Evaluation of Pose Only SLAM

Gibson Hu, Shoudong Huang and Gamini Dissanayake

Abstract— **In recent SLAM (simultaneous localization and mapping) literature, Pose Only or Graph Based optimization methods have become increasingly popular. This is greatly supported by the fact that these algorithms are computationally more efficient, as they focus more on the robots trajectory rather than dealing with complex map. Implantation simplicity allows these to handle both 2D and 3D environments with ease. This paper presents a detailed evaluation of the reliability and the accuracy of Pose Only SLAM, and aims at providing a definitive answer to whether optimizing poses is more advantages than optimizing features. Focus is centered around TORO, a Tree based network optimization algorithm, which has gained increase recognition within the robotics community. We compare this with Least Squares, which is considered one of the best Maximum Likelihood method available. Results based on both simulated and real data in 2D environments, are presented to substantiate the conclusions.**

I. INTRODUCTION

One of the main focuses of current SLAM research is on developing solutions to improve SLAM efficiency without compromising the accuracy. SLAM itself is an optimization problem, solved by combining the information gained from sensor observations and robot odometry. Researchers have typically resorted to simplifying data and fitting it around point feature based solutions which aims to compute optimal locations of both features and poses [1].

Recently, Pose Only SLAM approaches have gained popularity [2] [3]. A Pose Only implantation typically divides SLAM into two separate phases, one is for the identification of position constraints and the second is the optimization of robot poses. During the first phase, the consistency of information use must be monitored and information reuse must be avoided. For the second phase, the major focus is on efficiency and accuracy, that is, how to get a good quality solution quickly.

A popular Pose Only SLAM algorithms is Tree-based Network Optimizer or TORO. It has been evaluated to be much faster than most standard maximum likelihood approaches and stated to work well in many different applications[3].

When it is difficult or impossible to extract features from the sensor data, Pose Only SLAM is probably a good choice for optimizing the robot poses and locating the robot in an unknown environment. However, if there are good quality features that can be extracted from the environment, how much accuracy or consistency is compromised for efficiency in TORO or Pose Only SLAM?

In this paper, we assume the point feature based SLAM set up and ask the above question. We also want to know whether it is necessary to compute the optimal locations of landmarks at all if an accurate position can be gained from Pose Only SLAM. In other words, how accurate are the implementations of Pose Only SLAM when compared with the optimal solution. We basically want to look at how TORO's result differs from a Least Squares based optimization where information use is maximized. We also look at how a conservative approach can be used to maintain consistency to obtain fair comparisons.

The paper is structured as follows. Section II explains the three optimization techniques we used to conduct our experiments. Section III explains our approach to maintaining consistency when obtaining Pose Only constraints. Section IV describes our evaluations methods. Section V presents experimental results and Section VI discussed some related work. Finally, Section VII draws conclusions on our findings.

II. THREE OPTIMIZATION TECHNIQUES

A fair comparison to any estimated solution can be made by providing a Least Squares benchmark. Unlike methods such as Extended Kalman Filtering (EKF), the Least Squares solution keeps all the variables in its state vector avoiding any information loss. Often search times can be decreased by using a Levenberg Marquardt implementation. The computational efficiency of the Least Squares implementation, is improvable by means of map joining or exploiting the sparseness of it information matrix [6]. The Least Squares result is arguably one of the most accurate estimations one can achieve.

The basic principle behind Least Squares is the minimization of the error function

$$(Z - F(X))^T P^{-1}(Z - F(X))$$

where X is the state vector, Z is the measurement information and P is the covariance of the measurement. The problem itself is not always linear therefore the state vector should be solved iteratively by

$$X_{k+1} = H^{-1} \cdot J^T \cdot P^{-1}(Z - F(X_k) + J \cdot X_k))$$

where J is the Jacobian and H is the Hessian $(J^T P^{-1} J)$.

In this paper we use Least Squares in two ways. Optimizing both landmarks and robot poses or just on poses.

A. Full Least Squares SLAM

Assuming the measurement information Z containing both odometry information and observations from robot poses to

Gibson Hu, Shoudong Huang and Gamini Dissanayake are with Faculty of Engineering and Information Technology, University of Technology Sydney, PO Box 123 Broadway NSW 2007, Australia {g.hu,sdhuang,gdissa}@eng.uts.edu.au

features. A Full Least Squares solution takes into consideration all this information and optimizes everything in one go.

The state vector X in this algorithm thus contains all the robot poses and all the feature positions. That is

$$X = (R_1, R_2, \cdots, R_m, L_1, L_2, \cdots, L_n)$$
$$= (x_1^r, y_1^r, \phi_1^r, \cdots, , x_1^l, y_1^l \cdots)$$

where R_i is the global robot pose and L_i is global landmark position.

The measurement vector Z contains all the available odometry and observation information

$$Z = (R_{1,2}, L_{1,1}, L_{1,2}, \cdots R_{2,3}, L_{2,1}, L_{2,2} \cdots).$$

Here $R_{i,i+1}$ is the odometry (a constraint between 2 adjacent poses) and $L_{i,j}$ is the observations made from pose i to landmark j.

A major issue associated with the Full Least Squares approach, is that when the environment becomes complex with multiple features the solution can often diverges. The non linearity of the observation and the motion models is one factor.

B. Pose Only Least Squares

If we can somehow transfer the original pose-to-feature observation information into relative pose/constraint information, then we can apply the Pose Only SLAM techniques. Still, when the relative pose constraints information is given, it can be argued that a Least Squares implementations will provide the best achievable solutions for optimizing the poses.

In this case, the state vector contains robot poses only and is expressed by

$$X_{pose} = (R_1, R_2, \cdots, R_m).$$

The measurement information available is now

$$Z_{pose} = (R_{(1,2)}, R_{(1,3)}, R_{(1,4)}, \cdots, R_{(2,3)}, R_{(2,4)} \cdots)$$

where $R_{(i,j)}$ is the odometry from i to pose$j$.

C. TORO

The input to TORO is also Z_{pose}. TORO is an efficient pose only SLAM algorithm combining the ideas Grisetti et al[3] and research done by Olson et al [13], who was one of the first to introduce Stochastic gradient decent (SGD) to graph based approaches. Which resulted in faster processing times without approximate factoring of the probability density function.

The SGD equation is governed by

$$X^{t+1} = X^t + \lambda \cdot KJ^T P^{-1}(Z - F(X^t))$$

Here the state vector X^t containing only poses. J is the Jacobian of the error function, and $(Z - F(X^t))$ is the residual and K is a pre-conditioning matrix computed from the Hessian H. It is used to scale the variations resulting from the Jacobian depending on the curvature of the error surface [10]. λ is the learning rate that improves convergence.

TORO uses a tree based parametrization approach to decomposing the optimization approach. This greatly increases the speed of the pose only algorithm but it is unclear how much information is sacrificed through this process. During this process constraints are randomly selected and categorized into either *open* or *closed* loop constraints. A SGD optimization is used to reduce the transformation error and a SLERP(Spherical linear interpolation) is further added to evenly distribute rotational error. Finally optimized pose state can be recovered [10].

The method argues that by selecting more important constraints to use, optimization can still produce a near accurate solution. TORO also claims that it runs fast enough to incrementally build the graph maps while simultaneously run any desired movement application. The major drawback of TORO which is not present in a Full Least Squares solution, is the inability to hand non spherical covariances. Therefore TORO can only be used to optimize robot poses and only performs well with spherical covariances.

III. OBTAINING CONSTRAINTS

Before applying the Pose Only SLAM approach, we find getting information Z_{pose} from Z becomes a critical step. The process needs to be carefully performed such that information can be extracted with limited or no information reused. For example, we can not simply use the observations made from robot pose i many times to obtain the relative pose constraints between pose i and other poses.

In this paper, we proposed two different methods to obtain the constraint without information reuse. The two methods are both based on the following basic idea: under Gaussian noise assumption, a single observation $L_{i,j}$ with covariance $P_{i,j}$ is equivalent to k observations with covariance $k \times P_{i,j}$ in terms of information content. These methods cannot be considered as novel approaches but satisfactory for addressing our Pose Only SLAM evaluation.

A. Method 1

Method one sticks to a basic framework. Observations are only used twice, which implicates the build up of relationships only between adjacent poses. eg 0-1, 1-2, 2-3, ... , n-0. Since each odometry information is only used once, we need to double the observation covariance for each observation that is used.

Algorithm 1 Simple Extraction Method

```
1: Associate features between only adjacent poses
2: Working Sequentially, find the relative pose between each adjacent pose while doubling the covariance on features observed from the first pose.
3: Loop for all poses
4: Perform (2-3) to obtain final constraint between first pose and last pose.
5: End
```

Method 1, while being simple, many other relative pose constraints may not be obtained, thus resulting in some information loss.

B. Method 2

In method 2 we aim to maximize the information usage by trying to build as many relative pose relationships as possible. Consistency with information usage is approached in an offline perspective. The reason being that we want to find out the full potential of Pose Only SLAM optimization and do not want to compromise any information loss. Method 2 also has the support for complex trajectories with multiply loop closing scenarios.

Algorithm 2 Multi Pose Relative Information Extraction

```
Obtain observations from all poses.
Choose pose pairs with enough common features to find constraints.
Include odometry information for adjacent poses
Multiply covariance of observation by observation frequency.
Apply Least Squares to obtain relative pose constraint.
Loop for all poses and build up constraints vector Z with new covariance P.
Use Least Squares to obtained optimized state vector
End
```

To confidently obtain constraints without divergence or environmental symmetry, an adequate amount of landmarks has to be associated. Also a good initial estimate on the constraint typically guarantees convergence. For the purpose of evaluation, we simply use the ground truth, from simulation, as our initial estimate.

To find out which method is more effective, an additional experiment is conducted, see section V.

IV. QUANTIFICATION OF ESTIMATION RESULTS

Now we have three different approaches to solve the SLAM problem.

A. input Z, [Full Least Squares], output X.
B. input Z, [Method 1 or 2], transfer to Z_{pose}, [Pose Only Least Squares], output X_{pose}.
C. input Z, [Method 1 or 2], transfer to Z_{pose}, [TORO], output X_{pose}.

How to compare the results? What to compare?

In this paper, the comparisons will focus on estimation consistency and accuracy. Estimation consistency is a crucial requirement for any algorithm. Roughly speaking, an estimation algorithm is consistent if the uncertainty of the algorithm accurately represents the actual estimation error. It is said that estimation consistency, in a SLAM algorithm, holds greater value than computational efficacy [6].

A. 2σ bound check on consistency

One simple way to evaluate the consistency is by comparing the actual estimation error with its 2σ bound.

Another way of quantify the consistency is to compute the average normalized estimation error squared or (NEES). Commonly known as a (χ^2) Chi Square test with n degree of freedom.

B. NEES 1, Consistency check on robot pose estimate

An average NEES can only be done where ground truth is available. This test allows us to see the exact consistency between our two pose only SLAM methods and a Full Least Squares result. Before we can do this test however, poses and respective covariances must be extracted from the Full Least Squares result.

The NEES equation is

$$(X_{pose}^{true} - X_{pose})^T P_{pose}^{-1} (X_{pose}^{true} - X_{pose})$$

where X_{pose}^{true} is ground truth robot positions. X_{pose} is estimated robot poses obtained from the algorithms, and P_{pose} is the covariance matrix of the estimate.

In addition a comparison can also be made against a Gate values , a 95% probability concentration region of a χ^2 distribution.

C. NEES 2, Consistency of Pose Only methods

A NEES on constraint error can be performed when evaluating the optimization accuracy of TORO when compared with Pose Only Least Squares. We are also able to see how much information is lost through the approximation techniques used in TORO.

To test the NEES using constraint information a modified NEES equation is used.

$$(Z_{pose} - F_{pose}(X_{pose}))^T P_{rel}^{-1} (Z_{pose} - F_{pose}(X_{pose}))$$

Here Z_{pose} is the relative pose constraint, and P_{rel} is the corresponding covariance matrix, $F(X_{pose})$ is the function relating the poses X_{pose} to the constraints Z_{pose}.

D. NEES 3, Consistency comparison to feature based SLAM

A way to compare all three algorithms (TORO, Pose Only Least Squares and Full Least Squares) is by comparing their results against the initial data Z. To justify an accurate comparison all state vectors must be equal. A problem arises where the Pose Only results do not have landmarks associated.

To over come this, we can simply take the robot poses obtained from TORO (or Pose Only Least Squares), fix their values and find the corresponding feature that best fit the data Z, thus an estimate of X is obtained and can be used to compute the chi square test.

In this test Z and P are the initial data and covariance values used for Full Least Squares. Containing information of all odometry and observations, X is the optimized results from Full Least Squares and Pose Only after pose fixing, now containing landmarks and robot poses.

V. Experiments

A. Simulation

Firstly, a test is conducted to identify the accuracy of our relative pose extraction methods. The aim here is to understand the significance of information loss present when method 1 is used over method 2. When looking at Fig. 1, we can see that the uncertainty generated from using method 1 is much greater with some values reaching 18 meters. It is clear that the majority of error lies outside the sigma bound of method 2. Knowing this, we can be confident that using method 2 over 1 would be more meaningful for further experiments.

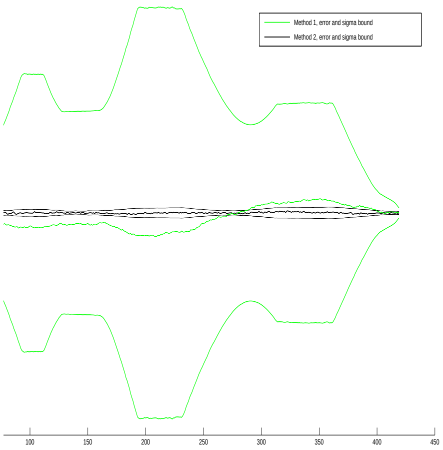

Fig. 1. Pose Only Least Squares, Method 1 (black) and Method 2(green) Sigma Bound Comparison

The experiments now follow the procedure outlined below.

- Simulate a trajectory, obtain retrieve odometry and observation information.
- Get constraints from method 2.
- Apply Least Squares and TORO Pose Only optimizer.
- Conduct a full Least Squares on initial data.
- Evaluate results by testing all 3 NEES functions.

4 Simulated trajectories are trailed with an additional experiment targeting on live data.

Each Simulation has a path surrounded by a normal distribution of 225 point features. Fig.1 shows the 4 different trajectories generated, 2 simple (82 Poses) and 2 complex scenarios (420 Poses). The more complex trajectories offer several loop closing positions where the simple scenario only has a single point of return.

For each step the robot moves 0.5 meters, turning in a predefined angle and tries to observe any features within its sensor range (5m with 180 degree field of view). For each trajectory we test for 3 different noise level scenarios. a) Regular environment where Sensor and Odometry noise is low. b) Changes in terrain resulting in higher odometry noise. c) Environmental effect resulting in higher sensor noise. The noise values are distributed using a gaussian model, the mean is described in Table I. Simulations for each noise type are repeated 10 times. The mean and standard deviation of their NEES values are listed in Tables II,III.

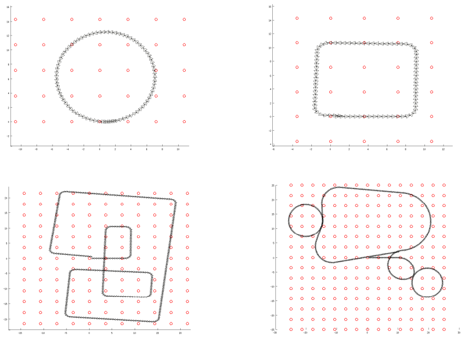

Fig. 2. Simulated Trajectories Top Left:1, Right:2, Bottom Left:3, Right:4

The DLR-Spatial-Cognition data set was the live data evaluated. The data set is available at *http://www.sfbtr8.spatial-cognition.de/insidedataassociation/data.html*. It was collected using a robot equipped with a camera, moving around in a building scattered with artificial landmarks (white/black circles) placed on the ground. The image data has been preprocessed and the relative position of the observed landmarks with respect to the observation point are provided. This data contains both odometry and landmark measurement with good covariance. Preprocessing of data has been performed with known data association. There is a total of 3296 poses and 576 landmarks with 14309 observation. Method 2 is used to obtain constraints, with a threshold of at least 5 common landmarks needed between observations. Because ground truth is not known the NEES 1 test cannot be performed.

B. Results

The results show high consistency between Pose Only Least Squared and Full Least Squared implementations. Unfortunately, it seems, TORO's sometimes unable to handle several cases in both simple and complex cases, resulting in inconsistency and high standard deviation. Especially when the noise in the sensors become high, case(c), TORO becomes more susceptible to bad estimation.

NEES 2 results confirms Pose Only to be a much more robust optimization technique. The constraint data is represented with far better accuracies.

A very interesting observation can be made when we look at the outcome of NEES 3. When using Pose Only Least Squares there is high indication that only a small amount of information is lost from the original data. This is evident when we compare this with the Full Least Squares Result. In most cases there is only a 40% increase in the NEES value. Which leads us to believe that optimization of landmark is not such a big component when it comes to back end SLAM.

Finally, results from the DLR data set, Table IV, supports our claim with its result reflecting those of the simulations.

TABLE I
MEAN NOISE SEED

Noise Type	Odometry			Observation	
	dx (m)	dy (m)	dtheta (rad)	dx (m)	dy (m)
a	0.1	0.1	0.05	0.1	0.1
b	0.2	0.2	0.2	0.1	0.1
c	0.1	0.1	0.05	0.3	0.3

TABLE II
NEES TESTS, MEAN VALUES FROM 10 SIMULATIONS

		NEES 1				NEES 2		NEES 3		
Trajectory	Noise Type	GATE	Pose Only LS	TORO	Full LS	Pose Only LS	TORO	Pose Only LS	TORO	Full LS
1	a	280.36	152.76	413.73	190.74	33.96	301.73	539.13	1039.25	327.50
1	b	280.36	128.79	179.76	189.35	18.47	67.78	612.41	1010.63	339.15
1	c	280.36	188.33	2280.94	193.19	34.13	2145.99	362.40	2505.23	327.76
2	a	280.36	157.80	193.73	195.88	29.089	82.85	501.19	569.44	339.05
2	b	280.36	125.33	144.53	192.56	16.62	39.08	501.28	614.02	336.09
2	c	280.36	181.44	722.07	189.25	31.45	613.88	398.41	1004.85	342.19
3	a	1340.59	628.77	2561.88	957.89	697.97	2958.97	3290.44	7146.97	3135.11
3	b	1340.59	551.32	3022.718	968.88	562.01	3120.39	3252.42	8666.17	3137.72
3	c	1340.59	798.08	10688.30	965.68	736.19	11180.28	3264.81	15581.44	3121.31
4	a	1340.59	661.12	2273.29	987.11	641.43	2523.50	3248.97	6826.11	2970.42
4	b	1340.59	581.17	1475.20	975.03	392.25	1351.07	3254.75	6550.57	2652.96
4	c	1340.59	809.64	8851.06	980.33	674.88	9264.40	3187.42	13135.66	2988.83

TABLE III
NEES TESTS, STANDARD DEVIATION VALUES FROM 10 SIMULATIONS

		NEES 1			NEES 2		NEES 3		
Trajectory Type	Noise Type	Pose Only LS	TORO	Full LS	Pose Only LS	TORO	Pose Only LS	TORO	Full LS
1	a	9.60	391.54	14.09	5.76	359.16	80.77	669.99	16.79
1	b	19.26	74.7	16.90	3.13	48.66	104.46	471.12	21.05
1	c	12.42	354.64	11.30	5.74	351.48	16.06	358.56	14.11
2	a	12.79	18.49	10.51	3.29	12.82	31.38	69.19	11.10
2	b	12.0	17.8	11.77	2.84	12.67	26.05	96.61	16.3
2	c	12.42	110.32	12.0	4.64	111.50	23.02	120.62	17.87
3	a	14.62	328.61	39.06	12.95	328.76	90.83	743.58	46.89
3	b	24.26	3288.33	49.23	14.88	3328.61	49.24	6520.58	48.59
3	c	42.39	692.99	46.80	17.04	701.09	42.69	1236.35	30.23
4	a	14.62	328.61	39.06	12.95	328.76	90.83	743.58	46.89
4	b	41.02	466.24	35.2	196.43	727.41	56.74	1258.82	536.50
4	c	42.39	692.9	46.80	17.04	701.09	42.69	1236.35	30.23

TABLE IV
NEES DLR DATA 3296 POSES

NEES 2	Pose Only LS	TORO	
	24413.73	5904053.101	
NEES 3	Pose LLS	TORO	Full LS
	64037.788	6041301.618	48346.732

VI. DISCUSSION AND RELATED WORK

Recent work into graph based SLAM methods show TORO to be far superior to SAM (Smoothing and Mapping) in terms of error reduction [10]. Especially under high rotational noise TORO converges confidently within 250 iterations where SAM does not. Grisetti also write, his method compared with MLR and Olson [13] is able to converge to the same error under much less iterations. This motivates us to perform a proper evaluation of trajectory based SLAM algorithms especially TORO.

Results from all 3 simulations indicate evidence of pose only SLAM performing exceptionally well. Upon closer examination, the Pose Only Least Squares result stays relatively similar to TORO when information is only used twice (and covariance matrix is doubled). When method 2 is applied the approximation effects of TORO are definitely noticed. The NEES 1 test justifies Pose Only SLAM being quiet effective in optimizing error, staying below the gate value during the majority of tests. When we do a full comparison with the Full Least Squares, indicated in NEES 3, we can see little information is lost when SLAM is split into two problems.

When we applied the same testing procedures on the DLR data set, the NEES 3 error of TORO increased significantly while Least Squares showed a slight rise. Due to fact that real data could have some bad uncertainty estimations, we can see that the TORO method may amplify this when arriving at its

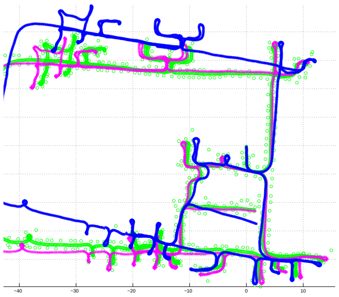

Fig. 3. DLR results (Green: Full Least Squares Pink: Pose Only Least Squares Blue: TORO)

solution.

Nowadays more and more SLAM algorithms are being developed. Evaluation of different SLAM algorithms is becoming an important issue and has attracted more attention in the past few years. For example, Burgard et. al [5] and Kummerle et.al [12] provided an objective benchmark for comparing different trajectory based SLAM algorithms. The metric use only the relative position of poses for comparing the accuracy of the trajectories obtained from different SLAM algorithms, which allows us to compare SLAM approaches that use different estimation techniques or different sensor modalities since all computations are made based on the corrected trajectory of the robot. In [6], some performance metrics for comparing the consistency, accuracy and efficiency of different point-feature based SLAM algorithms are proposed. Moreover, a number of research groups [7][8] have collected large-scale experimental data with accurate ground truth such that different SLAM algorithms can be evaluated using real data.

One important issue in SLAM or any other information fusion techniques is information reuse. One way to deal with information reuse is first use whatever information possible to get the estimate, then applying Covariance intersection (CI) (see [14] and [15]), that facilitates combining two correlated pieces of information, when the extent of correlation itself is unknown is used to fuse these two estimates. Another way to separate the observations made from a particular pose into two parts, one part is used to compute the relative pose with respect to the previous pose and the other part is used to compute the relative pose to the next pose [9]. However, both ways cause some information loss.

In this paper, we dealt with this issue differently. We assume the Gaussian noise assumption and separate one single observation into different parts each with reduced amount of information (enlarged covariance matrix). This provides us the information fusion results without information reuse and without information loss.

VII. Conclusion

After careful evaluation we can see that Pose Only Least Squares is able to achieve results both accurately and reliably without much information loss, provided that the relative pose information is extracted properly. When benchmarks with a full Least Squares solution Pose Only seems show good consistency as long as information is used optimally. The results from method 1 and 2 show that method 2 to be far superior for extracting relative pose information, and should be used to maintain consistency.

It is evident that approximations involved in TORO does seem to affect its ability to fully optimize the performance. The tradeoff between efficiency and accuracy is still undetermined. We can conclude that accurate poses can definitely be achieved with Pose Only SLAM, and the increase in computational efficiency is definitely a bonus. However it is still hard to argue if just optimizing poses is a good reason for ignore features completely. More experiments would need to be conducted, but using Pose Only SLAM is definitely a positive direction to take.

References

[1] F. Dellaert and M. Kaess, “Square root SAM: Simultaneous localization and mapping via square root information smoothing”. *International Journal of Robotics Research*, vol. 25, no. 12, December 2006, pp. 1181-1203.

[2] F. Lu and E. Milios. “Globally consistent range scan alignment for environment mapping”. *Autonomous Robots*, 4:333-349, 1997.

[3] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard: A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Decent., *Robotics: Science and Systems (RSS)*, 2007

[4] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson and W. Burgard, “Online constraint network optimization for efficient maximum likelihood mapping”. *In Proceedings of 2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, California, on May 19-23, 2008.

[5] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kuemmerle, C. Dornhege, M. Ruhnke, A. Kleiner, J. D. Tardos, A Comparison of SLAM Algorithms Based on a Graph of Relations. IROS, 2009

[6] S. Huang, Z. Wang, G. Dissanayake, and U. Frese, Iterated D-SLAM Map Joining: Evaluating its performance in terms of consistency, accuracy and efficiency. *Autonomous Robots*, (2009) 27: 409-429

[7] J. L. Blanco, F. A. Moreno, J. Gonzalez, A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, (2009) 27: 327-351.

[8] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, P. Taddei, Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, (2009) 27: 353-371.

[9] Eustice, R. M., Singh, H., and Leonard, J. 2005. Exactly sparse delayed-state filters. In *Proc. IEEE International Conference on Robotics and Automation*, pp. 2428-2435.

[10] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard: Non-linear Constraint Network Optimization for Efficient Map Learning, *IEEE Transactions on Intelligent Transportation Systems*, Volume 10, Issue 3, Pages 428-439, 2009

[11] Grisetti Giorgio, Slawomir Grzonka, Cyrill Stachniss, Patrick Pfaff, and Wolfram Burgard: Efficient Estimation of Accurate Maximum Likelihood Maps in 3D., *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007

[12] R. Kummerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, A. Kleiner, On measuring the accuracy of SLAM algorithms, *Autonomous Robots*, (2009) 27: 387-407.

[13] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates, *In Proc. of the IEEE Int Conf on Robotics and Automation (ICRA)*, pages 2262 -2269, 2006

[14] Chen, L., Arambel, P. O., and Mehra, R. K. 2002. Estimation under unknown correlation: covariance intersection revisited. *IEEE Transactions on Automatic Control* 47(11):1879-1882.

[15] Julier, S. J., and Uhlmann, J. K. 2001. Simultaneous localization and map building using split covariance intersection. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1257-1262.