Message Passing Least Squares Framework and its Application to Rotation Synchronization

Yunpeng Shi [1] Gilad Lerman [1]

Abstract

We propose an efficient algorithm for solving group synchronization under high levels of corruption and noise, while we focus on rotation synchronization. We first describe our recent theoretically guaranteed message passing algorithm that estimates the corruption levels of the measured group ratios. We then propose a novel reweighted least squares method to estimate the group elements, where the weights are initialized and iteratively updated using the estimated corruption levels. We demonstrate the superior performance of our algorithm over state-of-the-art methods for rotation synchronization using both synthetic and real data.

1. Introduction

The problem of group synchronization is critical for various tasks in data science, including structure from motion (SfM), simultaneous localization and mapping (SLAM), Cryo-electron microscopy imaging, sensor network localization, multi-object matching and community detection. Rotation synchronization, also known as rotation averaging, is the most common group synchronization problems in 3D reconstruction. It asks to recover camera rotations from measured relative rotations between pairs of cameras. Permutation synchronization, which has applications in multi-object matching, asks to obtain globally consistent matches of objects from possibly erroneous measurements of matches between some pairs of objects. The simplest example of group synchronization is $\mathbb{Z}_2$ synchronization, which appears in community detection.

The general problem of group synchronization can be mathematically formulated as follows. Assume a graph $G([n],E)$ with n vertices indexed by $[n]=\{1,...,n\}$, a group $\mathcal{G}$, and set of group elements $\{g_i^*\}_{i=1}^n \subseteq \mathcal{G}$. The problem asks to recover $\{g_i^*\}_{i=1}^n$ from noisy and corrupted measurements $\{g_{ij}\}_{ij\in E}$ of the group ratios $\{g_i^* g_j^{*-1}\}_{ij\in E}$. We use the star superscript to emphasize the ground truth solution of group synchronization. We note that one can only recover, or approximate, the original group elements $\{g_i^*\}_{i\in[n]}$ up to a right group action. Indeed, for any $g_0\in\mathcal{G}$, g_{ij}^* can also be written as $g_i^* g_0 (g_j^* g_0)^{-1}$ and thus $\{g_i^* g_0\}_{i\in[n]}$ is also a solution. The above mentioned synchronization problems (rotation, permutation, and $\mathbb{Z}_2$ synchronization) correspond to the groups $SO(3)$, S_N, and $\mathbb{Z}_2$, respectively.

The most challenging issue for group synchronization is the practical scenario of highly corrupted and noisy measurements. Traditional least squares solvers often fail to produce accurate results in such a scenario. Moreover, some basic estimators that seem to be robust to corruption often do not tolerate in practice high level of noise. We aim to propose a general method for group synchronization that may tolerate high levels and different kinds of corruption and noise. While our basic ideas are formally general, in order to carefully refine and test them, we focus on the special problem of rotation synchronization, which is also known as rotation averaging (Hartley et al., 2013). We choose this problem as it is the most common, and possibly most difficult, synchronization problem in 3D computer vision.

1.1. Related Works

Most previous group synchronization solvers minimize an energy function. For the discrete groups $\mathbb{Z}_2$ and S_N, least squares energy minimization is commonly used. Relevant robustness results, under special corruption and noise models, are discussed in Abbe et al. (2014); Abbe (2017); Bandeira (2018); Chen et al. (2014); Huang & Guibas (2013); Huroyan (2018); Pachauri et al. (2013).

For Lie groups, such as $SO(D)$, that is, the group of $D\times D$ orthogonal matrices with determinant 1, where $D\geq 2$, least squares minimization was proposed to handle Gaussian noise (Bandeira et al., 2017; Eriksson et al., 2018; Govindu, 2004). However, when the measurements are also adversarially corrupted, this framework does not work well and other corruption-robust energy functions need to be used (Chatterjee & Govindu, 2013; 2018; Hartley et al., 2011; Maunu & Lerman, 2020; Wang & Singer, 2013). The most common corruption-robust energy function uses least absolute deviations. Wang & Singer (2013) prove that under a very special probabilistic setting with $\mathcal{G}=SO(D)$, the pure minimizer of this energy function can exactly recover the underlying group elements with high probability. However, their assumptions are strong and they use convex relaxation, which changes the original problem and is expensive to compute. Maunu & Lerman (2020) apply a trimmed averaging procedure for robustly solving $SO(2)$ synchronization. They are able to recover the ground truth group elements under a special deterministic condition on the topology of the corrupted subgraph. However, the verification of this condition and its extension to

[1]School of Mathematics, University of Minnesota, Minneapolis, MN, USA. Correspondence to: Yunpeng Shi<shixx517@umn.edu>.

Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

$SO(D)$, where $D>2$, are nontrivial. Hartley et al. (2011) used the Weiszfeld algorithm to minimize the least-absolute-deviations energy function with $\mathcal{G} = SO(3)$. Their method iteratively computes geodesic medians. However, they update only one rotation matrix per iteration, which results in slow empirical convergence and may increase the possibility of getting stuck at local minima. Chatterjee & Govindu (2018) proposed a robust Lie-algebraic averaging method over $\mathcal{G}=SO(3)$. They apply an iteratively reweighted least squares (IRLS) procedure in the tangent space of $SO(3)$ to optimize different robust energy functions, including the one that uses least absolute deviations. They claim that the use of the $\ell_{1/2}$ norm for deviations results in highest empirical accuracy. The empirical robustness of the two latter works is not theoretically guaranteed, even in simple settings. A recent deep learning method (Purkait et al., 2019) solves a supervised version of rotation synchronization, but it does not apply to the above unsupervised formulation of the problem.

Huang et al. (2017) use least absolute deviations minimization for solving 1D translation synchronization, where $\mathcal{G}=\mathbb{R}$ with addition. They propose a special version of IRLS and provide a deterministic exact recovery guarantee that depends on properties of the graph and its Laplacian. They do not explain their general result in an adversarial setting, but in a very special noisy setting.

Robustness results were established for least absolute deviations minimization in camera location estimation, which is somewhat similar to group synchronization (Hand et al., 2018; Lerman et al., 2018). These results assume special probabilistic setting, however, they have relatively weak assumptions on the corruption model.

Several energy minimization solutions have been proposed to $SE(3)$ synchronization (Birdal et al., 2018; Briales & Jiménez, 2017; Rosen et al., 2019; Arrigoni et al., 2016; 2018). This problem asks to jointly estimate camera rotations and locations from relative measurements of both. Neither of these solutions successfully address highly corrupted scenarios.

Other works on group synchronization, which do not minimize energy functions but aim to robustly recover corrupted solutions, screen corrupted edges using cycle consistency information. For a group $\mathcal{G}$ with group identity denoted by e, any $m \geq 3$, any cycle $L = \{i_1 i_2, i_2 i_3 \dots i_m i_1\}$ of length m and any corresponding product of ground-truth group ratios along L, $g_L^* = g_{i_1 i_2^*} g_{i_2 i_3^*} \cdots g_{i_m i_1^*}$, the cycle-consistency constraint is $g_L^* = e$. In practice, one is given the product of measurements, that is, $g_L = g_{i_1 i_2} g_{i_2 i_3} \cdots g_{i_m i_1}$, and in order to "approximately satisfy the cycle-consistency constraint" one tries to enforce g_L to be sufficiently close to e. Zach et al. (2010) uses the cycle-consistency constraint to detect corrupted relative rotations in $SO(3)$. It seeks to maximize a log likelihood function, which is based on the cycle-consistency constraint, using either belief propagation or convex relaxation. However, no theoretical guarantees are provided for the accuracy of outlier detection. Moreover, the log likelihood function implies very strong assumptions on the joint densities of the given relative rotations. Shen et al. (2016) classify the relative rotations as uncorrupted if they belong to any cycle that approximately satisfies the cycle-consistency constraint. However, this work only exploits local information and cannot handle the adversarial corruption case, where corrupted cycles can be approximately consistent.

An iterative reweighting strategy, IR-AAB (Shi & Lerman, 2018), was proposed to detect and remove corrupted pairwise directions for the different problem of camera location estimation. It utilizes another notion of cycle-consistency to infer the corruption level of each edge. Lerman & Shi (2019) extend the latter idea, and interpret it as a message passing procedure, to solve group synchronization with any compact group. They refer to their new procedure as cycle-edge message passing (CEMP). While We follow ideas of Lerman & Shi (2019); Shi & Lerman (2018), we directly solve for group elements, instead of estimating corruption levels, using them to initial cleaning of edges and solving the cleaner problem with another method.

To the best of our knowledge, the unified frameworks for group synchronization are Gao & Zhao (2019); Lerman & Shi (2019); Perry et al. (2018). However, Gao & Zhao (2019) and Perry et al. (2018) assume special probabilistic models that do not address adversarial corruption. Furthermore, Gao & Zhao (2019) only applies to Lie groups and the different setting of multi-frequencies.

1.2. Contribution of This Work

Current group synchronization solvers often do not perform well with highly corrupted and noisy group ratios. In order to address this issue, we propose a rotation synchronization solver that can address in practice high levels of noise and corruption. Our main ideas seem to generalize to group synchronization with any compact group, but more careful developments and testing are needed for other groups. We emphasize the following specific contributions of this work:

- We propose the message passing least squares (MPLS) framework as an alternative paradigm to IRLS for group synchronization, and in particular, rotation synchronization. It uses the theoretically guaranteed CEMP algorithm for estimating the underlying corruption levels. These estimates are then used for learning better weights for the weighted least squares problem.
- We explain in Section 3 why the common IRLS solver may not be accurate enough and in Section 4 why MPLS can overcome these obstacles.
- While MPLS can be formally applied to any compact group, we refine and test it for the group $\mathcal{G}=SO(3)$. We demonstrate state-of-the-art results for rotation synchronization with both synthetic data having nontrivial scenarios and real SfM data.

2. Setting for Robust Group Synchronization

Some previous robustness theories for group synchronization typically assume a very special and often unrealistic corruption probabilistic model for very special groups (Pachauri et al., 2013; Wang & Singer, 2013). In general, simplistic probabilistic models for corruption, such as generating potentially corrupted group ratios according to the Haar measure on $\mathcal{G}$ (Wang & Singer, 2013), may not generate some nontrivial scenarios that often occur in practice. For example, in the application of rotation synchronization that arise in SfM, the corrupted camera relative rotations can be self-consistent due to the ambiguous scene structures (Wilson & Snavely, 2014). However, in common probabilistic models, such as the one in Wang & Singer (2013), cycles with corrupted edges are self-consistent with probability zero. A more realistic model is the adversarial corruption model for the different problem of camera location (Lerman et al., 2018; Hand et al., 2018). However, it also assumes very special probabilistic models for the graph and camera locations, which are not realistic. A more general model of adversarial corruption with noise is due to Lerman & Shi (2019) and we review it here.

We assume a graph $G([n],E)$ and a compact group $\mathcal{G}$ with a bi-invariant metric d, that is, for any g_1, g_2, $g_3 \in \mathcal{G}$, $d_{\mathcal{G}}(g_1,g_2) = d_{\mathcal{G}}(g_3g_1,g_3g_2) = d_{\mathcal{G}}(g_1g_3,g_2g_3)$. For $\mathcal{G} = SO(3)$, or any Lie group, d is commonly chosen to be the geodesic distance. Since $\mathcal{G}$ is compact, we can scale d and assume that $d(\cdot) \leq 1$.

We partition E into E_g and E_b, which represent sets of good (uncorrupted) and bad (corrupted) edges, respectively. We will need a topological assumption on E_b, or equivalently, E_g. A necessary assumption is that $G([n],E_g)$ is connected, though further restrictions on E_b may be needed for establishing theoretical guarantees (Lerman & Shi, 2019).

In the noiseless case, the adversarial corruption model generates group ratios in the following way.

$$g_{ij} = \begin{cases} g_{ij}^* := g_i^* g_j^{*-1}, & ij \in E_g; \\ \tilde{g}_{ij} \neq g_{ij}^*, & ij \in E_b. \end{cases} \tag{1}$$

That is, for edges $ij \in E_b$, the corrupted group ratio $\tilde{g}_{ij} \neq g_{ij}^*$ can be arbitrarily chosen from $\mathcal{G}$. The corruption is called adversarial since one can maliciously corrupt the group ratios for $ij \in E_b$ and also maliciously choose E_b as long as the needed assumptions on E_b are satisfied. One can even form cycle-consistent corrupted edges, so that they can be confused with the good edges.

In the noisy case, we assume a noise model for $d(g_{ij},g_{ij}^*)$, where $ij \in E_g$. In theory, one may need to restrict this model (Lerman & Shi, 2019), but in practice we test highly noisy scenarios.

For $ij \in E$ we define the corruption level of ij as

$$s_{ij}^* = d(g_{ij},g_{ij}^*).$$

We use ideas of Lerman & Shi (2019) to estimate $\{s_{ij}^*\}_{ij\in E}$, but then we propose new ideas to estimate $\{g_i^*\}_{i\in[n]}$. While exact estimation of both quantities is equivalent in the noiseless case (Lerman & Shi, 2019), this property is not valid when adding noise.

3. Issues with the Common IRLS

We first review the least squares minimization, least absolute and unsquared deviations minimization and IRLS for group synchronization. We then explain why IRLS may not form a good solution for the group synchronization problem, and in particular for Lie algebraic groups, such as the rotation group.

The least squares minimization can be formulated as follows:

$$\min_{\{g_i\}_{i=1}^n \subseteq \mathcal{G}} \sum_{ij\in E} d^2(g_{ij},g_i g_j^{-1}), \tag{2}$$

where one often relaxes this formulation. This formulation is generally sensitive to outliers and thus more robust energy functions are commonly used when considering corrupted group ratios. More specifically, one may choose a special function $\rho(x) \neq x^2$ and solve the following least unsquared deviation formulation

$$\min_{\{g_i\}_{i=1}^n \subseteq \mathcal{G}} \sum_{ij\in E} \rho\big(d(g_{ij},g_i g_j^{-1})\big). \tag{3}$$

The special case of $\rho(x) = x$ (Hand et al., 2018; Hartley et al., 2011; Ozyesil & Singer, 2015; Wang & Singer, 2013) is referred to as least absolute deviations. Some other common choices are $\rho(x) = x^2/(x^2+\sigma^2)$ (Chatterjee & Govindu, 2013) and $\rho(x) = \sqrt{x}$ (Chatterjee & Govindu, 2018).

The least unsquared formulation is typically solved using IRLS, where at iteration t one solves the weighted least squares problem:

$$\{g_{i,t}\}_{i\in[n]} = \operatorname*{argmin}_{\{g_i\}_{i=1}^n \subseteq \mathcal{G}} \sum_{ij\in E} w_{ij,t} d^2(g_{ij},g_i g_j^{-1}). \tag{4}$$

In the first iteration the weights can be initialized in a certain way, but in the next iterations the weights are updated using the residuals of this solution. Specifically, for $ij \in E$ and iteration t, the residual is $r_{ij,t} = d(g_{ij},g_{i,t}g_{j,t}^{-1})$ and the weight $w_{ij,t}$ is

$$w_{ij,t} = F(r_{ij,t-1}), \tag{5}$$

where the function F depends on the choice of ρ. For $\rho(x) = x^p$, where $0 < p < 2$, $F(x) = \min\{x^{p-2}, A\}$, where $1/A$ is a regularization parameter and here we fix $A = 10^8$.

The above IRLS procedure poses the following three issues. First, its convergence to the solution $\{g_i^*\}_{i\in[n]}$ is not guaranteed, especially under severe corruption. Indeed, IRLS succeeds when it accurately estimates the correct weights $w_{ij,t}$ for each edge. Ideally, when the solution $\{g_{i,t}\}_{i\in[n]}$ is close to the ground truth $\{g_i^*\}_{i\in[n]}$, the residual $r_{ij,t}$ must be close to the corruption level s_{ij}^* so that weight $w_{ij,t}$ must be close to $F(s_{ij}^*)$. However, if edge $ij \in E_b$ is severely corrupted (or edge $ij \in E_g$ has high noise) and either g_i^* or g_j^* is wrongly estimated, then the residual $r_{ij,t}$ might have a very small value. Thus the weight $w_{ij,t}$ in (5) can be extremely large and may result in an inaccurate solution in the next iteration and possibly low-quality solution at the last iteration.

The second issue is that for common groups each iteration of (4) requires either SDP relaxation or tangent space approximation (for Lie groups). However, if the weights of IRLS are wrongly estimated in the beginning, then they may affect the tightness of the SDP relaxation and the validity of tangent space approximation. Therefore, such procedures tend to make the IRLS scheme sensitive to corruption and initialization of weights and group elements.

At last, when dealing with noisy data where most of s_{ij}^*, $ij \in E$, are significantly greater than 0, the current reweighting strategy usually gives non-negligible positive weights to outliers. This can be concluded from the expression of F (e.g., for ℓ_p minimization) and the fact that in a good scenario $r_{ij,t} \approx s_{ij}^*$ and s_{ij}^* can be away from 0. Therefore, outliers can be overweighed and this may lead to low-quality solutions. We remark that this issue is more noticeable in Lie groups, such as the rotation group, as all measurements are often noisy and corrupted; whereas in discrete groups some measurements may be rather accurate (Shi et al., 2020).

4. Message Passing Least Squares (MPLS)

In view of the drawbacks of the common IRLS scheme, we propose the MPLS (Message Passing Least Squares), or Minneapolis, algorithm. It carefully initializes and reevaluates the weights of a weighted least squares problem by our CEMP algorithm (Lerman & Shi, 2019) or a modified version of it. We first review the ideas of CEMP in Section 4.1. We remark that its goal is to estimate the corruption levels $\{s_{ij}^*\}_{ij\in E}$ and not the group elements $\{g_i^*\}_{i\in[n]}$. Section 4.2 formally describes MPLS for the general setting of group synchronization. Section 4.3 carefully refines MPLS for rotation synchronization. Section 4.4 summarizes the complexity of the proposed algorithms.

4.1. Cycle-Edge Message Passing (CEMP)

The CEMP procedure aims to estimate the corruption levels $\{s_{ij}^*\}_{ij\in E}$ from the cycle inconsistencies, which we define next. For simplicity and for ease of computation, we work here with 3-cycles, that is, triangles in the graph. For each edge $ij\in E$, we independently sample with replacement 50 nodes that form 3-cycles with i and j. That is, if k is such a node then ik and jk are in E. We denote this set of nodes by C_{ij}. We remark that the original version of CEMP in Lerman & Shi (2019) uses all 3-cycles and can be slower. We define the cycle inconsistency of the 3-cycle ijk associated with edge ij and $k\in C_{ij}$ as follows

$$d_{ij,k} := d(g_{ij}g_{jk}g_{ki}, e_{\mathcal{G}}), \quad k\in C_{ij}, ij\in E. \tag{6}$$

The idea of CEMP is to iteratively estimate each corruption level s_{ij}^* for $ij\in E$ from a weighted average of the cycle inconsistencies $\{d_{ij,k}\}_{k\in[n]}$. To motivate this idea, we assume the noiseless adversarial corruption model and formulate the following proposition whose easy proof appears in Lerman & Shi (2019).

Proposition 4.1. *If $s_{ik}^*=s_{jk}^*=0$, that is, ik, $jk\in E_g$, then* $s_{ij}^*=d_{ij,k}$.

If the condition of the above proposition holds, we call ijk a good cycle with respect to ij, otherwise we call it a bad cycle.

CEMP also aims to estimate the conditional probability $p_{ij,k}^t$ that the cycle ijk is good, so $s_{ij}^*=d_{ij,k}$. The conditioning is on the estimates of corruption levels computed in the previous iteration. CEMP uses this probability as a weight for the cycle inconsistency $d_{ij,k}$. The whole weighted sum thus aims to estimate the conditional expectation of s_{ij}^* at iteration t. This estimate is denoted by $s_{ij,t}$. The iteration of CEMP thus contains two main steps: 1) Computation of the weight $p_{ij,k}^t$ of $d_{ij,k}$; 2) Computation of $s_{ij,t}$ as a weighted average of the cycle inconsistencies $\{d_{ij,k}\}_{k\in[n]}$. In the former stage messages are passed from edges to cycles and in the latter stage messages are passed from cycles to edges. The simple procedure is summarized in Algorithm 1. We formulate it in generality for a compact group $\mathcal{G}$ with a bi-invariant metric d and a graph $G([n],E)$, for which the cycle inconsistencies, $\{d_{ij,k}\}_{ij\in E,k\in C_{ij}}$, were computed in advance. Our default parameters are later specified in Section 5.1. For generality, we write $|C_{ij}|$ instead of 50.

Algorithm 1 CEMP (Lerman & Shi, 2019)

```
Input: $\{d_{ij,k}\}_{ij\in E,k\in C_{ij}}$, time step $T$, increasing $\{\beta_t\}_{t=0}^T$
  Steps:
  $s_{ij,0}=\frac{1}{|C_{ij}|}\sum_{k\in C_{ij}} d_{ij,k}$                          $ij\in E$
  for $t=0:T$ do
    Message passing from edges to cycles:
    $p_{ij,k}^t=\exp(-\beta_t(s_{ik,t}+s_{jk,t}))$                 $k\in C_{ij}, ij\in E$
    Message passing from cycles to edges:
    $s_{ij,t+1}=\frac{1}{Z_{ij,t}}\sum_{k\in C_{ij}} p_{ij,k}^t d_{ij,k}$,                      $ij\in E$
    where $Z_{ij,t}=\sum_{k\in C_{ij}} p_{ij,k}^t$ is a normalization factor
  end for
Output: $\{s_{ij,T}\}_{ij\in E}$
```

Algorithm 1 can be explained in two different ways (Lerman & Shi, 2019). First of all, it can be theoretically guaranteed to be robust to adversarial corruption and stable to low level of noise (see Theorem 5.4 in Lerman & Shi (2019)). Second of all, our above heuristic explanation can be made more rigorous using some statistical assumptions. Such assumptions are common in other message passing algorithms in statistical mechanics formulations and we thus find it important to motivate them here. We remark that our statistical assumptions are weaker than those of previous works on message passing (Donoho et al., 2009; Yedidia et al., 2003), and in particular, those for group synchronization (Perry et al., 2018; Zach et al., 2010). We also remark that they are not needed for establishing the above mentioned theory.

Our first assumption is that ijk is a good cycle if and only if $s_{ij}^*=d_{ij,k}$. Proposition 4.1 implies the only if part, but the other part is generally not true. However, under special random corruption models (e.g., models in Wang & Singer (2013); Ozyesil & Singer (2015)), the assumed equivalence holds with probability 1. We further assume that $\{s_{ij}^*\}_{ij\in E}$ and $\{s_{ij,t}\}_{ij\in E}$ are both i.i.d. random variables and that for any $ij\in E$, s_{ij}^* is independent of $s_{kl,t}$ for $kl\neq ij\in E$. We further assume that for any $ij\in E$

$$\Pr(s_{ij}^*=0|s_{ij,t}=x)=\exp(-\beta_t x). \tag{7}$$

We also assume the existence of good cycle ijk for any $ij\in E$.

In view of these assumptions, in particular the i.i.d. sampling and (7), we obtain that the expression for $p_{ij,k}^t$ used in Algorithm 1 coincides with the conditional probability that ijk is a good cycle, that is, the conditional probability that $s_{ik}^*=s_{jk}^*=0$:

$$\begin{aligned} p_{ij,k}^t &= \Pr(s_{ik}^*=s_{jk}^*=0|\{s_{ab,t}\}_{ab\in E}) \\ &= \Pr(s_{ik}^*=0|s_{ik,t})\Pr(s_{jk}^*=0|s_{jk,t}) \\ &= \exp(-\beta_t(s_{ik,t}+s_{jk,t})). \end{aligned} \tag{8}$$

Using the definition of conditional expectation, the equivalence assumption, the above i.i.d. sampling assumptions and (7), we show that the expression for $s_{ij,k}^t$ used in Algorithm 1 coincides with the conditional expectation of s_{ij}^*:

$$\begin{aligned} &\mathbb{E}\big(s_{ij}^*|\{s_{ab,t}\}_{ab\in E}\big) \\ &= \frac{1}{Z_{ij,t}}\sum_{k\in C_{ij}} \Pr\big(s_{ij}^*=d_{ij,k}|\{s_{ab,t}\}_{ab\in E}\big) d_{ij,k} \end{aligned}$$

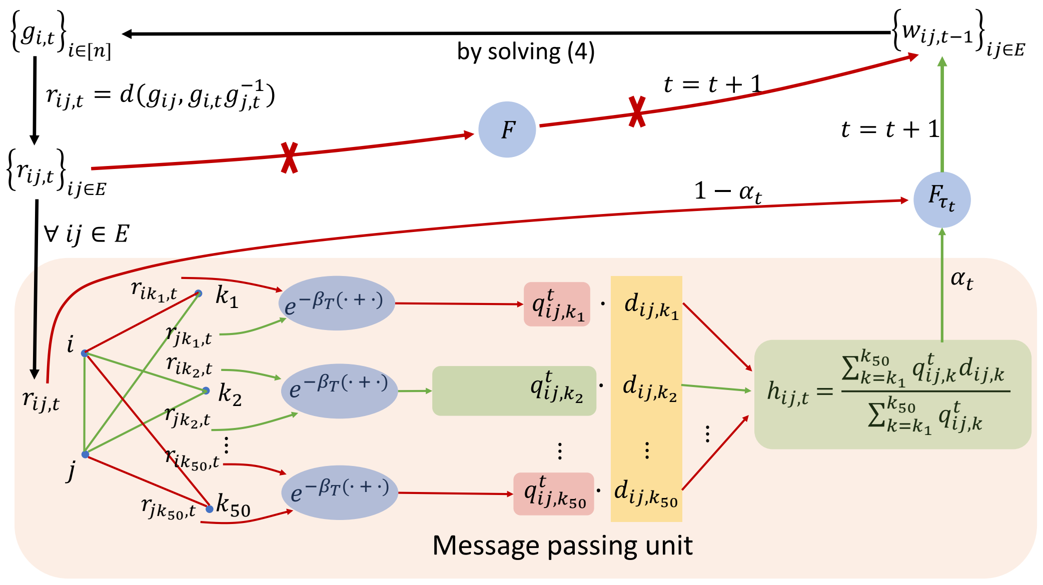

Figure 1. Demonstration of MPLS in comparison with IRLS. IRLS updates the graph weights directly from the residuals, which can be inaccurate. It is demonstrated in the upper loop of the figure, where the part different than MPLS is crossed out. In contrast, MPLS updates the graph weights by applying a CEMP-like procedure to the residuals, demonstrated in the "message-passing unit". Good edges, such as jk_1, are marked with green, and bad edges are marked with red. For $ij \in E$ and $k \in \{k_1, k_2, ... k_{50}\}$, $q^t_{ij,k}$ is updated using the two residuals $r_{ik,t}$ and $r_{jk,t}$ according to the indicated operation. The length of a bar around the computed value of each $q^t_{ij,k}$ is proportional to magnitude and the green or red colors designate good or bad corresponding cycles, respectively. The weighted sum $h_{ij,t}$ aims to approximate s^*_{ij} and this approximation is good when the green $q^t_{ij,k}$ bars are much longer than the red bars. The weight $w_{ij,t}$ is formed as a convex combination of $r_{ij,t}$ and $h_{ij,t}$. The rest of the procedure is similar to IRLS.

$$\begin{aligned}
&= \frac{1}{Z_{ij,t}} \sum_{k\in C_{ij}} \Pr\big(s^*_{ik} = s^*_{jk} = 0 \big| \{s_{ab,t}\}_{ab\in E}\big) d_{ij,k} \\
&= \frac{1}{Z_{ij,t}} \sum_{k\in C_{ij}} \exp(-\beta_t(s_{ik,t} + s_{jk,t})) d_{ij,k} \qquad (9) \\
&= \frac{1}{Z_{ij,t}} \sum_{k\in C_{ij}} p^t_{ij,k} d_{ij,k}.
\end{aligned}$$

Note that our earlier motivation of Algorithm 1 assumed both (8) and (9). Demonstration of a procedure similar to CEMP, but with different notation, appears in the lower part of Figure 1.

4.2. General Formulation of MPLS

MPLS uses the basic ideas of CEMP in order to robustly estimate the residuals $\{r_{ij,t}\}_{ij\in E}$ and weights $\{w_{ij,t}\}_{ij\in E}$ of the IRLS scheme as well as to carefully initialize this scheme. It also incorporates a novel truncation idea. We explain in this and the next section how these new ideas address the drawbacks of the common IRLS procedure, which was reviewed in Section 3. We sketch MPLS in Algorithm 2, demonstrate it in Figure 1 and explain it below. For generality, we assume in this section a compact group $\mathcal{G}$, a bi-invariant metric d on $\mathcal{G}$ and a graph $G([n], E)$ with given relative measurements and cycle inconsistencies computed in advance. Our default parameters are later specified in Section 5.1.

Instead of using the traditional IRLS reweighting function $F(x)$ explained in Section 3, we use its truncated version $F_\tau(x) = F(x)\mathbf{1}_{\{x\le\tau\}} + 10^{-8}\mathbf{1}_{\{x>\tau\}}$ with a parameter $\tau > 0$. We decrease τ as the iteration number increases in order to avoid overweighing outliers. By doing this we aim to address the third drawback of IRLS mentioned in Section 3. We remark that the truncated function is $F(x)\mathbf{1}_{\{x\le\tau\}}$ and the additional term $10^{-8}\mathbf{1}_{\{x>\tau\}}$ is needed to ensure that the graph with weights resulting from F_τ is connected.

Algorithm 2 Message Passing Least Squares (MPLS)

```
Input: $\{g_{ij}\}_{ij\in E}$, $\{d_{ij,k}\}_{k\in C_{ij}}$, nonincreasing $\{\tau_t\}_{t\ge0}$,
  increasing $\{\beta_t\}_{t=0}^T$, decreasing $\{\alpha_t\}_{t\ge1}$
  Steps:
  Compute $\{s_{ij,T}\}_{ij\in E}$ by CEMP
  $w_{ij,0} = F_{\tau_0}(s_{ij,T})$                                   $ij \in E$
  $t = 0$
  while not convergent do
    $t = t+1$
    $\{g_{i,t}\}_{i\in[n]} = \operatorname{argmin}_{g_i\in\mathcal{G}} \sum_{ij\in E} w_{ij,t-1} d^2(g_{ij}, g_i g_j^{-1})$
    $r_{ij,t} = d(g_{ij}, g_{i,t} g_{j,t}^{-1})$                          $ij \in E$
    $q^t_{ij,k} = \exp(-\beta_T(r_{ik,t} + r_{jk,t}))$           $k \in C_{ij}, ij \in E$
    $h_{ij,t} = \frac{\sum_{k\in C_{ij}} q^t_{ij,k} d_{ij,k}}{\sum_{k\in C_{ij}} q^t_{ij,k}}$                       $ij \in E$
    $w_{ij,t} = F_{\tau_t}(\alpha_t h_{ij,t} + (1-\alpha_t) r_{ij,t})$          $ij \in E$
  end while
Output: $\{g_{i,t}\}_{i\in[n]}$
```

The initial step of the algorithm estimates the corruption levels $\{s_{ij}^*\}_{ij\in E}$ by CEMP. The initial weights for the IRLS procedure follow (5) with additional truncation. At each iteration, the group ratios $\{g_{i,t}\}_{i\in[n]}$ are estimated from the weighted least squares procedure in (4). However, the weights $w_{ij,t}$ are updated in a very different way. First of all, for each $ij\in E$ the corruption level s_{ij}^* is re-estimated in two different ways and a convex combination of the two estimates is taken. The first estimate is a residual $r_{ij,t}$ computed with the newly updated estimates $\{g_{i,t}\}_{i\in[n]}$. This is the error of approximating the given measurement g_{ij} by the newly estimated group ratio. The other estimate practically applies CEMP to re-estimate the corruption levels. For edge $ij\in E$, the latter estimate of s_{ij}^* is denoted by $h_{ij,t}$. For interpretation, we can replace (7) with $\Pr(s_{ij}^*|r_{ij,t})=\exp(-\beta_T x)$ and use it to derive analogs of (8) and (9). Unlike CEMP, we use the single parameter, β_T, as we assume that CEMP provides a sufficiently good initialization. At last, a similar weight as in (5), but truncated, is applied to the combined estimate $\alpha_t h_{ij,t}+(1-\alpha_t)r_{ij,t}$.

We remark that utilizing the estimate $h_{ij,t}$ for the corruption level addresses the first drawback of IRLS discussed in Section 3. Indeed, assume the case where $ij\in E_b$ and $r_{ij,t-1}$ is close to 0. Here, $w_{ij,t}$ computed by IRLS is relatively large; however, since $ij\in E_b$, $w_{ij,t}$ needs to be small. Unlike $r_{ij,t-1}$ in IRLS, we expect that $h_{ij,t}$ in MPLS should not be too small as long as for some $k\in C_{ij}$, $d_{ij,k}$ are sufficiently large. This happens as long as there exists some $k\in C_{ij}$ for which the cycle ijk is good. Indeed, in this case s_{ij}^* is sufficiently large and for good cycles $d_{ij,k}=s_{ij}^*$.

We further remark that $h_{ij,t}$ is a good approximation of s_{ij}^* under certain conditions. For example, if for all $k\in C_{ij}$, $r_{ik,t-1}\approx s_{ik}^*$ and $r_{jk,t-1}\approx s_{jk}^*$, then plugging in the definition of $p_{ij,k}^t$ to the expression of $h_{ij,t}$, using the fact that β_T is sufficiently large and at last applying Proposition 4.1, we obtain that

$$\begin{aligned} h_{ij,t} &= \sum_{k\in C_{ij}} \frac{\exp(-\beta_T(r_{ik,t-1}+r_{jk,t-1}))}{\sum_{k\in C_{ij}}\exp(-\beta_T(r_{ik,t-1}+r_{jk,t-1}))} d_{ij,k} \\ &\approx \sum_{k\in C_{ij}} \frac{\exp(-\beta_T(s_{ik}^*+s_{jk}^*))}{\sum_{k\in C_{ij}}\exp(-\beta_T(s_{ik}^*+s_{jk}^*))} d_{ij,k} \qquad (10) \\ &\approx \sum_{k\in C_{ij}} \frac{\mathbf{1}_{\{ijk \text{ is a good cycle}\}}}{\sum_{k\in C_{ij}}\mathbf{1}_{\{ijk \text{ is a good cycle}\}}} d_{ij,k} = s_{ij}^*. \end{aligned}$$

This intuitive argument for a restricted case conveys the idea that "local good information" can be used to estimate s_{ij}^*. The theory of CEMP (Lerman & Shi, 2019) shows that under weaker conditions such information can propagate through the whole graph within a few iterations, but we cannot extend it to MPLS.

If the graph $G([n],E)$ is dense with sufficiently many good cycles, then we expect that this good information can propagate in few iterations and that $h_{ij,t}$ will have a significant advantage over $r_{ij,t}$. However, in real scenarios of rotation synchronization in SfM, one may encounter sparse graphs, which may not have enough cycles and, in particular, not enough good cycles. In this case, utilizing $h_{ij,t}$ is mainly useful in the early iterations of the algorithm. On the other hand, when $\{g_{i,t}\}_{i\in[n]}$ are close to $\{g_i^*\}_{i\in[n]}$, $\{r_{ij,t}\}_{i\in[n]}$ will be sufficiently close to $\{s_{ij}^*\}_{i\in[n]}$. Aiming to address rotation synchronization, we decrease α_t, the weight of $h_{ij,t}$, with t. In other applications, different choices of α_t can be used (Shi et al., 2020).

The second drawback of IRLS, discussed in Section 3, is the possible difficulty of implementing the weighted least squares step of (4). This issue is application-dependent, and since in this work we focus on rotation synchronization (equivalently, $SO(3)$ synchronization), we show in the next subsection how MPLS can deal with the above issue in this specific problem. Nevertheless, we claim that our framework can also be applied to other compact group synchronization problems and we demonstrate this claim in a follow up work (Shi et al., 2020).

4.3. MPLS for $SO(3)$ synchronization

Rotation synchronization, or $SO(3)$ synchronization, aims to solve 3D rotations $\{\boldsymbol{R}_i^*\}_{i\in[n]} \in SO(3)$ from measurements $\{\boldsymbol{R}_{ij}\}_{ij\in E} \in SO(3)$ of the 3D relative rotations $\{\boldsymbol{R}_i^*\boldsymbol{R}_j^{*-1}\}_{ij\in E} \in SO(3)$. Throughout the rest of the paper, we use the following normalized geodesic distance for $\boldsymbol{R}_1,\boldsymbol{R}_2\in SO(3)$:

$$d(\boldsymbol{R}_1,\boldsymbol{R}_2)=\|\log(\boldsymbol{R}_1\boldsymbol{R}_2^{-1})\|_F/(\sqrt{2}\pi), \qquad (11)$$

where log is the matrix logarithm and the normalization factor ensures that the diameter of $SO(3)$ is 1. We provide some relevant preliminaries of the Riemannian geometry of $SO(3)$ in Section 4.3.1 and then describe the implementation of MPLS for $SO(3)$, which we refer to as MPLS-$SO(3)$, in Section 4.3.2.

4.3.1. Preliminaries: $SO(3)$ and $\mathfrak{so}(3)$

We note that $SO(3)$ is a Lie group, and its corresponding Lie algebra, $\mathfrak{so}(3)$, is the space of all skew symmetric matrices, which is isomorphic to $\mathbb{R}^3$. For each $\boldsymbol{R}\in SO(3)$, its corresponding element in $\mathfrak{so}(3)$ is $\boldsymbol{\Omega}=\log(\boldsymbol{R})$, where log denotes matrix logarithm. Each $\boldsymbol{\Omega}\in\mathfrak{so}(3)$ can be represented as $[\boldsymbol{\omega}]_\times$ for some $\boldsymbol{\omega}=(\omega_1,\omega_2,\omega_3)^T\in\mathbb{R}^3$ in the following way:

$$[\boldsymbol{\omega}]_\times := \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}.$$

In other words, we can map any $\boldsymbol{\omega}\in\mathbb{R}^3$ to $\boldsymbol{\Omega}=[\boldsymbol{\omega}]_\times\in\mathfrak{so}(3)$ and $\boldsymbol{R}=\exp([\boldsymbol{\omega}]_\times)\in SO(3)$, where exp denotes the matrix exponential function. We remark that geometrically $\boldsymbol{\omega}$ is the tangent vector at $\boldsymbol{I}$ of the geodesic path from $\boldsymbol{I}$ to $\boldsymbol{R}$.

4.3.2. Details of MPLS-$SO(3)$

We note that in order to adapt MPLS to the group $SO(3)$, we only need a specific algorithm to solve the following formulation of the weighted least squares problem at iteration t

$$\begin{aligned} &\min_{\boldsymbol{R}_{i,t}\in SO(3)} \sum_{ij\in E} w_{ij,t} d^2(\boldsymbol{R}_{ij},\boldsymbol{R}_{i,t}\boldsymbol{R}_{j,t}^{-1}) \\ =&\min_{\boldsymbol{R}_{i,t}\in SO(3)} \sum_{ij\in E} w_{ij,t} d^2(\boldsymbol{I},\boldsymbol{R}_{i,t}^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_{j,t}), \qquad (12) \end{aligned}$$

where the last equality follows from the bi-invariance of d. The constraints on orthogonality and determinant of $\boldsymbol{R}_i$ are non-convex. If one relaxes those constraints, with an appropriate choice of the metric d, then the solution of the least squares problem in the relaxed Euclidean space often lies away from the embedding of $SO(3)$ into that space. For this reason, we follow the common choice of d according to (11) and implement the Lie-algebraic Averaging (LAA) procedure (Govindu, 2004; Chatterjee & Govindu, 2013; 2018; Tron et al., 2008). We review LAA, explain why it may be problematic and why our overall implementation may overcome its problems. LAA aims to move from $\boldsymbol{R}_{i,t}$ to $\boldsymbol{R}_{i,t+1}$ along the manifold using the right group action $\boldsymbol{R}_{i,t} = \boldsymbol{R}_{i,t-1}\Delta\boldsymbol{R}_{i,t}$, where $\Delta\boldsymbol{R}_{i,t} \in SO(3)$. For this purpose, it defines $\Delta\boldsymbol{R}_{ij,t} = \boldsymbol{R}_{i,t-1}^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_{j,t-1}$ so that

$$(\Delta\boldsymbol{R}_{i,t})^{-1}\Delta\boldsymbol{R}_{ij,t}\Delta\boldsymbol{R}_{j,t} = (\Delta\boldsymbol{R}_{i,t})^{-1}\boldsymbol{R}_{i,t-1}^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_{j,t-1}\Delta\boldsymbol{R}_{j,t} = \boldsymbol{R}_{i,t}^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_{j,t}$$

and (12) can be transformed to the still hard to solve equation

$$\min_{\Delta\boldsymbol{R}_{i,t}\in SO(3)} \sum_{ij\in E} w_{ij,t} d^2(\boldsymbol{I},(\Delta\boldsymbol{R}_{i,t})^{-1}\Delta\boldsymbol{R}_{ij,t}\Delta\boldsymbol{R}_{j,t}). \quad (13)$$

LAA then maps $\{\Delta\boldsymbol{R}_{i,t}\}_{i\in[n]}$ and $\{\Delta\boldsymbol{R}_{ij,t}\}_{ij\in E}$ to the tangent space of $\boldsymbol{I}$ by $\Delta\boldsymbol{\Omega}_{i,t} = \log\Delta\boldsymbol{R}_{i,t}$ and $\Delta\boldsymbol{\Omega}_{ij,t} = \log\Delta\boldsymbol{R}_{ij,t}$. Applying (11) and the fact that the Riemannian logarithmic map, which is represented by log, preserves the geodesic distance and using a "naive approximation": $d(\boldsymbol{I},(\Delta\boldsymbol{R}_{i,t})^{-1}\Delta\boldsymbol{R}_{ij,t}\Delta\boldsymbol{R}_{j,t})$. Therefore, LAA uses the following approximation

$$\begin{aligned} d(\boldsymbol{I},(\Delta\boldsymbol{R}_{i,t})^{-1}\Delta\boldsymbol{R}_{ij,t}\Delta\boldsymbol{R}_{j,t}) &= \\ \|\log((\Delta\boldsymbol{R}_{i,t})^{-1}\Delta\boldsymbol{R}_{ij,t}\Delta\boldsymbol{R}_{j,t})\|_F/(\sqrt{2}\pi) &\approx \\ \|-\log(\Delta\boldsymbol{R}_{i,t})+\log(\Delta\boldsymbol{R}_{ij,t})+\log(\Delta\boldsymbol{R}_{j,t}))\|_F/(\sqrt{2}\pi) &= \\ \|\Delta\boldsymbol{\Omega}_{i,t}-\Delta\boldsymbol{\Omega}_{j,t}-\Delta\boldsymbol{\Omega}_{ij,t}\|_F/(\sqrt{2}\pi). \quad (14) \end{aligned}$$

Consequently, LAA transforms (13) as follows:

$$\min_{\Delta\boldsymbol{\Omega}_{i,t}\in\mathfrak{so}(3)} \sum_{ij\in E} w_{ij,t}\|\Delta\boldsymbol{\Omega}_{i,t}-\Delta\boldsymbol{\Omega}_{j,t}-\Delta\boldsymbol{\Omega}_{ij,t}\|_F^2. \quad (15)$$

However, the approximation in (14) is only valid when $\Delta\boldsymbol{R}_{ij,t}$, $\Delta\boldsymbol{R}_{i,t}$, $\Delta\boldsymbol{R}_{j,t} \approx \boldsymbol{I}$, which is unrealistic.

One can check that the following conditions: $\boldsymbol{R}_{ij} \approx \boldsymbol{R}_i^*\boldsymbol{R}_j^{*-1}$ ($s_{ij}^* \approx 0$), $\boldsymbol{R}_{i,t} \approx \boldsymbol{R}_i^*$ and $\boldsymbol{R}_{j,t} \approx \boldsymbol{R}_j^*$ for $t \geq 0$ imply that $\Delta\boldsymbol{R}_{ij,t}$, $\Delta\boldsymbol{R}_{i,t}$, $\Delta\boldsymbol{R}_{j,t} \approx \boldsymbol{I}$ and thus imply (14). Therefore, to make LAA work we need to give large weights to edges ij with small s_{ij}^* and provide a good initialization $\{\boldsymbol{R}_{i,0}\}_{i\in[n]}$ that is reasonably close to $\{\boldsymbol{R}_i^*\}_{i\in[n]}$ and so that $\{\boldsymbol{R}_{i,t}\}_{i\in[n]}$ for all $t \geq 1$ are still close to the ground truth. Our heuristic argument is that good approximation by CEMP, followed by MPLS, addresses these requirements. Indeed, to address the first requirement, we note that good initialization by CEMP can result in $s_{ij,T} \approx s_{ij}^*$ and by the nature of F, $w_{ij,0}$ is large when $s_{ij,T}$ is small. As for the second requirement, we assign the weights $s_{ij,T}$, obtained by CEMP, to each $ij \in E$ and find the minimal spanning tree for the weighted graph by Prim's algorithm. We initialize the rotations by fixing $\boldsymbol{R}_{1,0} = \boldsymbol{I}$, multiplying relative rotations along the computed minimal spanning tree and consequently obtaining $\boldsymbol{R}_{i,0}$ for any node i. We summarize our MPLS version of rotation averaging in Algorithm 3.

4.4. Computational Complexity

CEMP requires the computation of $d_{ij,k}$ for $ij \in E$ and $k \in C_{ij}$. Its computational complexity per iteration is thus of order $O(|E|)$

Algorithm 3 MPLS-$SO(3)$

```
Input: $\{\boldsymbol{R}_{ij}\}_{ij\in E}$, $\{d_{ij,k}\}_{k\in C_{ij}}$, $\{\tau_t\}_{t\geq 0}$, $\{\beta_t\}_{t=0}^T$, $\{\alpha_t\}_{t\geq 1}$
  Steps:
  Compute $\{s_{ij,T}\}_{ij\in E}$ by CEMP
  Form an $n\times n$ weight matrix $\boldsymbol{W}$, where $W_{ij}=W_{ji}=s_{ij,T}$ for $ij\in E$, and $W_{ij}=W_{ji}=0$ otherwise
  $G([n],E_{ST})$ = minimal spanning tree of $G([n],W)$
  $\boldsymbol{R}_{1,0}=\boldsymbol{I}$
  find $\{\boldsymbol{R}_{i,0}\}_{i>1}$ by $\boldsymbol{R}_i=\boldsymbol{R}_{ij}\boldsymbol{R}_j$ for $ij\in E_{ST}$
  $t=0$
  $w_{ij,0}=F_{\tau_0}(s_{ij,T})$
  while not convergent do
    $t=t+1$
    $\Delta\boldsymbol{\Omega}_{ij,t}=\log(\boldsymbol{R}_{i,t-1}^{-1}\boldsymbol{R}_{ij}\boldsymbol{R}_{j,t-1})$    $ij\in E$
    $\{\Delta\boldsymbol{\Omega}_{i,t}\}_{i\in[n]}=$
        $\operatorname{argmin}_{\Delta\boldsymbol{\Omega}_{i,t}\in\mathfrak{so}(3)} \sum_{ij\in E} w_{ij,t}\|\Delta\boldsymbol{\Omega}_{i,t}-\Delta\boldsymbol{\Omega}_{j,t}-\Delta\boldsymbol{\Omega}_{ij,t}\|_F^2$
    $\boldsymbol{R}_{i,t}=\boldsymbol{R}_{i,t-1}\exp(\Delta\boldsymbol{\Omega}_{i,t})$    $i\in[n]$
    $r_{ij,t}=\|\Delta\boldsymbol{\Omega}_{i,t}-\Delta\boldsymbol{\Omega}_{j,t}-\Delta\boldsymbol{\Omega}_{ij,t}\|_F/(\sqrt{2}\pi)$    $ij\in E$
    $q_{ij,k}^t=\exp(-\beta_T(r_{ik,t}+r_{jk,t}))$    $k\in C_{ij}, ij\in E$
    $h_{ij,t}=\frac{\sum_{k\in C_{ij}} q_{ij,k}^t d_{ij,k}}{\sum_{k\in C_{ij}} q_{ij,k}^t}$    $ij\in E$
    $w_{ij,t}=F_{\tau_t}(\alpha_t h_{ij,t}+(1-\alpha_t)r_{ij,t})$    $ij\in E$
  end while
Output: $\{\boldsymbol{R}_{i,t}\}_{i\in[n]}$
```

as we use $|C_{ij}| = 50$ for all $ij \in E$. Since we advocate few iterations ($T = 5$) of CEMP, or due to its fast convergence under special settings (Lerman & Shi, 2019), we can assume that its total complexity is $O(|E|)$. The computational complexity of MPLS depends on the complexity of solving the weighted least squares problem, which depends on the group. For MPLS-$SO(3)$, the most expensive part is solving the weighted least squares problem in the tangent space, whose complexity is at most $O(n^3)$. This is thus also the complexity of MPLS-$SO(3)$ per iteration. Unlike CEMP, we have no convergence guarantees yet for MPLS.

5. Numerical Experiments

We test the proposed MPLS algorithm on rotation synchronization, while comparing with state-of-the-art methods. We also try simpler ideas than MPLS that are based on the basic strategy of CEMP. All computational tasks were implemented on a machine with 2.5GHz Intel i5 quad core processors and 8GB memory.

5.1. Implementation

We use the following default parameters for Algorithm 1: $|C_{ij}| = 50$ for $ij \in E$; $T = 5$; $\beta_t = 2^t$ and $t = 0,...,5$. If an edge is not contained in any 3-cycle, we set its corruption level as 1. For MPLS-$SO(3)$, which we refer to in this section as MPLS, we use the above parameters of Algorithm 1 and the following ones for $t \geq 1$:

$$\alpha_t = 1/(t+1) \text{ and } \tau_t = \inf_x \left\{\hat{P}_t(x) > \max\{1-0.05t, 0.8\}\right\}.$$

Here, $\hat{P}_t$ denotes the empirical distribution of $\{\alpha_t h_{ij,t} + (1-\alpha_t) r_{ij,t}\}_{ij\in E}$. That is, for $t=0$, 1, 2, 3, we ignore 0%, 5%, 10%, 15% of edges that have highest $\alpha_t h_{ij,t} + (1-\alpha_t) r_{ij,t}$, and for $t \geq 4$ we ignore 20% of such edges. $F(x)$ for MPLS is chosen as $x^{-3/2}$ and it corresponds to $\rho(x) = \sqrt{x}$. For simplicity and consistency, we use these choices of parameters for all of our experiments. We remark that our choice of β_t in Algorithm 1 is supported by the theory of Lerman & Shi (2019). We found that MPLS is not so sensitive to its parameters. One can choose other values of $\{\beta_t\}_{t\geq 0}$, for example any geometric sequence with ratio 2 or less, and stop after several iterations. Similarly, one may replace 0.8 and 0.05 in the definition of τ_t with $0.7-0.9$ and $0.01-0.1$, respectively, and perform similarly on average.

We test two previous state-of-the-art IRLS methods: IRLS-GM (Chatterjee & Govindu, 2013) with $\rho(x) = x^2/(x^2+25)$, $F(x) = 25/(x^2+25)^2$ and IRLS-$\ell_{1/2}$ (Chatterjee & Govindu, 2018) with $\rho(x) = \sqrt{x}$, $F(x) = x^{-3/2}$. We use their implementation by Chatterjee & Govindu (2018).

We have also separately implemented the part of initializing the rotations of MPLS, and refer to it by CEMP in this section. Recall that it solves rotations by direct propagation along the minimal weighted spanning tree of the graph with weights obtained by Algorithm 1 (note that we previously referred by CEMP only to the latter algorithm). We also test the application of this initialization to the main algorithms in Chatterjee & Govindu (2013) and Chatterjee & Govindu (2018) and refer to the resulting methods by CEMP+IRLS-GM and CEMP+IRLS-$\ell_{1/2}$, respectively. We remark that the original algorithms initialize by a careful least absolute deviations minimization. We use the convergence criterion $\sum_{i\in[n]} \|\Delta\boldsymbol{\Omega}_{i,t}\|_F/(\sqrt{2}n) < 0.001$ of Chatterjee & Govindu (2018) for all the above algorithms.

Because the solution is determined up to a right group action, we align our estimated rotations $\{\hat{\boldsymbol{R}}_i\}$ with the ground truth ones $\{\boldsymbol{R}_i^*\}$. That is, we find a rotation matrix $\boldsymbol{R}_{\text{align}}$ so that $\sum_{i\in[n]} \|\hat{\boldsymbol{R}}_i \boldsymbol{R}_{\text{align}} - \boldsymbol{R}_i^*\|_F^2$ is minimized. For synthetic data, we report the following mean estimation error in degrees: $180 \cdot \sum_{i\in[n]} d(\hat{\boldsymbol{R}}_i \boldsymbol{R}_{\text{align}}, \boldsymbol{R}_i^*)/n$. For real data, we also report the median of $\{180 \cdot d(\hat{\boldsymbol{R}}_i \boldsymbol{R}_{\text{align}}, \boldsymbol{R}_i^*)\}_{i\in[n]}$.

5.2. Synthetic Settings

We test the methods in the following two types of artificial scenarios. In both scenarios, the graph is generated by the Erdős-Rényi model $G(n,p)$ with $n=200$ and $p=0.5$.

5.2.1. Uniform Corruption

We consider the following random model for generating $\boldsymbol{R}_{ij}$:

$$\boldsymbol{R}_{ij} = \begin{cases} \text{Proj}(\boldsymbol{R}_{ij}^* + \sigma \boldsymbol{W}_{ij}), & \text{w.p. } 1-q; \\ \tilde{\boldsymbol{R}}_{ij} \sim \text{Haar(SO(3))}, & \text{w.p. } q, \end{cases} \quad (16)$$

where Proj denotes the projection onto $SO(3)$; $\boldsymbol{W}_{ij}$ is a 3×3 Wigner matrix whose elements follow i.i.d. standard normal distribution; $\sigma \geq 0$ is a fixed noise level; q is the probability that an edge is corrupted and Haar$(SO(3))$ is the Haar probability measure on $SO(3)$. We clarify that for any 3×3 matrix $\boldsymbol{A}$, $\text{Proj}(\boldsymbol{A}) = \text{argmin}_{\boldsymbol{R}\in SO(3)} \|\boldsymbol{R} - \boldsymbol{A}\|_F$.

We test the algorithms with four values of σ : 0, 0.1, 0.5, and 1. We average the mean error over 10 random samples from the uniform model and report it as a function of q in Figure 2.

We note that MPLS consistently outperforms the other methods for all tested values of q and σ. In the noiseless case, MPLS exactly recovers the group ratios even when 70% of the edges are corrupted. It also nearly recovers with 80% corrupted edges, where the estimation errors for IRLS-GM and IRLS-$\ell_{1/2}$ are higher than 30 degrees. MPLS is also shown to be stable under high level of noise. Since all algorithms produce poor solutions when $q=0.9$, we only show results for $0 \leq q \leq 0.8$.

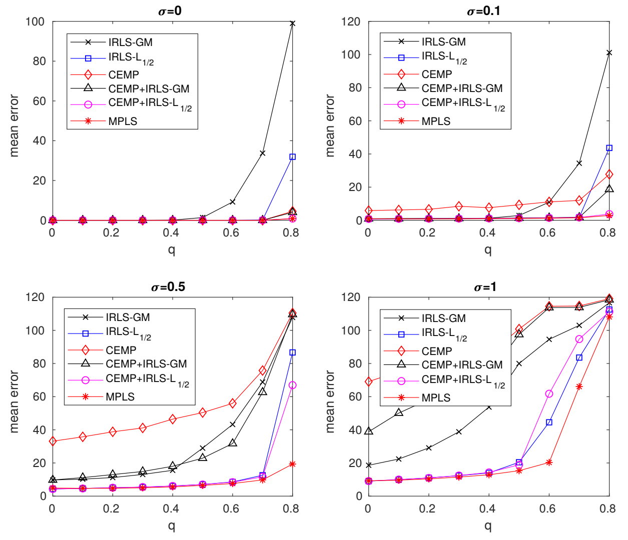

Figure 2. Performance under uniform corruption. The mean error (in degrees) is plotted against the corruption probability q for 4 values of σ.

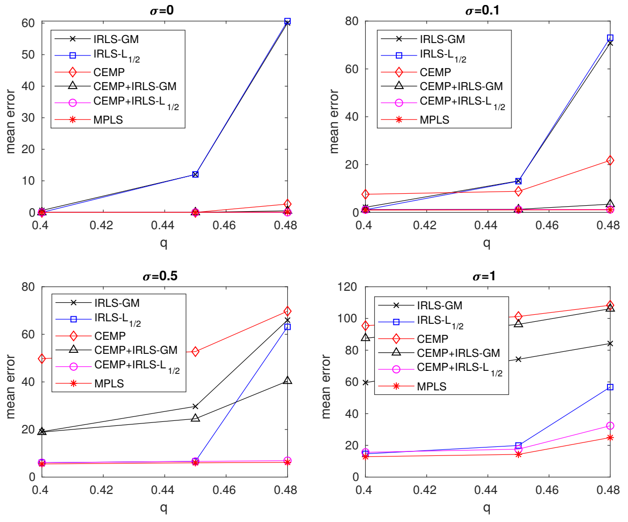

Figure 3. Performance under self-consistent corruption. The mean error is plotted against the corruption probability q for 4 values of σ.

Algorithms			IRLS-GM				IRLS-$\ell_{\frac{1}{2}}$				CEMP				MPLS			
Dataset	n	m	$\tilde{e}$	$\hat{e}$	runtime	iter #	$\tilde{e}$	$\hat{e}$	runtime	iter #	$\tilde{e}$	$\hat{e}$	runtime	iter #	$\tilde{e}$	$\hat{e}$	runtime	iter #
Alamo	564	71237	3.64	1.30	14.2	10+8	3.67	1.32	15.5	10+9	4.05	1.62	**10.38**	6	**3.44**	**1.16**	20.6	6+8
Ellis Island	223	17309	3.04	1.06	3.2	10+9	2.71	0.93	2.8	10+13	2.94	1.11	**2.4**	6	**2.61**	**0.88**	4.0	6+11
Gendarmenmarkt	655	32815	**39.24**	**7.07**	6.5	10+14	39.41	7.12	7.3	10+19	45.33	8.62	**4.7**	6	44.94	9.87	17.8	6+25
Madrid Metropolis	315	14903	5.30	1.78	3.8	10+30	4.88	1.88	2.7	10+12	5.10	1.66	**2.1**	6	**4.65**	**1.26**	5.2	6+23
Montreal N.D.	442	44501	1.25	0.58	6.5	10+6	1.22	0.57	7.3	10+8	1.33	0.79	**6.3**	6	**1.04**	**0.51**	9.3	6+7
Notre Dame	547	88577	2.63	0.78	17.2	10+7	2.26	0.71	22.5	10+10	2.35	0.94	**13.2**	6	**2.06**	**0.67**	31.5	6+8
NYC Library	307	13814	2.71	1.37	2.5	10+14	2.66	1.30	2.6	10+15	3.00	1.41	**1.9**	6	**2.63**	**1.24**	4.5	6+14
Piazza Del Popolo	306	18915	4.10	2.17	2.8	10+9	3.99	2.09	3.1	10+13	**3.44**	**1.57**	**2.6**	6	3.73	1.93	3.5	6+3
Piccadilly	2031	186458	5.12	2.02	153.5	10+16	5.19	2.34	170.2	10+19	4.66	1.98	**45.8**	6	**3.93**	**1.81**	191.9	6+21
Roman Forum	989	41836	2.66	1.58	8.6	10+9	2.69	1.57	11.4	10+17	2.80	1.45	**6.1**	6	**2.62**	**1.37**	8.8	6+8
Tower of London	440	15918	3.42	2.52	2.6	10+8	3.41	2.50	2.4	10+12	**2.84**	**1.57**	**2.2**	6	3.16	2.20	2.7	6+7
Union Square	680	17528	6.77	3.66	5.0	10+32	6.77	3.85	5.6	10+47	7.47	3.64	**2.5**	6	**6.54**	**3.48**	5.7	6+21
Vienna Cathedral	770	87876	8.13	1.92	28.3	10+13	8.07	**1.76**	45.4	10+23	**6.91**	2.63	**13.1**	6	7.21	2.83	42.6	6 +19
Yorkminster	410	20298	2.60	1.59	**2.4**	10+7	**2.45**	1.53	3.3	10+9	2.49	**1.37**	2.8	6	2.47	1.45	3.9	6+7

Table 1. Performance on the Photo Tourism datasets: n and m are the number of nodes and edges, respectively; $\tilde{e}$ and $\hat{e}$ indicate mean and median errors in degrees, respectively.; runtime is in seconds; and numbers of iterations (explained in the main text).

5.2.2. Self-Consistent Corruption

In order to simulate self-consistent corruption, we independently draw from Haar($SO(3)$) two classes of rotations: $\{\boldsymbol{R}_i^*\}_{i\in[n]}$ and $\{\tilde{\boldsymbol{R}}_i\}_{i\in[n]}$. We denote their corresponding relative rotations by $\boldsymbol{R}_{ij}^* = \boldsymbol{R}_i^*\boldsymbol{R}_j^{*\intercal}$ and $\tilde{\boldsymbol{R}}_{ij} = \tilde{\boldsymbol{R}}_i\tilde{\boldsymbol{R}}_j^\intercal$ for $ij \in E$. The idea is to assign to edges in E_g and E_b relative rotations from two different classes, so cycle-consistency occurs in both $G([n],E_g)$ and $G([n],E_b)$. We also add noise to these relative rotations and assign them with Bernoulli model to the two classes, so one class is more significant. More specifically, for $ij \in E$

$$\boldsymbol{R}_{ij} = \begin{cases} \text{Proj}(\boldsymbol{R}_{ij}^* + \sigma\boldsymbol{W}_{ij}), & \text{w.p. } 1-q; \\ \text{Proj}(\tilde{\boldsymbol{R}}_{ij} + \sigma\boldsymbol{W}_{ij}), & \text{w.p. } q, \end{cases} \tag{17}$$

where q, σ, and $\boldsymbol{W}_{ij}$ are the same as in the above uniform corruption model. We remark that an information-theoretic threshold for the exact recovery when $\sigma = 0$ is $q = 0.5$. That is, for $q \geq 0.5$ there is no hope of exactly recovering $\{\boldsymbol{R}_i^*\}_{i\in[n]}$.

We test the algorithms with four values of σ : 0, 0.1, 0.5, and 1. We average the mean error over 10 random samples from the self-consistent model and report it as a function of q in Figure 3. We focus on values of q approaching the information-theoretic bound 0.5 (q = 0.4, 0.45 and 0.48). We note that MPLS consistently outperforms the other algorithm and that when $\sigma = 0$ it can exactly recover the ground truth rotations when $q = 0.48$.

5.3. Real Data

We compare the performance of the different algorithms on the Photo Tourism datasets (Wilson & Snavely, 2014). Each of the 14 datasets consists of hundreds of 2D images of a 3D scene taken by cameras with different orientations and locations. For each pair of images of the same scene, we use the pipeline proposed by Sengupta et al. (2017) to estimate the relative 3D rotations. The ground truth camera orientations are also provided. Table 1 compares the performance of IRLS-GM, IRLS-$\ell_{1/2}$, CEMP and MPLS, while reporting mean and median errors, runtime and number of iterations. The number of iterations is the sum of the number of iterations to initialize the rotations and the number of iterations of the rest of the algorithm, where CEMP only has iterations in the initialization step.

MPLS achieves the lowest mean and median error on 9 out of 14 datasets with runtime comparable to both IRLS, while IRLS-GM only outperforms MPLS on the Gendarmenmarkt dataset. This dataset is relatively sparse and lacks cycle information. It contains a large amount of self-consistent corruption and none of the methods solve it reasonably well. Among the tested 4 methods, the fastest approach is CEMP. It achieves shortest runtime on 13 out of 14 dataset. Moreover, CEMP is 3 times faster than other tested methods on the largest dataset (Piccadilly). We remark that CEMP is able to achieve comparable results to common IRLS on most datasets, and has superior performance on 2 datasets, which have some perfectly estimated edges. In summary, for most of the datasets, MPLS provides the highest accuracy and CEMP obtains the fastest runtime.

6. Conclusion

We proposed a framework for solving group synchronization under high corruption and noise. This general framework requires a successful solution of the weighted least squares problem, which depends on the group. For $SO(3)$, we explained how a well-known solution integrates well with our framework. We demonstrated state-of-the-art performance of our framework for $SO(3)$ synchronization. We have motivated our method as an alternative to IRLS and explained how it may overcome the limitations of IRLS when applied to group synchronization.

There are many directions to expand our work. One can carefully adapt and implement our proposed framework to other groups that occur in practice. One may develop certain theoretical guarantees for convergence and exact recovery of MPLS.

Acknowledgement

This work was supported by NSF award DMS-18-21266. We thank Tyler Maunu for his valuable comments on an earlier version of this manuscript.

References

Abbe, E. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Abbe, E., Bandeira, A. S., Bracher, A., and Singer, A. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *IEEE Trans. Network Science and Engineering*, 1(1):10–22, 2014.

Arrigoni, F., Rossi, B., and Fusiello, A. Spectral synchronization of multiple views in SE(3). *SIAM J. Imaging Sciences*, 9(4): 1963–1990, 2016.

Arrigoni, F., Rossi, B., Fragneto, P., and Fusiello, A. Robust synchronization in SO(3) and SE(3) via low-rank and sparse matrix decomposition. *Comput. Vis. Image Underst.*, 174: 95–113, 2018.

Bandeira, A. S. Random laplacian matrices and convex relaxations. *Foundations of Computational Mathematics*, 18 (2):345–379, 2018. doi: 10.1007/s10208-016-9341-9.

Bandeira, A. S., Boumal, N., and Singer, A. Tightness of the maximum likelihood semidefinite relaxation for angular synchronization. *Mathematical Programming*, 163(1-2): 145–167, 2017.

Birdal, T., Simsekli, U., Eken, M. O., and Ilic, S. Bayesian pose graph optimization via bingham distributions and tempered geodesic MCMC. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems (NeurIPS)*, pp. 306–317, 2018.

Briales, J. and Jiménez, J. G. Cartan-sync: Fast and global se(d)-synchronization. *IEEE Robotics Autom. Lett.*, 2(4): 2127–2134, 2017.

Chatterjee, A. and Govindu, V. M. Efficient and robust large-scale rotation averaging. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pp. 521–528, 2013.

Chatterjee, A. and Govindu, V. M. Robust relative rotation averaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4): 958–972, 2018. doi: 10.1109/TPAMI.2017.2693984.

Chen, Y., Guibas, L. J., and Huang, Q. Near-optimal joint object matching via convex relaxation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 100–108, 2014.

Donoho, D. L., Maleki, A., and Montanari, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009. ISSN 0027-8424. doi: 10.1073/pnas.0909892106.

Eriksson, A. P., Olsson, C., Kahl, F., and Chin, T. Rotation averaging and strong duality. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22*, pp. 127–135. IEEE Computer Society, 2018.

Gao, T. and Zhao, Z. Multi-frequency phase synchronization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 2132–2141, 2019.

Govindu, V. M. Lie-algebraic averaging for globally consistent motion estimation. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 27 June - 2 July 2004, Washington, DC, USA*, pp. 684–691, 2004.

Hand, P., Lee, C., and Voroninski, V. Shapefit: Exact location recovery from corrupted pairwise directions. *Communications on Pure and Applied Mathematics*, 71(1):3–50, 2018.

Hartley, R. I., Aftab, K., and Trumpf, J. L1 rotation averaging using the weiszfeld algorithm. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 3041–3048, 2011.

Hartley, R. I., Trumpf, J., Dai, Y., and Li, H. Rotation averaging. *Int. J. Comput. Vis.*, 103(3):267–305, 2013. doi: 10.1007/s11263-012-0601-0.

Huang, Q. and Guibas, L. J. Consistent shape maps via semidefinite programming. *Comput. Graph. Forum*, 32(5): 177–186, 2013.

Huang, X., Liang, Z., Bajaj, C., and Huang, Q. Translation synchronization via truncated least squares. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 1459–1468, 2017.

Huroyan, V. *Mathematical Formulations, Algorithm and Theory for Big Data Problems*. PhD thesis, University of Minnesota, 2018.

Lerman, G. and Shi, Y. Robust group synchronization via cycle-edge message passing. *arXiv preprint arXiv:1912.11347*, 2019.

Lerman, G., Shi, Y., and Zhang, T. Exact camera location recovery by least unsquared deviations. *SIAM J. Imaging Sciences*, 11(4):2692–2721, 2018. doi: 10.1137/17M115061X.

Maunu, T. and Lerman, G. A provably robust multiple rotation averaging scheme for SO(2). *arXiv preprint arXiv:2002.05299*, 2020.

Ozyesil, O. and Singer, A. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2674–2683, 2015.

Pachauri, D., Kondor, R., and Singh, V. Solving the multi-way matching problem by permutation synchronization. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 1860–1868. Curran Associates, Inc., 2013.

Perry, A., Wein, A. S., Bandeira, A. S., and Moitra, A. Message-passing algorithms for synchronization problems over compact groups. *Communications on Pure and Applied Mathematics*, 2018.

Purkait, P., Chin, T., and Reid, I. D. Neurora: Neural robust rotation averaging. *arXiv preprint arXiv:1912.04485*, 2019.

Rosen, D. M., Carlone, L., Bandeira, A. S., and Leonard, J. J. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *I. J. Robotics Res.*, 38(2-3), 2019.

Sengupta, S., Amir, T., Galun, M., Goldstein, T., Jacobs, D. W., Singer, A., and Basri, R. A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 4798–4806, 2017.

Shen, T., Zhu, S., Fang, T., Zhang, R., and Quan, L. Graph-based consistent matching for structure-from-motion. In *European Conference on Computer Vision*, pp. 139–155. Springer, 2016.

Shi, Y. and Lerman, G. Estimation of camera locations in highly corrupted scenarios: All about that base, no shape trouble. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 2868–2876, 2018.

Shi, Y., Li, S., and Lerman, G. Robust multi-object matching via iterative reweighting of the graph connection laplacian, 2020.

Tron, R., Vidal, R., and Terzis, A. Distributed pose averaging in camera networks via consensus on SE(3). In *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras, Stanford, CA, USA, September 7-11, 2008*, pp. 1–10, 2008.

Wang, L. and Singer, A. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, 2013.

Wilson, K. and Snavely, N. Robust global translations with 1dsfm. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pp. 61–75, 2014.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

Zach, C., Klopschitz, M., and Pollefeys, M. Disambiguating visual relations using loop constraints. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 1426–1433, 2010.