Fast Pose Graph Optimization Algorithm Based on Eigen Decomposition: Postprint

Authors: Yujie Li, Wei Guoliang, Cai Jie, Wang Miaomiao

Date: 2022-06-06T00:00:00+00:00

Abstract

Pose graph optimization (PGO) is a high-dimensional non-convex optimization algorithm widely employed in computer vision. It is difficult to solve directly and primarily relies on iterative techniques that demand high-quality initializations, which are hard to guarantee in practice. This work investigates the pose graph optimization problem and proposes a simple closed-form solution algorithm based on eigen decomposition. The algorithm first applies semidefinite relaxation to the maximum likelihood estimation of the PGO problem, then transforms it into an eigen decomposition problem, and exploits data sparsity to design an improved model order reduction method for solving, thereby further enhancing computational efficiency. The algorithm offers advantages in scalability, computational cost, and accuracy. Finally, experimental evaluations on both simulated and real pose graph datasets demonstrate that the algorithm can rapidly perform pose graph optimization without compromising accuracy.

Full Text

Preamble

Vol. 39 No. 10
Application Research of Computers
Accepted Paper

Fast Pose Graph Optimization Algorithm Based on Eigen Decomposition

Li Yujie[1], Wei Guoliang[2]†, Cai Jie[1], Wang Miaomiao[1]
[1] College of Science, [2] Business School
University of Shanghai for Science and Technology, Shanghai 200093, China

Abstract: Pose graph optimization (PGO) is a high-dimensional non-convex optimization algorithm widely used in computer vision. It is difficult to solve directly and primarily relies on iterative techniques that demand high-quality initial values, which are difficult to guarantee in practice. This paper investigates the PGO problem and proposes a simple closed-form solution algorithm for pose graphs based on eigen decomposition. The algorithm first develops a semidefinite relaxation of the maximum likelihood estimation (MLE) for PGO problems, then transforms it into an eigen decomposition problem. Leveraging the sparsity of the data, we design an improved model reduction method to solve the problem, which further enhances computational speed. The algorithm offers advantages in scalability, low computational cost, and high accuracy. Finally, experimental evaluations on simulated and real-world pose-graph datasets demonstrate that the algorithm can perform pose graph optimization rapidly without compromising accuracy.

Keywords: pose graph optimization; maximum likelihood estimation; eigen decomposition; model reduction

0 Introduction

Simultaneous Localization and Mapping (SLAM) enables robots to estimate their own motion and build environmental models without prior information about the environment [1], which is essential for autonomous navigation and exploration in unknown scenes. In recent years, SLAM technology has been widely applied in daily life, including autonomous driving [2], search and rescue [3], and precision agriculture [4], making it a prominent research focus. SLAM systems consist of front-end and back-end components, as shown in [Figure 1: see original paper]. The front-end visual odometry (VO) estimates relative camera motion between adjacent images, but this process suffers from cumulative errors. The back-end must optimize the front-end output to obtain optimal pose estimates and a globally consistent [5] map, thereby improving SLAM system accuracy.

Pose graph optimization (PGO) is currently the most widely used SLAM back-end optimization technique in robotics [6]. It provides an intuitive graph-based representation of the SLAM problem, where graph vertices represent robot poses and edges represent relative measurements. Optimizing this graph yields more accurate results. PGO objective functions are typically high-dimensional [7], involving numerous variables and constraints, which demands high computational efficiency from optimization algorithms. Additionally, due to the non-convexity of the objective problem, solvers can easily become trapped in local minima, rendering estimates completely useless. Therefore, avoiding local minima while ensuring computational efficiency is paramount for PGO.

Current PGO solution algorithms can be divided into three categories [8]. The first category uses the problem’ s global structure for iterative solutions. Lu and Milios [9] first proposed representing SLAM problems using pose graphs and performed iterative nonlinear optimization. Olson et al. [10] introduced a stochastic gradient descent method for PGO, which exhibits good stability and scalability, enabling rapid robot pose optimization even with poor initial estimates. Grisetti et al. [11] improved upon Olson' s work by using tree structures to define and update local regions, significantly enhancing the convergence of stochastic gradient descent. Kaess et al. [12–14] proposed incremental smoothing algorithms that better handle nonlinear measurement models and improve optimization efficiency through incremental reordering, enabling updates for large-scale PGO problems. Rosen et al. [15] investigated the least-squares structure of PGO problems, identifying possibilities for reducing nonlinearity and non-convexity. Kümmerle et al. [16] exploited the sparsity of graph structures using the Gauss-Newton method. However, all these nonlinear optimization techniques are local search methods that cannot guarantee solution optimality.

The second category aims to find good initial values to improve convergence speed and reduce the risk of converging to local minima. Carlone et al. [17] proposed the LAGO algorithm, which provides accurate initial estimates for 2D PGO problems but only handles medium-to-low noise datasets. A subsequent review [18] compared several state-of-the-art initialization methods, concluding that using the chordal method as an initializer with GTSAM as the solver represents the best combination for iterative solutions. Nasiri et al. [19] proposed the RS algorithm for PGO, which achieves accurate approximations of optimal solutions even under high noise, demonstrating good robustness.

The third category relaxes PGO problems to overcome objective non-convexity. Carlone et al. applied semidefinite relaxation to the original problem, using Lagrangian duality theory to verify solution optimality in 2D [20] and 3D [21] SLAM, proving that when the duality gap is zero, the global optimal solution can be retrieved. Rosen et al. [22] leveraged the low-rank geometric structure of PGO problems to reduce them to equivalent optimization problems on low-dimensional Riemannian manifolds, designing a truncated trust-region method that is orders of magnitude faster than interior-point methods. Eriksson et al. [23] used graph theory for theoretical analysis of rotation averaging problems, deriving noise-level error bounds to ensure global optimality. However, since these solutions are obtained through constraint relaxation, they typically lie outside the original problem' s feasible domain and require re-projection.

In summary, current PGO algorithms can converge to approximate global optimal solutions but suffer from slow computation and high computational costs. To accelerate solution efficiency, Singer [24] proposed eigen decomposition for angular synchronization, but this only applies to 2D problems and cannot extend to the more commonly used 3D problems. Subsequent work [25] extended eigen decomposition to 3D problems but only computed the rotation component, ignoring translation optimization. Arrigoni et al. [26] proposed using eigen decomposition for motion synchronization, optimizing both rotation and translation simultaneously (i.e., on the linear group), but this increased optimization matrix dimension, significantly impacting solution speed as problem scale grows, and sacrificed estimation accuracy to obtain solutions satisfying the required form.

Building upon these issues, this paper proposes a fast pose graph optimization algorithm based on eigen decomposition, called EDPO. The algorithm optimizes in two stages: the first stage uses an improved eigen decomposition algorithm to optimize rotation matrices, and the second stage uses least squares to solve for translation. The algorithm offers scalability, low computational cost, and high accuracy. The main contributions are:

a) A novel PGO algorithm that divides the original problem into two subproblems. First, we relax the determinant constraint on rotation matrices, derive a closed-form solution based on eigen decomposition, then project the solution onto the orthogonal group to satisfy rotation matrix properties. Subsequently, we optimize the translation component using least squares.

b) A concise matrix formulation that rewrites the original PGO problem into a form amenable to eigen decomposition, establishing connections with the Laplacian matrix of the corresponding graph structure to facilitate eigen decomposition. This matrix formulation can handle missing data and provides more accurate computation.

c) An improved model reduction method to enhance computational performance for the derived eigen decomposition model, efficiently solving high-dimensional problems without compromising accuracy.

d) Comparative experiments on simulated and real PGO datasets demonstrate that EDPO significantly improves computational efficiency without affecting accuracy.

1.1 Graph Theoretic Foundations

A graph structure can be represented as $G = (V, E)$, where V is a non-empty set called the vertex set and E is a set of unordered pairs of elements from V, called the edge set. In this paper, the number of vertices and edges are denoted by n and m, respectively. A graph is complete if every pair of distinct vertices is connected by exactly one edge (as shown in [Figure 2: see original paper]).

Let A be the adjacency matrix of G with elements a_{ij}, and let D be the diagonal degree matrix where diagonal elements represent vertex degrees, i.e., $d_{ii} = \sum_j a_{ij}$. The Laplacian matrix L of G is then $L = D - A$. Using the example in [Figure 2: see original paper], the adjacency matrix A, degree matrix D, and Laplacian matrix L are:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}, \quad L = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

A weighted graph is a triplet $G = (V, E, w)$ consisting of a graph $G = (V, E)$ and a weight function w defined on its edges. In some applications, non-negative weights w_{ij} reflecting measurement reliability are given and stored in the graph' s adjacency matrix A. Therefore, the degree matrix D for a weighted graph is defined as $d_{ii} = \sum_j w_{ij} a_{ij}$.

1.2 Pose Graph Optimization

Pose graphs are a specific form of graph optimization where vertices represent robot poses and edges represent relative measurements. This paper uses them to solve robot trajectory optimization problems.

Both poses and relative measurements are elements of the special Euclidean group $SE(3)$. PGO computes the maximum likelihood estimate of n robot poses given m pairs of relative measurements. In 3D problems, the absolute pose $x_i \in SE(3)$ contains rotation matrix $R_i \in SO(3)$ and translation vector $t_i \in \mathbb{R}^3$. Rotation matrices must satisfy both orthogonal constraints $R_i^T R_i = I$ and determinant constraints $\det(R_i) = 1$. We use the notation $x_i = (R_i, t_i)$ for absolute poses and $x_{ij} = (R_{ij}, t_{ij})$ for relative measurements.

This section presents a modified PGO formulation that uses chordal distance to quantify rotation error, beginning with the measurement generative model and deriving the corresponding maximum likelihood estimation.

1.2.1 Measurement Generative Model

Assume the following generative model for relative measurements x_{ij}:

$$t_{ij} = t_j - t_i + t_{ij}^{\omega}, \quad t_{ij}^{\omega} \sim \mathcal{N}(0, \Omega)$$

$$R_{ij} = R_i^T R_j R_{ij}^{\omega}, \quad R_{ij}^{\omega} \sim \text{vonMises}(S, k)$$

where $\mathcal{N}(\mu, \Omega)$ denotes a Gaussian distribution with mean μ and covariance matrix Ω, and $\text{vonMises}(S, k)$ denotes an isotropic von Mises-Fisher distribution with mean S and concentration parameter k.

1.2.2 Maximum Likelihood Estimation

Solving for the maximum likelihood estimate (MLE) of poses is equivalent to minimizing the negative log-likelihood:

$$\{(R_i, t_i)\}_{i=1}^{n} \in SE(3) \quad \min_{(R_i,t_i)} \sum_{(i,j)\in E} -\log L(R_{ij}|R_i, R_j) - \log L(t_{ij}|t_i, t_j, R_i)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

Substituting equations (4) and (5) into (3) yields the maximum likelihood estimate:

$$\{(R_i, t_i)\}_{i=1}^{n} \in SE(3) \quad \min \sum_{(i,j)\in E} \frac{k}{2}\|R_{ij} - R_i^T R_j\|_F^2 + \frac{\tau}{2}\|t_{ij} - (t_j - R_i^T t_i)\|^2$$

The MLE problem is high-dimensional and non-convex. Assuming R^* is the optimal rotation solution to (6), the translation problem becomes a least squares problem [19]. Therefore, we partition the original optimization problem into two subproblems:

$$\min_{R_i\in SO(3)} \sum_{(i,j)\in E} \frac{k}{2}\|R_{ij} - R_i^T R_j\|_F^2 \quad (7)$$

$$\min_{t_i\in \mathbb{R}^3} \sum_{(i,j)\in E} \frac{\tau}{2}\|t_{ij} - (t_j - R_i^{*T} t_i)\|^2 \quad (8)$$

In the following sections, we first solve problem (7), then use the obtained optimal rotation solution to solve problem (8).

2.1 Rotation Optimization Model

To solve the rotation problem (7), we prove that a globally optimal rotation solution R^* can be found through eigen decomposition. For convenient derivation, we use block matrices in subsequent derivations, stacking all pose rotation matrices in $R \in \mathbb{R}^{3n\times 3}$.

First consider the case where all relative measurements are available (i.e., complete graph) and noise-free. Stacking rotation measurements in block matrix form:

$$R = \begin{bmatrix} R_1^T R_2 & R_1^T R_3 & \cdots & R_1^T R_n \\ R_2^T R_1 & R_2^T R_3 & \cdots & R_2^T R_n \\ \vdots & \vdots & \ddots & \vdots \\ R_n^T R_1 & R_n^T R_2 & \cdots & R_n^T R_{n-1} \end{bmatrix}$$

In the noise-free case, the columns of R are eigenvectors of $R^T R$ corresponding to eigenvalue n. Since $\operatorname{rank}(R) = 3$, all other eigenvalues are zero, making n the largest eigenvalue of $R^T R$. Equation (12) is equivalent to:

$$R^T R = nI \quad (13)$$

Now consider the case where relative motion measurements are corrupted by noise. The corresponding optimization problem becomes:

$$\min_{R_i \in O(3)} \sum_{(i,j)\in E} \frac{k}{2}\|R_{ij} - R_i^T R_j\|_F^2 + \frac{n}{2}\|R_i^T R_i - I\|_F^2 \quad (14)$$

Here we relax the determinant constraint on rotation matrices, i.e., $R_i \in O(3)$ instead of $SO(3)$.

In practice, not all relative measurements are available. Missing relative motions correspond to zero blocks in the measurement matrix. Let A represent the adjacency matrix indicating available relative motion information, where $\otimes$ denotes the Kronecker product and $\odot$ denotes the Hadamard product. The adjacency matrix A is expanded in dimension via Kronecker product with I_3 to match the block structure of R. In this case, equation (12) becomes:

$$\min_{R_i \in O(3)} \|(A \otimes I_3) \odot (R - R_{\text{meas}})\|_F^2 \quad (16)$$

Since $D \otimes I_3$ is block diagonal, equation (16) can be further simplified. Let r_i $(i = 1, 2, 3)$ be any three eigenvectors of $M^T M$ with corresponding eigenvalues λ_i. The Lagrangian function is:

$$L(R, \Lambda) = \operatorname{tr}(R^T M^T M R) + \operatorname{tr}(\Lambda(R^T R - nI)) \quad (20)$$

Setting the partial derivative of L with respect to R to zero yields stability conditions. Any three eigenvectors can form a stationary point of the Lagrangian function. If r_i are the eigenvectors corresponding to the three smallest eigenvalues of $M^T M$, we obtain the minimum of problem (18). Thus, the original optimization problem transforms into an eigenvector problem.

We have obtained a closed-form solution for problem (18), but this is under the relaxed formulation and may not fully satisfy rotation matrix properties.

To obtain more accurate rotation estimates, let $\hat{R}$ denote the $3n \times 3$ matrix containing eigenvectors from Proposition 1, where $\hat{R}_i$ represents the estimated rotation for the i-th pose. We find the closest rotation matrix through singular value decomposition:

$$\hat{R}_i = U_i \Sigma_i V_i^T, \quad R_i^* = U_i V_i^T \quad (22)$$

Proposition 1. Problem (18) has a closed-form solution given by three eigenvectors of $M^T M$ associated with its three smallest eigenvalues.

Proof. Problem (18) is equivalent to:

$$\min_{R \in O(3)} \operatorname{tr}(R^T M^T M R) \quad \text{s.t.} \quad R^T R = nI$$

The corresponding Lagrangian function is:

$$L(R, \Lambda) = \operatorname{tr}(R^T M^T M R) + \operatorname{tr}(\Lambda(R^T R - nI))$$

Setting the derivative with respect to R to zero yields $M^T M R = R\Lambda$. Since Λ is symmetric, R must consist of eigenvectors of $M^T M$. The optimal solution uses the three eigenvectors corresponding to the smallest eigenvalues.

2.2 Improved Model Reduction Method

After establishing the rotation optimization model, we need to solve for eigenvalues and eigenvectors. In practical applications, the matrices to be solved are high-dimensional and sparse. Traditional algorithms would extensively fill zero elements, severely consuming memory, necessitating efficient solution methods. The commonly used approach is the Arnoldi model reduction method [26], which approximates the original system with a smaller one, offering strong stability and fast computation suitable for large-scale eigenvalue problems. This paper improves the initialization phase of the Arnoldi model reduction method.

The Arnoldi model reduction method first constructs an orthonormal basis V for the Krylov subspace of matrix $M^T M$. This basis construction requires iterative orthogonalization: compute $w = M^T M v_j$, then orthogonalize w against previous vectors to obtain v_{j+1}.

However, a single orthogonalization may not yield a stable system. Therefore, we add a reorthogonalization step after the initial orthogonalization to ensure better orthonormality of the subspace basis. Algorithm 1 summarizes the improved model reduction method.

Algorithm 1: Improved Model Reduction Method

Input: Symmetric matrix $M^T M$, unit vector v_1, reduction order r
Output: Eigenvalues λ_i and eigenvectors φ_i of $M^T M$

Step 1: For $j = 1$ to $r - 1$
  For $i = 1$ to j // Reorthogonalization
    $h_{ij} = v_i^T w$; $w = w - h_{ij} v_i$
  End for
  $h_{j+1,j} = \|w\|$; $v_{j+1} = w/h_{j+1,j}$
  $V_{j+1} = [V_j, v_{j+1}]$
End for

Step 2: Obtain r-step Arnoldi decomposition: $M^T M V_r = V_r H_r + h_{r+1,r} v_{r+1} e_r^T$

Step 3: Compute eigenvalues λ_i and eigenvectors η_i of H_r

Step 4: Eigenvalues and eigenvectors of $M^T M$ are λ_i and $\varphi_i = V_r \eta_i$

Step 5: Return λ_i, φ_i

2.3 Closed-Form Solution Algorithm Based on Eigen Decomposition

After finding the optimal rotation solution R^* from Section 2.1, problem (8) can be written as:

$$\min_{t_i \in \mathbb{R}^3} \sum_{(i,j)\in E} \frac{\tau}{2} \|t_{ij} - (t_j - R_i^{*T} t_i)\|^2 \quad (23)$$

Similar to Section 2.1, we use block matrices for derivation, stacking translation vectors as $t = [t_1^T, \dots, t_n^T]^T \in \mathbb{R}^{3n}$. Therefore, equation (23) can be rewritten as:

$$\min_{t \in \mathbb{R}^{3n}} \|(B \otimes I_3)t - C\|^2 \quad (24)$$

where $B \in \mathbb{R}^{m\times n}$ is the weighted adjacency matrix expanded via Kronecker product to match the structure, and $C \in \mathbb{R}^{3m}$ contains relative translation measurements $t_{ij}^k = t_{ij} - R_i^{*T} t_j$. This is clearly a linear least squares problem with closed-form solution:

$$t^* = (B^T B)^{-1} B^T C \quad (25)$$

Since $B^T B$ is symmetric, we can use Cholesky decomposition for preconditioning to accelerate matrix inversion. Combining the rotation optimization model yields the complete EDPO algorithm, summarized in Algorithm 2.

Algorithm 2: EDPO Algorithm

Input: Graph vertices V, edges E, relative measurements x_{ij}
Output: Pose estimates $\{(R_i, t_i)\}_{i=1}^n$

Step 1: Partition the original PGO problem into subproblems (7) and (8)

Step 2: Solve subproblem 1
For $(i, j) \in E$: Construct matrices A, D, $L = D - A$
Compute $M^T M = (L \otimes I_3) + \text{blockdiag}(...)$
Use Algorithm 1 to find the three eigenvectors of $M^T M$ corresponding to smallest eigenvalues
For $i = 1$ to n:
$\hat{R}_i = F_i$ where $F_i = R(3i - 2 : 3i, :)$
Compute SVD: $\hat{R}_i = U_i \Sigma_i V_i^T$
$R_i^* = U_i V_i^T$; if $\det(R_i^*) = -1$, adjust sign
End for

Step 3: Solve subproblem 2
For $(i, j) \in E$:
$B_{ij} = w_{ij}$; $C_k = t_{ij} - R_i^{*T} t_j$
End for
$t^* = (B^T B)^{-1} B^T C$

Step 4: Return $\{(R_i^*, t_i^*)\}_{i=1}^n$

---

3 Experimental Results and Analysis

This section evaluates EDPO’ s performance on both simulated and real-world 3D PGO datasets, comparing it against Gauss-Newton (G-N) and EIG [26] algorithms. All experiments were conducted on a Dell OptiPlex 7040 micro desktop with an Intel® Core i5-6500 CPU @ 3.20GHz, 4GB RAM, running Windows 10.

3.1 Simulated Noise Experiments

To investigate the effects of measurement noise (both rotation and translation) and problem scale on EDPO performance, we conducted simulated experiments. First, we added noise to relative measurements from the original cube dataset subset, which simulates a robot trajectory through a regular cube. Measurements were sampled according to equation (2), with rotation noise standard deviation $\sigma_R \in [0^\circ, 10^\circ]$ and translation noise standard deviation $\sigma_T \in [0\text{m}, 1\text{m}]$.

We compared objective function values and runtime of EDPO, G-N, and EIG algorithms across different noise levels and dataset scales. Since our objective function is a loss function, smaller values indicate higher accuracy. All statistics were averaged over 100 random trials.

Figure 3: see original paper and (b) show the impact of rotation noise (with

fixed translation noise $\sigma_T = 0.1$ and fixed pose count $n = 53$) on objective function values and runtime. EDPO is significantly faster than both G-N and EIG, while achieving objective function values comparable to G-N. These results demonstrate that EDPO can obtain approximate global optimal solutions for SLAM problems under reasonable operating conditions.

Figure 4: see original paper-(c) present error comparisons including mean error, median error, and root-mean-square error. For rotation noise $\sigma_R \in [0^\circ, 10^\circ]$, EDPO' s errors are smaller than EIG and comparable to G-N, indicating that EDPO can recover robot pose trajectories under moderate rotation noise levels.

Figure 5: see original paper and (b) show the impact of translation noise (with fixed rotation noise $\sigma_R = 5^\circ$ and fixed pose count $n = 53$). EDPO remains faster than G-N and EIG while maintaining objective function values comparable to G-N, demonstrating overall superior performance.

Figure 6: see original paper and (b) illustrate the effect of increasing pose count (with fixed $\sigma_R = 5^\circ$, $\sigma_T = 0.1$). Problem scale increases have less impact on EDPO' s runtime compared to G-N and EIG, while maintaining objective function values comparable to G-N, verifying EDPO' s scalability for large-scale problems.

3.2 Algorithm Performance Comparison

We selected six public 3D SLAM datasets: sphere, torus, and cube (simulated); cubicle, rim, and garage (real-world). The sphere dataset simulates robot motion on a spherical surface; torus simulates motion on a toroidal surface; cube simulates walking on a 3D grid world. Garage is a real dataset from Vertigo; cubicle and rim are real datasets from the RIM center, with relative pose constraints obtained via ICP on point clouds from 3D laser scans. Dataset details including pose count (n) and measurement count (m) are summarized in .

Table 1: Dataset Information

Dataset	Poses (n)	Measurements (m)
Sphere	2500	4949
Torus	5000	9048
Cube	8000	23718
Cubicle	5750	16869
Rim	10100	36540
Garage	1661	6275

Table 2: Experimental Results on 3D SLAM Datasets

Dataset	Initial Cost	EDPO Cost	G-N Cost	EIG Cost	EDPO Time (s)	G-N Time (s)	EIG Time (s)	Min Eigen-value
Sphere	$1.291\times10^{[illegible]}$	[illegible]	[illegible] $\times10^{3}$	3.393×10^{3}	0.45	1.53	7.51	1.28×10^{-5}
Torus	$1.996\times10^{[illegible]}$	[illegible]	[illegible] $\times10^{4}$	2.475×10^{4}	[illegible]	[illegible]	[illegible]	[illegible]
Cube	$7.327\times10^{[illegible]}$	[illegible]	[illegible] $\times10^{4}$	8.733×10^{4}	[illegible]	[illegible]	[illegible]	[illegible]
Cubicle	$4.532\times10^{[illegible]}$	[illegible]	[illegible] $\times10^{4}$	4.147×10^{5}	[illegible]	[illegible]	[illegible]	[illegible]
Rim	$4.994\times10^{[illegible]}$	[illegible]	[illegible] $\times10^{4}$	7.227×10^{4}	1.29	1.53	1.42	9.43×10^{-6}
Garage	$8.36\times10^{[illegible]}$	[illegible]	[illegible] $\times10^{0}$	1.422×10^{0}	0.02	0.05	0.07	4.25×10^{-7}

Table 3: Trajectory Error Comparison on 3D SLAM Datasets

Dataset	Mean Error	Median Error	RMSE	Std Dev	Mean Error	Median Error	RMSE	Std Dev	Mean Error	Median Error	RMSE	Std Dev
	EDPO				**G-N**				**EIG**			
Sphere	0.12	0.10	0.15	0.08	0.11	0.09	0.14	0.07	0.18	0.15	0.22	0.12
Torus	0.08	0.07	0.09	0.04	0.08	0.07	0.09	0.04	0.12	0.10	0.14	0.08
Cube	0.15	0.13	0.18	0.09	0.15	0.13	0.18	0.09	0.21	0.18	0.25	0.13
Cubicle	0.22	0.19	0.26	0.12	0.22	0.19	0.26	0.12	0.31	0.27	0.35	0.18
Rim	0.18	0.16	0.21	0.10	0.18	0.16	0.21	0.10	0.25	0.22	0.29	0.15
Garage	0.05	0.04	0.06	0.03	0.05	0.04	0.06	0.03	0.08	0.07	0.09	0.05

Results show EDPO achieves comparable performance to G-N on torus, cube, cubicle, and garage datasets, with slightly worse but negligible differences on sphere and rim. Trajectory errors are lower than EIG and comparable to G-N across all datasets. EIG performs worse because it simultaneously optimizes rotation and translation, performing a single eigen decomposition to obtain four eigenvectors before projecting to the required form, resulting in suboptimal translation estimates.

EDPO' s solution speed is significantly faster than G-N and EIG because it performs direct global optimization with lower computational complexity, whereas G-N is an iterative search technique requiring repeated iterations until convergence. EIG simultaneously optimizes rotation and translation, expanding matrix dimensions and increasing computational complexity. Therefore, EDPO offers superior speed, making it ideal for low-noise applications where speed is critical and differences from G-N are negligible.

These results further demonstrate that EDPO provides an effective method for recovering approximate global optimal solutions for SLAM under real operating conditions, with clear computational advantages. Improved speed enhances PGO performance, and as pose graph dimensions increase, EDPO offers an effective balance between accuracy and computation time.

---

4 Conclusion

In PGO problems, maximum likelihood estimation can solve poses with high accuracy, but the involved optimization strategies are often cumbersome and require good initialization to achieve approximate global optimal solutions. This paper proposes a closed-form solution algorithm based on eigen decomposition that enables fast optimization and applies to large-scale SLAM problems. The algorithm consists of two parts: first optimizing the rotation component to obtain the optimal solution, then solving for translation. An improved model reduction method is introduced for this model, greatly improving computational efficiency without compromising accuracy. Experiments on simulated and real SLAM datasets demonstrate that EDPO can quickly recover approximate global optimal solutions for SLAM problems under reasonable operating conditions. Future work aims to integrate this algorithm into a complete SLAM system to further validate and enhance its practical performance.

References

[1] Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: toward the Robust-Perception age [J]. IEEE Trans on Robotics, 2016, 32 (6): 1309-1332.

[2] Gao Xingbo, Shi Xuhua, Ge Qunfeng, et al. A review of visual SLAM for dynamic object scenarios [J]. Robot, 2021, 43 (06): 733-750.

[3] Wang Yan, Suo Hansheng, Jia Guijin. Research on mobile robot SLAM for petrochemical hazardous accident rescue [J]. Control Engineering of China, 2018, 25 (2): 6.

[4] Li Chenyang, Peng Cheng, Zhang Zhenqian, et al. Localization and map construction method of agricultural robot integrating odometer information [J]. Trans of the Chinese Society of Agricultural Engineering, 2021, 37 (21): 16-23.

[5] Bazeille S, Filliat D. Combining odometry and visual loopclosure detection for consistent topo-metrical mapping [J]. RAIRO: Operations Research, 2010, 44 (4): 365-377.

[6] Liang Mingjie, Min Huaqing, Luo Ronghua. Graph-based SLAM: a survey [J]. Robot, 2013, 35 (4): 13.

[7] Grisetti G, Kyummerle R, Stachniss C, et al. A tutorial on graph-based SLAM [J]. IEEE Intelligent Transportation Systems Magazine, 2010, 2 (4): 31-43.

[8] Zhang Honghua, Liu xuan, Chen Fuhao, et al. Research and development of SLAM back-end optimization based on graph optimization [J]. Application Research of Computers, 2019 (1): 7.

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

[10] Olson E, Leonard J, Teller S. Fast iterative alignment of pose graphs with poor initial estimates [C]// Proc of IEEE International Conference on Robotics and Automation. [S. l.]: IEEE Press, 2006: 2262-2269.

[11] Grisetti G, Stachniss C, Grzonka S, et al. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent [C]// Proc of Robotics: Science and Systems. Boston: MIT Press, 2007.

[12] Dellaert F, Kaess M. Square Root SAM: Simultaneous localization and mapping via square root information smoothing [J]. The International Journal of Robotics Research, 2006, 25 (12): 1181-1203.

[13] Kaess M, Ranganathan A, Dellaert F. iSAM: Incremental smoothing and mapping [J]. IEEE Trans on Robotics, 2008, 24 (6): 1365-1378.

[14] Kaess M, Johannsson H, Roberts R, et al. iSAM2: Incremental smoothing and mapping using the Bayes tree [J]. The International Journal of Robotics Research, 2012, 31 (2): 216-235.

[15] Rosen D M, Kaess M, Leonard J J. RISE: An incremental trust-region method for robust online sparse least-squares estimation [J]. IEEE Trans on Robotics, 2014, 30 (5): 1091-1108.

[16] Kyummerle R, Grisetti G, Strasdat H, et al. g2o: A general framework for graph optimization [C]// Proc of IEEE International Conference on Robotics and Automation. [S. l.]: IEEE Press, 2011: 3607-3613.

[17] Carlone L, Aragues R, Castellanos J A, et al. A linear approximation for graph-based simultaneous localization and mapping [C]// Proc of Robotics: Science and Systems. Boston: MIT Press, 2012: 41-48.

[18] Carlone L, Tron R, Daniilidis K, et al. Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization [C]// Proc of IEEE International Conference on Robotics and Automation. [S. l.]: IEEE Press, 2015: 4597-4604.

[19] Nasiri S M, Moradi H, Hosseini R. A linear least square initialization method

for 3D pose graph optimization problem [C]// Proc of IEEE International Conference on Robotics and Automation. [S. l.]: IEEE Press, 2018: 2474-2479.

[20] Carlone L, Dellaert F. Duality-based verification techniques for 2D SLAM [C]// Proc of IEEE International Conference on Robotics and Automation. [S. l.]: IEEE Press, 2015: 4589-4596.

[21] Carlone L, Rosen D M, Calafiore G, et al. Lagrangian duality in 3D SLAM: verification techniques and optimal solutions [C]// Proc of IEEE International Conference on Intelligent Robots and Systems. Vancouver: IEEE Press, 2015: 125-132.

[22] Rosen D M, Carlone L, Bandeira A S, et al. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group [J]. The International Journal of Robotics Research, 2019, 38 (2-3): 95-125.

[23] Eriksson A, Olsson C, Kahl F, et al. Rotation averaging and strong duality [C]// Proc of the IEEE Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Press, 2018: 127-135.

[24] Singer A. Angular synchronization by eigenvectors and semidefinite programming [J]. Applied & Computational Harmonic Analysis, 2011, 30 (1): 20-36.

[25] Singer A, Shkolnisky Y. Three-dimensional structure determination from common lines in cryo-EM by eigenvectors and semidefinite programming [J]. SIAM Journal on Imaging Sciences, 2011, 4 (2): 543-572.

[26] Arrigoni F, Rossi B, Fusiello A. Spectral synchronization of multiple views in SE(3) [J]. SIAM Journal on Imaging Sciences, 2016, 9 (4): 1963-1990.

[27] Eriksson A, Olsson C, Kahl F, et al. Rotation averaging with the chordal distance: Global minimizers and strong duality [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2019, 43 (1): 256-268.

[28] Jbilou K. An Arnoldi based algorithm for large algebraic Riccati equations [J]. Applied Mathematics Letters, 2006, 19 (5): 437-444.