

# iSPCG: Incremental Subgraph-Preconditioned Conjugate Gradient Method for Online SLAM with Many Loop-Closures

Yong-Dian Jian and Frank Dellaert

**Abstract**— We propose a novel method to solve online SLAM problems with many loop-closures on the basis of two state-of-the-art SLAM methods, iSAM and SPCG. We first use iSAM to solve a sparse sub-problem to obtain an approximate solution. When the error grows larger than a threshold or the optimal solution is requested, we use subgraph-preconditioned conjugate gradient method to solve the original problem where the subgraph preconditioner and initial estimate are provided by iSAM. Finally we use the optimal solution from SPCG to regularize iSAM in the next steps. The proposed method is consistent, efficient and can find the optimal solution. We apply this method to solve large simulated and real SLAM problems, and obtain promising results.

## I. INTRODUCTION

Simultaneous localization and mapping (SLAM) refers to the problem of localizing a robot in an unknown environment while simultaneously building a consistent map. Being able to efficiently solve SLAM in large and complex environments are important for many tasks in autonomous mobile robotics such as navigation and planning [1], [2]. Here we are interested in solving online SLAM problems with many loop-closures, which are common when the robot has long-range sensors, e.g., cameras, and explores in large open space.

Modern SLAM methods often formulate SLAM as a graph-based optimization problem [3], [4], [5]. The state-of-the-art online SLAM methods stem from incremental smoothing and mapping (iSAM) [6], and hierarchical optimization (HOG-Man) [7]. Yet these methods do not scale well when there are many loop-closures because they aim to incrementally or hierarchically factorize the information matrix which is expensive for these problems.

Many techniques have been proposed to efficiently solve SLAM problems with many loop-closures. Methods based on graph sparsification aim to reduce the problem size by discarding redundant edges, marginalizing redundant vertices, or using convex optimization techniques, but they lead to either inconsistent estimate or high computational complexity [8], [9], [10], [11], [12], [13], [14]. Iterative methods have been used to solve the SLAM problems with many loop-closures, but they either assume all measurements are available in advance or require good initialization to obtain good results [15], [16], [17], [18], [19], [20], [21].

In this paper, we propose a new method, incremental subgraph-preconditioned conjugate (iSPCG) method, to ef-

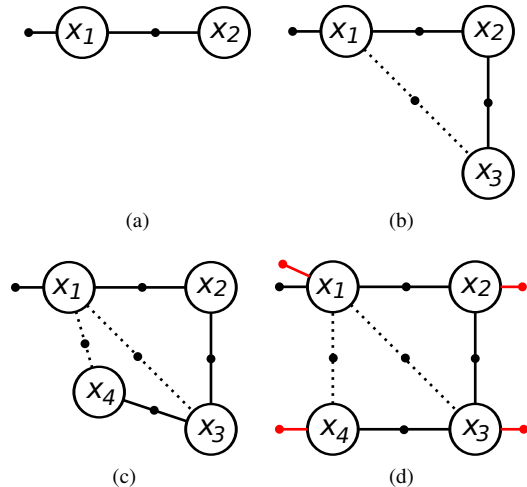


Fig. 1: Illustration of the iSPCG method on a simple graph. The solid factors belong to the subgraph while the dashed factors correspond to the remaining part. (a) Initially the graph is still sparse. Hence all factors belong to the subgraph, and iSAM can solve it very efficiently. (b) There is one loop-closure constraint, but leaving it out of the subgraph does not introduce significant error. (c) There are more loop-closures, but this time leaving them out of the subgraph leads to unsatisfactory results. Hence SPCG is invoked to optimize the entire graph. (d). The solution obtained from SPCG are used to regularized iSAM in the next iterations.

ficiently solve online SLAM problems with many loop-closures. The main idea is to use iSAM to incrementally solve a sparse subgraph to obtain an approximate solution. When the error grows larger than a threshold or the optimal solution is requested, the subgraph-preconditioned conjugate gradient method (SPCG) [20], [21] is applied to solve the entire graph to obtain a locally optimal solution. We note that the subgraph preconditioner and the initial estimate for SPCG are provided by iSAM. Then we use the optimal solution from SPCG to regularize iSAM's estimates in the following steps. Unlike previous work which might lead to information loss or inconsistent estimate [9], [10], [13], [14], we show that iSPCG can be consistent, efficient and optimal.

This paper has the following contributions: (1) We propose iSPCG to efficiently solve online SLAM problems with many loop-closures by combining two state-of-the-art SLAM methods. (2) We provide a theoretical analysis of iSPCG and show that it can provide consistent and optimal estimate. (3) We apply iSPCG to solve large-scale simulated and real SLAM problems and obtain promising results.

The authors are affiliated with the Georgia Institute of Technology, USA {yujian@gatech.edu, dellaert@cc.gatech.edu}, and they acknowledge the support from the National Science Foundation Award 11115678 "RI: Small: Ultra-Sparsifiers for Fast and Scalable Mapping and 3D Reconstruction on Mobile Robots".

## II. SLAM

### A. Formulation

Here we review SLAM formulation to facilitate the exposition. We define  $\theta = \{\theta_i\}_{i=1}^n$  as the state variables (e.g., robot poses), and  $\mathbf{Z} = \{z_j\}_{j=1}^m$  as the measurements (e.g., odometry and loop-closure). The goal is to obtain the maximum a posteriori (MAP) estimation

$$\theta_{\text{MAP}}(\mathbf{Z}) = \underset{\theta}{\operatorname{argmax}} P(\theta)P(\mathbf{Z} | \theta). \quad (1)$$

Assuming the variables are independent, and the measurements are conditionally independent, we can factorize the right-hand side of (1) into

$$P(\theta)P(\mathbf{Z}|\theta) \propto \prod_{i=1}^n P(\theta_i) \prod_{j=1}^m P(z_j | \theta_j) \quad (2)$$

where  $\theta_j$  denotes the variables of the  $j$ th measurement.

The SLAM problem can also be formulated with the factor graph representation [22] where each vertex denotes a state variable, and each factor (edge) is represented by the squared error term associated with a probability density function in (2). More specifically, we assume prior and measurement models are Gaussian, defined by

$$P(\theta_i) \propto \exp(-\|g_i(\theta_i)\|_{\Gamma_i}^2) \quad (3)$$

$$P(z_j | \theta_j) \propto \exp(-\|h_j(\theta_j)\|_{\Psi_j}^2). \quad (4)$$

where  $g_i(\cdot)$  denotes the prior model over the  $i$ th variable and  $h_j(\cdot)$  denotes the model of the  $j$ th measurement. In both models, we assume zero-mean and normally distributed noise with covariance matrices  $\Gamma_i$  and  $\Psi_j$  respectively. Here  $\|e\|_{\Sigma} = \sqrt{e^T \Sigma^{-1} e}$  denotes the Mahalanobis distance. By substituting the probability densities in (2) with the functions in (3) and (4), and taking negative logarithm, we obtain the following factor graph representation for the SLAM problem

$$\theta_{\text{MAP}}(\mathbf{Z}) = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \|g_i(\theta_i)\|_{\Gamma_i}^2 + \sum_{j=1}^m \|h_j(\theta_j)\|_{\Psi_j}^2 \quad (5)$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{k=1}^{m+n} \|e_k(\theta_k)\|_{\Sigma_k}^2 \quad (6)$$

where  $e_k(\cdot)$  is a function  $\theta_k$  with covariance matrix  $\Sigma_k$ .

### B. Nonlinear Optimization

We show how to solve (6) with nonlinear optimization [4], [5]. The function in (6) is generally non-convex and has no closed-form expression to compute the global optimum, but assuming we have some initial estimates of the variables, we can find a local optimum by using any nonlinear least-squares optimization algorithm (e.g., the Gauss-Newton or the Levenberg-Marquardt algorithm) [23]. The key is to apply first-order Taylor expansion to linearize the function

$$e_k(\theta_k) \approx e_k(\theta_k^0) + \mathbf{J}_k \Delta \theta_k \quad (7)$$

where  $\mathbf{J}_k$  is the Jacobian matrix of  $e_k(\cdot)$  with respect to  $\theta_k$  at the linearization point  $\theta_k^0$ :

$$\mathbf{J}_k = \left. \frac{\partial e_k(\theta_k)}{\partial \theta_k} \right|_{\theta_k^0}. \quad (8)$$

If we set (7) to zero, then we obtain  $\mathbf{J}_k \Delta \theta_k = -e_k(\theta_k^0)$  which is linear in  $\Delta \theta_k$ . Repeating this procedure for all of the  $e_k(\cdot)$  functions, we can derive a linear system

$$\mathbf{A} \Delta \theta = \mathbf{b} \quad (9)$$

where  $\mathbf{A}$  is a rectangular matrix whose  $k$ th (block) row contains the Jacobian matrix  $\mathbf{J}_k$  in (8), and  $\mathbf{b}$  is a vector whose  $k$ th (block) row equals  $-e_k(\theta_k^0)$ . Equation (9) can be considered as a linearized version of the SLAM problem whose graph structure is represented by the sparsity pattern of  $\mathbf{A}$ . Hereafter we will refer to (9) as the *linear system* or the *Gaussian factor graph* of a SLAM problem, and refer to  $\mathbf{A}$  as the *Jacobian matrix*, refer to  $\mathbf{\Lambda} = \mathbf{A}^T \mathbf{A}$  as the information matrix.

In batch scenario, we assume an initial estimate and all of the measurements are available, so we typically iteratively solve (9) to update the current estimates until convergence. Yet in online SLAM problems, the measurements are not available at once but come in sequentially. Intuitively, the techniques designed for batch scenario can be applied to solve online problems, e.g., fixed-lag smoothing or periodic global optimization, but their performance are suboptimal.

### C. Incremental Smoothing and Mapping

iSAM aims to solve online SLAM problems by performing fast incremental updates of the square root information matrix when new measurements are added [24]. Recently, it has been shown that Bayes trees provide a better understanding of the matrix factorization and allows incremental re-ordering and just-in-time relinearization [6]. These desirable properties make iSAM one of the state-of-the-art methods for online SLAM problems. Generally speaking, iSAM can achieve constant computational complexity when the robot is exploring the environment, but the performance degrades when there are many loop-closures because iSAM will have to frequently reorder the Bayes Tree as well as update the corresponding conditional densities.

### D. Subgraph-Preconditioned Conjugate Gradient Method

To efficiently solve SLAM with many loop-closures, the subgraph-preconditioned conjugate gradient (SPCG) method has been proposed [20], [21]. The main idea is to combine the advantages of direct and iterative methods by identifying a *sparse* subgraph and then solve it with direct methods to build a prior probability density to precondition the original problem. Using a sparse subgraph has the advantage that solving the sub-problem and applying the preconditioner can both be performed efficiently. Yet as a batch technique, SPCG requires good initial estimates to obtain good results.

### III. INCREMENTAL SPCG

We present the incremental subgraph-preconditioned conjugate gradient (iSPCG) method that combines two state-of-the-art techniques, iSAM and SPCG, to efficiently solve online SLAM problems with many loop-closures. In SLAM, a loop-closure refers to an event that the robot recognizes a previously mapped area. Loop-closures are essential to limit the growth of uncertainty and improve the estimate.

First we use iSAM to incrementally solve a sparse subgraph to obtain an approximate solution. When the error grows larger than a threshold or the optimal solution is requested, we apply SPCG [20], [21] to solve the entire graph to obtain the optimal solution. We note that the subgraph preconditioner and the initial estimate for SPCG are provided by iSAM. Then we use the optimal solution from SPCG to regularize iSAM's estimate in the following steps. The detail of each step will be explained in the following sections.

#### A. Solving Subgraphs with iSAM

Considering a SLAM problem as a factor graph  $G = (V, E)$ , where  $V$  denotes the robot poses, and  $E$  denotes the measurements (factors). We incrementally separate the graph into two parts: the *subgraph* part  $H = (V, E_H)$  and the *constraint* part  $C = (V, E_C)$ . Then we use iSAM to incrementally solve the subgraph part, and constantly keep an approximate solution. To evaluate the quality of the approximate solution, we compute the normalized chi-square error on the subgraph part  $\chi_H^2$ , on the constraint part  $\chi_C^2$  and on the entire graph  $\chi_G^2$ , respectively. If the error is small, i.e.,

$$\chi_G^2 \leq \tau_g \quad \text{and} \quad \frac{\chi_C^2}{\chi_H^2} \leq \tau_r, \quad (10)$$

where  $\tau_g$  and  $\tau_r$  are thresholds, we accept iSAM's solution and proceed to the next iteration.

We note that this approximation scheme is efficient if the subgraph is sparse because the associated Bayes Tree and conditional densities can be updated efficiently.

#### B. Solving Original Graphs with SPCG

If iSAM's approximate solution leads to high error or the optimal solution is requested by the user, we apply SPCG to solve the entire graph with iSAM's solution as the initial estimate and iSAM's factorization of the approximate information matrix as the subgraph preconditioner. Since iSAM typically provides good initial estimates, and the subgraph preconditioner can effectively reparametrize the problem, SPCG will converge in a few iterations. Consequently, we will obtain a locally optimal solution.

#### C. Regularizing iSAM with SPCG's Solutions

Finally, we have to inform iSAM with the optimal solution from SPCG, otherwise iSAM will drift again in the next iterations. To this end, we add prior factors to the subgraph

$$e_p(\theta_k) = \frac{1}{2} \|\theta_k - \theta_k^{spcg}\|_{\Sigma_k}^2 \quad \forall \quad k = 1, 2, \dots, n \quad (11)$$

where  $\theta_k^{spcg}$  denotes SPCG's solution of the  $k$ th variable,  $\Sigma_k$  is the covariance matrix of the prior factor, and  $n$  is the

---

#### Algorithm 1: One step of the iSPCG algorithm

---

**Input:**  $G_{t-1}$  is the current factor graph, and  $H_{t-1}$  is a subgraph of  $G_{t-1}$ .  $F_t$  denotes new factors.

**Output:** new estimate  $\theta_t$

1. split  $F_t = F_t^H \cup F_t^C$  into subgraph and constraints parts
  2. use iSAM to solve the new subgraph  $H_t = H_{t-1} \cup F_t^H$
  3. **if** iSAM's solution  $\theta_t^{isam}$  is acceptable **then return**  $\theta_t^{isam}$
  4. use SPCG to solve the new graph  $G_t = G_{t-1} \cup F_t$  with  $\theta_t^{isam}$  as the initial estimate
  5. use SPCG's solution  $\theta_t^{spcg}$  to regularize iSAM hereafter
  6. **return**  $\theta_t^{spcg}$
- 

number of variables. Note that these prior factors only exist in iSAM and will not affect the optimal solution, and they will be replaced after next invocation of SPCG. The key steps of iSPCG are illustrated with a simple example in Fig. 1 and summarized in Algorithm 1.

### IV. THE CONSISTENCY OF iSPCG

Here we prove a sufficient condition that iSPCG is consistent. The consistency of an online SLAM method is important because it can prevent us from being over-confident about the current estimates and therefore help us make conservative data association in the SLAM frontend. While previous work used convex optimization techniques to enforce the consistency [25], [12], here we show that the proposed method can be proved to be consistent.

We start by defining the notion of consistency:

**Definition 1** (Consistency). *The estimate of the mean and covariance  $\Sigma$  of Gaussian random variables is consistent if*

$$E[\hat{\mu} - \mu] = 0 \quad (12)$$

$$\hat{\Sigma} \succeq \Sigma \quad (13)$$

where  $(\mu, \Sigma)$  denote the true values, and  $(\hat{\mu}, \hat{\Sigma})$  denote the estimate [26].

Proving (12) is typically simple due to the Gaussian assumption and the central limit theorem. Therefore, we focus on proving (13). We need the following four lemmas to prove the consistency of iSPCG.

**Lemma 2.** *Suppose  $\mathbf{X}$  and  $\mathbf{Y}$  are symmetric and positive-definite matrices, then  $\mathbf{X} \succeq \mathbf{Y}$  if and only if  $\rho(\mathbf{X}^{-1}\mathbf{Y}) \leq 1$ , where  $\rho(\cdot)$  denotes the largest eigenvalue [27, p.471].*

**Lemma 3.** *Suppose  $\mathbf{X}$  and  $\mathbf{Y}$  are positive-definite matrices. If  $\mathbf{X} \succeq \mathbf{Y}$ , then  $\mathbf{X}^{-1} \preceq \mathbf{Y}^{-1}$ .*

*Proof.* Since  $\mathbf{X} \succeq \mathbf{Y}$ , according to Lemma 2, we know  $\rho(\mathbf{X}^{-1}\mathbf{Y}) \leq 1$ . Since  $\rho(\mathbf{X}^{-1}\mathbf{Y}) = \rho(\mathbf{Y}\mathbf{X}^{-1})$ , therefore we obtain  $\rho(\mathbf{Y}\mathbf{X}^{-1}) \leq 1$  and then  $\mathbf{X}^{-1} \preceq \mathbf{Y}^{-1}$ .  $\square$

**Lemma 4.** *Given a real symmetric matrix  $\mathbf{X} \in \mathbb{R}^{n \times n}$ , and its eigenvalues  $\{\lambda_i\}_{i=1}^n$ , then the eigenvalues of  $(\mathbf{X} + t\mathbf{I})$  are  $\{\lambda_i + t\}_{i=1}^n$ .*

*Proof.* Suppose  $v_i$  is the  $i$ th eigenvector of  $\mathbf{X}$ , then we obtain  $(\mathbf{X} + t\mathbf{I})v_i = \mathbf{X}v_i + t\mathbf{I}v_i = \lambda_i v_i + t v_i = (\lambda_i + t)v_i$   $\square$

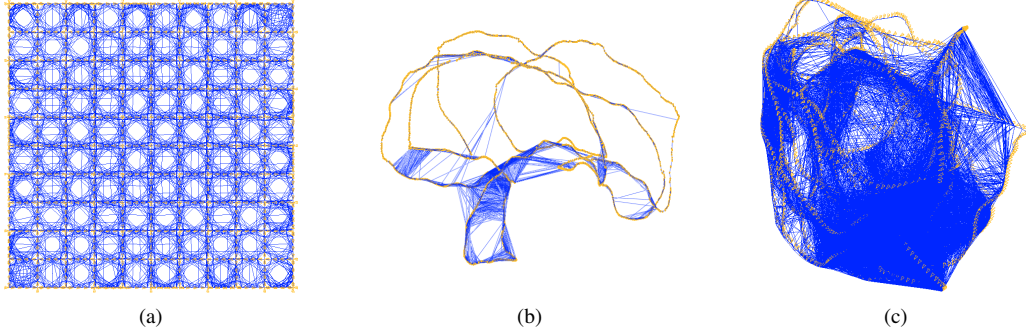


Fig. 2: The bird's-eye views of (a) a Blockworld problem with 1,000 robot poses (yellow) and 10,000 constraints (blue), (b) the "Lab02" problem, and (c) the "Cubicle02" problem.

**Lemma 5.** *Solving a subgraph leads to consistent estimates.*

*Proof.* Consider rearranging the Jacobian matrix in (9) into

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_H \\ \mathbf{A}_C \end{bmatrix} \quad (14)$$

where  $\mathbf{A}_H$  denotes the Jacobians associated to the subgraph, and  $\mathbf{A}_C$  denotes that of the remaining part. Suppose  $\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$  denotes the information matrix, we need to prove that  $\mathbf{\Sigma}_H \succeq \mathbf{\Sigma}$ . From (14), we can see that  $\mathbf{\Lambda} - \mathbf{\Lambda}_H = \mathbf{A}^T \mathbf{A} - \mathbf{A}_H^T \mathbf{A}_H = \mathbf{A}_C^T \mathbf{A}_C \succeq \mathbf{0}$ . Therefore,  $\mathbf{\Lambda} \succeq \mathbf{\Lambda}_H$ . Using Lemma 3, we can obtain  $\mathbf{\Sigma}_H = \mathbf{\Lambda}_H^{-1} \succeq \mathbf{\Lambda}^{-1} = \mathbf{\Sigma}$ .  $\square$

**Lemma 6.** *Adding the regularization terms in (11) to a subgraph maintains the consistency if  $\mathbf{\Sigma}_k = t^{-1}\mathbf{I}$  and  $t \leq \lambda_{\min}(\mathbf{\Lambda}_C)$ , where  $\lambda_{\min}(\cdot)$  denotes the smallest eigenvalue.*

*Proof.* The information matrix of the regularized subgraph is  $(\mathbf{\Lambda}_H + t\mathbf{I})$ . To prove the lemma, we need to show  $\mathbf{\Lambda} \succeq (\mathbf{\Lambda}_H + t\mathbf{I})$ . We can see that  $\mathbf{\Lambda} - (\mathbf{\Lambda}_H + t\mathbf{I}) = (\mathbf{\Lambda}_H + \mathbf{\Lambda}_C) - (\mathbf{\Lambda}_H + t\mathbf{I}) = (\mathbf{\Lambda}_C - t\mathbf{I})$ . Suppose the eigenvalues of  $\mathbf{\Lambda}_C$  are  $\{\lambda_i\}_{i=1}^n$ . Using Lemma 4, we know the eigenvalues of  $(\mathbf{\Lambda}_C - t\mathbf{I})$  are  $\{\lambda_i - t\}_{i=1}^n$ . Since  $t \leq \min_i \lambda_i$ , we know  $\lambda_i - t \geq 0$ , for all  $i$ . Hence  $(\mathbf{\Lambda}_C - t\mathbf{I}) \succeq \mathbf{0}$  and  $\mathbf{\Lambda} \succeq (\mathbf{\Lambda}_H + t\mathbf{I})$ .  $\square$

**Corollary 7.** *iSPCG gives consistent and optimal estimates.*

*Proof.* The discussion can be splitted into two parts: (1) For the iSAM part, using Lemma 5, we know that using iSAM to solve a subgraph always leads to consistent estimates. Moreover, using iSAM to solve a *regularized* subgraph also leads to consistent estimates if we assign the covariance matrices according to Lemma 6. The solution from iSAM is close to optimal in the sense that the normalized chi-square error is always smaller than a predefined threshold. (2) For the SPCG part, the estimates are both consistent and optimal because they are obtained by solving the original graph.

Note that in Lemma 6 we assume the linearization points for both the subgraph and the original graph are identical, but this assumption is not always true for nonlinear SLAM problems. Nevertheless, the difference between two linearization points is bounded because iSAM would relinearize whenever there are sufficient changes in the current estimates.  $\square$

## V. RESULTS

We conducted experiments to evaluate the accuracy, speed and scalability of iSPCG, and compare it with iSAM [6] on simulated and real datasets. For iSPCG, we used a subgraph consisting of the odometry chain of the robot poses plus  $n$  randomly selected edges, where  $n$  is the number of robot poses. Such a simple choice has shown its effectiveness in [21]. The thresholds in (10) are empirically set to  $\tau_g = 10^{-2}$  and  $\tau_r = 5.0$  respectively. We also used the inverse iteration method [28] to estimate the smallest eigenvalue to determine the proper covariance matrices of the regularization terms in (11). For iSAM, we used the implementation in GTSAM [29] with default parameters. We ran all of the experiments with single thread on a PC with an Intel Core i7 CPU.

### A. Simulated Datasets

To facilitate the comparison, we generated a number of synthetic Blockworld problems, simulating a robot traversing a block world. The bird's-eye view of this problem is illustrated in Fig. 2a. For each robot pose, we added various number of constraints to its closest neighbors, and these measurements are contaminated by zero-mean and normally distributed noise. To make the SLAM problem well-posed, we attached a prior factor to the first robot pose.

1) *Accuracy:* We evaluated the accuracy of different solvers on a Blockworld problem with 1,000 poses and 20,000 measurements, and showed the results in Fig. 3. Note that "iSAM-full" means using iSAM to solve the entire graph, while "iSAM-subgraph" means using iSAM to solve the subgraph as in iSPCG. In Fig. 3a, we can see that both iSPCG and iSAM-full can achieve lower errors because they both aim to solve the original problem. Yet iSAM-subgraph consistently has larger errors because it only used part of the information. Moreover, in some of our trials, we observed that iSAM-subgraph cannot solve the problems.

2) *Timing:* We also evaluated the running time of different solvers on the same dataset, and reported the results in Figs. 3b and 3c. We can see that iSAM-full quickly becomes expensive because of many loop-closures, which makes it unsuitable for large-scale problems. iSAM-subgraph is very

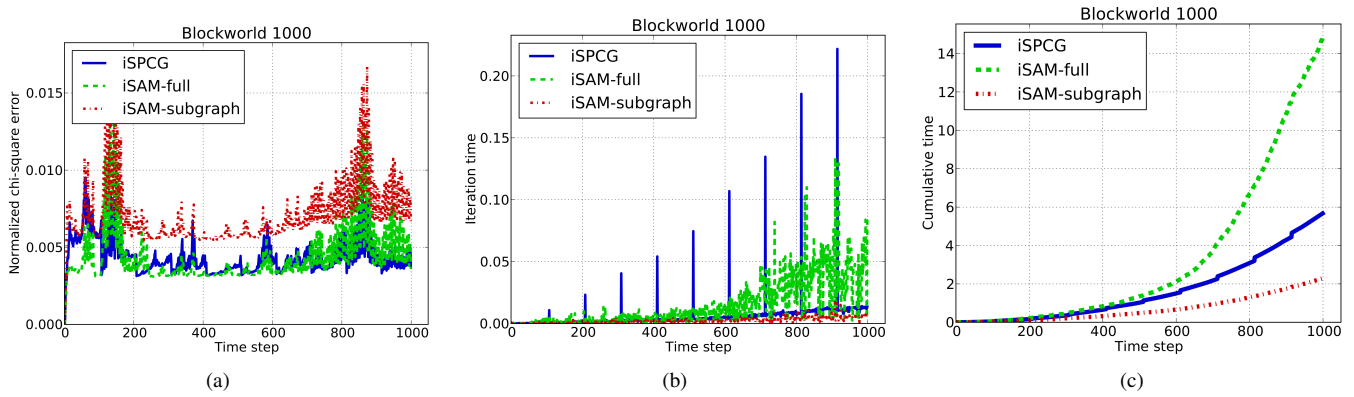


Fig. 3: The results on a synthetic Blockworld dataset with 1,000 poses and 20,000 measurements. (a) The normalized chi-square error. (b) The processing time per time step. (c) The cumulative processing time.

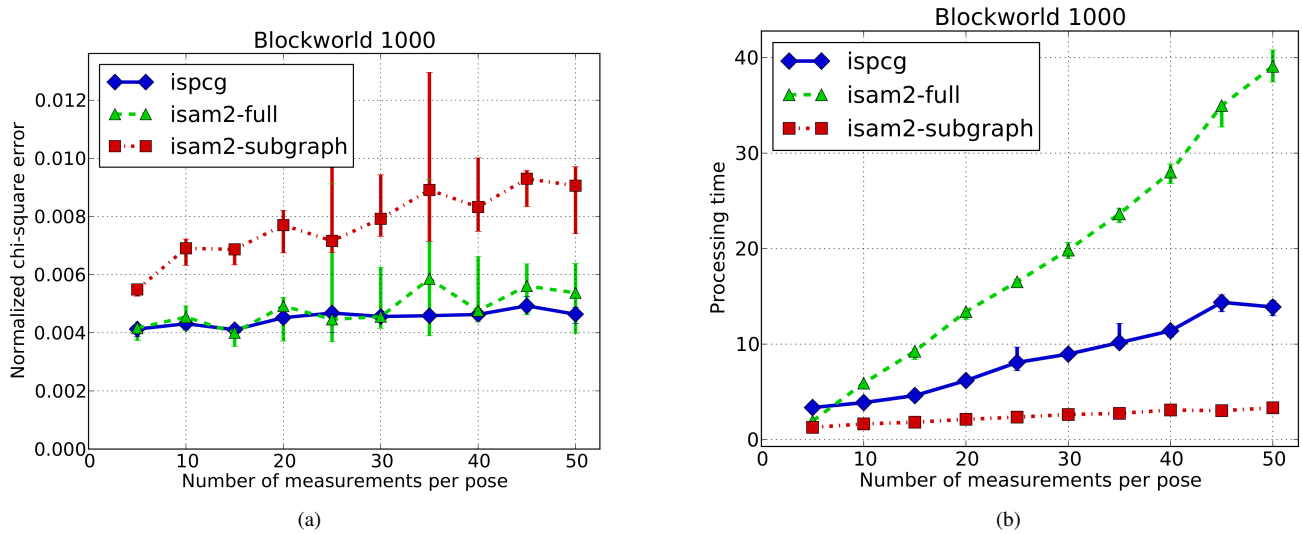


Fig. 4: The results on Blockworld datasets with different number of loop-closures.

efficient but it also leads to higher errors or potentially wrong solutions. By combining the advantages of iSAM and SPCG, iSPCG can be more than two times faster than iSAM-full and also obtain high-quality solutions.

Notably, from Fig. 3b, we can see that iSPCG periodically has a spike, which is undesirable for online applications. We observed that this happened when the solution of iSAM is unsatisfactory and SPCG has to be invoked to optimize the full graph. One way to resolve this problem is by splitting iSPCG into two threads: one thread running iSAM in the frontend, and the other thread running SPCG in the backend. In addition, we observed that there is a tradeoff between the quality of solutions obtained from iSPCG as well as the efficiency of iSPCG. That is, the smaller the thresholds  $\tau_g$  and  $\tau_r$ , the better solutions we obtain, but the more often we have to run to solve the full graphs. How to automatically determine these thresholds is another interesting question. We plan to explore these two directions in future work.

3) *Scalability*: We evaluated the performance of different solvers on Blockworld datasets with various number of loop-closures. We reported the tenth percentile, the median and

the ninetieth percentile over twenty trials of the final errors and total processing times in Fig. 4.

We can see that iSAM-full consistently achieves lower errors as in the previous experiments because it aims to solve the original problem. iSAM-full is also efficient when the number of loop-closures is small, but quickly becomes expensive when the number of loop-closures increases, which makes it unsuitable for large-scale problems.

For iSAM-subgraph, since it aims to solve a sparse subproblem, its efficiency is always good and independent of the number of loop-closures. Yet we observed that its error is not only consistently higher than that of iSAM-full and iSPCG, but also increases with the number of loop-closures. These properties also make iSAM-subgraph unsuitable for obtaining high-quality solutions.

For iSPCG, it can provide high-quality solutions, and also be up to four times faster than iSAM-full. These properties make iSPCG a better choice for large-scale SLAM problems with many loop-closures.



TABLE I: The timing results on real datasets in seconds. The "Ratio" column indicates the ratio between the number of measurements to the number of poses, which can be an indicator of the difficulty of the problem.

Name	Poses	Measurements	Ratio	iSPCG	iSAM
Killian	1,941	3,995	2.1	4.6	<b>3.1</b>
Intel	910	4,454	4.9	2.0	<b>1.1</b>
Lab02	1,998	15,505	7.8	<b>17.5</b>	18.1
Cubicle02	1,998	33,234	16.6	<b>69.2</b>	441.6

TABLE II: Comparison between different methods for SLAM problems with many loop-closures.

Method	Efficient	Consistent	Optimal
iSAM	×	○	⊙
Marginalization	×	○	⊙
Vertex removal	○	×	×
Edge removal (subgraph)	○	○	×
Chow-Liu tree approx.	○	×	×
Convex optimization	×	○	×
iSPCG	○	○	○

### B. Real Dataset

We also evaluated the performance of iSPCG and iSAM on four real datasets. The "Killian" and "Intel" datasets are publicly available. The "Lab02" and "Cubicle02" datasets are collected by the authors with a Videre STOC camera in an open office environment, where the camera constantly visits the same place to create many loop-closure constraints. We used the vocabulary tree technique [30] implemented in [31] to generate loop-closure constraints. The latter two datasets can be downloaded from the authors' website.

From the results in Table I, we can see that iSAM is more efficient when the SLAM problems do not have many loop-closures, i.e., the ratio between the number of measurements to the number of poses is low. We observed that iSAM is up to two times faster in "Killian" and "Intel" datasets. However, when the ratio becomes larger, iSPCG starts to show its advantages. On the "Cubicle02" dataset, we observed that iSPCG is 6.3 times faster than iSAM. We omitted the normalized chi-square errors because both methods achieve similar errors for all datasets.

## VI. RELATED WORK

Solutions to the online SLAM problem have been well-studied in literature. Here we focus on recent results of pose graph optimization [3], [4], [5], and refer the readers to [1] and [2] for the developments of filtering-based methods.

One of the main challenges to SLAM methods is scalability. In this paper, we addressed the scalability to the number of loop-closures. Many techniques have been proposed and they can be divided into the following categories.

The first category aims to build an intermediate representation of the problem so that the estimate can be obtained ef-

ficiently, e.g., incremental smoothing and mapping [24], [6], and hierarchical optimization [7]. Although these methods are efficient for sparse problems, they do not scale well when there are many loop-closures. The main reason is that they all aim to factorize the information matrix which is expensive for large-scale problems. Nevertheless, the concepts of these techniques are useful, and therefore we design our method based on one of the state-of-the-art methods in this category.

The second category aims to *sparsify* the robot poses. Earlier work selects keyframes or skeleton graphs [32], [8]. Although these techniques can effectively downsize the problem, they typically lead to information loss and inconsistent estimation. Recent work marginalizes redundant robot poses and induces additional constraints (pseudo loop-closures) between the adjacent poses [9], [10], [11], [12], [33]. These techniques can effectively reduce the number of poses and may lead to consistent estimation, but the graphs after marginalization typically become more dense than the original graphs. This implies that marginalization has to stop at some point because it would eventually become expensive due to the increasing size of the associated clique. Therefore one still has to solve a graph with many loop-closures in the end, which is the place the proposed method can be applied.

The third category aims to sparsify the loop-closures. This process can be guided by thresholding the number of loop-closures per robot pose [9], thresholding the expected information gain [10], locally approximating with a Chow-Liu tree in the information matrix [13], [14]. Yet these techniques lead to either information loss or inconsistent estimate, which is suboptimal. Consistent edge sparsification methods have been proposed, but they require solving a convex optimization problem, which might be too expensive for large-scale problems [12]. In contrast, the proposed method constantly solves a regularized sparse subgraph, which can be done efficiently, and also proven to be consistent.

The fourth category aims to reparametrize the problem so that the solution can be obtained faster. Incremental pose reparametrization over the odometry chain or a spanning tree of the graph has been used to improve the convergence speed of the stochastic gradient descent method [18], [19]. Using sparse subgraphs to reparametrize (precondition) the SLAM problems has been shown to be able to effectively improve the convergence speed of the conjugate gradient method [20], [21]. Yet these techniques are designed for batch SLAM problems. Notably, Sibley et al. [34] showed that using relative pose parametrization makes it possible to incrementally solve SLAM in constant time.

iSPCG combines the advantages of the first, the third and the fourth categories and is a consistent and efficient method for online SLAM problems with many loop-closures. The solutions are close to optimal in the iSAM steps and optimal in the SPCG steps. The comparison between the above methods is summarized in Table II.

## VII. CONCLUSIONS AND FUTURE WORK

We proposed a new method, iSPCG, to efficiently solve online SLAM problems with many loop-closures. To sum

up, iSPCG has the following advantages: (1) iSPCG is efficient because it combines the advantages of two state-of-the-art SLAM methods, iSAM and SPCG. The iSAM part is efficient because it only has to solve sparse subgraphs. The SPCG part is also efficient because it scales well to the number of loop-closures, utilizes the subgraph preconditioners and initial estimate provided by iSAM, and only being invoked whenever necessary. Finally, iSPCG used the optimal solution from SPCG to regularize iSAM in the next steps. (2) We proved that iSPCG can be consistent, while in previous work such property is usually not guaranteed or has to be enforced by convex optimization techniques. Although the data association problem is not addressed in this paper, the consistency of iSPCG actually can help make conservative data association in the SLAM frontend. (3) iSPCG aims to find the optimal solution because it does not discard any measurements. We applied iSPCG to solve large SLAM problems and obtained promising results.

There are several directions for future work. The first is to design a new metric to evaluate the quality of a subgraph for iSPCG, and then use this metric to design an algorithm to incrementally find good subgraphs. Intuitively, this metric should consider the computational complexity of using iSAM to solve the subgraph, the quality of the approximate estimate obtained from iSAM, and the quality of subgraph preconditioner for SPCG. The second is to derive more versatile sufficient conditions to guarantee the consistency of iSPCG. The third is to develop an algorithm to automatically decide the thresholds in iSPCG. The algorithm should consider the tradeoff between the quality of the solutions obtained from iSAM and the time spent on running SPCG. At last, we would like to improve the efficiency of iSPCG by utilizing multiple cores on modern CPUs. Similar to the idea in PTAM [32], we can split iSPCG into two threads: one thread running iSAM to obtain the current estimates in the frontend, and the other thread running SPCG to obtain the optimal estimates in the backend.

## REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *Robotics & Automation Magazine*, Jun 2006.
- [2] T. Bailey and H. Durrant-Whyte, "Simultaneous Localisation and Mapping (SLAM): Part II State of the Art," *Robotics & Automation Magazine*, Sep 2006.
- [3] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, pp. 333–349, Apr 1997.
- [4] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.
- [5] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [6] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree," *Intl. J. of Robotics Research*, vol. 31, pp. 217–236, Feb 2012.
- [7] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.
- [8] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008.
- [9] E. Eade, P. Fong, and M. E. Munich, "Monocular graph slam with complexity reduction," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, October 2010.
- [10] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact pose slam," *IEEE Trans. Robotics*, vol. 26, no. 1, pp. 78–93, 2010.
- [11] H. Johannsson, M. Kaess, M. Fallon, and J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013.
- [12] G. Huang, M. Kaess, and J. J. Leonard, "Consistent sparsification for graph optimization."
- [13] H. Kretschmar, C. Stachniss, and G. Grisetti, "Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.
- [14] N. Carlevaris-Bianco and R. M. Eustice, "Generic factor-based node marginalization and edge sparsification for pose-graph slam," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013.
- [15] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *IEEE International Conference on Robotics and Automation*, 2000.
- [16] A. Howard, M. Mataric, and G. Sukhatme, "Relaxation on a mesh: a formalism for generalized localization," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2001.
- [17] U. Frese, P. Larsson, and T. Duckett, "A Multilevel Relaxation Algorithm for Simultaneous Localisation and Mapping," *IEEE Trans. Robotics*, vol. 21, no. 2, pp. 196–207, April 2005.
- [18] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. of IEEE International Conference on Robotics and Automation*, 2006.
- [19] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proc. of Robotics: Science and Systems (RSS)*, 2007.
- [20] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. E. Thorpe, "Subgraph-preconditioned Conjugate Gradient for Large Scale SLAM," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [21] Y.-D. Jian, D. Balcan, I. Panageas, P. Tetali, and F. Dellaert, "Support-Theoretic Subgraph Preconditioners for Large-Scale SLAM," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.
- [22] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [23] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Verlag, 1999.
- [24] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec 2008.
- [25] J. Vial, H. Durrant-Whyte, and T. Bailey, "Conservative sparsification for efficient and consistent approximate estimation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.
- [26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.
- [27] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
- [28] L. Trefethen and D. Bau, *Numerical linear algebra*. SIAM, 1997.
- [29] "GTSAM <https://collab.cc.gatech.edu/borg/gtsam>."
- [30] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [31] C. Beall, B. Lawrence, V. Ila, and F. Dellaert, "3D reconstruction of underwater structures," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [32] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, 2007.
- [33] N. Carlevaris-Bianco and R. M. Eustice, "Long-term simultaneous localization and mapping with generic linear constraint node removal," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, November 2013.
- [34] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," in *Robotics: Science and Systems (RSS)*, 2009.