Graph Optimization with Unstructured Covariance: Fast, Accurate, Linear Approximation

Luca Carlone[1], Jingchun Yin[1], Stefano Rosa[2], and Zehui Yuan[1]

[1] Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy
{luca.carlone,jingchun.yin,zehui.yuan}@polito.it
[2] Italian Institute of Technology (IIT), Torino, Italy
stefano.rosa@iit.it

Abstract. This manuscript addresses the problem of optimization-based Simultaneous Localization and Mapping (SLAM), which is of concern when a robot, traveling in an unknown environment, has to build a world model, exploiting sensor measurements. Although the optimization problem underlying SLAM is nonlinear and nonconvex, related work showed that it is possible to compute an accurate linear approximation of the optimal solution for the case in which measurement covariance matrices have a block diagonal structure. In this paper we relax this hypothesis on the structure of measurement covariance and we propose a linear approximation that can deal with the general unstructured case. After presenting our theoretical derivation, we report an experimental evaluation of the proposed technique. The outcome confirms that the technique has remarkable advantages over state-of-the-art approaches and it is a promising solution for large-scale mapping.

Keywords: Pose graph optimization, Simultaneous Localization and Mapping, Mobile robotics.

1 Introduction

In several application scenarios (e.g., search and rescue, planetary exploration, disaster response) mobile robots are deployed in an unknown environment and are required to build a model (*map*) of the surroundings. The map is often used for planning human intervention or for enhancing situational awareness. Therefore, the mapping process is guided by three main requirements: (i) *accuracy*, since a misleading representation of the environment can seriously compromise the operation of human (or robotic) operators within the scenario, (ii) *efficiency*, since it is crucial to have time-sensitive information on the environment, (iii) *scalability*, since the robot may be in charge of mapping large areas.

Pose graph optimization has recently emerged as an effective problem formulation for SLAM. In a *pose graph*, each node represents a pose assumed by a mobile robot at a certain time, whereas an edge exists between two nodes if a relative measurement (inter-nodal constraint) is available between the corresponding

I. Noda et al. (Eds.): SIMPAR 2012, LNAI 7628, pp. 261–274, 2012.

poses. Inter-nodal constraints are usually obtained by means of proprioceptive sensors (*odometry*) or exteroceptive sensor-based techniques (*vector registration*, *scan matching*, etc.) [2]. Then, the objective of pose graph optimization is to estimate the nodes' poses (*pose graph configuration*) that maximize the likelihood of inter-nodal measurements. The utility of estimating the configuration of the pose graph stems from the fact that, from the estimated poses and from sensor measurements, it is then easy to construct a map of the environment. After the seminal paper [12], several authors put their efforts in devising sustainable and accurate solutions to pose graph optimization. Thrun and Montemerlo [17] enabled the estimation of large maps using a conjugate gradient-based scheme. Konolige [9] investigated a reduction scheme for reducing the number of nodes involved in the optimization. Frese et al. proposed a multilevel relaxation approach for full SLAM [7]. Olson et al. [14] proposed the use of incremental pose parametrization for improving efficiency and convergence. Grisetti et al. [8] extended such framework, taking advantage of the use of stochastic gradient descent in planar and three-dimensional scenarios. In [11] the authors presented a general framework for the optimization of graph-based nonlinear error functions. More recently, Sünderhauf et al. [16] stressed the topic of outliers rejection in pose graph optimization, proposing a strategy for discarding erroneous loop closure constraints. All the aforementioned techniques are iterative, in the sense that, at each iteration, they solve a local convex approximation of the original problem, and use such local solution to update the configuration [6]. This process is then repeated until the optimization variable converges to a local minimum of the cost function. As a consequence, all mentioned techniques require the availability of an initial guess for nonlinear optimization, which needs to be sufficiently accurate for the technique to converge to a global solution of the problem. A partial answer to these two problems (computational complexity and need of an accurate initial guess) came from the work [3]. In [3] the authors proposed a linear approximation for the pose graph configuration, assuming that the measurement covariance matrices have a block diagonal structure. The approach requires no initial guess and was shown to be accurate in practice.

In this article we extend and complement the previous work [3], proposing two contributions: (i) we relax the hypothesis of structured measurement covariances and we propose an approach that is able to deal with the full covariance case; (ii) we present an extensive evaluation of the performance of the proposed approach, compared with the approach of [3] and with other state-of-the-art techniques. The first contribution (Section 3) is more theoretical: we describe an algorithm for estimating the pose graph configuration and we then prove that it corresponds to a Gauss-Newton steps around a suitable suboptimal solution. The algorithm is an extended version of the approach proposed in [3]; also the proof proceeds on the same line, although it encompasses the more general case of full covariances. The second contribution (Section 4) is experimental. We test several state-of-the-art techniques on real and simulated datasets and we propose a performance evaluation in terms of accuracy, efficiency, and scalability.

2 Problem Formulation

The objective of pose graph optimization is to provide an estimate of the poses assumed by a mobile robot, namely $X = \{x_0, \dots, x_n\}$. X is called a *configuration* of poses; the $n+1$ poses are in the form $x_i = [p_i^\top \ \theta_i]^\top \in \mathrm{SE}(2)$, where $p_i \in \mathbb{R}^2$ is the Cartesian position of the i-th pose, and θ_i is its orientation. The input for the estimation problem are m measurements of the relative pose between pairs of nodes. For instance a measurement $\bar{\xi}_{i,j}$ between nodes i and j is in the form

$$\bar{\xi}_{i,j} = \xi_{i,j} + \epsilon_{i,j} = \begin{bmatrix} R_i^\top (p_j - p_i) \\ \langle \theta_j - \theta_i \rangle_{2\pi} \end{bmatrix} + \begin{bmatrix} \epsilon_{i,j}^{\Delta} \\ \epsilon_{i,j}^{\delta} \end{bmatrix}, \tag{1}$$

where $\xi_{i,j}$ is the true (unknown) relative pose between node i and node j, $\epsilon_{i,j} \in \mathbb{R}^3$ is the *measurement noise*, $R_i \in \mathbb{R}^{2\times 2}$ is a planar rotation matrix of an angle θ_i, $\langle \cdot \rangle_{2\pi}$ is a modulo-(2π) operator that forces angular measurements in the manifold $\mathrm{SO}(2)$, and $\epsilon_{i,j}^{\Delta}$ and $\epsilon_{i,j}^{\delta}$ are the (possibly correlated) *Cartesian* and *orientation noise*. According to related literature, we assume $\epsilon_{i,j}$ to be zero mean Gaussian noise, i.e., $\epsilon_i^j \sim \mathcal{N}(\mathbf{0}_3, P_{i,j})$, being $P_{i,j}$ a 3 by 3 covariance matrix. $\xi_{i,j}$ describes the relative transformation that leads pose i to overlap with pose j. We can rewrite each measurement as $\bar{\xi}_{ij} = [(\bar{\Delta}_{i,j}^l)^\top \ \check{\delta}_{i,j}]^\top$, where $\bar{\Delta}_{i,j}^l \in \mathbb{R}^2$ denotes the *relative position* measurement, and $\check{\delta}_{i,j} \in \mathrm{SO}(2)$ denotes the *relative orientation* measurement. The superscript l in $\bar{\Delta}_{i,j}^l$ remarks that the relative position vector is expressed in a local frame. By convention, the pose of the first node is assumed to be the reference frame in which we want to estimate all the other poses, i.e., $x_0 = [0 \ \ 0 \ \ 0]^\top$.

In [3] the authors showed that the relative orientation measurements can be made linear by adding a suitable multiple of 2π, i.e, $\langle \theta_j - \theta_i \rangle_{2\pi} = \theta_j - \theta_i + 2k_{i,j}\pi$, with $\theta_i, \theta_j \in \mathbb{R}$, and $k_{i,j} \in \mathbb{Z}$ ($k_{i,j}$ is called *regularization term*). In this paper we assume that the regularization terms have been correctly computed, according to [3], and we call $\bar{\delta}_{i,j}$ the regularized measurements, i.e., we define $\bar{\delta}_{i,j} = \check{\delta}_{i,j} - 2k_{i,j}\pi$. Then the measurement model becomes:

$$\begin{bmatrix} \bar{\Delta}_{i,j}^l \\ \bar{\delta}_{i,j} \end{bmatrix} = \begin{bmatrix} R_i^\top (p_j - p_i) \\ \theta_j - \theta_i \end{bmatrix} + \begin{bmatrix} \epsilon_{i,j}^{\Delta} \\ \epsilon_{i,j}^{\delta} \end{bmatrix}. \tag{2}$$

Therefore, the maximum likelihood estimate of nodes configuration X attains the minimum of the following cost function (see [3] and the references therein):

$$f(X) = \sum_{(i,j)\in\mathcal{E}} \begin{bmatrix} R_i^\top (p_j - p_i) - \bar{\Delta}_{i,j}^l \\ \theta_j - \theta_i - \bar{\delta}_{i,j} \end{bmatrix}^\top \Omega_{i,j} \begin{bmatrix} R_i^\top (p_j - p_i) - \bar{\Delta}_{i,j}^l \\ \theta_j - \theta_i - \bar{\delta}_{i,j} \end{bmatrix} \tag{3}$$

where $\Omega_{i,j} = P_{i,j}^{-1}$ is the *information matrix* of measurement (i,j). Pose graph optimization reduces to find a global minimum of the weighted sum of the residual errors, i.e., $X^* = \arg\min f(X)$. In the following we use $\mathbf{I}_n$, $\mathbf{0}_n$, and $\otimes$ to denote an identity matrix, a vector of all zeros, and the Kronecker product.

3 A Linear Approximation

In this section we present the first contribution of this manuscript: a linear approximation for problem (3) that relaxes the assumption of previous work [3]. In [3] it was assumed that $P_{i,j}$ (and then $\Omega_{i,j}$) has the following structure:

$$P_{i,j} = \begin{bmatrix} P_{i,j}^{\Delta} & \mathbf{0}_2 \\ \mathbf{0}_2^\top & P_{i,j}^{\delta} \end{bmatrix}. \tag{4}$$

Roughly speaking, this essentially requires that the relative position and relative orientation measurements (that together give the relative pose measurement) are uncorrelated. In order to present the subsequent derivation we need to rewrite the cost function (3) in a more compact form. For this purpose we define the unknown *nodes' position* $p = [p_1^\top \ \dots \ p_n^\top]^\top$ and the unknown *nodes' orientation* $\theta = [\theta_1 \ \dots \ \theta_n]^\top$; therefore the to-be-computed network configuration may be written as $x = [p^\top \quad \theta^\top]^\top$ (note that we have excluded from x the pose x_0 that was assumed to be known). Then, we number the available measurements from 1 to m and we stack the relative position measurements in the vector $\bar{\Delta}^l = [(\bar{\Delta}_1^l)^\top \ (\bar{\Delta}_2^l)^\top \ \dots \ (\bar{\Delta}_m^l)^\top]^\top$, and the relative orientation measurements in the vector $\bar{\delta} = [\bar{\delta}_1 \ \bar{\delta}_2 \ \dots \ \bar{\delta}_m]^\top$. Accordingly, we reorganize the measurement information matrices $\Omega_{i,j}, (i,j) \in \mathcal{E}$, into a large matrix

$$\Omega \doteq \begin{bmatrix} \Omega_{\Delta} & \Omega_{\Delta\delta} \\ \Omega_{\delta\Delta} & \Omega_{\delta} \end{bmatrix}, \tag{5}$$

such that Ω is the information matrix of the vector of measurements $[(\bar{\Delta}^l)^\top \ \ \bar{\delta}^\top]^\top$.

Then, the cost (3) can be written as:

$$f(x) = \begin{bmatrix} A_2^\top p - R\bar{\Delta}^l \\ A^\top\theta - \bar{\delta} \end{bmatrix}^\top \begin{bmatrix} R\Omega_{\Delta}R^\top & R\Omega_{\Delta\delta} \\ \Omega_{\delta\Delta}R^\top & \Omega_{\delta} \end{bmatrix} \begin{bmatrix} A_2^\top p - R\bar{\Delta}^l \\ A^\top\theta - \bar{\delta} \end{bmatrix} \tag{6}$$

where:

- A is the *reduced incidence matrix* of graph $\mathcal{G}$, see [3];
- $A_2 = A \otimes \mathbf{I}_2$ is an expanded version of A, see [1, 3];
- $R = R(\theta) \in \mathbb{R}^{2m,2m}$ is a block diagonal matrix, whose nonzero entries are in positions $(2k-1, 2k-1), (2k-1, 2k), (2k, 2k-1), (2k, 2k),\ k = 1, \dots, m$, such that, if the k-th measurement correspond to the relative pose between i and j, then the k-th diagonal block of R is a planar rotation matrix of an angle θ_i.

The residual errors in the cost function (6) are described by the following vector, whose entries represent the mismatch between the relative poses of a given configuration x and the actual relative measurements.

$$r(x) \doteq \begin{bmatrix} A_2^\top p - R\bar{\Delta}^l \\ A^\top\theta - \bar{\delta} \end{bmatrix} \tag{7}$$

Before presenting the proposed approach we anticipate the main intuition behind the algorithm. The cost function (6) is quite close to a quadratic function, since

the last part of the residual errors in (7) is linear, and the overall cost function (6) becomes quadratic as soon as the rotation matrix R is known. Therefore, the basic idea is (i) to obtain an estimate of nodes orientations θ exploiting the linear part of the residual errors in (7), (ii) to use the estimated orientation to compute an estimate of R, and (iii) to solve the overall problem in the optimization variable x. This basic intuition is the same motivating [3], although here the derivation is made more complex by the presence of the correlation between position measurements $\bar{\Delta}^l$ and orientation measurements $\bar{\delta}$.

We are now ready to present the proposed linear approximation for pose graph optimization, whose properties will be analyzed in Theorem 1.

Algorithm 1. *A linear approximation for the maximum likelihood pose graph configuration can be computed in three phases, given the relative measurements $\bar{\Delta}^l$ and $\bar{\delta}$, the corresponding information matrix Ω, and the graph incidence matrices A and A_2:*

1. *Solve the following linear system in the unknown $z \doteq [(\Delta^l)^\top \ \theta]^\top$:*

$$\Omega_z \, z = b_z \tag{8}$$

with:

$$Z = \begin{bmatrix} \mathbf{I}_{2m} & \mathbf{0}_{2m,n} \\ \mathbf{0}_{m,2m} & A^\top \end{bmatrix}, \quad \textit{and,} \quad \begin{matrix} b_z = Z^\top \Omega \left[(\bar{\Delta}^l)^\top \ \bar{\delta}^\top \right]^\top \\ \Omega_z = Z^\top \Omega Z \end{matrix} \tag{9}$$

Call the solution of the linear system $\hat{z} \doteq [(\hat{\Delta}^l)^\top \ \hat{\theta}]^\top$.

2. *Compute an estimate of the quantity $R\bar{\Delta}^l$ in (7) from $\hat{z}$, preserving the correlation with the estimate $\hat{\theta}$:*

$$\hat{y} = T(\hat{z}) \doteq \begin{bmatrix} \hat{R} & \mathbf{0}_{2m \times n} \\ \mathbf{0}_{2m \times n}^\top & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \hat{\Delta}^l \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} \tau_1(z) \\ \tau_2(z) \end{bmatrix}_{z=\hat{z}} \tag{10}$$

with $\hat{R} = R(\hat{\theta})$; compute the corresponding information matrix:

$$\Omega_y = (\hat{T} \Omega_z^{-1} \hat{T}^\top)^{-1} = (\hat{T}^{-1})^\top \Omega_z (\hat{T}^{-1}), \tag{11}$$

where $\hat{T}$ is the Jacobian of the transformation $T(\cdot)$:

$$\hat{T} \doteq \begin{bmatrix} \frac{\partial \tau_1}{\partial \Delta^l} & \frac{\partial \tau_1}{\partial \theta} \\ \frac{\partial \tau_2}{\partial \Delta^l} & \frac{\partial \tau_2}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \hat{R} & J \\ \mathbf{0}_{n \times 2m} & \mathbf{I}_n \end{bmatrix}. \tag{12}$$

3. *Solve the following linear system in the unknown $x = [p^\top \ \theta^\top]^\top$, given $\hat{y}$, see (10), and Ω_y, see (11):*

$$\Omega_x \, x = b_x \tag{13}$$

with:

$$B = \begin{bmatrix} A_2^\top & \mathbf{0}_{2m \times n} \\ \mathbf{0}_{n \times 2n} & \mathbf{I}_n \end{bmatrix}, \quad \textit{and,} \quad \begin{matrix} b_x = B^\top \Omega_y \hat{y} \\ \Omega_x = B^\top \Omega_y B \end{matrix} \tag{14}$$

The solution of the linear system (13) is the proposed linear approximation of the pose graph configuration: $x^* = [(p^*)^\top \ (\theta^*)^\top]^\top$. □

The effectiveness of the linear approximation computed using Algorithm 1 is assessed by the following result.

Theorem 1. *Given the inputs* $\{\bar{\Delta}^l, \bar{\delta}, \Omega, A, A_2\}$*, and assuming the information matrix* Ω *to be positive-definite, the following statements hold for the quantities computed in Algorithm 1:*

1. $\Omega_z, \Omega_y, \Omega_x$ *are full rank;*
2. *The combination of the three phases is equivalent to applying a Gauss-Newton step to the cost function* (6)*, starting from the initial guess* $\hat{x} = [\hat{p}^\top \ \ \hat{\theta}^\top]^\top$*, with* $\hat{\theta} = \left[A\Omega_\delta A^\top - A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}A^\top\right]^{-1} A\left(\Omega_\delta - \Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\right)\bar{\delta}$ *and* $\hat{p} = (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R}\Omega_\Delta\bar{\Delta}^l$.

Proof. See Appendix. □

The first claim assures the uniqueness of the outcome of the proposed algorithm (no indetermination in the solution of the linear systems). The second claim assures that the proposed approximation improves over an initial guess $\hat{x}$, applying a Gauss-Newton step. It is worth noticing that $\hat{\theta}$ can be rewritten as: $\hat{\theta} = \left[AP_\delta^{-1}A^\top\right]^{-1} AP_\delta^{-1}\bar{\delta}$, where $P_\delta^{-1} = \left(\Omega_\delta - \Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\right)$ is the marginal information matrix of the orientation measurements $\bar{\delta}$. Therefore, the initial guess $\hat{\theta}$ is the BLUE (Best Linear Unbiased Estimator) for θ, given the sole orientation measurements, see [3]; moreover, $\hat{p}$ is the optimal estimate of nodes' positions, under the assumption that the actual orientations of the robot coincide with $\hat{\theta}$ [3]. The practical advantage of the algorithm is that $\hat{x}$ is already quite close to the optimal solution in practice, then the approximation is accurate in common problem instances. Moreover, the vector $\hat{p}$ is not computed explicitly by the approach, saving computation time.

4 Experimental Analysis

In this section we present the results of an extensive numerical evaluation on optimization-based SLAM. We compare the methodology proposed in this paper (Algorithm 1) with several state-of-the-art optimization approaches, namely a Gauss-Newton method [12], TORO [8], g^{2}o [11], and the linear approximation proposed in [3]. The Gauss-Newton approach is a standard implementation of the Gauss-Newton method for solving nonlinear least squares problems [13]. The halting condition for this approach is based on the norm of the local correction. Roughly speaking, if in two consecutive iterations the change in the configuration is smaller than a threshold the algorithm stops. In our tests the threshold on the norm of the local correction was set to 0.1. The results from TORO and g^{2}o are obtained using the C++ code available online [15]. For the tests we used default settings for both approaches. Our implementations of the linear approximation [3] and of Algorithm 1 are available online [4]. Also the implementation of the Gauss-Newton approach we used in the test campaign was released online [4].

The compared approaches are tested on publicly available datasets: Freiburg Indoor Building 079 (FR079), MIT CSAIL Building (CSAIL), Intel Research Lab (INTEL), Manhattan World M3500 (M3500), Manhattan World M10000 (M10000). The relative pose measurements of the datasets {FR079, CSAIL, M3500, M10000} are available online [10], while the measurements of the dataset INTEL were obtained through a scan matching procedure, from the raw sensor data, available at [10]. The INTEL dataset is the same studied in [3]. The relations available online [10] only describe the relative pose measurements, while we are interested to test the behavior of the approaches for different measurement covariance matrices. In particular, for each dataset we consider three variants, each one corresponding to a different choice of the covariance matrix. The first variant (e.g., FR079-I) uses identity matrices as measurement covariances, i.e., the noise of the relative pose measurement between node i and node j is $\epsilon_i^j \sim \mathcal{N}(\mathbf{0}_3, \mathbf{I}_3)$. The second variant (e.g., FR079-P_s) uses a structured covariance matrix, as in eq. (4). The third variant (e.g., FR079-P_f) uses full covariance matrices obtained as follows. According to the standard *odometry model* [18], we parametrize the relative pose between node i and j as a rotation γ_r^1, followed by a translation γ_t^1, and by a second rotation γ_r^2, see Section 5.4 in [18]. Then, fixing the uncertainty in the parameters, we can define the corresponding covariance matrix for the relative pose measurement. For our numerical experiments we set the standard deviations of γ_r^1, γ_t^1, γ_r^2, to 0.05 rad, 0.05 m, and 0.01 rad, respectively. For the sake of repeatability and for stimulating further comparisons with related approaches the datasets considered in this paper were released online [4].

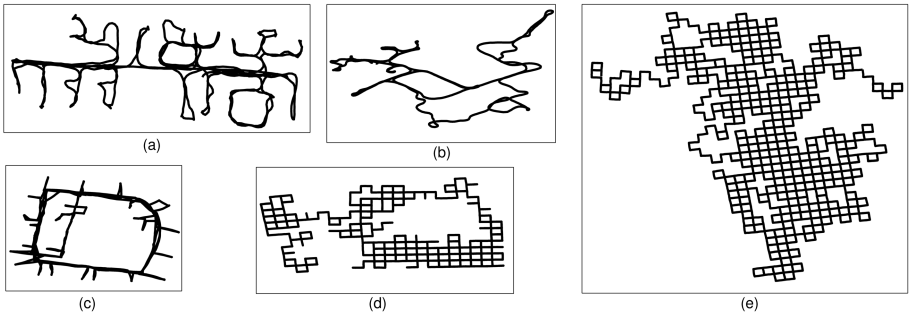

Fig. 1. Estimated trajectory for each of the considered datasets: (a) FR079, (b) CSAIL, (c) INTEL, (d) M3500, (e) M10000

Accuracy. In Figure 1 we show some qualitative results for the proposed approach on the considered datasets (for simplicity we only show the variant with the identity matrix as measurement covariance). For a quantitative evaluation of the accuracy of the approaches, we recorded the optimal value of the cost function (6), attained by each of the compared techniques. Since each approach is required to minimize the cost (6), the best solution is the one attaining the smallest value of the objective function. The results for the compared approaches

and for each dataset are reported in Table 1. The Gauss-Newton method and g^{2}o attain the same solution (which is also the smallest observed cost function) in most cases. This comes at no surprise since they both solve the original optimization problem without any approximation involved. Only in two scenarios g^{2}o performs worse than the Gauss-Newton method: in the INTEL dataset (in which the initial guess is particularly bad), and in the M10000 dataset (which contains a large number of nodes). The difference is explained as follows: g^{2}o applies a fixed number of iterations, then in the two mentioned cases, the iterations are not sufficient to reach the optimal value. TORO shows the worst performance in all tests, due to the involved approximations, see [8]. The linear approximations of [3] (structured covariance) and the one proposed in this paper (unstructured covariance) produce intermediate results. They are practically optimal in the first variants (I) of each scenario (using identity matrices as covariances), and they are close to the Gauss-Newton solution in the second variants (P_s). Only in the scenario INTEL the cost function was remarkably larger than the Gauss-Newton approach, although being lower than TORO. It is worth noticing that in the first and in the second variants of each dataset, the linear approximation of [3] and the approach proposed in this paper attain the same objective. This is due to the fact that the approach proposed in this paper reduces to the linear approximation of [3] when the measurements covariance matrices of the input data are structured (as it happens in the variants I and P_s). The two approaches, instead, differ when measurement covariance is unstructured, as in the P_f variant reported in Table 1. In this case the linear approximation [3] simply neglects the correlation terms while the proposed approach can deal with the full covariance case. The numerical results show that in the third variant (P_f) of the datasets {INTEL, M10000} the proposed approximation remarkably improves the attained objective value. In the remaining datasets, the difference between the linear approximations is small. We conclude this paragraph by noticing that the proposed approximation and the approach in [3] are more accurate than g^{2}o on the large-scale dataset M10000.

Efficiency. The efficiency of the compared approaches is connected with the computational effort that each method requires for producing the estimate of node configuration. The average CPU time required by the compared approaches for each of the dataset is reported in Table 1. The reported statistics are averaged over 10 runs. The tests are conducted on a standard laptop, with an Intel Core i7 3.4 GHZ and 8 GB of RAM. The CPU times required by TORO and g^{2}o are the ones returned by the code available online. The Gauss-Newton method, the linear approximation [3], and the approach proposed in this paper are implemented in C++ and use the *CSparse* library [5]. Moreover, for the two linear approximations, the CPU time includes the computation of the *regularization terms* [3].

From the table it is possible to see that the Gauss-Newton method, although being very accurate, quickly becomes unsustainable for large datasets. TORO is slightly faster, but still remains not competitive w.r.t. the other techniques. g^{2}o is highly optimized and allows a remarkable speed-up w.r.t. the Gauss-Newton method. Table 1 highlights that the linear approximation [3] and the linear

Table 1. Objective function values and average computation time (in seconds) for the compared approaches

			Linear Approximation (structured covariance)	Linear Approximation (unstructured covariance)	Gauss-Newton	TORO	g^{2}o
FR079	I	Objective	7.20E-02	7.20E-02	7.20E-02	8.60E-02	7.19E-02
		Time (s)	5.80E-03	8.15E-03	1.99E-01	3.19E-01	1.05E-02
	Ps	Objective	3.94E+01	3.94E+01	3.88E+01	4.74E+02	3.89E+01
		Time (s)	5.76E-03	7.87E-03	2.00E-01	3.39E-01	1.07E-02
	Pf	Objective	2.76E+02	2.90E+02	1.47E+02	8.99E+03	1.47E+02
		Time (s)	5.81E-03	8.16E-03	2.00E-01	3.04E-01	1.06E-02
CSAIL	I	Objective	1.07E-01	1.07E-01	1.07E-01	1.18E-01	1.07E-01
		Time (s)	5.72E-03	7.55E-03	2.65E-01	2.89E-01	1.01E-02
	Ps	Objective	4.06E+01	4.06E+01	4.06E+01	2.41E+03	4.06E+01
		Time (s)	5.53E-03	7.46E-03	2.01E-01	2.90E-01	1.01E-02
	Pf	Objective	2.45E+02	2.33E+02	1.57E+02	4.57E+04	1.57E+02
		Time (s)	5.59E-03	7.50E-03	2.66E-01	2.88E-01	1.03E-02
INTEL	I	Objective	8.07E-01	8.07E-01	7.89E-01	1.17	7.89E-01
		Time (s)	7.10E-03	9.49E-03	5.87E-01	4.15E-01	1.32E-02
	Ps	Objective	1.45E+04	1.45E+04	2.15E+02	1.03E+05	2.15E+02
		Time (s)	7.01E-03	9.49E-03	4.90E-01	3.89E-01	1.31E-02
	Pf	Objective	1.51E+06	1.07E+05	3.95E+02	2.53E+07	1.08E+03
		Time (s)	6.98E-03	9.47E-03	5.91E-01	4.01E-01	1.31E-02
M3500	I	Objective	3.03	3.03	3.02	5.42	3.02
		Time (s)	3.26E-02	4.04E-02	5.81	1.57	7.07E-02
	Ps	Objective	3.73E+03	3.73E+03	3.55E+03	2.18E+06	3.55E+03
		Time (s)	3.25E-02	4.03E-02	4.84	1.61	7.06E-02
	Pf	Objective	1.15E+04	6.81E+03	2.09E+03	5.78E+08	2.09E+03
		Time (s)	3.26E-02	4.05E-02	5.82	1.61	7.14E-02
M10000	I	Objective	3.03E+02	3.03E+02	3.03E+02	3.29E+02	3.03E+02
		Time (s)	3.55E-01	4.86E-01	2.21E+02	1.73E+01	6.93E-01
	Ps	Objective	1.99E+05	1.99E+05	1.98E+05	7.65E+06	2.28E+05
		Time (s)	3.57E-01	4.89E-01	2.21E+02	1.83E+01	6.96E-01
	Pf	Objective	9.00E+05	9.61E+05	6.79E+05	2.07E+08	1.90E+07
		Time (s)	3.55E-01	4.86E-01	4.11E+02	1.77E+01	6.91E-01

approximation proposed in this paper outperform all state-of-the-art techniques in terms of computational time. In particular, they assure a reduction of the computational time of $30-50\%$ w.r.t. to g^{2}o and improve the computational time of orders of magnitude w.r.t. the other state-of-the-art techniques. We conclude this section observing that the proposed approach is able to compute an estimate of the configuration of a pose graph with 10000 nodes and 64311 edges in less than 0.5 seconds.

Finally a video showing an experimental test in which the edges of the graph are obtained online using a scan matching algorithm is available online [4].

5 Conclusion

The contribution of this article is twofold: a linear approximation for optimization-based SLAM and an extensive evaluation of the performance of state-of-the-art approaches on benchmarking datasets. The first contribution includes the presentation of an algorithm for estimating an approximation of pose graph configuration. The algorithm is an extended version of the approach proposed in [3] and can deal with the case of generic unstructured measurement covariance. The second contribution includes an experimental analysis of pose graph optimization approaches in terms of accuracy, efficiency, and scalability. As a results we demonstrate that the accuracy of the proposed linear approximation is comparable with the one of state-of-the-art techniques, although it requires a fraction of their computational effort.

Appendix

In this appendix we report the proof of Theorem 1. We omit for brevity the proof of the first claim, which is a straightforward extension of the results reported in [3]. We instead prove the second claim by direct calculation. We need to demonstrate that the outcome of the proposed algorithm is equivalent to a Gauss-Newton step from the initial guess $\hat{x} = [\hat{p}^\top \quad \hat{\theta}^\top]^\top$. The structure of the proof is the following: (i) we compute by direct calculation the solution of the proposed approach x^*, (ii) we compute the estimate x^{GN}, obtained from a Gauss-Newton step with initial guess $\hat{x}$, (iii) we show that $x^* = x^{GN}$. We start by computing $\Omega_z = Z^\top \Omega Z$:

$$\Omega_z = \begin{bmatrix} \Omega_\Delta & \Omega_{\Delta\delta} A^\top \\ A\Omega_{\delta\Delta} & A\Omega_\delta A^\top \end{bmatrix}$$

We can use blockwise inversion rule to compute the inverse of Ω_z. Notice that the explicit inverse needs not be computed in practice, since computationally effective methods can be used to solve the sparse linear system (8). For the sake of the proof, we instead evaluate:

$$P_z \doteq \Omega_z^{-1} = \begin{bmatrix} P_{z_{11}} & P_{z_{12}} \\ P_{z_{21}} & P_{z_{22}} \end{bmatrix}$$

with:

$$P_{z_{22}} = \left[A\Omega_\delta A^\top - A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}A^\top\right]^{-1},$$
$$P_{z_{11}} = \Omega_\Delta^{-1} + \Omega_\Delta^{-1}\Omega_{\Delta\delta}A^\top P_{z_{22}} A\Omega_{\delta\Delta}\Omega_\Delta^{-1}, \quad P_{z_{12}} = P_{z_{21}}^\top = -\Omega_\Delta^{-1}\Omega_{\Delta\delta}A^\top P_{z_{22}}.$$

Then we can compute $\hat{z} = \Omega_z^{-1} b_z$:

$$\hat{z} = \begin{bmatrix} \bar{\Delta}^l + \Omega_\Delta^{-1}\Omega_{\Delta\delta}\left[\bar{\delta} - A^\top P_{z_{22}} A\left(\Omega_\delta - \Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\right)\bar{\delta}\right] \\ P_{z_{22}} A\left(\Omega_\delta - \Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\right)\bar{\delta} \end{bmatrix}$$

If we call $\hat{\theta} = P_{z_{22}} A \left(\Omega_\delta - \Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\right)\bar{\delta}$, the vector $\hat{z}$ can be written in compact form as:

$$\hat{z} = \begin{bmatrix} \bar{\Delta}^l + \Omega_\Delta^{-1}\Omega_{\Delta\delta}\left(\bar{\delta} - A^\top\hat{\theta}\right) \\ \hat{\theta} \end{bmatrix} \tag{15}$$

We can then compute $\hat{y}$ and Ω_y according to (10) and (11):

$$\hat{y} = \begin{bmatrix} \hat{R}\bar{\Delta}^l + \hat{R}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\left(\bar{\delta} - A^\top\hat{\theta}\right) \\ \hat{\theta} \end{bmatrix}, \quad \Omega_y = (\hat{T}^{-1})^\top \Omega_z (\hat{T}^{-1}) \doteq \begin{bmatrix} \Omega_{y_{11}} & \Omega_{y_{12}} \\ \Omega_{y_{12}} & \Omega_{y_{22}} \end{bmatrix}$$

with:

$$\begin{aligned}
\hat{T}^{-1} &= \begin{bmatrix} \hat{R} & J \\ \mathbf{0}_{n\times 2m} & \mathbf{I}_n \end{bmatrix}^{-1} = \begin{bmatrix} \hat{R}^\top & -\hat{R}^\top J \\ \mathbf{0}_{n\times 2m} & \mathbf{I}_n \end{bmatrix}, \quad \Omega_{y_{11}} = \hat{R}\Omega_\Delta\hat{R}^\top \\
\Omega_{y_{12}} &= \Omega_{y_{12}}^\top = -\hat{R}\Omega_\Delta\hat{R}^\top J + \hat{R}\Omega_{\Delta\delta}A^\top \\
\Omega_{y_{22}} &= A\Omega_\delta A^\top + J^\top\hat{R}\Omega_\Delta\hat{R}^\top J - A\Omega_{\delta\Delta}\hat{R}^\top J - J^\top\hat{R}\Omega_{\Delta\delta}A^\top.
\end{aligned}$$

Now we compute Ω_x, its inverse Ω_x^{-1}, and b_x, from which it is easy to derive x^*. According to (3), Ω_x can be written explicitly as:

$$\Omega_x \doteq B^\top\Omega_y B = \begin{bmatrix} \Omega_{x_{11}} & \Omega_{x_{12}} \\ \Omega_{x_{21}} & \Omega_{x_{22}} \end{bmatrix}$$

with:

$$\begin{aligned}
\Omega_{x_{11}} &= A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top \\
\Omega_{x_{12}} &= \Omega_{x_{21}}^\top = -A_2\hat{R}\Omega_\Delta\hat{R}^\top J + A_2\hat{R}\Omega_{\Delta\delta}A^\top \\
\Omega_{x_{22}} &= A\Omega_\delta A^\top + J^\top\hat{R}\Omega_\Delta\hat{R}^\top J - A\Omega_{\delta\Delta}\hat{R}^\top J - J^\top\hat{R}\Omega_{\Delta\delta}A^\top.
\end{aligned}$$

After long and tedious calculations we obtain Ω_x^{-1} using standard blockwise-inversion rules:

$$P_x \doteq \Omega_x^{-1} = \begin{bmatrix} P_{x_{11}} & P_{x_{12}} \\ P_{x_{21}} & P_{x_{22}} \end{bmatrix}$$

with:

$$\begin{aligned}
&P_{x_{22}} = \left[A\Omega_\delta A^\top + J^\top\hat{R}\Omega_\Delta\hat{R}^\top J - J^\top\hat{R}\Omega_{\Delta\delta}A^\top - A\Omega_{\delta\Delta}\hat{R}^\top J + \right. \\
&\left. -(A\Omega_{\delta\Delta}\hat{R}^\top A_2^\top - J^\top\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}(A_2\hat{R}\Omega_{\Delta\delta}A^\top - A_2\hat{R}\Omega_\Delta\hat{R}^\top J)\right]^{-1} \\
&P_{x_{12}} = P_{x_{21}}^\top = -(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}(A_2\hat{R}\Omega_{\Delta\delta}A^\top - A_2\hat{R}\Omega_\Delta\hat{R}^\top J)P_{x_{22}} \\
&P_{x_{11}} = (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1} + (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}(A_2\hat{R}\Omega_{\Delta\delta}A^\top - A_2\hat{R}\Omega_\Delta\hat{R}^\top J)\times \\
&P_{x_{22}}(A\Omega_{\delta\Delta}\hat{R}^\top A_2^\top - J^\top\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}.
\end{aligned} \tag{16}$$

From matrix-vector multiplication we also compute $b_x \doteq B^\top\Omega_y\hat{y} = [b_{x_1}^\top \;\; b_{x_2}^\top]^\top$, with:

$$\begin{aligned}
b_{x_1} &= A_2\hat{R}\left(\Omega_\Delta\bar{\Delta}^l - \Omega_\Delta\hat{R}^\top J\hat{\theta} + \Omega_{\Delta\delta}\bar{\delta}\right) \\
b_{x_2} &= A\Omega_\delta A^\top\hat{\theta} + J^\top\hat{R}\Omega_\Delta\hat{R}^\top J\hat{\theta} - A\Omega_{\delta\Delta}\hat{R}^\top J\hat{\theta} + A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\hat{\delta} + \\
&\quad -J^\top\hat{R}\Omega_{\Delta\delta}\bar{\delta} - A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}A^\top\hat{\theta} + A\Omega_{\delta\Delta}\bar{\Delta}^l - J^\top\hat{R}\Omega_\Delta\bar{\Delta}^l.
\end{aligned}$$

Finally we can calculate $x^* = P_x b_x \doteq [(p^*)^\top \ (\theta^*)^\top]^\top$, with:

$$\begin{aligned}\theta^* &= \hat{\theta} + P_{x_{22}}\big[(J^\top \hat{R}\Omega_\Delta \hat{R}^\top A_2^\top)(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R} - (A\Omega_{\delta\Delta}\hat{R}^\top A_2^\top)\times \\ &\quad (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R} - J^\top\hat{R} + A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\big]\big[\Omega_\Delta\bar{\Delta}^l + \Omega_{\Delta\delta}(\bar{\delta} - A^\top\hat{\theta})\big], \\ p^* &= \hat{p} + (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R}\left[\Omega_{\Delta\delta}(\bar{\delta} - A^\top\theta^*) + \Omega_\Delta\hat{R}^\top J(\theta^* - \hat{\theta})\right],\end{aligned}$$

where $\hat{\theta} = P_{z_{22}} A\left(\Omega_\delta - \Omega_{\delta\Delta}\Omega_\Delta^{-1}\Omega_{\Delta\delta}\right)\bar{\delta}$ and $\hat{p} = (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R}\Omega_\Delta\bar{\Delta}^l$. After obtaining x^* we have to compute the outcome of the Gauss-Newton step from $\hat{x} \doteq [\hat{p}^\top \ \hat{\theta}^\top]^\top$, since we claim that the result is the same. According to the standard Gauss-Newton approach, a single step from the guess $\hat{x}$ produces the estimate $x^{GN} = \hat{x} + \tilde{x}$, where $\tilde{x} \doteq [\tilde{p}^\top \ \tilde{\theta}^\top]^\top$ is the minimum of the quadratic cost obtained by linearizing the residual errors in (6) around $\hat{x}$. Let us start by linearizing the residual errors in the cost function around $\hat{x}$:

$$r(\hat{x}+\tilde{x}) \approx \begin{bmatrix} A_2^\top\hat{p} + A_2^\top\tilde{p} - \hat{R}\bar{\Delta}^l - J\tilde{\theta} \\ A^\top\hat{\theta} + A^\top\tilde{\theta} - \bar{\delta}\end{bmatrix} \doteq \tilde{r}(\tilde{x}) \tag{17}$$

Considering the linearized residue and evaluating the covariance matrix in $\theta = \hat{\theta}$ the cost function (6) becomes quadratic:

$$f(\tilde{x}) \approx \tilde{r}(\tilde{x})^\top \begin{bmatrix} \hat{R}\Omega_\Delta\hat{R}^\top & \hat{R}\Omega_{\Delta\delta} \\ \Omega_{\delta\Delta}\hat{R}^\top & \Omega_\delta\end{bmatrix}\tilde{r}(\tilde{x}) \doteq \tilde{f}(\tilde{x}) \tag{18}$$

The global minimum of the previous cost function can be computed by taking the gradient of the $\tilde{f}$ with respect to $\tilde{x} = [\tilde{p}^\top \ \tilde{\theta}^\top]^\top$ and imposing it to be zero. Let us compute the gradient with respect to the variable $\tilde{p}$ and $\tilde{\theta}$:

$$\begin{aligned}\nabla_{\tilde{p}}(\tilde{f}) &= 2A_2\hat{R}\Omega_\Delta\hat{R}^\top(A_2^\top\hat{p} - \hat{R}\bar{\Delta}^l + A_2^\top\tilde{p} - J\tilde{\theta}) + 2A_2\hat{R}\Omega_{\Delta\delta}(A^\top\hat{\theta} + A^\top\tilde{\theta} - \bar{\delta}) \\ \nabla_{\tilde{\theta}}(\tilde{f}) &= -2J^\top\hat{R}\Omega_\Delta\hat{R}^\top(A_2^\top\hat{p} - \hat{R}\bar{\Delta}^l + A_2^\top\tilde{p} - J\tilde{\theta}) - 2J^\top\hat{R}\Omega_{\Delta\delta}(A^\top\hat{\theta} + A^\top\tilde{\theta} - \bar{\delta}) + \\ &\quad + 2A\Omega_{\delta\Delta}\hat{R}^\top(A_2^\top\hat{p} - \hat{R}\bar{\Delta}^l + A_2^\top\tilde{p} - J\tilde{\theta}) + 2A\Omega_\delta(A^\top\hat{\theta} + A^\top\tilde{\theta} - \bar{\delta})\end{aligned}$$

The global minimum has to satisfy the following linear system of equations:

$$\begin{cases}\nabla_{\tilde{p}}(\tilde{f}) = \mathbf{0}_{2n} \\ \nabla_{\tilde{\theta}}(\tilde{f}) = \mathbf{0}_n\end{cases} \tag{19}$$

It is possible to explicit the unknown $\tilde{p}$ from the first equation in (19), writing it in function of $\tilde{\theta}$:

$$\tilde{p} = (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R}\left[\Omega_{\Delta\delta}(\bar{\delta} - A^\top\hat{\theta} - A^\top\tilde{\theta}) + \Omega_\Delta\hat{R}^\top J\tilde{\theta}\right].$$

Now we can substitute $\tilde{p}$ in the second equation of (19), obtaining a linear equation containing only $\tilde{\theta}$. Solving such equation we obtain:

$$\begin{aligned}\tilde{\theta} = P_{x_{22}}\big[&(J^\top\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R} + \\ &-(A\Omega_{\delta\Delta}\hat{R}^\top A_2^\top)(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R} + \\ &-J^\top\hat{R} + A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\big]\big[\Omega_\Delta\bar{\Delta}^l + \Omega_{\Delta\delta}(\bar{\delta} - A^\top\hat{\theta})\big],\end{aligned}$$

where $P_{x_{22}}$ is defined as in (16). We can finally compute x^{GN}:

$$x^{\mathrm{GN}} = \hat{x} + \tilde{x} = \begin{bmatrix}\hat{p} + \tilde{p} \\ \hat{\theta} + \tilde{\theta}\end{bmatrix} = \begin{bmatrix} p^{\mathrm{GN}} \\ \theta^{\mathrm{GN}}\end{bmatrix}$$

with:

$$\theta^{\mathrm{GN}} = \hat{\theta} + P_{x_{22}}\big[(J^\top \hat{R}\Omega_\Delta \hat{R}^\top A_2^\top)(A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R} - (A\Omega_{\delta\Delta}\hat{R}^\top A_2^\top)\times (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R} - J^\top\hat{R} + A\Omega_{\delta\Delta}\Omega_\Delta^{-1}\big]\big[\Omega_\Delta\bar{\Delta}^l + \Omega_{\Delta\delta}(\bar{\delta} - A^\top\hat{\theta})\big],$$
$$p^{\mathrm{GN}} = \hat{p} + (A_2\hat{R}\Omega_\Delta\hat{R}^\top A_2^\top)^{-1}A_2\hat{R}\left[\Omega_{\Delta\delta}(\bar{\delta} - A^\top\theta^*) + \Omega_\Delta\hat{R}^\top J(\theta^* - \hat{\theta})\right].$$

By inspection, it is easy to verify that $p^{\mathrm{GN}} = p^*$ and $\theta^{\mathrm{GN}} = \theta^*$, hence proving the second claim of Theorem 1.

References

[1] Barooah, P., Hespanha, J.P.: Estimation on graphs from relative measurements. IEEE Control Systems Magazine 27(4), 57–74 (2007)

[2] Carlone, L., Aragues, R., Castellanos, J.A., Bona, B.: A first-order solution to simultaneous localization and mapping with graphical models. In: Proc. of the IEEE lnternational Conf. on Robotics and Automation (2011)

[3] Carlone, L., Aragues, R., Castellanos, J.A., Bona, B.: A linear approximation for graph-based simultaneous localization and mapping. In: Proc. of Robotics: Science and Systems (2011)

[4] Carlone, L., Rosa, S., Yin, J.: Robotics research group: Resources – graph optimization with unstructured covariance (2012), `www.polito.it/LabRob`

[5] Davis, T.A.: Direct Methods for Sparse Linear Systems. Fundamentals of Algorithms, vol. 2. Society for Industrial and Applied Mathematics, Philadelphia (2006) ISBN 0898716136

[6] Dellaert, F., Carlson, J., Ila, V., Ni, K., Thorpe, C.: Subgraph-preconditioned conjugate gradients for large scale SLAM. In: Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (2010)

[7] Frese, U., Larsson, P., Duckett, T.: A multilevel relaxation algorithm for simultaneous localization and mapping. IEEE Trans. on Robotics 21(2), 196–207 (2005)

[8] Grisetti, G., Stachniss, C., Burgard, W.: Non-linear constraint network optimization for efficient map learning. IEEE Trans. on Intelligent Transportation Systems 10(3), 428–439 (2009)

[9] Konolige, K.: Large-scale map-making. In: Proc. of the AAAI National Conf. on Artificial Intelligence (2004)

[10] Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., Kleiner, A.: Slam benchmarking webpage (2009), `http://ais.informatik.uni-freiburg.de/slamevaluation`

[11] Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2o: A general framework for graph optimization. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 3607–3613 (May 2011), doi:10.1109/ICRA.2011.5979949

[12] Lu, F., Milios, E.: Globally consistent range scan alignment for environment mapping. Autonomous Robots 4, 333–349 (1997)

[13] Nocedal, J., Wright, S.J.: Numerical Optimization. Springer (2006)

[14] Olson, E., Leonard, J.J., Teller, S.: Fast iterative optimization of pose graphs with poor initial estimates. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 2262–2269 (2006)

[15] Stachniss, C., Frese, U., Grisetti, G.: Open SLAM webpage (2007), http://openslam.org/
[16] Sunderhauf, N., Protzel, N.: Towards a robust back-end for pose graph slam. In: Proc. of IEEE International Conference on Robotics and Automation, ICRA (2012)
[17] Thrun, S., Montemerlo, M.: The GraphSLAM algorithm with applications to large-scale mapping of urban structures. Int. J. Robot. Res. 25, 403–429 (2006)
[18] Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. MIT Press (2005)