explicitly contained in the pose graph structure build so far, robots can request specific images from locations they believe are most likely to coincide with their own path and hence well suited to improve the overall joined map. Additionally, when remaining bandwidth allows, likely candidate images can be pro-actively sent by robots to their peers to maximize the exploitation of the available bandwidth.

Algorithm 1: Pseudo code for determining the image to send from robot i to robot j

```
Data: pose graph $G^+$
Data: The set of sent data to robot $j$ $S_j$ (initially empty)
Data: distance threshold $d_{max}$
Output: Sensor observation to send to robot $j$
$v^* = \text{NIL}$;
$d^* = \infty$;
for $v_i \in V_i \setminus S_j$ do
    $d = \underset{v_j \in V_j}{\text{argmin}}\ dist(pose(v_i), pose(v_j))$;
    if $d < d^*$ then
        $d^* = d$;
        $v^* = v_i$;
    end
end
if $v^* \neq \text{NIL} \wedge d^* < d_{max}$ then
    $S_j = S_j \cup \{v^*\}$;
    return $sensorob(v^*)$;
end
return NIL;
```

Algorithm 1 sketches the strategy to determine the image to be sent from robot i to robot j at a given time step. The Euclidean distance between the estimate vertex positions is used to determine a good sensor observation (here a camera image) to share between the robots. Once robot j has the image, it will try to register it with any nearby images. All generated edges are then broadcast to all robots.

Using this strategy, a team of robots can fully localize each other, and generate a joint photo-mosaic map. However, each robot mainly stores specific images for the area it covered itself. It is important to note that the pose graph constructed by this method contains full uncertainty information. Global optimization algorithms, as mentioned in section III-A, can be applied to this logical graph structure without requiring sensor observations (e.g. camera images, etc) to be transmitted between the peers. This allows to make a trade off between computational cost to re-optimize the graph on each robot and the bandwidth cost to transmit the optimized graph. Since computational resources are significantly cheaper in an underwater setting than bandwidth resources, the update strategy limits the amount of data sent to just topological updates. Metric updates, such as specific pose estimates of vertices, are never transmitted as they can be recomputed by each robot with the available information.

C. Data Transmission

In the experiments below, all data is transmitted in binary form. Data-specific knowledge is used to choose a format that requires the least amount of bytes to transmit the message. Compression is only used for the images transmitted between the robot peers.

In total, three distinct messages are used to communicate all mapping related information.

1) *Start*: A message containing the ID of the sending robot and its starting pose (3D vector and a quaternion), if available.
2) *Update*: A message containing the complete edge information, which consists of IDs of the two vertices linked by this edge, and a pose difference (3D vector and a quaternion) with an accompanying covariance matrix. This matrix is either stored as a 7x7 float array or a list of matrix indices and float values (with the indices being two four bit integers), whichever is smaller.
3) *Image*: This message contains the ID of the vertex this image belongs to and the compressed JPEG image itself. The images were compressed with 30% quality using `ImageMagick`.

Thus, a *Start* message is $2 + 7 \cdot 4 = 30$ bytes long. It contains a short integer as the robot ID and a pose, consisting of a 3D float vector and a quaternion, also using floats.

An *Image* message contains a two byte and a four byte integer as the vertex reference (robot ID and vertex ID), a four byte integer to store the size of the binary data, and the binary compressed image data itself. In the data set used for the experiment, this message is on average 1212.87 bytes long ($\sigma = 347.0156$).

An *Update* message has a length of $2 \cdot (2 + 4) + 7 \cdot 4 + 1 + sizeof(cov)$ bytes, which is at least 41 bytes with a zero covariance matrix, and at most 237 bytes with a dense covariance matrix. Matrices are either stored as a list of indices and values, with two four bit integers as indices and a four byte float value, or a dense list of 49 four byte float values. In practice, given the amount of uncertainty information extracted from the image registration method, the message is always 92 bytes long, which includes 10 non-zero matrix values. Since the employed registration method decouples translation and rotation, the matrices are mostly sparse.

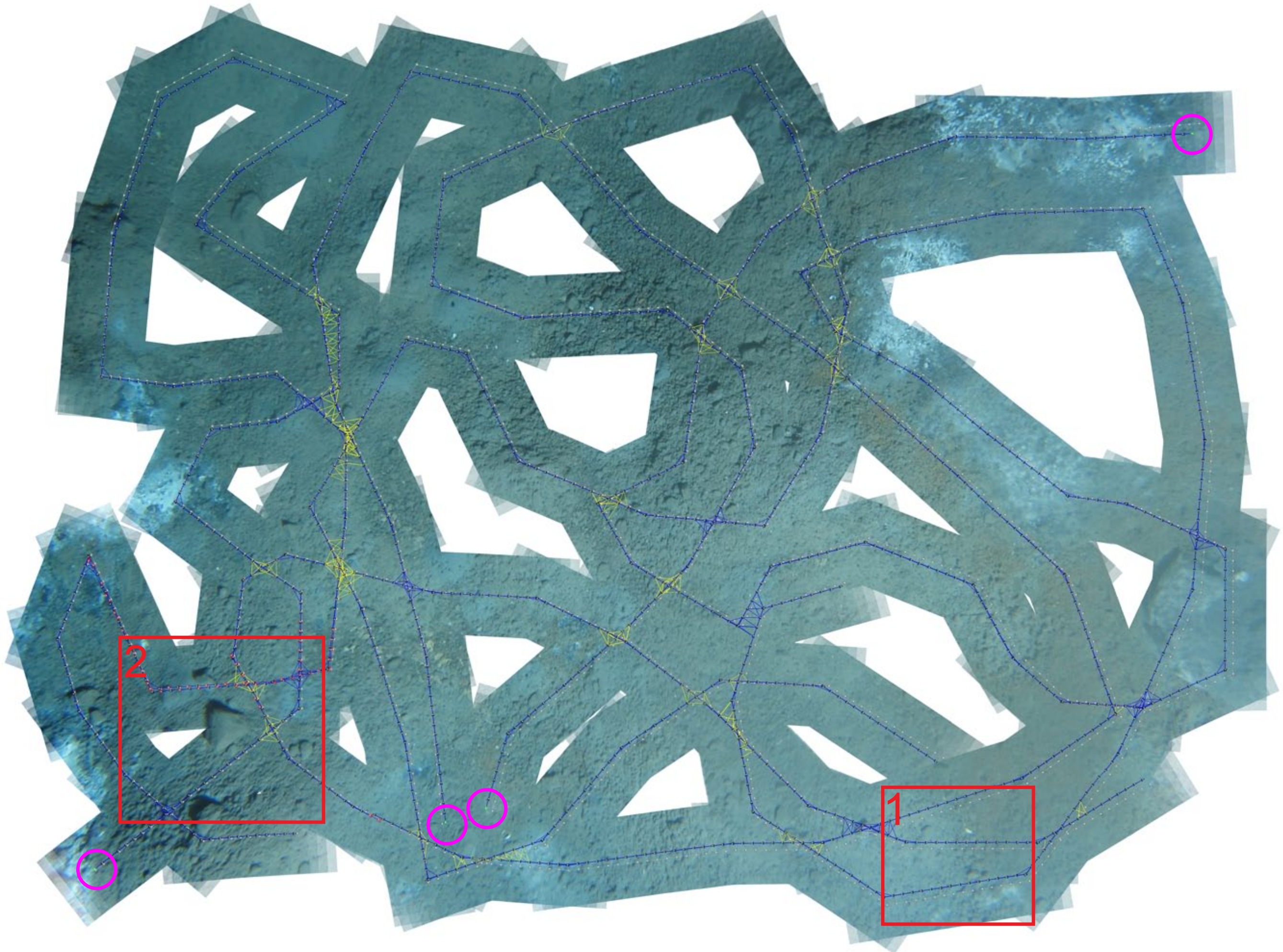

Fig. 2. An example result of the cooperative mapping strategy. The pose graph structure is superimposed (poses are black arrow heads, edges are blue, inter-robot edges are yellow, residual errors are red, startposes are marked with circles, ground truth is shown in gray). Details of the areas indicated by red squares are shown in figure 3

Since only covariances for 2D translation and a rotational variance are extracted, the message could theoretically be as small as 44 $(2 \cdot (2+4) + 3 \cdot 4 + 5 \cdot 4)$ bytes if the pose was 2D and the translation and rotation were independent. However, the cooperative pose graph update strategy itself allows for other - possibly full 6 DoF - motion estimation methods to be used. Using the presented multi-robot map update strategy, it is thus possible to build full 3D maps within severe communication constraints.

D. Minimum required bandwidth

At a minimum, all robots shall receive pose graph updates from all other robots. This is conceptually similar to just broadcasting the current robot poses, however it allows for later optimization of the trajectories by the receiving robot. Such extra information is especially useful for autonomous behaviors during the mission.

A *Start* message has to be sent at the beginning if the robot starting pose is known. Every time step, an *Update* message has to be sent to allow other robots to reconstruct the trajectory. Thus, at a minimum, 41 bytes per time step per robot has to be available. Using the registration method and the uncertainty estimation described in [21], at least 92 bytes per time step have to be transferred. More detailed uncertainty information, as might be computed from other registration methods, usually results in larger message sizes.

However, this only allows the transmission of incremental edges, not the transmission of loop closures within a robot's trajectory, i.e. registrations of sensor data with data stored in previously visited nodes. This information is needed by other robots to be actually able to optimize the trajectory. A single trajectory without loops is informative already, but loops allow the definition of a meaningful cost function of vertex positions that can be optimized. With a more accurate trajectory of its peers after optimization, a robot can also more accurately estimate the potential usefulness of its images to other robots. Therefore, all possible edges to previous already known vertices need to be transmitted as well. This is of course dependent on the specific robot trajectory and how many times it crosses its own path.

The average number of back edges in the example trajectories shown in figure 1 is 2.314 ($\sigma = 0.725$). Thus, on average, every vertex is involved in 1.314 loop closing edges to previous vertices, not counting incremental registration edges. Finally, on average for the trajectories in figure 1, 212.92 bytes per time step need to be transmitted ($\sigma = 66.71$) in order to communicate all incremental and loop closing edges.

IV. EXPERIMENTS

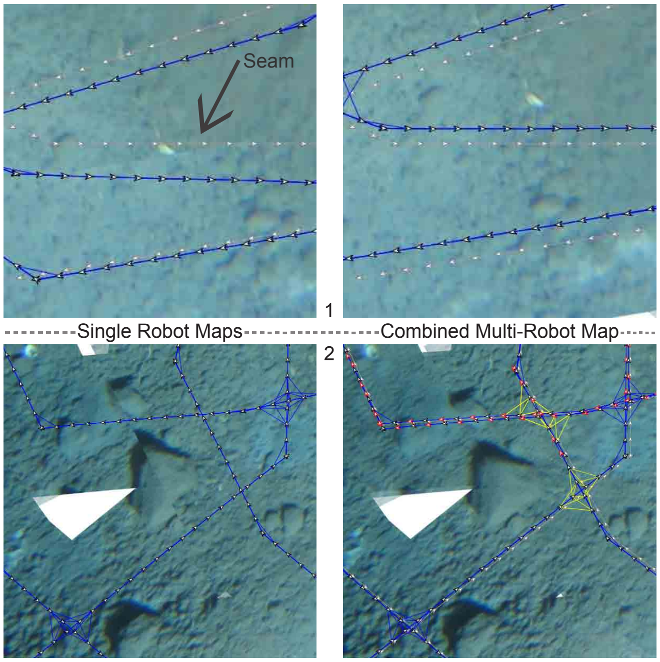

Fig. 3. Map details from figure 2. *No Comms* case (left) is compared with the map built with the *Multi-Robot* strategy (right). Numbers correspond to indicated areas in figure 2.

The presented multi-robot map update strategy is tested in simulation using real world data. Image sequences were extracted from high resolution underwater image data. One example of the underlying data for generating the video streams is shown in figure 1. The stream of images generated along the paths drawn on the high resolution source image was used as camera data for image mapping. No odometry or similar motion estimates are used in the experiments.

A team of four robots is simulated, including their communication. Following assumptions are made to keep the simulation realistic. The first assumption is that the underwater communication system is limited to a data rate of 4,800 bit/sec per channel. Each channel is assumed to allow two-way communication, with one robot sending data and one robot receiving at a time. Full duplex communication, i.e. where both robots send and receive at the same time, is explicitly assumed to be not possible. Secondly, the communication system is assumed to have three channels to allow a full network. Finally, the data rate of the channels is assumed to be constant given the distance of the two communicating robots. Usually, the last assumption should be fulfilled if the robots stay within a local area, given the rather large range of underwater acoustic modems of several hundred meters to kilometers.

Errors:	Positions (pixel)	Rotations (rad)
Full Comms	15.9090 ($\sigma = 8.85$)	0.5652
Multi-Robot	13.5177 ($\sigma = 7.7041$)	0.5086
No Comms	28.9373 ($\sigma = 20.99$)	1.3217

TABLE I
COMPARISONS OF MAP QUALITY WITH RESPECT TO GROUND TRUTH.

Figure 1 shows the set of pre-programmed paths of the first experiment. The source image contains 4416x3312 pixels, the extracted video frames are all 128x128 pixels large. A total of 1581 video frames were extracted. The longest sequence was 438 images long.

To assign some real world values to the simulation world, it is assumed that the image covers an area of 50 by 66 m ($3300m^2$). One pixel then corresponds to an area of approximately 15 by 15 cm. One time step is one second, the following bandwidth capacities follow accordingly. On average, each robot traveled 30.68 pixels per time step, which translates to an average speed of $0.463m/sec$ or 0.9 knots. The robots move at a constant height of $3.35m$ above the sea-floor, and the opening angle of the camera is assumed to be $60°$. This results in an area of $14.94m^2$ covered by the camera image.

Please note that the above speed estimate of the simulated vehicles is quite high for underwater robots and that the distance to the floor is quite small. Thus, the following experiments and the preceding assumptions actually represent a conservative estimate of the potential performance. If the available bandwidth is smaller, or less channels are available, the robots can decrease their speed to reduce bandwidth requirements and the following results remain valid. There is also the option to increase the distance to the seabed to cover a larger area of the sea floor with one camera image to accommodate lower communication rates.

Figure 4 shows a plot of the average amount of transferred bytes by each robot pair (i.e. per communication channel) in the team per time step. Faster map updates would have resulted in a congestion of the communication channels. Just transferring the pose graph updates at a faster rate would have exceeded the available bandwidth.

The plot compares required data rates for an update strategy that does not take the available bandwidth into account with the proposed multi-robot update strategy. Concretely, the *Full Comms* strategy transmits all acquired images at every time step and thus exceeds the available limited bandwidth by a factor of two.

Naturally, it has to be expected that transmitting less information in map updates results in deteriorated maps. However, as large portions of the global map are of no consequence to individual robots, especially regions that have few or no overlap, parts of this information are likely to be safely omitted. Table I shows a comparison between three possible update strategies: Full communication (i.e. transmitting all

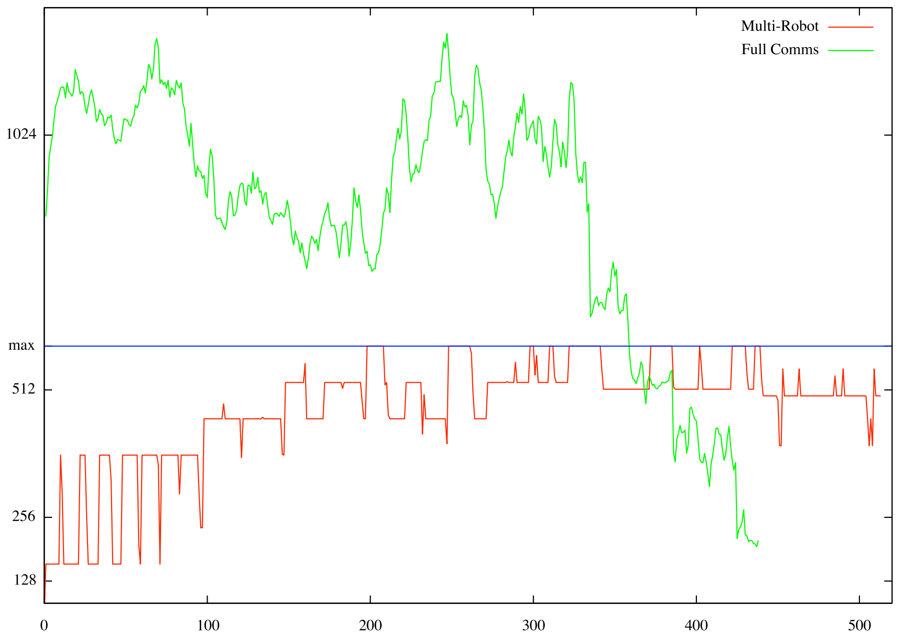

Fig. 4. Bytes transferred per time step, first experiment. The multi-robot update strategy uses the available bandwidth almost optimally.

video streams to all robots), no communication at all (i.e. only single robot maps overlaid by known starting poses), and the proposed multi-robot update strategy. The mean euclidean distance, the corresponding standard deviation and the mean angle difference between the estimated and ground truth poses of each vertex in the pose graph map is shown. The proposed multi-robot update strategy performs very similar to the *Full Comms*, and theoretically best, strategy. Concretely, it even produces slightly less errors than the optimal strategy, which can be explained by the stochastic nature of the optimization processes in the pose graph optimization. Both strategies produce significantly more accurate maps than non-cooperative, communication-free mapping.

Generating the map shown in figure 2 took a total of 1752.23 seconds on a Core 2 Duo 2.8GHz with 4GB of RAM. This simulation was done for four robots in a single thread, which results in an average simulation time of 438.06 seconds per robot, about 7.5 minutes where each robot travels about 225 m. The team was active for a total of 513 time steps, or seconds given the simulation rate of 1 Hz, so the process is real-time capable.

The image registration algorithm runs at a rate of 10Hz in Matlab, a C++ prototype implementation exists that can achieve 30Hz. The main bottleneck is actually the loop search, which is accelerated by indexing the vertex positions in an octtree and takes $O(NM\log(M))$, where N is the number of robots and $M = \max(|V_i|)$. For all other $j = 1 \dots N$ robots, a nearest neighbor v_j has to be found for every vertex v_i. Each nearest neighbor query in an octtree takes $O(\log(M))$ time. Note that the case $i = j$ is explicitly included, as that corresponds to the loop search within the robot's own map.

As mentioned before, the robots travel quite fast with 0.9 knots and they are close to the ocean floor, thus there is a lot of change and little overlap between consecutive images in the video streams. Nevertheless, the approach is already real-time capable with significant parts implemented in MATLAB.

V. CONCLUSION

A cooperative approach to generate underwater image maps was presented that takes the particular challenges of the severe constraints on communication bandwidth in the underwater domain into account. In doing so, a special update strategy was introduced to maintain a joined pose graph for Simultaneous Localization and Mapping (SLAM). The strategy tries to transmit the most relevant information within the limits of the communication bandwidth to maximize the quality of the cooperative map. It was shown in experiments with simulations based on real world data that the strategy leads to near optimal results while obeying the severe bandwidth constraints of realistic underwater communication.

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`. Furthermore we would like to thank the Marum - Center for Marine Environmental Sciences, University of Bremen for providing the underwater image data.

REFERENCES

[1] J. Pollio, “Stereo-photographic mapping from submersibles,” *SPIE*, vol. 12, pp. 67–71, 1968.

[2] R. Haywood, “Acquisition of a micro scale photographic survey using an autonomous submersible,” in *OCEANS 1984*, vol. 18, Sep 1986, pp. 1423 – 1426.

[3] R. Marks, S. Rock, and M. Lee, “Real-time video mosaicking of the ocean floor,” *Oceanic Engineering, IEEE Journal of*, vol. 20, no. 3, pp. 229 –241, Jul 1995.

[4] Y. Rzhanov, L. Linnett, and R. Forbes, “Underwater video mosaicking for seabed mapping,” in *ICIP, 2000.*, 2000, pp. 224–227.

[5] K. Richmond and S. Rock, “An operational real-time large-scale visual mosaicking and navigation system,” in *OCEANS 2006*, Sept. 2006, pp. 1 –6.

[6] M. Caccia, G. Bruzzone, F. Ferreira, and G. Veruggio, “Online video mosaicing through slam for rovs,” in *OCEANS 2009-EUROPE, 2009. OCEANS '09.*, May 2009, pp. 1 –6.

[7] 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.

[8] A. Kim and R. Eustice, “Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct. 2009, pp. 1559 –1565.

[9] O. Pizarro and H. Singh, “Toward large-area mosaicing for underwater scientific applications,” *Oceanic Engineering, IEEE Journal of*, vol. 28, no. 4, pp. 651 – 672, Oct. 2003.

[10] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.

[12] O. Pizarro, H. Singh, and S. Lerner, “Towards image-based characterization of acoustic navigation,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, 2001, pp. 1519–1524 vol.3.

[13] A. Bahr, J. J. Leonard, and M. F. Fallon, “Cooperative Localization for Autonomous Underwater Vehicles,” *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.

[14] J. Diosdado and I. Ruiz, “Decentralised simultaneous localisation and mapping for auvs,” in *OCEANS 2007 - Europe*, June 2007, pp. 1 –6.

[15] S. Botelho, R. Neves, and L. Taddei, “Localization of a fleet of auvs using visual maps,” in *Oceans 2005 - Europe*, vol. 2, June 2005, pp. 1320 – 1325 Vol. 2.

[16] B. Johnson, N. Hallin, H. Leidenfrost, M. O'Rourke, and D. Edwards, “Collaborative mapping with autonomous underwater vehicles in low-bandwidth conditions,” in *OCEANS 2009-EUROPE, 2009. OCEANS '09.*, May 2009, pp. 1 –7.

[17] A. Rajala and D. Edwards, “Allocating auvs for mine map development in mcm,” in *OCEANS 2006 - Asia Pacific*, May 2006, pp. 1 –8.

[18] T. Maki, H. Kondo, T. Ura, and T. Sakamaki, T. A4 Sakamaki, “Photo Mosaicing of Tagiri Shallow Vent Area by the AUV ”Tri-Dog 1” using a SLAM based Navigation Scheme,” in *OCEANS 2006 OCEANS 2006 VO -*, H. Kondo, Ed., 2006, pp. 1–6.

[19] I. Nygren and M. Jansson, “Terrain navigation for underwater vehicles using the correlator method,” *Oceanic Engineering, IEEE Journal of*, vol. 29, no. 3, pp. 906–915, 2004.

[20] H. Bulow, A. Birk, and V. Unnithan, “Online generation of an underwater photo map with improved fourier mellin based registration,” in *OCEANS 2009-EUROPE, 2009. OCEANS '09.*, May 2009, pp. 1–6.

[21] M. Pfingsthorn, A. Birk, S. Schwertfeger, H. Bülow, and K. Pathak, “Maximum likelihood mapping with spectral image registration,” in *Robotics and Automation, 2010. ICRA 2010. Proceedings of the 2010 IEEE International Conference on*, 2010.

[22] M. Pfingsthorn and A. Birk, “Efficiently communicating map updates with the pose graph,” in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[23] S.-C. Yu, T. Ura, and N. Yoshiaki, “Multi-AUV based cooperative observations,” in *Autonomous Underwater Vehicles, 2004 IEEE/OES*, T. Ura, Ed., 2004, pp. 7–13.

[24] S. Jaruwatanadilok, “Underwater Wireless Optical Communication Channel Modeling and Performance Evaluation using Vector Radiative Transfer Theory,” *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 9, pp. 1620–1627, 2008.

[25] F. Hanson and S. Radic, “High bandwidth underwater optical communication,” *Applied Optics*, vol. 47, no. 2, pp. 277–283, 2008.

[26] M. Stojanovic, “Recent advances in high-speed underwater acoustic communications,” *Oceanic Engineering, IEEE Journal of*, vol. 21, no. 2, pp. 125–136, 1996.

[27] E. Sozer, M. Stojanovic, and J. Proakis, “Underwater acoustic networks,” *Oceanic Engineering, IEEE Journal of*, vol. 25, no. 1, pp. 72–83, 2000.

[28] H. Riksfjord, O. Haug, and J. Hovem, “Underwater Acoustic Networks - Survey on Communication Challenges with Transmission Simulations,” in *Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on*, 2009, pp. 300–305.

[29] Woods Hole Oceanographic Institute, “Acoustic communications,” http://acomms.whoi.edu, 2006.

[30] Evologics GmbH, http://www.evologics.de, 2010.

[31] LinkQuest Inc., http://www.link-quest.com, 1999-2010.

[32] 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.

[33] 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.

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

[35] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, “A tree parameterization for efficiently computing maximum likelihood maps using gradient descent,” in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

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

[41] R. C. Smith and P. Cheeseman, “On the Representation and Estimation of Spatial Uncertainty,” *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.