Verification Techniques for Pose Graph Optimization

Philippe Weingertner

AA273, Spring 2024

Contact: pweinger@stanford.edu

Table of Contents

Table of Contents

Problem Description

- **Focus:** Pose Graph Optimization (PGO)
- **Standard Approaches:**
  - Utilize nonlinear solvers
  - Commonly employ Gauss-Newton or Levenberg-Marquardt methods
  - Achieve locally optimal solutions
  - Do not guarantee the quality of the solution
- **Objective:** crucial for safety-critical applications
  - Establish a quality certificate for solutions
  - Determine the need for further solution refinement
- **Seminal Work:** Rosen, Carlone et al. in SE-Sync
- **Main Contribution:** Alternative approach using Riemannian optimization over a product of SO(3) manifolds

Pose Graph Optimization Problem

$$f^*_{\mathrm{ML}} = \min_{t_i, R_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \left\| t_j - t_i - R_i \tilde{t}_{ij} \right\|_2^2 + \frac{\omega_R^2}{2} \left\| R_j - R_i \tilde{R}_{ij} \right\|_F^2$$

$$f^*_{\mathrm{ML}} = \min_{t_i, r_i} \sum_{(i,j)\in\mathcal{E}} \omega_t^2 \left\| t_j - t_i - T_{ij} r_i \right\|_2^2 + \frac{\omega_R^2}{2} \left\| r_j - Q_{ij} r_i \right\|_2^2$$

$$\begin{cases} f^*_{\mathrm{ML}} = \min\limits_{x} \quad x^T Q x \\ \text{with } x \in \mathbb{R}^{12n},\, Q \in \mathbb{R}^{12n \times 12n} \\ \text{and Q}= \sum\limits_{(i,j)\in\mathcal{E}} \omega_t^2 B_{ij}^T B_{ij} + \frac{\omega_R^2}{2} C_{ij}^T C_{ij} \end{cases}$$

Problem Size reduction: down to $9n \times 9n$

$$f(x) = f(t,r) = \begin{bmatrix} t & r \end{bmatrix}^T \begin{bmatrix} Q_{tt} & Q_{tr} \\ Q_{rt} & Q_{rr} \end{bmatrix} \begin{bmatrix} t \\ r \end{bmatrix}$$

$$\frac{\partial f}{\partial t}(t,r) = 2Q_{tt}t + 2A_{rt}^T r = 0$$

$$t^* = -Q_{tt}^{-1} Q_{tr} r$$

$$\begin{cases} f_{\mathrm{ML}}^* = \min\limits_{r} \quad r^T Q_r r \\ \text{with } r \in \mathbb{R}^{9n}, Q_r \in \mathbb{R}^{9n \times 9n} \\ \text{and } Q_r = Q_{rr} - Q_{rt} Q_{tt}^{-1} Q_{tr} \end{cases}$$

Problem Size reduction: down to $3n \times 3n$

$$Q_r = \tilde{Q} \otimes I_3$$

$$r^T Q_r r = \text{Trace}\left(\tilde{Q} R^T R\right)$$

$$\begin{cases} f^*_{\text{ML}} = \min\limits_{R} \quad \text{Trace}\left(\tilde{Q} R^T R\right) \\ \text{with } R \in \text{SO}(3)^n, \tilde{Q} \in \text{Sym}(3n) \end{cases}$$

PGO Optimzation and its SDP relaxation

$$
\begin{cases}
f^*_{\text{ML}} = \min\limits_{R} \quad \text{Trace}\left(\tilde{Q} R^T R\right) \\
\text{with } R \in \text{SO}(3)^n, \tilde{Q} \in \text{Sym}(3n)
\end{cases}
$$

Drop the constraints $\text{rank}(X) = 3, \det(R_i) = 1$ to get a convex problem:

$$
\begin{cases}
f^*_{\text{SDP}} = \min\limits_{X \in \text{Sym}(3n)} \quad \text{Trace}\left(\tilde{Q} X\right) \\
\text{s.t. } X = \begin{bmatrix} I_3 & * & * \\ * & \ddots & * \\ * & * & I_3 \end{bmatrix} \succeq 0
\end{cases}
$$

Table of Contents

Method 1: Solve f_{SDP} and f_{ML} with a SDP solver

$$
\begin{cases}
f^*_{\mathrm{SDP}} = \min\limits_{X \in \mathrm{Sym}(3n)} \quad \mathrm{Trace}\left(\tilde{Q}X\right) \\
\text{s.t. } X = \begin{bmatrix} I_3 & * & * \\ * & \ddots & * \\ * & * & I_3 \end{bmatrix} \succeq 0
\end{cases}
$$

- Solve f_{SDP} with cvxpy and e.g. the SCS solver
- Solve f_{ML}:
  - Retrieve R from $X = R^T R$ with $R \in \mathrm{SO}(3)^n$
  - $X = U\Sigma V^T, R = \left[\sqrt{\sigma_1}V_1^T \quad \sqrt{\sigma_2}V_2^T \quad \sqrt{\sigma_3}V_3^T\right]$
  - $R = U_r \Sigma_r V_r^T, R_{\text{nearest rotation}} = U_r V_r^T$
  - Adjust if $\det\left(U_r V_r^T\right) < 0$

Method 2: Solve f_{ML} with Riemannian optimization

$$\begin{cases} f_{\text{ML}}^* = \min_{R} \quad \text{Trace}\left(\tilde{Q} R^T R\right) \\ \text{subject to } R \in \text{SO}(3)^n \end{cases}$$

- $\text{SO}(3)$ is a Riemannian manifold
- Solved as an unconstrained optimization problem with manopt

What is a manifold ?

- A manifold is a topological space that locally resembles Euclidean space near each point
- More formally, an n-dimensional manifold is a space where every point has a neighborhood that is homeomorphic to the Euclidean space $\mathbb{R}^n$
- Examples of manifolds include:
  - S^n: The n-dimensional sphere
  - $\mathbf{SO}(n)$: the space of all $n \times n$ orthogonal matrices with determinant 1
  - $\mathbf{SE}(n)$: the space of all rigid body transformations in n dimensions
  - $\mathbf{St}(n, p)$: Stiefel manifold, the space of all orthonormal p-frames (sets of p orthonormal vectors) in $\mathbb{R}^n$
  - $\mathbf{Sym}(n)$: Space of all $n \times n$ symmetric matrices
  - $\mathbf{SPD}(n)$: Space of all $n \times n$ symmetric positive definite matrices
- In robot state estimation, the variables of interest live on-manifold (e.g. poses $\in$ SE(3))

Example: Gradient Descent on a Sphere Manifold

- Updates are constrained to the surface of a sphere.

$$\mathbf{x}_{k+1} = \text{Retract}(\mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k))$$

- $\mathbf{x}_k$ is the current point on the sphere.
- α is the learning rate.
- $\nabla f(\mathbf{x}_k)$ is the gradient of the function at $\mathbf{x}_k$.
- Retract is a retraction operator that maps back to the sphere.

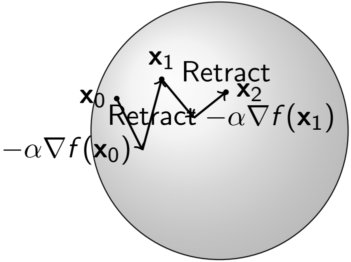
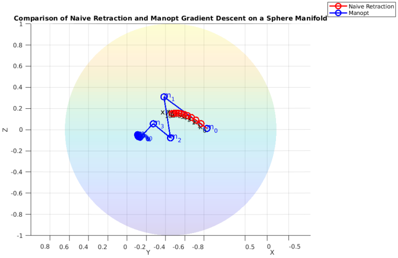


Figure 1: Two steps of gradient descent on a sphere manifold.

Naive Approach vs Riemaniann Optimization: Test 1



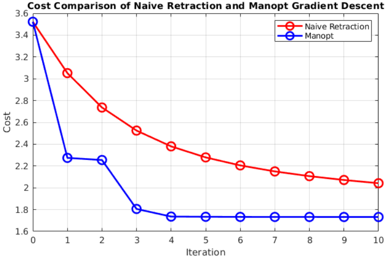
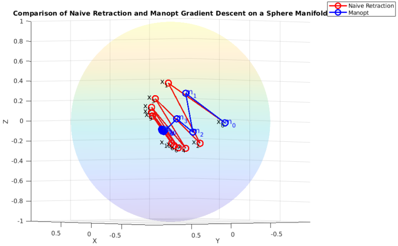


Figure 2: Optimize a Quadratic function constrained on a Sphere

Naive Approach vs Riemaniann Optimization: Test 2



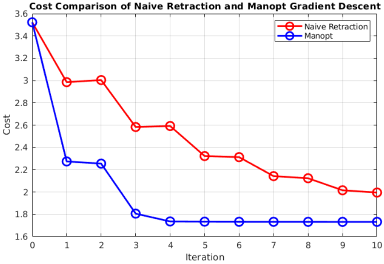


Figure 3: Optimize a Quadratic function constrained on a Sphere

- Much more information here: manopt learning

Method 3: Solve f_{SDP} with a SDP staircase algorithm

$$\begin{cases} f^*_{\text{SDP}} = \min\limits_{X \in \text{Sym}(3n)} \quad \text{Trace}\left(\tilde{Q}X\right) \\ \text{s.t. } X \succeq 0, \text{blocks diagonal} X_{ii} = I_3 \end{cases}$$

Why not use a low-rank factorization: $X = YY^T$?

$$\begin{array}{ll} \min\limits_{Y} & \text{Trace}\left(QYY^T\right) \\ s.t. & X = YY^T \\ & Y \in \mathbb{R}^{n \times p} \text{ leading to rank}\left(X\right) \leq p \end{array}$$

We are back to a non linear, non convex problem !!!

Method 3: Solve f_{SDP} with a Low-Rank Staircase Algorithm

Linear, Convex	Non-linear, Non-convex
$\min_X f(X)$	$\min_Y f\left(YY^T\right)$
$X \succeq 0,\ X_{ii} = I_d$	$Y \in \mathrm{Stiefel}(d, p)^n$

Theorem (Boumal et al. [BVB16], updated in 2018)

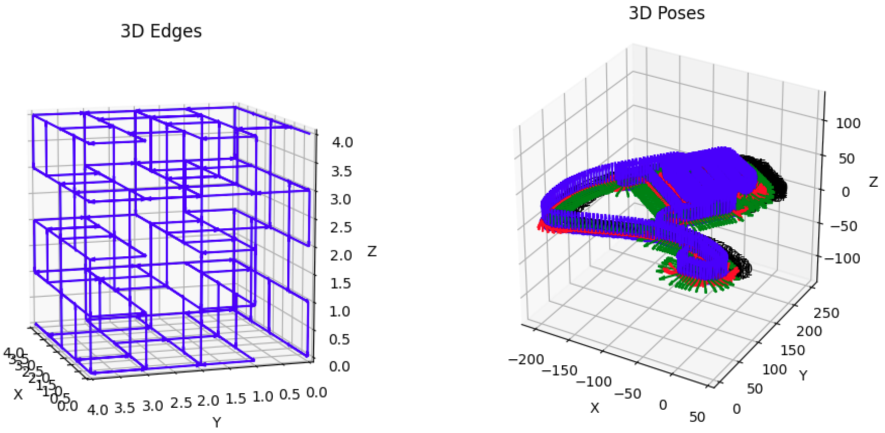
If Y is a local minimizer of the non-linear program and if Y is rank deficient or if Y is square, then $X = YY^T$ is a global minimizer of the initial convex program.

Method 3: Solving f_{SDP} with a low-rank staircase algorithm

- This leads to the Riemannian Staircase Algorithm
- It is guaranteed to return a globally optimal X

1. Set $p = d + 1$.
2. Compute Y_p, a Riemannian local optimizer.
3. If Y_p is rank deficient, stop.
4. Otherwise, increase p and go to step 2.

Table of Contents



(a) Grid3D N=4 test

(b) Parking-garage test

Figure 4: Grid3D N=4 and parking-garage tests

SE-sync Tests Results

Table 1: SE-Sync tests

Dataset	Sol.	$\hat{f}_{\mathrm{SDP}}$	$\hat{f}_{\mathrm{ML}}$	$\left\lvert \frac{\hat{f}_{\mathrm{SDP}} - \hat{f}_{\mathrm{ML}}}{\hat{f}_{\mathrm{SDP}}} \right\rvert$	Time
tinyGrid3D $n = 9$ $m = 11$	sdp scs	18.52	18.52	0	0.05 s
	riemann so(3)	-	18.52	0	0.08 s
	sdp staircase	18.52	18.52	0	0.03 s
	se-sync	18.52	18.52	0	0.13 s
smallGrid3D $n = 125$ $m = 297$	sdp scs	1025.4	1025.4	0	167 s
	riemann so(3)	-	1025.4	0	3.4 s
	sdp staircase	1025.4	1025.4	0	0.3 s
	se-sync	1025.4	1025.4	0	0.35 s
parking $n = 1661$ $m = 6275$	sdp scs	-	-	-	-
	riemann so(3)	-	369	282%	600 s time out
	sdp staircase	1.37	1.37	0	188 s
	se-sync	1.26	1.26	0	206 s

Table 2: Grid3D tests

Dataset	Sol.	$\hat{f}_{\mathrm{SDP}}$	$\hat{f}_{\mathrm{ML}}$	$\left\lvert \frac{\hat{f}_{\mathrm{SDP}} - \hat{f}_{\mathrm{ML}}}{\hat{f}_{\mathrm{SDP}}} \right\rvert$	Time
Grid3D N=2 $n = 27$ $m = 30$	sdp scs	1549	1549	0	4.6 s
	riemann so(3)	-	2158	39.3%	0.78 s
	sdp staircase	1549	1549	0	0.26 s
	se-sync	1549	1549	0	0.29 s
Grid3D N=4 $n = 125$ $m = 283$	sdp scs	65355	78523	20.1%	428.0 s
	riemann so(3)	-	**74597**	**14.1%**	7.15 s
	sdp staircase	65352	78534	20.1%	2.2 s
	se-sync	65352	78534	20.1%	2.1 s
Grid3D N=5 $n = 216$ $m = 482$	sdp scs	109424	136109	26.2%	600 s timeout
	riemann so(3)	-	**131774**	**22.1%**	22.1 s
	sdp staircase	107918	137259	27.1%	3.2 s
	se-sync	107947	137259	27.1%	3.1 s
Grid3D N=6 $n = 343$ $m = 783$	sdp scs	-	-	-	-
	riemann so(3)	-	**222550**	**22.1%**	22.1 s
	sdp staircase	190437	254583	16.9%	4.5 s
	se-sync	190400	254583	33.7%	4.4 s
Grid3D N=8 $n = 729$ $m = 1686$	sdp scs	-	-	-	-
	riemann so(3)	-	**469889**	**16.2%**	93.2 s
	sdp staircase	404610	541615	33.9%	24.4 s
	se-sync	404494	541590	33.9%	14.2 s
Grid3D N=9 $n = 1000$ $m = 2280$	sdp scs	-	-	-	-
	riemann so(3)	-	**618526**	**18.4%**	115.3 s
	sdp staircase	522279	707652	35.4%	26.7 s
	se-sync	522548	707672	35.4%	14.2 s

Algorithms Runtime

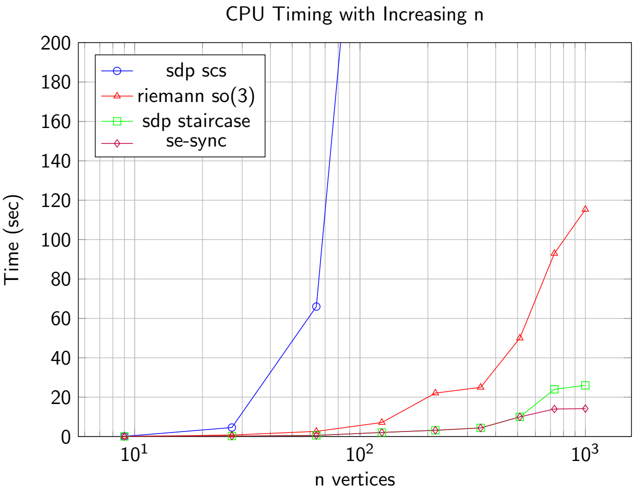


Figure 5: CPU Timing with Increasing n

Area of Improvements: Pre-Conditionner

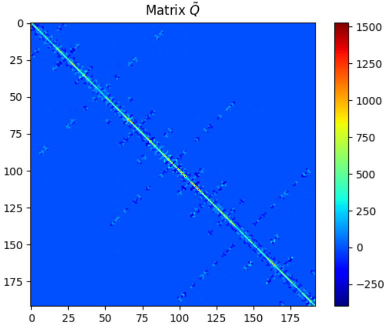


Figure 6: Matrix Sparsity for Grid3D N=3

Table of Contents

Conclusion

- Developed custom algorithm implementations and test sets for in-depth evaluation in SORI
- Benchmarked various methods for Pose Graph Optimization (PGO)
- **Method 1:** Does not scale well
- **Method 3:** Highly efficient, provides quality certificates
- **Method 2:** Riemannian optimization over SO(3) manifolds
  - Yielded promising results
  - Unexplored in the literature we reviewed
- **Future Work:** Investigate preconditioning techniques for faster convergence of Method 2

References I

Nicolas Boumal, *A riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints*, arXiv preprint arXiv:1506.00575 (2015).

Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira, *The non-convex burer-monteiro approach works on smooth semidefinite programs*, Advances in Neural Information Processing Systems **29** (2016).

David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard, *Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group*, The International Journal of Robotics Research **38** (2019), no. 2-3, 95–125.

Jie Wang and Liangbing Hu, *Solving low-rank semidefinite programs via manifold optimization*, arXiv preprint arXiv:2303.01722 (2023).