2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

CLEAR: A Consistent Lifting, Embedding, and Alignment Rectification Algorithm for Multiview Data Association

Kaveh Fathian, Kasra Khosoussi, Yulun Tian, Parker Lusk, and Jonathan P. How

***Abstract*—Many robotics applications require alignment and fusion of observations obtained at multiple views to form a global model of the environment. Multiway data association methods provide a mechanism to improve alignment accuracy of pairwise associations and ensure their consistency. However, existing methods that solve this computationally challenging problem are often too slow for real-time applications. Furthermore, some of the existing techniques can violate the cycle consistency principle, thus drastically reducing the fusion accuracy. This article presents the consistent lifting, embedding, and alignment rectification (CLEAR) algorithm to address these issues. By leveraging insights from the multiway matching and spectral graph clustering literature, CLEAR provides cycle-consistent and accurate solutions in a computationally efficient manner. Numerical experiments on both synthetic and real datasets are carried out to demonstrate the scalability and superior performance of our algorithm in real-world problems. This algorithmic framework can provide significant improvement in the accuracy and efficiency of existing discrete assignment problems, which traditionally use pairwise (but potentially inconsistent) correspondences. An implementation of CLEAR is made publicly available online.**

***Index Terms*—Data association, multi-way matching, simultaneous localization and mapping.**

I. Introduction

DATA association across *multiple* views, known as the multiview or multiway [1] matching, is a fundamental problem in robotic perception and computer vision. Conceptually, the goal in this problem is to establish correct associations between the sightings of "items" across multiple "views" (see Fig. 1). Examples include feature matching across multiple frames [1]–[3], and associating landmarks across multiple maps for map fusion in single/multi-robot simultaneous localization and mapping (SLAM) [4].

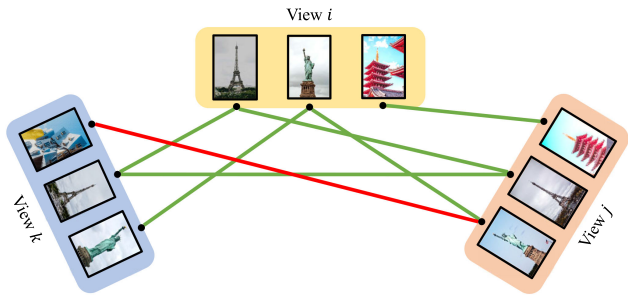


Fig. 1. Illustrative example of cycle consistency for the association of images observed in views i, j, k. Associations of "Eiffel tower" are cycle consistent. On the other hand, the "statue of liberty" associations are inconsistent since the images matched between views i and j and views i and k are not matched between views j and k (i.e., violation of transitivity).

The traditional approach treats the multiview data association problem as a sequence of decoupled *pairwise* matching subproblems, each of which can be formulated and solved, e.g., as a linear assignment problem [5]. Such techniques, however, cannot leverage the redundancy in the observations and, furthermore, often result in *nontransitive* (also known as cycle *inconsistent*) associations; see Fig. 1. One can address these issues by *synchronizing* all noisy pairwise associations via enforcing the cycle consistency constraint. Cycle consistency serves two crucial purposes: 1) it provides a natural mechanism for the discovery and correction of wrong (or missing) associations obtained through pairwise matching; and 2) it establishes an equivalence relation on the set of observations, which is *necessary* for global fusion in the so-called clique-centric applications such as map merging (Section VII).

Synchronizing pairwise associations is a combinatorial optimization problem with an exponentially large search space. This problem has been extensively studied in recent years (see [1]–[3], [6] and references therein). These efforts have resulted in several algorithms that can improve the erroneous initial set of pairwise associations. However, providing solutions that are computationally tractable for real-time applications remains a fundamental challenge. Further, the rounding techniques used by some of relaxation-based methods may violate the cycle consistency and *distinctness* constraints (distinctness implies

Manuscript received July 30, 2019; revised February 27, 2020; accepted May 14, 2020. Date of publication July 10, 2020; date of current version December 3, 2020. This work was supported in part by NASA Convergent Aeronautics Solutions project Design Environment for Novel Vertical Lift Vehicles (DELIVER), in part by ONR under BRC Award N000141712072, and in part by ARL DCIST under Cooperative Agreement Number W911NF-17-2-0181. This article was recommended for publication by Associate Editor S.-J. Chung and Editor J. Patrick upon evaluation of the reviewers' comments. *(Corresponding author: Kaveh Fathian.)*

The authors are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: kavehf@mit.edu; kasra@mit.edu; yulun@mit.edu; plusk@mit.edu; jhow@mit.edu).

This article has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The material consists of a CLEAR source code and the code for generating comparison results: https://github.com/mit-acl/clear, and a video of article summary: https://youtu.be/RBxq9KYcgTY.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2020.3002432

that the items observed in each view are unique, and thus must not be associated with each other).

This article aims to address these critical challenges via a novel spectral graph–theoretic approach. Specifically, we leverage the natural graphical representation of the problem and propose a spectral graph clustering technique uniquely tailored for producing accurate solutions to the multiway data association problem in a computationally tractable manner. Our solutions, by construction, are guaranteed to satisfy the cycle consistency and distinctness constraints under any noise regime. These are demonstrated in our extensive empirical evaluations based on synthetic and real datasets in the context of feature matching and map fusion in landmark-based SLAM.

A. Related Work

With the exception of combinatorial methods [7], [8] that do not scale well to large problems, and a recent deep leaning approach in [9], the majority of permutation synchronization algorithms that aim to solve this computationally challenging problem can be classified as (i) convex relaxation; (ii) spectral relaxation; and (iii) graph clustering.

Methods in the first category include [10], which uses a semidefinite programming relaxation of the problem and solves it via the alternating direction method of multipliers [11]. A distributed variation of this method with a similar formulation has been recently presented in [12]. Toward the same goal, [2] uses a low-rank matrix factorization to improve the computational complexity. Works such as [13] and [6] require full observability, whereas methods such as [14] can perform in a partially observable setting, where only a subset of overall items is observed at each view. The aforementioned algorithms often return solutions that have the highest accuracy; however, due to lifting to high-dimensional spaces, they are slow and not suitable for real-time applications.

Methods in the second category are based on a spectral relaxation of the problem, with prominent works including [1] and [15]. The method proposed in [1] returns consistent solutions from noisy pairwise associations using a spectral relaxation in the fully observable setting. Maset *et al.* [15] proposed an eigendecomposition approach that works in a partially observable setting; however, cycle consistency is lost in higher noise regimes. The recent work of [16] leverages a non-negative matrix factorization approach to solve the problem. This method works in a partially observable setting and preserves cycle consistency. Algorithms that use spectral relaxation are relatively fast and return solutions that have comparably high accuracy.

Methods in the third category use a graph representation of the problem. In [3] and [17], the authors have elegantly observed the equivalence relation between cycle consistency and cluster structure of the association graph. This observation is used to find approximate solutions to the problem based on existing graph clustering algorithms. The work done in [17] has considered a constrained clustering approach using a method similar to k-means. In [3], the existing density-based graph clustering algorithm in [18] is leveraged to solve the problem. Methods in this category could be very fast; however, the accuracy may be compromised.

Finally, we point out that the multiway data association problem can be viewed and solved from a graph-matching perspective [19], [20]. Unlike all previously discussed methods (and the present work), which only leverage the association information across views, graph matching additionally incorporates geometrical information between the items in each view. The additional complexity, in general, results in significantly slower algorithms.

B. Our Contributions

Our article provides new insights into connections between the multiway data association problem and the spectral graph clustering literature. We leverage these insights to push the boundaries of accuracy and speed—which are crucial for real-time robotics applications—to solve the multiway data association problem. The main contributions of this work are as follows:

1) To our knowledge, this work provides the first approach that formulates and solves the multiway association problem using a normalized objective function. This normalization is crucial to recover the correct solution when the association graph is a mixture of large and small clusters (Remark 1).
2) We leverage the natural graphical structure of the problem to estimate the unknown universe size[1] from erroneous associations. Specifically, we prove that our technique is guaranteed to recover the correct universe size under certain bounded noise regimes (Proposition 2). Moreover, we empirically demonstrate that the proposed approach is more robust to noise than the standard eigengap heuristic [21] used in the spectral graph clustering literature (Remark 3).
3) We propose a projection (rounding) method that, by construction, is guaranteed to produce solutions that satisfy the cycle consistency and distinctness constraints, whereas these constraints can be violated by some of the state-of-the-art algorithms in high-noise regimes (Section VIII).

In addition, we address an important subtlety regarding the choice of suitable metrics for evaluating the performance of multiway matching algorithms in applications such as map fusion (Section VII). Finally, we provide extensive numerical experiments on both synthetic and real datasets in the context of feature matching and map fusion (Sections VIII and IX). Our empirical results demonstrate the superior performance of our algorithm in comparison to the state-of-the-art methods in terms of both accuracy and speed.

C. Outline

The rest of this article is organized as follows. Section II introduces the notation and definitions, followed by the problem formulation in Section IV. Section IV presents the CLEAR algorithm, followed by a numerical example in Section V. Section

[1]By definition, universe size is the total number of unique items in all views.

TABLE I
SUMMARY OF IMPORTANT NOMENCLATURE USED THROUGHOUT THE ARTICLE

Notation	Domain	Definition and properties
n	$\mathbb{N}$	Total number of views
m	$\mathbb{N}$	Size of universe; number of unique items; number of cliques in the association graph
m_i	$\mathbb{N}$	Number of items observed in view i
l	$\mathbb{N}$	Total number of items observed across all views; $l \stackrel{\text{def}}{=} \sum_i m_i$
~	-	Accent used for variables corresponding to the noisy input
P_j^i	$\{0,1\}^{m_i \times m_j}$	Partial permutation matrix; association matrix between items at views i and j
P	$\{0,1\}^{l \times l}$	Aggregate association matrix consisting of P_j^i's; see (1)
A	$\{0,1\}^{l \times l}$	Adjacency matrix of the association graph; $A = P - I$
D	$\mathbb{N}_0^{l \times l}$	Degree matrix of the association graph
C	$\mathbb{N}_0^{l \times l}$	Diagonal matrix with entries $c_i \stackrel{\text{def}}{=} \sum_j (P)_{ij}$; $C = D + I$
L	$\mathbb{Z}^{l \times l}$	Laplacian matrix of $\mathcal{G}$; $L \stackrel{\text{def}}{=} D - A = C - P$
L_{nrm}	$\mathbb{R}^{l \times l}$	Normalized Laplacian matrix; $L_{\text{nrm}} \stackrel{\text{def}}{=} C^{-\frac{1}{2}} L C^{-\frac{1}{2}}$
P_{nrm}	$\mathbb{R}^{l \times l}$	Normalized association matrix; $P_{\text{nrm}} \stackrel{\text{def}}{=} C^{-\frac{1}{2}} P C^{-\frac{1}{2}}$
P^i	$\{0,1\}^{m_i \times m}$	Lifting permutation matrix; association of items observed at views i to items of the universe
V	$\{0,1\}^{l \times m}$	Aggregate lifting permutation matrix consisting of P^i's; see (3)
U	$\mathbb{R}^{l \times m}$	Normalized aggregate lifting permutation; $U \stackrel{\text{def}}{=} C^{-\frac{1}{2}} V$; eigenvectors associated to m smallest eigenvalues of L_{nrm}
u_i	$\mathbb{R}^m$	Row of U
u_i'	$\mathbb{R}^m$	Pivot row of U

VI discusses the theoretical justification behind the algorithm with proofs presented in the Appendix. Section VII discusses the application domains for the CLEAR algorithm. In Section VIII, CLEAR is benchmarked against the state-of-the-art algorithms using synthetic data. Finally, experimental evaluations of CLEAR on real-world datasets are presented in Section IX. Finally, Section X concludes this article.

II. NOTATION AND DEFINITIONS

We denote the set of natural numbers by $\mathbb{N}$, integers by $\mathbb{Z}$, $\mathbb{N}_0 \stackrel{\text{def}}{=} \{0\} \cup \mathbb{N}$, and define $\mathbb{N}_n \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$. Scalars and vectors are denoted by lower case (e.g., a), matrices by uppercase (e.g., A), and sets by script letters (e.g., $\mathcal{A}$). Cardinality of set $\mathcal{A}$ is denoted by $|\mathcal{A}|$. The element on row i and column j of matrix A is denoted by $(A)_{ij}$. The Frobenius inner product is defined as $\langle A, B \rangle \stackrel{\text{def}}{=} \text{tr}(A^\top B)$, where A and B are matrices of the same size. Finally, $\|\cdot\|$ denotes the (induced) 2-norm. Table I lists the key variables used throughout the article.

A. Permutation Matrices

Matrix $P_j^i \in \{0,1\}^{m_i \times m_j}$ is said to be a *partial permutation* matrix if and only if each row and column of P_j^i *at most* contains a single 1 entry. Matrix P is called a *full permutation* matrix if and only if each row and column has *exactly* a single 1 entry. We denote the space of all (partial or full) permutation matrices by $\mathbb{P}$. Matrix $P^i \in \mathbb{P}$ is said to be a *lifting permutation matrix* if and only if each row of P^i contains a single 1 entry (however, column entries could be all zero). We denote the space of all lifting permutation matrices by $\mathbb{P}^L$. The *aggregate association matrix* consisting of matrices $P_j^i \in \{0,1\}^{m_i \times m_j}$, $i, j \in \mathbb{N}_n$, is defined as

$$P \stackrel{\text{def}}{=} \begin{bmatrix} I & P_2^1 & \cdots & P_n^1 \\ P_1^2 & I & \cdots & P_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ P_1^n & P_2^n & \cdots & I \end{bmatrix} \in \mathbb{R}^{l \times l} \tag{1}$$

where I is the identity matrix with appropriate size, and $l \stackrel{\text{def}}{=} \sum_{i=1}^n m_i$.

B. Graph Theory

We denote a graph with l vertices by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices, and $\mathcal{E}$ is the set of undirected edges. The adjacency matrix $A \in \{0,1\}^{l \times l}$ of $\mathcal{G}$ is defined by $(A)_{ij} = a_{ij}$, where $a_{ij} = 1$ if there is an edge between vertices $v_i, v_j \in \mathcal{V}$, otherwise $a_{ij} = 0$. We assume $a_{ii} = 0$, i.e., graph has no self-loops. The degree of a vertex $v_i \in \mathcal{V}$ is defined as $d_i \stackrel{\text{def}}{=} \sum_{j=1}^l a_{ij}$, and the $l \times l$ degree matrix D is defined as a diagonal matrix with $d_1, \ldots, d_l$ on the diagonal. We define $C \stackrel{\text{def}}{=} D + I$, where I is identity matrix. If c_i's denote the diagonal entries of C, then $C^{-\frac{1}{2}}$ is a diagonal matrix with diagonal entries $1/\sqrt{c_i}$. The Laplacian matrix of $\mathcal{G}$ is defined as $L \stackrel{\text{def}}{=} D - A$. A *cluster graph* $\mathcal{G}$ is a disjoint union of cliques (i.e., complete subgraphs). That is, $\mathcal{G}$ can be partitioned into subgraphs $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$, where each $\mathcal{A}_i$ is a complete graph and there is no edge between any two $\mathcal{A}_i$, $\mathcal{A}_j$. The cliques in a cluster graph are called clusters.

III. PROBLEM FORMULATION

Simply put, the objective of this article is to reconstruct a set of *cycle-consistent* associations from a set of pairwise associations, which may contain error and lack cycle consistency. This problem can be approached from either an optimization or a graph–theoretic viewpoint. In what follows, we will first describe each formulation separately, and then shed light on their connections.

A. Optimization-Based Formulation

We consider n views and assume that view i contains m_i items. Associations (or matchings) between items in views i and j can be represented by a binary matrix $P_j^i \in \{0,1\}^{m_i \times m_j}$, in which the one entries indicate the associations. An example of pairwise associations among three views is shown in Fig. 2.

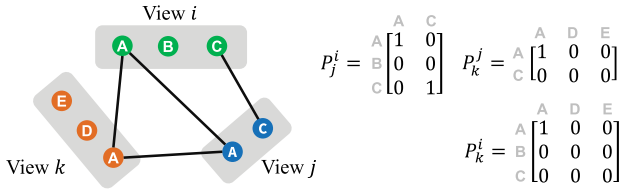


Fig. 2. Association of items labeled as A, B,..., E observed at three views.

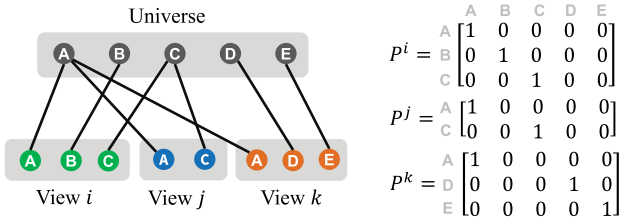


Fig. 3. Lifting permutation matrices associating observations at views i, j, k to the universe, which consists of items labeled as A, B, ..., E.

A lifting permutation represents the association between items observed in a view and the universe (which by definition consists of all items). An example is provided in Fig. 3.

Definition 1: Pairwise associations P_j^i are *cycle-consistent* if and only if there exist lifting permutations $P^i \in \mathbb{P}^L$ such that

$$P_j^i = P^i \, P^{j\top}, \qquad \forall i, j \in \mathbb{N}_n. \tag{2}$$

The cycle consistency condition (2) can be presented more concisely as $P = V\,V^\top$, where P is the aggregate association matrix defined in (1), and

$$V \overset{\text{def}}{=} \begin{bmatrix} P^{1\top} & P^{2\top} & \cdots & P^{n\top} \end{bmatrix}^\top \in \{0,1\}^{l \times m} \tag{3}$$

where $l \overset{\text{def}}{=} \sum_i m_i$. Here, $m \in \mathbb{N}_l$ is the number of columns of lifting permutations that is referred to as the *size of universe*.

Throughout this article, we use the accent to distinguish the variables that are associated with the noisy input. Therefore, $\tilde{P}_j^i \in \{0,1\}^{m_i \times m_j}$ denotes the noisy association between items in views i and j, where $\tilde{P}_i^i = I$ by definition. Note that $\tilde{P}_j^i$'s can be erroneous and inconsistent. Let $\tilde{P} \in \mathbb{R}^{l \times l}$, defined via (1), denote the noisy aggregate association matrix. Further, let $\tilde{C}$ be an $l \times l$ diagonal matrix with positive diagonal entries $\tilde{c}_1, \ldots, \tilde{c}_l$ defined as the sum of corresponding rows of $\tilde{P}$, i.e., $\tilde{c}_i \overset{\text{def}}{=} \sum_{j=1}^{l} (\tilde{P})_{ij}$. Using definitions above, we now formulate the main problem.

Problem 1: Given noisy associations $\tilde{P}_j^i$, find cycle-consistent associations P_j^i that solve the program

$$\begin{aligned} \underset{P_j^i \in \mathbb{P}}{\text{maximize}} \quad & \langle P_{\text{nrm}}, \tilde{P}_{\text{nrm}} \rangle \\ \text{subject to} \quad & P = V\,V^\top, \end{aligned} \tag{4}$$

where $P_{\text{nrm}} \overset{\text{def}}{=} C^{-\frac{1}{2}} P\, C^{-\frac{1}{2}}$, $\tilde{P}_{\text{nrm}} \overset{\text{def}}{=} \tilde{C}^{-\frac{1}{2}} \tilde{P}\, \tilde{C}^{-\frac{1}{2}}$.

In Problem 1, diagonal matrices C and $\tilde{C}$ are used to normalize the aggregate association matrices. The justification behind this normalization will be explained in Remark 1 after the graph formulation of the problem is introduced. The constraint $P_j^i \in \mathbb{P}$ enforces the permutation structure, preventing the rows and columns of P_j^i from having more than a single one entry. This enforces the *distinctness constraint*, which implies that items in the same view are unique, and thus should not be associated with each other. The constraint $P = V\,V^\top$ imposes cycle consistency, capturing the fact that correct associations should be transitive (i.e., if item i is associated to item j, and item j is associated to item k, then item i must also be associated to item k).

B. Graph-Based Formulation

The problem of data association has a graph representation. This representation provides the key insights that are leveraged by our algorithm to improve accuracy and runtime. A set of pairwise associations P_j^i can be represented as a colored graph, where items in each view are denoted by vertices with identical color, and each nonzero entry of P_j^i represents an edge between the corresponding vertices (e.g., Fig. 2). Formally, an *association graph* is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the coloring map $g : \mathcal{V} \to \mathbb{N}_n$. The set of vertices $\mathcal{V}$ consists of subsets $\mathcal{V}_i$ corresponding to items in view i, where $g(v_j) \overset{\text{def}}{=} i$ for all $v_j \in \mathcal{V}_i$. The set of edges $\mathcal{E}$ consists of subsets $\mathcal{E}_{ij}$, $i \neq j \in \mathbb{N}_n$, corresponding to associations, where $\{v_r, v_s\} \in \mathcal{E}_{ij}$ if and only if $(P_j^i)_{rs} = 1$.

The variables P, C, and V defined previously in the optimization formulation (4) also have graph interpretations. Specifically, the adjacency matrix of the association graph $\mathcal{G}$ is given by $A = P - I$. Further, we have that $C = D + I$, where D is the degree matrix of the graph. To understand the graph interpretation of V, we first note the following relation between the cycle consistency and the graph representation.

Proposition 1: A set of pairwise associations is cycle-consistent if and only if the corresponding association graph is a cluster graph (i.e., a disjoint union of complete subgraphs).

Proof of Proposition 1 is given in [3, Prop. 2] and hence omitted here. The proof reveals the connection between the algebraic definition of cycle consistency, $P = V\,V^\top$, and clusters of the association graph, denoted by $\mathcal{A}_1, \ldots, \mathcal{A}_m$. In particular, row i of the aggregate lifting permutation matrix $V \in \{0,1\}^{l \times m}$ represents vertex v_i of the association graph. The one entries in jth column of V indicate the vertices that belong to cluster $\mathcal{A}_j$ of $\mathcal{G}$. That is, if $(V)_{ij} = (V)_{kj} = 1$, then vertices v_i and v_k are connected by an edge and belong to cluster $\mathcal{A}_j$. We will leverage this observation in the theoretical analysis of the algorithm.

Given a noisy association graph $\tilde{\mathcal{G}}$ with adjacency matrix $\tilde{A}$, degree matrix $\tilde{D}$, and $\tilde{C} = \tilde{D} + I$, the graph-based formulation of the multiway association problem is as follows.

Problem 2: Given the noisy association graph $\tilde{\mathcal{G}}$, find undirected graph $\mathcal{G}$ with adjacency matrix A that solves

$$\begin{aligned} \underset{A}{\text{maximize}} \quad & \langle A_{\text{nrm}}, \tilde{A}_{\text{nrm}} \rangle \\ \text{subject to} \quad & \mathcal{G} \text{ consists of clusters } \mathcal{A}_1, \ldots, \mathcal{A}_m \\ & g(v_i) \neq g(v_j), \quad \forall v_i, v_j \in \mathcal{A}_k \end{aligned} \tag{5}$$

where $A_{\text{nrm}} \overset{\text{def}}{=} C^{-\frac{1}{2}} A\, C^{-\frac{1}{2}}$ and $\tilde{A}_{\text{nrm}} \overset{\text{def}}{=} \tilde{C}^{-\frac{1}{2}} \tilde{A}\, \tilde{C}^{-\frac{1}{2}}$.

Note that Problems 1 and 2 are equivalent. As elaborated above, the indices of the vertices belonging to clusters

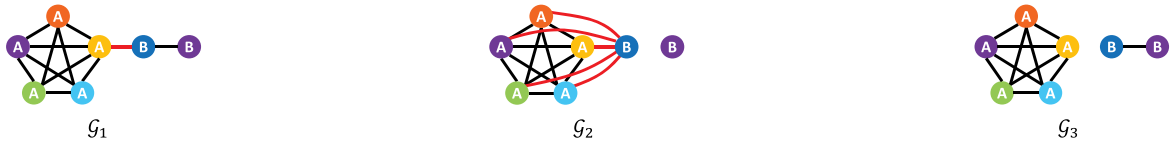


Fig. 4. (Best viewed in color) Graph $\mathcal{G}_1$ indicates the association of two items labeled as A, B, in six views identified by colors ●, ●, ●, ●, ●, ●. The incorrect association, which connects A and B, is indicated by the red edge. If in (4) the unnormalized objective $\langle P, \tilde{P}\rangle$ is used instead, $\mathcal{G}_2$ (and also $\mathcal{G}_3$) would be the optimal solution (with optimal values of 29). On the other hand, by using the proposed normalized objective $\langle P_{\text{nrm}}, \tilde{P}_{\text{nrm}}\rangle$, the correct association graph $\mathcal{G}_3$ is the only optimal solution (with optimal value of 1.79; the value for $\mathcal{G}_2$ is 1.43).

$\mathcal{A}_1, \ldots, \mathcal{A}_m$ uniquely determine V in Problem 1. Further, since $A = P - I$, both objective functions have the same optimizer. In (5), the first two constraints respectively correspond to the cycle consistency and distinctness of associations, where the latter is achieved by the fact that the colors of vertices in each cluster must be distinct.

Remark 1: The normalized objective function in (4) is a key distinction from several state-of-the-art methods [1], [15], [16] that consider the unnormalized objective $\langle P, \tilde{P}\rangle$. By weighting edges based on the degrees of their adjacent vertices, the normalized objective provides a measure to "balance" the number of edges that are removed from or added to the noisy association graph $\tilde{\mathcal{G}}$ to obtain $\mathcal{G}$. The unnormalized objective, on the other hand, is indifferent to the number of added edges. This can lead to (undesired) optimal solutions that consist of many additional edges. This point is illustrated in Fig. 4, where, in contrast to the normalized objective, the optimal solution with an unnormalized objective could fail to recover the ground truth even in a relatively low-noise regime.

We point out that the example shown in Fig. 4 is only one of countless scenarios in which the optimal solution of an *unnormalized* objective could fail to recover the desired association in a low-noise regime. Such examples can be constructed by (wrongly) associating clusters with small and large number of vertices.

IV. Consistent Lifting, Embedding, and Alignment Rectification (CLEAR) Algorithm

In this section, we present a concise description of the CLEAR algorithm used for solving the permutation synchronization problem, followed by a numerical example to further illustrate the steps of the algorithm in the next section. Theoretical justifications of the algorithm are discussed in Section VI. The pseudocode of CLEAR is given in Algorithm 1, where each step is discussed in details below.

• *Step 1:* Let $\tilde{\mathcal{G}}$ denote the association graph corresponding to a set of noisy pairwise associations $\tilde{P}^i_j$. Define the normalized Laplacian of $\tilde{\mathcal{G}}$ as

$$\tilde{L}_{\text{nrm}} \stackrel{\text{def}}{=} \tilde{C}^{-\frac{1}{2}}\, \tilde{L}\, \tilde{C}^{-\frac{1}{2}} \tag{6}$$

where $\tilde{L} = \tilde{D} - \tilde{A}$, $\tilde{C} \stackrel{\text{def}}{=} \tilde{D} + I$, and $\tilde{D}$, $\tilde{A}$ are, respectively, the degree and adjacency matrix of $\tilde{\mathcal{G}}$. Compute the eigenvalues and eigenvectors of $\tilde{L}_{\text{nrm}}$.

Algorithm 1: CLEAR (pseudocode).

Input: Noisy pairwise associations $\tilde{P}^i_j$.
Output: Cycle consistent associations P^i_j.

- **Step 1:** Compute $\tilde{L}_{\text{nrm}}$ from (6) and find its eigendecomposition.
- **Step 2:** Estimate size of universe $\hat{m}$ from (7).
- **Step 3:** Set U as the $\hat{m}$ first eigenvectors of $\tilde{L}_{\text{nrm}}$.
- **Step 4:** Normalize rows of U and chose $\hat{m}$ most orthogonal rows as pivots. Find lifting permutations P^i by assigning rows to pivots based on distance.
- **Step 5:** Set $P^i_j \leftarrow P^i P^{j\top}$.

To reduce the computational complexity, eigendecomposition of $\tilde{L}_{\text{nrm}}$ is done by first finding the connected components of $\tilde{\mathcal{G}}$ using the breadth-first search (BFS) algorithm [22]. Eigenvalues of $\tilde{L}_{\text{nrm}}$ are then given as the disjoint union of each component's normalized Laplacian eigenvalues. Similarly, eigenvectors are given by appropriately padding the eigenvectors of connected components with zeros.

We point out that if $\tilde{L}_{\text{nrm}}$ is not symmetric, its symmetric component $(\tilde{L}_{\text{nrm}} + \tilde{L}^{\top}_{\text{nrm}})/2$ should be used instead in the eigendecomposition (the skew-symmetric component does not contribute to the optimal answer; see Remark 2). Note that the symmetry implies that all eigenvalues and eigenvectors are real.

• *Step 2:* Obtain an estimate for the *size of universe* as

$$\hat{m} \stackrel{\text{def}}{=} \max\{\tilde{m}, m_1, m_2, \ldots, m_n\} \tag{7}$$

where m_i is the number of items in view i, and $\tilde{m}$ is defined as

$$\tilde{m} \stackrel{\text{def}}{=} \left|\{\lambda \in \text{eig}(\tilde{L}_{\text{nrm}}) \,:\, \lambda < 0.5\}\right| \tag{8}$$

i.e., the number of eigenvalues of $\tilde{L}_{\text{nrm}}$ that are less than 0.5.

• *Step 3:* Define matrix $U \in \mathbb{R}^{l\times m}$ as the $\hat{m}$ first eigenvectors of $\tilde{L}_{\text{nrm}}$, that is, the eigenvectors associated with the smallest eigenvalues.

• *Step 4:* Normalize rows of U to have unit norm, i.e., the ith row of U, denoted by u_i, is replaced by $u_i/\|u_i\|$. Choose the $\hat{m}$ most orthogonal rows as *pivots*.

This can be done based on a greedy strategy where the first row of U is chosen as the first pivot. To find the remaining pivots, the row with the smallest inner product magnitude with previously chosen pivots is picked consecutively. That is, if u'_k denotes the kth pivot, u'_{k+1} is selected such that $\sum_{i=1}^{k} |\langle u'_i, u'_{k+1}\rangle|$ is minimized.

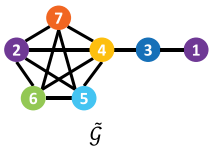


Fig. 5. Association graph corresponding to observations in six views distinguished by color. View 1 is colored as ●, and views 2 through 6 are successively colored as ●, ●, ●, ●, ●. Vertices are numbered from 1 to 7.

For each view i, define matrix $F^i \in \mathbb{R}^{m_i \times m}$ by $(F^i)_{jk} \stackrel{\text{def}}{=} \|u_j - u'_k\|^2$, where u_j denotes the rows of U associated to items in view i, and u'_k denotes the pivot rows.[2] Solve a linear assignment problem based on F^i as the cost matrix to obtain a lifting permutation $P^i \in \mathbb{P}^L$ that associates the items in view i (rows u_j) to the items of the universe (pivot rows u'_k). The Hungarian algorithm [5] can be used to solve the linear assignment problem and find the optimal answer. However, to reduce the computational complexity, faster (suboptimal) algorithms can be used instead while the distinctness constraint is preserved by ensuring that each u'_k is associated to at most one u_j, and each u_j is associated to exactly one u'_k.

• *Step 5:* Compute pairwise associations as $P^i_j = P^i P^{j\top}$. From Definition 1, pairwise associations are cycle-consistent by construction.

V. Numerical Example

We present an example to illustrate the steps of the CLEAR algorithm and show how pivot rows are chosen.

Example 1: In this example, we use the CLEAR algorithm to recover cycle-consistent associations from the (noisy) association graph $\tilde{\mathcal{G}}$ shown in Fig. 5 . Note that $\tilde{\mathcal{G}}$ is identical to $\mathcal{G}_1$ in Fig. 4, where the correct associations and the labels A, B are unknown and should be recovered. The aggregate association matrix (which is equal to the adjacency matrix plus identity) is given by

$$\tilde{P} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{9}$$

The first two rows of P correspond to items in view 1 and the remaining rows successively correspond to views 2 through 6.

• *Step 1:* From (9), the Laplacian matrix is computed as $\tilde{L} = \tilde{C} - \tilde{P}$, where $\tilde{C} = \text{diag}(2, 5, 3, 6, 5, 5, 5)$ and diag creates a diagonal matrix from input arguments. The normalized Laplacian matrix is given by $\tilde{L}_{\text{nrm}} = \tilde{C}^{-\frac{1}{2}} \tilde{L} \tilde{C}^{-\frac{1}{2}}$, which has eigenvalues $\{1.18,\ 1,\ 1,\ 1,\ 0.85,\ 0.17,\ 0\}$.

• *Step 2:* The number of eigenvalues of $\tilde{L}_{\text{nrm}}$ that are less than 0.5 are two. Hence, $\tilde{m} = 2$. The number of items in views is either two (for view 1) or one (for the rest of views). Thus, the estimated size of universe is obtained as $\hat{m} = 2$.

• *Step 3:* Matrix U consisting of the first two eigenvectors of $\tilde{L}_{\text{nrm}}$ is computed.

• *Step 4:* Rows of U are normalized to obtain (up to two decimals)

$$U = \begin{bmatrix} -0.94 & 0.34 \\ 0.47 & 0.88 \\ -0.88 & 0.48 \\ 0.07 & 0.99 \\ 0.47 & 0.88 \\ 0.47 & 0.88 \\ 0.47 & 0.88 \end{bmatrix}. \tag{10}$$

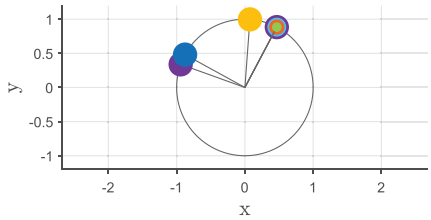


Fig. 6. Embedding of rows of matrix U in Example 1.

Fig. 6 depicts rows of (10) as vectors, where the endpoint of each vector is colored based on the view that it corresponds to, and the unit circle is drawn to indicate that rows have unit norm. The pivots are chosen by taking the first row as the first pivot $u'_1 = [-0.94,\ 0.34]$. The second pivot is chosen as the row of U that has the smallest (absolute value of) inner product with u'_1, which gives $u'_2 = [0.47,\ 0.88]$.

From $(F^i)_{jk} \stackrel{\text{def}}{=} \|u_j - u'_k\|^2$, where u_j are rows of U that correspond to view i and u'_k are pivot rows, we obtain

$$F^1 = \begin{bmatrix} 0 & 2.28 \\ 2.28 & 0 \end{bmatrix}, F^2 = \begin{bmatrix} 0.02 & 1.97 \end{bmatrix},$$
$$F^3 = \begin{bmatrix} 1.46 & 0.17 \end{bmatrix}, F^4 = \begin{bmatrix} 2.28 & 0 \end{bmatrix},$$
$$F^5 = \begin{bmatrix} 2.28 & 0 \end{bmatrix}, \quad F^6 = \begin{bmatrix} 2.28 & 0 \end{bmatrix}.$$

By solving a linear assignment problem for each F^i as the cost matrix (which aims to find the permutation matrix P^i such that $\langle P^i, F^i \rangle$ is minimized), we obtain lifting permutations

$$P^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, P^2 = \begin{bmatrix} 1 & 0 \end{bmatrix}, P^3 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$
$$P^4 = \begin{bmatrix} 0 & 1 \end{bmatrix}, P^5 = \begin{bmatrix} 0 & 1 \end{bmatrix}, P^6 = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

• *Step 5:* Cycle-consistent pairwise associations are obtained by $P^i_j = P^i P^j$. Note that these associations correspond to the graph $\mathcal{G}_3$ in Fig. 4.

[2]Specifically, u_j denotes rows $\sum_{k=1}^{i-1} m_k + 1$ through $\sum_{k=1}^{i} m_k$ of U.

VI. Theoretical Justifications

In this section, we discuss the insights and theoretical justifications behind steps of the CLEAR algorithm. To improve the readability, proof of all lemmas and propositions are presented in the Appendix.

The discrete and combinatorial nature of the multiway data association problem makes finding the optimal solution computationally prohibitive in practice. Hence, similar to the state-of-the-art methods, the CLEAR algorithm aims to find a suboptimal solution via a series of approximations of the original problem.

A. Step 1: Reformulation

Before proceeding with obtaining an approximate solution, we reformulate (4) to obtain an equivalent problem. This equivalent problem, given in the following proposition, is amenable to a relaxation, which grants us an approximate solution in a computationally tractable manner.

Proposition 2: Problem 1 is equivalent to

$$\underset{U\in\mathbb{U}}{\text{minimize}}\ \operatorname{tr}\left(U^\top \tilde{L}_{\text{nrm}}\, U\right) \tag{11}$$

where $\mathbb{U} \overset{\text{def}}{=} \{U : U = C^{-\frac{1}{2}}V,\ V \in \mathbb{V}\}$, $\mathbb{V}$ is defined as the set of all matrices of form (3), and $U^\top U = I$.

Remark 2: The skew-symmetric part of $\tilde{L}_{\text{nrm}}$ does not affect the solution of (11) since for all U and any skew-symmetric matrix B, $\operatorname{tr}(U^\top B\, U) = 0$. This observation justifies using only the symmetric part of $\tilde{L}_{\text{nrm}}$ in step 1 of the CLEAR algorithm.

B. Step 2: Estimating Size of Universe

From (7) and (8), CLEAR obtains an estimate for the size of universe based on the spectrum of $\tilde{L}_{\text{nrm}}$. By definition, the size of universe is the total number of unique items observed in all views (e.g., the size of universe in Fig. 3 is five), which essentially determines the number of columns of U in (11) [or equivalently V in (4)]. This approach is justified in the following analysis, which aims to show that, under certain bounded noise regimes, the estimated size $\hat{m}$ is guaranteed to be identical to its true value m. Let us denote the ground truth association graph by $\mathcal{G}$. Note that $\mathcal{G}$ consists of m clusters, each representing an item of the universe.

Lemma 1: If L_{nrm} is the normalized Laplacian matrix of the cluster graph $\mathcal{G}$, then $\operatorname{eig}(L_{\text{nrm}})$ consists of zeros and ones. Moreover, the multiplicity of the zero eigenvalues is the number of clusters.

Lemma 1 implies that in the noiseless setting, the number of zero eigenvalues of L_{nrm} is the size of universe, which is correctly recovered from (8) by counting the eigenvalues that are less than 0.5. We now consider the noisy association graph $\tilde{\mathcal{G}}$ with normalized Laplacian $\tilde{L}_{\text{nrm}} = L_{\text{nrm}} + N$, where N is a symmetric matrix that represents the noise. Here, the symmetry assumption follows from using only the symmetric component of $\tilde{L}_{\text{nrm}}$ in the algorithm (see Remark 2).

Lemma 2: Consider the estimate $\tilde{m}$ obtained by (8) from $\tilde{L}_{\text{nrm}} = L_{\text{nrm}} + N$. If $\|N\| < 0.5$, then $\tilde{m} = m$.

Lemma 2 implies that, under a bounded noise regime, the estimated size of universe is equal to the true value. In order to obtain a bound in terms of the number of wrong associations for which $\tilde{m} = m$ is guaranteed, let us consider a noise model where $\tilde{C} = C$. In this model, correct associations are potentially replaced with wrong ones; however, the degrees of vertices in $\mathcal{G}$ and $\tilde{\mathcal{G}}$ remain the same. Let $\tilde{A} = A + E$, where A and $\tilde{A}$ are respectively the adjacency matrices of $\mathcal{G}$ and $\tilde{\mathcal{G}}$, and $E \in \{-1, 0, 1\}^{l\times l}$ represents the noise. Further, let $e_{\max} \overset{\text{def}}{=} \max\{e_1, e_2, \ldots, e_l\}$, where $e_i \overset{\text{def}}{=} \sum_{j=1}^{l} |(E)_{ij}|$ denotes the number of wrong associations at vertex i of the graph $\tilde{\mathcal{G}}$. Let $c_{\min} > 0$ denote the smallest diagonal entry of the C matrix.

Proposition 3: Given $e_{\max}$, $c_{\min}$ defined above and $\tilde{m}$ obtained from (8), if $e_{\max} < 0.5\, c_{\min}$, then $\tilde{m} = m$.

Proposition 3 implies that when the noise magnitude (in terms of the number of mismatches) is sufficiently small, the estimated size of universe $\hat{m}$ is equal to the true value m. We point out that, in practice, the bound in Proposition 3 is conservative and correct estimates may be obtained in larger noise regimes or for noise with a more realistic model. In higher noise regimes where the estimate can have a large error, taking the maximum in (7) ensures the distinctness constraint (which implies that items in each view are unique), and therefore the estimated m cannot be less than the maximum number of items observed at a view.

The estimated value of m obtained from (7) fixes the size of U in (11) throughout the algorithm. Since (as we will show) each iteration of the CLEAR algorithm has a small execution time, instead of using a fixed value an alternative approach is to consider multiple values for $\hat{m}$ (e.g., by looping over all feasible $\hat{m}$) and choosing the solution that maximizes the objective in (4). In our empirical evaluations, we observed that this approach, which comes at the expense of a higher execution time, does not notably improve the accuracy of the results. This empirical observation hints that the estimated value of $\hat{m}$ is often close to its optimal value, advocating the proposed estimation approach.

Remark 3: In the spectral graph clustering methods, the "eigengap" heuristic is often used to estimate the number of clusters [21]. In this approach, $\tilde{m}$ is chosen such that $|\lambda_{\tilde{m}} - \lambda_{\tilde{m}+1}|$ is maximized, where λ_k's are sorted eigenvalues of $L_{\text{sym}} \overset{\text{def}}{=} D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. Unlike the normalized Laplacian L_{nrm} proposed in this work (see Lemma 1), in the noiseless setting, the nonzero eigenvalues of L_{sym} depend on the size of clusters. We believe that this can make the eigengap method more sensitive to noise. As we will see in our empirical analysis in Section VIII, the estimated universe size based on the eigengap heuristic can deviate considerably from the true value in moderate noise regimes, while our approach exhibits more robustness.

C. Step 3: Lifting and Relaxation

In order to solve (11) in a computationally tractable manner, the second approximation used in the CLEAR algorithm is to drop the discrete constraint $U \in \mathbb{U}$ and allow U to take values in $\mathbb{R}^{l\times m}$. This leads to the relaxed problem

$$\begin{aligned} &\underset{U\in\mathbb{R}^{l\times m}}{\text{minimize}} && \operatorname{tr}\left(U^\top \tilde{L}_{\text{nrm}}\, U\right) \\ &\text{subject to} && U^\top U = I \end{aligned} \tag{12}$$

which is a generalized Rayleigh quotient problem. From the Rayleigh–Ritz theorem [23, Sect. 5.2.2], it follows that the solution of (12) is given by the eigenvectors corresponding to the m-smallest eigenvalues of $\tilde{L}_{\text{nrm}}$ (note that m is estimated in the previous step).

We point out that the relaxation technique used here is similar to the relaxation used to solve the normalized minimum-cut problem in the spectral graph clustering literature [21]. This similarity is not surprising given the graph–theoretic interpretation of our problem discussed in Section III-B. Nevertheless, note that spectral graph clustering is based on $\tilde{L}_{\text{sym}} \stackrel{\text{def}}{=} \tilde{D}^{-\frac{1}{2}} \tilde{L} \tilde{D}^{-\frac{1}{2}}$ (or other normalized Laplacians) instead of $\tilde{L}_{\text{nrm}}$.

D. Step 4: Projection and Embedding

In order to obtain an approximate solution for the original problem (11), the solution $U^* \in \mathbb{R}^{l \times m}$ obtained from solving (12) should be projected back to the discrete set $\mathbb{U}$. This step is critical for ensuring the cycle consistency and distinctness constraints. In fact, as we will show in Section VIII, the solutions returned by some state-of-the-art methods could violate the cycle consistency or distinctness constraints in high-noise regimes due to bad projections.

To project U^* onto $\mathbb{U}$, several approaches can be considered. Our approach is inspired by the spectral graph clustering literature [21], [24], [25], where rows of U^* are normalized and embedded as points in $\mathbb{R}^m$. These points are then grouped into m disjoint sets based on their distance to cluster centers. Despite the aforementioned similarity, a key difference in our setting is the existence of the distinctness constraint (i.e., vertex coloring), which is not present in spectral graph clustering [25]. Hence, the k-means algorithm commonly used for grouping the embedded points in general violates the distinctness constraint. Furthermore, compared to other projection techniques that consider this constraint (e.g., methods in [16], [26]), our approach has a lower complexity that leads to superior execution time.

Our approach is based on noting that rows of V [defined in (3)] consist of standard bases vectors which are orthogonal. Furthermore, as explained earlier, V identifies graph clusters $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$, where vertices that belong to the same cluster have identical rows in V. Since $U \stackrel{\text{def}}{=} C^{-\frac{1}{2}} V$ and C is a diagonal matrix, it follows that in the *noiseless setting*, the set of *normalized* rows of U consists exclusively of m orthogonal vectors. Additionally, similar to V, normalized rows of U that are identical correspond to vertices that belong to the same cluster.

In the noisy setting, from the Davis–Kahan theorem [27], the eigenspace of the ground truth Laplacian matrix and its noisy version are "close" to each other (where "closeness" can be quantified by the noise magnitude, cf. the discussion in [21, Sect. 7]). Hence, in modest noise regimes, the rows of U^* that belong to the same clusters are expected to remain close (in terms of the Euclidean distance) and almost perpendicular to other rows. This observation is leveraged by choosing m rows of U^* that are most orthogonal to each other (called pivots) to represent the clusters. The remaining rows are then associated to pivots (while preserving distinctness) based on distance in order to identify which cluster they belong to.

If u_i denotes the ith row of U^*, the problem of finding the m most orthogonal rows can be formulated as finding a subset $\mathcal{S}$ of rows that solves

$$\begin{aligned} \underset{\mathcal{S} \subset \mathbb{N}_l}{\text{minimize}} \quad & \sum_{i,j \in \mathcal{S}} |\langle u_i, u_j \rangle| \\ \text{subject to} \quad & |\mathcal{S}| = m. \end{aligned} \tag{13}$$

The greedy strategy explained in step 3 of the CLEAR algorithm can be leveraged to efficiently find an approximate solution for (13).

After choosing the pivot rows, which are denoted by u'_k and represent clusters, the remaining rows of U^* are assigned to pivot rows through minimizing the within-cluster distances. This is formally stated as

$$\begin{aligned} \underset{\mathcal{A}_1, \ldots, \mathcal{A}_m}{\text{minimize}} \quad & \sum_{k=1}^{m} \sum_{v_j \in \mathcal{A}_k} \|u_j - u'_k\|^2 \\ \text{subject to} \quad & g(v_i) \neq g(v_j), \quad \forall v_i, v_j \in \mathcal{A}_s. \end{aligned} \tag{14}$$

The constraint in (14) enforces the distinctness constraint (i.e., items in a view should not be in the same cluster). Let us define $F \in \mathbb{R}^{l \times m}$ such that $(F)_{jk} \stackrel{\text{def}}{=} \|u_j - u'_k\|^2$, and denote its row blocks by

$$F = \begin{bmatrix} F^{1\top} & F^{2\top} & \cdots & F^{n\top} \end{bmatrix}^\top \tag{15}$$

where the number of rows of block F^i is equal to the number of items at view i. Using this notation, and since V encapsulates both the distinctness constraint and the cluster structure,[3] (14) can be represented in matrix form as $\min_{V \in \mathbb{V}} \langle V, F \rangle$. Noting that

$$\min_{V \in \mathbb{V}} \langle V, F \rangle = \min_{P^i \in \mathbb{P}^L} \textstyle\sum_{i=1}^{n} \langle P^i, F^i \rangle \tag{16a}$$

$$= \textstyle\sum_{i=1}^{n} \min_{P^i \in \mathbb{P}^L} \langle P^i, F^i \rangle \tag{16b}$$

and since each subproblem in (16b) is a linear assignment problem [5], the optimal solution can be obtained by, e.g., applying the Hungarian (Kuhn–Munkres) algorithm on each block F^i.

From the implementation point of view, as long as the lifting permutation structure of P^i is preserved, faster suboptimal methods can be used instead to solve (16b). To improve the runtime, instead of the Hungarian algorithm, CLEAR uses a suboptimal greedy strategy based on sorting, where the index of the smallest entries of F^i are used to determine the index of one entries in P^i. These indices are chosen with care to ensure that P^i is a lifting permutation (i.e., each row has a single one entry and each column has at most a single one entry). In our empirical evaluations, we observed that this suboptimal strategy performs as well as the optimal Hungarian algorithm most of the time in terms of accuracy, but has a considerable speed advantage.

Finally, we emphasize that the proposed projection technique is based on the orthogonality property of the embedded rows. Hence, the results are not affected by any transformation that preserves the orthogonality. This is particularly important since

[3] If the jth entry in column k of V is nonzero, then $v_j \in \mathcal{A}_k$.

solutions of (12) are only recovered up to an orthogonal transformation (i.e., if U^* is a solution so is U^*R for any orthogonal matrix R).

E. Computational Complexity

The computational complexity of CLEAR is determined by the eigendecomposition algorithm (used for estimating the universe size and computing the eigenvectors of $\tilde{L}_{\text{nrm}}$) and the linear assignment problem (used for the projection step). The time complexity of the eigendecomposition and optimal linear assignment (e.g., Hungarian algorithm) are, respectively, $\mathrm{O}(l^3)$ and $\mathrm{O}(n\,m^3)$, where l is the number of vertices in the assignment graph, n is the number of views, and m is the size of universe.

In order to improve the speed and scalability of CLEAR, a BFS, which has the worst computational complexity of $\mathrm{O}(l^2)$, can be used to first find the connected components of $\tilde{\mathcal{G}}$. The spectrum (i.e., eigenvalues of normalized Laplacian) of $\tilde{\mathcal{G}}$ is then obtained by taking the disjoint union of components' spectra (similarly, eigenvectors are given by zero padding the components' eigenvectors) [28]. Through this approach, the complexity of the eigendecomposition is reduced to $\mathrm{O}(l_1^3)$, where l_1 is the number of vertices in the largest connected component of $\tilde{\mathcal{G}}$. In practice, often the association graph consists of many disjoint components (e.g., see examples in Section IX), and the aforementioned procedure considerably improves the runtime and scalability.

The second improvement in speed is achieved by replacing the Hungarian algorithm with the suboptimal sorting strategy. This approach reduces the computational complexity of the projection step to $\mathrm{O}(n\,m^2\log(m))$.

VII. Applications: Edge-Centric vs. Clique-Centric

In this section, we divide the applications that benefit from solving the multiview matching problem into two categories, namely *edge-centric* and *clique-centric*. It will become clear shortly that making this subtle distinction is crucial for choosing the appropriate evaluation metric for each category.

In edge-centric applications, one ultimately seeks to establish associations between *pairs of views* (and not *all* views). In graph terms, this corresponds to seeking individual edges of the association graph (hence the name). For example, using multiview matching algorithms to associate features between multiple images for estimating relative transformation between the corresponding pair of camera poses [2] belongs to this category. The purpose of using multiview matching techniques in such applications is to *refine* the initial noisy associations by incorporating information from multiple views and enforcing cycle consistency. Based on this definition, even a cycle-*inconsistent* set of associations is still a *feasible* (although erroneous) solution in edge-centric applications. As a result, computing standard metrics such as precision/recall based on *individual* edges of the association graph is appropriate for evaluating the performance of multi-view association algorithms in such applications.

By contrast, in what we refer to as clique-centric applications, one ultimately seeks to fuse information *globally* (i.e., across *all* views) as prescribed by the cliques of the association graph.

Fig. 7. Evaluating the performance of cycle-inconsistent solutions (e.g., $\mathcal{G}_1$) for clique-centric applications must be done *after* completing the connected components of the association graphs (i.e., for the transitive closure $\mathcal{G}_2$). Even a single incorrect edge (drawn in red) may have catastrophic consequences in clique-centric applications.

For example, consider the map fusion problem that arises in single/multi-robot SLAM [4]. After identifying every sighting of each unique landmark in all maps (i.e., encoded in the cliques of a cycle-consistent association graph) via multiview matching techniques, the corresponding measurements (across *all* maps) must be fused together in the SLAM back-end to generate the global map. Note that such notion of *global* fusion is well-defined only if association, as a binary relation on the set of observations, is an equivalence relation.[4] Therefore, unlike edge-centric applications, cycle consistency of associations is a must in clique-centric applications where the observations in each equivalence class are fused together. Cycle-inconsistent solutions must therefore be made cycle-consistent before being used in such applications. An implicit and natural way of doing this is via the so-called transitive closure of associations which gives the smallest equivalence relation containing the original associations. In graph terms, this is equivalent to completing each connected component of the association graph into a clique. Thus, evaluating such cycle-inconsistent solutions by computing precision/recall based on individual edges can be highly misleading in the case of clique-centric applications. In such cases, precision/recall must be computed *after* completing the connected components of the association graph (i.e., for the transitive closure).

Note that a single incorrect association only affects local (pairwise) fusions in edge-centric applications, while it may have a catastrophic global impact in clique-centric domains. This is illustrated in Fig. 7 using a simple example. Here, the association graph $\mathcal{G}_1$ contains only a single incorrect edge drawn in red. Although $\mathcal{G}_1$ has a high precision and a high recall for edge-centric applications, it is not cycle-consistent and thus does not immediately prescribe a valid solution to clique-centric applications. As mentioned above, for such applications, one must first compute the transitive closure of $\mathcal{G}_1$. The transitive closure of $\mathcal{G}_1$ is given by $\mathcal{G}_2$ which performs poorly in terms of precision. Note that each red edge in $\mathcal{G}_2$ indicates an incorrect fusion of two observations.

Although CLEAR, by construction, always returns cycle-consistent solutions, as we will see in the following sections, several existing algorithms may violate cycle consistency in noisy regimes. It is thus crucial to be aware of the distinction between local (pairwise) and global fusion in order to use the appropriate performance metric in a particular application.

[4]This mainly refers to transitivity since for all practical purposes in robotics, associations are always reflexive and symmetric.

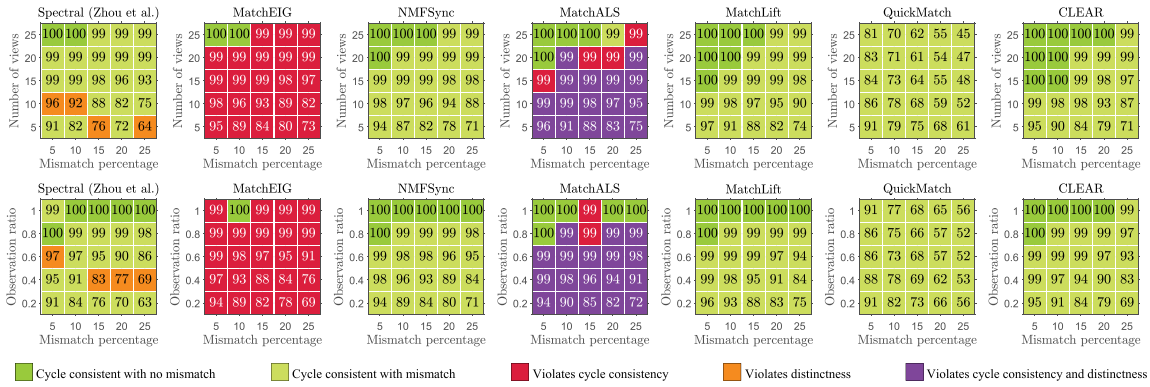


Fig. 8. Comparison of the state-of-the-art algorithms with CLEAR for uniformly sampled observations. Various number of views and observation ratios versus the percentage of mismatch in the input are considered. The F_1 score is reported in percentage in each grid (the higher the better). These values are computed based on individual edges in the association graph (i.e., for edge-centric applications); see Section VII.

VIII. Simulation Results

In this section, we use Monte–Carlo analysis with synthetic data to compare CLEAR with several state-of-the-art algorithms across different noise regimes. The aim of these comparisons is to 1) analyze the accuracy of the returned solutions; 2) identify algorithms that violate the cycle consistency or distinctness constraints in high-noise regimes; and 3) evaluate the accuracy of the proposed technique for estimating the universe size.

Algorithms used in our comparisons, which span across three aforementioned domains, include 1) MatchLift [10] and MatchALS [2] that are based on a convex relaxation; 2) spectral algorithm [1] extended for partial permutations by Zhou *et al.* [2], MatchEig [15], and NMFSync [16] that are based on a spectral relaxation; 3) and QuickMatch [3] that is a graph clustering approach.

We consider scenarios with various number of views and observations across different mismatch percentage in the pairwise correspondences. The mismatch in correspondences is introduced by randomly reassigning correct matches to wrong ones according to a uniform distribution. In all comparisons, the universe is set to contain 100 items, where this value is assumed to be unknown to algorithms and should be estimated. For algorithms that require the knowledge of universe size (all except QuickMatch), the same estimated value obtained for CLEAR from (7) is used.

We report the F_1 score, which is commonly used in the literature and is defined as $f \stackrel{\text{def}}{=} \frac{2pr}{p+r} \in [0,1]$, to evaluate the performance of the algorithms. Here, precision $p \in [0,1]$ is defined as the number of correct associations divided by the total number of associations in the output, and recall $r \in [0,1]$ is the number of correct associations in the output divided by the total number of associations in the ground truth. The best performance is achieved when $f = 1$ (when $p = q = 1$) and the worst when $f = 0$ (zero precision and/or zero recall).

In the first comparison, the algorithms are evaluated for different number of views and percentage of mismatch in the input. The observation ratio is fixed at 0.5; i.e., in each view, 50 (out of 100) items of the universe are observed. These items are sampled uniformly at random. For each number of views and mismatch percentage, 10 Monte–Carlo simulations are generated and the average F_1 score of the outputs across these simulations is reported in the first row of Fig. 8 (in percentage). In the second comparison (second row in Fig. 8), the number of views in all Monte–Carlo simulations is fixed at the value of $n = 10$, and results for various observation ratios of universe items and input mismatch percentage are reported. Similar to the first comparison (first row in Fig. 8), each observation ratio indicates the number of items that were observed (i.e., uniformly sampled at random) in a view. For example, observation ratio of 0.2 indicates that each agent observed 20 (out of 100) items of the universe.

Fig. 8 shows that for a fixed observation ratio, as the number of views increases, the F_1 score also increases. This indicates that the algorithms are able to leverage the redundancy in observations with the help of the cycle-consistency constraint. For the same reason, for a fixed number of views, the F_1 score improves as the observation ratio increases.

We also tested the returned solutions for cycle consistency (transitive associations) and distinctness (two observations in a view cannot be associated to each other). The results are displayed using colors in Fig. 8. In particular, here dark green indicates that the (cycle consistent) ground truth was recovered in all Monte–Carlo iterations. Light green indicates that the returned solutions satisfied cycle consistency and distinctness, but contained wrong associations in at least one of the simulations. Furthermore, red indicates that, in at least one simulation, the output was not cycle-consistent, orange indicates violation of the distinctness constraint, and, finally, purple indicates violation of both cycle consistency and distinctness constraints.

In addition, Fig. 8 demonstrates that the extended spectral algorithm, MatchEig, and MatchALS may return results that violate the cycle consistency and/or distinctness constraints in moderate to high noise regimes. Recall from Section VII that although a cycle-inconsistent solution may exhibit a high F_1 score in terms of individual associations, in clique-centric applications, its F_1 score can dramatically decrease after completing the connected components of the association graph (i.e., transitive closure). This is demonstrated in Fig. 9 for MatchEIG and MatchALS algorithms (compare Fig. 9 with Fig. 8). For example, the average F_1 score of MatchEIG with 10 views and under 15% mismatch drops from 0.93 (Fig. 8) to 0.08 (Fig. 9).

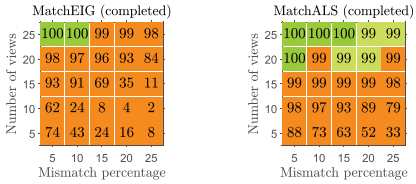


Fig. 9. Average F_1 score of the inconsistent algorithms after making them cycle-consistent by completing the graph's connected components for clique-centric applications (see Section VII).

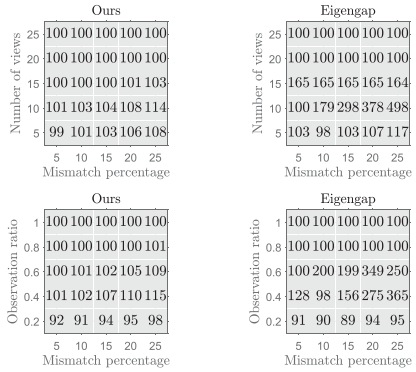


Fig. 10. Average of estimated universe sizes in the Monte–Carlo runs of Fig. 8 by CLEAR and the eigengap method based on the symmetric Laplacian. The closer to 100, the better.

As discussed in Section VII, here the F_1 score of 0.93 can be very misleading if the solution obtained by the algorithm is going to be used for fusion in the context of clique-centric applications.

Among the algorithms that do not violate the consistency and distinctness constraints, on average, MatchLift, NMFSync, and CLEAR have the highest F_1 scores. The poor performance of QuickMatch is mainly due to the fact that this algorithm was originally designed and tuned for matching image features based on *weighted* associations, whereas in our setting, the associations are binary.[5] In conclusion, synthetic comparisons demonstrate that CLEAR returns cycle-consistent solutions with high F_1 scores. In the next section, we evaluate the runtime and scalability of the algorithms in real-world examples, where the total number of observations can reach several thousands.

Finally, we compare the estimated size of universe, obtained from (7), with the eigengap method commonly used in the spectral graph clustering literature (see Remark 3). The results are reported in Fig. 10. The number written inside each square is the average of estimated universe sizes (rounded) in the Monte–Carlo runs of Fig. 8. The correct universe size is 100. According to the results depicted in Fig. 10, although both techniques perform equally well under a high signal-to-noise ratio (top two rows in each figure), the proposed approach is more robust to noise and significantly outperforms the standard eigengap heuristic (bottom three rows in each figure).

[5]Nonetheless, it is straightforward to generalize CLEAR and other algorithms to the weighted case.

IX. Experimental Results

To further evaluate the accuracy and speed of CLEAR in real-world robotics applications, we consider two scenarios, namely multi-image feature matching and map fusion in landmark-based SLAM. Feature matching datasets have become standard benchmarks for comparing the performance of multi-way data association algorithms. Hence, we report the results on two publicly available standard benchmark datasets, namely Graffiti[6] and CMU Hotel.[7] The aim of our experimental comparisons is to 1) compare the runtime of algorithms; and 2) evaluate the precision/recall for the returned solutions.

A. CMU Dataset

The CMU Hotel dataset consists of 101 images. The ground truth provided by this dataset consists of 30 feature points per image and their correct associations. These feature points are visible across all images, leading to a total of 3030 features across all images. Due to the large ratio of the number of images (101 images) to the number of feature points per image (30 features), this dataset represents scenarios where observations have high redundancy. To obtain the input for algorithms, we compute the SIFT descriptor [29] of each feature point using the VLFeat library[8] [30]. The standard `vl_ubcmatch` routine in VLFeat is used to match feature points across image pairs based on the Euclidean distance between their descriptor vectors. By taking this input (as a 3030×3030 aggregate association matrix), each algorithm returns an output which is then compared with the ground truth to evaluate its accuracy. We further record the execution time of each algorithm. All results are based on Matlab implementation of algorithms on a machine with an Intel Core i7-7700 K CPU @ 4.20 GHz and 16 GB RAM.

Fig. 11 shows an example of three images in the CMU Hotel sequence, where feature points and their associations across images are shown for the input and the output of four algorithms. Note that the input associations, which are obtained by matching features on image pairs, are cycle inconsistent and contain errors. The output of the algorithms should ideally identify and remove these errors based on the cycle consistency principle.

Fig. 12 reports the precision (i.e., number of correct matches divided by the total number of returned matches) versus the rate of the solutions returned by algorithms. The rate (i.e., the inverse of execution time) indicates the number of times an algorithm can run in one second. Due to the large difference between the runtimes of the algorithms, the rate axis is scaled logarithmically. The precision of the input is indicated by the orange line on the plot and approximately has the value of 0.92. Note that this value is calculated based on individual edges and thus is only meaningful for edge-centric applications; see Section VII. Solutions that were not cycle consistent are colored in red. An ideal algorithm should have a high rate (i.e., small runtime) and a high-precision output (i.e., based on individual edges and for edge-centric applications). Among the cycle-consistent algorithms, QuickMatch is the fastest; however, the returned

[6][Online]. Available: http://www.robots.ox.ac.uk/∼vgg/research/affine/
[7][Online]. Available: http://pages.cs.wisc.edu/∼pachauri/perm-sync/
[8][Online]. Available: http://www.vlfeat.org/

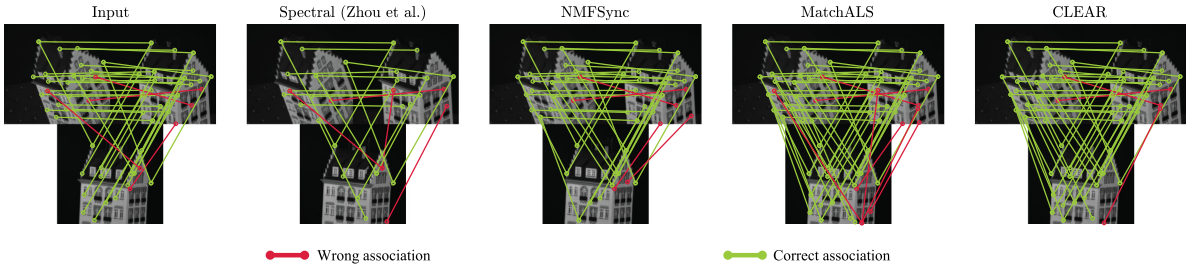


Fig. 11. Example of matched feature points across three images of the CMU Hotel dataset. Input, obtained by matching features across image pairs independently, contains error and is inconsistent. CLEAR returns cycle-consistent results and improves the precision of the input.

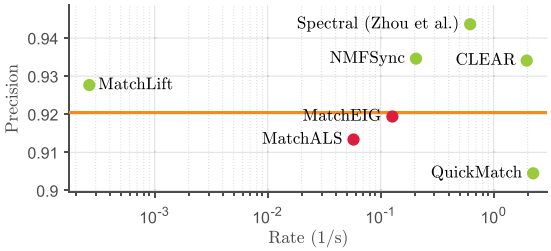


Fig. 12. Precision vs. rate (the inverse of execution time) in CMU Hotel dataset. The rate axis has a logarithmic scale (CLEAR is about 3× faster than Spectral, 10× faster than NMFSync, and 7500× faster than MatchLift). The precision of the input (based on individual edges and for edge-centric applications) is denoted by the orange line; see Section VII. Cycle-consistent/inconsistent outputs are, respectively, denoted by ● and ●. The closer to the top-right corner, the better.

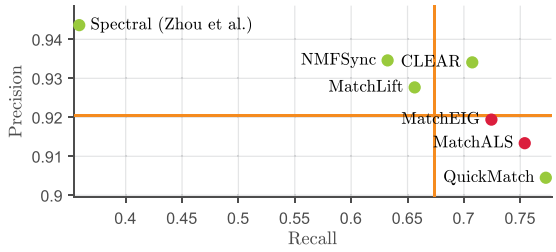


Fig. 13. Precision vs. recall in CMU Hotel dataset. Precision and recall of input (based on individual edges and for edge-centric applications) are denoted by the orange lines; see Section VII. The closer to the top-right corner, the better.

solution does not improve the precision of the input. CLEAR, Spectral, NMFSync, and MatchLift algorithms improve the precision, while CLEAR has a higher rate: CLEAR is about 3× faster than Spectral, 10× faster than NMFSync, and 7500× faster than MatchLift.

The faster runtime of CLEAR is due to 1) the structure of the input association graph, which consists of several disjoint connected components (this graph consists of 81 connected components, where the largest component has 297 vertices). This structure is exploited by the proposed eigendecomposition approach, which uses the BFS algorithm to find the spectrum of the graph as the union of its connected components' spectra. 2) The projection technique uses a suboptimal sorting strategy (instead of, e.g., the Hungarian algorithm) to improve the speed while ensuring consistency and distinctness. More specifically, running CLEAR with the Hungarian algorithm results in the same output (i.e., the same value for precision and recall); however, the execution time increases from 0.5 to 0.7 s.

Fig. 13 reports the precision and recall of returned solutions. An ideal solution simultaneously has high precision and recall. The output of the Spectral algorithm has the highest precision and lowest recall. On the other hand, the output of QuickMatch has the highest recall and lowest precision. In comparison, the output of CLEAR shows a balanced precision versus recall.

We note that the precision and recall of MatchEig after making its solution cycle consistent by completing the association graph's connected components (Section VII) become 0.67 and 0.8, respectively. Similarly, MatchALS's output after completion takes the precision and recall of 0.73 and 0.76, respectively. This sharp drop in precision underlines the importance of taking cycle consistency into account in evaluating multiview matching algorithms for clique-centric applications.

B. *Graffiti Dataset*

The Graffiti dataset consists of six images, each taken from a different viewpoint of a textured planar wall. Due to the large difference between the viewpoints, this dataset is particularly challenging for feature point detection/matching algorithms (thus, pairwise associations have a lower precision compared to the CMU hotel dataset). The dataset provides ground truth homography transformations between the viewpoints. We use the VLFeat library to extract the SIFT feature points for each image. To obtain the ground truth associations, the provided homography matrices are used to match the extracted features (correct matches must satisfy the planar homography mapping [31, see (5.35)]). To make sure that ground truth associations are error-free, we only take feature points and associations that are cycle consistent across all images and discard the rest. These associations are further visually inspected to ascertain that they do not contain mismatches. The number of feature points retained after this process ranges from 313 to 657 per image. The total number of feature points across all images is 3176. Unlike the CMU hotel dataset, the Graffiti dataset has a small ratio of the number of images to the number of feature points per image. Thus, it represents scenarios where observations have little to no redundancy.

The precision and rate of algorithms are reported in Fig. 14. Among the cycle-consistent algorithms, QuickMatch is the fastest; however, it does not improve the precision of the input computed based on individual edges and for edge-centric applications (Section VII). CLEAR improves the precision and is considerably faster compared to the other algorithms that

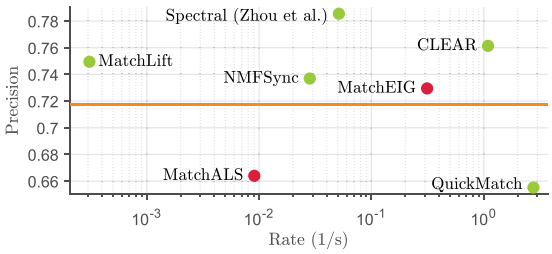


Fig. 14. Precision vs. rate (the inverse of execution time) in Graffiti dataset. The rate axis has a logarithmic scale (CLEAR is about 21× faster than Spectral, 39× faster than NMFSync, 3800× faster than MatchLift). Precision of the input is denoted by the orange line. Cycle-consistent and -inconsistent outputs are, respectively, denoted by ● and ●. The closer to the top-right corner, the better.

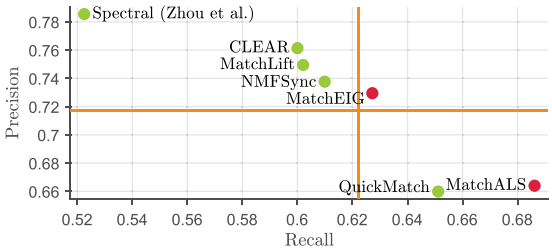


Fig. 15. Precision vs. recall in Graffiti dataset. Precision and recall of input are denoted by the orange lines. The closer to the top-right corner, the better.

improve the input's precision: about 21× faster than Spectral, 39× faster than NMFSync, and 3800× faster than MatchLift.

In the Graffiti dataset, the input association graph consists of 1506 connected components, where the largest component has 22 vertices. Running CLEAR with the Hungarian algorithm results in an output with the same value for precision and recall (up to three decimals); however, the execution time of the algorithm increases considerably from 0.92 to 49.5 s.

The precision and recall of returned solutions are reported in Fig. 15. Among cycle-consistent algorithms, the Spectral algorithm has the highest precision and lowest recall, while QuickMatch has the highest recall and lowest precision. In comparison, CLEAR, MatchLift, and NMFSynch have a balanced precision versus recall. The precision and recall of MatchEig after making its solution cycle consistent (for clique-centric applications) become 0.53 and 0.69, respectively. Similarly, MatchALS's output after completion takes the precision and recall of 0.54 and 0.69, respectively. Once again, the difference between these values and those reported in Fig. 15 highlights the importance of taking cycle consistency into account in evaluating multiview matching algorithms for clique-centric applications.

C. Forest Landmark-Based SLAM Dataset

Map fusion is an important clique-centric application of the multiview matching problem in single/multi-robot SLAM [4]. The goal in this problem is to identify unique landmarks across a given set of local maps (created by one or multiple robots) in order to fuse the corresponding measurements in the landmark-based SLAM back-end.[9] In this section, we report the

Fig. 16. Single UAV autonomous exploration at NASA LaRC. The vehicle (highlighted in red) performs landmark-based SLAM based on detected trees in order to estimate its position within the forest.

performance of CLEAR in the context of map fusion based on a SLAM dataset collected in the forest at the NASA Langley Research Center (LaRC) [32]. In this dataset, a single unmanned aerial vehicle equipped with an inertial measurement unit (IMU) and a 2D LIDAR is tasked with autonomously exploring an area under the tree canopy (Fig. 16). The exploration mission lasts 120 s. As the vehicle traverses the forest, it performs LIDAR-inertial odometry by fusing IMU measurements with incremental motion estimates from the iterative closest point algorithm at 40 Hz. In addition, the vehicle also uses a customized detector to identify trees from the LIDAR scans at a rate of 1 Hz. The objective is to correctly match and fuse identical tree landmarks detected during the exploration, and subsequently optimize the landmark positions and vehicle trajectory inside a landmark-based SLAM framework.

To obtain the initial pairwise data association, we apply the correspondence graph-matching algorithm [33] that associates two sets of landmarks based on their local configurations. Crucially, we note that this process does not use any global pose estimates, and thus is not affected by drift in the LIDAR-inertial odometry. Due to the presence of spurious detections and the lack of informative descriptors (e.g., SIFT), the initial data association matrix (of dimension 1091×1091) contains many mismatches and is not cycle consistent. We thus call CLEAR and other multiview matching algorithms to achieve cycle consistency. Recall from our discussion in Section VII that map fusion is inherently a clique-centric application. Therefore, we make any inconsistent data associations cycle-consistent by completing the connected components in the association graph (Section VII). In addition, we also introduce a baseline algorithm that directly completes the connected components in the input associations.

Since ground truth data association is not available, we adopt the following alternative performance metrics. A pair of associated trees is classified as either a *definite negative* or a *potential positive*, based on whether their distance as estimated by the LIDAR-inertial odometry is higher than a threshold of 2 m. We note that these definitions are precise assuming that the threshold value of 2 m accounts for the drift in the LIDAR-inertial odometry.[10] Since the number of definite negatives (denoted

[9]Here, each local map represents a "view" in the multiview matching problem. In practice, local maps may represent one or multiple frames, and may be built by one or multiple robots.

[10]Since the vehicle is flying at a low speed (2 m/s) for a relative short amount of time 120 s, we expect the estimation drift at any time is reasonably bounded.

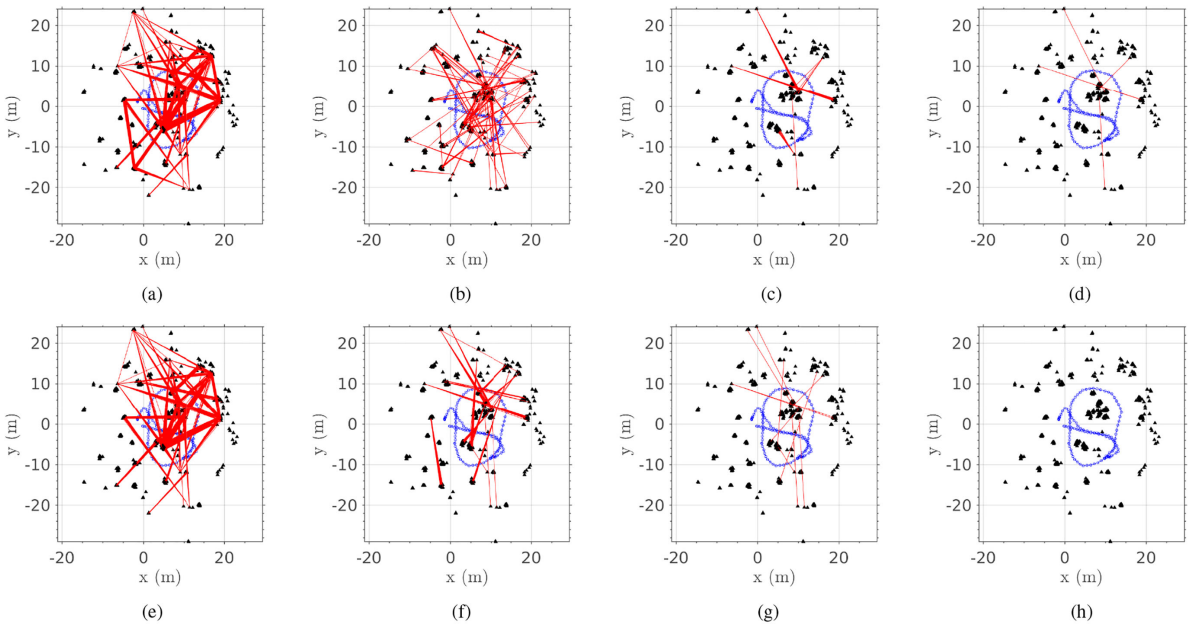


Fig. 17. Output association of baseline (a) and each algorithm (b)–(g) in the landmark-based SLAM dataset. Cycle-inconsistent solutions are completed due the clique-centric nature of the problem (Section VII). Each black triangle represents a single tree observation. *The LIDAR-inertial odometry is shown in blue*. Definite negatives identified using the odometry estimates are highlighted as red edges. We note that the output of CLEAR (g) still contains a few mismatches, but in practice, these can be filtered out by removing small clusters from the returned association, as shown in (h). (a) Baseline. (b) NMFSync. (c) MatchALS (completed). (d) MatchLift. (e) MatchEIG (completed). (f) QuickMatch. (g) CLEAR. (h) CLEAR (postprocessed).

by DN) is an underestimate of the true number of mismatches, and the number of potential positives (denoted by PP) is an overestimate of the true number of correct matches, we can further calculate an upper bound on the true precision as follows:

$$\bar{\mathrm{P}} \stackrel{\text{def}}{=} \frac{\mathrm{PP}}{\mathrm{DN}+\mathrm{PP}}. \tag{17}$$

We note that for landmark-based SLAM, the number of definite negatives (DN) is particularly important, since it is well known that any false data association could inflict catastrophic impact on the final solution. Therefore, an ideal data association should contain no definite negatives, or equivalently achieve a value of 100% for $\bar{\mathrm{P}}$ (upper bound on precision).

Fig. 17 visualizes the data associations returned by each algorithm in the world frame. All definite negatives are highlighted in red. The solutions of MatchALS and MatchEIG are not cycle-consistent initially, and are made cycle-consistent by completing the connected components. The solution of the spectral algorithm is omitted, as it contains significantly more mismatches due to its sensitivity in estimating the universe size. Table II shows the complete set of quantitative results and Fig. 18 shows the precision and rate of each evaluated algorithm.

Due to the existence of mismatches in the input associations, the baseline algorithm which directly completes the connected components yields more than 25 000 definite negatives; see Fig. 17(a). In contrast, most other algorithms are able to significantly reduce the number of definite negatives. Among these algorithms, CLEAR and MatchLift nearly eliminate all definite negatives; see Table II. However, MatchLift requires 124.7 s to converge while CLEAR only takes 0.084 s. The superior speed

TABLE II
CROSS COMPARISON OF ALGORITHMS IN TERMS OF THE NUMBER OF DEFINITE NEGATIVES (DN), POTENTIAL POSITIVES (PP), UPPER BOUND ON PRECISION ($\bar{\mathrm{P}}$), AND RUNTIME. THE UPPER BOUND ON PRECISION IS COMPUTED FROM (17)

Algorithm	DN	PP	$\bar{\mathrm{P}}$ (%)	Runtime (s)
CLEAR	**11**	3393	**99.677**	**0.084**
MatchLift [10]	**5**	2394	**99.792**	124.7
MatchALS [2] (completed)	89	15230	99.419	4.580
QuickMatch [3]	897	15757	94.614	0.118
NMFSync [16]	290	3233	91.768	4.272
MatchEIG [15] (completed)	21415	20381	48.763	1.808
Baseline	26249	**20487**	43.836	N/A

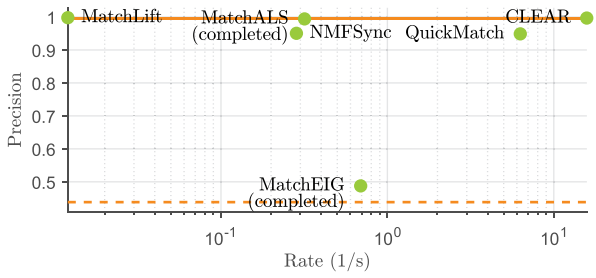


Fig. 18. Precision upper bound $\bar{\mathrm{P}}$ vs. rate in the landmark-based SLAM dataset. The rate axis has logarithmic scale. The closer to the top-right corner, the better. The solid orange line corresponds to the precision of input data associations. The dashed orange line corresponds to the precision of the baseline, obtained by completing the connected components in the input association graph. Further quantitative results are reported in Table II.

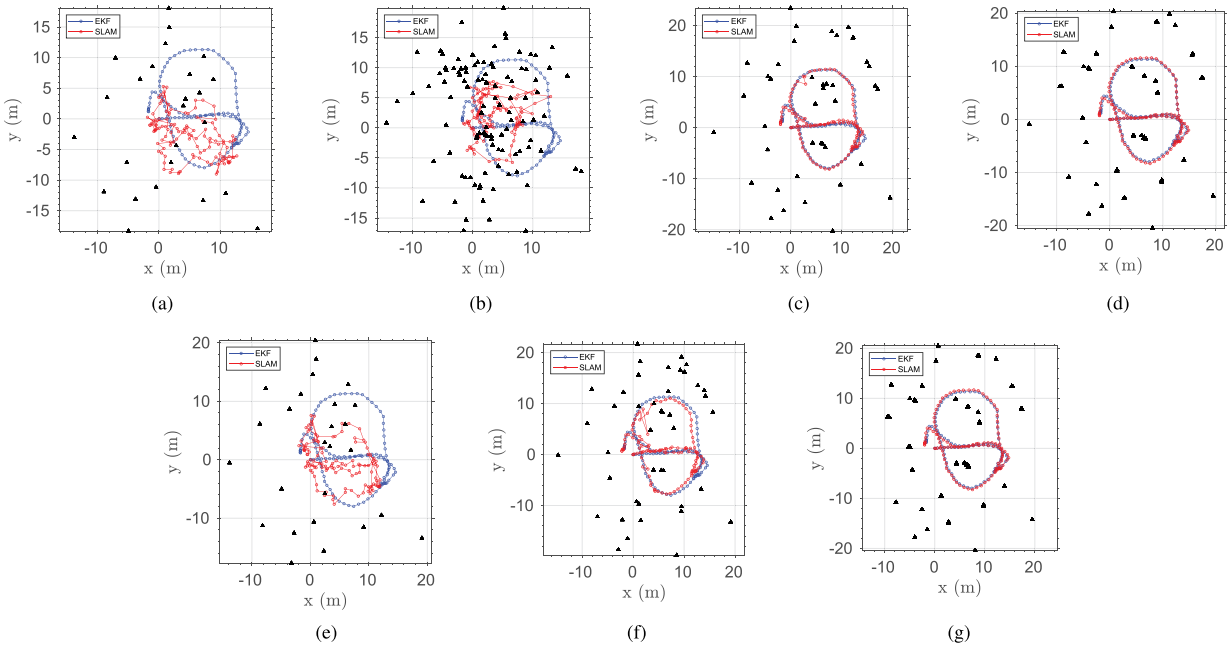


Fig. 19. Fused map after optimization with g2o. The solutions of MatchALS and MatchEIG are made cycle consistent by completing each connected components in the induced association graph. Each association is postprocessed to remove any clusters of size smaller than four. Each black triangle represents a single tree in the fused map. Trajectory estimates from EKF-based LIDAR-inertial odometry and after landmark-based SLAM are shown in blue and red, respectively. (a) Baseline (postprocessed). (b) NMFSync (postprocessed). (c) MatchALS (postprocessed). (d) MatchLift (postprocessed). (e) MatchEIG (postprocessed). (f) QuickMatch (postprocessed). (g) CLEAR (postprocessed).

of CLEAR thus makes the algorithm favorable for real-time SLAM applications. On the other hand, we note that the output of CLEAR still contains a few definite negatives; see Fig. 17(g). This is undesirable for landmark-based SLAM, as any incorrect fusion of landmarks could inflict catastrophic impact on the final SLAM solution. In practice, these mismatches can be filtered out by removing clusters of small size from the returned solution. For example, Fig. 17(h) shows the resulting association after removing clusters of size smaller than four from the output of CLEAR. After this postprocessing step, the final association is accurate and can be used by any SLAM back-end to solve for the vehicle trajectory and landmark positions.

Fig. 19 demonstrates the results of landmark-based SLAM using the data association returned by each algorithm. Prior to optimization, we apply the same postprocessing procedure described earlier, by removing clusters of size smaller than four from each data association. Subsequently, we initialize a single tree for each remaining cluster in the fused map. All tree positions and vehicle poses are then jointly optimized using g2o [34]. We note that Fig. 19 mainly provides a qualitative comparison of the trajectory estimates. Intuitively, we expect that SLAM trajectories that are discontinuous are likely to be wrong due to false data associations. These include the trajectories optimized with the baseline, NMFSync, MatchALS, MatchEIG, and QuickMatch. While CLEAR and MatchLift produce similar results, CLEAR is more than 1000 times faster as indicated by the results in Table II.

Finally, we note that some quantitative results presented in this section are different from those obtained in earlier comparisons based on synthetic data. For example, we observed that algorithms such as NMFSync perform better in simulation. A major cause of this discrepancy is the noise model. In our synthetic data, the input is solely corrupted by mismatches that reassign correct matches to wrong ones. In the forest dataset, however, the input is corrupted by both mismatches and a significant number of *missing* correct associations, thus further reducing the signal-to-noise ratio.

X. Conclusion

Data association across multiple views was a fundamental problem in robotic applications. Traditionally, this problem was decomposed into a sequence of pairwise subproblems. Multiview matching algorithms could leverage observation redundancy to improve the accuracy of pairwise associations. However, the use of these algorithms in robotic applications was often prohibited by their high computational complexity, as well as critical issues such as cycle inconsistency and high number of mismatches which may have catastrophic consequences.

To address these critical challenges, we presented CLEAR, an algorithm that leverages the natural graphical representation of the multiview association problem. CLEAR used a spectral graph clustering technique, which was uniquely tailored to solve this problem in a computationally efficient manner. Empirical results based on extensive synthetic and experimental evaluations demonstrated that CLEAR outperforms the state-of-the-art algorithms in terms of both accuracy and speed. This general framework could provide significant improvements in the accuracy and efficiency of data association in many applications such as metric/semantic SLAM, multiobject tracking, and multiview point cloud registration that traditionally rely on pairwise matchings.

APPENDIX

Proof of Proposition 2: Consider the optimization problem (4). Since trace is invariant under cyclic permutations,[11] we obtain

$$\max_{P=V\,V^\top} \langle P_{\text{nrm}},\, \tilde{P}_{\text{nrm}}\rangle = \max_{P=V\,V^\top} \operatorname{tr}(P_{\text{nrm}}^\top \tilde{P}_{\text{nrm}}) \qquad \text{(from definition of inner product } \langle\cdot,\cdot\rangle) \tag{18a}$$

$$= \max_{V\in\mathbb{V}} \operatorname{tr}(C^{-\frac{1}{2}}\, V\, V^\top C^{-\frac{1}{2}}\, \tilde{P}_{\text{nrm}}) \qquad \text{(since } P_{\text{nrm}} \stackrel{\text{def}}{=} C^{-\frac{1}{2}}\, P\, C^{-\frac{1}{2}} \text{ and } P = V\,V^\top) \tag{18b}$$

$$= \max_{V\in\mathbb{V}} \operatorname{tr}(V^\top C^{-\frac{1}{2}}\, \tilde{P}_{\text{nrm}}\, C^{-\frac{1}{2}}\, V). \qquad \text{(from cyclic permutation)} \tag{18c}$$

As discussed in Section III-B, in the graph formulation of the problem, V corresponds to partitions of the association graph $\mathcal{G}$ into clusters $\mathcal{A}_1,\ldots,\mathcal{A}_m$, where $(V)_{ij}=1$ if and only if vertex $v_i\in\mathcal{A}_j$. This implies that $\sum_{i=1}^{l}(V)_{ij}=|\mathcal{A}_j|$, and diagonal entries of C are $c_i=|\mathcal{A}_j|$ for each vertex $v_i\in\mathcal{A}_j$. Consequently, $V^\top C^{-1}V=I$. Since solution of (18c) is invariant to adding/subtracting a constant to the objective function, by subtracting $\operatorname{tr}(V^\top C^{-1}V)=\operatorname{tr}(I)=l$ from (18c) and defining $U\stackrel{\text{def}}{=}C^{-\frac{1}{2}}\,V$, we obtain the equivalent program:

$$\max_{V\in\mathbb{V}} \operatorname{tr}(V^\top C^{-\frac{1}{2}}\, \tilde{P}_{\text{nrm}}\, C^{-\frac{1}{2}}\, V) - \operatorname{tr}(V^\top C^{-1} V) \tag{19a}$$

$$= \max_{V\in\mathbb{V}} \operatorname{tr}(V^\top C^{-\frac{1}{2}}\, \tilde{P}_{\text{nrm}}\, C^{-\frac{1}{2}}\, V) - \operatorname{tr}(V^\top C^{-\frac{1}{2}}\, C^{-\frac{1}{2}}\, V) \qquad (C^{-1}=C^{-\frac{1}{2}}\,C^{-\frac{1}{2}}) \tag{19b}$$

$$= \max_{U\in\mathbb{U}} \operatorname{tr}(U^\top \tilde{P}_{\text{nrm}}\, U) - \operatorname{tr}(U^\top U) \qquad \text{(replacing } U\stackrel{\text{def}}{=}C^{-\frac{1}{2}}\,V) \tag{19c}$$

$$= \max_{U\in\mathbb{U}} \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}}\, \tilde{P}\, \tilde{C}^{-\frac{1}{2}}\, U) - \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}}\, \tilde{C}\, \tilde{C}^{-\frac{1}{2}} U) \qquad (\text{since } \tilde{C}^{-\frac{1}{2}}\,\tilde{C}\,\tilde{C}^{-\frac{1}{2}}=I) \tag{19d}$$

$$= \max_{U\in\mathbb{U}} \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}}\, (\tilde{P}-\tilde{C})\, \tilde{C}^{-\frac{1}{2}}\, U) \qquad \text{(by factoring terms)} \tag{19e}$$

$$= \min_{U\in\mathbb{U}} \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}}\, \tilde{L}\, \tilde{C}^{-\frac{1}{2}}\, U) \qquad (\text{since } \tilde{P}-\tilde{C}=-\tilde{L}) \tag{19f}$$

$$= \min_{U\in\mathbb{U}} \operatorname{tr}(U^\top \tilde{L}_{\text{nrm}}\, U). \qquad \text{(using definition } \tilde{L}_{\text{nrm}} \stackrel{\text{def}}{=} \tilde{C}^{-\frac{1}{2}}\,\tilde{L}\,\tilde{C}^{-\frac{1}{2}}) \tag{19g}$$

From the definition $U\stackrel{\text{def}}{=}C^{-\frac{1}{2}}\,V$ and since $V^\top C^{-1}V=I$, it follows that $U^\top U=I$. ■

Proof of Lemma: 1 The spectrum of a *complete* graph with l_i vertices and Laplacian $L_i\in\mathbb{R}^{l_i\times l_i}$ consists of eigenvalues 0 and l_i, with multiplicities 1 and l_i-1, respectively [28, Chap. 1]. Since in this case the diagonal matrix $C_i=D_i+I$ has diagonal entries l_i, eigenvalues of the normalized Laplacian $C_i^{-\frac{1}{2}}\,L_i\,C_i^{-\frac{1}{2}}=\frac{1}{l_i}L_i$ are 0 and 1, with multiplicities 1 and l_i-1, respectively. By definition, a cluster graph is a disjoint union of complete graphs. Since spectrum of a graph is the union of its connected components' spectra [28], the conclusion follows. ■

Proof of Lemma: 2 Let $\lambda_1\le\lambda_2\le\cdots\le\lambda_l$ denote ordered eigenvalues of L_{nrm}, where from Lemma 1, we have $\lambda_1=\lambda_2=\cdots=\lambda_m=0$ and $\lambda_{m+1}=\lambda_{m+2}=\cdots=\lambda_l=1$. If $\tilde{\lambda}_1\le\tilde{\lambda}_2\le\cdots\le\tilde{\lambda}_l$ are the ordered eigenvalues of $\tilde{L}_{\text{nrm}}=L_{\text{nrm}}+N$, from the Weyl's eigenvalue theorem [35], we have $|\tilde{\lambda}_i-\lambda_i|<\|N\|$ for all $i\in\mathbb{N}_l$. This implies, if $\|N\|<0.5$, that $|\{\tilde{\lambda}\,:\,\lambda<0.5\}|=m$, which shows the correct number of clusters is recovered. ■

Proof of Proposition: 3 We have

$$\|\tilde{L}_{\text{nrm}}-L_{\text{nrm}}\| = \|\tilde{C}^{-\frac{1}{2}}\,\tilde{L}\,\tilde{C}^{-\frac{1}{2}} - C^{-\frac{1}{2}}\,L\,C^{-\frac{1}{2}}\| \qquad \text{(from definitions of } \tilde{L}_{\text{nrm}}, L_{\text{nrm}}) \tag{20a}$$

$$= \|C^{-\frac{1}{2}}\,(\tilde{L}-L)\,C^{-\frac{1}{2}}\| \qquad \text{(since by assumption } \tilde{C}=C) \tag{20b}$$

$$\le \|C^{-1}\|\,\|\tilde{L}-L\| \qquad \text{(since 2-norm is submultiplicative)} \tag{20c}$$

$$= \|C^{-1}\|\,\|(\tilde{D}-\tilde{A})-(D-A)\| \qquad (\text{since } L\stackrel{\text{def}}{=}D-A) \tag{20d}$$

$$= \|C^{-1}\|\,\|\tilde{A}-A\| \qquad (\text{since } \tilde{C}=C \text{ and } D=C-I) \tag{20e}$$

$$= \|C^{-1}\|\,\|E\| \qquad (\text{since } E\stackrel{\text{def}}{=}\tilde{A}-A) \tag{20f}$$

$$\le \frac{1}{c_{\min}}\|E\| \qquad \text{(since } C \text{ is diagonal)} \tag{20g}$$

$$\le e_{\max}/c_{\min} \qquad (\text{since } \|E\|\le e_{\max}) \tag{20h}$$

where the last inequality follows from the Gershgorin circle theorem [35, Sect. 6.1]. The conclusion follows from Lemma 2 and observing that $\|\tilde{L}_{\text{nrm}}-L_{\text{nrm}}\|=\|N\|<0.5$ implies $e_{\max}<0.5\,c_{\min}$. ■

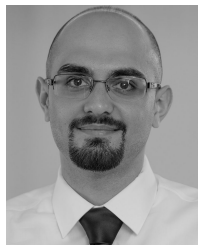
[11] e.g., $\operatorname{tr}(A\,B\,C)=\operatorname{tr}(B\,C\,A)=\operatorname{tr}(C\,A\,B)$.

ACKNOWLEDGMENT

The authors would like to thank Florian Bernard, Kostas Daniilidis, Spyros Leonardos, Stephen Phillips, Roberto Tron, and Xiaowei Zhou for their valuable insights that considerably improved this work.

REFERENCES

[1] D. Pachauri, R. Kondor, and V. Singh, "Solving the multi-way matching problem by permutation synchronization," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 1860–1868.
[2] X. Zhou, M. Zhu, and K. Daniilidis, "Multi-image matching via fast alternating minimization," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 4032–4040.
[3] R. Tron, X. Zhou, C. Esteves, and K. Daniilidis, "Fast multi-image matching via density-based clustering," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 4077–4086.
[4] R. Aragues, E. Montijano, and C. Sagues, "Consistent data association in multi-robot systems with limited communications," in *Proc. Robot. Sci. Syst.*, 2011, pp. 97–104.
[5] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. SIAM, 2012, vol. 106, doi: 10.1137/1.9781611972238.
[6] S. Leonardos, X. Zhou, and K. Daniilidis, "Distributed consistent data association via permutation synchronization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 2645–2652.
[7] C. Zach, M. Klopschitz, and M. Pollefeys, "Disambiguating visual relations using loop constraints," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1426–1433.
[8] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas, "An optimization approach to improving collections of shape maps," in *Computer Graphics Forum*, Wiley Online Library, 2011, vol. 30, no. 5, pp. 1481–1491.
[9] S. Phillips and K. Daniilidis, "All graphs lead to rome: Learning geometric and cycle-consistent representations with graph convolutional networks," 2019, *arXiv:1901.02078*.
[10] Y. Chen, L. Guibas, and Q. Huang, "Near-optimal joint object matching via convex relaxation," in *Proc. Int. Conf. Mach. Learn.*, Jun. 22–24, 2014, pp. 100–108.
[11] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[12] N. Hu, Q. Huang, B. Thibert, and L. J. Guibas, "Distributable consistent multi-object matching," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 2463–2471.
[13] J.-G. Yu, G.-S. Xia, A. Samal, and J. Tian, "Globally consistent correspondence of multiple feature sets using proximal Gauss–Seidel relaxation," *Pattern Recognit.*, vol. 51, pp. 255–267, 2016.
[14] S. Leonardos and K. Daniilidis, "A distributed optimization approach to consistent multiway matching," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 89–96.
[15] E. Maset, F. Arrigoni, and A. Fusiello, "Practical and efficient multi-view matching," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 4578–4586.
[16] F. Bernard, J. Thunberg, J. Goncalves, and C. Theobalt, "Synchronisation of partial multi-matchings via non-negative factorisations," *Pattern Recognition*, Elsevier, vol. 92, 2019, pp. 146–155.
[17] J. Yan, Z. Ren, H. Zha, and S. Chu, "A constrained clustering based approach for matching a collection of feature sets," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2016, pp. 3832–3837.
[18] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. Eur. Conf. Comput. Vision*. Springer, 2008, pp. 705–718.
[19] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.
[20] P. Swoboda, D. Kainmuller, A. Mokarian, C. Theobalt, and F. Bernard, "A convex relaxation for multi-graph matching," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 11148–11157.
[21] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[22] S. S. Skiena, *The Algorithm Design Manual*. Springer Science & Business Media, vol. 1, 1998. [Online]. Available: https://www.springer.com/gp/book/9781848000698
[23] H. Lutkepohl, *Handbook of Matrices*. Wiley Chichester, vol. 1, 1996. [Online]. Available: https://www.wiley.com/en-us/Handbook+of+Matrices-p-9780471970156
[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[25] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Advances Neural Inf. Process. Syst.*, 2002, pp. 849–856.
[26] J. Gallier, "Notes on elementary spectral graph theory. Applications to graph clustering using normalized cuts," 2013, *arXiv:1311.2492*.
[27] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation," *SIAM J. Numer. Anal.*, vol. 7, no. 1, pp. 1–46, 1970.
[28] F. R. Chung and F. C. Graham, *Spectral Graph Theory*. American Mathematical Society, 1997. [Online]. Available: https://books.google.com/books/about/Spectral_Graph_Theory.html?id=YUc38_MCuhAC
[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[30] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 1469–1472.
[31] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. New York, NY, USA: Springer Science & Business Media, 2012, vol. 26. [Online]. Available: https://www.springer.com/gp/book/9780387008936
[32] Y. Tian *et al.*, "Search and rescue under the forest canopy using multiple UAS," in *Proc. Int. Symp. Expe. Robot.*, 2018, pp. 140–152.
[33] T. Bailey, E. M. Nebot, J. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," in *Proc. Int. Conf. Robot. Automat.*, vol. 3, 2000, pp. 2512–2517.
[34] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3607–3613.
[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge University Press, 2012.



Kaveh Fathian received the M.S. and Ph.D. degrees in electrical engineering and the M.S. degree in mathematics from the University of Texas at Dallas, TX, USA, in 2013 and 2018, respectively.

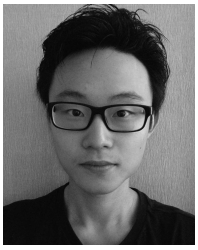
He is a Postdoctoral Associate with the Massachusetts Institute of Technology's Aerospace Controls Laboratory, Cambridge, MA, USA. His current research interests include topics in control theory, distributed and multi-robot systems, vision-based estimation, and spectral graph theory.

Dr. Fathian is a Member of the IEEE Control Systems Society and the Robotics and Automation Society.

Kasra Khosoussi received the bachelor's degree in computer engineering from the Department of Electrical and Computer Engineering at the K. N. Toosi University of Technology, Tehran, Iran, in 2011, and the Ph.D. degree in robotics from the Centre for Autonomous Systems (CAS), University of Technology Sydney, Australia, in 2017. He was a visiting Ph.D. student at the Department of Computer Science, University of Southern California, Los Angeles, CA, USA (2015–2016).

He is currently a Research Scientist with the Department of Aeronautics and Astronautics, and the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA, USA, where he previously was a Postdoctoral Associate (2017–2018). The overarching goal of his research is to develop robust and reliable algorithms with provable performance guarantees for single- and multi-robot spatial perception. His works include seminal results on certifiably correct algorithms for distributed pose-graph optimization, resource-aware algorithms with provable performance guarantees for single- and multi-robot spatial perception, and characterization of the impact of graph topology on estimation reliability in estimation-over-graphs problems such as SLAM. His current research interests include collaborative spatial perception, optimization (distributed, combinatorial, convex, and Riemannian), estimation theory, spectral graph theory, approximation algorithms, and numerical linear algebra, with applications to robotics.

He was a Best Paper Award Finalist in Multirobot Systems at the 2018 IEEE International Conference on Robotics and Automation (ICRA).



Yulun Tian received the B.A. degree in computer science from UC Berkeley, Berkeley, CA, USA, in 2017, and the S.M. degree in aeronautics and astronautics in 2019 from the Massachusetts Institute of Technology, Cambridge, MA, USA, where he is currently working toward the Ph.D. degree in aeronautics and astronautics.

His current research interests include geometric estimation and optimization for distributed multiagent systems.

Parker Lusk received the B.S. and M.S. degrees in electrical engineering from Brigham Young University, Provo, UT, USA, in 2016 and 2018, respectively. He is currently working toward the Ph.D. degree with the Massachusetts Institute of Technology, Cambridge, MA, USA.

His current research interests include visual-inertial navigation, geometric control and estimation, and data association.

Jonathan P. How received the B.A.Sc. degree in aerospace from the University of Toronto, Canada, in 1987, and the S.M. and Ph.D. degrees in aeronautics and astronautics from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1990 and 1993, respectively.

He studied for 1.5 years at MIT as a postdoctoral associate for the Middeck Active Control Experiment (MACE) that flew onboard the Space Shuttle Endeavour in March 1995. Prior to joining MIT in 2000, he was an Assistant Professor in the Department of Aeronautics and Astronautics at Stanford University, Stanford, CA, USA. He is currently the Richard C. Maclaurin Professor of Aeronautics and Astronautics with MIT.

Prof. How is a Fellow of the American Institute of Aeronautics and Astronautics (AIAA), former Editor-in-Chief of the IEEE CONTROL SYSTEMS magazine, and an Associate Editor of the *AIAA Journal of Aerospace Information Systems* and the IEEE TRANSACTIONS OF NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). He has won multiple AIAA Best Paper Awards, including those of 2011, 2012, and 2013. Additionally, he earned the 2011 IFAC Automatica Award for the Best Application Paper, the 2020 AIAA Intelligent Systems Award, and 2002 Institute of Navigation Burka Award.