Mini-SLAM: Minimalistic Visual SLAM in Large-Scale Environments Based on a New Interpretation of Image Similarity

Henrik Andreasson*, Tom Duckett† and Achim Lilienthal*

* Applied Autonomous Sensor Systems
Örebro University
SE-701 82 Örebro, Sweden
henrik.andreasson, achim@lilienthals.de

† Department of Computing and Informatics
University of Lincoln
Brayford Pool, Lincoln, LN6 7TS, UK
tduckett@lincoln.ac.uk

Abstract— **This paper presents a vision-based approach to SLAM in large-scale environments with minimal sensing and computational requirements. The approach is based on a graphical representation of robot poses and links between the poses. Links between the robot poses are established based on odomety and image similarity, then a relaxation algorithm is used to generate a globally consistent map. To estimate the covariance matrix for links obtained from the vision sensor, a novel method is introduced based on the relative similarity of neighbouring images, without requiring distances to image features or multiple view geometry. Indoor and outdoor experiments demonstrate that the approach scales well to large-scale environments, producing topologically correct and geometrically accurate maps at minimal computational cost. Mini-SLAM was found to produce consistent maps in an unstructured, large-scale environment (the total path length was 1.4 km) containing indoor and outdoor passages.**

I. Introduction

This paper presents a new approach to the problem of simultaneous localization and mapping (SLAM). The suggested method is called "Mini-SLAM" since it is minimalistic in several ways. On the hardware side, it solely relies on odometry and an omnidirectional camera. Using a camera as the external source of information for the SLAM algorithm provides a much cheaper solution compared to state-of-the-art 2D and 3D laser scanners with a typically even longer range. It is further known that vision can enable solutions in highly cluttered environments where laser range scanner based SLAM algorithms fail [1]. In this paper, the readings from a laser scanner are used to demonstrate the consistency of the created maps. Please note, however, that the laser scanner is only utilised for visualisation and is not used in the visual SLAM algorithm.

Apart from the frugal hardware requirements, the method is also minimalistic in its computational demands. Map estimation is performed online by a linear time SLAM algorithm that operates on an efficient graph representation. The main difference to other vision-based SLAM approaches is that there is no estimate of the positions of a set of landmarks involved, enabling the algorithm to scale up better with the size of the environment. Instead, a measure of image similarity is used to estimate the pose difference in between corresponding images and the uncertainty of this estimate. Given these "visual relations" and "odometry relations" between consecutive images, the Multilevel Relaxation algorithm [2] is then used to determine the maximum likelihood estimate of all image poses. The relations are expressed as an estimate of the relative pose and the corresponding covariance. A key point of the approach is that the estimate of the pose difference in the relations (particularly in the "visual relations") does not need to be extremely accurate as long as the corresponding covariance is modeled correctly. This is because the pose difference is only used as an initial estimate that the Multilevel Relaxation algorithm can adjust according to the covariance of the relation. Therefore, even with an initial estimate of the pose difference that is fairly imprecise, it is possible to build maps with improved geometric accuracy using the geometric information expressed by the covariance of the relative pose estimate. In this paper, the initial estimate of the relative pose is determined assuming that similar images are taken at the same place. Despite this simplistic assumption, Mini-SLAM was found to produce consistent maps in an unstructured, large-scale environment (the total path length was 1.4 km) containing indoor and outdoor passages.

A. Related Work

Using a camera as the external source of information in SLAM has received increasing attention during the past years. Many approaches extract landmarks using local features in the images and track the positions of these landmarks. As the feature descriptor, Lowe's scale invariant feature transform (SIFT) [3] has been used widely [4], [5]. An initial estimate of the relative pose change is often obtained from odometry [5], [6], [7], or where multiple cameras are available as in [8], [9], multiple view geometry can be applied to obtain depth estimates of the extracted features. To update and maintain visual landmarks, Extended Kalman Filters (EKF) [10], [6], Rao-Blackwellised Particle Filters (RBPF) [8] and FastSLAM [5] are among the most popular methods applied. The visual SLAM method in [10] uses a single camera. Particle filters were utilised to obtain the depth of landmarks, while the landmark positions were updated with an EKF. In order to obtain metrically correct estimates, initial landmark positions had to be provided by the user. A similar approach described in [7] also uses a single camera but applies a converse methodology. The landmark positions were estimated with a Kalman filter (KF) and a particle filter was used to estimate the path.

Since vision is particularly suited to solve the correspondence problem, vision-based systems have been applied as an addition to laser scanning based SLAM approaches for detecting loop closing. The principle has been applied to SLAM systems based on a 2D laser scanner [11] and a 3D laser scanner [12]. A totally different approach that combines vision and laser range scanning is described in [13]. Here, contours extracted from an aerial image were used to improve the consistency of a map, which was initially created using the laser scanner.

Other mapping approaches have combined omni-directional vision for place recognition with odometry for obtaining geometric information in a graph. For example, Ranganathan and Dellaert [14] use odometry information to evaluate the likelihood of topological map hypotheses in a MCMC framework. However, the emphasis of their work is on selecting the correct topology using very coarse visual features, and their approach is unlikely to scale to environments of the size presented here.

The rest of the paper is structured as follows. Section II describes the suggested SLAM approach. It includes a brief overview of the SLAM optimization technique (Section II-A) a description of the way in which relations are computed from odometry (Section II-B), and from visual similarity (Section II-C). Then the experimental set-up is detailed and the results are presented in Section III. The paper ends with conclusions and suggestions for future work (Section IV).

II. Mini-SLAM

Our approach is based on two principles. First, odometry is fairly precise if the distance traveled is short. Second, by using visual matching, correspondence between robot poses can be detected reliably even though the covariance of the current pose estimate, i.e. the search region, is large.

We therefore have two different types of relations. Relation based on odometry r_o and relation based on visual similarities r_v.

A. Multi-Level Relaxation

The SLAM problem is solved at the graph-level, where the Multilevel Relaxation (MLR) method of Frese and Duckett [2] is applied. In this method, a map is represented as a set of nodes connected in a graph. Each node or frame corresponds to the robot pose at a particular time (in our case when an omni-image was taken), and each link corresponds to a relative measurement of the spatial relation between the two nodes it connects, see Fig. 1.

The function of the MLR algorithm can be briefly explained as follows. The input to the algorithm is a set $\mathscr{R}$ of $m = |\mathscr{R}|$ relations on n planar frames (i.e., the algorithm in [2] assumes a flat, two-dimensional world). Each relation $r \in \mathscr{R}$ describes the likelihood distribution of the pose of frame a relative to frame b. It is modeled as a Gaussian distribution with mean μ^r and covariance C^r. The output is the maximum likelihood (ML) estimation vector $\hat{x}$ for the poses of all the frames. In other words, the purpose of the algorithm is to assign a globally consistent set of Cartesian coordinates to the nodes of the graph based on local (relative), inconsistent (noisy) measurements, by trying to maximize the total likelihood of all measurements.

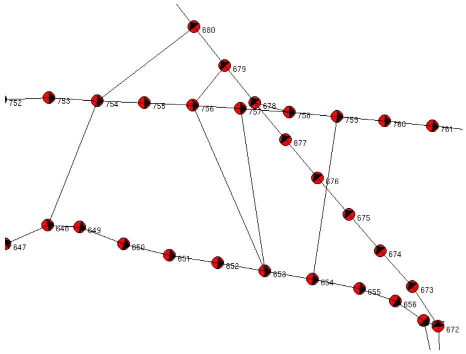

Fig. 1. The graph representation used in MLR. The figure shows the frames (nodes) and the relations (edges) both from odometry r_o and visual similarities r_v. Each frame a contains a reference to a set of features F_a extracted from the omni-directional image I_a, an odometry pose x_a^o, a covariance estimate of the odometry pose $C_{x_a^o}$, the estimated pose $\hat{x}_a$ and an estimate of its covariance $C_{\hat{x}_a}$. See also Fig. 2, which shows images from this region.

B. Relation Based on Odometry

By using odometry to add a relation r_o, the relative position change μ_{r_o} can easily be extracted directly from the odometry readings and the covariance C_{r_o} can be estimated by a motion model. In our implementation the model suggested in [15] is used where the covariance is modeled as:

$$C_{r_o} = \begin{bmatrix} d^2\delta_{X_d}^2 + t^2\delta_{X_t}^2 & 0 & 0 \\ 0 & d^2\delta_{Y_d}^2 + t^2\delta_{Y_t}^2 & 0 \\ 0 & 0 & d^2\delta_{\theta_d}^2 + t^2\delta_{\theta_t}^2 \end{bmatrix} \quad (1)$$

where d and t is the total distance traveled and total angle rotated between the two frames respectively. The 6 parameters adjust the influence of the distance d and rotation t in the calculation of the covariance matrix and were tuned manually. The δ_X parameters denote the forward motion, the δ_Y parameters the side motion and the δ_θ parameters the rotation of the robot. Note that an odometry relation r_o is only added between successive frames.

C. Visual Similarity Relation

To add a relation r_v which relies on visual similarities the likelihood distribution between two frames based on the visual similarity aspects has to be estimated. This is explained below.

1) Similarity Measure: Given two images I_a for frame a and I_b for frame b. For both images, features are extracted using the SIFT algorithm [3], which results in two sets of features F_a and F_b. Each feature $F = [x,y],H$ comprises the pixel position $[x,y]$ and a histogram H containing the

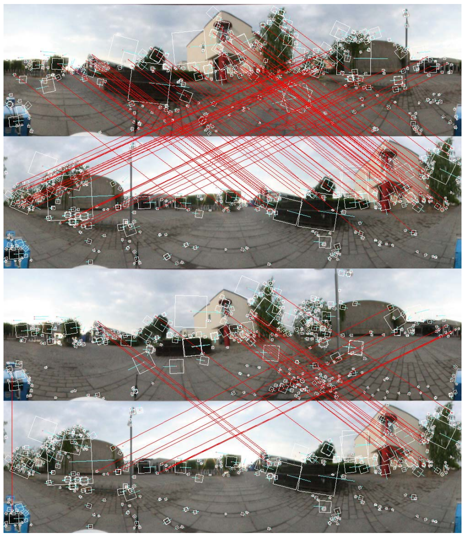

Fig. 2. Example of loop closing detection outdoors. The distance to the extracted features is comparably large. The top figure shows feature matches at a peak of the similarity value $S_{678,758} = 0.728$, whereas the lower figure shows the matches two steps away $S_{680,758} = 0.286$ ($\sim$3 meters distance). The pose variance $\sigma^2_{x^{rv}}$ and $\sigma^2_{y^{rv}}$ was estimated to be 2.16 m^2.

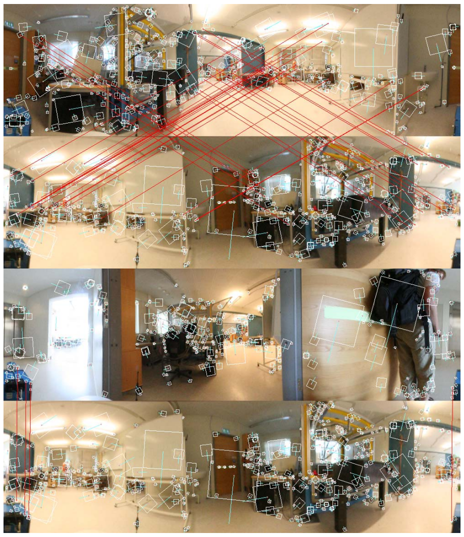

Fig. 3. Example of loop closing detection indoors. Here the distance to the features is smaller compared to Figure 2. The top figure shows matches at the local peak with a similarity value $S_{7,360} = 0.322$, whereas the lower figure shows the matches two steps away $S_{9,360} = 0.076$ ($\sim$3 meters distance). The pose variance $\sigma^2_{x^{rv}}$ and $\sigma^2_{y^{rv}}$ was estimated to be 1.27 m^2.

SIFT descriptor. The similarity measure $S_{a,b}$ is based on the number of features that match between F_a and F_b.

The feature matching algorithm calculates the Euclidean distance between each feature in image I_a and all the features in image I_b. A potential match is found if the smallest distance is smaller than 60% of the second smallest distance. This criterion was found empirically and also used in [16]. It guarantees that interest point match substansially better compared to the other feature pairs, see Fig. 2,3. In addition, no feature is allowed to be matched against more than one other feature. If a feature has more than one candidate match, the match which has the lowest Euclidean distance among the candidate matches is selected.

The feature matching step results in a set of feature pairs $P_{a,b}$, with a total number $M_{a,b}$ of matched pairs. Since the number of extracted features varies heavily depending on the image, the number of matches is normalized, hence the similarity measure $S_{a,b} \in [0,1]$ is defined as:

$$S_{a,b} = \frac{M_{a,b}}{\frac{1}{2}(n_{F_a} + n_{F_b})} \tag{2}$$

where n_{F_a} and n_{F_b} are the number of features in F_a and F_b respectively.

A high similarity meassure gives an indication that we are at a perceptually similar position. However, a single similarity measure cannot provide us with a relative position or variance estimate.

2) Estimate of the Relative Rotation and Variance: The relative rotation between two frames a and b can easily be estimated in a panoramic image by looking at the change in y pixel coordinate of the matched feature pairs $P_{a,b}$ since the width of a panoramic image encloses a complete revolution of the scene.

The relative rotations θ_p for all matched pairs $p \in P_{a,b}$ are added into a 10 bins histogram, which is smoothed with a $[1,1,1]$ kernel and the relative rotation $\mu_\theta^{r_v}$ is determined as the maximum point of a polynomial of degree two fitted to the smoothed histogram.

The rotation variance $\sigma^2_{\theta^{r_v}}$ is estimated from the sum of squared differences between the estimate of the relative rotation $\mu_\theta^{r_v}$ and the relative rotation of the matched pairs $P_{a,b}$.

$$\sigma^2_{\theta^{r_v}} = \frac{1}{M_{a,b} - 1} \sum_{p \in P_{a,b}} (\mu_\theta^{r_v} - \theta_p)^2 \tag{3}$$

3) Estimate of the Relative Position and Covariance: Our approach does not attempt to determine the position of the detected features. Therefore, the relative position between two frames a and b cannot be determined accurately. Instead we use only image similarity and set $[\mu_x^{r_v}, \mu_y^{r_v}]$ to $[0,0]$. One could of course use an estimate based on multiple view geometry, for example, but this would introduce additional complexity that we want to avoid.

However, it is possible to determine a meaningful estimate of the covariance of the relative position between frame a and b using only the similarity measure $S_{a,b}$. The number of matched features between frames will vary depending on the physical distance of the extracted features, see Fig. 4,2,3. For example, consider a robot located in an empty car park lot, where the physical distance to the features is large, and the features are fairly stable if the robot is moved one step forward. Compare this with a robot located in a narrow corridor where the physical distance to the extracted features is small. The number of matches would most likely be smaller if the robot was moved the same distance in the corridor compared to the car park.

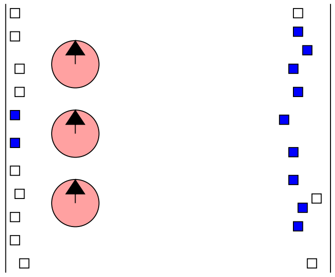

Fig. 4. Left: The physical distance to the features will influence the number of features that can be detected from different poses of the robot. The filled squares represent features that could be matched in all three robot poses while the unfilled squares represent the features were the correspondence cannot be found. The left wall in the figure is closer to the robot meaning that the features change more rapidly due to faster changes in appearance compared to the right wall which is further away.

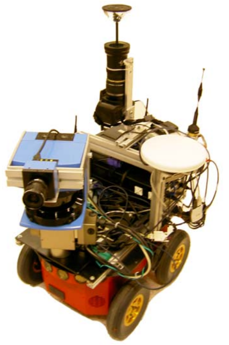

Fig. 5. Right : Our outdoor robot with the Canon EOS 350D camera and a panoramic lens from 0-360.com which were used to collect the data, a DGPS unit to determine ground truth positions, and an LMS SICK scanner used for visualization.

Hence the covariance of the robot's pose [x,y]

$$C_{r_v} = \begin{bmatrix} \sigma^2_{x^{r_v}} & \sigma_{x^{r_v}}\sigma_{y^{r_v}} \\ \sigma_{x^{r_v}}\sigma_{y^{r_v}} & \sigma^2_{y^{r_v}} \end{bmatrix} \quad (4)$$

is based on how the similarity measure varies within the neighbouring frames $N(a)$ of frame a. In order to avoid estimating the covariance orthogonal to the path of the robot if the robot was driven along a straight path, the covariance is simplified by setting $\sigma^2_{x^{r_v}} = \sigma^2_{y^{r_v}}$.

The variance is estimated by least squares fitting a 1D Gaussian function to the similarity measures $S_{N(a),b}$ and the Euclidean distance obtained from odometry. In the experimental evaluation the Gaussian was estimated using 5 consecutive frames.

4) Selecting frames to match: Consider two frames a and b, where b is the latest added frame.

By only calculating those similarity values $S_{a,b}$ for which it is likely that a and b are sufficiently close, the matching step can be speeded up. This also makes the method more robust to perceptual aliasing (where different regions have very similar appearance). If the similarity measure were to be calculated between frame b and all previously added frames, the number of feature pairs P to be matched would increase with the number of added frames.

From the SLAM method, see Section II-A we obtain a maximum likelihood estimate of the frame $\hat{x}_b$. There is, however, no estimate of the covariance $C_{\hat{x}}$ to distinguish whether frame a is likely to be close enough to calculate a similarity measure $S_{a,b}$.

If no visual relation r_v has been added, either between a and b or any of the frames between a and b, the relative covariance $C_{\hat{x}_{a,b}}$ can be determined directly from the odometry covariance $C_{x^o_a}$ and $C_{x^o_b}$. However when a visual relation $r_v^{a,b}$ between a and b is added, the covariance of the estimate $C_{\hat{x}_b}$ should be decreased. The covariance for frame b is updated with

$$C_{\hat{x}_b} = C_{\hat{x}_a} + C_{r_v^{a,b}} \quad (5)$$

if the new covariance is smaller then the previous one. The calculation was made by using the eigen vectors of the covariance matrices, i.e. representing the covariances as ellipses. The new covariance estimate is also used to update the previous frames between a and b by adding the odometry covariances $C_{x^o_{a..b}}$ in opposite order (i.e. simulate that the robot is moving backwards from frame b to a). A new covariance estimate for frame j is calculated with

$$C_{\hat{x}_j} = C_{\hat{x}_b} + C_{x^o_b} - C_{x^o_j}, \quad (6)$$

where $j \in (a,b)$. Note that the covariance is only updated if the new covariance is smaller than the previous one.

5) Visual Relation Filtering: To avoid adding visual relations based on low similarity, visual similarity relations $r_v^{a,b}$ between frame a and frame b are only added if the similarity measure exceeds a threshold t: $S_{a,b} > t$. In addition, similarity relations are only added if the similarity value $S_{a,b}$ has its peak at frame a (compared to the neighbouring frames $N(a)$). There is no limitation on the number of visual relations that can be added for each frame.

III. EXPERIMENTAL RESULTS

A large set of 945 omni-directional images was collected over a total distance of 1.4 kilometers with height differences of 3 meters. The robot was manually driven and the data were collected in both indoor and outdoor areas during a period of 2 days.

The omni-directional images were first converted to panoramic images with a resolution of 1000 x 289 before any processing was done. When extracting SIFT features the initial doubling of the images was not performed, i.e. SIFT features from the first octave were ignored, simply to lower the amount of extracted features. The mean number of extracted feature per image was 498 with a standard deviation of 170.0. The threshold t, described in Section II-C.5, was set to 0.2.

A. Visualized results

To visualize the maximum likelihood (ML) estimate $\hat{x}$ of the robot poses, laser scans acquired at the same time (and pose) as the omni-images were used to render a occupancy map. See Fig. 7 for the whole map using a 25x25 cm^2 grid size. In Fig. 8 only the center part is shown with a grid size of 10x10 cm^2.

B. Comparison to ground truth obtained from DGPS

To evaluate the accuracy of the created map, the robot position was measured with differential GPS (DGPS) and collected together with the omni-directional images, i.e. for every SLAM pose estimate there is a corresponding DGPS position $< \hat{x}_i, x_i^{DGPS} >$.

DGPS gives a smaller position error than GPS. However since only the signal noise is corrected, e.g. the problem with reflection still remains. DGPS is also only available if the radio link between the robot and the stationary GPS is functional. Therefore a subset of the pose pairs $< \hat{x}_i, x_i^{DGPS} >_{i=1..N}$ is selected. Measurements were considered only where at least five satellites were visible and the radio link to the stationary GPS was functional. The valid DGPS readings are indicated as blue dots in Fig. 6. The total number of pairs used to calculate the MSE for the whole map was 377 compared to the total number of frames which was 945.

To measure the difference between the estimated poses from SLAM $\hat{x}$ and the DGPS positions x^{DGPS} (using UTM WGS84, which provides a metric coordinate system) the two data sets have to be aligned. Since the correspondence of the filtered pose pairs is known, $< \hat{x}_i, x_i^{DGPS} >$, rigid alignment can be applied directly, e.g. using ICP [17] without searching for the closest point, see Fig. 6.

The mean square error (MSE) between x^{DGPS} and $\hat{x}$ for the map shown in Fig. 7 is 4.62 meters. This can be compared to a result of 6.22 meters for a constant average covariance of 1.72 (the average of the estimated covariances), demonstrating the increased geometric accuracy due to the new similarity-based covariance estimation method, see Fig. 9. To see how the MSE evolves over time when creating the map, MSE was calculated from the new estimates $\hat{x}$ after each new frame was added. The result is shown in Fig. 10 where the MSE that would result from using only odometry to estimate the robot's position is also plotted.

Note that the MSE was evaluated for every frame added. Therefore when the DGPS data is not available the MSE will stay constant for these frames with respect to odometry x^o. This can be seen between frames $250-440$. The MSE of the SLAM estimate $\hat{x}$ will not be constant since new estimates are computed for each frame added and loop closing also occurs indoors. The first visual relation r_v was added around frame 260. Until then, the error of the SLAM estimate $\hat{x}$ and odometry were the same. In consequence, the MSE can change quite abruptly.

IV. CONCLUSIONS AND FUTURE WORK

This paper combines two existing methods: (1) using similarity of panoramic images to close loops at the topological level, and (2) graph relaxation from odometric information and the given topology to obtain the geometric level of the map representation, and a novel method to estimate the required covariance matrix for links obtained from the vision sensor based on the visual similarity of neighbouring poses. This method uses the similarity of images to compensate for the lack of range information for the local image features, avoiding computationally expensive and less general methods such as tracking of individual image features.

From an experimental point of view, the method seems to scale well to large environments. The experimental results are presented by visual means (as occupancy maps rendered

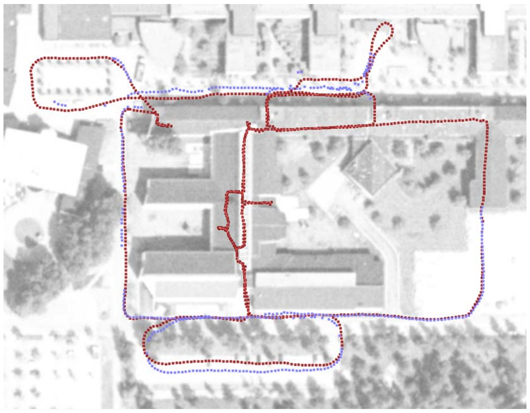

Fig. 6. DGPS data x^{DGPS} (blue) with aligned SLAM estimates $\hat{x}$ (red) displayed on an aerial image. The squares show the SLAM and DGPS poses for which the number of satellites used to obtained the DGPS measurement was considered acceptable.

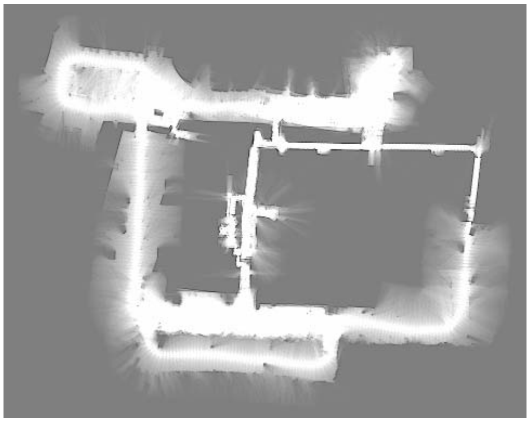

Fig. 7. Visualized map using laser range data for each image node. Note that the laser data is only used for visualization. In the rendered map the grid size is 25x25 cm^2.

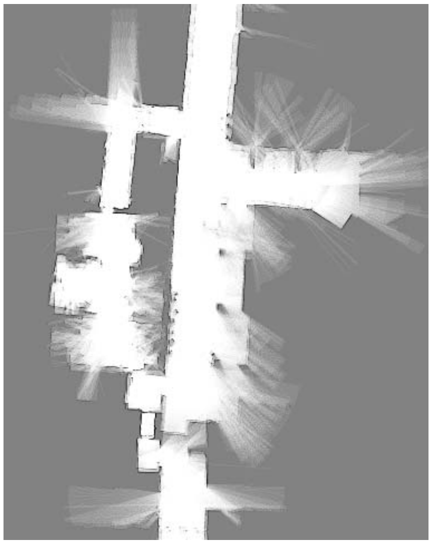

Fig. 8. Visualized map using laser range data for the center parts of the map. In the rendered map the grid size is 10x10 cm^2.

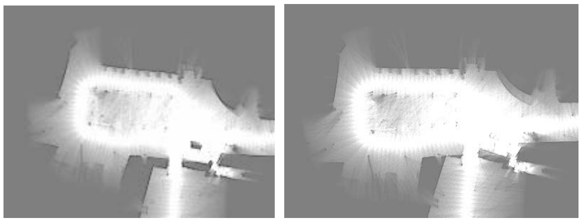

Fig. 9. Visualized map using laser range data to compare the accuracy of using (left) the esimtated covariance for each poses and (right) using a constant covariance, where the constant covariance is the mean of the estimated covariances.

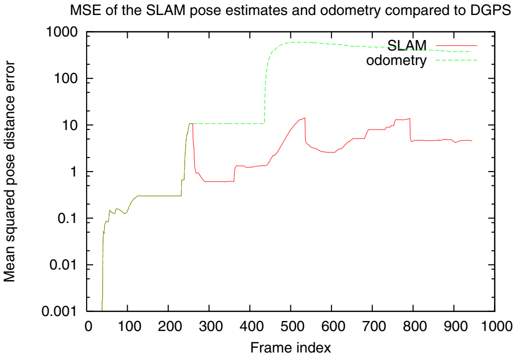

Fig. 10. The MSE of the Euclidean distance between the ground truth position obtained from DGPS readings x^{DGPS} and the SLAM estimate of the robot pose $\hat{x}$ for different states of the maps as frames are added. Drops in the MSE indicate that the consistency of the map has been increased.

from laser scans and poses determined by the SLAM algorithm) and comparison with ground truth (obtained from DGPS). These results demonstrated that the Mini-SLAM method is able to produce topologically correct and geometrically accurate maps of a large-scale environment at minimal computational cost.

The approach generates 2-dimensional maps based on 2-d motions (x, y, θ). However, it is worth noting that the ground truth positions in our experiments also contained variations of up to 3 meters in height. This indicates that the method can cope with 3-d motions to a certain extent, and we would expect a graceful degradation in map accuracy as the roughness of the terrain increases. The representation should still be useful for self-localization using 2-d odometry and image similarity, e.g., using the global localization method in [18]. In extreme cases, of course, it is possible that the method would create inconsistent maps, and a 3-d representation should be considered.

The bottleneck of the current implementation in terms of computation time is the calculation of image similarity, which involves the comparison of many local features. The suggested approach, however, is not limited to the particular measure of image similarity used in this work. There are many possibilities to increase the computation speed either by using alternative similarity measures that are faster to compute while being still distinctive enough, or by optimizing the implementation, for example, by executing image comparisons on a graphics processing unit (GPU) [19].

Plans for future work include a thorough run-time evaluation of the approach, an investigation of the possibility of using a standard camera instead of an omni-directional camera, and to incorporate vision-based odometry to realise a completely vision-based system.

References

[1] D. C. A. Samer M. Abdallah and J. S. Zelek, “Towards benchmarks for vision SLAM algorithms,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 1542–1547.

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

[3] D. Lowe, “Object recognition from local scale-invariant features,” in *Proc. Int. Conf. Computer Vision ICCV, Corfu*, 1999, pp. 1150–1157.

[4] S. Se, D. Lowe, and J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks,” *International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[5] T. Barfoot, “Online visual motion estimation using FastSLAM with SIFT features,” in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005, pp. 579–585.

[6] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, “A framework for vision based bearing only 3D SLAM,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 1944–1950.

[7] N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, “The vSLAM algorithm for robust localization and mapping,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2005, pp. 24–29.

[8] P. Elinas, R. Sim, and J. Little, “σSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 1564–1570.

[9] J. Sez and F. Escolano, “6dof entropy minimization slam,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 1548–1555.

[10] A. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV’03)*, 2003, pp. 1403–1410.

[11] K. L. Ho and P. Newman, “Loop closure detection in SLAM by combining visual and spatial appearance,” *Robotics and Autonomous System*, vol. 54, no. 9, pp. 740–749, September 2006.

[12] P. M. Newman, D. M. Cole, and K. L. Ho, “Outdoor SLAM using visual appearance and laser ranging,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 1180–1187.

[13] C. Chen and H. Wang, “Large-scale loop-closing with pictoral matching,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 1194–1199.

[14] A. Ranganathan, E. Menegatti, and F. Dellaert, “Bayesian inference in the space of topological maps,” *IEEE Transactions on Robotics*, vol. 22, pp. 92– 107, 2005.

[15] A. I. Eliazar and R. Parr, “Learning probabilistic motion models for mobile robots,” in *Proc. of the twenty-first Int. Conf. on Machine learning (ICML)*, 2004, p. 32.

[16] J. Gonzalez-Barbosa and S. Lacroix, “Rover localization in natural environments by indexing panoramic images,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2002, pp. 1365–1370.

[17] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[18] H. Andreasson, A. Treptow, and T. Duckett, “Localization for mobile robots using panoramic vision, local features and particle filter,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2005, pp. 3348–3353.

[19] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, “GPU-based video feature tracking and matching,” in *Workshop on Edge Computing Using New Commodity Architectures (EDGE 2006)*, Chapel Hill, 2006.