An Iterative Projection method for Synchronization of Invertible Matrices Over Graphs

Johan Thunberg*, Nicolo Colombo, Zuogong Yue, and Jorge Goncalves

***Abstract*— This paper addresses synchronization of invertible matrices over graphs. The matrices represent pairwise transformations between n euclidean coordinate systems. Synchronization means that composite transformations over loops are equal to the identity. Given a set of measured matrices that are not synchronized, the synchronization problem amounts to fining new synchronized matrices close to the former. Under the assumption that the measurement noise is zero mean Gaussian with known covariance, we introduce an iterative method based on linear subspace projection. The method is free of step size determination and tuning and numerical simulations show significant improvement of the solution compared to a recently proposed direct method as well as the Gauss-Newton method.**

I. INTRODUCTION

This paper presents a method for synchronization of invertible matrices (or transformations) over graphs. The word "synchronization" in this context, does not refer to consensus [1] or rendezvous, e.g., attitude synchronization [2], but to transitive consistency [3], [4]. It means that transformations over loops are equal to the identity. This is a property that must be fulfilled if the transformations are bijections. If the property is not fulfilled, i.e., the matrices are not synchronized, the objective is to find new matrices that are synchronized and "close" to the former in an appropriate sense.

The synchronization problem has been addressed in the literature before, but mostly for the case of orthogonal matrices. In the case of orthogonal matrices, Govindu et al. use Lie-group averaging, where a first-order approximation in the tangent space is employed [5]–[7]. Singer et al., present several optimization-based approaches [4], [8], [9]. The same group of authors have also presented several application-oriented results [10]–[13]. Their works have also been adapted by Pachauri et al. to the case where the transformations are permutation matrices [14].

Practical examples where synchronization is present include rigid bodies in space and multiple-images registration [15]. In the former, the transformations are rigid and in the latter the transformations are affine (if not nonlinear); by using homogeneous coordinates, the transformations can be represented as matrices in $GL(d,\mathbb{R})$. Furthermore, the rotational respective linear part can be chosen to be synchronized independently from the translational part. Synchronization is also closely related to the 3D localization problem, where rigid transformations are calculated from camera measurements [16]–[18], and the Generalized Procrustes Problem, where rotations, translations and scales are calculated from point-clouds [19]–[23].

In this paper, we introduce a new iterative method for synchronization. Under the assumptions that the matrices are drawn from Gaussian distributions with known covariance, the method achieves a suboptimal solution to the maximum log-likelihood problem. The key procedure in each iteration of the method comprises a projection step, where the matrices calculated in the previous iteration are projected onto a linear subspace calculated by spectral factorization of a positive semi-definite matrix. The method is free from step-size determination and tuning. For initialization, a problem is solved with a convex objective function and quadratic constraints; it is solved by means of spectral factorization of a Hessian matrix. This initialization method improves on a recently presented direct method [23].

II. GRAPHS AND CONSISTENT MATRICES

In order to define the problem we want to solve, we need to state some concepts of directed graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the node set and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the edge set.

Definition II.1. (connected graph, undirected path)
The directed graph $\mathcal{G}$ is connected if there is an undirected path from any node in the graph to any other node. An undirected path is defined as a (finite) sequence of unique nodes such that for any pair (i,j) of consecutive nodes in the sequence it holds that

$$((i,j) \in \mathcal{E}) \text{ or } ((j,i) \in \mathcal{E}).$$

Definition II.2. (quasi-strongly connected graph, center, directed path)
The connected directed graph $\mathcal{G}$ is quasi-strongly connected (QSC) if it contains a center. A center is a node in the graph to which there is a directed path from any other node in the graph. A directed path is defined as a (finite) sequence of unique nodes such that any pair of consecutive nodes in the sequence comprises an edge in $\mathcal{E}$.

Definition II.3. (strongly connected graph)
The directed graph $\mathcal{G}$ is strongly connected if for each pair $(i,j) \in \mathcal{V} \times \mathcal{V}$, there is a directed path from i to j.

Definition II.4. (strongly connected component)
A strongly connected component of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is a strongly connected graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, such that $\mathcal{V}_c \subset \mathcal{V}$ and $\mathcal{E} \supset \mathcal{E}_c \subset \mathcal{V}_c \times \mathcal{V}_c$

This work was supported by Fonds National de la Recherche Luxemourg FNR8864515, I1R-SCG-PFN-14CRCL.

All authors are with Luxembourg Centre for Systems Biomedicine, University of Luxembourg, 6, avenue du Swing, L-4367 Belvaux, Luxembourg.
*For correspondence, johan.thunberg@uni.lu

Definition II.5. *The complete graph is* $(\mathcal{V}, \mathcal{V} \times \mathcal{V})$.

Now we connect the directed graphs with collections of matrices in $GL(d, \mathbb{R})$.

Definition II.6. (transitive consistency)

1) *The matrices in the collection* $\{G_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ *of matrices in* $GL(d, \mathbb{R})$ *are* transitively consistent for the complete graph *if there is a collection* $\{G_i\}_{i\in\mathcal{V}}$ *of matrices in* $GL(d, \mathbb{R})$ *such that*
$$G_{ij} = G_i^{-1} G_j \text{ for all } i, j. \tag{1}$$
2) *Given a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, *the matrices in the collection* $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ *of matrices in* $GL(d, \mathbb{R})$ *are* transitively consistent for $\mathcal{G}$ *if there is a collection* $\{G_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}} \supset \{G_{ij}\}_{(i,j)\in\mathcal{E}}$ *such that* $\{G_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ *is transitively consistent for the complete graph.*

Another word for transitive consistency, which will be used in this paper, is *synchronization*. We say that two collections $\{G_i\}_{i\in\mathcal{V}}$ and $\{\bar{G}_i\}_{i\in\mathcal{V}}$ are *equal up to transformation from the left* if there is a matrix $Q \in GL(d, \mathbb{R})$ such that
$$QG_i = \bar{G}_i \text{ for all } i.$$

Lemma II.1 ([3]). *For any graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *and collection* $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ *of matrices in* $GL(d, \mathbb{R})$ *that are transitively consistent for* $\mathcal{G}$ *the following holds*

1) *All collections* $\{G_i\}_{i\in\mathcal{V}}$ *satisfying* (1) *are equal up to transformation from the left if and only if* $\mathcal{G}$ *is connected,*
2) *there is a unique collection* $\{G_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}} \supset \{G_{ij}\}_{(i,j)\in\mathcal{E}}$ *of transitively consistent matrices for the complete graph, if and only if all collections* $\{G_i\}_{i\in\mathcal{V}}$ *satisfying* (1) *are equal up to transformation from the left.*

Lemma II.2. *The matrices in the collection* $\{G_{ij}\}_{(i,j)\in\mathcal{V}\times\mathcal{V}}$ *of matrices in* $GL(d, \mathbb{R})$ *are transitively consistent for the complete graph if and only if*
$$G_{ik} = G_{ij} G_{jk}$$
for all i, j *and* k.

For a proof of Lemma II.2 we refer to [24].

III. Problem formulation

In this section we formulate the problem. On a high level, the problem is described as follows. Given a directed and connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $\mathbb{R}^{d\times d}$, the problem is to find a collection of transitively consistent (or synchronized) invertible $\bar{G}_{ij}$-matrices (i.e., elements in $GL(d, \mathbb{R})$) that are close to the G_{ij} matrices (we return to the meaning of closeness in the detailed description of the problem below). Due to Definition II.6, this general problem is equivalent to finding a collection $\{\bar{G}_i\}_{i\in\mathcal{V}}$ of invertible matrices such that the matrices $\bar{G}_{ij} = (\bar{G}_i^{-1}\bar{G}_j)$ are close to the G_{ij}-matrices.

Now we provide a detailed description of the problem. It is formulated as a maximum likelihood estimation problem. We assume that there is a collection $\{G_i^{\text{true}}\}_{i\in\mathcal{V}}$ of matrices in $GL(d, \mathbb{R})$ such that that (the vectorization of) each G_{ij}-matrix in the collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ has been drawn from the distribution
$$\mathcal{N}(\text{vec}((G_i^{\text{true}})^{-1}G_j^{\text{true}}), Q_{ij}^{-1}).$$
Each $Q_{ij} \in \mathbb{R}^{d^2\times d^2}$ is positive definite. The vec$(\cdot)$-operator transforms a matrix into a vector by stacking the columns after each other in consecutive order.

The maximum-likelihood matrix synchronization problem is formulated as
$$\begin{aligned} \underset{G}{\text{minimize}} \quad & f(G) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\text{vec}(G_{ij} - G_i^{-1}G_j)^T \cdot \\ & Q_{ij}\text{vec}(G_{ij} - G_i^{-1}G_j), \\ \text{subject to} \quad & G = [G_1, G_2, \ldots, G_n], \\ & G_i \in GL(d, \mathbb{R}) \text{ for all } i. \end{aligned} \tag{2}$$

Remark III.1. *In the special case when the* Q_{ij}*-matrices are equal to the identity, problem* (2) *can be written more compactly as*
$$f(G) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij} - G_i^{-1}G_j\|_F^2.$$

The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is assumed to be connected. Connectivity, is necessary to guarantee that the G_i matrices are unique up to transformations from the left in the case of synchronized G_{ij} matrices, see Lemma II.1.

It should be noted that problem (2) is a nonlinear (nonconvex) least squares problem over an open (non-convex) set. A valid concern is that of the well-posedness of Problem (2), i.e., can we rightfully write "minimize" instead of "inf". In fact, in its most general form, without any requirements for the $\mathcal{G}$-graph and the G_{ij}-matrices, the problem is not well-posed. To be sure that there exists a minimizer to the problem, we have to impose additional constraints on those objects. Proposition III.1 introduces such constraints.

According to Proposition III.1 1), the well-posedness is guaranteed if $\mathcal{G}$ is strongly-connected. The condition 2) in the proposition violates the assumption that the matrices are drawn from Gaussian distributions. Thus, this second condition is nothing but a requirement for the Problem (2) to be well posed. It should be noticed however that the event that at least one of the drawn matrices is rank deficient has probability zero.

Before we proceed, we introduce the set
$$\mathcal{S} = \{G : G = [G_1, G_2, \ldots, G_n], G_i \in GL(d, \mathbb{R}) \text{ for all } i\}.$$

Proposition III.1. *If one of the two assumptions below are fulfilled, then* f *defined in problem* (2) *attains a minimum on the set* $\mathcal{S}$.

1) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *is strongly connected,*

2) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *is connected, and there is at most one strongly connected component* $\mathcal{G}_c$ *of* $\mathcal{G}$ *that contains two or more nodes. For any edge* $(i, j) \in \mathcal{E}$*, where at least one of the nodes* i *and* j *is not in* $\mathcal{G}_c$*, it holds that* $G_{ij} \in GL(\mathbb{R}, d)$*.*

Proof of Proposition III.1: All norms in this proof shall be read as the Frobenius norm. The proof begins with a general part and proceeds with parts tailored for conditions 1) and 2), respectively.

General part: Since, the function f is bounded from below (by 0), there is a sequence $\{G^s\}_{s=1}^{\infty}$ with $G^s \in \mathcal{S}$ for all s such that

$$\inf_{G \in \mathcal{S}} f(G) = \lim_{s \to \infty} f(G^s) \text{ and } f(G^{s+1}) \leq f(G^s) \text{ for all } s.$$

Let $G^s = [G_1^s, G_2^s, \ldots, G_n^s]$ for all s, where $G_i^s \in GL(d, \mathbb{R})$ for all i, s. Since $\{f(G^s)\}_{s=1}^{\infty}$ is decreasing, due to the structure of f, it is easy to verify that $\{\|(G_i^s)^{-1} G_j^s\|\}_{s \in \mathbb{N}, (i,j) \in \mathcal{E}}$ is uniformly bounded, which means that the elements in $\{(G_i^s)^{-1} G_j^s\}_{s \in \mathbb{N}, (i,j) \in \mathcal{E}}$ are uniformly bounded. Hence, there is a sub-sequence $\{s_k\}_{k=1}^{\infty}$ such that $\{(G_i^{s_k})^{-1} G_j^{s_k}\}_{k=1}^{\infty}$ converges for all $(i, j) \in \mathcal{E}$ (application of Weierstrass Theorem). Thus, there are $\bar{G}_{ij} \in \mathbb{R}^{d \times d}$ for all $(i, j) \in \mathcal{E}$ such that

$$\lim_{k \to \infty} (G_i^{s_k})^{-1} G_j^{s_k} = \bar{G}_{ij} \text{ for all } (i, j) \in \mathcal{E}. \qquad (3)$$

One can now show that

$$\inf_{G \in \mathcal{S}} f(G) = \sum_{(i,j) \in \mathcal{E}} \frac{1}{2} \text{vec}(G_{ij} - \bar{G}_{ij})^T Q_{ij} \text{vec}(G_{ij} - \bar{G}_{ij}).$$

Condition 1) The first step is to show that the $\bar{G}_{ij}$-matrices are invertible. Suppose that there is an edge $(i_1, i_2) \in \mathcal{E}$ such that $\bar{G}_{i_1 i_2}$ is not invertible. Since the graph $\mathcal{G}$ is strongly connected (i, j) is part of a loop with distinct nodes. Let the edges in the loop be $(i_1, i_2), (i_2, i_3), \ldots, (i_N, i_1)$, where N is the number nodes in the loop. Now it holds that

$$\begin{aligned}
& \lim_{k \to \infty} (G_{i_1}^{s_k})^{-1} G_{i_2}^{s_k} (G_{i_2}^{s_k})^{-1} G_{i_3}^{s_k} \cdots (G_{i_N}^{s_k})^{-1} G_{i_1}^{s_k} \\
= \; & (\lim_{k \to \infty} (G_{i_1}^{s_k})^{-1} G_{i_2}^{s_k}) (\lim_{k \to \infty} (G_{i_2}^{s_k})^{-1} G_{i_3}^{s_k}) \cdots \\
& (\lim_{k \to \infty} (G_{i_N}^{s_k})^{-1} G_{i_1}^{s_k}) = \bar{G}_{i_1 i_2} \bar{G}_{i_2 i_3} \cdots \bar{G}_{i_N i_1} = I.
\end{aligned}$$

Since the right-hand side in the last equation is inverible, all the matrices in the left-hand side must also be. Especially $G_{i_1 i_2}$, which was claimed not to. This is a contradiction. Hence, all the $\bar{G}_{ij}$-matrices are invertible.

Now we can augment the collection of $\bar{G}_{ij}$-matrices. We define

$$\bar{G}_{ij} = \lim_{k \to \infty} (G_i^{s_k})^{-1} G_j^{s_k} \text{ for all } (j, i) \in \mathcal{E}, \qquad (4)$$

For any $(i, l) \notin \mathcal{E}$, for which there is j such that $(i, j), (j, l) \in \mathcal{E}$ holds, we define $\bar{G}_{il}$ by

$$\begin{aligned}
& \lim_{k \to \infty} (G_i^{s_k})^{-1} G_j^{s_k} (G_j^{s_k})^{-1} G_l^{s_k} \\
= & (\lim_{k_1 \to \infty} (G_i^{s_{k_1}})^{-1} G_j^{s_{k_1}}) (\lim_{k_2 \to \infty} (G_j^{s_{k_2}})^{-1} G_l^{s_{k_2}}) \\
= & \; \bar{G}_{ij} \bar{G}_{jl} = \bar{G}_{il}.
\end{aligned}$$

Now we can add new matrices to the collection of existing $\bar{G}_{ij}$-matrices by considering new (i, j, l)-triplets as set forth above. As a consequence of the fact that the graph $\mathcal{G}$ is connected, this procedure can be continued until all the $\bar{G}_{ij}$-matrices are added (i.e., all index pairs are considered). All the matrices in the collection $\{\bar{G}_{ij}\}_{(i,j) \in \mathcal{V} \times \mathcal{V}}$ are invertible. Furthermore, by using a limit argument, one can show that

$$\bar{G}_{ij} \bar{G}_{jk} = \bar{G}_{il} \text{ for all } i, j, l.$$

The latter implies – due to Lemma II.2 – that there are $\bar{G}_i$-matrices in $GL(d, \mathbb{R})$ such that

$$\bar{G}_{ij} = \bar{G}_i^{-1} \bar{G}_j \text{ for all } (i, j) \in \mathcal{V} \times \mathcal{V}.$$

Hence, the problem has a minimizer under condition 1).

Condition 2) We partition $\mathcal{E}$, i.e., the edge-set of $\mathcal{G}$, into two sets. The first, S_1, comprises edges that are contained in the strongly connected component $\mathcal{G}_c$. The second, S_2, comprises the rest of the edges.

First, we consider edges in S_1. Along the lines of the proof for condition 1), one can prove that $\bar{G}_{ij} \bar{G}_{jk} = \bar{G}_{ik}$ for all $i, j, k \in \mathcal{G}_c$. Furthermore, for all edges in S_1, there are $\bar{G}_i$-matrices such that $\bar{G}_{ij} = \bar{G}_i^{-1} \bar{G}_j$ for all $i, j \in \mathcal{G}_c$.

Secondly, we consider edges in S_2. Each edge in S_2 is contained in a tree (which is one of perhaps many trees consisting of nodes that are not in $\mathcal{G}_c$) where one and only one "leaf-edge" ends or starts in a node i_c that is a node in $\mathcal{G}_c$. Now one can show that there are $\bar{G}_{ij}$-matrices for the edges in the tree such that

$$\bar{G}_{ij} = \bar{G}_i^{-1} \bar{G}_j,$$

where the $\bar{G}_i$-matrices (for all nodes but i_c) are computed recursively, starting with G_k, where k is the other node, besides i_c, in the "leaf-edge". Depending on whether the edge has the form (k, i_c) or (i_c, k), the matrix $\bar{G}_k$ is either given by $\bar{G}_k^{-1} = G_{k i_c} G_{i_c}^{-1}$ or $\bar{G}_k = G_{i_c} G_{i_c k}$. The matrix $G_{k i_c}$ was assumed to be invertible, see condition 2). Once $\bar{G}_k$ is computed, we can continue with the matrices of the nodes directly connected to k and so on. The equivalent procedure is then performed for all the trees (if there are more than one). □

Remark III.2. *Note that since the graph* $\mathcal{G}$ *is assumed to be connected, the* $\bar{G}_{ij}$*-matrices in the proof of Proposition III.1 correspond to a unique (up to transformation from the left that is) collection of* $\bar{G}_i$*-matrices, see Lemma II.1. However, we do not claim the solution to be unique (up to transformation from the left) of the optimal solution to this problem.*

IV. THE ITERATIVE PROJECTION METHOD

In this section we introduce our method, which is an iterative method that achieves a suboptimal solution to Problem (2). At each iteration k, the matrix

$$G(k) = [G_1(k), G_2(k), \dots, G_n(k)]$$

is calculated. After a suitable number of iterations, T, the algorithm stops and the final $\bar{G}$-matrix is chosen as the best $G(k)$-matrix for all the iterations $k = 1, 2, \dots, T$.

A. Initialization

As initialization for the method (step 0 in the main algorithm), an other optimization problem than (2) is solved. The problem is defined below.

$$\begin{aligned} \underset{G}{\text{minimize}} \quad & \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\|G_{ij}G_j - G_i\|_F^2 \\ \text{subject to} \quad & G = [G_1, G_2, \dots, G_n], \\ & G_i \in \mathbb{R}^{d\times d}, \\ & GG^T = nI. \end{aligned} \tag{5}$$

The $G_i(0)$-matrices are obtained from the optimal solution to this problem. Provided that the G_{ij}-matrices are close enough to be synchronized, one can show that that the optimal $\bar{G}_i$-matrices are invertible and the problem is well posed. Solving (5) amounts to performing a Spectral Factorization or a Singular Value Decomposition (SVD). More specifically, we reformulate the problem (5) as:

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \text{trace}(W^T H W), \\ \text{subject to} \quad & W \in \mathbb{R}^{nd\times d}, W^T W = I. \end{aligned} \tag{6}$$

The matrix H is defined as

$$H = \text{diag}(A\mathbf{1}) \otimes I + \text{diag}(W^T W) - W - W^T,$$

where $\mathbf{1} \in \mathbb{R}^n$ is the vector where each element is equal to 1, $W = [W_{ij}]$, $A = [A_{ij}]$, and

$$W_{ij} = \begin{cases} G_{ij} & \text{if } (i,j)\in\mathcal{E}, \\ 0 & \text{else,} \end{cases}, \quad A_{ij} = \begin{cases} 1 & \text{if } (i,j)\in\mathcal{E}, \\ 0 & \text{else.} \end{cases}$$

Now, let V be the optimal solution to Problem (6) (obtained by preforming SVD or spectral factorization of H) and identify the $G_i(0)$-matrices by

$$V^T = [(G_1(0))^{-T}, (G_2(0))^{-T}, \dots, (G_n(0))^{-T}].$$

B. Main algorithm

Step 0: Set $k = 0$, choose $T \in \mathbb{N}$, and choose $G(0)$ according to the procedure in Section IV-A.

Step 1: Let $\bar{H}(k) = [\bar{H}_{ij}(k)]$, where

$$\begin{aligned} \bar{H}_{ii}(k) = & \sum_{j\in\mathcal{N}_i} (G_j(k)^T \otimes I)^T Q_{ij} (G_j(k)^T \otimes I) + \\ & \sum_{\{j: i\in\mathcal{N}_j\}} (G_i(k)^T \otimes G_{ji})^T Q_{ji} (G_i(k)^T \otimes G_{ji}) \end{aligned}$$

for all i and

$$\bar{H}_{ij}(k) =$$

$$\begin{cases} 0 \end{cases} \quad \text{if } \begin{cases} (i,j) \notin \mathcal{E}, \\ (j,i) \notin \mathcal{E}, \end{cases}$$

$$\begin{cases} -(G_j(k)^T \otimes I)^T Q_{ij} (G_j(k)^T \otimes G_{ij}) \end{cases} \quad \text{if } \begin{cases} (i,j) \in \mathcal{E}, \\ (j,i) \notin \mathcal{E}, \end{cases}$$

$$\begin{cases} -(G_i(k)^T \otimes G_{ji})^T Q_{ji} (G_i(k)^T \otimes I) \end{cases} \quad \text{if } \begin{cases} (i,j) \notin \mathcal{E}, \\ (j,i) \in \mathcal{E}, \end{cases}$$

$$\begin{cases} -(G_j(k)^T \otimes I)^T Q_{ij} (G_j(k)^T \otimes G_{ij}) \\ \quad - (G_i(k)^T \otimes G_{ji})^T Q_{ji} (G_i(k)^T \otimes I) \end{cases} \quad \text{if } \begin{cases} (i,j) \in \mathcal{E}, \\ (j,i) \in \mathcal{E}, \end{cases}$$

for all $i \neq j$. The object $\otimes$ is the Kronecker product.

Step 2: Let $Z(k)$ be the matrix (up to permutation), whose column vectors are given by the d^2 eigenvectors corresponding to the d^2 smallest eigenvalues of the matrix $\bar{H}(k)$. These eigenvectors are computed by spectral factorization of $\bar{H}(k)$.

Step 3: Define $R = [R_1, R_2, \dots, R_n] \in \mathbb{R}^{d\times nd}$ via

$$\begin{aligned} & \text{vec}(R) \\ = & Z(k)Z(k)^T \text{vec}([(G_1(k))^{-1}, (G_2(k))^{-1}, \dots, (G_n(k))^{-1}]) \end{aligned} \tag{7}$$

where $R_i \in \mathbb{R}^{d\times d}$ for all i.

Step 4: If the R_i-matrices are invertible and $k + 1 \leq T$, let $k = k + 1$, $\bar{G}(k) = R$, and goto **Step 1.** Else, let

$$\bar{G} = G(l), \text{ where } l = \arg \min_{m\in\{1,2,\dots,k\}} f(G(m))$$

and terminate the algorithm.

V. EXPLANATION AND MOTIVATION OF THE ALGORITHM

In this section we explain and motivate the proposed algorithm. To begin with, we turn to Section IV-A and the initialization procedure. The function f in (2) reduces to the objective function in Problem (6) if two restrictions are fulfilled. The first is that all the Q_{ij}-matrices are equal to I – in this case the function is on the form given in Remark III.1 – the second is that the matrices are constrained to be orthogonal. If these two restrictions are fulfilled, the procedure in Section IV-A generates a solution very close to the global optimal solution of the restricted Problem (2), see [3] and especially Section 3.4 for further details on the method. The constraint $GG^T = nI$ is a relaxation of the constraint that $G_i G_i^T = I$ for all i. A related method, also for the case where $Q_{ij} = I$ for all i, j was presented in [23]. That method was specialized for affine transformation matrices.

One of the more interesting properties of Problem (6) (or rather the matrix H defined in Section IV-A) is provided by the following proposition.

Proposition V.1 ([3]). *The collection $\{G_{ij}\}_{(i,j)\in\mathcal{E}}$ of matrices in $GL(d,\mathbb{R})$ is transitively consistent for the connected graph $\mathcal{G}$ if and only if*

$$dim(ker(H)) = d,$$

where H is defined in Section IV-A.

Furthermore, one can also show that the rank of H is always larger or equal to $(n-1)d$ if the G_{ij}-matrices are invertible.

Now we turn to the motivation behind the main algorithm. We want to solve problem (2). However, not only is the feasible set non-convex, the objective function in (2) is non-convex too. A key idea behind the algorithm is to approximate the objective function with a convex one. This is done in the manner described below. The objective function f in Problem (2) is given by

$$\begin{aligned} f(G) &= \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\text{vec}(G_{ij} - G_i^{-1}G_j)^T Q_{ij}\text{vec}(G_{ij} - G_i^{-1}G_j) \\ &= \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}\text{vec}((G_{ij}G_j^{-1} - G_i^{-1})G_j)^T \cdot \\ &\quad Q_{ij}\text{vec}((G_{ij}G_j^{-1} - G_i^{-1})G_j). \end{aligned}$$

Now, let us introduce the following function

$$\begin{aligned} \bar{f}(G,R) = \sum_{(i,j)\in\mathcal{E}} &\frac{1}{2}\text{vec}((G_{ij}R_j - R_i)G_j)^T \cdot \\ &Q_{ij}\text{vec}((G_{ij}R_j - R_i)G_j), \end{aligned}$$

where $R = [R_1, R_2, \dots, R_n]$ and each $R_i \in \mathbb{R}^{d\times d}$ for all i. The problem (2) can be rewritten as

$$\begin{aligned} &\underset{G}{\text{minimize}} && \bar{f}(G,R) \\ &\text{subject to} && R_i = G_i^{-1} \text{ for all } i. \end{aligned} \tag{8}$$

Now, let us consider the following optimization problem. In each iteration we aim to minimize $\bar{f}(G(k),R)$ with respect to R. To be more specific, we solve a the following problem

$$\begin{aligned} &\underset{P}{\text{minimize}} && \|G(k) - P\|_F \\ &\text{subject to} && P = [P_1, P_2, \dots, P_n], \\ &&& P_i \in \mathbb{R}^{d\times d} \text{ for all } i, \\ &&& \text{vec}(P) \in \text{im}(Z(k)), \\ &&& Z(k) = \arg\min_Z \{Z^T\bar{H}(k)Z : \dots \\ &&& \qquad Z \in \mathbb{R}^{nd^2\times d^2}, Z^TZ = I\}, \\ &&& \bar{H}(k) = (\nabla_R)^2\bar{f}(G(k),R). \end{aligned} \tag{9}$$

The Hessian matrix of the function $\bar{f}(G(k),R)$ with respect to R is $\bar{H}(k)$. If the G_{ij}-matrices are transitively consistent and the $G_i(k)$-matrices fulfill (1), then $\bar{g}(k) = \text{vec}([G_1^{-1}(k), G_2^{-1}(k), \dots, G_n^{-1}(k)])$ is contained in the d^2-dimensional nullspace of $\bar{H}(k)$. If the G_{ij}-matrices are not synchronized, the optimal solution to a certain optimization problem is contained in $\text{im}(Z(k))$. The optimization problem is that of minimizing $\bar{f}(G(k),R)$ with respect to R, subject to the constraint that $\text{vec}(R)$ has unit norm. Now, one could propose a method similar to the one in Section IV-A, where the optimal solution is obtained from $Z(k)$ (which is the equivalent to the matrix V in Section IV-A). However, $\text{im}(Z(k))$ is a d^2-dimensional subspace and it is hard to extract the optimal solution from this space. The trick is instead to project the (vectorization of the) near-optimal solution, consisting of the inverses of the $G_i(k)$-matrices, onto $\text{im}(Z(k))$, and by this procedure, removing the parts that are not contained in $\text{im}(Z(k))$. Solving Problem (9) corresponds to performing this projection procedure. The projection is given by the matrix R in (7).

VI. Numerical experiments

To evaluate the performance of the algorithm, numerical simulations were conducted. Results of those simulations are shown in Fig. 1. For each out of eight different parameter settings, 100 simulations were run and the minimum and the median of the normalized f-values are shown for each iteration of the algorithms (see explanation of the normalization below). Red lines correspond to our proposed method, whereas black lines correspond to the Gauss-Newton method, which is a standard method for nonlinear least squares problems. In [3], explicit expressions are provided for the Gauss-Newton method for the case where all the Q_{ij}-matrices are equal to the identity matrix.

For simplicity, we have not considered the damped version of the Gauss-Newton method, nor the Levenberg-Marquad generalization. Thus, we reserve ourselves for possible improvements present, would these aspects been accounted for. As is the case for our method, the initialization procedure for the Gauss-Newton method is the direct method in Section IV-A.

The objective values have been normalized compared to that obtained via the initialization procedure. The initialization procedure is (to the best of our knowledge) the state of the art in terms of direct methods. The problem at hand, (2) that is, has (to the best of our knowledge) not been studied before in the general form we consider. Due to this reason, we did not consider more comparison methods than the Gauss-Newton method and the direct method in Section IV-A.

The parameters in Fig. 1 are defined as follows.

- n is the number of coordinate systems.
- d is the dimension of the matrices, i.e., the matrices are contained in $\mathbb{R}^{d\times d}$.
- ρ is the graph density. We always assume that the graphs are connected, so $\rho = 0$ corresponds to a tree graph and $\rho = 1$ corresponds to the complete graph, with linear interpolation in between.
- σ_g is a parameter used in the generation of the G_{ij}-matrices. It is used in the following procedure.
  - First random orthogonal matrices are generated, call them R_i, for all $i \in \mathcal{V}$. Each such matrix is generated from $\bar{R}_i$ matrices whose elements

are samples drawn from the uniform distribution with $(-0.5, 0.5)$ as support. The $\bar{R}_i$-matrices are projected onto $O(n)$ in the least square sense to get the R_i-matrices.
  - Then G_i^{true}-matrices are created by element wise addition of samples drawn from $\mathcal{N}(0, \sigma_g)$ to the R_i-matrices.
  - Now $\bar{G}_{ij}^{\text{true}} = (G_i^{\text{true}})^{-1} G_j^{\text{true}}$ for all i, j. These synchronized $\bar{G}_{ij}^{\text{true}}$-matrices are then used for the means in the distributions $\mathcal{N}(\text{vec}(G_{ij}^{\text{true}}), Q_{ij}^{-1})$, from which the (vectorization of the) G_{ij}-matrices are drawn.

- σ_Q is a parameter used in the generation of the Q_{ij} matrices. $\bar{Q}_{ij}$-matrices are created by element wise addition of samples drawn from $\mathcal{N}(0, \sigma_Q)$ to the identity matrix. Then $Q_{ij} = \bar{Q}_{ij}^T \bar{Q}_{ij}$. Thus, if $\sigma_Q = 0$, it holds that $Q_{ij} = I$ for all i, j.

There are a couple of things to note here. The first is that the graph is not necessarily strongly connected, albeit connected. The conditions in Proposition III.1 are not necessarily fulfilled. Yet, the method works well in practice (in the medium sense), see the simulations. Improvement of the initial solution was observed for all the different parameter settings, which is not the case for the Gauss-Newton method. The second thing to note is the involvement of orthogonal matrices in the generation of the G_i^{true}-matrices. Orthogonal matrices are the least ill-conditioned matrices (with respect to matrix inversion). Thus, for near-orthogonal G_i^{true}-matrices, the optimization problem is more tractable than for more ill-conditioned matrices. For all parameter settings in Fig. 1, $\sigma_g = 0.5$, which means that the G_i^{true}-matrices are (statistically speaking) not that close to orthogonal matrices.

We should emphasize here that in each simulation, for each setting of parameters, new matrices and graphs were generated. Also, note that no median-improvement is achieved when the the Gauss-Newton method is used for parameter settings corresponding to the bottom sub-figures. As already mentioned, one might argue that by choosing proper step-sizes and by augmenting the Gauss-Newton method with a gradient term (the Levenberg-Marquad algorithm) we would improve the perfomance. Now, this is underpinning one of the advantages with the proposed method – it is free of such step-size determination and tuning, which makes it easy to use in practice.

VII. Conclusions

A new iterative method has been presented for synchronization of invertible matrices over graphs. The matrices are assumed to be drawn from Gaussian distributions with known covariance. The method is projection method, where the updated matrices in each iteration are obtained by means of projection onto a linear subspace. The method has fast convergence and is free of step-size determination and tuning. In numerical simulations, the method performs better than the Gauss-Newton method as well as a recently proposed direct method.

References

[1] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[2] J. Thunberg, W. Song, E. Montijano, Y. Hong, and X. Hu. Distributed attitude synchronization control of multi-agent systems with switching topologies. *Automatica*, 50(3):832–840, 2014.

[3] J. Thunberg, F. Bernard, and J. Goncalves. On transitive consistency for linear invertible transformations between euclidean coordinate systems. *arXiv preprint arXiv:1509.00728*, 2015.

[4] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, page iat005, 2013.

[5] V.M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR*. IEEE, 2004.

[6] V.M. Govindu. Robustness in motion averaging. In *Computer Vision–ACCV 2006*, pages 457–466. Springer, 2006.

[7] V.M. Govindu and A. Pooja. On averaging multiview relations for 3d scan registration. *Transactions on Image Processing*, 23(3):1289–1302, 2014.

[8] A.S. Bandeira, A. Singer, and D.A. Spielman. A cheeger inequality for the graph connection laplacian. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1611–1630, 2013.

[9] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20–36, 2011.

[10] K.N. Chaudhury, Y. Khoo, and A. Singer. Global registration of multiple point clouds using semidefinite programming. *arXiv.org*, June 2013.

[11] R. Hadani and A. Singer. Representation theoretic patterns in three dimensional Cryo-Electron Microscopy I: The intrinsic reconstitution algorithm. *Annals of mathematics*, 174(2):1219, 2011.

[12] R. Hadani and A. Singer. Representation Theoretic Patterns in Three-Dimensional Cryo-Electron Microscopy II—The Class Averaging Problem. *Foundations of computational mathematics (New York, N.Y.)*, 11(5):589–616, 2011.

[13] A. Singer and Y. Shkolnisky. Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming. *SIAM journal on imaging sciences*, 4(2):543–572, June 2011.

[14] D. Pachauri, R. Kondor, and V. Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in neural information processing systems*, pages 1860–1868, 2013.

[15] F. Bernard, J. Thunberg, A. Husch, L. Salamanca, P. Gemmar, F. Hertel, and J. Goncalves. Transitively consistent and unbiased multi-image registration using numerically stable transformation synchronisation. In *MICCAI Workshop on Spectral Analysis in Medical Imaging (SAMI)*, 2015.

[16] R. Aragues, C. Sagues, and Y. Mezouar. *Parallel and Distributed Map Merging and Localization: Algorithms, Tools and Strategies for Robotic Networks*. Springer, 2015.

[17] R. Tron and R. Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *Transactions on Automatic Control*, 59(12):3325–3340, 2014.

[18] E. Montijano, D. Zhou, M. Schwager, and C. Sagues. Distributed formation control without a global reference frame. In *American Control Conference (ACC), 2014*, pages 3862–3867. IEEE, 2014.

[19] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987.

[20] J.C. Gower and G.B. Dijksterhuis. *Procrustes problems*, volume 3. Oxford University Press Oxford, 2004.

[21] P.H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, March 1966.

[22] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127, 1988.

[23] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves. A Solution for Multi-Alignment by Transformation Synchronisation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

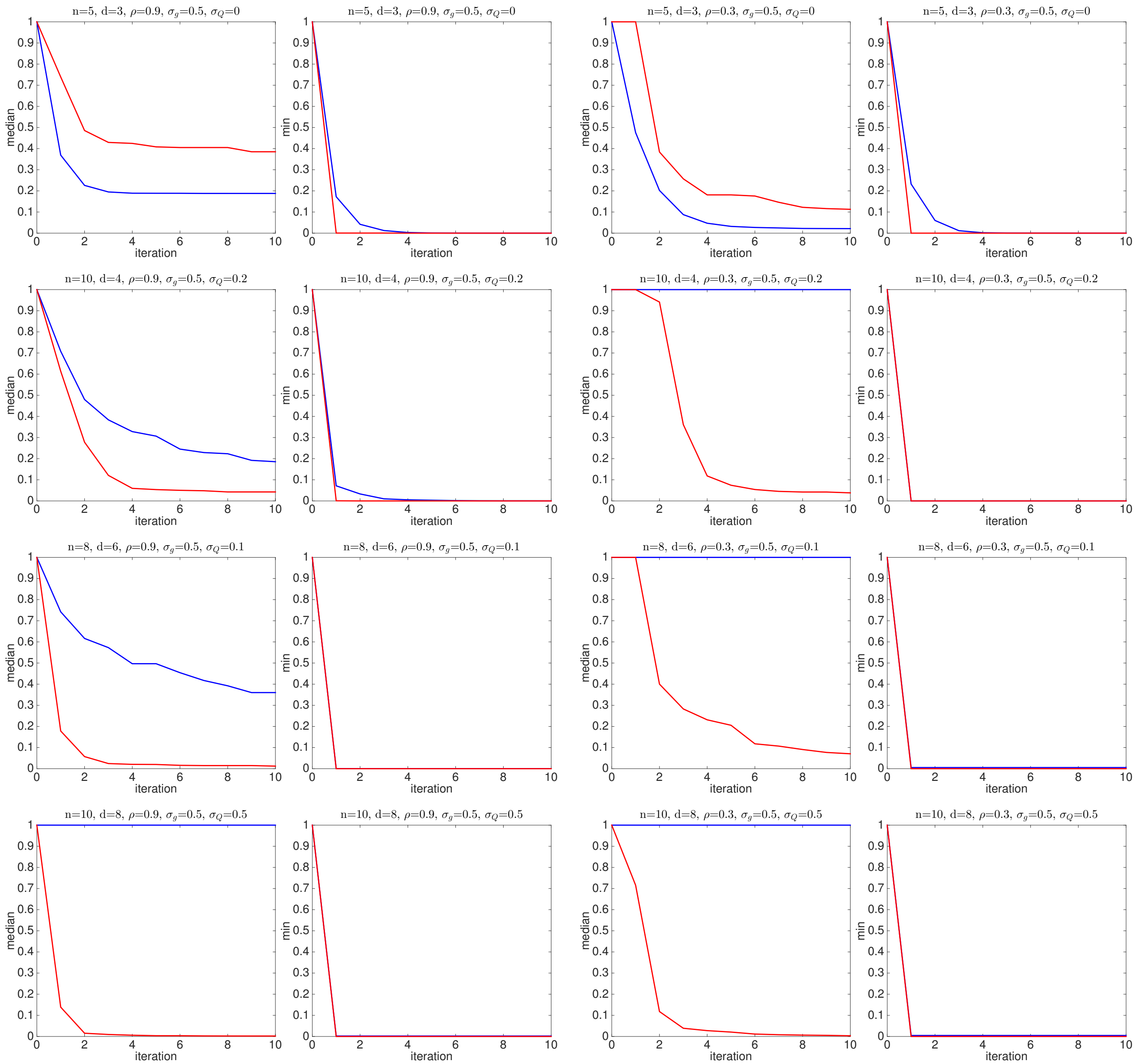

Fig. 1: Improvement over the direct method for the proposed algorithm and the Gauss-Newton method. Eight different parameter settings are considered. Values below 1 are improvements compared to the direct method, which is being used for initialization.

[24] F. Bernard, J. Thunberg, P. Gemmar, F. Hertel, A. Husch, and J. Goncalves. A solution for multi-alignment by transformation synchronisation. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.