Data Association Algorithms and Representations for Robust Geometric Perception

by

Parker C. Lusk

B.S., Brigham Young University (2016)
M.S., Brigham Young University (2018)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2023

Authored by: Parker C. Lusk
Department of Aeronautics and Astronautics
July 14, 2023

Certified by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics, MIT
Thesis Supervisor

Certified by: John J. Leonard
S. C. Collins Professor of Mechanical and Ocean Engineering, MIT
Thesis Supervisor

Certified by: Nikolay Atanasov
Assistant Professor of Electrical and Computer Engineering, UCSD
Thesis Supervisor

Accepted by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Data Association Algorithms and Representations for Robust Geometric Perception

by

Parker C. Lusk

Submitted to the Department of Aeronautics and Astronautics
on July 14, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Data association is a fundamental requirement of geometric estimation in robotics. Identifying correspondences between measurements and models enables estimation processes to incorporate more data, in general leading to better estimates. However, sensor data is replete with noise and spurious measurements, making data association considerably more challenging. This thesis addresses data association problems that arise in realistic robotic perception, thus enabling robust geometric estimation. The first contribution of this thesis is the introduction of a scalable algorithm that efficiently identifies pairwise correspondences in high-outlier scenarios without initial data alignment. By modeling the pairwise association problem using a weighted graph, large complete subgraphs of highly consistent associations can be found without sacrificing information through thresholding, unlike previous methods. The second contribution is the introduction of a novel representation for lines and planes using the affine Grassmannian manifold. This thesis looks beyond points to higher-order geometric abstractions and provides a means for robustly aligning line and plane landmarks without an initial guess. When applied to lidar-based localization and loop closure, which face challenges in registering sparse point clouds, higher accuracy and success rates are achieved compared to typical representations. The third contribution of this thesis is to extend pairwise data association to multiway data association, wherein multiple pairs of associations are jointly analyzed to improve their accuracy and to ensure their consistency. By leveraging insights from the spectral graph clustering literature, this thesis develops an algorithm that is computationally efficient and provides accurate solutions with guaranteed global consistency. The final contribution of this thesis is to develop a multiway association algorithm that is capable of operating directly on pairwise affinities, unlike previous work which assumes the availability of pairwise binary permutation matrices. By delaying pairwise decision making until many pairwise affinities can be analyzed together, higher accuracy associations can be made. Taken together, these contributions improve the robustness of data association, allowing reliable geometric estimation in the presence of uncertainty.

Acknowledgments

The work of earning a PhD is accompanied by many highs and lows and would be insurmountable without the support of so many mentors, colleagues, and friends. Support systems are important, and I am very grateful for mine.

First, many thanks to my advisor, Professor Jonathan How, who sets the bar high and works tirelessly to help his students be successful. He not only helps his students *do* effective research, but gives them the tools to *become* effective researchers. I am glad to have worked with him on various research projects over the years and am happy to have played some small role in his `git`/Python expertise.

I am fortunate to have had excellent committee members. Professor John Leonard and Professor Nikolay Atanasov have both given me invaluable guidance and encouragement throughout the research process. I am especially grateful for the technical discussions and constructive feedback which has helped develop my critical thinking and understanding of the research landscape.

The Aerospace Controls Laboratory (ACL) has been an incredible group to be associated with and I am grateful for the impact that each person has had on me. In particular, thanks to Drs. Kaveh Fathian and Kasra Khosoussi for being excellent collaborators, for always being willing to work through problems together, and for being my thesis readers. Thanks to Drs. Brett Lopez and Michael Everett for all of the inspiration, for teaching me many things and for providing me with examples of how to be a successful researcher (and grad student). Dr. David Rosen made a huge impact on me by expanding my optimization knowlege and teaching me how to formulate problems and how to approach solutions. So many of my peers have taught me and encouraged me: Dr. Jesus Tordesillas (who has also flown—and crashed—FPV drones with me), Dr. Dong-Ki Kim, Dr. Kris Frey, Lena Downes, Xiaoyi (Jeremy) Cai. I'm especially grateful to Yulun Tian and Andrea Taglibue for their support and for all the interesting and helpful discussions. Thanks also to Savva Morozov, Jacqueline Ankenbauer, Kota Kondo, and Mason Peterson for collaborating with me on interesting problems.

Finally, thank you to my family: my mom from whom I learned my street smarts, my dad from whom I learned how to problem solve, my siblings from whom I learned resilience, my daughter from whom I learned balance. Most importantly, thank you to my wife Candice. Thank you. Your immeasurable support and sacrifice during graduate school cannot be overstated.

This work is supported by the Ford Motor Company, The Boeing Company under Cooperative Agreement MRA#2017-656, ARL DCIST under Cooperative Agreement W911NF-17-2-0181, and by UPenn under ONR award 584551.

Contents

List of Figures

List of Tables

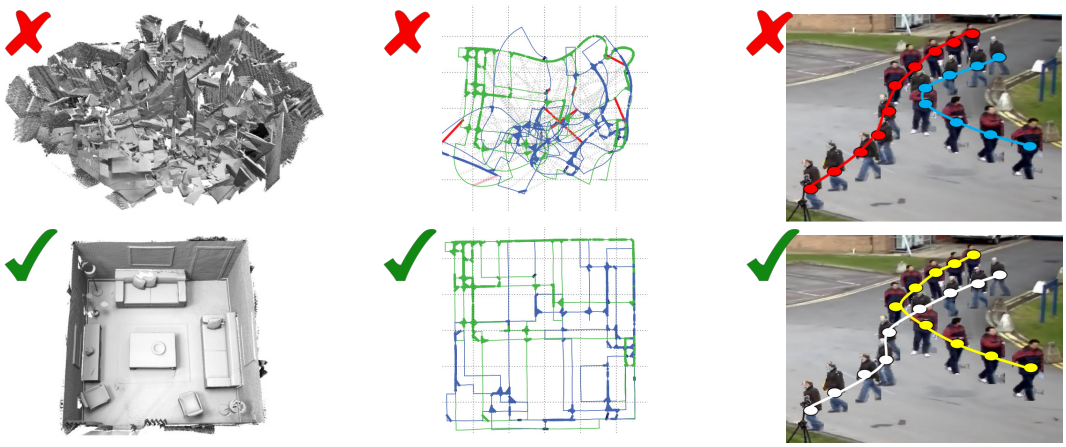
Chapter 1

Introduction

1.1 Overview

The widespread deployment of autonomous mobile robots remains a challenge, largely due to uncertainty [1–3]. Uncertainty arises from multiple factors, including 1) sensor limitations such as noise, narrow field of view, occlusions, and perceptual aliasing, 2) navigation and localization errors, often in the form of inaccurate egomotion estimation with respect to other robots and objects, and 3) dynamic environments, where the misclassification of static and dynamic objects can lead to poor situational awareness and mapping errors. Typically, uncertainty is accounted for in the context of estimation theory by leveraging assumed-known measurement noise statistics to obtain an optimal state estimate using the maximum-likelihood paradigm [4–7]. However, these methods—typically relying on least squares estimation under the assumption of Gaussian noise—are not robust to out-of-distribution measurements or long-tailed events [8]. Indeed, sensor data gathered in the field is frequently contaminated with spurious measurements, which are difficult to identify.

A central challenge caused by these uncertainties is *data association* [9–11], which is the process of identifying correspondences between observation sets—either as raw measurements or abstracted representations (e.g., keypoints, geometries, objects). Successful data association is crucial for real-world estimation applications such as localization [12], simultaneous localization and mapping (SLAM) [13, 14], multiview



Redwood Data [22] (a)

City10000 Dataset [23] (b)

PETS 09 Dataset [24] (c)

Figure 1-1. Example estimation applications highlighting the importance of successful data association. When data association fails—sometimes even slightly, e.g., (c)—estimation quality suffers significantly compared to results with known correspondences (bottom). (a) Indoor RGB-D reconstruction from point cloud fragments [22]. Each fragment pair must be correctly aligned using point correspondences. (b) Multirobot loop closure detection between green and blue robot trajectories [23,25]. Even a handful of invalid loop closures can be disastrous for estimation. (c) Multi-object tracking from a stationary camera [24,26]. Two pedestrians abruptly change motion profile, crossing in front of each other and causing the tracker to associate measurements to the wrong existing track.

reconstruction [15], object pose estimation [16, 17], object tracking [18, 19], multi-robot frame alignment [20], and collaborative mapping [21]. However, data association performance can become severely degraded or costly in real-world settings due to the aforementioned uncertainties, resulting in estimation failures, such as those demonstrated in Fig. 1-1. Therefore, it is imperative that algorithms exist which are *robust*—i.e., capable of detecting and rejecting measurements that are inconsistent with or substantially deviate from the estimation model. This thesis provides data association algorithms that are robust, scalable, and suitable for online use, enabling the use of estimation algorithms for real-world and reliable robot systems.

Broadly, data association is the operation of associating measurements (either raw or processed) to some process or model. For example, in the object tracking literature [27], data association typically refers to associating uncertain measurements to known tracks. Similarly, in SLAM, data association often refers to relating sensor observations with landmarks in the map [28]. In multirobot settings, data association is necessary for aligning representations and coordinate frames [29]. Data association

can also be synonymous with information fusion [30, 31], where data from multiple sensors is fused to provide a more complete description of a signal of interest [32], often in a recursive Bayes filter framework [33]. In this thesis, data association is defined as finding correspondences between two or more sets of objects, where these sets are defined by the problem of interest. For example, in point cloud registration a rigid-body transform is sought that best aligns two point clouds. To estimate the alignment, correspondences between points of the two point clouds must be identified, necessitating *pairwise data association*. When many sets of objects need to be associated, such as in multi-view image matching [34] where the features of each image form a set, *multiway data association* is used. Throughout this thesis we enforce that correspondences should be *one-to-one*, i.e., an element of one set can only be matched with one other element, and *distinct*, i.e., an element cannot be matched with another element from the same set.

In practice, observations are "noisy" and "partial", i.e., where an unknown number of objects may not have a counterpart in one or more of the other sets. Some of these objects may correspond to spurious measurements (e.g., noise from RGB-D sensor) and any *correspondence* to them would constitute an *outlier*—similarly so with any incorrect correspondence between valid objects. To deal with these robustness issues, three main lines of work have emerged. First, robust model fitting in the presence of spurious measurements can be posed as the maximum consensus problem [35]. Due to its NP-hardness, approximate algorithms are instead employed, with the most used algorithm being RANSAC [36], a non-deterministic heuristic that has a running time exponentially proportional to the desired probability of success [37]. Second, motivated by robust statistics and originating in the work of Huber [38] is the use of robust loss functions, which are less sensitive to the deleterious effects of outliers, but introduce a local iterative re-weighting procedure. While provably insensitive to a bounded fraction of outliers [39], many of these so-called M-estimators introduce non-convexity, thus increasing reliance on a good initial guess, and often preclude the underlying optimization from admitting a tight convex relaxation [40]. A related method is graduated non-convexity (GNC) [41], a heuristic which can be used

to optimize non-convex cost functions [42]. Third, the robust selection of pairwise correspondences can be formulated as the NP-hard quadratic assignment problem (QAP) [43]. This class includes graph matching [44] and maximum clique [12, 45] formulations, which are in general not scalable and so heuristic approximations are frequently sought. Additional robustness can be gained by leveraging multiple pairs of correspondences in a *multiview matching* [46] framework, often at the expense of increased computational complexity. This class of approaches is explored in this thesis by leveraging graph-based representations for assignment problems. In contrast to other works, continuous relaxations to the underlying NP-hard combinatorial optimizations are found that (1) have an empirically large basin of attraction and (2) allow for weighted graphs, increasing the amount of information available and leading to improved data association.

This thesis asserts that *measurement consistency* provides a rich signal for robust data association, leading to successful geometric estimation. The classification of correspondences are correct or incorrect is unobservable a priori, therefore correspondences that are consistent with each other and with the estimation model should be sought. Because data association tends to be the "hardest part" of many estimation problems [47–49], this thesis presents algorithms that recover correspondences and that can be coupled with geometric estimators to provide solutions even in the presence of many noisy and spurious measurements. The availability of such algorithms enables the reliable deployment of autonomous systems, due to their reliance on data association for navigation and situational awareness.

The conceptual and computational challenges that must be overcome to achieve this goal are outlined in Section 1.2. Section 1.3 gives an overview of the technical contributions. Finally, Section 1.4 presents the thesis structure.

1.2 Problem Statement

This thesis develops data association algorithms to address key challenges in deploying mobile robots in uncertain environments. The following questions are investigated: (i)

How to robustly identify pairwise correspondences in the presence of a large number of incorrect options? (ii) How to best represent data from modern sensors (e.g., 3D lidar, RGB-D) for reliable and efficient data association? (iii) How to ensure consistent data association across multiple observations? (iv) How to utilize uncertain affinities in consistent, multiway data association? The following sections provide motivation and elaborate on these questions, while the technical problem formulations are left to subsequent chapters.

1.2.1 Robust pairwise data association

Successful geometric estimation requires fitting data to some model. This is fundamentally challenging in the presence of many noisy measurements because it is not immediately clear which measurements should be used or which measurements correspond to which aspects of the model. For example, to enjoy the loop closing benefits of SLAM, currently observed landmarks must be identified with previously seen landmarks. Similarly, when aligning two point clouds, point correspondences between the data and the model are necessary. Traditionally, this data association task is accomplished using the Hungarian [50] algorithm for bipartite matching, which requires each data point to have a descriptor that is used to score how similar two data points are. However, descriptors may not be discriminative enough (e.g., learned descriptors [51, 52] used in different settings than they were trained in), leading to incorrect matches. Further, the Hungarian algorithm typically assumes that all data should be matched, which is not necessarily true and is rarely known a priori. The objective of this work is to enable robust pairwise data association without relying on descriptors and even in the presence of noisy, spurious, and partially observed data.

1.2.2 Representations for data association

Modern sensors like 3D lidars or RGB-D cameras are ubiquitous in robotics because they provide accurate depth sensing and typically cover a wide sensing area. However, they produce an enormous amount of data, with some spinning lidars producing

approximately 5 million points per second[1]. While high-resolution 3D data is immensely useful for perception and navigation, it is challenging to process, store, and transfer due to memory, storage, and bandwidth limitations. Further, sensor processing is complicated by various sensing patterns (e.g., spinning vs solid state vs overlapping cells[2]), non-uniform sampling, and blind spots. To address these issues, geometric primitives like lines and planes can be used to abstract raw sensor data into more manageable measurements and maps [53–55]. These geometric primitives significantly reduce the size of the map required for navigation and improve the structural understanding of the scene (e.g., instead of seeing thousands of co-planar points, you see one wall). However, existing works tend to use heuristics for matching lines and planes that are local and sensitive to viewing angle, thus limiting a SLAM system's ability to produce loop closures or a calibration system's ability to register sensors without providing a good initial guess. The objective of this work is to remove the requirement of a good initial guess for matching lines and planes and to study the effect that the mathematical representation of lines and planes has on data association.

1.2.3 Consistent multiway data association

Establishing correspondence between data points across multiple sets is a challenging problem in robotic perception and computer vision and is commonly known as multi-view or multiway matching [46]. Conceptually, the goal in this problem is to establish correct associations between the sightings of objects across multiple "views". Example applications include feature matching across multiple frames [46,56,57], and associating landmarks across multiple maps for map fusion in single/multirobot SLAM [20]. In practice, data points are contaminated with noise and outliers, rendering classical assignment techniques (such as the Hungarian [50] or auction [58] algorithms) ineffective as they cannot reject outliers and may produce inconsistent results. Traditional approaches to the multiway matching problem leverage redundant observations

[1] Velodyne VLS-128 can output 4.8M points per second and Ouster OS1 can output 5.2M points per second

[2] Cepton introduced Micro Motion Technology®, which consists of an overlapping grid of forward-facing lasers that randomly sample points within their cell.

and attempt to synchronize all noisy pairwise associations via enforcing a *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 the equivalence of observation subsets, which is necessary for global fusion in applications such as map merging. Synchronizing pairwise associations requires combinatorial optimization over an exponentially large search space. While many state-of-the-art synchronization methods exist [46, 57, 59–66], solution strategies that are computationally tractable for real-time applications remain an active area of research. Further, the rounding techniques used by some relaxation-based methods may actually violate cycle consistency and other important constraints (i.e., distinctness). The objective of this work is to address these challenges by identifying novel optimization formulations and relaxations, thus enabling real-time performance in robotics applications.

1.2.4 Data fusion from multiple views and sensing modalities

Multiway data association is often posed as the synchronization of pairwise correspondences, known as permutation synchronization [46, 59–66]. The pairwise correspondences are recovered from an initial data association process, which typically generate binary pairwise matchings based on uncertain pairwise similarities (affinities). The output of the pairwise data association is then improved and made consistent by joint optimization. Methods that take this approach are effective at multiway matching provided that binary matchings are available; however, their use of (partial) permutation matrices is akin to late fusion [67, 68] and precludes them from using all available information, i.e., multiway matching is performed *after* each pairwise affinity matrix is pre-processed to create binary pairwise matches. The objective of this work is to develop an algorithm that works directly on the pairwise affinities rather than the pre-processed binary pairwise matches, enabling early fusion of data and an appropriate way to mix the scores from multiple sensing modalities.

1.3 Technical Contributions

This section gives an overview of the technical contributions of this thesis. Each subsection highlights a contribution that addresses a problem described in Section 1.2. These contributions utilize two key insights: data association can be made robust by (1) leveraging *measurement consistency* and (2) by developing problem formulations that make use of *weighted* information. Compared to the prior works that require an initial guess or that ignore weighted information, these contributions enable robust pairwise and multiway data association algorithms.

1.3.1 Contribution 1: Robust Pairwise, Global Data Association using Graph-Theoretic Concepts

This contribution [69] addresses the problem of pairwise data association even in the presence of extreme outlier regimes. Outlier associations arise due to noisy/bad measurements, occlusions, and the weak discrimination ability of feature descriptors. The key idea of this contribution is to identify the largest group of pairwise compatible measurements by leveraging *geometric consistency*. A pair of associations are said to be geometrically consistent if the distance between objects in the same set is invariant under rigid-body transformation. A graph-theoretic formulation is developed, and since it results in a mixed-integer optimization, a continuous relaxation is proposed which can be solved via a first-order gradient-based algorithm. The benefit of the proposed algorithm is the ability to efficiently solve geometric data association problems, enabling the use of non-minimal least squares solvers for robust estimation.

1.3.2 Contribution 2: Abstract Geometric Representations for Global Data Association

This contribution [70, 71] addresses the shortcomings of existing work on matching geometric primitives. Lines and planes can be used to abstract high-volume, raw point cloud data, enabling much more lightweight maps in terms of memory and storage.

However, existing works frequently use Euclidean distance between line or plane pairs, using representations such as the centroid [72], which is not well-defined for infinite lines and planes, or the "closest point" (CP) parameterization [54], which is heavily dependent on the sensor pose. These deficiencies ultimately lead to data association algorithms that require a good initial guess because of viewpoint sensitivity. The key idea of this contribution is to leverage the proper geometric interpretation of lines and planes as elements of the affine Grassmannian manifold. By utilizing the natural metric associated with this manifold, we show that the distance between lines and planes can be calculated in a way that is *invariant* to arbitrary rotation and translation. The benefit of introducing this manifold view is the ability to use our graph-theoretic framework for data association. Thus, lines and planes can be associated across wide-baselines (e.g., in the case of place recognition or loop closure detection) without any initial guess.

1.3.3 Contribution 3: Consistent Multiway Synchronization of Pairwise Data Associations

This contribution [73] extends robust and consistent data association from pairwise to multiway. In multirobot or multiview settings, it is crucial that correspondences are *cycle consistent*, i.e., all robots or sensors agree on the a shared identity for each object of interest. If cycle consistency is not properly satisfied, data fusion can result in *catastrophic merging*, where unrelated objects, features, or landmarks are collapsed into a single entity. The key idea of this contribution is to leverage the natural graphical representation of the permutation synchronization problem. By providing new insights into the connections between permutation synchronization and spectral graph clustering, a multiway matching algorithm is developed that pushes the boundaries of accuracy and speed.

1.3.4 Contribution 4: Multiattribute, Multiway Fusion of Uncertain Pairwise Affinities

This contribution [74] enhances the ability of multiway data association algorithms by enabling direct processing of pairwise affinities without the need of a pairwise correspondence selection step. Because sensors are noisy, perceive objects from different perspectives, or utilize different modalities, different pairs of sensors may agree on different correspondences leading to cycle consistency violations which must be rectified. This work extends pairwise maximum-weight matching to the multiway case, allowing the fusion of uncertain pairwise affinities, rather than focusing on binary permutation synchronization as has been done in other work. The key idea of this work is to formulate a mixed-integer quadratic program with a particular continuous relaxation that leads to guaranteed constraint satisfaction and an efficient gradient-based algorithm. By processing affinities directly, this contribution enables the combination of multiple attributes (e.g., features from different modalities) and avoids unnecessary pre-processing of affinities into hard pairwise yes-no correspondences.

1.4 Thesis Structure

The rest of the thesis is structured as follows

- Chapter 2 provides an overview of the literature relevant to this thesis.

- Chapter 3 presents the framework for robust pairwise data association. Validation of the approach is performed on public datasets in the context of point cloud registration. The content of this chapter is based on:

  > Parker C Lusk, Kaveh Fathian, and Jonathan P How. CLIPPER: A graph-theoretic framework for robust data association. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13828–13834, 2021. `https://arxiv.org/abs/2011.10202`

- Chapter 4 presents the novel abstract geometric representation for pairwise data association of lines and planes. The content of this chapter is based on:

  > Parker C Lusk and Jonathan P How. Global data association for slam with 3d grassmannian manifold objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4463–4470, 2022. `https://arxiv.org/pdf/2205.08556.pdf`
  >
  > Parker C Lusk, Devarth Parikh, and Jonathan P How. GraffMatch: Global matching of 3d lines and planes for wide baseline lidar registration. *IEEE Robotics and Automation Letters*, 8(2):632–639, 2022. `https://arxiv.org/abs/2212.12745`

- Chapter 5 presents the multiway fusion framework for data with multiple attributes and uncertain pairwise affinities. The content of this chapter is based on:

  > Kaveh Fathian, Kasra Khosoussi, Yulun Tian, Parker C Lusk, and Jonathan P How. CLEAR: A consistent lifting, embedding, and alignment rectification algorithm for multiview data association. *IEEE Transactions on Robotics*, 36(6):1686–1703, 2020

- Chapter 6 presents the multiway fusion framework for data with multiple attributes and uncertain pairwise affinities. The content of this chapter is based on:

  > Parker C Lusk, Kaveh Fathian, and Jonathan P How. Mixer: Multiattribute, multiway fusion of uncertain pairwise affinities. *IEEE Robotics and Automation Letters*, 8(5):2462–2469, 2023. `https://arxiv.org/abs/2210.08360`

- Chapter 7 summarizes the thesis contributions and describes future research directions.

Chapter 2

Related Work

This thesis is concerned with robustly establishing correspondences, thus enabling reliable estimation of geometric quantities in robotic perception. In this chapter, an overview of literature related to this goal is provided. Section 2.1 discusses geometric estimation and specifically focuses on approaches that aim to improve robustness in the presence of spurious measurements. Section 2.2 provides an overview of approaches typically used for data association. Finally, Section 2.3 explores relevant literature on object and map representations and their effects on data association.

2.1 Geometric Perception

Geometric perception is a fundamental task in robotics and computer vision applications. It is the problem of estimating unknown geometric variables (e.g., poses, rotations, 3D structure) from sensor data (e.g., camera images, lidar scans, inertial data, wheel odometry) [49]. In practice, many of the measurements used for estimation are completely erroneous due to various sensing imperfections and are thus called outliers. Estimation in the presence of outliers has been widely studied in the statistical, robotics, and computer vision communities. In this section, three leading strategies to handling outliers are discussed: maximum consensus, M-estimation, and graduated non-convexity. Additionally, the recent research theme of *certifiable perception* is discussed in relation to robustness.

Two major themes prevail when dealing with outliers: their effects are either alleviated or they are rejected altogether. Often, these outlier management schemes are performed in tandem with the estimation task, which can result in larger, more expensive problems. Instead, this thesis focuses on adding robustness to the data association step, *rejecting* outliers *before* data is processed by the estimator.

2.1.1 Robust Estimation

Maximum consensus [35] estimation is an NP-hard problem [75] that aims to estimate model parameters which are consistent with as many inliers as possible, where the inlier–outlier decision is governed by a pre-determined threshold. RANSAC [36] and its variants are frequently used to approximately solve this problem; however, the core hypothesize-and-verify heuristic has fundamental shortcomings. Namely, such randomized methods tend to be computationally expensive in the presence of high outlier proportions because the probability of selecting an outlier-free minimal subset decreases exponentially with the number of outliers, resulting in an exponential increase in the number of required iterations [76]. Approximate deterministic approaches have recently been presented for the general case in [77] and for point cloud registration in [78]. A related approach in the context of planar SLAM is that of Carlone et al. [79], where a large set of consistent measurements are approximately sought under a linear estimation model. Exact maximum consensus methods are commonly based on branch-and-bound [80] or tree search [81], which limits the scalability of such methods.

Another framework for robust model fitting is M-estimation, which supplants the typical, yet highly-sensitive least-squares loss function with a robust alternative. Commonly used loss functions include the convex Huber loss, which has a theoretical breakdown point of 50% and has often been known to exhibit breakdown even with less than 50% outliers in the data [82, 83]. Non-convex loss functions exhibit increased robustness due to their redescending influence functions [84], such as the Geman-McClure (GM) [85] loss used for camera localization in [86] or the truncated least squares (TLS) loss used for multiple view geometry in [87], for point cloud reg-

istration in [88], and for robust pose graph optimization in [89]. Other applications of M-estimation in pose graph optimization include well-known techniques such as switchable constraints [90], dynamic covariance scaling [91], and max-mixtures [92]. Wang and Singer [93] used an unsquared deviations loss function for robust rotation synchronization. For point cloud registration, Fitzgibbon [94] incorporated the Huber loss function into the model fitting step of the local iterative closest point (ICP) method [95, 96] and a comparison of robust loss functions used in ICP was presented by Babin et al. [97]. The principal strategy for solving these M-estimation problems is through local search via iteratively re-weighted least squares (IRLS) [84]. Therefore, the use of the more robust, but non-convex loss functions can unfortunately cause the M-estimation to become (even more) sensitive to initial conditions.

An alternate strategy to solve non-convex M-estimation problems is to leverage graduated non-convexity (GNC) [41], a heuristic in which a convex surrogate problem is first solved, after which the problem is gradually made more non-convex—using the previous solution as the initial guess—until arriving at the original non-convex problem. In [98], this strategy was employed to solve the point cloud registration problem with GM loss. Yang et al. [42] studied GNC in combination with TLS and GM for variety of spatial perception problems and found TLS to be more robust, generally up to 80% outliers, and have additionally applied GNC-TLS to point cloud registration [99], shape reconstruction [100], and shape alignment [17].

2.1.2 Certifiable Perception

The success of TLS as a robust loss in spatial perception problems, but the local and heuristic nature of GNC has led researchers to consider ways of globally solving or certifying solutions to robust perception problems. Globally optimal estimation algorithms have been developed in the setting of outlier-free geometric perception using nonlinear least squares formulations (i.e., standard squared residual loss). These algorithms typically relax the problem using Shor's relaxation [101] and solve the resulting convex semidefinite program (SDP) to compute a global solution and a numerical certificate of optimality, typically based on the rank of the SDP solution.

The certificate of optimality relies on a zero-duality gap and leads to a certifiable algorithm [102]. See Carlone [49] for an excellent introduction and overview of robust and certifiable geometric perception.

2.2 Data Association

2.2.1 Pairwise Correspondence

Associating elements from two sets based on inter-element similarity scores is traditionally formulated as a linear assignment problem (LAP) [103, 104], which can be solved to global optimality in polynomial time using, e.g., the Hungarian (Kuhn-Munkres) [50] algorithm or the auction algorithm [58]. The Hungarian algorithm produces *perfect matchings* (one-to-one correspondence with all elements matched) for *balanced* matching problems (same number of elements in both sets). Imperfect (one-to-one correspondence, all items need not be matched) or unbalanced matching problems can be reduced to perfect, balanced matching and solved with the Hungarian algorithm [105]. The case where an imperfect matching of any size is sought that maximizes the possible benefit is called the maximum-weight matching (MWM) problem [106].

Other approaches to solving the LAP for pairwise association include greedy techniques, such as matching feature points between two views based on nearest neighbors. Often smart heuristics are included to detect outlier correspondences, such as Lowe's ratio test [107] in feature-based image matching. These heuristics are frequently used in practice because of their simplicity and speed, but can lead to severely sub-optimal correspondence selection.

If elements have underlying structure (e.g., geometric structure of 3D point clouds) that should be included in the association decision, the problem can be formulated as a quadratic assignment problem (QAP) [43, 108, 109]. The QAP is equivalent to the graph matching [110] problem. Unlike the LAP, the QAP (and its equivalent graph matching formulation) is, in general, NP-hard [111]. Exact methods for solv-

ing quadratic assignment use expensive branch and bound techniques [112, 113]. To improve computational efficiency, approximate solutions based on relaxations of the original problem are obtained, with examples including spectral relaxations [114], dual decomposition [115], linear relaxations [116], convex relaxations [117–120], path following [121–123], or alternating directions [124].

Consistency Graph Formulations

While the QAP considers pairwise similarity between associations, it does not guard against selecting inconsistent associations. In contrast, consistency graph formulations operate under the notion of identifying sets of pairwise correspondences that are *geometrically consistent*. Each putative correspondence is represented as a node in a graph and edges are formed between nodes depending on the pairwise consistency of those correspondences. Finding the largest set of pairwise consistent correspondences (i.e., connected nodes) then becomes a graph optimization problem. Most commonly, the weighted consistency graph is thresholded to an unweighted graph and the largest set of pairwise consistent correspondences is found by solving the NP-hard maximum clique problem [125], which in general is very difficult to approximate [126].

One of the first instances of a consistency graph construction was given by Ambler et al. [127] for model-based visual recognition of parts in an early computer-controlled assembly system. By extracting discrete properties of part structures (e.g., an edge-length could be long, medium, or short), initial correspondences can be made between structures with the same property. If a pair of initial correspondences had the same binary relation (e.g., a long edge followed by a short edge), then an edge would be added between the nodes representing the correspondences and the largest set of matching relational structures was found by solving the maximum clique problem. Bolles [128] adopted this framework and extended its use to continuous similarities, such as edge length or hole size, still creating unweighted edges based on a similarity threshold and finding the maximum clique. More recently, Bailey et al. [12] used this graph framework in 2D LiDAR scan matching and sought the correct data association via the maximum common subgraph, which can be equivalently solved via the

maximum clique problem. Enqvist et al. [129] developed a method for 3D-3D and 3D-2D registration by approximate vertex covering. Bustos et al. [130] proposed an exact method for identifying the maximum clique in point cloud registration problems based on branch and bound and graph coloring. Mangelson et al. [25] developed the pairwise consistent measurement set maximization (PCM) algorithm for loop closure filtering, where the graph encodes the consistency of loop closures in SLAM and the maximum clique provides an estimate of the inlier set. Yang and Carlone [88] provide an algorithm for robust point cloud registration, where estimation in extreme outlier rates are feasible due to a maximum clique inlier selection step, which was further explored and formalized in [99, 131].

An alternate approach to mining pairwise-consistent correspondences from consistency graphs was presented in parallel by Leordeanu and Hebert [114] and Olson et al. [132]. In these works, the weighted consistency graph is used directly (i.e., without thresholding) to identify the *densest edge-weighted subgraph*, which can be solved in polynomial time [133] using the max-flow, min-cut theorem. Note that the key difference between the maximum clique problem and the dense edge-weighted subgraph problem (apart from unweighted vs weighted) is the inclusion of the clique constraint. Enforcing this constraint ensures that the selected correspondences are all mutually pairwise consistent, and so excluding this constraint leads to poor performance in high-outlier settings. Recent work continues to leverage these algorithms; for example, in wide-disparity RGB-D image matching [134] and RADAR-based odometry [135].

2.2.2 Multiway Correspondence

Pairwise data association is frequently used due to its simplicity, the availability of established techniques, and the computational efficiency of those techniques. However, pairwise data association is limited in its ability to exploit global structure or relationships among multiple sets of elements, and it can be sensitive to noise or outliers. For example, in vision-based object tracking, frame-to-frame pairwise associations are often utilized, but a single incorrect association will cause performance degradation

in terms of localization accuracy and track ID switching. Instead, if data association decisions can be delayed, a window of $n > 2$ frames can be used to capture the global structure of measurements using more sophisticated multiway association techniques and noisy pairwise data associations can be corrected by ensuring *cycle consistency* of associations. The tradeoff of including the additional information from many data association pairs is an increase in computational complexity.

Multiway data association is predominately formulated as a permutation synchronization problem [46], which is computationally challenging due to its binary domain. With the exception of exact combinatorial methods [136, 137] that do not scale well to large problems, and a recent deep leaning approach in [138], 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. Other approaches include filtering by cluster-consistency statistics [65], message passing [66], and iteratively reweighted least squares [139].

Methods in the first category include [59], which proposes to solve a semidefinite programming relaxation of the problem via ADMM [140]. A distributed variation of this method with a similar formulation has been recently presented in [141]. Toward the same goal, works such as [56,62,64] use low-rank matrix factorizations to improve the computational complexity. Works such as [142] and [61] require full observability, whereas methods such as [63,143] 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 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 [46] and [60]. The method proposed in [46] returns consistent solutions from noisy pairwise associations using a spectral relaxation in the fully observable setting. The work done by [60] proposes an eigendecomposition approach that works in a partially observable setting, however cycle consistency is lost in higher noise regimes. The recent work of [144] leverages a non-negative matrix factorization approach to solve the problem. This method works in a partially observ-

able 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 [145] and [57], 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 [145] has considered a constrained clustering approach using a method similar to k-means. In [57], the existing density-based graph clustering algorithm in [146] is leveraged to solve the problem. The method of Tron et al. [57] is extended in [147] and applied to a multirobot feature matching problem. Methods in this category could be very fast, though accuracy may be compromised.

With the exception of [56, 57], the aforementioned methods were originally designed with permutation synchronization in mind, thus expecting binary associations as input. This expectation requires additional front-end processing, where an initial pairwise data association algorithm must produce hard association decisions, thus reducing the amount of information available to the multiway association algorithm.

Lastly, note that when underlying structure is incorporated, the multiway data association becomes a multi-graph matching problem [148–152] (equivalently, multi-QAP), which is considerably more computationally challenging. In these works, geometrical information between the items in each view is incorporated into the problem. However, the additional complexity, in general, results in significantly slower algorithms.

2.3 Map Representations and Abstractions for Data Associations

Data association and registration algorithms are commonly referred to as either local or global, depending on if an initial guess is required. Local methods, like ICP [95], are often used in scan matching, where consecutive scans typically have small dis-

placement between them. In contrast, global methods do not need an initial guess to succeed and are often preferred in settings like loop closure detection because no good initial guess exists. In point-based registration, global methods first generate candidate point correspondences, typically based on local descriptors [51, 52, 153]. The set of putative correspondences are likely to contain incorrect matches, called outliers, and so robust iterative estimation techniques like RANSAC [36] or graduated non-convexity [42, 98, 99] can be used to select a subset of correspondences that best support the model. Consistency-graph-based approaches (see Section 2.2.1) provide an alternative technique that is both deterministic and extremely robust in the presence of outliers.

Lines (e.g., poles) and planes commonly exist in man-made environments and have recently been used as landmarks in LiDAR navigation. The benefits of using higher-order geometric primitives include lower storage and processing requirements since there are fewer line and plane objects than points [72, 154, 155], and more accurate odometry because it is infeasible to get exact point-to-point correspondences from sparse LiDAR point clouds [156]. Local methods for landmark-based registration rely on identifying the closest landmarks between two scans, given an initial alignment guess. Nearest neighbor search requires calculating distances between landmarks, which is dependent on the landmark representation. The most common representations include vector form for lines, Hesse normal form for planes, and the so-called "closest point" (CP) vector for both lines and planes [157]. The CP representation compactly encodes position and vector orientation in a 3-vector, and the Euclidean distance is often used to find similar landmarks [54, 156]. For lines in vector form or planes in Hesse normal form, both the angle between vectors and the distance between points is used [53]. If points are retained in one of the scans, then the point-to-landmark RMSE can also be used [55]. Other methods [154, 155, 158] project the centroids of the detected landmark onto the ground plane, creating 2D points. However, these methods assume the landmark (infinite line or plane) has a well-defined centroid (which is not true for elements of infinite extent, and is extremely sensitive to detection accuracy for elements of finite extent), the ground plane is known, and

that 2D registration is sufficient.

Since descriptors for 3D lines and planes have not been thoroughly explored, most global methods rely on a series of geometric tests to assign correspondences [159]. LiPMatch [160] adopts an interpretation tree for plane matching [161] in loop closure detection, using unary and binary constraints between candidate plane matches to determine the largest set of consistent matches. However, some of these constraints are sensitive to viewpoint change (e.g., centroid, area). ClusterMatch [72, 162] matches poles/lines and planes by iteratively searching for landmark pairs with similar pairwise centroid distance until a large number of matches supports the resulting transformation. A critical drawback of these methods is that, although global methods, the heuristic geometric tests that are used are often heavily view dependent.

Chapter 3

Graph-Theoretic Framework for Robust, Pairwise Data Association

3.1 Introduction

Data association is a fundamental requirement of geometric estimation in robotics. Identifying correspondences between measurements and models enables estimation processes to incorporate more data, in general leading to better estimates. However, sensor data is replete with noise and spurious measurements, making data association considerably more challenging. Further, while some estimation processes (e.g., ICP [95]) leverage an initial estimation guess to reduce the difficulty of data association, high-quality initial guesses are frequently not available. Thus, putative correspondences generated by the data association step are often contaminated with incorrect matches. Additional effort is then required to identify which measurements are useful, as even a small number of corrupted measurements can cause an estimator to diverge [39]. Note that the difficulty of this requirement is in the meaning of "useful"—in general, the classification of measurements and their correspondences as inliers or outliers is unobservable.

Conventional methods employ *consensus maximization* [35] to reject outliers *during* estimation, so that model parameters are sought that explain the largest subset of the input data. However, this requires distinguishing between inliers and outliers

via a pre-specified error threshold, which is difficult to determine in practice. In fact, measurements which satisfy the error threshold may still be outliers in the sense that they were actually generated from a different model [79]. Alternatively, estimation in the presence of contaminated data can be performed via *M-estimation* [163]. In this framework, the effects of outliers are discounted *during* estimation by minimizing a set of robust loss functions defined over residuals, thereby using residuals as a means of outlier detection. This technique has been successfully used in computer vision [164] and robotics [165] settings with a moderate number of outliers. However, common robust losses (e.g., the convex Huber loss) have low theoretical breakdown points and often exhibit breakdown with data having a low proportion of outliers [82, 83, 89]; in fact, even a single "bad" outlier can lead to estimator bias [89].

Rather than attempting to distinguish inliers from outliers, we instead aim to select the largest set of *mutually consistent correspondences*, thus avoiding the unobservable nature of inlier–outlier classification and also avoiding the necessity of an initial estimation guess. By selecting good measurements *before* estimation, we focus on making the data association stage more robust, thereby increasing the quality of data before it is used for estimation. By substantially reducing the number of incorrect correspondences in the data association module, classical robust estimation techniques become applicable again, even when the input data is heavily contaminated with outliers. Additionally, by clipping outlier associations early in the data processing pipeline, the computational burden of processing invalid measurements during estimation is reduced.

We present a method for robust data association based on an edge-weighted graph, where edge weights encode the pairwise consistency of potential associations. Using this graph, we formulate consistent measurement selection as a novel combinatorial optimization problem which seeks the *densest* edge-weighted clique (DEWC). Due to the computational challenges associated with solving for the DEWC in practice, we explore a continuous relaxation based on the maximum spectral radius clique (MSRC). We find that the MSRC problem has an empirically large basin of attraction with respect to the global optima of the DEWC on a variety of relevant data

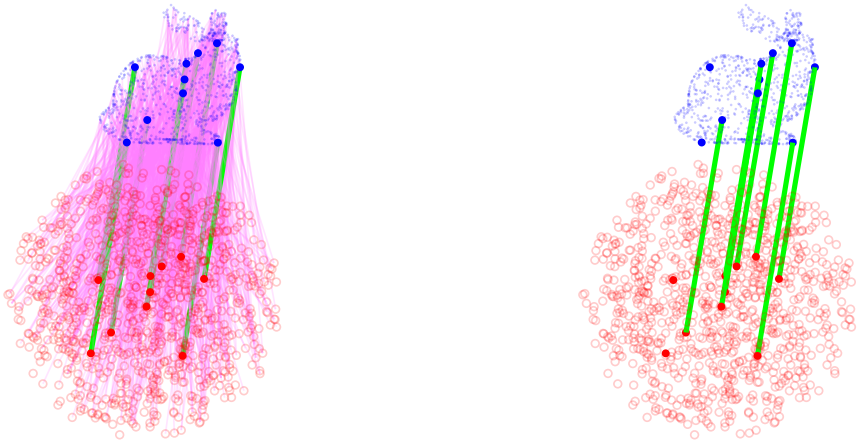


Figure 3-1. Robust data association using CLIPPER on Stanford Bunny. Only 10 of 1000 blue points in view 1 were seen in view 2 (discs), simulating a partial view with 1000 additional outlier points (circles). (left) Putative associations between the two point clouds having 99% outlier associations, shown by magenta lines, and 1% inlier associations, shown by green lines. (right) CLIPPER removes all outlier associations with 80% recall in 6 ms.

association problems and has connections to spectral matching [114,132], while being distinct in the fact that it enforces solutions to be *complete* subgraphs (i.e., cliques). Taken together, the use of edge weights in the density objective allows for more expressive modelling of association consistency while the clique constraint ensures that selected associations are jointly consistent, hence affording increased robustness.

To approach the non-convex MSRC problem, we introduce a convex semidefinite relaxation following Shor's relaxation [101], enabling recovery of the globally optimal MSRC provided satisfaction of the rank constraint, notwithstanding the non-convexity of the original MSRC problem. In practice, we find that the rank constraint is satisfied in most cases we consider and that the MSRC-SDR can be solved in one second or less for problems with 100 or less putative associations. Finally, we present a separate, computationally-efficient MSRC solver by leveraging a homotopy-based [166], graduated projected gradient ascent framework. We find this first-order method to perform well, solving problems with 8000 associations in one second or less and frequently returning the globally optimal solution in the problems we consider. This algorithmic framework, called CLIPPER (Consistent LInking, Pruning, and Pairwise Error Rectification), outperforms the state-of-the-art in consistency graph formulations. An example application of CLIPPER is shown in Fig. 3-1.

The contributions of this chapter can be summarized as:

1. A novel DEWC formulation for pairwise data association that provides a high degree of robustness to outlier associations as compared to existing pairwise data association techniques.

2. The MSRC problem as a relaxation of the DEWC with an empirically large basin of attraction to the DEWC and a connection to spectral matching.

3. A convex semidefinite relaxation of MSRC that empirically produces globally optimal solutions.

4. A scalable, computationally efficient projected gradient ascent algorithm called CLIPPER that is shown to outperform the state-of-the-art in pairwise data association.

3.2 Background

3.2.1 Consistent Correspondence Selection

Data association is hard because the classification of correspondences as inliers or outliers is unobservable. Instead, we aim to select the group of correspondences that are most consistent with each other and the estimation model. To do so, we first discuss the consistency graph and then present our problem formulation for consistent correspondence selection.

Given two sets of data $\mathcal{P}, \mathcal{Q}$ and a set of putative associations $\mathcal{A} \subset \mathcal{P} \times \mathcal{Q}$, the data association problem can be viewed as a bipartite graph $\mathcal{B} = (\mathcal{P}, \mathcal{Q}, \mathcal{A})$ where $\mathcal{P}, \mathcal{Q}$ are disjoint and independent sets of vertices and $\mathcal{A}$ are the edges between $\mathcal{P}$ and $\mathcal{Q}$. Defining $u := (p, q) \in \mathcal{A}$, a pair of associations $u_i, u_j \in \mathcal{A}$ is called *consistent* if the underlying data p_i, p_j can be mapped into q_i, q_j using a single mapping. However, noise prevents a perfect mapping and so a *consistency score* $s : \mathcal{A} \times \mathcal{A} \to [0, 1]$ is used, with 0 being inconsistent and 1 being fully consistent. The scoring function typically relies on an *invariant*—that is, a property that does not change under the

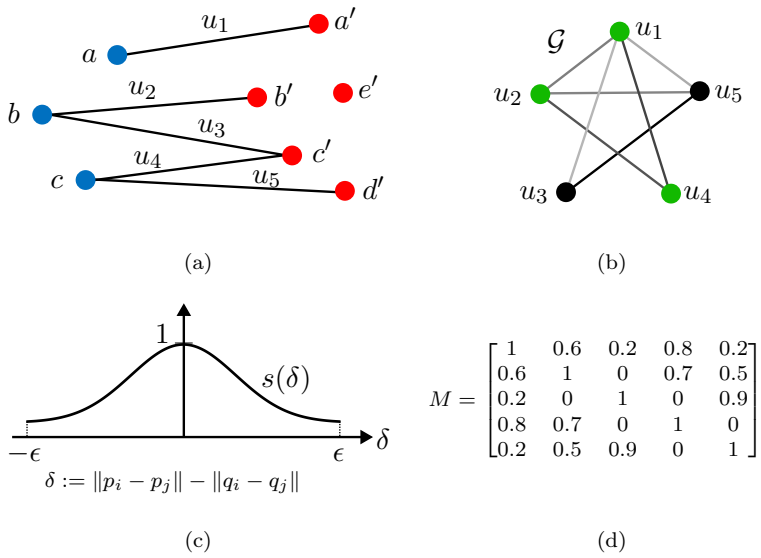


Figure 3-2. Consistency graph construction example for point cloud registration. (a) Putative associations $u_1, \ldots, u_5 \in \mathcal{A}$ are given between red and blue point clouds. (b) The consistency graph $\mathcal{G}$ with vertices representing the associations and edges between two vertices indicating their geometric consistency. In the noiseless case, any two associations u_i, u_j mapping points p_i, p_j to q_i, q_j are consistent if $\delta = 0$, where $\delta := \|p_i - p_j\| - \|q_i - q_j\|$. The correct associations are colored green. (c) Edges of the consistency graph are weighted according to the pairwise consistency score function $s(\delta)$. If $\delta > \epsilon$ or if two associations start/end at the same point, the association pair is deemed inconsistent. (d) The affinity matrix M is the numerical representation of the consistency graph $\mathcal{G}$.

mapping (e.g., Euclidean distance under rigid-body transformation [69,131]), though non-invariant properties have also been used (e.g., Euclidean distance under projective transformation [114]). By scoring each of the $\binom{|\mathcal{A}|}{2}$ association pairs, a *consistency graph* $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ can be formed, with associations $\mathcal{A}$ as the vertices and edges $\mathcal{E}$ weighted according to the pairwise consistency score s. Note that various problem-specific constraints (i.e., element $p \in \mathcal{P}$ can only be matched to one element $q \in \mathcal{Q}$) can be enforced by removing certain edges in $\mathcal{E}$. These constraints are especially important in the all-to-all hypothesis case, i.e., no prior information is available (e.g., no descriptor matching) and so every possible association is given in the putative association set $\mathcal{A}$.

Without loss of generality, we focus on consistency graph construction for the point cloud registration problem. However, consistency graphs can be created for other data association and geometric estimation problems (e.g., rotation averaging [131], plane/line alignment [70, 71]). Point cloud registration is concerned with finding the

rotation and translation that best align a point set to their corresponding points in another set. Identifying correct point correspondences is the main challenge due to noise and outliers. *Outlier points* are spurious measurements, typically generated due to noisy sensing or partial observation, while *outlier associations* are correspondences that incorrectly match two points from either set. Fig. 3-2 illustrates consistency graph construction given a set of putative associations (i.e., the lines denoted with u_i) between blue and red point clouds (see Fig. 3-2a). Inlier points are denoted by $a, b, c \in \mathcal{P}$ in the blue point cloud and their transformed counterparts by $a', b', c' \in \mathcal{Q}$ in the red point cloud. Points $d', e' \in \mathcal{Q}$ do not correspond to any blue point and hence are considered as outliers. Thus, the associations denoted $u_1, u_2, u_4 \in \mathcal{A}$ are inliers and $u_3, u_5 \in \mathcal{A}$ are outliers.

To build the consistency graph (illustrated in Fig. 3-2b), the consistency score function s must be defined for point cloud registration problems. Because rigid-body transformation (i.e., the unknown variable of the registration problem) is distance preserving, the Euclidean distance between points in one set should be identical (in the noiseless setting) to the Euclidean distance between their counterparts in the other set. Thus, with $\delta := \|p_i - p_j\| - \|q_i - q_j\|$ for associations $u_i, u_j \in \mathcal{A}$ and slight abuse of notation on the domain of s, we define

$$s(\delta) := \begin{cases} \exp(-\frac{1}{2}\frac{\delta^2}{\sigma^2}) & |\delta| \leq \epsilon \\ 0 & |\delta| > \epsilon \end{cases}, \tag{3.1}$$

where the threshold ϵ is based on a bounded noise model with a noise radius of $\epsilon/2$ on point coordinates [99] and the parameter σ controls how consistent an association pair with noisy underlying points is (see Fig. 3-2c). Note that consistency scores of correct association pairs are higher, which will motivate our problem formulation in Section 3.3.1.

Finally, the *affinity matrix* $M \in [0,1]^{m \times m}$ of the consistency graph in Fig. 3-2b is shown in Fig. 3-2d. The affinity matrix is an $m \times m$ symmetric matrix, where $m := |\mathcal{A}|$ is the number of putative associations. While the off-diagonal M_{ij} terms encode

the pairwise consistency of associations u_i, u_j, the diagonal M_{ii} terms encode single association consistency by measuring the similarity of points directly, for example, by comparing descriptor similarity of points p_i, q_i. These terms M_{ii} are simply set to 1 when this information is unavailable.

3.3 Approach

3.3.1 Problem Formulation

Given a consistency graph $\mathcal{G}$ and assuming that inlier associations $\mathcal{A}_{\text{in}} \subset \mathcal{A}$, our goal is to identify the subgraph $\mathcal{C} \subset \mathcal{G}$ whose vertex set is equal to $\mathcal{A}_{\text{in}}$. Towards this goal, we make two observations about the properties of $\mathcal{C}$. First, because each $(p, q) \in \mathcal{A}_{\text{in}}$ is such that p can be mapped onto q given a single mapping, then inlier associations are *mutually consistent*, which forces $\mathcal{C}$ to be a clique (each vertex is connected to every other vertex). Second, under the assumption that noise and outliers are random and unstructured, then $\mathcal{C}$ is the "largest" (in some sense) clique in $\mathcal{G}$ [114]. In particular, we propose that $\mathcal{C}$ can be identified as the *densest-edge weighted clique* (DEWC) via the following optimization formulation

$$\begin{aligned} &\underset{u \in \{0,1\}^m}{\text{maximize}} \quad \frac{u^\top M u}{u^\top u} \\ &\text{subject to} \quad u_i u_j = 0 \quad \text{if } M_{ij} = 0, \ \forall_{i,j}. \end{aligned} \tag{3.2}$$

Here, the optimization variable u is a binary vector with 1's indicating selected associations and 0's otherwise. Since u is binary and the objective is to maximize, the constraint $u_i\, u_j = 0$ enforces the subgraph induced by u to be a clique. The objective evaluates the *density* of the induced subgraph, which is defined as the total sum of edge weights divided by the number of selected vertices. Thus, (3.2) shares the objective function of the spectral matching technique [114], while enforcing mutual consistency via the clique constraint.

When M is binary (e.g., $s(\delta) := 1$ for $|\delta| \leq \epsilon,\ 0$ otherwise and has 1's on the diagonal), it is straightforward to show that (3.2) simplifies to the *maximum clique* (MC)

problem

$$\begin{aligned} \underset{u\in\{0,1\}^m}{\text{maximize}} \quad & \sum_{i=1}^{m} u_i \\ \text{subject to} \quad & u_i u_j = 0 \quad \text{if } M_{ij} = 0,\ \forall_{i,j}. \end{aligned} \tag{3.3}$$

This definition of s corresponds to frequently used *unweighted* consistency graph frameworks [12,99,129,131], which effectively *ignore* the consistency information captured by (3.1). This can be problematic in the case of *competing cliques*, e.g., if the graph in Fig. 3-2b where unweighted, problem (3.3) cannot disambiguate between $\{u_1, u_2, u_4\}$ (correct) or $\{u_1, u_3, u_5\}$.

To highlight the importance of density in (3.2), consider the following affinity matrix M with two solution candidates u, u'

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.2 & 0.2 \\ 0 & 0 & 0.2 & 1 & 0.2 \\ 0 & 0 & 0.2 & 0.2 & 1 \end{bmatrix}, \; u = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; u' = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.4}$$

Both the MC objective (3.3) (with non-zero M_{ij} set to 1) and the unnormalized objective of $u^\top M\, u$ return u' as the optimum solution. However, the block of M corresponding to u' has low pairwise consistency scores of 0.2. On the other hand, the density objective (3.2) takes values of $2, 1.4$ for u, u', respectively, leading to selection of the smaller, but more consistent subgraph. While this problem could have been avoided by choosing a smaller ϵ in (3.1), a conservative threshold may lead to rejecting correct associations (i.e., lower recall).

3.3.2 Non-Convex Continuous Relaxation

A core challenge in solving (3.2) is the combinatorial complexity of the problem due to its binary domain. This makes it intractable to solve (3.2) to global optimality in real time, even for small-sized problems (see Fig. 3-6). A standard approach is to relax the

binary domain to the reals, which often facilitates faster optimization. This approach is further motivated in our case due to the observation that the objective of (3.2) is the Rayleigh quotient, whose maximizer over the reals is simply the principal eigenvector of M. Thus, we relax the binary domain and leverage the scaling invariance of the Rayleigh quotient, yielding an optimization over the non-negative reals

$$\begin{aligned} \underset{v \in \mathbb{R}^m_+}{\text{maximize}} \quad & v^\top M\, v \\ \text{subject to} \quad & v_i v_j = 0 \quad \text{if } M_{ij} = 0,\ \forall_{i,j} \\ & \|v\|_2^2 \leq 1, \end{aligned} \tag{3.5}$$

which can be understood as a constrained eigenvalue problem. In fact, due to the graph interpretation of M, problem (3.5) seeks the *maximum spectral radius clique* (MSRC) of M. To recover a solution that is feasible with respect to (3.2), a *rounding* step is required to project v onto the binary domain. This projection can be performed exactly by using Goldberg's polynomial-time algorithm [133] to solve the *densest subgraph* (DS) problem (e.g., (3.2) without clique constraints) on the clique induced by the non-zero elements of v.

3.3.3 CLIPPER Algorithm

Optimizing for the MSRC (3.5) would simply be the principal eigenvector if the clique constraints were omitted. Note that because $M \in [0,1]^{m \times m}$, by the Perron-Frobenius theorem, the principal eigenvector $v^* \in [0,1]^m$, thus satisfying the non-negative real constraint [114]. However, the inclusion of the clique constraints makes the problem more challenging; therefore, we leverage the following penalty form

$$\begin{aligned} \underset{v \in \mathbb{R}^m_+}{\text{maximize}} \quad & v^\top M_d\, v \\ \text{subject to} \quad & \|v\|_2^2 \leq 1, \end{aligned} \tag{3.6}$$

where $M_d := M - d\,C$ encodes the clique constraints using the penalty parameter $d > 0$ and the matrix $C \in \{0,1\}^{m \times m}$ with $C_{ij} := 1$ if and only if $M_{ij} = 0$. Thus,

when $C_{ij} = 1$, the joint selection of v_i, v_j is penalized by the amount $-2d\, v_i v_j$. Hence, as d increases the entries of solution v that violate the constraints are pushed to zero. In fact, following the reasoning of [167] for binary M (since M can be rounded to a binary matrix), when $d \geq m$, solutions of (3.5) are guaranteed to satisfy the clique constraints. The intuition of the penalty parameter d is that of other continuation or homotopy approaches [166]: when $d = 0$ the problem is easily solved and so the solution can be used to warm start the next optimization and so on as d is incrementally increased. At each value of d, we optimize (3.5) using projected gradient ascent (PGA).

Given a feasible solution v^* to MSRC (3.5), let $\mathcal{G}|_{v^*} \subseteq \mathcal{G}$ denote the (necessarily complete) subgraph of $\mathcal{G}$ induced by the non-zero elements of v^*. As stated in Section 3.3.2, $v^* \in \mathbb{R}^m_+$ can be rounded to a binary $u^* \in \{0,1\}^m$ by solving the DS problem. Instead of solving for the DS exactly, we use a fast heuristic that can immediately return a binary u^* by selecting the $\hat{\omega} := \text{round}(v^{*\top} M\, v^*)$ largest elements of v^* as vertices of $\mathcal{G}|_{u^*} \subseteq \mathcal{G}|_{v^*} \subseteq \mathcal{G}$. The justification follows from the facts that $v^{*\top} M\, v^*$ (i.e., the spectral radius of $\mathcal{G}|_{v^*}$), is a tight upper bound for the graph's density [168] and that nonzero elements of v^*, (i.e., the principal eigenvector of $M|_{v^*}$) represent centrality of their corresponding vertices, which is a measure of connectivity for a vertex in the graph [169].

These steps, i.e., 1) obtaining a solution v^* of (3.5) via repeated PGA and 2) estimating the densest clique $\hat{\mathcal{C}} := \mathcal{G}|_{u^*}$ in $\mathcal{G}|_{v^*}$ by selecting the vertices corresponding to the $\hat{\omega}$ largest elements of v^*, constitute the CLIPPER algorithm, which is outlined in Algorithm 1. The core PGA method (Lines 6–9) utilizes backtracking line search for step size selection, followed by a projection onto the constraint manifold. Because solutions $v^* \in \mathbb{R}^m_+$ lie on the boundary of $\|v^*\| \leq 1$, the constraint manifold can be reduced to $\mathbb{R}^m_+ \cap \mathcal{S}^m$, where $\mathcal{S}^m$ is the unit sphere. Once the inner PGA has converged for a given value of d, the penalty is increased and the process repeats until $\mathcal{G}|_{v^*}$ satisfies the clique constraints given in C.

The update schedule chosen for d is motivated by the desire to quickly, but carefully converge to a feasible solution. Focusing on elements of v that contribute to

Algorithm 1 CLIPPER

1: **Input** affinity matrix $M \in [0,1]^{m \times m}$ of consistency graph $\mathcal{G}$
2: **Output** $\hat{\mathcal{C}} := \mathcal{G}|_{u^*}$, dense clique of feasible subgraph $\mathcal{G}|_{v^*} \subseteq \mathcal{G}$
3: $v \leftarrow \lambda_1(M)$ % initialize with global solution to (3.6) with $d = 0$
4: $d \leftarrow \text{mean}\{[Mv]/[Cv] : [Cv] > 0, [v] > 0\}$ % $[\cdot]$ element-wise
5: **while** clique constraints not satisfied **do**
6: **while** v not converged **do**
7: $v \leftarrow v + \alpha \nabla_v F$ % α via backtracking line search
8: $v \leftarrow \max(v/\|v\|, 0)$ % project back onto $\mathbb{R}^m_+ \cap S^m$
9: $d \leftarrow d + \text{mean}\{[Mv]/[Cv] : [Cv] > 0, [v] > 0\}$
10: $\hat{\omega} \leftarrow \text{round}({v^*}^\top M\, v^*)$ % estimate dense clique size
11: $\hat{\mathcal{C}} \leftarrow$ vertices corresponding to largest $\hat{\omega}$ elements of v^*

the violation of the clique constraints allows us to do so. The gradient of the objective (3.6) is $\nabla_v F = 2\,(Mv - d\,Cv)$. Observe that Cv is always non-negative and that non-zero entries $[Cv]_i$ indicate the entries of v that if increased would incur more penalty—these are the *potentially* problematic entries of v. Note that all the entries of v are in $[0, 1]$ and only the non-zero entries could contribute to clique constraint violations. Thus, to precisely identify the problematic entries, we find entries satisfying $[Cv]_i > 0, v_i > 0$. For each of those problematic entries v_i, we solve for the value of d that would cause $\nabla_v F_i \geq 0$ so that the step $-\nabla_v F$ causes v_i to diminish. The set of values that have this property for problematic entries is (i.e., set $\nabla_v F = 0$ and solve for d)

$$\mathcal{D} = \left\{ \frac{[Mv]}{[Cv]} : [Cv] > 0, [v] > 0 \right\}, \tag{3.7}$$

where the notation $[\cdot]$ indicates an element-wise operation. The number of entries diminished to zero is controlled by taking the maximum, median, or minimum of this set. We found that incrementing the penalty by the average of all such d's (Lines 4, 10) produces solutions that balance convergence speed and accuracy.

3.3.4 Globally Optimal CLIPPER

Due to the non-convexity of (3.5), searching for the MSRC in a consistency graph $\mathcal{G}$ by optimizing (3.5) may lead to local, suboptimal solutions. Thus, we transform (3.5) into a convex semidefinite program (SDP) which has only one (global) solution

by definition. The benefit is that if a rank condition on the SDP solution is satisfied, then we have actually recovered the global optimum of (3.5). Observing that (3.5) is a quadratically-constrained quadratic program, we utilize Shor's relaxation [101,170], which amounts to *lifting* the decision variable to $X = vv^\top$ and discarding the implied rank-1 constraint, which is non-convex. Following these steps yields

$$\begin{aligned} \underset{X \in \mathbb{D}^m}{\text{maximize}} \quad & \text{Tr}(MX) \\ \text{subject to} \quad & X_{ij} = 0 \quad \text{if } M_{ij} = 0, \ \forall_{i,j} \\ & \text{Tr}(Z) \leq 1, \end{aligned} \tag{3.8}$$

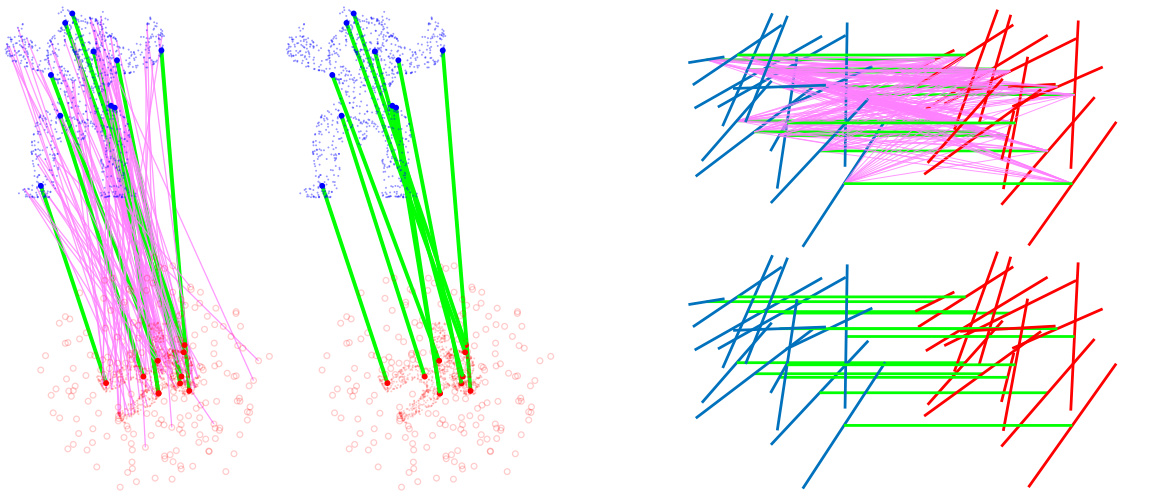
where the doubly non-negative cone $\mathbb{D}^m := \{X \in \mathbb{S}^m_+ | X \geq 0\}$ and $\mathbb{S}^m_+$ is the positive semidefinite cone. If the solution X^* to (3.8) has rank 1, then v^* can be recovered by the principal eigenvector of X^* and v^* is the global optimum of (3.5).

3.3.5 Constructing The Consistency Graph for Common Robotics Applications

CLIPPER can be applied to a broad array of data association problems found in robotics. All that is required is to identify a *geometric invariant* in the data, i.e., a quantity that is invariant under transformation. This invariant feature is then used to score geometric consistency for which a consistency graph $\mathcal{G}$ can be constructed and CLIPPER can be used to quickly remove outlier associations. We briefly review how to score geometric consistency for the application examples in Fig. 3-3.

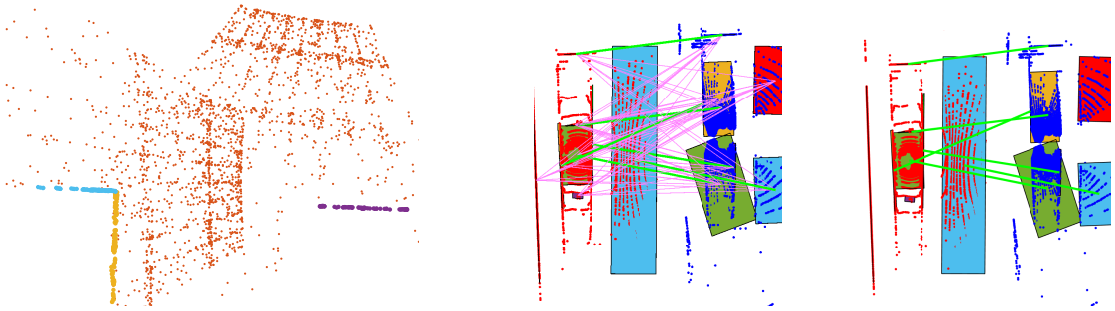
Point Clouds In Section 3.2.1 we described how two points seen in each cloud will have the same pairwise distance if the association is correct. This idea can be extended to scaled point clouds by using three point correspondences to form triangles for which correct associations will preserve the ratio of side lengths. This leads to a tensor formulation which can be marginalized into an $n \times n$ affinity matrix [172].

Line Clouds A line $l : (p, v)$ is given by a point $p \in \mathbb{R}^3$ and direction $v \in \mathbb{R}^3$. Given two sets of lines $\{l_1, \ldots, l_n\}$ and $\{l'_1, \ldots, l'_m\}$, a simple invariant feature is the angle between pairs of lines, resulting in a geometric consistency score defined



(a) Scaled point cloud registration

(b) Line cloud registration



(c) Plane registration (top-down view)

(d) Planar patch registration

Figure 3-3. Applications were the CLIPPER framework is used for robust data association: (a) noisy point cloud registration with unknown scale and outliers; (b) noisy line cloud registration; (c) plane cloud registration for LiDAR sensor calibration in outdoor, urban environment (planes indicated by colored points, sensor scans are correctly registered); (d) planar patch registration (extracted from LiDAR scans of [171]). In (a), (b), (c) magenta and green lines indicate incorrect and correct associations, respectively. Input associations can be generated from an external matching procedure or an all-to-all hypothesis in the case of no prior information. Each setting generates a consistency graph that CLIPPER operates on to identify which of the input associations are the most geometrically consistent.

by $s(|\text{acos}\, v_i^\top v_j - \text{acos}\, v_i'^\top v_j'|)$. The point p could also be incorporated to improve precision.

Plane Clouds A plane $\pi : (n, d)$ is given by its normal $n \in \mathbb{R}^3$ and distance from the origin d. An invariant feature of four planes is the four-way intersection point. However, the requirement of choosing four plane correspondences increases computational complexity. Instead, the simpler invariant of angle between normals n_i, n_j can be used resulting in the same consistency score as for line clouds.

Patch Clouds A cloud of planar patches, e.g., extracted from LiDAR using [173],

additionally provides the centroid and area of each patch. Although neither the centroid nor area are guaranteed to be invariant across views (e.g., partial view), these values can be used to assign a similarity score to corresponding planar patches by weighting the diagonal entries of the affinity matrix M. Geometric consistency is scored based on pairs of normals as with plane clouds.

3.4 Results

We evaluate our problem formulation and algorithms in the domain of point cloud registration. We demonstrate that our framework enables extreme robustness, even in the presence of many outliers and few inliers. Specifically, we find that correspondences selected by the DEWC lead to the lowest registration error compared to state-of-the-art formulations and that the CLIPPER algorithm produces solutions that are near-optimal with respect to DEWC. In addition to its accuracy, CLIPPER is computationally efficient and is shown to process as many as 8000 initial correspondences in one second or less.

The DEWC (3.2) and MSRC (3.5) are solved using Gurobi 10.0.2 and the DS is solved Goldberg's polynomial-time algorithm [133], implemented in C++. These global solutions are denoted as DEWC*, MSRC*, and DS*. The MSRC semidefinite relaxation (MSRC-SDR) (3.8) is solved using a custom C++ parser and optimized using SCS v3.0.0 [174]. In our experiments, we found that the the rank constraint was always satisfied for MSRC-SDR, indicating that the global optimizer of the MSRC (3.5) was found, and thus we denote its solutions as MSRC-SDR*. Two variations of the DS are also compared against: spectral matching (SM) [114] and single-cluster graph partitioning (SCGP) [132], both implemented in Python. We use the open-source, optimized C++ code of TEASER++ [99] to test ROBIN* [131], which solves the MC (3.3). TEASER++ first uses ROBIN* to reject outliers, followed by a custom TLS-GNC estimator for robust registration—we denote TLS-GNC tests without the maximum clique inlier selection stage as GNC. In the correspondence-based tests, we use the RANSAC implementation of Open3D 0.17 using the default

desired confidence of 99.9 % and denote variants limited to 10K, 100K, and 1M iterations. Fast global registration (FGR) [98] is also included. The C++ source code of CLIPPER and MSRC-SDR can be found at https://github.com/mit-acl/clipper. All experiments are run on a Linux computer with an Intel i9-7920X CPU with 64 GB of RAM.

3.4.1 Stanford Bunny Dataset

The Stanford Bunny [175] model is randomly downsampled to 500 points and scaled to fit within a $[0, 1]^3$ cube to obtain a source point cloud. The target point cloud is then created by adding bounded noise to each point. Following [99], bounded noise is added by sampling $\eta_i \sim \mathcal{N}(0, \gamma^2 I)$ and resampling if $\|\eta_i\| > \beta$. We set $\gamma = 0.01$, with $\beta = 5.54\gamma$ chosen so that $\mathbb{P}(\|\eta_i\|^2 > \beta^2) \leq 10^{-6}$. By adding noise, it is no longer clear which points correspond between the source and target point clouds. To identify inlier correspondences, the mutual nearest neighbor bounded by β is found for each point (when possible) in the source cloud. False correspondences are constructed by taking the complement of the true correspondences, i.e., an all-to-all correspondence with the ground truth correspondences removed. Putative correspondences are then simulated by combining a fraction of ground truth correspondences with false correspondences in accordance with the desired outlier rate. Finally, a random rigid-body transform (R, t) with $R \in \mathrm{SO}(3)$ and $t \in \mathbb{R}^3$ is applied to the target point cloud. At each desired outlier rate, we perform 30 Monte Carlo trials.

We compare and evaluate data association strategies using precision p and recall r, defined as

$$p = \frac{\#\text{ correctly selected}}{\#\text{ selected}}, \quad r = \frac{\#\text{ correctly selected}}{\#\text{ correct}}.$$

Further, we evaluate these algorithms and robust registration strategies in terms of rotation and translation error, defined as

$$r_{\text{err}} = \|\text{Log}(\hat{\boldsymbol{R}}^\top \boldsymbol{R})\|, \quad t_{\text{err}} = \|\boldsymbol{t} - \hat{\boldsymbol{t}}\|.$$

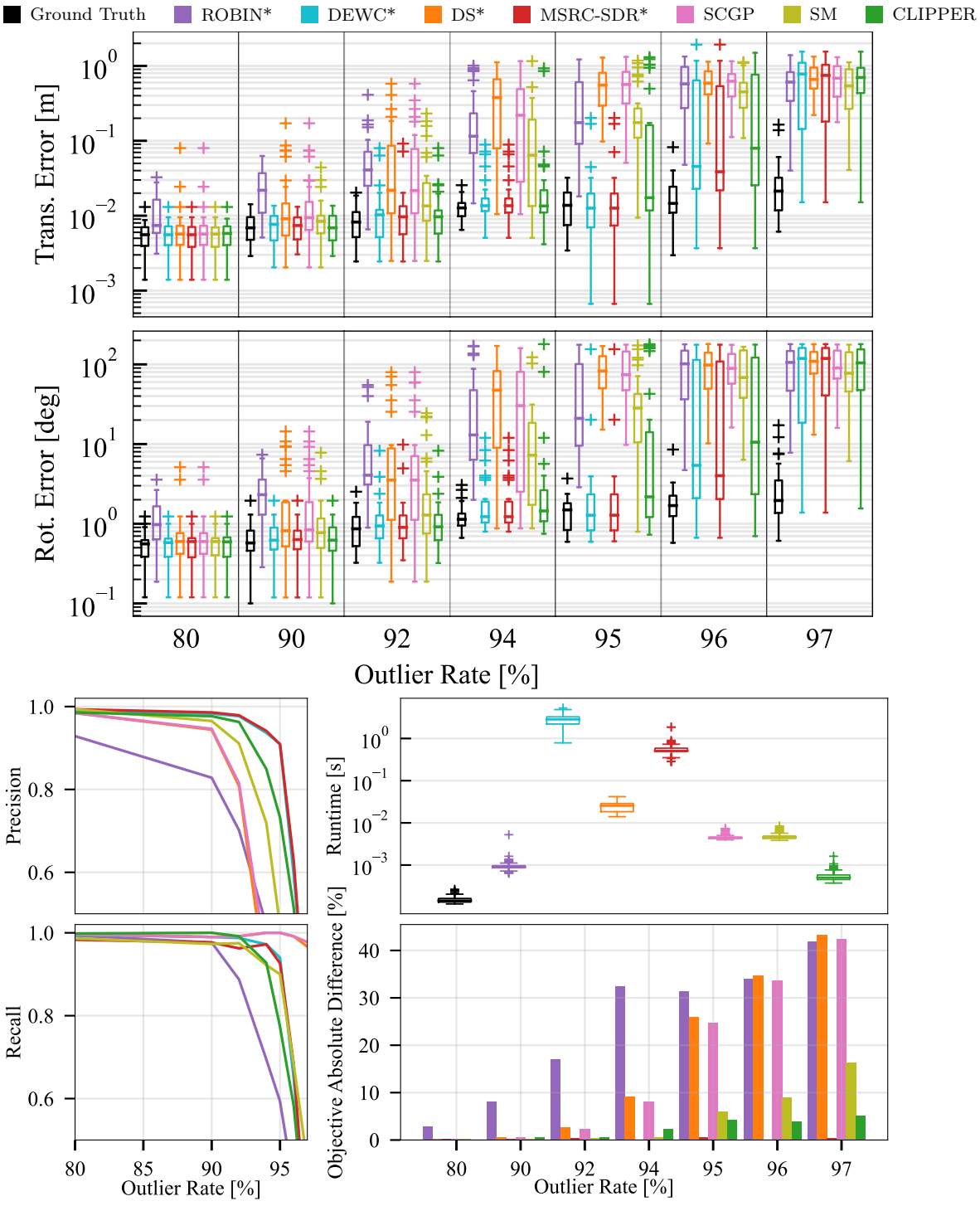


Figure 3-4. (1) DEWC*, MSRC* are better formulations (use weighted info), but are NP hard (2) CLIPPER is a relaxation that finds solutions close to DEWC / MSRC, (3) An important observation is that DS/SM/SCGP fail in high outlier regimes because of violation of clique constraint.

For the data association algorithms, the estimate $(\hat{\boldsymbol{R}}, \boldsymbol{t})$ is produced using Arun's (non-robust) least squares method [176].

Results. Because of the computationally intensive nature of DEWC*, we select $m = 100$ putative correspondences in the manner explained above over a range of high-outlier regimes. Note that at 97 % outlier ratio only 3 inlier correspondences remain—this is the minimum number of points for alignment. Registration error, precision and recall, runtime, and suboptimality results are shown in Fig. 3-4. Precision and recall plots in Fig. 3-4 verify that the DEWC (3.2) formulation and its relaxation MSRC (3.5) lead to robust data association, both yielding at least 98 % precision until 92 % outliers. CLIPPER performs nearly as well, but with a runtime that is 3 orders of magnitude faster. The effect of maintaining edge weights in the consistency graph is highlighted here—in terms of precision, the next best algorithms are SM, SCGP, and DS*. Finally, ROBIN* is the most sensitive to high outlier rates, confirming that thresholding consistency scores leads to a information loss. Rotation error is also shown in Fig. 3-4, where the "Ground Truth" uses all of the correct (noisy) correspondences in Arun's method and is included as an indication of the best achievable estimation error. As implied by the precision results, DEWC*, MSRC-SDR*, and CLIPPER achieve the best estimation error. Finally, the suboptimality of each solution with respect to DEWC* is shown in Fig. 3-4, showing that (1) MSRC and CLIPPER solutions are nearly optimal and (2) that the disregarding weighted consistency information or clique constraints cause relatively higher suboptimallity.

3.4.2 3DMatch Dataset

The 3DMatch dataset [177] contains RGB-D scans of 8 indoor scenes, each scene broken into a number of fragments. On average, each scene has approximately 200 ± 135 potential pairwise fragment registrations, as determined from the percent overlap. The dataset provides 5000 randomly sampled keypoints and their associated 512-dimensional FPFH [153] features. We randomly subselect 1000 of these keypoints, making the problem more challenging as there are fewer correct correspondences while being in high outlier regimes. Given a pair of fragments, we generate putative corre-

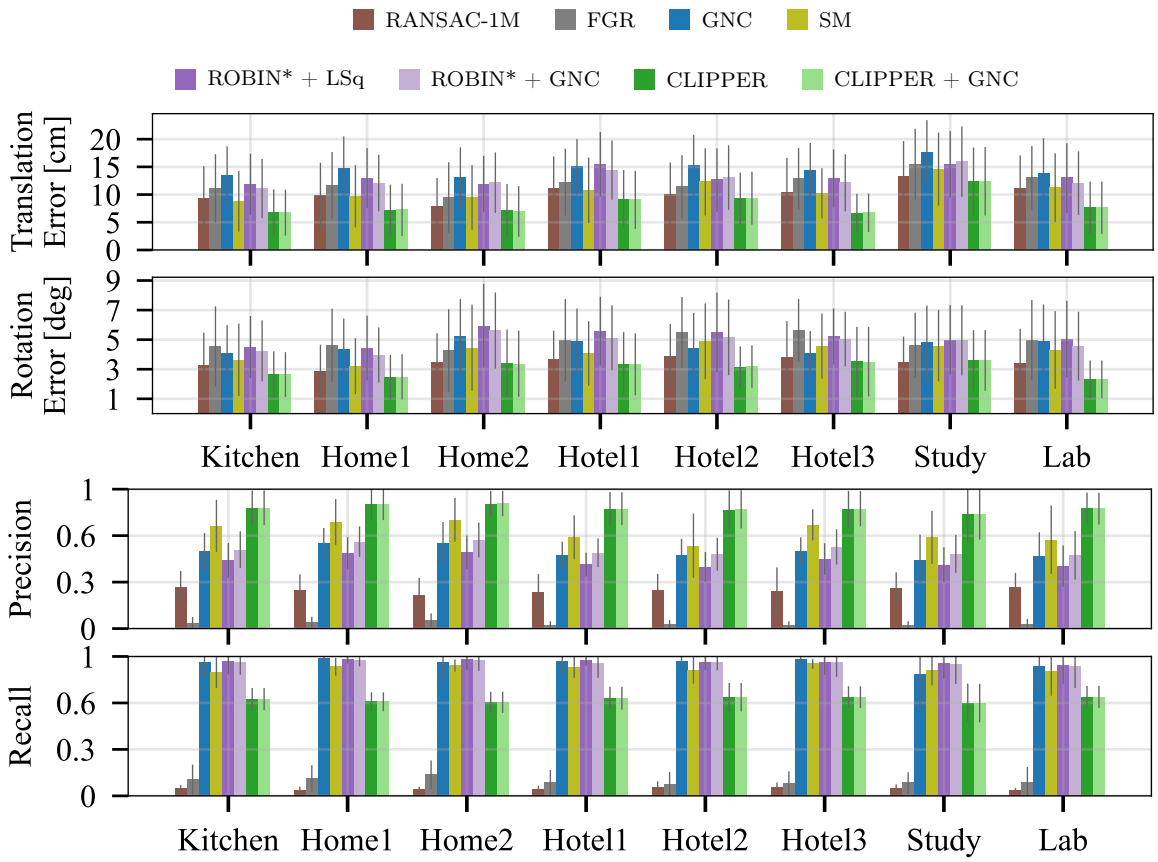


Figure 3-5. Average registration error, precision, and recall of successful registrations ($t_{\text{err}} \leq 30\,\text{cm}$ and $R_{\text{err}} \leq 15\,\text{deg}$) from the eight 3DMatch datasets. Not only does CLIPPER enable the most successful registrations (cf. Table 3.1), but it also produces the lowest registration error. This is due to its ability to achieve high precision, even in high outlier regimes. Because CLIPPER selects a high-precision set of correspondences, GNC does little to improve its performance.

spondences by searching for 2 nearest mutual neighbors in descriptor space for each keypoint of the target fragment. Following [99, 131, 177], we consider a registration successful if its error with respect to ground truth is less than 15 deg and 30 cm.

Results. Algorithm success rates are given in Table 3.1 for each dataset, where the input outlier rate and the average algorithm runtime is also shown. In addition to using Arun's non-robust least squares method for registration, ROBIN* and CLIPPER matches are also used for GNC-based registration. Note that ROBIN*+GNC is effectively TEASER++ [99] and that the effect of GNC is far less significant on the CLIPPER results. The average precision, recall, and registration error for successful matches are shown for each algorithm in Fig. 3-5.

Table 3.1. 3DMatch Registration Success Rates

	Kitchen	*Home1*	*Home2*	*Hotel1*	*Hotel2*	*Hotel3*	*Study*	*Lab*	Average Runtime [ms]
Outlier Rate [%]	99.0	98.7	99.0	99.0	99.2	99.3	99.2	99.1	
RANSAC-10K	13.8	16.7	18.8	5.8	6.7	5.56	6.16	10.4	18.8
RANSAC-100K	29.4	46.8	38.9	24.8	22.1	38.9	17.5	16.9	179.9
RANSAC-1M	55.5	60.9	52.4	44.2	34.6	46.3	33.6	41.6	1786.6
FGR	50.4	57.7	44.2	43.4	31.7	38.8	24.7	44.2	12.1
GNC	56.7	49.4	45.7	47.3	43.3	46.3	26.3	50.6	352.6
SM	50.6	60.3	50.9	47.3	38.5	40.7	29.8	50.6	307.8
ROBIN* + LSq	65.6	62.2	48.6	57.5	47.1	51.9	29.1	54.5	279.1
ROBIN* + GNC	69.2	64.1	57.2	61.5	51.9	55.5	32.2	**55.8**	284.4
CLIPPER + LSq	**77.4**	74.4	68.3	**69.0**	**64.4**	**70.4**	45.2	**55.8**	141.2
CLIPPER + GNC	**77.4**	**75.0**	**68.8**	**69.0**	**64.4**	**70.4**	**45.9**	**55.8**	141.7

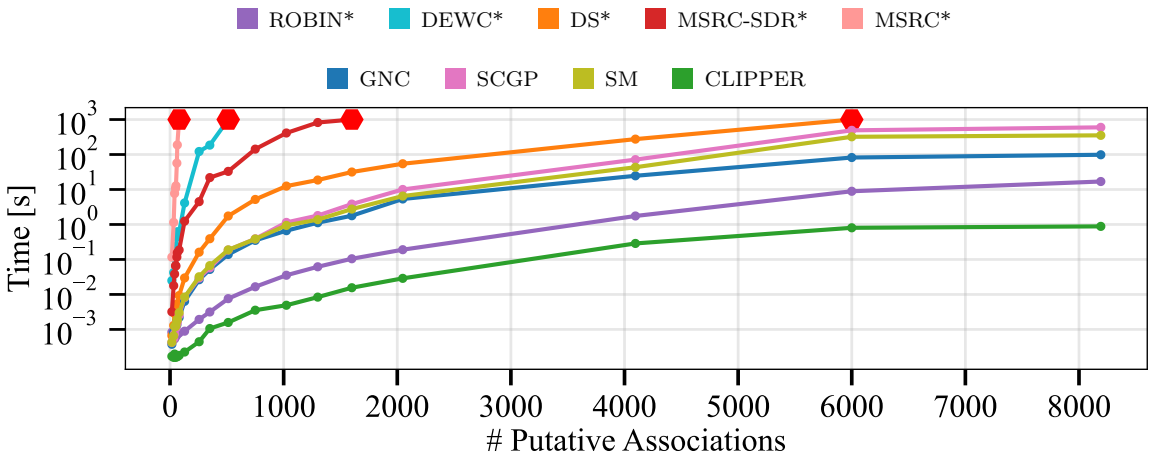


Figure 3-6. Scalability results. MSRC*, DEWC*, MSRC-SDR, and DS* had to be early stopped, indicated via the symbol ⬣.

3.4.3 Scalability Evaluation

Fig. 3-6 shows how the runtime of each algorithm scales with the number of putative associations, which were generated in the same way as described in Section 3.4.1 for an outlier ratio of 80 % such that each algorithm had an average of 90 % precision. Runtime results are generated on a single CPU core and are averaged over 10 trials, except for MSRC*, DEWC*, and MSRC-SDR*, which are averaged over two trials. These results indicate that CLIPPER can process 8000 associations in one second or less, achieving orders of magnitude faster performance compared to the state of the

art.

3.5 Summary

This chapter presented CLIPPER, a graph-theoretic framework for robust data association using the notion of geometric consistency. The key observations of this chapter are that (1) consistency-graph approaches allow for robust data association and (2) using the weighted edges in the consistency graph is key to achieving a high degree of robustness. CLIPPER was shown to consistently execute with low runtime and to outperform the state of the art, especially in very high outlier regimes where only 3-5 inliers remain. These gains were found by implementing an efficient projected gradient descent algorithm and by formulating the data association problem on weighted graphs rather than binary.

Chapter 4

Abstract Geometric Representations for Pairwise Data Association

4.1 Introduction

Estimating the rigid-body transformation between two sensors using their observations is a fundamental component of many mobile robotic systems. Wide-baseline registration is particularly challenging since an odometry signal may not be accurate, or even available, to use as an initial guess. This situation arises in core tasks such as loop closure generation, multi-robot map merging, extrinsic calibration, and global (re)localization. In these cases, the relative rotation and translation between sensors can instead be accurately estimated by matching co-visible features and optimizing for the best feature alignment.

In visual settings, appearance-based descriptors are commonly used for image retrieval or place recognition [178–180]. However, appearance-based techniques are often limited due to their sensitivity to illumination, weather, and viewpoint changes. Working with 3D sensors can alleviate these issues because of the geometric nature of the data [51, 181], but this requires storing and processing large point clouds, which can hinder online operation. In the context of LiDAR-based navigation, using geometric landmarks such as lines and planes has resulted in storage-efficient maps [72, 154, 155], better scene understanding [162, 182], and low-drift odometry and

mapping [53–55, 183]. However, existing geometric landmark matching techniques tend to assume 2D motion only [72, 162], use local association strategies given an initial guess from odometry [55, 156], or use a series of heuristic checks [159–161] that lead to low matching success rate.

Recent successes in consistency-graph-based data association (see Chapter 3) have significantly increased the robustness of the correspondence selection process in spatial perception problems by leveraging the notion of pairwise consistency. By matching constellations of objects that have consistent pairwise distances across two views, good correspondences can be identified even in the presence of many bad ones. Global data association techniques do not require an initial registration guess, but rely on a correctly defined pairwise distance that is invariant to transformation; for example, the Euclidean distance between two rigidly-attached points does not change even as those points are rotated and translated. Existing works frequently use Euclidean distance between line or plane pairs, using representations such as the centroid [72], which is not well-defined for infinite lines and planes, or the "closest point" (CP) parameterization [54], which lacks the necessary translation invariance because of its dependence on sensor origin.

Instead, this chapter represents line and plane landmarks naturally as elements of a Grassmannian manifold, which is the space of all linear subspaces. In particular, *affine* Grassmannian manifold is utilized, which allows for the representation of affine subspaces (i.e., linear subspaces not necessarily containing the origin). Thus, invariant distances between geometric primitives can easily be defined in a principled manner using the Grassmannian metric, enabling the use of CLIPPER for robust, global data association (cf. Chapter 3). Then, the rigid transformation between a pair of candidate loop closure scans can be estimated by solving a line and plane registration problem with known correspondences in the least-squares sense.

In summary, the main contributions presented in this chapter are:

1. the introduction of the affine Grassmannian for global data association of lines and planes, leading to 3D registration without requiring an initial guess;

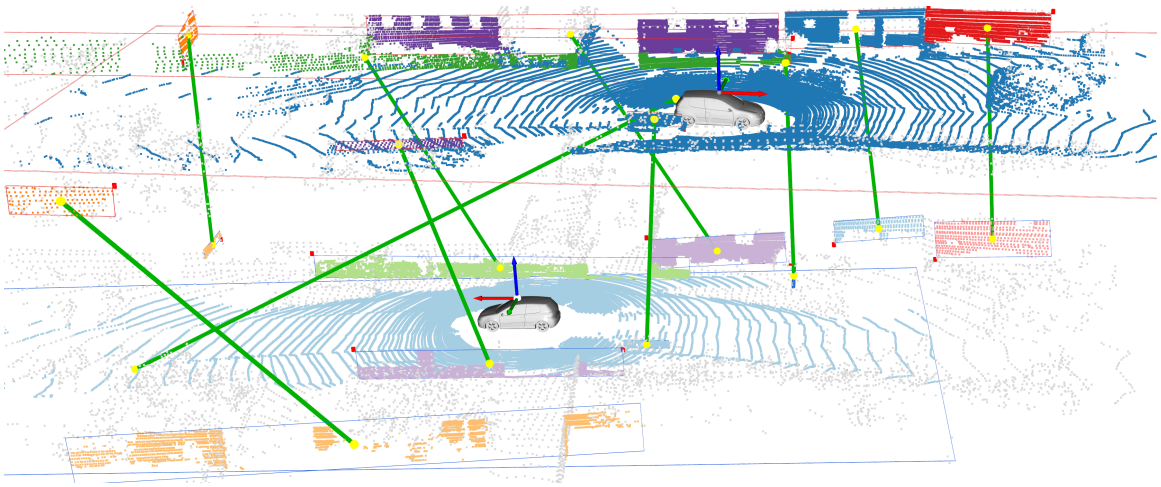


Figure 4-1. Successful matching and alignment of line and plane landmarks from two LiDAR scans 180 deg and 14 m apart, without an initial guess. Sensor origins are denoted by the coordinate axes on top of the cars and scans are offset in the z direction for visualization. Poles and planar patches extracted from each LiDAR scan are represented as 3D affine Grassmannian elements. Using the associated Riemannian metric allows for the evaluation of geometric consistency between landmark pairs. Correspondences (shown with green lines connecting landmark centroids) are identified using our GraffMatch algorithm and then used to estimate the rotation and translation between the two sensors, yielding an alignment error of 0.8 deg and 12 cm.

2. a least squares estimator for rigid transformation using lines and planes instead of points, leading to a more accurate estimate for rotation and translation;

3. experimental evaluation in the context of loop closure, using challenging scan pairs of LiDAR datasets, showing superior recall and accuracy over the state-of-the-art.

This is the first work using the affine Grassmannian manifold for data association, which provides a unifying and principled framework for associating points, lines, planes (or higher dimensional linear objects) in spatial and geometric perception problems encountered in robotics.

4.2 Background

We briefly introduce the Grassmannian manifold, but a more comprehensive introduction is provided in [184]. The Grassmannian is the space of k-dimensional subspaces of $\mathbb{R}^n$, denoted $\mathrm{Gr}(k, n)$. For example, $\mathrm{Gr}(1, 3)$ represent 3D lines containing the origin.

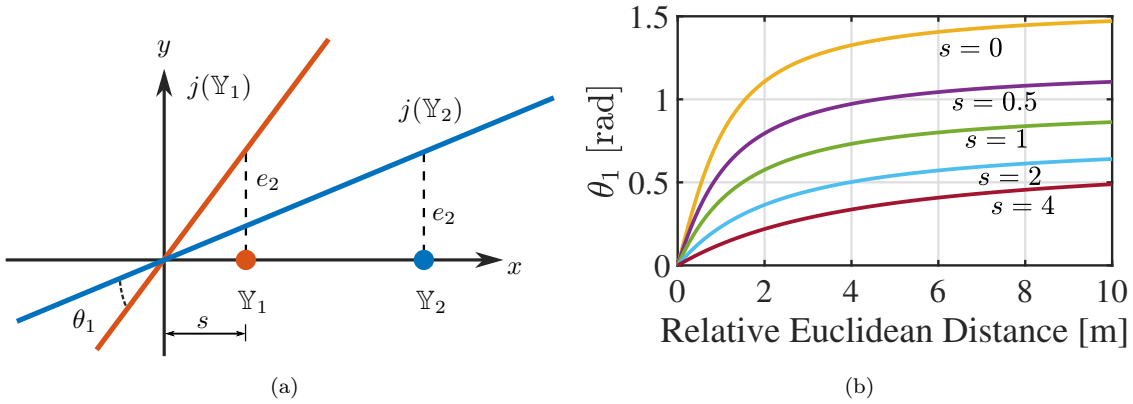


Figure 4-2. (a) Example of points in $\mathrm{Graff}(0,1)$ being embedded as lines in $\mathrm{Gr}(1,2)$. The principal angle between these two linear subspaces is θ_1. (b) When applied directly, d_{Graff} is not invariant to global translation s.

An element $\mathbb{A} \in \mathrm{Gr}(k,n)$ is represented by an orthonormal matrix $A \in \mathbb{R}^{n\times k}$ whose columns form an orthonormal basis of $\mathbb{A}$. Note that the choice of A is not unique. The geodesic distance between two subspaces $\mathbb{A}_1 \in \mathrm{Gr}(k_1,n)$ and $\mathbb{A}_2 \in \mathrm{Gr}(k_2,n)$ is

$$d_{\mathrm{Gr}}(\mathbb{A}_1,\mathbb{A}_2) = \left(\sum_{i=1}^{\min(k_1,k_2)} \theta_i^2\right)^{1/2} \tag{4.1}$$

where θ_i are known as the principal angles [185]. These angles can be computed via the singular value decomposition (SVD) of the corresponding orthonormal matrices of $\mathbb{A}_1$ and $\mathbb{A}_2$,

$$A_1^\top A_2 = U\,\mathrm{diag}(\cos\theta_1,\ldots,\cos\theta_{\min(k_1,k_2)})\,V^\top. \tag{4.2}$$

We are specifically interested in affine subspaces of $\mathbb{R}^3$, e.g., lines and planes that may be at some distance away from the origin. In analogy to $\mathrm{Gr}(k,n)$, the set of k-dimensional affine subspaces constitute a smooth manifold called the *affine Grassmannian* and denoted $\mathrm{Graff}(k,n)$ [186]. We write an element of this manifold as $\mathbb{Y} = \mathbb{A} + b \in \mathrm{Graff}(k,n)$ with affine coordinates $[A,b] \in \mathbb{R}^{n\times(k+1)}$, where $A \in \mathbb{R}^{n\times k}$ is an orthonormal matrix and $b \in \mathbb{R}^n$ is the displacement of $\mathbb{A}$ from the origin. We emphasize that $\mathrm{Graff}(k,n) \neq \mathrm{Gr}(k,n) \times \mathbb{R}^n$. Instead, an element $\mathbb{Y} \in \mathrm{Graff}(k,n)$ is

treated as a higher-order subspace via the embedding

$$\begin{aligned} j : \mathrm{Graff}(k, n) &\hookrightarrow \mathrm{Gr}(k+1, n+1), \\ \mathbb{A} + b &\mapsto \mathrm{span}(\mathbb{A} \cup \{b + e_{n+1}\}), \end{aligned} \tag{4.3}$$

where $e_{n+1} = (0, \ldots, 0, 1)^\top \in \mathbb{R}^{n+1}$ (see [186, Theorem 1]). Fig. 4-2a shows an example of two points $\mathbb{Y}_1$ and $\mathbb{Y}_2$ in $\mathbb{R}$ being embedded as different lines $j(\mathbb{Y}_1)$ and $j(\mathbb{Y}_2)$ in $\mathbb{R}^2$.

The Stiefel coordinates of $\mathbb{Y} \in \mathrm{Graff}(k, n)$,

$$Y = \begin{bmatrix} A & b_0/\sqrt{1 + \|b_0\|^2} \\ 0 & 1/\sqrt{1 + \|b_0\|^2} \end{bmatrix} \in \mathbb{R}^{(n+1)\times(k+1)}, \tag{4.4}$$

allow for the computation of distances between two affine subspaces using the Grassmannian metric,

$$d_{\mathrm{Graff}}(\mathbb{Y}_1, \mathbb{Y}_2) = d_{\mathrm{Gr}}(j(\mathbb{Y}_1), j(\mathbb{Y}_2)), \tag{4.5}$$

with principal angles computed via the SVD of $Y_1^\top Y_2$. The vector $b_0 \in \mathbb{R}^n$ is the orthogonal displacement of $\mathbb{A}$, which is the projection of b onto the left nullspace of A s.t. $A^\top b_0 = 0$.

For convenience, the line $\mathbb{Y}^\ell \in \mathrm{Graff}(1, 3)$ may also be represented in vector form as $\ell = [A; b] \in \mathbb{R}^6$, and a plane $\mathbb{Y}^\pi \in \mathrm{Graff}(2, 3)$ may be represented in Hesse normal form as $\pi = [n; d] \in \mathbb{R}^4$ where $n = \ker A^\top$ and $d = \|b_0\|$. Under a rigid transformation $T = (R, t) \in \mathrm{SE}(3)$, the transformation law of lines and planes can be written

$$\ell' = f_\ell(\ell, R, t) := \begin{bmatrix} RA & Rb + t \end{bmatrix}^\top \tag{4.6}$$

$$\pi' = f_\pi(\pi, R, t) := T^{-\top}\pi. \tag{4.7}$$

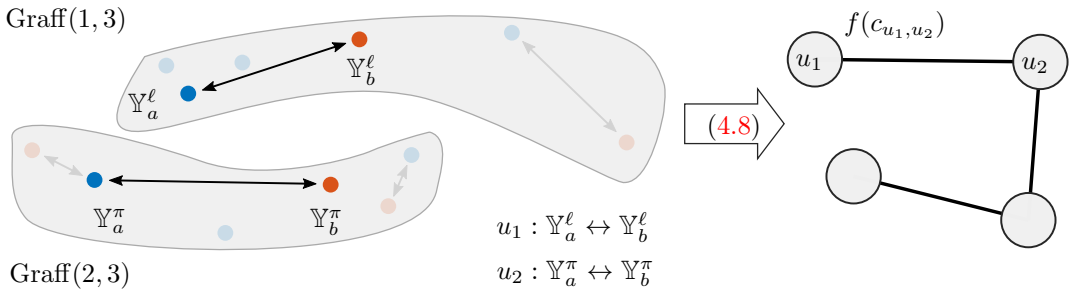


Figure 4-3. Construction of a consistency graph. Using d_{Graff}, the distance between a line and a plane in set $\mathcal{S}_i$ (•) is compared to the distance between the two corresponding landmarks in set $\mathcal{S}_j$ (•). The consistency of these two distances is evaluated using (4.8) and the edge (u_1, u_2) is so weighted.

4.3 Approach

Given a set $\mathcal{S}_i = \{\mathbb{Y}_1^\ell, \ldots, \mathbb{Y}_{l_i}^\ell, \mathbb{Y}_1^\pi, \ldots, \mathbb{Y}_{p_i}^\pi\}$ with l_i lines and p_i planes, we refer to landmark a as $s_{i,a} \in \mathcal{S}_i$. Our method is comprised of the following steps: (i) constructing a consistency graph based on pairwise landmark distances, (ii) identifying landmark correspondences via the densest fully-connected subgraph in the consistency graph, and (iii) estimating a rigid transformation based on correspondences.

4.3.1 Consistency Graph Construction

A consistency graph for two sets $\mathcal{S}_i, \mathcal{S}_j$ is an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ with potential landmark correspondences $s_{i,a} \leftrightarrow s_{j,b}$ as vertices, edges between consistent correspondences, and a weighting function $w : \mathcal{E} \to [0, 1]$ that evaluates the strength of consistency. A pair of correspondences $u_1, u_2 \in \mathcal{V}$ is consistent if the distance between the underlying objects $s_{i,a} \in \mathcal{S}_i, s_{j,b} \in \mathcal{S}_j$ satisfies

$$c_{u_1,u_2} = |d(s_{i,u_1^a},\, s_{i,u_2^a}) - d(s_{j,u_1^b},\, s_{j,u_2^b})| < \epsilon, \tag{4.8}$$

for some distance function d. Note that the two distances in (4.8) are between landmarks *internal* to sets $\mathcal{S}_i$ and $\mathcal{S}_j$, respectively. If a pair of correspondences are deemed consistent, the corresponding edge is attributed the weight $w(u_1, u_2) := f(c_{u_1,u_2})$, for some choice of $f : \mathbb{R}_+ \to [0, 1]$ that scores very consistent pairs close to 1. In this chapter, we choose $f(c) := \exp(-c^2/2\sigma^2)$ for simplicity, though other appropriate

functions could be used. Given a consistency graph, correspondences are selected that maximize consistency, further explained in Section 4.3.2.

The distance function d must be carefully chosen to ensure accuracy of graph-based data association. In particular, we desire (4.8) to hold when s_{j,u_1^b}, s_{j,u_2^b} are the transformed versions of s_{i,u_1^a}, s_{i,u_2^a}, respectively. This invariance property leads to subgraphs of the consistency graph that indicate a set of landmark matches.

Definition 4.3.1. A distance $d : X \times X \to \mathbb{R}$ is *invariant* if $d(x_1, x_2) = d(x_1', x_2')$, where $x_1', x_2' \in X$ are the transformation of $x_1, x_2 \in X$ under $T \in \mathrm{SE}(3)$, respectively.

We establish the invariance of the metric d_{Graff} to rotation and, under careful application, translation.

Proposition 4.3.1. *For elements* $\mathbb{Y}_1 \in \mathrm{Graff}(k_1, 3)$, $\mathbb{Y}_2 \in \mathrm{Graff}(k_2, 3)$ *with affine coordinates* $[A_1, b_1]$ *and* $[A_2, b_2]$, *the affine Grassmannian metric* d_{Graff} *is invariant if the affine components are first* shifted *to the origin, i.e., if both* b_1 *and* b_2 *are first translated by* $-b_1$.

Proof. Suppose $\mathbb{Y}_1, \mathbb{Y}_2$ are shifted by $-b_1$ such that $b_1 = 0$. This implies that the orthogonal displacement of $\mathbb{Y}_1$ is $b_{01} = 0$ and the inner product of the Stiefel coordinates of $\mathbb{Y}_1, \mathbb{Y}_2$ is

$$Y_1^\top Y_2 = \begin{bmatrix} A_1^\top A_2 & \frac{1}{\eta_2} A_1^\top b_{02} \\ \frac{1}{\eta_2} b_{01}^\top A_2 & \frac{1}{\eta_1 \eta_2}\left(b_{01}^\top b_{02} + 1\right) \end{bmatrix} = \begin{bmatrix} A_1^\top A_2 & \frac{1}{\eta_2} A_1^\top b_{02} \\ 0 & \frac{1}{\eta_1 \eta_2} \end{bmatrix},$$

with $\eta_i := \sqrt{\|b_{0i}\|^2 + 1}$. Recall that computing $d_{\mathrm{Graff}}(\mathbb{Y}_1, \mathbb{Y}_2)$ uses the SVD of $Y_1^\top Y_2$. Given $T = (R, t) \in \mathrm{SE}(3)$, let $\bar{\mathbb{Y}}_1, \bar{\mathbb{Y}}_2$ be the transformations of $\mathbb{Y}_1, \mathbb{Y}_2$, with affine coordinates

$$\mathbb{Y}_i : [A_i, b_i] \xrightarrow{T} \bar{\mathbb{Y}}_i : [RA_i, Rb_i + t].$$

Shifting $\bar{\mathbb{Y}}_1, \bar{\mathbb{Y}}_2$ by $-\bar{b}_1 := -(Rb_1 + t)$ leads to the affine coordinates $\bar{\mathbb{Y}}_1 : [RA_1, 0]$ and $\bar{\mathbb{Y}}_2 : [RA_2, Rb_2]$ so that

$$\bar{Y}_1^\top \bar{Y}_2 = \begin{bmatrix} A_1^\top A_2 & \frac{1}{\eta_2} A_1^\top b_{02} \\ 0 & \frac{1}{\eta_1 \eta_2} \end{bmatrix} = Y_1^\top Y2,$$

implying that $d_{\mathrm{Graff}}(\mathbb{Y}_1, \mathbb{Y}_2)$ is invariant to (R, t). □

The intuition of Proposition 4.3.1 can be understood from Fig. 4-2. As $\mathbb{Y}_1$ and $\mathbb{Y}_2$ are together translated further from the origin, the principal angle between $j(\mathbb{Y}_1)$ and $j(\mathbb{Y}_2)$ decreases to zero in the limit. However, the distance between the affine components of $\mathbb{Y}_1$ and $\mathbb{Y}_2$ remains the same, no matter the translation. By first shifting the affine components, we remove the dependence of the absolute translation in the computation of the principal angle, while maintaining the dependence on the *relative* translation between $\mathbb{Y}_1$ and $\mathbb{Y}_2$.

A remaining challenge is to address the insensitivity of d_{Graff} to the Euclidean distance between landmarks' affine components. The yellow curve ($s = 0$) in Fig. 4-2b represents the principal angle between $\mathbb{Y}_1, \mathbb{Y}_2 \in \text{Graff}(0, 1)$ after shifting them as per Proposition 4.3.1, as a function of the relative translation between $\mathbb{Y}_1$ and $\mathbb{Y}_2$. Observe that after a distance of approximately 2 m, the curve quickly asymptotes towards $\frac{\pi}{2}$. This nonlinearity leads to poor discrimination between pairs of correspondences whose internal landmarks are far apart in the Euclidean sense. To combat this when calculating pairwise affine Grassmannian distances, we first scale the affine component of each $\mathbb{Y}_i$ by a constant parameter ρ so that the affine coordinates of $\mathbb{Y}_i$ become $[A_i, b_i/\rho]$. The choice of ρ depends on the average Euclidean distance between landmarks in the environment and its effect is to bring principal angles into the linear regime. The selection of ρ is discussed further in Section 4.4.2.

With Proposition 4.3.1 and the scaling parameter ρ in hand, a consistency graph between landmarks in $\mathcal{S}_i$ and $\mathcal{S}_j$ can be constructed. We establish initial correspondences between each landmark in $\mathcal{S}_i$ with each landmark of $\mathcal{S}_j$ so long as the landmarks are of the same dimensions k (i.e., we do not allow lines to be associated to planes). Given additional information such as color, scan intensity, planar patch area, or pole radius, this initial set of correspondences could be refined, but would rely on accurately segmenting lines and planes across potentially wide baselines. While we restrict landmark correspondences to be of the same dimension, the machinery we have developed allows for computing the consistency of two correspondences whose internal pair of objects have differing dimension, thereby aiding in subgraph selection. Fig. 4-3 illustrates evaluating the consistency of a correspondence pair using the affine

Grassmannian.

4.3.2 Graph-based Global Data Association

Given a consistency graph, the task of matching objects from two scans is reduced to identifying the densest clique of consistent correspondences, formalized as the problem

$$\begin{aligned} &\underset{u\in\{0,1\}^m}{\text{maximize}} \quad \frac{u^\top M\, u}{u^\top u} \\ &\text{subject to} \quad u_i\, u_j = 0 \quad \text{if } M(i,j) = 0,\ \forall i, j, \end{aligned} \tag{4.9}$$

where $M \in [0,1]^{m\times m}$ is the weighted adjacency matrix (i.e., from w as defined in Section 4.3.1) with ones on the diagonal, and $u \in \{0,1\}^m$ indicates a consistent set of correspondences. Note that we choose to maximize the *density* of correspondences rather than the cardinality (i.e., maximum clique) as our previous work has found this objective to produce more accurate results (see Section 3.4). Problem (4.9) is NP-hard, therefore we solve a particular relaxation which yields high-accuracy solutions via our efficient CLIPPER algorithm (see Chapter 3).

4.3.3 Transformation Estimation

Given pairwise correspondences between landmarks in $\mathcal{S}_i$ and $\mathcal{S}_j$, consider finding the best rigid transformation to simultaneously align matched lines and planes by solving the optimization problem

$$\min_{\substack{R\in\mathrm{SO}(3),\\ t\in\mathbb{R}^3}} \sum_{i=1}^{p} \|\pi_i' - f_\pi(\pi_i, R, t)\|^2 + \sum_{i=1}^{l} \|\ell_i' - f_\ell(\ell_i, R, t)\|^2. \tag{4.10}$$

This problem can be solved in closed-form by first solving for the rotation via SVD, then solving for the translation via least squares, similar to Arun's method for point cloud registration [176]. The benefit of using the line and plane geometry directly, as opposed to a point parameterization, is twofold. First, it allows the use of the full information present in the infinite plane or line, i.e., distance from origin as well

as orientation. Second, it does not require assumptions about where the "centroid" of the plane or line is, which is undefined for infinite planes and lines and requires consistent segmentation of landmarks from point clouds. Together, these benefits lead to a more accurate rigid transformation estimate when aligning line and plane landmarks.

4.4 Results

We evaluate our method, called GraffMatch, using LiDAR scans from three datasets: KITTI [187] sequences 00, 02, 05, 08; KITTI-360 [188] sequences 00, 04, 06, 09; NCLT [189] sessions 2012-04-29, 2012-05-11, 2012-12-01. We include comparisons with CPMatch [156], BruteForceRMSE [55], ClusterMatch [72], and LiPMatch [160], all of which are used for geometric landmark registration in state-of-the-art pipelines. CPMatch uses nearest neighbor search on CP vectors and BruteForceRMSE exhaustively calculates the point-to-line/plane RMSE between each potential landmark match, followed by the selection of correspondences with low RMSE. ClusterMatch creates putative correspondences using pairwise landmark centroid distances followed by inlier validation, and LiPMatch uses an interpretation tree to find plane correspondences with similar geometric properties. We also include comparisons using our CLIPPER algorithm, using Euclidean distance of centroids and CP vectors to score consistency, both of which lack the required invariance for lines and planes. The algorithms are implemented[1] using Python and C++ and executed on an i9-7920X CPU with 64 GB RAM. The parameters used for GraffMatch (see (4.8)) are $\epsilon = 0.2$ and $\sigma = 0.05$. In our comparisons, we show that GraffMatch successfully matches more landmarks and is capable of producing more successful registrations.

4.4.1 Dataset Preparation

Motivated by place recognition and loop closure detection applications, we sample poses along the ground-truth trajectory of each dataset sequence with a stride of 2 m

[1] https://github.com/mit-acl/graffmatch

to create *places* (i.e., a pose with the LiDAR scan at that pose), following [181]. For each place p, true *revisits* are found by identifying any previously visited place p' such that the path length between p and p' is greater than 50 m and the Euclidean distance between the places is less than 20 m. Ground-truth landmark matches between revisited places are generated by aligning the landmarks using ground truth and solving a linear-sum assignment problem based on d_{Graff} distances. Valid place pairs are selected as *loop closures* from revisits having more than 4 true landmark matches and having registration error less than 5 deg and 1 m using the true landmark matches. This setup provides a scene matching dataset similar to 3DMatch [51], but having line and plane landmarks instead of point features.

Lines are detected in point clouds by first selecting pole-like points found by clustering the LiDAR range image, as in [190]. For KITTI, pole-like points are selected using semantic information from SemanticKITTI [191]. These points are clustered using DBSCAN [192] implemented in Open3D [193] and PCA is used to estimate lines from pole-like clusters. Planar patches are extracted from the LiDAR scan using our own implementation[2] of [194]. Because planar patches are bounded, there may be multiple planar patches that correspond to the same infinite plane, so we merge planes that are similar. The statistics of correct line and plane landmark matches in the dataset are shown in Fig. 4-4, separated into three cases according to input inlier ratio (IIR), which is defined as the number of correct matches out of the number of possible matches and indicates the difficulty of each place pair. Cases 1, 2, and 3 consist of place pairs with IIRs of 0.05+, 0.03 to 0.05, and 0 to 0.03, respectively. These IIR ranges are chosen to expose the relationship between IIR and sensor baseline distance and to ensure enough place pairs exist in each case.

4.4.2 Selection of Scaling Parameter

The scaling parameter ρ (see Section 4.3.1) is chosen so that the pairwise affine Grassmannian distance lies in the linear regime and is therefore more sensitive when scoring consistencies. The Velodyne HDL-64E used in KITTI has a range up to

[2] https://github.com/plusk01/pointcloud-plane-segmentation

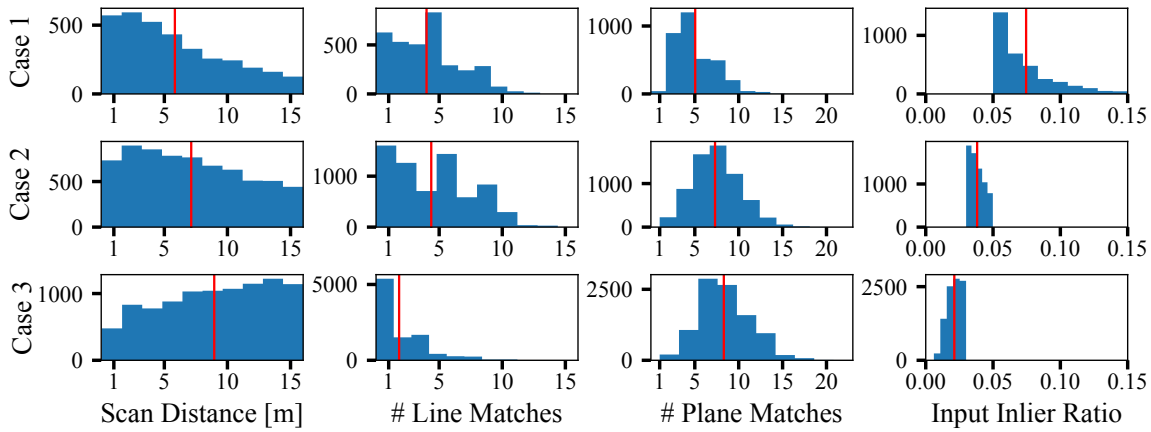


Figure 4-4. Statistics of the 20k loop closures in the test dataset, divided into three cases according to each loop closure's input inlier ratio. The distance between sensor scans and input inlier ratio are correlated, with the most difficult case (Case 3) having longer sensor baselines. On average, there are more correct plane landmark matches than correct line landmark matches.

120 m, with an average point range in the KITTI dataset of approximately 80 m. Additionally, the average Euclidean distance between landmark centroids in KITTI is 26 ± 16 m, as shown in Fig. 4-5. Therefore, we select $\rho = 40$ so that relative Euclidean distances of 80 m will be scaled to 2 m, which is at the end of the linear regime (see Fig. 4-2b). However, GraffMatch is not extremely sensitive to this choice and yields similar matching results for a range of ρ values while holding other parameters constant, as shown in Fig. 4-5.

4.4.3 Evaluation Metrics

We evaluate both landmark matching ability and registration quality. Correctness of landmark matching is evaluated via landmark-match recall (LMR) [52, 195], which measures the percentage of place pairs that can be registered with high confidence given landmark matches. LMR is defined as

$$\text{LMR} = \frac{1}{N} \sum_{s=1}^{N} \mathbb{1} \left(\left[\frac{1}{|\mathcal{C}_s|} \sum_{(i,j) \in \mathcal{C}_s} \mathbb{1} \left(d_{ij} < \tau_d \right) \right] > \tau_{\text{OIR}} \right), \tag{4.11}$$

where N is the number of place pairs, $\mathcal{C}_s$ is a set of landmark correspondences between place pair s, and d_{ij} is the d_{Graff} distance between landmark matches after registering

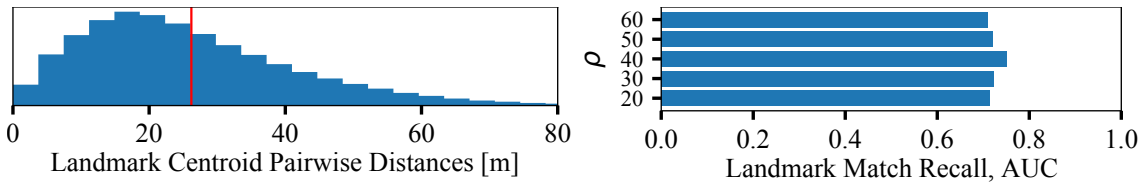


Figure 4-5. (left) Pairwise distances of landmark centroids in KITTI. The mean is 26 ± 16 m. Using this data, we choose the scaling parameter as $\rho = 40$. (right) GraffMatch is not extremely sensitive to this choice of scaling. Other values of ρ yield similar matching results in KITTI, indicated by the AUC.

landmarks using ground truth. The area under the LMR curve (AUC) using $\tau_d = 6\,\text{deg}$, $\tau_{\text{OIR}} = 0.6$ is used as a summary measure, with an AUC of 1 being ideal. The inlier distance threshold τ_d controls how close two landmarks must be (after registering the landmarks using the ground truth) to be considered an inlier correspondence. The output inlier ratio (OIR) measures the precision of an algorithm's correspondence set $\mathcal{C}_s$ and the threshold τ_{OIR} is used to examine the percentage of place pairs an algorithm can recover with at least the specified OIR. Point-based registration works have evaluated performance with an OIR threshold as low as 0.05 [52, 195], arguing that RANSAC can be effective even with only a 5% inlier ratio, even though many iterations would be required [195]. In contrast, we set $\tau_{\text{OIR}} \gg 0.05$ to emphasize the robustness of GraffMatch and we do not use RANSAC for inlier refinement.

Registration quality is evaluated via rotation, translation error and registration recall. Error is calculated with respect to the ground truth transformation $(\boldsymbol{R}^\star, \boldsymbol{t}^\star)$ as $\arccos((\text{Tr}(\hat{\boldsymbol{R}}^\top \boldsymbol{R}^\star) - 1)/2)$ and $\|\hat{\boldsymbol{t}} - \boldsymbol{t}^\star\|$. Registration recall is the percentage of successfully registered place pairs (i.e., loop closures). A successful registration has an estimation error within 5 deg and 1 m to the ground truth transformation.

4.4.4 Landmark Matching and Registration Results

Data association is attempted on each place pair, after which the landmark matches are used to estimate the relative rotation and translation. Fig. 4-6 plots algorithms' LMR curves for the NCLT dataset. GraffMatch is able to match more than 80% of place pairs with an OIR of 0.6 or more, while other methods are only able to match less than 40% of place pairs for the same inlier regime. For correspondence sets having

Table 4.1. Results for each dataset, divided into three cases based on the feature inlier ratio. We compare each algorithm on recall rate and landmark-match recall AUC and the highest for each dataset case is bolded. The average rotation and translation errors of the successful registrations are also listed.

			CPMatch [156]				BruteForceRMSE [55]				ClusterMatch [72]				LiPMatch [160]				CLIPPER-$\mathbb{R}^n$				CLIPPER-CP				GraffMatch			
		N	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]	Recall [%]	LMR AUC	R_e [deg]	t_e [cm]
KITTI	Case 1	1392	24	0.47	1.1	21	9	0.33	1.2	62	35	0.34	1.7	18	16	0.30	1.1	19	62	0.70	1.5	24	42	0.56	1.3	21	**81**	**0.91**	1.1	20
	Case 2	4560	19	0.53	1.2	23	3	0.34	1.3	69	16	0.18	1.5	17	15	0.32	1.2	22	46	0.55	1.5	27	22	0.38	1.4	25	**52**	**0.78**	1.1	19
	Case 3	2329	7	0.42	1.5	29	<1	0.23	2.1	66	6	0.10	1.5	21	6	0.23	1.7	30	20	0.30	1.7	31	7	0.17	1.7	33	**21**	**0.59**	1.2	24
KITTI-360	Case 1	182	17	0.51	1.6	29	3	0.34	1.5	28	21	0.28	1.4	26	8	0.23	1.8	35	16	0.32	1.5	26	10	0.35	1.7	32	**53**	**0.85**	1.6	32
	Case 2	1555	10	0.43	1.6	27	1	0.26	2.4	20	8	0.14	1.3	25	5	0.19	1.7	32	12	0.26	2.0	31	6	0.28	1.7	30	**41**	**0.74**	1.8	36
	Case 3	7143	4	0.27	1.8	28	<1	0.14	2.6	20	2	0.06	1.7	31	2	0.11	2.0	36	6	0.14	2.2	36	1	0.14	2.1	34	**21**	**0.55**	1.9	42
NCLT	Case 1	1854	6	0.33	2.0	30	1	0.23	2.6	81	21	0.36	2.9	30	4	0.16	2.5	41	25	0.42	2.7	35	14	0.39	2.7	35	**60**	**0.91**	2.7	35
	Case 2	676	5	0.38	2.0	25	1	0.20	2.8	89	7	0.17	2.9	34	1	0.09	1.9	44	20	0.36	2.9	39	9	0.32	2.8	36	**43**	**0.78**	2.8	38
	Case 3	150	4	0.32	2.5	29	<1	0.14	2.6	76	2	0.15	3.2	19	0	0.07	–	–	12	0.31	3.2	43	2	0.23	2.4	42	**16**	**0.62**	2.9	36

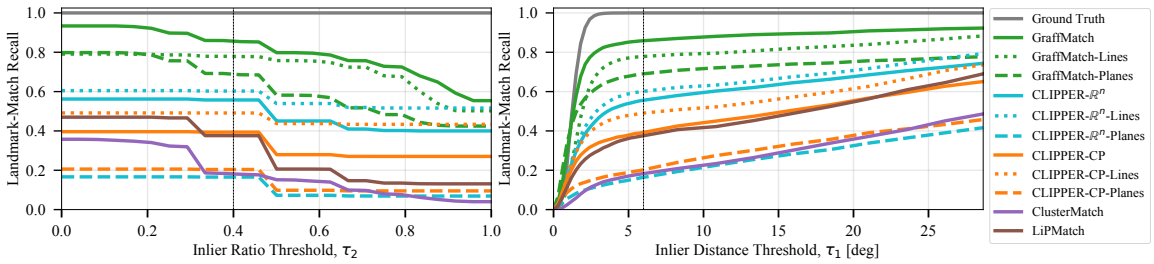


Figure 4-6. Landmark-match recall (LMR) curves for the NCLT datasets. (left) LMR varied over τ_{OIR}, indicating for each algorithm the percentage of place pairs having selected correspondences with at least the given OIR. (right) The maximum distance a pair of ground-truth registered landmarks can be while considered an inlier match.

such a high inlier ratio, additional methods like RANSAC could be used to further refine the correspondence set if desired. However, we show that even without this extra step, GraffMatch is able to successfully select the most correct correspondences. Fig. 4-6 also includes line-only and plane-only variants of GraffMatch to highlight the value of creating a consistency graph utilizing both landmark types. In the NCLT dataset, there are fewer detected poles than planes, leading to a gap between the line-only and plane-only variants; however, using both landmarks results in a greater number of place pairs having a higher OIR. In the right of Fig. 4-6, we see how the LMR curve changes with the inlier distance threshold τ_d. By using d_{Graff} for these geometric landmarks, GraffMatch is able to achieve an LMR curve most similar to the ground truth. Due to space limitations, the LMR curves for each dataset are not shown, but the AUC for each dataset and case is listed in Table 4.1. In all datasets and cases, GraffMatch successfully matches more landmarks and has higher registration recall than other methods. As expected, the local methods CPMatch and BruteForceRMSE perform worse than global methods because of wide sensor baselines present in the datasets. However, with the exception of GraffMatch, the global methods leverage landmark properties that are view dependent. ClusterMatch and CLIPPER-$\mathbb{R}^n$ use landmark centroids, which are not well-defined for infinite geometries and may shift depending on the detection. Likewise, LiPMatch uses centroids as well as properties such as the area and the extent of planar patches. CLIPPER-CP uses CP vectors, which are not invariant to translation and cannot be shifted as in Proposition 4.3.1 because the CP vector is undefined for landmarks at the origin.

These results underscore the importance of using view-independent geometric representations for lines and planes, e.g., the affine Grassmannian mainfold used in the GraffMatch framework.

Fig. 4-7 shows the alignment error of all attempted place pair registrations from KITTI as a grid of density heatmaps, where columns correspond to algorithms and rows (from top to bottom) correspond to Cases 1, 2, and 3. The first column shows the alignment error using ground truth landmark matches with respect to ground truth alignment, indicating the best achievable landmark-based registration without a point-based refinement step (e.g., using ICP). GraffMatch is the only data association method that consistently scores in the low-translation, low-rotation error regime.

Runtime statistics are shown in Fig. 4-8. Compared to CLIPPER-$\mathbb{R}^n$, GraffMatch incurs an additional 50 ms on average due to the calculation of d_{Graff}, but is still capable of real-time at 10 Hz LiDAR rate. GraffMatch runtime could be reduced by encoding prior knowledge in the putative associations, e.g., ground planes should be matched, or large planar patches are not likely to be matched to small patches.

4.4.5 Automatic LiDAR-LiDAR and Camera-Depth Calibration

High-quality extrinsic calibration is a crucial prerequisite for many autonomous systems. A cumbersome step in many calibration pipelines is to manually perform data association, especially when multiple modalities like camera and LiDAR are present. Using GraffMatch, geometric landmarks extracted from each modality can be matched without requiring an initial registration guess. Fig. 4-9 shows two instances of extrinsic calibration using GraffMatch to globally match planes. In Fig 4-9a, many chess boards are held in the field-of-view of the LiDARs and planes are extracted from the two point clouds (red and blue). GraffMatch correctly matches 10 planes in 75 ms with calibration error of 1.1 deg and 8 cm. In Fig 4-9b, an example of multi-modal calibration using is presented, where 3D planes are extracted from chess boards in the image frame and matched to planes extracted from the depth sensor. GraffMatch

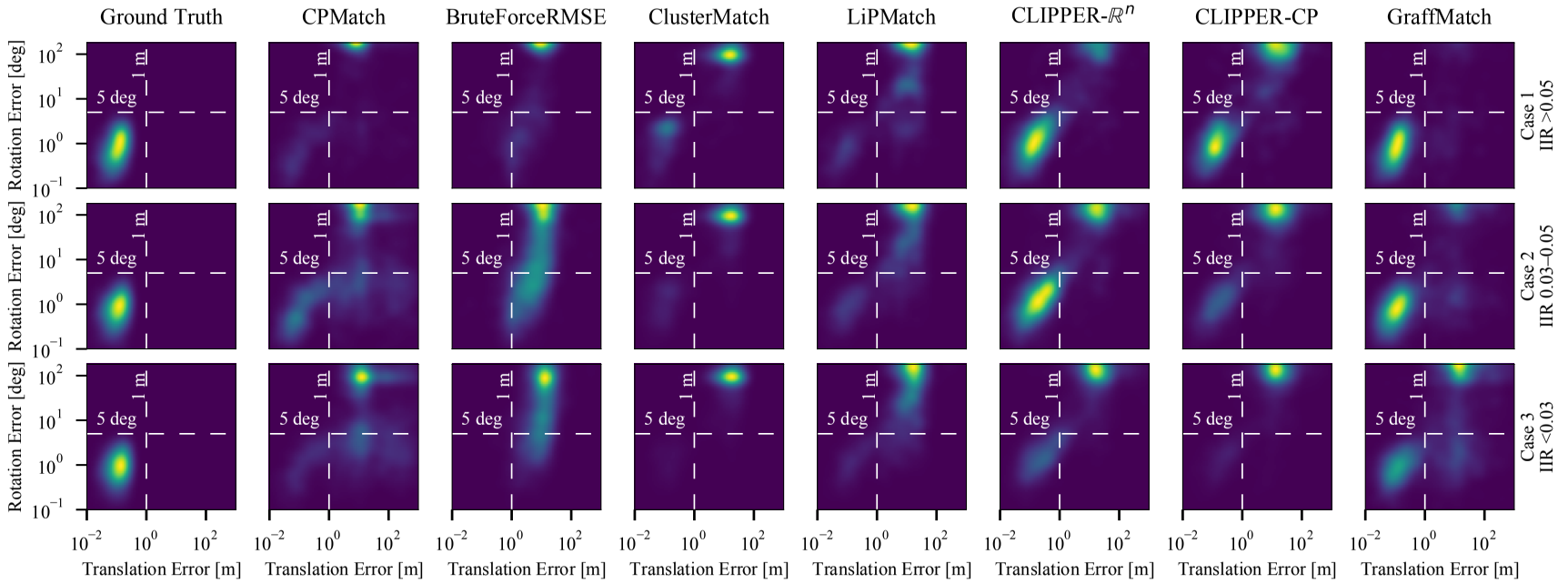


Figure 4-7. Alignment error for place pairs, visualized as likelihood-normalized density plots. Columns correspond to algorithms. The first column shows the best alignment error achievable when registering landmarks using the true landmark matches. From top to bottom, each row corresponds to Cases 1, 2, and 3, with Case 1 being easiest and Case 3 being hardest. In each case, GraffMatch produces the greatest number of registrations with low alignment error.

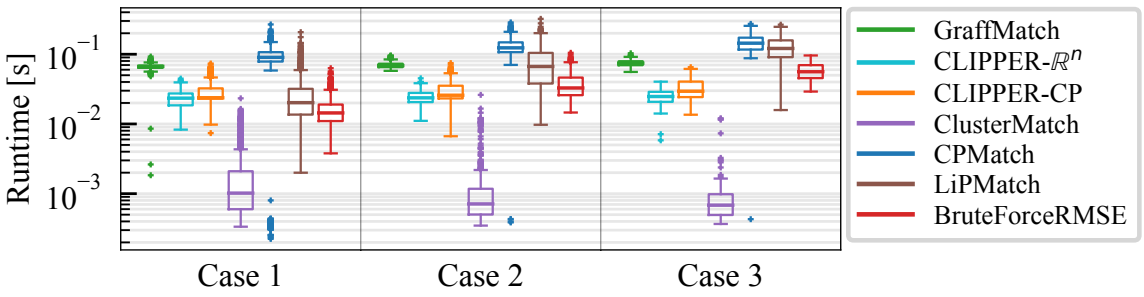


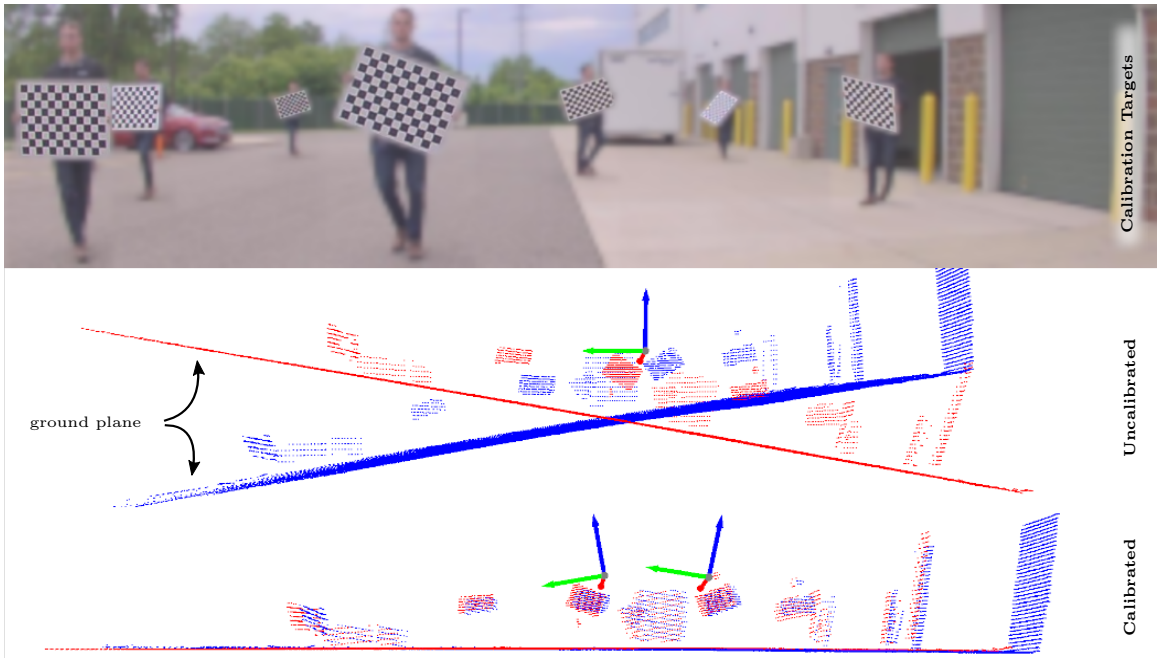
Figure 4-8. Runtime statistics. GraffMatch is capable of running in real-time at the typical 10 Hz LiDAR rate. The runtime of GraffMatch could be reduced using landmark descriptors to decrease the number of putative associations.

correctly matches 7 planes in 23 ms resulting in a calibration error of 0.9 deg and 5 cm.

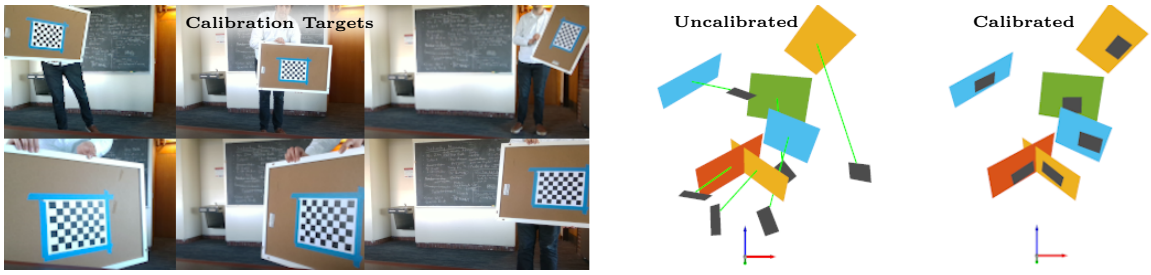
4.5 Summary

We presented GraffMatch, a global method for matching 3D lines and planes from two landmark sets and subsequently estimating the relative rotation and translation. The main novelty of the GraffMatch algorithm is in the representation of lines and planes as elements of the affine Grassmannian manifold. By representing these geometric landmarks in this natural way (e.g., as affine subspaces), it is possible to leverage the Grassmannian metric to calculate the distance between two landmarks. We prove that the distance between two affine Grassmannian elements is invariant to both rotation and translation if a shift operation is performed before applying the metric. This invariance property enables the use of efficient and robust graph-theoretic data association methods.

No initial alignment guess is required for GraffMatch, allowing registration in settings where landmark sets have a large displacement or relative rotation due to observing the landmarks from two different locations. We evaluate GraffMatch and compare against state-of-the-art geometric landmark matching algorithms by generating a challenging dataset of LiDAR scan pairs with displacements up to 16 m and 180 deg, motivated by wide-baseline scan registration for loop closure. In this evaluation, GraffMatch is able to correctly match the most landmarks and achieve



(a)



(b)

Figure 4-9. Automatic extrinsic calibration using GraffMatch. (a) 3D planes are extracted from two Ouster LiDAR scans, illustrated with red and blue point clouds. Plane-to-plane correspondences are identified, allowing the sensors to be calibrated without an initial guess. (b) 3D plane detections of calibration targets are extracted from depth sensor and from intrinsically-calibrated RGB sensor using PnP. RGB and depth data collected from Intel D435i are arbitrarily transformed to simulate uncalibrated sensors.

the highest number of successful registrations. We also show the applicability of GraffMatch for multi-modal extrinsic sensor calibration.

Currently, GraffMatch selects landmark correspondences from an initial set of correspondences where any landmark could be matched to any landmark of the same type. This can lead to a large number of initial associations to process, increasing runtime and potentially introducing symmetries (i.e., multiple internal pairs of landmarks with the same pairwise distance). A remedy would be to reduce the number of initial associations by developing landmark descriptors from which to generate putative matches. Future research also includes estimating lines and planes directly via subspace tracking methods and using manifold-based optimization techniques to perform online bundle adjustment of affine Grassmannian landmarks within a SLAM framework.

Chapter 5

Multiway Synchronization of Pairwise Correspondences

5.1 Introduction

Data association across *multiple* views, known as multiview or multiway matching [46], 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. 5-1). Examples include feature matching across multiple frames [46, 56, 57], and associating landmarks across multiple maps for map fusion in single/multi-robot simultaneous localization and mapping (SLAM) [20].

The traditional approach treats the multi-view 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 [103]. Such techniques, however, cannot leverage the redundancy in the observations and, furthermore, often result in *non-transitive* (a.k.a., cycle *inconsistent*) associations; see Fig. 5-1. 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

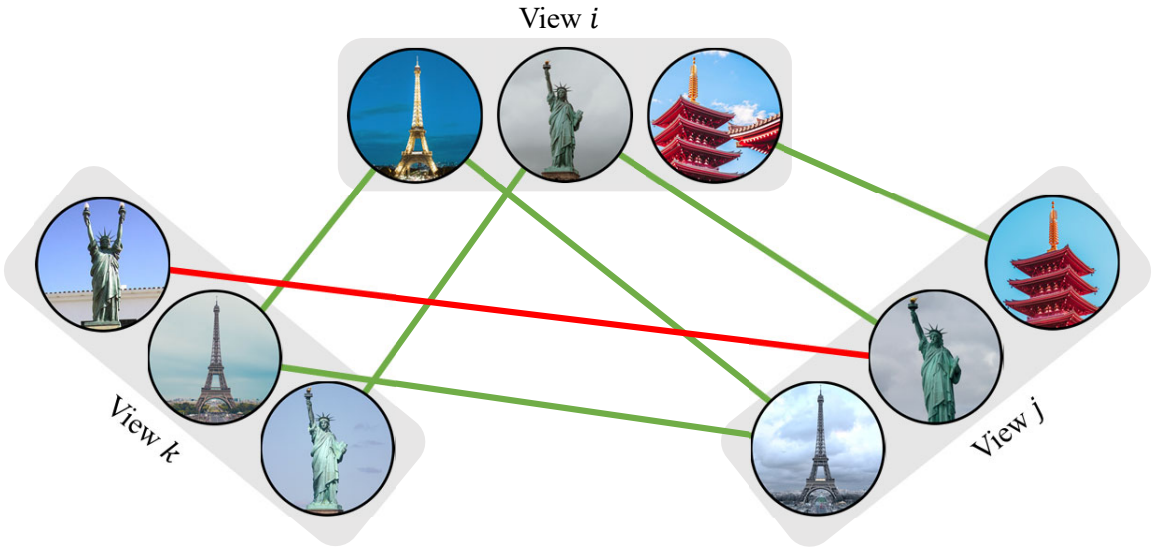


Figure 5-1. An illustrative example of cycle consistency for the association of images observed in views i, j, k. Associations of "Eiffel tower" are cycle consistent. On the other hand, the "statue of liberty" associations are inconsistent since the images matched between views i and j and views i and k are not matched between views j and k (i.e., violation of transitivity).

relation on the set of observations, which is *necessary* for global fusion in the so-called clique-centric applications such as map merging (Section 5.3.6).

Synchronizing pairwise associations is a combinatorial optimization problem with an exponentially large search space. This problem has been extensively studied in recent years (see [46,56,57,61] 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 that the items observed in each view are unique, and thus must not be associated with each other).

This chapter 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.

The contributions of the work in this chapter provide new insights into the 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. 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 5.3.1).

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 [196] used in the spectral graph clustering literature (Remark 5.3.3).

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 5.4.1).

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 5.3.6). Finally, we provide extensive numerical experiments on both synthetic and real datasets in the context of feature matching and map fusion (Sections 5.4.1 and 5.4.2). 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.

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

Outline

The organization of this chapter is as follows. The notation and definitions are introduced in Section 5.2.1, followed by the problem formulation in Section 5.3. The CLEAR algorithm is presented in Section 5.3.3, followed by a numerical example in Section 5.3.4. The theoretical justifications behind the algorithm are discussed in Section 5.3.5 with proofs presented in the Appendix. Application domains for the CLEAR algorithm are discussed in Section 5.3.6. CLEAR is benchmarked against the state-of-the-art algorithms using synthetic data in Section 5.4.1. Finally, experimental evaluations of CLEAR on real-world datasets are presented in Section 5.4.2.

5.2 Background

5.2.1 Notation and Definitions

We denote the set of natural numbers by $\mathbb{N}$, integers by $\mathbb{Z}$, $\mathbb{N}_0 := \{0\} \cup \mathbb{N}$, and define $\mathbb{N}_n := \{1, 2, \cdots, n\}$. 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 $\langle A, B \rangle := \text{tr}(A^\top B)$, where A and B are matrices of the same size. Finally, $\|\cdot\|$ denotes the (induced) 2-norm. Table 5.1 lists the key variables used throughout this chapter.

5.2.2 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

$$P := \begin{bmatrix} I & P_2^1 & \cdots & P_n^1 \\ P_1^2 & I & \cdots & P_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ P_1^n & P_2^n & \cdots & I \end{bmatrix} \in \mathbb{R}^{l \times l}, \tag{5.1}$$

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

5.2.3 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 $d_i := \sum_{j=1}^{l} a_{ij}$, 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 $C := D + I$, 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 $L := D - A$. 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.

5.3 Approach

Simply put, the objective of this chapter 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.

Table 5.1. Summary of important nomenclature used throughout this chapter.

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; $l := \sum_i m_i$
$\tilde{}$	-	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 (5.1)
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 $c_i := \sum_j (P)_{ij}$; $C = D + I$
L	$\mathbb{Z}^{l \times l}$	Laplacian matrix of $\mathcal{G}$; $L := D - A = C - P$
L_{nrm}	$\mathbb{R}^{l \times l}$	Normalized Laplacian matrix; $L_{\text{nrm}} := C^{-\frac{1}{2}} L C^{-\frac{1}{2}}$
P_{nrm}	$\mathbb{R}^{l \times l}$	Normalized association matrix; $P_{\text{nrm}} := C^{-\frac{1}{2}} P C^{-\frac{1}{2}}$
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 (5.3)
U	$\mathbb{R}^{l \times m}$	Normalized aggregate lifting permutation; $U := C^{-\frac{1}{2}} V$; 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

5.3.1 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. 5-2. 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. 5-3.

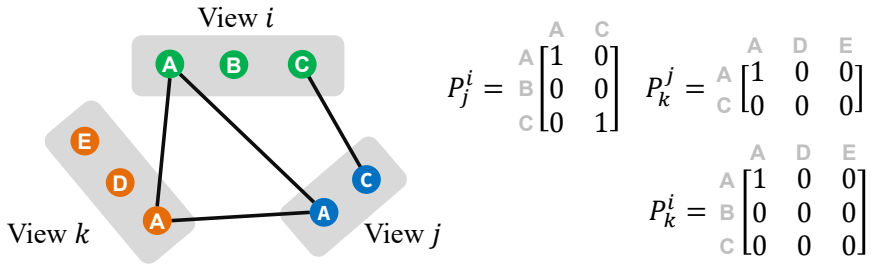


Figure 5-2. Association of items labeled as A, B, C, D, E observed at three views.

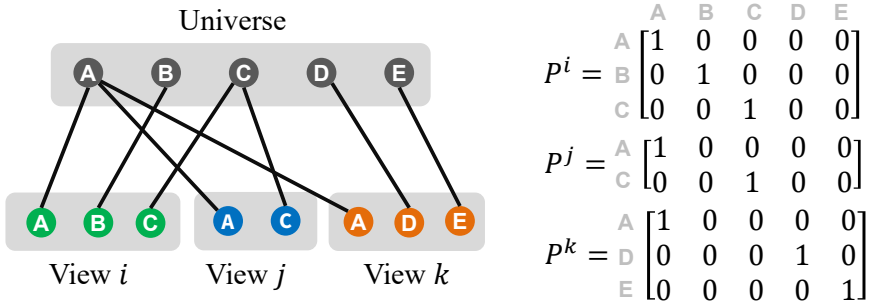


Figure 5-3. Lifting permutation matrices associating observations at views i, j, k to the universe, which consists of items labeled as A, B, C, D, E.

Definition 5.3.1 (Cycle consistency)**.** Pairwise associations P_j^i are *cycle consistent* if and only if there exist lifting permutations $P^i \in \mathbb{P}^L$ such that

$$P_j^i = P^i \, P^{j\top}, \qquad \forall i, j \in \mathbb{N}_n. \tag{5.2}$$

The cycle consistency condition (5.2) can be presented more concisely as $P = V\,V^\top$, where P is the aggregate association matrix defined in (5.1), and

$$V := \begin{bmatrix} P^{1\top} & P^{2\top} & \dots & P^{n\top} \end{bmatrix}^\top \in \{0,1\}^{l \times m}, \tag{5.3}$$

where $l := \sum_i m_i$. 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 chapter, we use the notation $\tilde{\cdot}$ 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 (5.1), 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., $\tilde{c}_i := \sum_{j=1}^{l} (\tilde{P})_{ij}$. Using definitions above, we now formulate the main problem.

Problem 5.3.1. Given noisy associations $\tilde{P}_j^i$, find cycle consistent associations P_j^i that solve the program

$$\begin{aligned} \underset{P_j^i \in \mathbb{P}}{\text{maximize}} \quad & \langle P_{\text{nrm}}, \tilde{P}_{\text{nrm}} \rangle \\ \text{subject to} \quad & P = V\,V^\top, \end{aligned} \tag{5.4}$$

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

In Problem 5.3.1, diagonal matrices C and $\tilde{C}$ are used to normalize the aggregate association matrices. The justification behind this normalization will be explained in Remark 5.3.1 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, 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).

5.3.2 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. 5-2). 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 $g(v_j) := i$ 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 (5.4) 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 5.3.1. *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 5.3.1 is given in [57, 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 j-th 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 multi-way association problem is as follows.

Problem 5.3.2. Given the noisy association graph $\tilde{\mathcal{G}}$, find undirected graph $\mathcal{G}$ with adjacency matrix A that solves

$$\begin{aligned} \underset{A}{\text{maximize}} \quad & \langle A_{\text{nrm}},\, \tilde{A}_{\text{nrm}} \rangle \\ \text{subject to} \quad & \mathcal{G} \text{ consists of clusters } \mathcal{A}_1, \ldots, \mathcal{A}_m \\ & g(v_i) \neq g(v_j), \quad \forall v_i, v_j \in \mathcal{A}_k \end{aligned} \tag{5.5}$$

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

Note that Problems 5.3.1 and 5.3.2 are equivalent. As elaborated above, the indices of the vertices belonging to clusters $\mathcal{A}_1, \ldots, \mathcal{A}_m$ uniquely determine V in

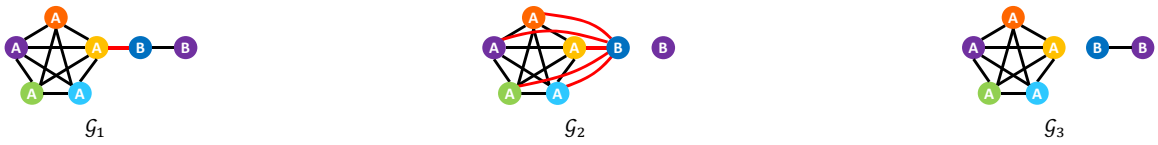


Figure 5-4. (Best viewed in color) Graph $\mathcal{G}_1$ indicates the association of two items labeled as A, B, in six views identified by colors ●, ●, ●, ●, ●, ●. The incorrect association, which connects A and B, is indicated by the red edge. If in (5.4) the unnormalized objective $\langle P, \tilde{P} \rangle$ 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).

Problem 5.3.1. Further, since $A = P - I$, both objective functions have the same optimizer. In (5.5), 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 5.3.1. *The normalized objective function in* (5.4) *is a key distinction from several state-of-the-art methods [46, 60, 144] 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. 5-4, 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. 5-4 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.

5.3.3 The 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 will be discussed in Section 5.3.5. The pseudocode of CLEAR is given in Algorithm 2, 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 $\tilde{P}_j^i$. Define the normalized Laplacian of $\tilde{\mathcal{G}}$ as

$$\tilde{L}_{\text{nrm}} := \tilde{C}^{-\frac{1}{2}}\,\tilde{L}\,\tilde{C}^{-\frac{1}{2}}, \tag{5.6}$$

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

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 [197]. 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 $(\tilde{L}_{\text{nrm}} + \tilde{L}_{\text{nrm}}^{\top})/2$ should be used instead in the eigendecomposition (the skew-symmetric component does not contribute to the optimal answer; see Remark 5.3.2). Note that the symmetry implies that all eigenvalues and eigenvectors are real.

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

$$\hat{m} := \max\{\tilde{m},\, m_1,\, m_2, \ldots, m_n\}, \tag{5.7}$$

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

$$\tilde{m} := \left|\{\lambda \in \text{eig}(\tilde{L}_{\text{nrm}}) \;:\; \lambda < 0.5\}\right|, \tag{5.8}$$

Algorithm 2 CLEAR (pseudocode)

Input noisy pairwise associations $\tilde{P}_j^i$
Output cycle consistent associations P_j^i
• **Step 1:** Compute $\tilde{L}_{\text{nrm}}$ from (5.6) and find its eigendecomposition.
• **Step 2:** Estimate size of universe $\hat{m}$ from (5.7).
• **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 $P_j^i \leftarrow P^i P^{j\top}$.

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.

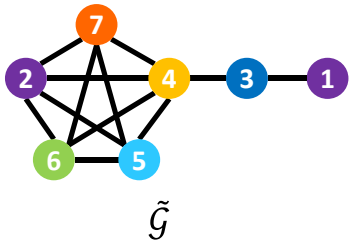
• **Step 4**: Normalize rows of U to have unit norm, i.e., the i-th row of U, denoted by u_i, is replaced by $u_i/\|u_i\|$. Choose the $\hat{m}$ most orthogonal rows as *pivots*.

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 k-th pivot, u'_{k+1} is selected such that $\sum_{i=1}^{k} |\langle u'_i, u'_{k+1}\rangle|$ is minimized.

For each view i, define matrix $F^i \in \mathbb{R}^{m_i\times m}$ by $(F^i)_{jk} := \|u_j - u'_k\|^2$, 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 [103] 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 $P_j^i = P^i P^{j\top}$. From Definition 5.3.1, pairwise associations are cycle consistent by construction.

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



$\tilde{\mathcal{G}}$

Figure 5-5. The 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.

5.3.4 Numerical Example

An example is presented to illustrate the steps of the CLEAR algorithm and show how pivot rows are chosen.

Example 5.3.1. In this example, we use the CLEAR algorithm to recover cycle-consistent associations from the (noisy) association graph $\tilde{\mathcal{G}}$ shown in Fig. 5-5. Note that $\tilde{\mathcal{G}}$ is identical to $\mathcal{G}_1$ in Fig. 5-4, 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

$$\tilde{P} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{5.9}$$

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 (5.9), 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 $\{1.18,\ 1,\ 1,\ 1,\ 0.85,\ 0.17,\ 0\}$.

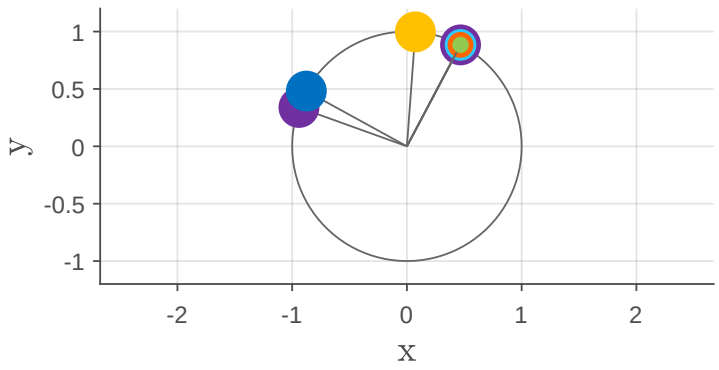


Figure 5-6. Embedding of rows of matrix U in Example 5.3.1.

• **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)

$$U = \begin{bmatrix} \text{-0.94} & 0.34 \\ 0.47 & 0.88 \\ \text{-0.88} & 0.48 \\ 0.07 & 0.99 \\ 0.47 & 0.88 \\ 0.47 & 0.88 \\ 0.47 & 0.88 \end{bmatrix}. \tag{5.10}$$

Fig. 5-6 depicts rows of (5.10) 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 $(F^i)_{jk} := \|u_j - u'_k\|^2$, where u_j are rows of U that correspond to view i and

u'_k are pivot rows we obtain

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

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

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

• **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. 5-4.

5.3.5 Theoretical Justifications

In this section, the insights and theoretical justifications behind the steps of the CLEAR algorithm are discussed.

The discrete and combinatorial nature of the multi-way 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.

Step 1: Reformulation

Before proceeding with obtaining an approximate solution, we reformulate (5.4) 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 5.3.2. *Problem 5.3.1 is equivalent to*

$$\underset{U \in \mathbb{U}}{\text{minimize}} \quad \operatorname{tr}\big(U^\top \tilde{L}_{\text{nrm}}\, U\big), \tag{5.11}$$

where $\mathbb{U} := \{U \,:\, U = C^{-\frac{1}{2}} V,\, V \in \mathbb{V}\}$, $\mathbb{V}$ *is defined as the set of all matrices of form* (5.3)*, and* $U^\top U = I$.

Proof. Consider the optimization problem (5.4). Because the trace operator is invariant under cyclic permutations[3], we obtain

$$\max_{P = V\,V^\top} \langle P_{\text{nrm}},\, \tilde{P}_{\text{nrm}} \rangle \tag{5.12a}$$

$$= \max_{P = V\,V^\top} \operatorname{tr}(P_{\text{nrm}}^\top \tilde{P}_{\text{nrm}}) \qquad \text{(defn of inner product } \langle \cdot, \cdot \rangle) \tag{5.12b}$$

$$= \max_{V \in \mathbb{V}} \operatorname{tr}(C^{-\frac{1}{2}}\, V\, V^\top C^{-\frac{1}{2}}\, \tilde{P}_{\text{nrm}}) \qquad \text{(since } P_{\text{nrm}} := C^{-\frac{1}{2}}\, P\, C^{-\frac{1}{2}} \text{ and } P = V\,V^\top) \tag{5.12c}$$

$$= \max_{V \in \mathbb{V}} \operatorname{tr}(V^\top C^{-\frac{1}{2}}\, \tilde{P}_{\text{nrm}}\, C^{-\frac{1}{2}}\, V). \qquad \text{(from cyclic permutation)} \tag{5.12d}$$

As discussed in Section 5.3.2, 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 (5.12d) 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 (5.12d) and defining $U := C^{-\frac{1}{2}}\, V$ we obtain

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

the equivalent program

$$\max_{V \in \mathbb{V}} \operatorname{tr}(V^\top C^{-\frac{1}{2}} \tilde{P}_{\text{nrm}} C^{-\frac{1}{2}} V) - \operatorname{tr}(V^\top C^{-1} V) \tag{5.13a}$$

$$= \max_{V \in \mathbb{V}} \operatorname{tr}(V^\top C^{-\frac{1}{2}} \tilde{P}_{\text{nrm}} C^{-\frac{1}{2}} V) - \operatorname{tr}(V^\top C^{-\frac{1}{2}} C^{-\frac{1}{2}} V) \quad (C^{-1} = C^{-\frac{1}{2}} C^{-\frac{1}{2}}) \tag{5.13b}$$

$$= \max_{U \in \mathbb{U}} \operatorname{tr}(U^\top \tilde{P}_{\text{nrm}} U) - \operatorname{tr}(U^\top U) \quad (\text{replacing } U := C^{-\frac{1}{2}} V) \tag{5.13c}$$

$$= \max_{U \in \mathbb{U}} \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}} \tilde{P} \tilde{C}^{-\frac{1}{2}} U) - \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}} \tilde{C} \tilde{C}^{-\frac{1}{2}} U) \quad (\text{since } \tilde{C}^{-\frac{1}{2}} \tilde{C} \tilde{C}^{-\frac{1}{2}} = I) \tag{5.13d}$$

$$= \max_{U \in \mathbb{U}} \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}} (\tilde{P} - \tilde{C}) \tilde{C}^{-\frac{1}{2}} U) \quad (\text{by factoring terms}) \tag{5.13e}$$

$$= \min_{U \in \mathbb{U}} \operatorname{tr}(U^\top \tilde{C}^{-\frac{1}{2}} \tilde{L} \tilde{C}^{-\frac{1}{2}} U) \quad (\text{since } \tilde{P} - \tilde{C} = -\tilde{L}) \tag{5.13f}$$

$$= \min_{U \in \mathbb{U}} \operatorname{tr}(U^\top \tilde{L}_{\text{nrm}} U). \quad (\text{by defn of } \tilde{L}_{\text{nrm}}) \tag{5.13g}$$

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

Remark 5.3.2. *The skew-symmetric part of $\tilde{L}_{\text{nrm}}$ does not affect the solution of (5.11) 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.*

Step 2: Estimating Universe Size

From (5.7) and (5.8), 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. 5-3 is five), which essentially determines the number of columns of U in (5.11) (or equivalently V in (5.4)). 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 5.3.1. *If L_{nrm} is the normalized Laplacian matrix of the cluster graph $\mathcal{G}$, then $\text{eig}(L_{\text{nrm}})$ consists of zeros and ones. Moreover, the multiplicity of the zero eigenvalues is the number of clusters.*

Proof. 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 [198, 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 [198], the conclusion follows. □

Lemma 5.3.1 implies that in the noiseless setting the number of zero eigenvalues of L_{nrm} is the size of universe, which is correctly recovered from (5.8) 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 5.3.2).

Lemma 5.3.2. *Consider the estimate $\tilde{m}$ obtained by (5.8) from $\tilde{L}_{\text{nrm}} = L_{\text{nrm}} + N$. If $\|N\| < 0.5$, then $\tilde{m} = m$.*

Proof. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_l$ denote ordered eigenvalues of L_{nrm}, where from Lemma 5.3.1 we have $\lambda_1 = \lambda_2 = \cdots = \lambda_m = 0$ and $\lambda_{m+1} = \lambda_{m+2} = \cdots = \lambda_l = 1$. If $\tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \cdots \leq \tilde{\lambda}_l$ are the ordered eigenvalues of $\tilde{L}_{\text{nrm}} = L_{\text{nrm}} + N$, from the Weyl's eigenvalue theorem [199] we have $|\tilde{\lambda}_i - \lambda_i| < \|N\|$ for all $i \in \mathbb{N}_l$. This implies, if $\|N\| < 0.5$, that $\left|\{\tilde{\lambda} : \lambda < 0.5\}\right| = m$, which shows the correct number of clusters is recovered. □

Lemma 5.3.2 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 $e_{\max} := \max\{e_1, e_2, \ldots, e_l\}$, where $e_i := \sum_{j=1}^{l} |(E)_{ij}|$ 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 5.3.3. *Given $e_{\max}$, $c_{\min}$ defined above and $\tilde{m}$ obtained from (5.8), if $e_{\max} < 0.5\, c_{\min}$, then $\tilde{m} = m$.*

Proof. We have

$$\|\tilde{L}_{\text{nrm}} - L_{\text{nrm}}\| = \|\tilde{C}^{-\frac{1}{2}}\, \tilde{L}\, \tilde{C}^{-\frac{1}{2}} - C^{-\frac{1}{2}}\, L\, C^{-\frac{1}{2}}\| \quad \text{(by defn of } \tilde{L}_{\text{nrm}}, L_{\text{nrm}}) \tag{5.14a}$$

$$= \|C^{-\frac{1}{2}}\, (\tilde{L} - L)\, C^{-\frac{1}{2}}\| \quad \text{(since by assumption } \tilde{C} = C) \tag{5.14b}$$

$$\leq \|C^{-1}\|\, \|\tilde{L} - L\| \quad \text{(since 2-norm is submultiplicative)} \tag{5.14c}$$

$$= \|C^{-1}\|\, \|(\tilde{D} - \tilde{A}) - (D - A)\| \quad \text{(since } L := D - A) \tag{5.14d}$$

$$= \|C^{-1}\|\, \|\tilde{A} - A\| \quad \text{(since } \tilde{C} = C \text{ and } D = C - I) \tag{5.14e}$$

$$= \|C^{-1}\|\, \|E\| \quad \text{(since } E := \tilde{A} - A) \tag{5.14f}$$

$$\leq \frac{1}{c_{\min}} \|E\| \quad \text{(since } C \text{ is diagonal)} \tag{5.14g}$$

$$\leq e_{\max}/c_{\min}, \quad \text{(since } \|E\| \leq e_{\max}) \tag{5.14h}$$

where the last inequality follows from the Gershgorin circle theorem [199, Sec. 6.1]. The conclusion follows from Lemma 5.3.2 and observing that $\|\tilde{L}_{\text{nrm}} - L_{\text{nrm}}\| = \|N\| < 0.5$ implies $e_{\max} < 0.5\, c_{\min}$. □

Proposition 5.3.3 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 5.3.3 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 (5.7) 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 (5.7) fixes the size of U in (5.11) 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 (5.4). 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 5.3.3. *In the spectral graph clustering methods, the "eigengap" heuristic is often used to estimate the number of clusters [196]. 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 $L_{\text{sym}} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. Unlike the normalized Laplacian L_{nrm} proposed in this work (see Lemma 5.3.1), 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 5.4.1, 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.*

Step 3: Lifting and Relaxation

In order to solve (5.11) 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

$$\begin{aligned} \underset{U\in\mathbb{R}^{l\times m}}{\text{minimize}} \quad & \operatorname{tr}\big(U^\top \tilde{L}_{\text{nrm}}\, U\big) \\ \text{subject to} \quad & U^\top U = I, \end{aligned} \tag{5.15}$$

which is a generalized Rayleigh quotient problem. From the Rayleigh-Ritz theorem [200, Sec 5.2.2], it follows that the solution of (5.15) 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 [196]. This similarity is not surprising given the graph-theoretic interpretation of our problem discussed in Section 5.3.2. Nevertheless, note that spectral graph clustering is based on $\tilde{L}_{\text{sym}} := \tilde{D}^{-\frac{1}{2}}\tilde{L}\tilde{D}^{-\frac{1}{2}}$ (or other normalized Laplacians) instead of $\tilde{L}_{\text{nrm}}$.

Step 4: Projection and Embedding

In order to obtain an approximate solution for the original problem (5.11), the solution $U^* \in \mathbb{R}^{l\times m}$ obtained from solving (5.15) 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 5.4.1, 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 [196, 201, 202], 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 [202]. 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 [144, 203]), our approach has a lower complexity that leads to superior execution time.

Our approach is based on noting that rows of V (defined in (5.3)) 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 $U := C^{-\frac{1}{2}} V$ 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 [204] 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 [196, Sec. 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 i-th 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

$$\begin{aligned} \underset{\mathcal{S} \subset \mathbb{N}_l}{\text{minimize}} \quad & \textstyle\sum_{i,j \in \mathcal{S}} |\langle u_i, u_j \rangle| \\ \text{subject to} \quad & |\mathcal{S}| = m. \end{aligned} \tag{5.16}$$

The greedy strategy explained in step 3 of the CLEAR algorithm can be leveraged to efficiently find an approximate solution for (5.16).

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

$$\begin{aligned} \underset{\mathcal{A}_1,\ldots,\mathcal{A}_m}{\text{minimize}} \quad & \textstyle\sum_{k=1}^{m} \sum_{v_j \in \mathcal{A}_k} \|u_j - u'_k\|^2 \\ \text{subject to} \quad & g(v_i) \neq g(v_j), \quad \forall v_i, v_j \in \mathcal{A}_s. \end{aligned} \tag{5.17}$$

The constraint in (5.17) 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 $(F)_{jk} := \|u_j - u'_k\|^2$, and denote its row blocks by

$$F = \begin{bmatrix} F^{1\top} & F^{2\top} & \cdots & F^{n\top} \end{bmatrix}^\top, \tag{5.18}$$

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,[4] (5.17) can be represented in matrix form as $\min_{V \in \mathbb{V}} \langle V, F \rangle$. Noting that

$$\min_{V \in \mathbb{V}} \langle V, F \rangle = \min_{P^i \in \mathbb{P}^L} \textstyle\sum_{i=1}^{n} \langle P^i, F^i \rangle \tag{5.19a}$$

$$= \textstyle\sum_{i=1}^{n} \min_{P^i \in \mathbb{P}^L} \langle P^i, F^i \rangle, \tag{5.19b}$$

and since each subproblem in (5.19b) is a linear assignment problem [103], 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 (5.19b). 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

[4] If the j-th entry in column k of V is nonzero, then $v_j \in \mathcal{A}_k$.

most of the time in term of accuracy, but has a considerable speed advantage.

Lastly, 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 solutions of (5.15) are only recovered up to an orthogonal transformation (i.e., if U^* is a solution so is U^*R for any orthogonal matrix R).

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 breadth-first search (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) [198]. 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 5.4.2), 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 $\mathrm{O}(n\,m^2\log(m))$.

5.3.6 Applications: Edge-Centric vs. Clique-Centric

In this section, we divide the applications that benefit from solving the multi-view 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 multi-view matching algorithms to associate features between multiple images for estimating relative transformation between the corresponding pair of camera poses [56] belongs to this category. The purpose of using multi-view 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.

In contrast, clique-centric applications ultimately seek to fuse information *globally* (i.e., across *all* views) as prescribed by the cliques of the association graph. For example, consider the map fusion problem that arises in single/multi-robot SLAM [20]. After identifying every sighting of each unique landmark in all maps (i.e., encoded in the cliques of a cycle-consistent association graph) via multi-view 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.[5] 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

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

Figure 5-7. 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.

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. 5-7 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.

5.4 Results

5.4.1 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 [59] and MatchALS [56] that are based on a convex relaxation; 2) Spectral algorithm [46] extended for partial permutations by Zhou et al. [56], MatchEig [60], and NMFSync [144] that are based on a spectral relaxation; 3) and QuickMatch [57] 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 (5.7) is used.

We report the F_1 score, which is commonly used in the literature and is defined as $f := \frac{2pr}{p+r} \in [0, 1]$, 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

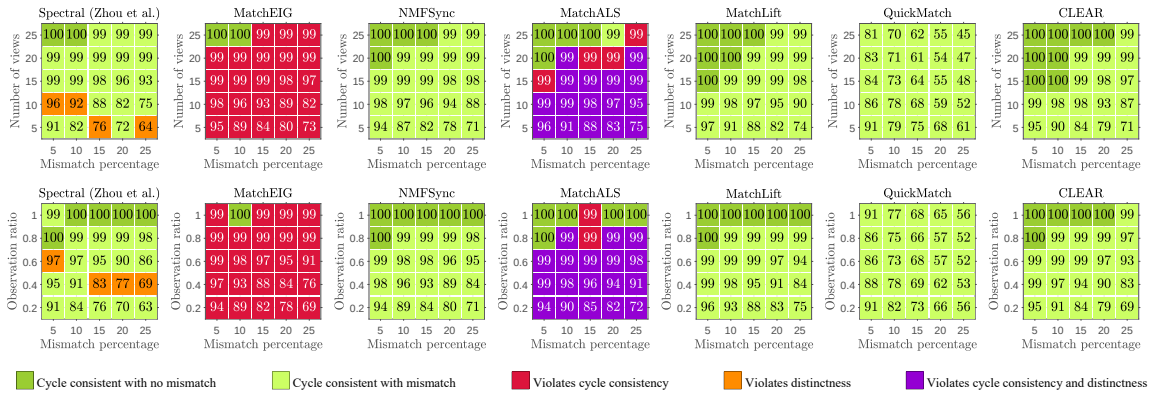


Figure 5-8. (Best viewed in color) Comparison of the state-of-the-art algorithms with CLEAR for uniformly sampled observations. Various number of views and observation ratios versus the percentage of mismatch in the input are considered. The F_1 score is reported in percentage in each grid (the higher the better). These values are computed based on individual edges in the association graph (i.e., for edge-centric applications); see Section 5.3.6.

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. 5-8 (in percentage). In the second comparison (second row in Fig. 5-8), 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. 5-8), 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. 5-8 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

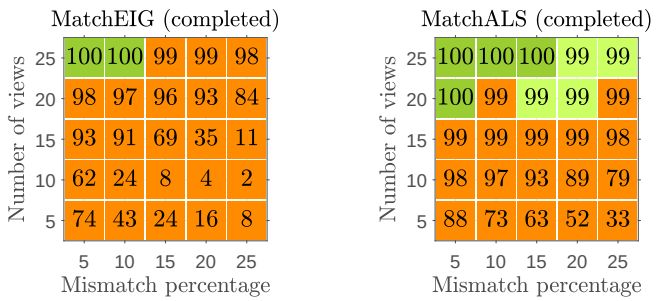


Figure 5-9. (Best viewed in color) The average F_1 score of the inconsistent algorithms after making them cycle consistent by completing the graph's connected components for clique-centric applications (see Section 5.3.6).

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. 5-8. 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. 5-8 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 5.3.6 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. 5-9 for MatchEIG and MatchALS algorithms (compare Fig. 5-9 with Fig. 5-8). For example, the average F_1 score of MatchEIG with 10 views and under 15% mismatch drops from 0.93 (Fig. 5-8) to 0.08 (Fig. 5-9). As discussed in Section 5.3.6, 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.

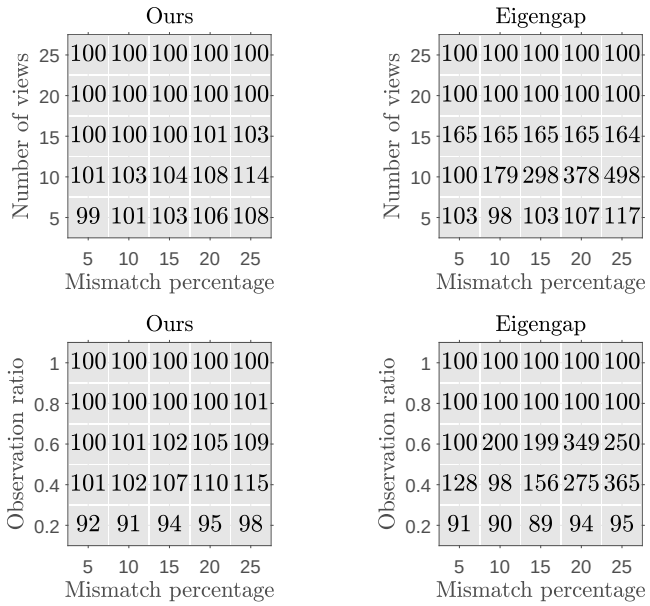


Figure 5-10. The average of estimated universe sizes in the Monte Carlo runs of Fig. 5-8 by CLEAR and the eigengap method based on the symmetric Laplacian. The closer to 100, the better.

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. 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 (5.7), with the eigengap method commonly used in the spectral graph clustering literature (see Remark 5.3.3). The results are reported in Fig. 5-10. The number written inside each square is the average of estimated universe sizes (rounded) in the Monte Carlo runs of Fig. 5-8. The correct universe size is 100. According to the results depicted in Fig. 5-10, 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

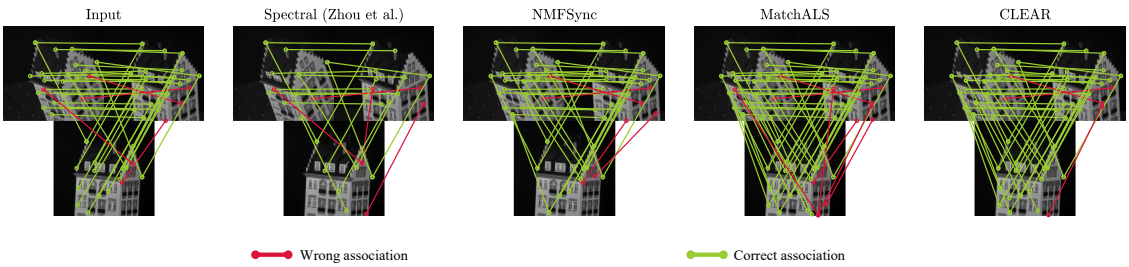


Figure 5-11. (Best viewed in color) An example of matched feature points across three images of the CMU Hotel dataset. Input, obtained by matching features across image pairs independently, contains error and is inconsistent. CLEAR returns cycle-consistent results and improves the precision of the input.

each figure).

5.4.2 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; 2) evaluate the precision/recall for the returned solutions.

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

[6]http://www.robots.ox.ac.uk/~vgg/data/data-aff.html
[7]http://pages.cs.wisc.edu/~pachauri/perm-sync/

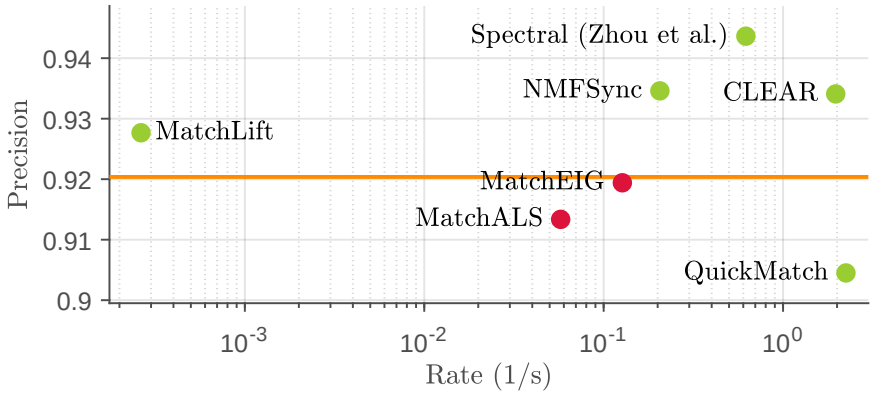


Figure 5-12. Precision vs. rate (the inverse of execution time) in CMU Hotel dataset. The rate axis has a logarithmic scale (CLEAR is about 3x faster than Spectral, 10x faster than NMFSync, 7500x faster than MatchLift). The precision of the input (based on individual edges and for edge-centric applications) is denoted by the orange line; see Section 5.3.6. Cycle-consistent/inconsistent outputs are respectively denoted by ● and ●. The closer to the top-right corner, the better.

the SIFT descriptor [107] of each feature point using the VLFeat library[8] [205]. 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 i7-7700K CPU @ 4.20GHz and 16GB RAM.

Fig. 5-11 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. 5-12 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 run-

[8] http://www.vlfeat.org/

times 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 5.3.6. 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 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 3x faster than Spectral, 10x faster than NMFSync, 7500x 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, which 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 0.5s to 0.7s.

Fig. 5-13 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 5.3.6) 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

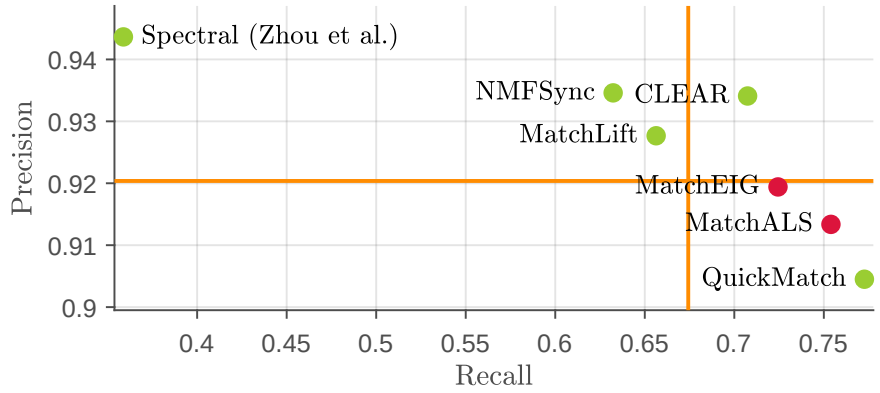


Figure 5-13. Precision vs. recall in CMU Hotel dataset. Precision and recall of input (based on individual edges and for edge-centric applications) are denoted by the orange lines; see Section 5.3.6. The input has a precision of 0.92 and a recall of 0.67 as denoted by the orange lines. The closer to the top-right corner, the better.

drop in precision underlines the importance of taking cycle consistency into account in evaluating multi-view matching algorithms for clique-centric applications.

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 [206, 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 is reported in Fig. 5-14. 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 5.3.6). CLEAR improves the precision and is considerably faster compared to the other algorithms that improve the input's precision: about 21x faster than Spectral, 39x faster than NMFSync, 3800x 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 0.92s to 49.5s.

The precision and recall of returned solutions are reported in Fig. 5-15. 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. 5-15 highlights the importance of taking cycle consistency into account in evaluating multi-view matching algorithms for clique-centric applications.

Forest Landmark-based SLAM Dataset

Map fusion is an important clique-centric application of the multi-view matching problem in single/multi-robot SLAM [20]. 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-

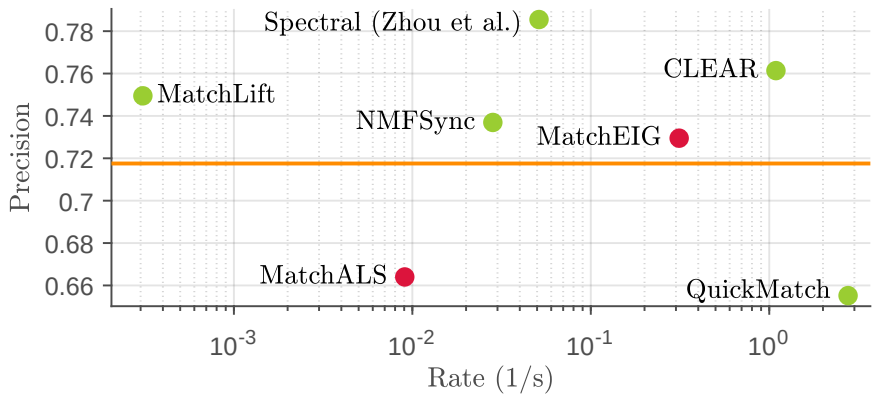


Figure 5-14. Precision vs. rate (the inverse of execution time) on the Graffiti dataset. The rate axis has a logarithmic scale (CLEAR is about 21x faster than Spectral, 39x faster than NMFSync, 3800x faster than MatchLift). Precision of the input is denoted by the orange line. Cycle consistent and inconsistent outputs are respectively denoted by ● and ●. The closer to the top-right corner, the better.

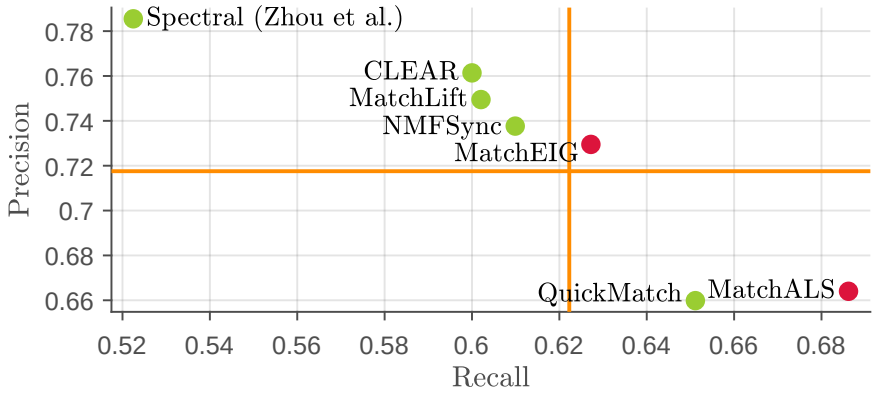


Figure 5-15. Precision vs. recall on Graffiti dataset. Precision and recall of input are denoted by the orange lines. The closer to the top-right corner, the better.

end.[9] In this section, we report the 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) [207]. 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. 5-16). The exploration mission lasts 120 seconds. 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

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

Figure 5-16. 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.

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 [12] 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 multi-view matching algorithms to achieve cycle consistency. Recall from our discussion in Section 5.3.6 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 5.3.6). 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 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,

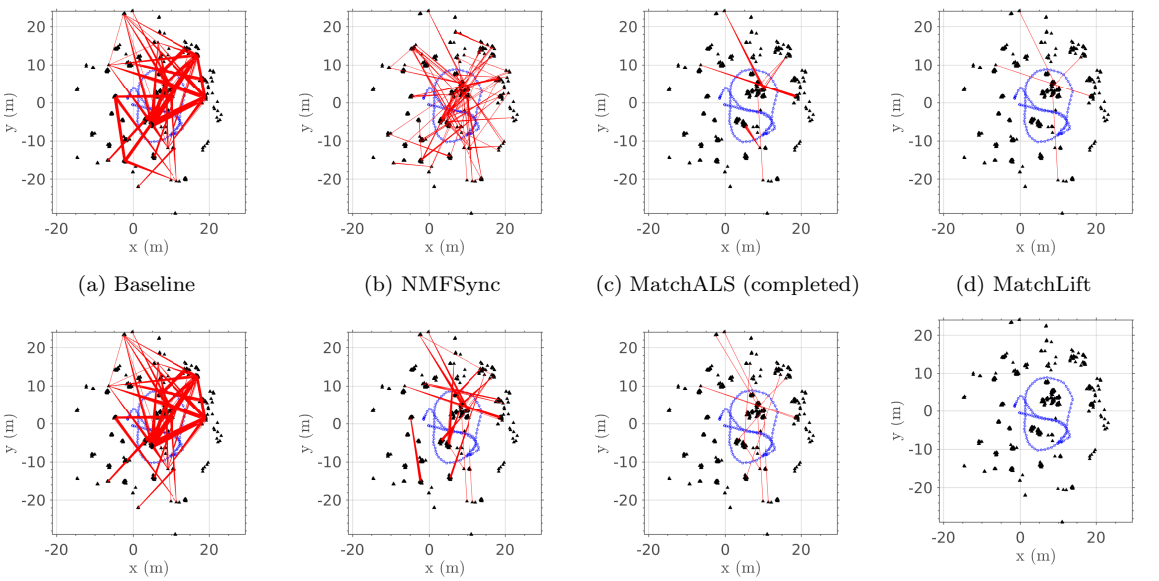
$$\bar{\mathrm{P}} := \frac{\mathrm{PP}}{\mathrm{DN} + \mathrm{PP}}. \tag{5.20}$$

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).

Table 5.2. Cross-comparison of algorithms in terms of the number of definite nagatives (DN), potential positives (PP), upper bound on precision ($\bar{\mathrm{P}}$), and runtime. The upper bound on precision is computed from (5.20).

Algorithm	DN	PP	$\bar{\mathrm{P}}$ (%)	Runtime (s)
CLEAR	**11**	3393	**99.677**	**0.084**
MatchLift [59]	**5**	2394	**99.792**	124.7
MatchALS [56] (completed)	89	15230	99.419	4.580
QuickMatch [57]	897	15757	94.614	0.118
NMFSync [144]	290	3233	91.768	4.272
MatchEIG [60] (completed)	21415	20381	48.763	1.808
Baseline	26249	**20487**	43.836	N/A

[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.



(a) Baseline (b) NMFSync (c) MatchALS (completed) (d) MatchLift

(e) MatchEIG (completed) (f) QuickMatch (g) CLEAR (h) CLEAR (post-processed)

Figure 5-17. 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 5.3.6). 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).

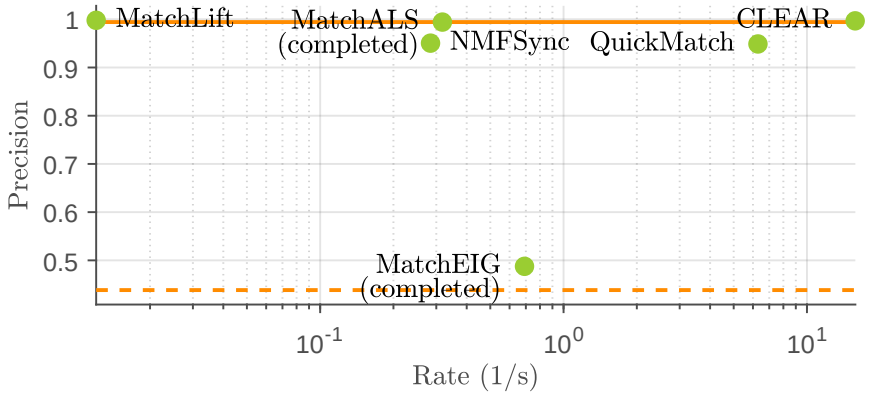


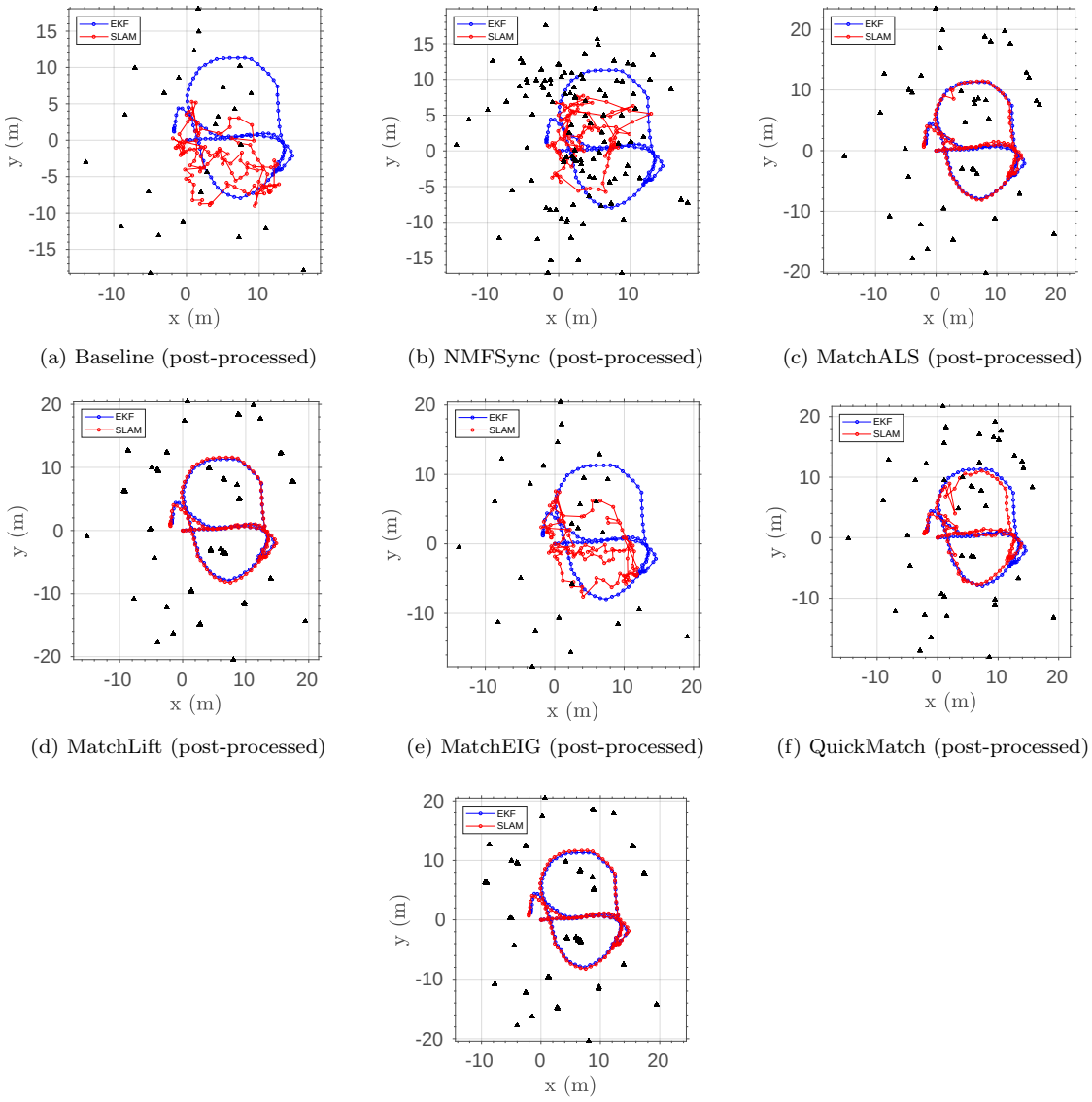
Figure 5-18. Precision upper bound $\bar{P}$ vs. rate in the landmark-based SLAM dataset. The rate axis has logarithmic scale. The closer to the top-right corner, the better. The solid orange line corresponds to the precision of input data associations. The dashed orange line corresponds to the precision of the baseline, obtained by completing the connected components in the input association graph. Further quantitative results are reported in Table 5.2.

Fig. 5-17 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 com-

pleting 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 5.2 shows the complete set of quantitative results and Fig. 5-18 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 25000 definite negatives; see Fig. 5-17(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 5.2. However, MatchLift requires 124.7 s to converge while CLEAR only takes 0.084 s. The superior speed 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. 5-17(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. 5-17(h) shows the resulting association after removing clusters of size smaller than four from the output of CLEAR. After this post-processing 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. 5-19 demonstrates the results of landmark-based SLAM using the data association returned by each algorithm. Prior to optimization, we apply the same post-processing 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 [208]. We note that Fig. 5-19 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 5.2.



(a) Baseline (post-processed)

(b) NMFSync (post-processed)

(c) MatchALS (post-processed)

(d) MatchLift (post-processed)

(e) MatchEIG (post-processed)

(f) QuickMatch (post-processed)

(g) CLEAR (post-processed)

Figure 5-19. 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 post-processed 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.

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.

5.5 Summary

Data association across multiple views is a fundamental problem in robotic applications. Traditionally, this problem is decomposed into a sequence of pairwise subproblems. Multi-view matching algorithms can leverage observation redundancy to improve the accuracy of pairwise associations. However, the use of these algorithms in robotic applications is often prohibited by their high computational complexity, as well as critical issues such as 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 multi-view association problem. CLEAR uses a spectral graph clustering technique, which is 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 can provide significant improvements in the accuracy and efficiency of data association in many applications that rely on pairwise matchings such as metric/semantic SLAM, multi-object tracking, and multi-view point cloud registration.

Chapter 6

Multiattribute, Multiway Fusion of Uncertain Pairwise Affinities

6.1 Introduction

As explored in the previous chapter, multiway matching attempts to correct outliers by exploiting observation redundancy and enforcing *cycle consistency*—a property stating that the composition of pairwise matchings over cycles must be identity. Many state-of-the-art multiway matching algorithms do this by permutation synchronization [46, 59–66, 73] of pairwise correspondences, which are improved and made consistent by joint optimization. These methods are effective at multiway matching when binary matchings are available; however, their use of (partial) permutation matrices is akin to late fusion [67, 68] and precludes them from using all available information, i.e., multiway matching is performed *after* each pairwise affinity matrix is pre-processed to create binary pairwise matches.

In contrast, this chapter presents MIXER (Multiway affinity matrIX fusER), an algorithm that produces cycle consistent multiway matches directly from noisy and uncertain pairwise affinities, i.e., in an early fusion sense. Similar to the conclusions of data fusion works in other contexts [67, 68], our experiments show that MIXER's early multiway fusion can yield a significant performance increase over existing late fusion approaches. Specifically, direct access to affinities enables a key property of

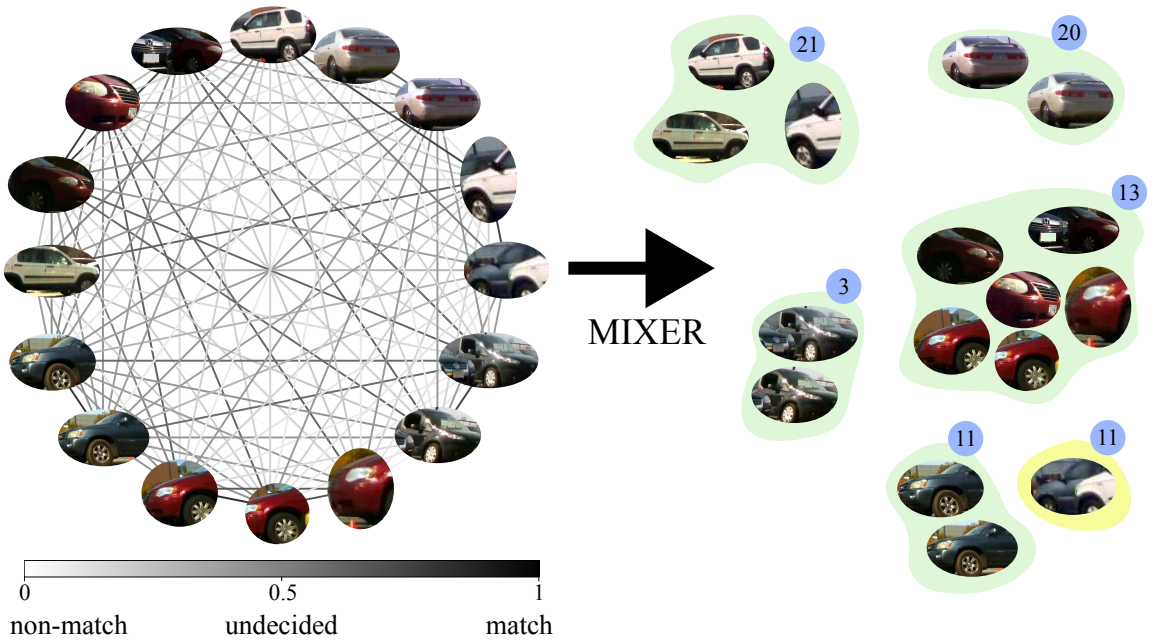


Figure 6-1. Multiattribute, multiway fusion example of car detections from different images. Each detection has SIFT, color, and bounding box attributes used to perform pairwise affinity scoring. Pairwise affinities are frequently uncertain, due to noisy detections and scoring processes, and the resulting multiway affinity matrix (graph representation, left) is inconsistent. Leveraging cycle consistency, distinctness constraints, and three modes of association, MIXER achieves high accuracy data fusion, correctly clustering identical cars (green clusters, right). If observations are too uncertain, MIXER tends to sacrifice recall in favor of precision, illustrated in the case of car 11—the detection in the yellow cluster is too ambiguous to be fused.

our optimization formulation of multiway fusion—three modes of association: non-match, uncertain match, and match, corresponding to 0, 0.5, and 1 affinities, respectively. These three modes also appear in a generalization of pairwise LAP called maximum-weight matching (MWM) [105, 106], where affinities less than or equal to a threshold (i.e., 0.5) will never be associated because they penalize the objective. MIXER extends these ideas to the multiway case (see Remark 6.3.1), balancing the three association modes in conjunction with cycle consistency and other problem constraints to achieve high accuracy in the presence of uncertain pairwise affinity scores. An example of MIXER fusing together car observations is shown in Fig. 6-1.

The availability of three association modes is especially important when working with data having multiple modalities or attributes (e.g., color, lidar reflectance, position, bag-of-words vector [180], SIFT descriptors [107], bounding box [209], shape [210], reID features [211], etc.). Because attributes may produce affinity scores in contention

(e.g., an object viewed from different viewpoints may have the same color, but not have bounding box overlap), the combined affinity becomes more uncertain, trending toward 0.5. This allows MIXER to defer to other pairwise affinities, and to the problem constraints, before making a decision.

We formulate the early multiway fusion problem as a mixed-integer quadratic program (MIQP). Since MIQPs are generally not scalable, we propose a novel continuous relaxation leading to approximate solutions. The main contribution of the MIXER algorithm over similar techniques is that its solutions are *guaranteed* to converge to feasible (binary) solutions of the original MIQP. Thus, rounding/projecting results to binary values—which is required when using other techniques and may lead to infeasible solutions—is avoided. To solve the relaxed problem efficiently, we present a projected gradient descent algorithm with backtracking line search. This polynomial-time algorithm has worst-case cubic complexity in problem size (from matrix-vector multiplies) at each iteration, and is guaranteed to converge to second-order stationary points [212, Prop. 7].

MIXER is evaluated on synthetic and real-world datasets and compare the results with state-of-the-art multiway data association algorithms. Our synthetic analysis shows an empirically tight optimality gap of MIXER solutions with respect to the global minimum of the MIQP, while achieving an average runtime of 8 ms for problems with 300 associations—four orders of magnitude faster than solving the MIQP with a general-purpose solver. Our real-world evaluation considers nine multiway matching benchmark datasets, showing that even in the *single* attribute early fusion setting, MIXER is able to achieve high accuracy, superior to the state of the art. Finally, we collect our own dataset of RGB images recorded in a parking lot and attempt to fuse observations of cars seen from multiple viewpoints. Three complementary attributes of cars are extracted (bounding box, color, and SIFT visual appearance) and we show that MIXER significantly outperforms existing algorithms, increasing F_1 score over the next-best result by 32% while being 49x faster. In summary, the main contributions of the work presented in this chapter are:

1. A principled formulation of multiway fusion as an MIQP that when approached

in an early fusion framework leads to three states of association and a multiway extension of the pairwise MWM problem.

2. A novel continuous relaxation of the multiway fusion MIQP leading to MIXER, a polynomial-time algorithm based on projected gradient descent that converges to stationary points.

3. Theoretical analysis of the continuous relaxation, showing that the MIXER algorithm is guaranteed to converge to feasible, binary solutions of the original MIQP.

4. Substantial accuracy and timing improvements over state-of-the-art multiway maching algorithms on standard benchmarks and in a challenging, self-collected RGB dataset with three distinct attributes.

6.2 Background

In this section, we formalize a principled framework to solve the multiway fusion problem. Consider n sets of data $\mathcal{S}_i$, $i = 1, \dots, n$, with cardinality $|\mathcal{S}_i| = m_i$ and let $m = \sum_{i=1}^{n} m_i$. We define the *universe* as $\mathcal{U} := \cup_i \mathcal{S}_i$, with $|\mathcal{U}| = k \leq m$ distinct elements across all sets. For example, Fig. 6-2 shows $n = 3$ images, each with m_i car detections with bounding box attributes. These are denoted by $\mathcal{S}_1 := \{a, b, c\}$, $\mathcal{S}_2 := \{d, e\}$, and $\mathcal{S}_3 := \{f\}$. Assuming that we know observations a, d, f and b, e represent the same cars, the universe is $\mathcal{U} := \{a, b, c\}$ and has $k = 3$ elements.

Given two sets $\mathcal{S}_i$ and $\mathcal{S}_j$, we define the *pairwise affinity matrix* between observations as

$$S_{ij} := \begin{bmatrix} s_{11} & \cdots & s_{1m_j} \\ \vdots & \ddots & \vdots \\ s_{m_i 1} & \cdots & s_{m_i m_j} \end{bmatrix} \in [0,1]^{m_i \times m_j}, \tag{6.1}$$

where $s_{ab} \in [0, 1]$ denotes the similarity between elements $a \in \mathcal{S}_i$ and $b \in \mathcal{S}_j$. Scores of 0, 0.5, and 1 correspond to maximum dissimilarity, maximum uncertainty/no preference, and maximum similarity, respectively. The *multiway affinity matrix* between

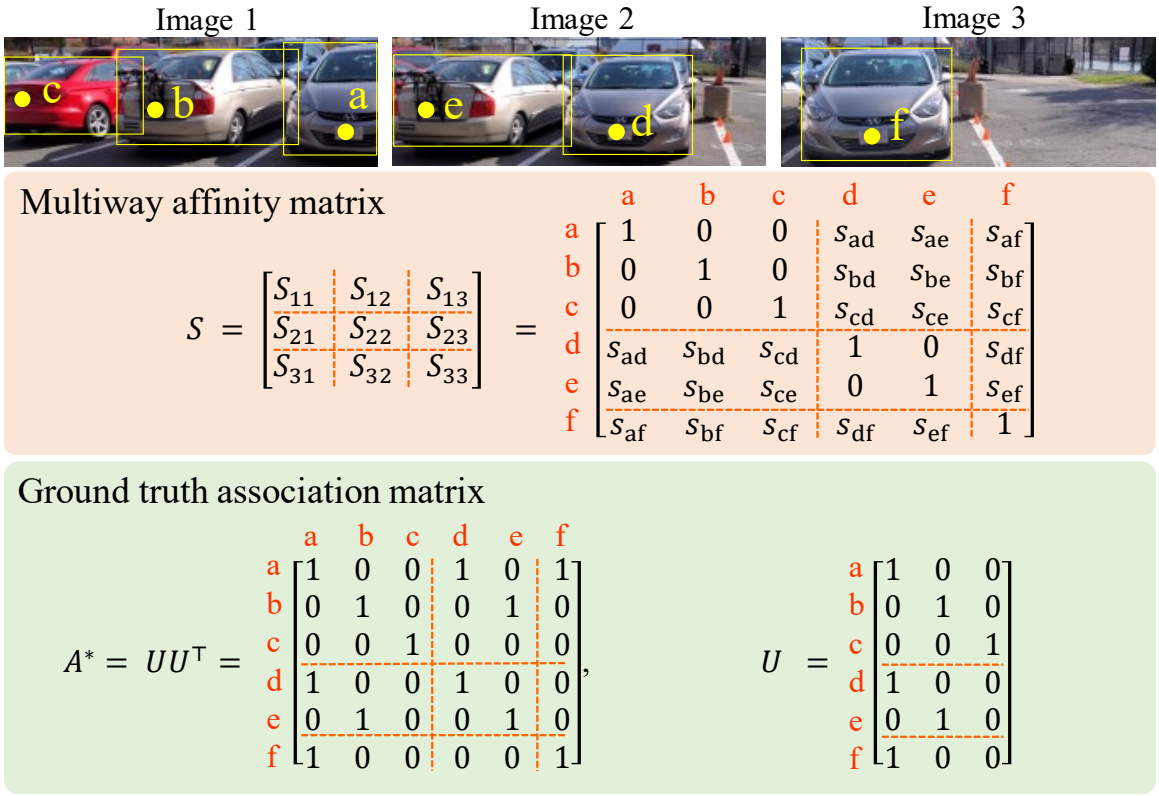


Figure 6-2. Example with $k = 3$ cars across $n = 3$ images, with a total of $m = 6$ observations. Bounding box overlap (had they been drawn on the same image) gives the similarity score between two cars. The multiway affinity matrix S and its corresponding ground truth A^* are shown.

all sets is defined as the symmetric matrix

$$S := \begin{bmatrix} S_{11} & \cdots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{n1} & \cdots & S_{nn} \end{bmatrix} \in [0,1]^{m \times m}, \tag{6.2}$$

where, by definition, $S_{ij} = S_{ji}^\top$. The example in Fig. 6-2 shows the multiway affinity matrix S (henceforth called the affinity matrix for simplicity), where $m = 6$. Pairwise association matrix blocks are separated by dashed lines.

Ground-truth association. The ground-truth pairwise affinity matrices take values of 1 for identical objects and 0 otherwise. When affinity matrices are binary, like in the ground truth case, we refer to them as association matrices. Furthermore, the ground-truth multiway association matrix, denoted by A^*, can be factorized as

$A^* = UU^\top$, with the *universal association matrix* U is defined as

$$U := \begin{bmatrix} U_1 & \cdots & U_n \end{bmatrix} \in \{0,1\}^{m \times k}. \tag{6.3}$$

Matrices $U_i \in \{0,1\}^{m_i \times k}$ represent associations between elements of sets S_i and $\mathcal{U}$. Fig. 6-2 shows A^* and U for the corresponding example, with $k = 3$ unique cars across all images and where matrices U_i are separated by dashed lines.

Constraints. Often, data association algorithms must meet certain constraints imposed by the high-level task. The *one-to-one* constraint states that an object can be associated with at most one other object. This constraint is satisfied if each row of U has a single 1 entry. The *distinctness* constraint states that objects within a set are distinct and therefore should not be associated. This is satisfied if there is at most a single 1 entry in each column of U_i. When the association problem is solved across more than two sets, associations must be *cycle consistent*, that is, if $a \leftrightarrow b$, and $b \leftrightarrow c$, then $a \leftrightarrow c$, where a, b, c are objects to be matched and $\leftrightarrow$ indicates a pairwise match. This condition is satisfied if the association matrix can be factorized as $A = UU^\top$ [56]. These three constraints are crucial for detecting and correcting erroneous similarity scores and associations, but increase the difficulty and complexity of the data association procedure.

6.3 Approach

6.3.1 Optimization Formulation

Given noisy and potentially incorrect similarity scores, our goal is to rectify their values to binary scores that respect the one-to-one, distinctness, and cycle consistency

constraints. This goal can be formally stated as finding U that solves the MIQP

$$\begin{aligned} \underset{U\in\{0,1\}^{m\times k}}{\text{minimize}} \quad & \left\|U\,U^\top - S\right\|_F^2 && \text{(cycle consistency)} \\ \text{subject to} \quad & U\,\mathbf{1}_m = \mathbf{1}_m && \text{(one-to-one)} \\ & U_i^\top \mathbf{1}_{m_i} \le \mathbf{1}_{m_i} && \text{(distinctness)} \end{aligned} \tag{6.4}$$

where $\|\cdot\|_F$ is the matrix Frobenius norm and $\mathbf{1}$ denotes the vector of all ones (with size shown as subscript). When the universe size k is known (or a good estimate $\hat{k}$ is available), this can be used directly in (6.4) to determine the dimensions of U. However, k is rarely known in practice and can be difficult to reliably estimate, so instead we allow the algorithm to simultaneously estimate U and k by setting $k = m$. Then, a solution U of (6.4) will have $\hat{k}$ nonzero columns, giving an estimate of the universe size.

Remark 6.3.1 (Multiway MWM)**.** *For $n = 2$, problem (6.4) reduces to a linear assignment problem (LAP). Replacing the affinity matrix by $S \leftarrow \frac{1}{2}(\mathbf{1} - S)$, problem (6.4) returns a perfect matching and solutions are equivalent to Hungarian solutions. With S as specified in Sec. 6.2, (6.4) solves the MWM problem [106] and can return imperfect matchings. This generalization of the LAP is often more natural for data association since not all data should always be fused.*

The traditional two-way maximum-weight matching (MWM) problem is often reduced so that a perfect matching exists [105] and is solved using the Hungarian algorithm. This is done by setting any affinity below a threshold to 0 and then discarding any match with 0 affinity in post-processing. Instead, (6.4) does not require this explicit thresholding and extends to $n > 2$, solving the multiway maximum-weight matching problem by allowing the information from multiple pairs of views to inform the final decision.

6.3.2 Equivalent Penalty Form

Before approaching a solution strategy for problem (6.4), we develop an equivalent penalty form. This penalty form will be useful for analysis and will lead to insights

that will be useful for designing the MIXER algorithm.

Equivalent problem. From the definition of Frobenius norm, the objective of (6.4) can be expanded and simplified as

$$\begin{aligned}\|UU^\top - S\|_F^2 &= \big\langle UU^\top - S,\ UU^\top - S\big\rangle \\ &= \|UU^\top\|_F^2 - 2\big\langle UU^\top,\ S\big\rangle + \|S\|_F^2,\end{aligned} \tag{6.5}$$

where $\big\langle A, B\big\rangle := \operatorname{tr}(A^\top B) = \sum_{ij} A_{ij}B_{ij}$ is the Frobenius inner product of matrices A and B. Further, it holds that

$$\|UU^\top\|_F^2 = \sum_{i,j}(UU^\top)_{ij}^2 = \sum_{i,j}(UU^\top)_{ij} = \big\langle UU^\top, \mathbf{1}_{m\times m}\big\rangle, \tag{6.6}$$

where the second equality follows by noting that entries of U are binary and therefore equal to their square. Combining (6.5) and (6.6), the objective can be written as

$$\|UU^\top - S\|_F^2 = \big\langle UU^\top,\ \mathbf{1}_{m\times m} - 2S\big\rangle + \|S\|_F^2. \tag{6.7}$$

Since S is given and is a constant of the optimization problem, the term $\|S\|_F^2$ does not affect the solution and can be omitted from the objective. Thus, defining $\bar{S} := \mathbf{1}_{m\times m} - 2S$, problem (6.4) is equivalent to the following problem

$$\begin{aligned}\underset{U\in\{0,1\}^{m\times m}}{\text{minimize}} \quad & \big\langle UU^\top, \bar{S}\big\rangle && \text{(cycle consistency)} \\ \text{subject to} \quad & U\,\mathbf{1}_m = \mathbf{1}_m && \text{(one-to-one constraint)} \\ & U_i^\top \mathbf{1}_{m_i} \le \mathbf{1}_m && \text{(distinctness constraint)}\end{aligned} \tag{6.8}$$

Penalty functions. Next, we introduce the standard simplex and two penalty functions, allowing the reformulation of the constraints of (6.8) into an equivalent problem. With slight abuse of notation, the *standard simplex* defined for each row of a matrix is

$$\Delta^{m\times m} := \{U \in \mathbb{R}_+^{m\times m} : U\mathbf{1}_m = \mathbf{1}_m\}. \tag{6.9}$$

Observe that $\{0,1\}^{m\times m} \subset \Delta^{m\times m}$ and that $\Delta^{m\times m}$ captures the one-to-one constraint. Using the standard simplex, we first show that a binary $U \in \Delta^{m\times m}$ must have orthogonal columns. This orthogonality property is *implicit* in the original formulation and we explictly include it in the relaxation.

Lemma 6.3.1. *A matrix $U \in \Delta^{m\times m}$ is binary if and only if the columns of U are orthogonal.*

Proof. Suppose, by contradiction, two columns v and w of U give $\langle v, w\rangle \neq 0$. Then, since U is binary, there exists at least one $k \in \{1,\dots,m\}$ such that $v_k = 1$ and $w_k = 1$. This implies that there are at least two 1 entries in the k'th row of U. Consequently, the k-th row of vector $U\mathbf{1}_m$ is strictly greater than 1, which violates the constraint $U\mathbf{1}_m = \mathbf{1}_m$.

Conversely, assume $U \in \Delta^{m\times m}$ has orthogonal columns. Without loss of generality, we show that the first row of U is binary and, by applying a similar argument to other rows, we conclude that U is binary. Denote the first row of U by v_1. Because $U \in \Delta^{m\times m}$, there exists $i \in \{1,\dots,m\}$ such that $(v_1)_i > 0$. Let u_i denote the i-th column of U. Orthogonality of columns implies that for all $j \neq i$,

$$\begin{aligned} 0 = u_i^\top u_j &= \sum_{k=1}^{m} (u_i)_k (u_j)_k \\ &= (u_i)_1 (u_j)_1 + \sum_{k=2}^{m} (u_i)_k (u_j)_k \\ &= (v_1)_i (v_1)_j + \sum_{k=2}^{m} (u_i)_k (u_j)_k. \end{aligned} \tag{6.10}$$

Because U is non-negative, for (6.10) to hold we must have $(v_1)_i(v_1)_j = 0$ for all $j \neq i$. Since $(v_1)_i > 0$, it follows that $(v_1)_j = 0$ for all $j \neq i$. Further, since the sum of the entries of v_1 is equal to 1, we have that $(v_1)_i = 1$ so that the first row of U is binary. □

Because we seek binary $U \in \Delta^{m\times m}$, we introduce a penalty function corresponding

to the column-wise orthogonality of U, defined as

$$\phi_{\text{orth}}(U) := \left\langle U^\top U,\ P_o \right\rangle, \tag{6.11}$$

where $P_o := \mathbf{1}_{m\times m} - I_{m\times m}$. Note that by definition, $\left\langle U^\top U,\ P_o \right\rangle := \sum_{i,j} (U^\top U)_{ij} (P_o)_{ij} = \sum_{i\neq j} (U^\top U)_{ij}$, which is the sum of non-diagonal entries. Therefore, $\phi_{\text{orth}}(U) = 0$ if and only if the columns of U are orthogonal.

The second penalty function corresponds to the distinctness constraint and is defined as

$$\phi_{\text{dist}}(U) := \left\langle UU^\top,\ P_d \right\rangle, \tag{6.12}$$

where $P_d := \text{blockdiag}(P_{d1}, \ldots, P_{dn})$ and each $m_i \times m_i$ matrix $P_{di} := 2\,(\mathbf{1}_{m_i\times m_i} - I_{m_i\times m_i})$ ensures that the m_i observations of view i are distinct.

Lemma 6.3.2. *Given $U \in \{0,1\}^{m\times m}$ and $\phi_{\text{dist}}(U)$ as defined, $\phi_{\text{dist}}(U) = 0$ if and only if $U_i^\top \mathbf{1}_{m_i} \leq \mathbf{1}_m$.*

Proof. Expanding $\phi_{\text{dist}}(U)$ based on the n matrix blocks in U and P_d gives

$$\begin{aligned}
\left\langle UU^\top,\ P_d \right\rangle &= \sum_{i=1}^{n} \left\langle U_i U_i^\top,\ P_{di} \right\rangle \\
&= \sum_{i=1}^{n} \left\langle U_i U_i^\top,\ \mathbf{1}_{m_i\times m_i} - I_{m_i\times m_i} \right\rangle \\
&= \sum_{i=1}^{n} \sum_{j\neq r} (U_i U_i^\top)_{jr}.
\end{aligned} \tag{6.13}$$

Since U_i is binary, the latter summation is zero if and only if all matrices $U_i U_i^\top$ are diagonal. Suppose $\left\langle UU^\top,\ P_d \right\rangle = 0$ and, by contradiction, there exists U_i for which $U_i U_i^\top$ is non-diagonal. This implies U_i has at least two non-orthogonal rows. From a similar proof-by-contradiction argument used in the proof of Lemma 6.3.1 (based on rows instead of columns), non-orthogonality implies that there exists k such that the k-th elements of these two non-orthogonal rows are 1. Therefore, the k-th element of $U_i^\top \mathbf{1}_{m_i}$ is strictly greater than 1, a contradiction. Now suppose $U_i^\top \mathbf{1}_{m_i} \leq \mathbf{1}_m$. Since U_i is binary, this implies that if the k-th element of a row of U_i is 1, then the k-th

element of all other rows must be 0. Consequently, rows of U_i are orthogonal, which implies $\sum_{j \neq r} (U_i U_i^\top)_{jr} = 0$ and therefore $\langle UU^\top, \ P_d \rangle = 0$. □

Using the standard simplex (6.9) and the penalty functions (6.11), (6.12), problem (6.8) can be equivalently expressed as

$$\begin{aligned}
\underset{U}{\text{minimize}} \quad & \langle UU^\top, \bar{S} \rangle && \text{(cycle consistency)} \\
\text{subject to} \quad & U \in \Delta^{m \times m} && \text{(one-to-one constraint)} \\
& U \in \{0,1\}^{m \times m} && \text{(binary constraint)} \\
& \phi_{\text{orth}}(U) = 0 && \text{(orthogonality constraint)} \\
& \phi_{\text{dist}}(U) = 0 && \text{(distinctness constraint)}
\end{aligned} \tag{6.14}$$

6.3.3 Continuous Relaxation

Due to its binary domain, solving (6.14) to global optimality requires combinatorial techniques that quickly become intractable as the problem size grows. To increase scalability, the standard workaround is to relax the domain of the problem to the positive orthant. However, these solutions must be subsequently *rounded* (i.e., projected back to binary values), which can be problematic since this may produce infeasible solutions that violate the original constraints. A key novelty of our relaxation approach and algorithm is that solutions are *guaranteed* to converge to *feasible*, *binary* solutions of the original problem (6.4), thereby obviating the potentially problematic rounding step.

To proceed, we relax the binary constraint of problem (6.14) and include the orthogonality and distinctness constraints into the objective, scaled by $d \geq 0$,

$$\begin{aligned}
\underset{U \in \mathbb{R}_+^{m \times m}}{\text{minimize}} \quad & F(U) := \langle UU^\top, \mathbf{1} - 2S \rangle \\
& \qquad + d\big(\phi_{\text{orth}}(U) + \phi_{\text{dist}}(U)\big). \\
\text{subject to} \quad & U\mathbf{1}_m = \mathbf{1}_m
\end{aligned} \tag{6.15}$$

Because $\phi_{\text{orth}} \geq 0$ and $\phi_{\text{dist}} \geq 0$, the parameter d must be non-negative so that any constraint violation incurs a cost.

The objective $F(U)$ of problem (6.15) is nonconvex. Thus, convergence to a first-order stationary point is not enough as critical points may either be local minima or saddle points. Therefore, higher-order information must be used so that convergence to second-order stationary points can be achieved, implying convergence to local minima [212]. We leverage the generic algorithmic framework of [212] for escaping strict saddle points in constrained optimization by searching for feasible directions in the nullspace of the gradient that have negative curvature with respect to the Hessian $\nabla^2 F$ [212, Thm. 4]. Equipped with the ability to converge to second-order stationary points, our main result concerning (6.15) is stated in the following theorem, where *solutions* refer to (local) minima.

Theorem 6.3.1. *For $d \geq m+1$, solutions $U^\star$ of problem (6.15) are feasible solutions of problem (6.14). In particular, $U^\star$ is cycle consistent, distinct, and binary.*

Proof. See Section 6.3.4. □

For a comparison of our relaxation and its guarantees with existing approaches having similar formulations, see Table 6.1. The closest formulations to (6.15) in Table 6.1 are MatchDGD [63] and MatchRTR [64], both of which include a regularizer that penalizes non-binary solutions, similar to the role of ϕ_{orth}. However, inclusion of this regularizer does not *guarantee* that solutions will be binary. Thus, a Hungarian-based rounding step is required, incurring additional computation and potentially resulting in a solution that causes an increase in objective. Additionally, MatchRTR [64] is not able to guarantee that solutions satisfy distinctness.

6.3.4 Theoretical Analysis

We present theoretical insights behind the relaxed problem (6.15) which lead to the MIXER algorithm. Consider the relaxation in standard form [213] so that the con-

Table 6.1. Comparison of our MIXER formulation with multiway matching algorithms that also relax combinatorial problems with a Frobenius objective. The resulting relaxations are similar, but MIXER guarantees that solutions are cycle consistent (cyc.), distinct (dis.), and binary (bin.). Further, MIXER obtains solution using an efficient projected gradient descent (PGD) method where the projection $\Pi_{\mathcal{C}}$ onto the constraint set $\mathcal{C}$ can be evaluated in closed form. When a binary solution is not guaranteed, superscripts 'c', 't', and 'h' indicate rounding via clustering, thresholding, and Hungarian, respectively. Parameters λ and α arise due to sparsity regularization (different from our perspective) and are set to the defaults suggested by the respective authors.

Algorithm	Objective	Constraint Set	Solution Method	$\Pi_{\mathcal{C}}(\cdot)$	Soln. Guarantees		
					cyc.	dis.	bin.
MatchLift [59]	$\underset{A\in\mathbb{R}_+^{m\times m}}{\text{minimize}}\ \langle A, \mathbf{1}-\frac{1}{\lambda}S\rangle$	$A_{ii}=I_{m_i},\ \forall_i,\ \left[\begin{smallmatrix} k & \mathbf{1}^\top \\ \mathbf{1} & A\end{smallmatrix}\right]\succeq \mathbf{0}$	SDP/ADMM	–	✓	✓	✗ c
MatchALS [56]	$\underset{A\in\mathbb{R}_+^{m\times m}}{\text{minimize}}\ \langle A, \mathbf{1}-\frac{1}{\alpha}S\rangle + \lambda\|A\|_*$	$A_{ii}=I_{m_i},\ \forall_i,\ A_{ij}=A_{ji}^\top,\ \forall_{i\neq j}$ $\mathbf{0}\le A_{ij}\mathbf{1}\le\mathbf{1},\ \mathbf{0}\le A_{ij}^\top\mathbf{1}\le\mathbf{1}$	ADMM	LP	✗	✓	✗ t
MatchDGD [63]	$\underset{U\in\mathbb{R}_+^{m\times k}}{\text{minimize}}\ \langle UU^\top, \mathbf{1}-2S\rangle + \sum_i \|I-U_iU_i^\top\|_F^2$	$U\mathbf{1}=\mathbf{1},\ U_i^\top\mathbf{1}\le\mathbf{1},\ \forall_i$	PGD	ADMM	✓	✓	✗ h
MatchRTR [64]	$\underset{U\in\mathbb{R}_+^{m\times k}}{\text{minimize}}\ \langle UU^\top, \mathbf{1}-2S\rangle + \sum_i \|I-U_iU_i^\top\|_F^2$	$U\mathbf{1}=\mathbf{1}$	Riemannian trust-region	closed form	✓	✗	✗ h
MIXER	$\underset{U\in\mathbb{R}_+^{m\times m}}{\text{minimize}}\ \langle UU^\top, \mathbf{1}-2S\rangle + d\big(\phi_{\text{orth}}(U)+\phi_{\text{dist}}(U)\big)$	$U\mathbf{1}=\mathbf{1}$	PGD	closed form	✓	✓	✓

straints are explicit and with $\bar{S} := \mathbf{1} - 2S$, restated here for convenience

$$\begin{aligned} \underset{U \in \mathbb{R}_+^{m \times m}}{\text{minimize}} \quad & F(U) := \left\langle UU^\top, \bar{S} \right\rangle + d\big(\phi_{\text{orth}}(U) + \phi_{\text{dist}}(U)\big) \\ \text{subject to} \quad & U \geq \mathbf{0}_{m \times m} \\ & U\mathbf{1}_m - \mathbf{1}_m = \mathbf{0}_m \end{aligned} \tag{6.16}$$

The scalar $d \geq 0$ controls the strength of the penalty functions. Intuitively, increasing d pushes solutions of (6.16) towards binary, distinct solutions. Because $\phi_{\text{orth}}(U) \geq 0$ and $\phi_{\text{dist}}(U) \geq 0$, d is restricted to be non-negative so that constraint violation penalizes the objective. In this appendix, we will show that using Algorithm 3, once d is larger than a finite value $\phi_{\text{orth}}(U) = \phi_{\text{dist}}(U) = 0$ so that local minima of (6.16) are feasible solutions of problem (6.14).

The Lagrangian of (6.16) is

$$\mathcal{L}(U; Y, \lambda) := F - \left\langle Y,\ U \right\rangle - \left\langle \lambda,\ U\mathbf{1}_m - \mathbf{1}_m \right\rangle, \tag{6.17}$$

where $\lambda \in \mathbb{R}^m$ and $Y \in \mathbb{R}^{m \times m}$ are the Lagrange multipliers for the equality and inequality constraints, respectively. From the first-order optimality conditions, stationary points $U^\star \in \Delta^{m \times m}$ of (6.16) must satisfy the KKT conditions

$$\nabla_U \mathcal{L} = \nabla F(U^\star) - Y^\star - \lambda^\star \mathbf{1}_m^\top = 0, \tag{6.18a}$$

$$U^\star \mathbf{1}_m - \mathbf{1}_m = \mathbf{0}_m, \tag{6.18b}$$

$$U^\star_{ij} \geq 0,\ \forall_{ij}, \tag{6.18c}$$

$$Y^\star_{ij} \geq 0,\ \forall_{ij}, \tag{6.18d}$$

$$Y^\star_{ij} U^\star_{ij} = 0,\ \forall_{ij}, \tag{6.18e}$$

where $\nabla F(U) = 2\bar{S}U + 2d\big(UP_o + P_d U\big)$.

We begin by analyzing gradient entries corresponding to non-zero entries of a given stationary point.

Lemma 6.3.3. *At a stationary point $U^\star \in \Delta^{m\times m}$ of (6.16), entries $U^\star_{ij} \neq 0$ of the i-th row have equal corresponding gradient entries ∇F_{ij}. In particular, $\nabla F_{ij} = \sum_{k=1}^{m} (\nabla F \odot U^\star)_{ik}$.*

Proof. Considering the stationarity condition (6.18a), we can remove the dependence on $Y^\star$ by element-wise right multiplication of $U^\star$. Because of complementarity (6.18e), we have

$$\nabla F \odot U^\star - \lambda^\star \mathbf{1}_m^\top \odot U^\star = 0, \tag{6.19}$$

where $\odot$ denotes the Haddamard product. Observing that $(\lambda^\star \mathbf{1}_m^\top \odot U^\star)\mathbf{1}_m = \lambda^\star$ because $U^\star \mathbf{1}_m = \mathbf{1}_m$, multiplying on the right by $\mathbf{1}_m$ and isolating $\lambda^\star$ yields

$$\lambda^\star = (\nabla F \odot U^\star)\mathbf{1}_m \tag{6.20}$$

Our aim is to eliminate explicit dependence of $\lambda^\star$ from (6.19). Substituting (6.20) into (6.19) yields

$$\begin{aligned} \nabla F \odot U^\star - (\nabla F \odot U^\star)\mathbf{1}_m \mathbf{1}_m^\top \odot U^\star &= 0 \\ (\nabla F - (\nabla F \odot U^\star)\mathbf{1}_{m\times m}) \odot U^\star &= 0, \end{aligned}$$

which implies one of the following two cases

$$\begin{cases} U^\star_{ij} = 0 \\ U^\star_{ij} \neq 0, \ (\nabla F - (\nabla F \odot U^\star)\mathbf{1}_{m\times m})_{ij} = 0 \end{cases}$$

Through simplification, when $U^\star_{ij} \neq 0$ we have

$$\begin{aligned} \nabla F_{ij} &= ((\nabla F \odot U^\star)\mathbf{1}_{m\times m})_{ij} \\ &= (\nabla F \odot U^\star)_{i:}(\mathbf{1}_{m\times m})_{:j} \\ &= \textstyle\sum_{k=1}^{m} (\nabla F \odot U^\star)_{ik}. \end{aligned} \tag{6.21}$$

Thus, on row i, for any j such that $U^\star_{ij} \neq 0$ we have that the corresponding ∇F_{ij} is equal to (6.21). Therefore, all ∇F_{ij} corresponding to a non-zero $U^\star_{ij}$ on row i are

equal. □

Next, we analyze non-binary stationary points $U^\star \in \Delta^{m\times m}$ of (6.16). From the analysis that follows, we will conclude that non-binary stationary points are *strict saddle points*, i.e., critical points where there is at least one direction along which the curvature is strictly negative. Note that this definition includes local maxima. The following proposition gives the necessary and sufficient conditions for identifying a strict saddle point.

Proposition 6.3.1 ([212,213])**.** *Given convex constraints $\mathcal{C}$, a stationary point $x^\star \in \mathcal{C}$ is a strict saddle point of the nonlinear problem $\min_{x\in\mathcal{C}} f(x)$ if it satisfies the first-order necessary conditions and there are directions that are undecided to first order, but have negative curvature (i.e., will decrease the objective). That is, the following conditions must be satisfied*

(i) $\nabla f(x^\star)^\top (x - x^\star) \geq 0, \quad \forall\, x \in \mathcal{C},$

(ii) $\exists\, y \in \mathcal{C}$ *such that* $\nabla f(x^\star)^\top (y - x^\star) = 0$
and $(y - x^\star)^\top \nabla^2 f(x^\star)(y - x^\star) < 0.$

To simplify the following analysis, we will use the vectorized form the objective in (6.16). Let $\text{vecr} : \mathbb{R}^{n\times m} \to \mathbb{R}^{nm}$ be the row-order vectorization operator that stacks matrix rows into a single column. Let $u := \text{vecr}(U)$, then the gradient of $F(U)$ in (6.16) can be expressed as the vectorization

$$\nabla F_{\text{v}}(u) := 2(\bar{S} \otimes I)u + 2d(I \otimes P_o + P_d \otimes I)u, \tag{6.22}$$

where $\otimes$ denotes the Kronecker product.

Lemma 6.3.4. *For $d \geq m+1$, if a stationary point $U^\star \in \Delta^{m\times m}$ is non-binary, then it is a strict saddle point.*

Proof. Assume $U^\star$ has non-binary entries. Because $\sum_j U^\star_{ij} = 1\ \forall_i$, there must be at least two non-zero elements a and b with $0 < a \leq b < 1$ in a single row. Without loss of generality, assume that these two non-zero, non-binary elements occur in the

first two columns of the first row of $U^\star$. Let $u^\star := \mathrm{vecr}(U^\star)$, thus $u^\star = [a, b, \dots]^\top$. Take $y = [a - \epsilon, b + \epsilon, \dots]^\top$ with $\epsilon > 0$ so that $0 \le a - \epsilon < b + \epsilon \le 1$ and let $v := y - u^\star = [-\epsilon, \epsilon, 0, \dots, 0]^\top$.

From Lemma 6.3.3, we know that the entries of $\nabla F_{\mathrm{v}}(u^\star)$ corresponding to a and b are equal; denote their value as c so that $\nabla F_{\mathrm{v}}(u^\star) = [c, c, \dots]^\top$. Therefore

$$\nabla F_{\mathrm{v}}(u^\star)^\top v = -c\epsilon + c\epsilon = 0,$$

so there exists a y suitable for condition (ii) of Proposition 6.3.1.

The curvature of F_{v} in the direction of v is given by

$$v^\top \nabla^2 F_{\mathrm{v}}(u^\star)\, v = 2\, v^\top \left[\bar{S} \otimes I + d(I \otimes P_o + P_d \otimes I)\right] v.$$

Thus, to satisfy condition (ii) of Proposition 6.3.1, we must have

$$d\, v^\top (I \otimes P_o + P_d \otimes I) v < -v^\top (\bar{S} \otimes I) v, \tag{6.23}$$

We will treat each side separately. Let v_1 denote the first row of $U^\star$. Then, the right side can be simplified as

$$\begin{aligned} &- v^\top (\bar{S} \otimes I) v \\ &\quad = - \begin{bmatrix} v_1 \\ \hdashline 0 \\ \hdashline \vdots \\ \hdashline 0 \end{bmatrix}^\top \begin{bmatrix} \bar{S}_{11} I & \cdots & \bar{S}_{1m} I \\ \vdots & \ddots & \vdots \\ \bar{S}_{m1} I & \cdots & \bar{S}_{mm} I \end{bmatrix} \begin{bmatrix} v_1 \\ \hdashline 0 \\ \hdashline \vdots \\ \hdashline 0 \end{bmatrix} \\ &\quad = -\bar{S}_{11} \|v_1\|^2. \end{aligned} \tag{6.24}$$

The left side can be simplified as

$$
\begin{aligned}
& d\, v^\top (I \otimes P_o + P_d \otimes I) v \\
& \quad = d \begin{bmatrix} v_1 \\ \hdashline 0 \\ \hdashline \vdots \\ \hdashline 0 \end{bmatrix}^\top \begin{bmatrix} P_o & 2I & 2I & \cdots \\ 2I & P_o & 2I & \cdots \\ 2I & 2I & P_o & \\ \vdots & \vdots & & \ddots \end{bmatrix} \begin{bmatrix} v_1 \\ \hdashline 0 \\ \hdashline \vdots \\ \hdashline 0 \end{bmatrix} \\
& \quad = d\, v_1^\top P_o\, v_1 = d \begin{bmatrix} -\epsilon \\ \epsilon \\ 0 \\ \vdots \end{bmatrix}^\top \begin{bmatrix} \epsilon \\ -\epsilon \\ 0 \\ \vdots \end{bmatrix} \\
& \quad = -d\|v_1\|^2. \qquad (6.25)
\end{aligned}
$$

Therefore, condition (6.23) is simplified as

$$
-d\|v_1\|^2 < -\bar{S}_{11}\|v_1\|^2 \quad \Longrightarrow \quad d > \bar{S}_{11}.
$$

Recall that $|\bar{S}_{ij}| \leq 1$ and $d \geq m+1$ by assumption. Hence,

$$
d \geq m+1 > 1 \geq |\bar{S}_{11}|,
$$

showing that condition (ii) of Proposition 6.3.1 is satisfied. □

Lemma 6.3.4 gives us the ability to detect if Algorithm 3 has converged to a strict saddle point (or local maxima). If it has, then a second-order stationary point may be found by escaping the saddle point. This is guaranteed using the generic framework proposed in [212, Alg. 1], wherein a feasible search direction $v := y - u^\star$ is found by

solving

$$\begin{aligned} \underset{y \in \Delta^{m\times m}}{\text{minimize}} \quad & q(y; u^\star) := (y - u^\star)^\top \nabla^2 F_{\text{v}}(u^\star)(y - u^\star) \\ \text{subject to} \quad & \nabla F_{\text{v}}(u^\star)^\top (y - u^\star) = 0 \end{aligned} . \tag{6.26}$$

If $q(y; u^\star) < 0$ is found, then $u^\star$ is a strict saddle point and F will be decreased by taking a step in the direction of v; otherwise $u^\star$ is already at a feasible local minimum.

Lemma 6.3.5. *For $d \geq m + 1$, if a stationary point $U^\star \in \Delta^{m\times m}$ is binary, then $\phi_{\text{dist}}(U^\star) = 0$, i.e., distinctness is satisfied.*

Proof. Without loss of generality, consider a single view, i.e., $n = 1$ and $m = m_1$. Therefore $(P_d)_{ij} = 2(1 - \delta_{ij})$, where $\delta_{ij} = 1$ for $i = j$ and 0 otherwise.

By contradiction, suppose $U^\star \in \Delta^{m\times m}$ is a binary stationary point of (6.16), but distinctness is violated. In particular and without loss of generality, assume that the k-th column of $U^\star$, denoted $U^\star_{:k}$, is such that ${U^\star_{:k}}^\top \mathbf{1}_m = 1 + \rho > 1$, where ρ is the number of excess entries in violation of distinctness (e.g., if there are 3 ones in a column then $\rho = 2$).

By stationarity (6.18a),

$$2\bar{S}U^\star + 2d\big(U^\star P_o + P_d U^\star\big) - Y^\star - \lambda^\star \mathbf{1}_m^\top = 0. \tag{6.27}$$

Considering the first row of (6.27), by definition of matrix multiplication we have the terms

$$2(\bar{S}U^\star)_{1j} = 2\sum_{i=1}^{m} \bar{S}_{1i} U^\star_{ij} = 2\delta_{kj} \sum_{i=1}^{m} \bar{S}_{1i},$$

$$\begin{aligned} 2d(U^\star P_o)_{1j} &= 2d \sum_{i=1}^{m} U_{1i}(P_o)_{ij} = 2d\, U_{1k}(P_o)_{kj} \\ &= 2d(1 - \delta_{kj}), \end{aligned}$$

$$
\begin{aligned}
2d(P_d U^\star)_{1j} &= 2d \sum_{i=1}^{m} (P_d)_{1i} U_{ij} = 4d \sum_{i=1}^{m} (1 - \delta_{1i}) U_{ij} \\
&= 4d \sum_{i=2}^{m} (1 - \delta_{1i}) U_{ij} = 4d \sum_{i=2}^{m} U_{ij} \\
&= 4d\rho\delta_{kj},
\end{aligned}
$$

so that the j-th entry in the first row of (6.27) can be written

$$
2\delta_{kj} \sum_{i=1}^{m} \bar{S}_{1i} + 2d(1 - \delta_{kj}) + 4d\rho\delta_{kj} = Y_{1j}^\star + \lambda_1^\star. \tag{6.28}
$$

Since $U_{1k}^\star = 1$ by assumption, $Y_{1k}^\star = 0$ by complementarity (6.18e). Therefore, solving (6.28) for $\lambda_1^\star$ when $j = k$ yields

$$
\lambda_1^\star = 2 \sum_{i=1}^{m} \bar{S}_{1i} + 4d\rho.
$$

Then, when $j \neq k$, we can use $\lambda_1^\star$ to solve for $Y_{1j}^\star$ as

$$
Y_{1j}^\star = 2d - 4d\rho - 2 \sum_{i=1}^{m} \bar{S}_{1i}.
$$

By dual feasibility (6.18d), $Y_{1j}^\star \geq 0$ so

$$
d(1 - 2\rho) \geq \sum_{i=1}^{m} \bar{S}_{1i} \geq -m
$$

would guarantee dual feasibility, where the lower bound $-m$ is due to $|\bar{S}_{ij}| \leq 1$. By assumption, $d \geq m + 1$, and assuming the worst case where all entries violate distinctness (i.e., $\rho = m - 1$) gives the tightest bounds

$$
d(1 - 2\rho) \geq (m + 1)(1 - 2\rho) \geq 2(1 - m^2) \geq -m,
$$

implying that $-2m^2 + m + 2 \geq 0$ for dual feasibility (6.18d) to hold, which is a contradiction for $m > 1$. □

Given the characterization of non-binary and binary stationary points in Lemma 6.3.4 and Lemma 6.3.5, we now present our main result concerning (6.15) (which was restated as (6.16)).

Theorem 6.3.1. *For $d \geq m+1$, solutions $U^\star$ of problem (6.15) are feasible solutions of problem (6.14). In particular, $U^\star$ is cycle consistent, distinct, and binary.*

Proof. By direct consequence of Lemma 6.3.4, when $d \geq m+1$, non-binary solutions are strict saddles. In this case, by solving (6.26) strict saddles can be escaped, ensuring convergence to a second-order stationary point $U^\star$, i.e., a (local) minima [212, Theorem 4]. By the contrapositive of Lemma 6.3.4, it follows that $U^\star$ is binary, and therefore must be distinct due to Lemma 6.3.5. Since $U^\star$ is cycle consistent by construction [56], the proof is complete. □

6.3.5 MIXER Algorithm

We approach solving (6.15) by gradually increasing the scalar parameter $d \geq 0$. The rationale is that the affinity S is indicative of the true solution and so an initially small d allows U to be biased towards a good solution, while a large d ultimately pushes U towards a feasible solution. According to Theorem 6.3.1, the output of MIXER is a cycle-consistent, distinct, and binary second-order (local) minimum of (6.14).

MIXER is summarized in Algorithm 3. The solution is initialized using the eigenvectors of the modified affinity matrix, $\mathbf{1} - 2S$ (Line 3). The penalty weight d is first initialized (Line 4) to a value that causes approximately half of the elements of U that violate distinctness or orthogonality to diminish towards zero in the first step (see Section 6.4.2); note that $[\cdot]$ refers to an element-wise operation. For each value of d, we solve (6.15) using projected gradient descent (PGD) with the projection Π_Δ onto the convex constraint set. The projection operation can be efficiently implemented as a non-iterative algorithm [214]. PGD over convex constraints is guaranteed to converge to first-order stationary points [212, Prop. 7] and the `DetectAndEscapeSaddle` framework of [212, Alg. 1] ensures convergence to second-order stationary points, i.e., feasible local minima (cf. Theorem 6.3.1). Note that in practice, we have found that

Algorithm 3 MIXER

1: **Input** affinity matrix $S \in [0,1]^{m\times m}$, set cardinalities $m_i \in \mathbb{N}^n$ (if distinctness is required)
2: **Output** $U \in \{0,1\}^{m\times m}$
3: $U \leftarrow \Pi_\Delta(\text{eigvec}(\mathbf{1}-2S))$ % initialize using eigenvectors
4: $d \leftarrow \text{med}\{-\frac{[\mathbf{1}-2S]}{[UP_o+P_dU]} : [UP_o + P_dU] > 0, [U] > 0\}$
5: **while** $d < m+1$ **do**
6: **while** U not converged **do**
7: $\nabla F = 2(\mathbf{1}-2S)U + 2d\big(UP_o + P_dU\big)$
8: $U \leftarrow \Pi_\Delta(U - \alpha\nabla F)$ % α via backtracking line search
9: $U \leftarrow$ `DetectAndEscapeSaddle`(U)
10: $d \leftarrow 2d$
11: **if** $\phi_{\text{orth}}(U) = 0$ and $\phi_{\text{dist}}(U) = 0$ **then return**

adding a small (10^{-1}) perturbation to P_o, P_d is efficient to implement and avoids non-binary saddles altogether. This is because non-binary saddles have strictly negative curvature (cf. Lemma 6.3.4) so moving in random directions is likely to cause objective minimization. Additionally, we frequently observed that Algorithm 3 converges to feasible solutions of (6.14) with a much smaller d than required by Theorem 6.3.1, and so early termination (Line 11) regularly occurs.

Computational complexity. Neglecting constant factors and replacing the `DetectAndEscapeSaddle` routine with small gradient perturbations as described above, the worst-case complexity of Algorithm 1 is bounded by $\mathcal{O}(m^3)$ per iteration, corresponding to matrix multiplications. We observed that most of the runtime is spent computing the matrix-vector products in ∇F and F and by the $\mathcal{O}(m^2 \log m)$ projection onto the matrix simplex, Π_Δ. A numerical analysis of runtime for different size problems is given in Section 6.4.3.

6.4 Results

We evaluate MIXER in three experiments. First, we use synthetic data to validate that our MIQP problem formulation (6.4) enables high accuracy. We find that MIXER achieves near-optimal performance compared to several existing state-of-the-art multiway matching algorithms. Second, we compare MIXER with other algorithms using standard multiway image feature matching benchmarks. Finally, we

apply multiway fusion to a challenging robotics dataset collected as part of this work, highlighting MIXER's ability to use multiple attributes to improve the overall fusion accuracy.

We report matching accuracy in terms of precision, recall, and the F_1 score. Precision p is the number of correct associations divided by the total number of associations identified, recall r is the number of correct associations identified divided by the total number of associations in the ground truth, and the F_1 score is defined as $\frac{2pr}{p+r} \in [0, 1]$, which captures the balance between precision and recall.

We compare against several state-of-the-art multiway matching algorithms. Spectral [46] and MatchEig [60] are based on spectral relaxation, CLEAR (see Chapter 5) and QuickMatch [57] are based on graph clustering, FCC [65] is based on cluster-consistency statistics, NMFSync [62] is based on matrix factorization, MatchALS [56] is based on low-rank matrix recovery, and MatchLift [59] is based on convex relaxation. Except for MIXER, QuickMatch and FCC, algorithms require a universe size estimate and we use CLEAR's estimate as the typical spectral approach is typically sensitive to noisy affinity matrices [56, 60, 73]. For MatchALS, we scale this estimate by 2 as suggested by the authors. As a baseline, we perform pairwise data association using Hungarian with a minimum score threshold (0.35), as commonly done in the tracking literature [209], demonstrating the type of noisy, cycle inconsistent results that arise when naively fusing a set of pairwise associations. For algorithms that do not guarantee cycle consistency (baseline, FCC, MatchEig, MatchALS), we post-process their solutions by extracting connected components and completing them into cliques (cf. [73, §VII]). This enforces cycle consistency, but exposes distinctness violations—thus highlighting the importance of simultaneously satisfying both constraints during solution search. These completed results are denoted with a "(C)". For small problems, we globally solve the MIQP (6.4) to optimality using Gurobi 9.5.2, referred to as MIXER*. All algorithms are implemented in MATLAB and executed on an i7-6700 CPU with 32 GB RAM.

6.4.1 Synthetic Dataset

We use Monte Carlo analysis with synthetic data to compare MIXER with several state-of-the-art algorithms across different noise regimes. We aim to 1) validate our formulation (6.4) for high-accuracy multiway fusion, 2) show the ability of MIXER to obtain near-optimal and thus high-accuracy solutions, 3) show that early fusion with MIXER achieves higher accuracy than late fusion with existing methods, and 4) show that MIXER provides a computationally scalable approach to approximately solve (6.4). Noisy pairwise associations are synthetically generated by considering $n = 10$ partial views of $k = 30$ objects, e.g., 10 images each containing at most 30 objects. Twenty-five different noise regimes are considered, wherein the percent mismatch (i.e., binary noise added to ground truth associations) and the probability of observation (i.e., selection of random subsets of the universe) are varied. A noisy association $a_{kl}^{ij} \in \{0, 1\}$ of objects k and l from views i and j is then made into an affinity $s_{kl}^{ij} \in [0, 1]$ by adding uncertainty according to $s(a; \theta) = (1 - \theta)a + 0.5\theta$, where θ is drawn from a standard uniform distribution. Each algorithm processes these noisy pairwise affinities, allowing us to study the accuracy of each method as an early fusion approach.

Fig. 6-3 shows average results over 10 Monte Carlo trials. While all algorithms generally perform well in low noise regimes with perfect observability (top left squares of Fig. 6-3), the performance of most algorithms quickly deteriorates as the probability of observing objects decreases. MIXER is the exception, performing better than other approaches on average, over the entire range of mismatch and partial observability. The low observation redundancy regimes (bottom rows in Fig. 6-3) are especially important because these settings occur when limited field-of-view sensors only see portions of the universe; for example, in multirobot SLAM [20, 215].

To assess how well our formulation (6.4) addresses the multiway fusion problem, we compute the optimality gap of each algorithm's solutions compared to MIXER*. Fig. 6-4 plots the accuracy of each solution against the optimality gap (percent change of objective value with respect to MIXER* solution), exposing the correlation be-

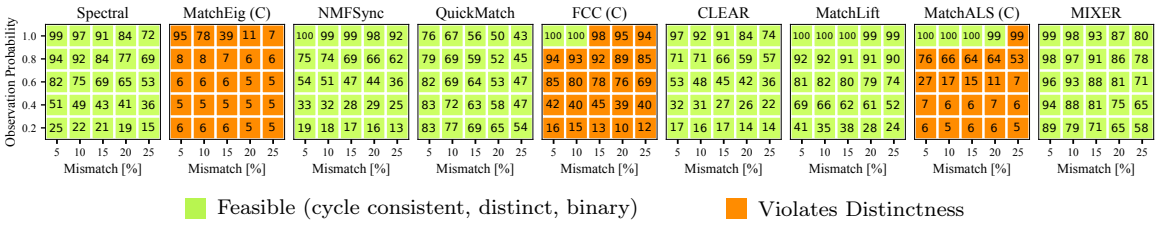


Figure 6-3. Multiway early fusion of noisy pairwise affinities generated from $n = 10$ simulated views of a universe with $k = 30$ distinct objects. MIXER accurately fuses pairwise affinities while enforcing cycle consistency and distinctness, performing better than the state-of-the-art over the entire range of noise regimes. Especially apparent is the high accuracy of MIXER compared to existing methods in low observation redundancy settings (bottom rows).

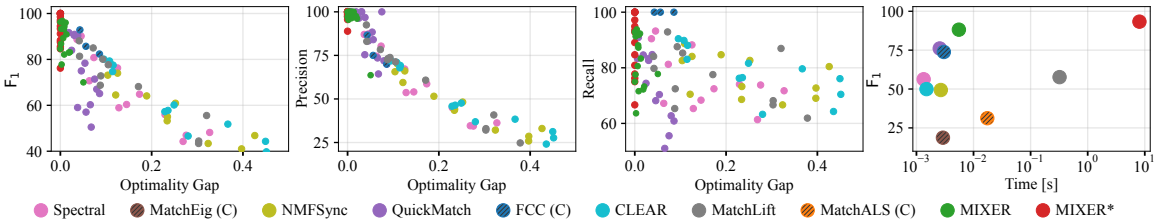


Figure 6-4. F_1, precision, and recall compared with optimality gaps relative to our MIQP formulation (6.4). Each point corresponds to a solution of a synthetic problem having some observation probability and mismatch percentage (e.g., see Fig. 6-3). As the optimality gap of a solution decreases, the accuracy increases, validating our multiway fusion formulation. MIXER solutions are tightly grouped in the high-accuracy, near-optimal regime. The last plot visualizes each algorithm's average accuracy vs runtime, showing that MIXER is much more scalable than directly solving the MIQP.

tween low optimality gap and high accuracy (F_1 and precision). MIXER solutions are tightly grouped in this near-optimal, high-accuracy regime, showing that our algorithm, although a local, first-order method, is frequently able to achieve good results. Interestingly, we observe in the recall results of Fig. 6-4 that our formulation leads MIXER to be conservative—it would rather *not* fuse objects if precision would be sacrificed. This property allows further improvements to be made as more data is collected.

Finally, Fig. 6-5 highlights that early fusion with MIXER outperforms both early and late fusion of uncertain affinities compared to existing state-of-the-art algorithms. In late fusion, pairwise affinities are first processed into hard matches before multiway matching is performed, and in this setting, MIXER does not perform as well. The principal reason is because MIXER is a multiway MWM formulation as discussed in Remark 6.3.1—when pairwise associations are first made, a large amount of infor-

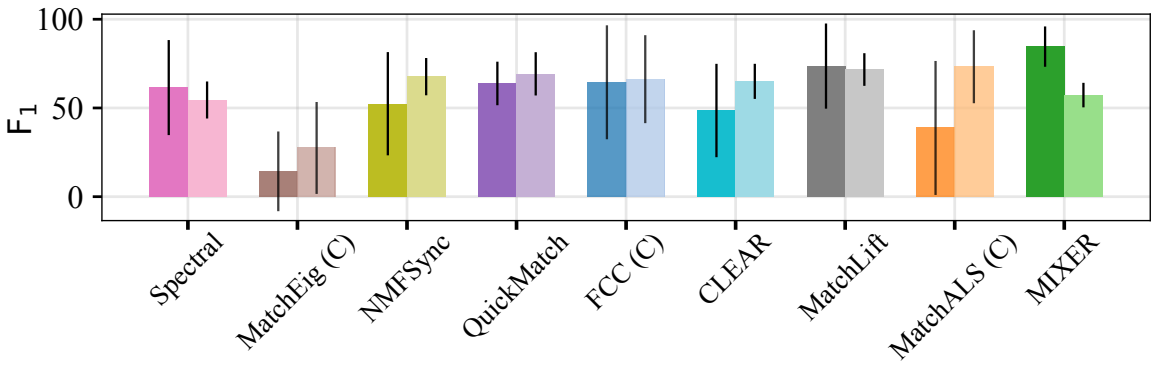


Figure 6-5. Average F_1 results using early fusion (left bars) and late fusion (right bars). Note that MatchEig (C) and MatchALS (C) violate distinctness constraints and so these infeasible results are indicated with hatch marks.

mation is discarded, precluding MIXER from using it in combination with multiway constraints.

6.4.2 Update Rule Analysis

We must increment the non-negative parameter d until $U \in \Delta^{m \times m}$ is binary and satisfies all the constraints of (6.14). Our strategy for incrementing d is motivated by the desire to quickly converge to a feasible solution. Focusing on the elements of U that contribute to the violation of distinctness and orthogonality allows us to do so. In what follows we refer to these two constraints together simply as the penalty $\Phi(U) := \phi_{\text{orth}}(U) + \phi_{\text{dist}}(U)$, with $\nabla\Phi = UP_o + P_dU$.

Observe that $\nabla\Phi$ is always non-negative and that non-zero entries $\nabla\Phi_{ij}$ indicate the entries of U that if increased would incur more penalty—these are the *potentially* problematic entries of U. Note that all the entries of U are in $[0, 1]$ and only the non-zero entries could contribute to distinctness or orthogonality violations. Thus, to precisely identify the problematic entries, we find entries satisfying $\nabla\Phi_{ij} > 0, U_{ij} > 0$. For each of those problematic entries, we solve for the value of d that would cause $\nabla F_{ij} \geq 0$ so that the step $-\nabla F$ causes U_{ij} to diminish. The set of values that have this property for problematic entries is (i.e., set $\nabla F = 0$ and solve for d)

$$\mathcal{D} = \left\{ -\frac{[\mathbf{1} - 2S]}{[\nabla\Phi]} : [\nabla\Phi] > 0, [U] > 0 \right\}, \tag{6.29}$$

where the notation $[\cdot]$ indicates an element-wise operation. The number of entries diminished to zero is controlled by taking the maximum, median, or minimum of this set.

We analyze ten different update rules for d in Table 6.2. The value of d chosen at each outer iteration is given by the sequence under the "Update Rule" column. Our study indicates that the best tradeoff between a low number of outer iterations and high accuracy can be achieved by first selecting $\operatorname{med} \mathcal{D}$ and then doubling the value of d each subsequent outer iteration. This method takes a principled approach to selecting the first penalty value based on the problem data and initial guess, followed by doubling the penalty weight to quickly lead to convergence to a feasible, binary solution. Note that initializing d too large leads to abysmal accuracy.

Table 6.2. Analysis of update rules for d (Line 10) in Algorithm 3. Ten rules are evaluated on the ten datasets of Table 6.3 and Table 6.4. We compare the average number of outer iterations necessary until convergence and the average solution accuracy. F_1 score is normalized according to the maximum F_1 across methods within a given dataset. The strategy that achieves the highest accuracy begins with $d = 0$ and slowly increments d by 0.1 each outer iteration, but requires more than 50 outer iterations on average. A better tradeoff between high accuracy and low iteration count can be found by first selecting $\operatorname{med} \mathcal{D}$ and then doubling the value of d each subsequent outer iteration.

Update Rule	Num. Outer Iters.	Normalized F_1
$d_i = 0, \max \mathcal{D}, 2d_{i-1}, \ldots$	2.9 ± 1.2	0.927 ± 0.050
$d_i = 0, \operatorname{med} \mathcal{D}, 2d_{i-1}, \ldots$	3.7 ± 0.5	0.938 ± 0.050
$d_i = 0, \min \mathcal{D}, 2d_{i-1}, \ldots$	6.1 ± 1.1	0.942 ± 0.040
$d_i = \max \mathcal{D}, 2d_{i-1}, \ldots$	2.5 ± 1.3	0.917 ± 0.070
$\boldsymbol{d_i = \operatorname{med} \mathcal{D}, 2d_{i-1}, \ldots}$	$\mathbf{4.8 \pm 0.9}$	$\mathbf{0.980 \pm 0.030}$
$d_i = \min \mathcal{D}, 2d_{i-1}, \ldots$	15.6 ± 1.2	0.936 ± 0.050
$d_i = 0, 0.1, 2d_{i-1}, \ldots$	8.0 ± 0.8	0.950 ± 0.040
$d_i = 0, 0.1, 0.2, \ldots$	52.5 ± 28.4	0.988 ± 0.030
$d_i = 1, 2d_{i-1}, \ldots$	3.6 ± 2.2	0.944 ± 0.050
$d_i = 10, 2d_{i-1}, \ldots$	1.0 ± 0.0	0.238 ± 0.190

6.4.3 Timing Analysis

We compare the runtime of Algorithm 3 to existing algorithms. Runtime complexity is driven by the problem size m, which is a function of the number of world objects k, the number of views n, and the probability p of a given view observing all k objects. Therefore, we generating synthetic data with nominal values of $k = 30$, $n = 10$, and

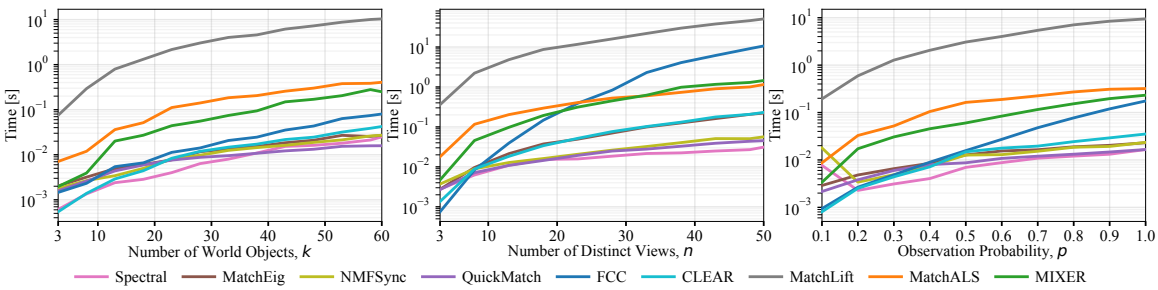


Figure 6-6. Runtime comparison when operating on synthetic data with various problem sizes. Problem size is a function of the number of world objects k, the number of views n, and the probability p of a given view observing all k objects. Nominal values for these parameters are $k = 30$, $n = 10$, and $p = 0.5$. Each subplot is generated by varying one of the parameters while holding the other two constant. In general, MIXER achieves the highest accuracy (see Section 6.4) while being faster than its closest competitors MatchLift and MatchALS (see Table 6.1). Although other algorithms are faster than MIXER, they frequently fail to return accurate solutions in realistic settings with a moderate number of world objects and low observation probability.

$p = 0.5$ and produce three plots in Fig. 6-6 corresponding to varying one parameter while holding the other two at their nominal values. The nominal values are chosen for consistency with the synthetic experiments in Section 6.4.1, with results in Fig. 6-3 and Fig. 6-4. For all three plots in Fig. 6-6, we use a mismatch of 25% and the runtime at each parameter setting is averaged over 10 trials.

Fig. 6-6 shows that in general the fastest algorithm is Spectral while the slowest is MatchLift. MIXER is faster than its most similar competitors, MatchLift and MatchALS (see Table 6.1). Indeed, these competitors are the closest in terms of accuracy (e.g., Table 6.4), but are considerably slower than MIXER. The remaining algorithms (Spectral, MatchEig, NMFSync, CLEAR) perform well in terms of runtime and obtain similar execution speeds to each other. Although these algorithms are faster than MIXER, previous results (Figs. 6-3 and 6-4, Tables 6.3 and 6.4) indicate that they are not able to achieve the high accuracy that MIXER does. The FCC algorithm varies in runtime the most, with the largest change in runtime occurring when a large number of distinct views (i.e., large n) exists in the dataset. In realistic scenarios with a moderate amount of world objects (e.g., using local submaps to reduce the number of objects needed to be processed), there are frequently low observation probabilities as robot sensors are noisy and field-of-view-limited so that only a subset of the world objects are seen in each view. In this scenario (e.g., Section 6.4.5), our unoptimized MATLAB implementation of MIXER achieves the best

accuracy at runtimes on the order of a few hundred milliseconds.

6.4.4 Benchmark Datasets

We use the CMU Hotel[1] and Affine Covariant Features[2] benchmark datasets to demonstrate multiway fusion with MIXER on real data and report results in Table 6.3. These datasets consist of collections of images from different perspectives and the goal is to extract features from each image, perform pairwise feature matching, and then perform multiway fusion of features. Instead of standard pairwise SIFT matching, which would cause late fusion since it makes hard decisions about pairwise association and thus limits the ability of multiway fusion to make fully-informed decisions, we instead construct pairwise affinity matrices using k-nearest neighbors. Pairwise affinities are created by finding the 10 closest SIFT matches (ℓ_1 distance) for each keypoint. The affinity score of the closest neighbor is set to 1, the other neighbors set to 0.5, and the rest to 0. This makes use of all three states of association and indicates that for the semi-close neighbors, there is some uncertainty as to whether or not they should be matched, while for far away neighbors they are very likely to be incorrect matches.

6.4.5 Car Fusion Dataset

In this section, we evaluate MIXER's ability to fuse observation of objects seen from multiple views. Multiway object fusion is a task that appears in settings like multirobot SLAM [20, 215] and multimodal object tracking [216]. We collect data by teleoperating a Clearpath Jackal equipped with an RGB camera around a parking lot as shown in Fig. 6-7. We select $n = 184$ images from two separate traversals to create large view changes and thus increase the difficulty. In total, there are $k = 22$ distinct cars visible from 339 car detections, resulting in only 8% of the universe being observed in each image. Using these RGB images, the objective is to fuse cars seen from different views using noisy, partial, and cropped detections (i.e., not every

[1] http://pages.cs.wisc.edu/~pachauri/perm-sync/
[2] https://www.robots.ox.ac.uk/~vgg/data/affine

Table 6.3. Benchmark results on CMU Hotel and Affine Covariant Features datasets. F_1 scores are reported as percentages. Results that violate cycle consistency are indicated in red and are completed to asses their true fusion accuracy. When fusing cycle inconsistent solutions, distinctness is often violated and these results are indicated in orange. In every case, MatchLift is at least 1 order of magnitude slower than MIXER (see Fig. 6-4).

Algorithm	Hotel	Wall	UBC	Bikes	Leuven	Trees	Graffiti	Bark	Boat
Baseline	89.2	49.0	71.1	52.4	68.9	44.0	40.9	39.4	47.6
Baseline (C)	16.8	2.2	5.9	1.4	4.9	1.5	0.3	10.7	0.9
Spectral	65.3	43.8	51.5	48.5	58.5	46.2	34.1	31.4	26.8
MatchEig	94.7	78.4	84.4	75.7	78.9	50.4	48.0	37.5	61.2
MatchEig (C)	27.2	16.1	10.8	1.9	4.1	1.6	0.3	4.0	1.1
NMFSync	89.7	39.4	63.2	58.4	59.5	33.5	29.3	18.9	44.1
QuickMatch	82.4	46.4	65.7	50.9	62.7	42.3	39.5	28.2	43.4
FCC	92.5	55.7	77.4	59.0	68.8	51.7	40.4	42.1	32.3
FCC (C)	92.8	46.3	70.8	52.8	58.5	41.0	32.4	30.2	29.9
CLEAR	76.9	40.9	47.8	40.3	53.1	26.1	28.1	25.8	38.9
MatchALS	95.4	71.4	77.1	71.1	73.9	50.8	46.7	38.4	58.7
MatchALS (C)	89.3	43.1	26.4	39.2	39.2	21.9	11.1	26.9	35.0
MatchLift	95.9	55.9	**79.2**	63.8	71.9	48.2	**45.4**	35.0	54.4
MIXER	**97.0**	**56.4**	77.7	**67.5**	**74.2**	**51.0**	43.3	**40.0**	**59.2**

car is seen in each frame, and some cars extend out of frame). Cars are detected using YOLOv3 [217] and we extract three attributes from each detection: bounding box, car color, and visual appearance. These three attributes have complementary strengths. For example, bounding boxes are good for scoring consecutive detections, but cannot be used over large time offsets; car color is a sensitive quantity to extract, but is good for associating detections over large time offsets; visual appearance based on the number of matching SIFT descriptors is typically robust to small/medium view changes, but breaks down with large view changes. Thus, using these attributes highlights the three-states-of-association feature of the formulation of problem (6.4), which allows MIXER to achieve high accuracy. Ground truth associations are generated by manually annotating the detections.

Each attribute creates its own set of pairwise affinities, which will be combined to create a multiway affinity matrix. Bounding box affinity is scored using intersection-over-union (IoU) for consecutive detections, while non-consecutive detections take a value of 0.5 as wide-baseline IoU scoring is inconclusive. Color affinity scores take either 0, 0.5, or 1, with 0.5 being used if either car's color could not be clearly as-

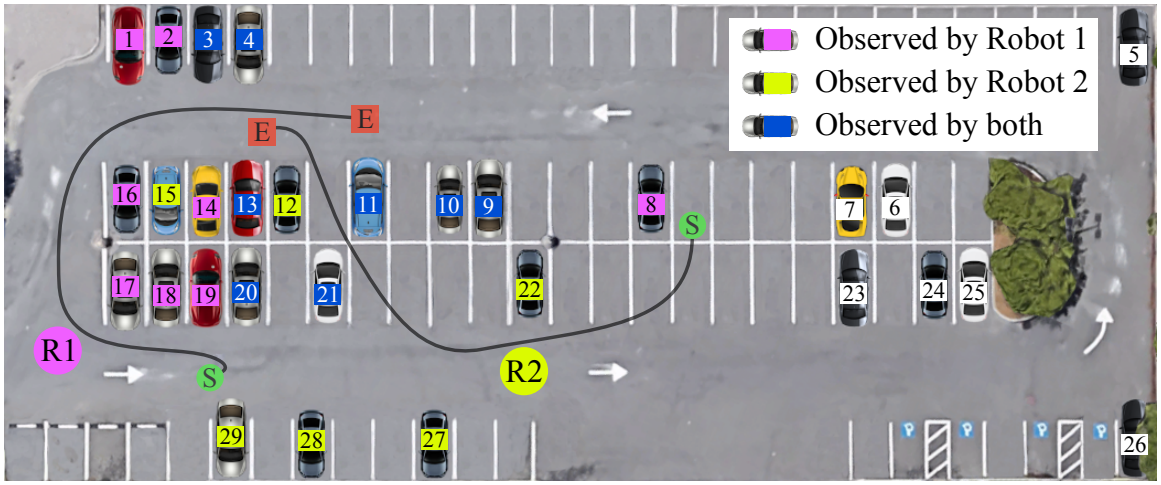


Figure 6-7. Illustration of the parking lot dataset, with the paths of two robots (R1 and R2), from start (S) to end (E). The robots observed $k = 22$ cars, with 8 cars being covisible. Colors indicate which robot saw which car.

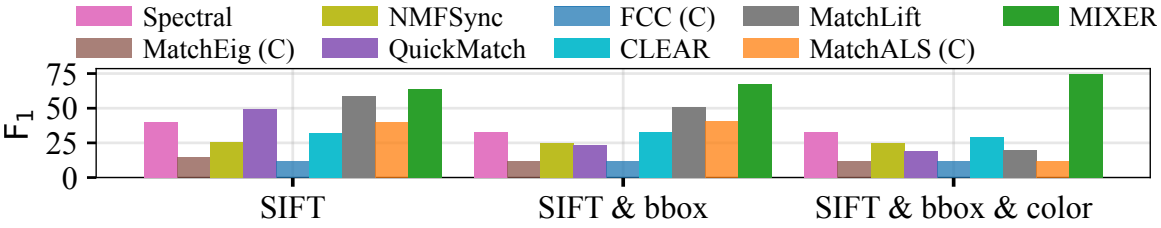


Figure 6-8. Multiway car fusion results, incrementally adding additional attribute affinities. Mixing together attributes allows MIXER to increase in F_1 score by more than 10%. In contrast, the combination of additional uncertain information causes other methods to decrease in F_1 accuracy.

certained. Visual similarity is scored based on the number of SIFT matches between two cars. Common attribute affinity combination/fusion methods include (weighted) averaging, non-maximum suppression, probabilistic ensembling [218], multi-layer perceptrons [216], graph neural networks [219], or other learned fusion processes [220]. For simplicity, we adopt the weighted averaging approach, using a weights of 1, 0.5, and 1 for bounding box overlap, color similarity, and appearance similarity, respectively. Color receives less weight due to its lack of robustness.

We also report results for $\text{MatchALS}_{\alpha=0.5}$ and $\text{MatchLift}_{\lambda=0.5}$ with parameters set so that the input affinity matrix is transformed to $\mathbf{1} - 2S$ like in MIXER (see Table 6.1). In MatchALS and MatchLift, these parameters—and therefore the coefficient scaling on S—are heuristically introduced to encourage sparsity, with default values suggested by their authors of $\alpha = 0.1$ and $\lambda = \sqrt{\binom{n}{2}/(2n)} \approx 0.3$, respec-

Table 6.4. Multiway car fusion results. Objective values are listed except for solutions that violate cycle consistency (red) or distinctness (orange).

Algorithm	Precision	Recall	F_1	Obj. (↓)	Time [ms]
Baseline	44.9	14.8	22.3	–	1648
Baseline (C)	8.2	84.5	14.9	–	1650
Spectral	22.8	58.0	32.7	146.9	**12**
MatchEig	13.0	72.8	22.1	–	1079
MatchEig (C)	6.3	99.7	11.8	–	1083
NMFSync	15.7	58.6	24.7	151.5	35
QuickMatch	13.8	31.2	19.2	148.9	37
FCC	47.4	70.8	56.8	–	3269
FCC (C)	6.3	98.1	11.9	–	3272
CLEAR	19.0	66.4	29.6	150.6	63
MatchALS	7.5	94.9	13.9	–	542
MatchALS (C)	6.3	99.9	11.8	–	545
MatchLift	12.4	48.8	20.1	149.7	15 463
$\text{MatchALS}_{\alpha=0.5}$	41.9	76.7	54.2	–	614
$\text{MatchALS}_{\alpha=0.5}$ (C)	8.9	87.6	16.1	–	617
$\text{MatchLift}_{\lambda=0.5}$	34.6	54.7	42.4	141.1	15 295
MIXER	**79.4**	**70.6**	**74.8**	**135.7**	312

tively. In contrast, the expression $\mathbf{1} - 2S$ arises naturally in our formulation due to the Frobenius objective (6.4) and our insight is that it allows for three states of association in multiway MWM (see Remark 6.3.1). Setting $\alpha = \lambda = 0.5$ allows for a direct comparison of the relaxations and algorithms listed in Table 6.1 and, as expected due to our insights, they provide an increase in accuracy. However, as reported in Table 6.4, MIXER substantially outperforms existing algorithms. It is followed by $\text{MatchLift}_{\lambda=0.5}$, which lags by 32% in F_1 accuracy and is 49x slower. Although $\text{MatchALS}_{\alpha=0.5}$ and $\text{MatchLift}_{\lambda=0.5}$ also have three states of association, their relaxations require rounding to binary matrices and search for solutions in association space $A \in \mathbb{R}_+^{m \times m}$, leading to sensitivity of the universe size estimate.

MIXER achieves the best objective and F_1 accuracy in Table 6.4. This is consistent with Fig. 6-4, which shows that MIXER attains near-optimal, high-accuracy solutions, even in low-observation-probability settings. The relationship between near-optimal and high accuracy solutions is due to our MIQP formulation (6.4), which allows for three states of association and therefore a way to combine uncertain affinities from multiple attributes, as shown in Fig. 6-8.

6.5 Summary

We presented the MIXER algorithm for multiattribute, multiway fusion of uncertain pairwise affinities. Our MIQP formulation leverages direct access to affinities and allows three modes of association (non-match, undecided, and match) over the range 0 to 1. This feature led to the insight that our formulation is a multiway extension of the maximum-weight matching problem (see Remark 6.3.1). Because of scalability issues of solving MIQPs, we proposed a novel continuous relaxation in a projected gradient descent scheme to converge to feasible, binary solutions of the original problem. The guarantee of our algorithm to converge to binary points sets it apart from related work, which relies on a final binarization step that can lead solutions to be infeasible. Finally, our experimental evaluations in three datasets showed that MIXER frequently achieves higher accuracy than the state of the art, especially in noisy regimes with low observation redundancy. These properties and results establish MIXER as an effective and scalable multiway fusion algorithm in the presence of uncertain affinities.

Chapter 7

Conclusion

7.1 Summary of Contributions

This thesis provides algorithms that enable robust data association for geometric estimation problems that arise in robotic perception. Robust and reliable geometric estimation is a crucial prerequisite for deploying autonomous systems, and the difficulty of robust estimation is largely due to the data association process, which operates on noisy, imperfect measurements. To combat the ill-effects of uncertain sensing on data association, this thesis asserts that measurement consistency provides a powerful means of robustly identifying correspondences, which ultimately leads to more successful geometric estimation. This conclusion was reached by investigating the following questions: (i) How to robustly identify pairwise correspondences in the presence of a large number of incorrect options? (ii) How to best represent data from modern sensors (e.g., 3D lidar, RGB-D) for reliable and efficient data association? (iii) How to ensure consistent data association across multiple observations? (iv) How to utilize uncertain affinities in consistent, multiway data association?

Chapter 3 presented a graph-theoretic algorithm for robust pairwise data association. The new approach, called CLIPPER (Consistent LInking, Pruning, and Pairwise Error Rectification) leverages the notion of geometric consistency to transform the pairwise assignment problem into a search for the largest set of mutually consistent correspondences. Compared with existing consistency-graph approaches, CLIPPER

does not threshold the naturally weighted graph representation of consistency into an unweighted graph, thus is able to exploit the most information. Further, CLIPPER enjoys scalability due to the first-order local search method it employs, which empirically is shown to have a good basin of attraction with respect to the global solution. In the experiments of chapter 3, CLIPPER consistently ran with low runtime and was able to outperform the state of the art in data association precision/recall and in overall estimation accuracy.

Chapter 4 presented GraffMatch, a global method for matching 3D lines and planes from two landmark sets and subsequently estimating the relative rotation and translation. The main novelty of GraffMatch is in the representation of lines and planes as elements of the affine Grassmannian manifold. By representing these geometric landmarks in this natural way (e.g., as affine subspaces), it is possible to leverage the Grassmannian metric to calculate the distance between two landmarks. We prove that the distance between two affine Grassmannian elements is invariant to both rotation and translation if a shift operation is performed before applying the metric. This invariance property enables the use of the efficient and robust graph-theoretic data association method developed in Chapter 3. As a result, no initial alignment guess is required for GraffMatch, allowing registration in settings where landmark sets have a large displacement or relative rotation due to observing the landmarks from two different locations.

Chapter 5 addressed computational complexity and accuracy challenges in multiway data association by presenting the CLEAR algorithm. CLEAR leverages the natural graphical representation of the multiway association problem and uses a spectral graph clustering technique that is uniquely tailored to solve this problem in a computationally efficient manner. An important characteristic of the CLEAR algorithm is that its solutions are cycle consistent, an important property in so-called clique-centric applications that can help avoid catastrophic merging of incorrectly associated objects. Empirical results based on extensive synthetic and experimental evaluations demonstrated that CLEAR outperforms state-of-the-art algorithms in terms of both accuracy and speed. This general framework can provide significant

improvements in the accuracy and efficiency of data association in many applications that rely on pairwise matchings such as metric/semantic SLAM, multi-object tracking, and multi-view point cloud registration.

Chapter 6 extended the ideas of Chapter 5 by developing the MIXER algorithm, which enables multiattribute, multiway fusion of uncertain pairwise affinities. Traditionally, multiway data association is formulated as a permutation synchronization problem, which takes as input binary pairwise data associations. However, pairwise data association typically begins with a similarity scoring process that generates continuous affinity matrices, not binary permutations. By requiring permutation matrices, existing multiway algorithms lose access to the additional uncertainty information encoded in the pairwise affinity matrices. The specific MIQP formulation presented in Chapter 6 leverages direct access to affinities and allows three modes of association (non-match, undecided, and match) over the range 0 to 1. This feature led to the insight that the MIQP formulation is a multiway extension of the maximum-weight matching problem (see Remark 6.3.1). Because of scalability issues of solving MIQPs, a novel continuous relaxation is presented and a projected gradient descent scheme is used to converge to feasible, binary solutions of the original problem. The guarantee of MIXER to converge to binary points sets it apart from related work, which relies on a final binarization step that can lead solutions to be infeasible. Finally, experimental evaluations in three datasets showed that MIXER frequently achieves higher accuracy than the state of the art, especially in noisy regimes with low observation redundancy. These properties and results establish MIXER as an effective and scalable multiway fusion algorithm in the presence of uncertain affinities.

Overall, the contributions of this thesis address key data association challenges in the presence of degraded measurements and outlier associations, thus producing more reliable geometric estimation. The core set of tools that were used to develop these contributions were graph theory, non-convex optimization, continuous relaxation strategies, and geometric understanding.

7.2 Future Directions

7.2.1 Learned-Invariants for Consistency Graph Construction

The work in Chapter 3 makes use of distances that are geometrically invariant to rotation and translation. In 3D point cloud scenarios, the most obvious distance that satisfies this property is the Euclidean metric. However, the robustness of CLIPPER would make benefit data association of features across images, where the distances are no longer strictly Euclidean due to the projective nature of image space. Although a perspective projection invariant could be formulated, it would be very costly to build the consistency matrix due to it requiring at least 5 points to compute, due to evaluating $\binom{n}{5}$ pairs. Instead, it would be very useful to identify invariants from a data-driven approach, allowing the invariance (or nearly-invariant) function to be learned from the particular application.

7.2.2 Mapping Using the Affine Grassmannian Manifold

As Chapter 4 demonstrated, the correct representation of geometric primitives is important. Future work should consider leveraging the affine Grassmannian representation for line and plane objects during the mapping stage, not just for localization. By building a Grassmannian observer [221], geometric landmarks could be refined in an online manner as point measurements are made. This would have the benefit of more stable landmarks (as opposed to one-shot detections) and potentially of more computationally efficient landmark detection. Further, once landmarks are matched using GraffMatch, the registration is currently accomplished via least squares alignment using point-normal-based representations. Instead, one could leverage the full geometric description of landmarks in the minimization over poses, which would lead to an intrinsic formulation of the optimization that could better exploit the geometry.

7.2.3 Hybrid Data Association for Multiple Object Tracking

Most multi object tracking frameworks produce frame-to-frame associations of new measurements and currently tracked objects. However, as discussed in Chapters 5 and 6, the use of many pairs of associations can improve data association overall. Future work should consider a hybrid approach to data association in multi object tracking, where data associations are made over a window of frames. Allowing a more global view of the problem would ultimately lead to better tracking performance and would add robustness in the presence of occlusions. Occlusions occur when an object is momentarily not seen by the sensor. The work in [222] begins to approach this problem, but the global consistency, accuracy, and runtime benefits of CLEAR or MIXER could provide higher fidelity tracking in the presence of occlusions.

References

[1] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[2] Mary B Alatise and Gerhard P Hancke. A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, 8:39830–39846, 2020.

[3] Miquel Kegeleirs, Giorgio Grisetti, and Mauro Birattari. Swarm slam: Challenges and perspectives. *Frontiers in Robotics and AI*, 8:23, 2021.

[4] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[5] Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.

[6] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[7] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

[8] Ursula Gather and Balvant K Kale. Maximum likelihood estimation in the presence of outiliers. *Communications in Statistics-Theory and Methods*, 17(11):3767–3784, 1988.

[9] Yaakov Bar-Shalom, Thomas E Fortmann, and Peter G Cable. Tracking and data association, 1990.

[10] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[11] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[12] Tim Bailey, Eduardo Mario Nebot, JK Rosenblatt, and Hugh F Durrant-Whyte. Data association for mobile robot navigation: A graph theoretic approach. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2512–2517, 2000.

[13] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[14] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.

[15] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[16] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[17] Jingnan Shi, Heng Yang, and Luca Carlone. Optimal pose and shape estimation for category-level 3d object perception. In *Robotics: Science and Systems (RSS)*, 2021.

[18] Samuel S Blackman. Multiple-target tracking with radar applications. *Dedham*, 1986.

[19] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[20] Rosario Aragues, Eduardo Montijano, and Carlos Sagues. Consistent data association in multi-robot systems with limited communications. In *RSS*, pages 97–104, 2011.

[21] Lukas Bernreiter, Shehryar Khattak, Lionel Ott, Roland Siegwart, Marco Hutter, and Cesar Cadena. Collaborative robot mapping using spectral graph analysis. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3662–3668. IEEE, 2022.

[22] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015.

[23] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

[24] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–6. IEEE, 2009.

[25] Joshua G Mangelson, Derrick Dominic, Ryan M Eustice, and Ram Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2916–2923. IEEE, 2018.

[26] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4091–4099, 2015.

[27] Yaakov Bar-Shalom and Xiao-Rong Li. Estimation and tracking- principles, techniques, and software. *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[28] José Neira and Juan D Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6):890–897, 2001.

[29] Vadim Indelman. Distributed perception and estimation: a short survey. In *Principles of Multi-Robot Systems, workshop in conjunction with Robotics Science and Systems*, 2015.

[30] Hugh F. Durrant-Whyte. *Sensor Models and Multisensor Integration*, pages 73–89. Springer New York, New York, NY, 1990.

[31] Federico Castanedo. A review of data fusion techniques. *The scientific world journal*, 2013, 2013.

[32] Hugh Durrant-Whyte and Thomas C. Henderson. *Multisensor Data Fusion*, pages 867–896. Springer International Publishing, Cham, 2016.

[33] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[34] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *European conference on computer vision*, pages 414–431. Springer, 2002.

[35] Tat-Jun Chin and David Suter. The maximum consensus problem: recent algorithmic advances. *Synthesis Lectures on Computer Vision*, 7(2):1–194, 2017.

[36] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[37] Konstantinos G Derpanis. Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3, 2010.

[38] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[39] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[40] David M Rosen, Kevin J Doherty, Antonio Terán Espinoza, and John J Leonard. Advances in inference and representation for simultaneous localization and mapping. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:215–242, 2021.

[41] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT press, 1987.

[42] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020.

[43] Eugene L Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

[44] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V 11*, pages 492–505. Springer, 2010.

[45] Qinghua Wu and Jin-Kao Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693–709, 2015.

[46] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In *Advances in neural information processing systems*, pages 1860–1868, 2013.

[47] Frank Dellaert. Monte carlo em for data-association and its applications in computer vision, September 2001.

[48] Michael Kaess and Frank Dellaert. Covariance recovery from a square root information matrix for data association. *Robotics and autonomous systems*, 57(12):1198–1210, 2009.

[49] Luca Carlone. Estimation contracts for outlier-robust geometric perception. *arXiv preprint arXiv:2208.10521*, 2022.

[50] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[51] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *IEEE CVPR*, pages 1802–1811, 2017.

[52] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *IEEE/CVF ICCV*, pages 8958–8966, 2019.

[53] Michael Kaess. Simultaneous localization and mapping with infinite planes. In *IEEE ICRA*, pages 4605–4611, 2015.

[54] Patrick Geneva, Kevin Eckenhoff, Yulin Yang, and Guoquan Huang. Lips: Lidar-inertial 3d plane slam. In *IEEE/RSJ IROS*, pages 123–130, 2018.

[55] Lipu Zhou, Guoquan Huang, Yinian Mao, Jincheng Yu, Shengze Wang, and Michael Kaess. Plc-lislam: Lidar slam with planes, lines and cylinders. *IEEE RA-L*, 2022.

[56] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *IEEE International Conference on Computer Vision*, pages 4032–4040, 2015.

[57] Roberto Tron, Xiaowei Zhou, Carlos Esteves, and Kostas Daniilidis. Fast multi-image matching via density-based clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4077–4086, Oct 2017.

[58] Dimitri P Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, 14(1):105–123, 1988.

[59] Yuxin Chen, Leonidas Guibas, and Qixing Huang. Near-optimal joint object matching via convex relaxation. In *International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 100–108, 22–24 Jun 2014.

[60] Eleonora Maset, Federica Arrigoni, and Andrea Fusiello. Practical and efficient multi-view matching. In *IEEE International Conference on Computer Vision*, pages 4578–4586, 2017.

[61] Spyridon Leonardos, Xiaowei Zhou, and Kostas Daniilidis. Distributed consistent data association via permutation synchronization. In *IEEE ICRA*, pages 2645–2652, 2017.

[62] Florian Bernard, Johan Thunberg, Jorge Goncalves, and Christian Theobalt. Synchronisation of partial multi-matchings via non-negative factorisations. *Pattern Recognition*, 92:146–155, 2019.

[63] Spyridon Leonardos and Kostas Daniilidis. A distributed optimization approach to consistent multiway matching. In *IEEE CDC*, pages 89–96, 2018.

[64] Spyridon Leonardos, Xiaowei Zhou, and Kostas Daniilidis. A low-rank matrix approximation approach to multiway matching with applications in multisensory data association. In *2020 IEEE International Conference on Robotics and Automation*, pages 8665–8671, 2020.

[65] Yunpeng Shi, Shaohan Li, Tyler Maunu, and Gilad Lerman. Scalable cluster-consistency statistics for robust multi-object matching. In *IEEE 3DV*, pages 352–360, 2021.

[66] Shaohan Li, Yunpeng Shi, and Gilad Lerman. Fast, accurate and memory-efficient partial permutation synchronization. In *IEEE/CVF CVPR*, pages 15735–15743, 2022.

[67] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*, pages 399–402, 2005.

[68] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. Early vs late fusion in multimodal convolutional neural networks. In *FUSION*, pages 1–6. IEEE, 2020.

[69] Parker C Lusk, Kaveh Fathian, and Jonathan P How. CLIPPER: A graph-theoretic framework for robust data association. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13828–13834, 2021. `https://arxiv.org/abs/2011.10202`.

[70] Parker C Lusk and Jonathan P How. Global data association for slam with 3d grassmannian manifold objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4463–4470, 2022. `https://arxiv.org/pdf/2205.08556.pdf`.

[71] Parker C Lusk, Devarth Parikh, and Jonathan P How. GraffMatch: Global matching of 3d lines and planes for wide baseline lidar registration. *IEEE Robotics and Automation Letters*, 8(2):632–639, 2022. `https://arxiv.org/abs/2212.12745`.

[72] Bingyi Cao, Claas-Norman Ritter, Daniel Göhring, and Raúl Rojas. Accurate localization of autonomous vehicles based on pattern matching and graph-based optimization in urban environments. In *IEEE ITSC*, 2020.

[73] Kaveh Fathian, Kasra Khosoussi, Yulun Tian, Parker C Lusk, and Jonathan P How. CLEAR: A consistent lifting, embedding, and alignment rectification algorithm for multiview data association. *IEEE Transactions on Robotics*, 36(6):1686–1703, 2020.

[74] Parker C Lusk, Kaveh Fathian, and Jonathan P How. Mixer: Multiattribute, multiway fusion of uncertain pairwise affinities. *IEEE Robotics and Automation Letters*, 8(5):2462–2469, 2023. `https://arxiv.org/abs/2210.08360`.

[75] Tat-Jun Chin, Zhipeng Cai, and Frank Neumann. Robust fitting in computer vision: Easy or hard? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 701–716, 2018.

[76] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *European Conference on Computer Vision*, pages 500–513. Springer, 2008.

[77] Huu Minh Le, Tat-Jun Chin, Anders Eriksson, Thanh-Toan Do, and David Suter. Deterministic approximate methods for maximum consensus robust fitting. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[78] Huu M Le, Thanh-Toan Do, Tuan Hoang, and Ngai-Man Cheung. SDRSAC: Semidefinite-based randomized approach for robust point cloud registration without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 124–133, 2019.

[79] Luca Carlone, Andrea Censi, and Frank Dellaert. Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2667–2674. IEEE, 2014.

[80] Hongdong Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1074–1080. IEEE, 2009.

[81] Tat-Jun Chin, Pulak Purkait, Anders Eriksson, and David Suter. Efficient globally optimal consensus maximisation with tree search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2413–2421, 2015.

[82] Charles V Stewart. Robust parameter estimation in computer vision. *SIAM review*, 41(3):513–537, 1999.

[83] Luca Carlone and Giuseppe C Calafiore. Convex relaxations for pose graph optimization with outliers. *IEEE Robotics and Automation Letters*, 3(2):1160–1167, 2018.

[84] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[85] Stuart Geman and D McClure. Bayesian image analysis: An application to single photon emission tomography. *Amer. Statist. Assoc*, pages 12–18, 1985.

[86] Kirk MacTavish and Timothy D Barfoot. At all costs: A comparison of robust cost functions for camera correspondence outliers. In *2015 12th Conference on Computer and Robot Vision*, pages 62–69. IEEE, 2015.

[87] Olof Enqvist, Erik Ask, Fredrik Kahl, and Kalle Aström. Robust fitting for multiple view geometry. In *European Conference on Computer Vision*, pages 738–751. Springer, 2012.

[88] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.

[89] Pierre-Yves Lajoie, Siyi Hu, Giovanni Beltrame, and Luca Carlone. Modeling perceptual aliasing in slam via discrete–continuous graphical models. *IEEE Robotics and Automation Letters*, 4(2):1232–1239, 2019.

[90] Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph slam. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884. IEEE, 2012.

[91] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *2013 IEEE International Conference on Robotics and Automation*, pages 62–69. Ieee, 2013.

[92] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *The International Journal of Robotics Research*, 32(7):826–840, 2013.

[93] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013.

[94] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and vision computing*, 21(13-14):1145–1153, 2003.

[95] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *IEEE TPAMI*, volume 14, pages 239–256, 1992.

[96] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[97] Philippe Babin, Philippe Giguere, and François Pomerleau. Analysis of robust functions for registration algorithms. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1451–1457. IEEE, 2019.

[98] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European conference on computer vision*. Springer, 2016.

[99] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE T-RO*, 37(2):314–333, 2020.

[100] Heng Yang and Luca Carlone. In perfect shape: Certifiably optimal 3d shape reconstruction from 2d landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 621–630, 2020.

[101] Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010.

[102] Afonso S Bandeira. A note on probably certifiably correct algorithms. *Comptes Rendus Mathematique*, 354(3):329–333, 2016.

[103] R Burkard, M Dell'Amico, and S Martello. Assignment problems. *SIAM*, 2009.

[104] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[105] Lyle Ramshaw and Robert E Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1*, 20, 2012.

[106] Ran Duan and Seth Pettie. Linear-time approximation for maximum weight matching. *Journal of the ACM*, 61(1):1–23, 2014.

[107] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[108] Tjalling C Koopmans and Martin Beckmann. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pages 53–76, 1957.

[109] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007.

[110] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[111] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.

[112] Mokhtar S Bazaraa and Alwalid N Elshafei. An exact branch-and-bound procedure for the quadratic-assignment problem. *Naval Research Logistics Quarterly*, 26(1):109–121, 1979.

[113] MS Bazaraa and Omer Kirca. A branch-and-bound-based heuristic for solving the quadratic assignment problem. *Naval research logistics quarterly*, 30(2):287–304, 1983.

[114] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision*, volume 2, pages 1482–1489, 2005.

[115] Lorenzo Torresani, Vladimir Kolmogorov, and Carsten Rother. A dual decomposition approach to feature correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):259–271, 2012.

[116] Paul Swoboda, Carsten Rother, Hassan Abu Alhaija, Dagmar Kainmuller, and Bogdan Savchynskyy. A study of lagrangean decompositions and dual ascent solvers for graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1607–1616, 2017.

[117] Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.

[118] Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d'Aspremont. Convex relaxations for permutation problems. *Advances in neural information processing systems*, 26, 2013.

[119] Itay Kezurer, Shahar Z Kovalsky, Ronen Basri, and Yaron Lipman. Tight relaxation of quadratic matching. In *Computer Graphics Forum*, volume 34, pages 115–128. Wiley Online Library, 2015.

[120] Florian Bernard, Christian Theobalt, and Michael Moeller. Ds*: Tighter lifting-free convex relaxations for quadratic matching problems. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4310–4319, 2018.

[121] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2008.

[122] Feng Zhou and Fernando De la Torre. Factorized graph matching. *Transactions on pattern analysis and machine intelligence*, 38(9):1774–1789, 2015.

[123] Bo Jiang, Jin Tang, Chris Ding, and Bin Luo. Binary constraint preserving graph matching. In *IEEE conference on computer vision and pattern recognition*, pages 4402–4409, 2017.

[124] D Khuê Lê-Huu and Nikos Paragios. Alternating direction graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4914–4922, 2017.

[125] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[126] Johan Hastad. Clique is hard to approximate within n1 minus epsilon. *Acta Math*, 182, 1999.

[127] A Patricia Ambler, Harry G. Barrow, Christopher M. Brown, Rod M. Burstall, and Robin J. Popplestone. A versatile system for computer-controlled assembly. *Artificial Intelligence*, 6(2):129–156, 1975.

[128] Robert C Bolles. Robust feature matching through maximal cliques. In *Imaging Applications for Automated Industrial Inspection and Assembly*, volume 182, pages 140–149. International Society for Optics and Photonics, 1979.

[129] Olof Enqvist, Klas Josephson, and Fredrik Kahl. Optimal correspondences from pairwise constraints. In *IEEE International Conference on Computer Vision*, pages 1295–1302, 2009.

[130] A Parra Bustos, Tat-Jun Chin, Frank Neumann, Tobias Friedrich, and Maximilian Katzmann. A practical maximum clique algorithm for matching with pairwise constraints. *arXiv preprint arXiv:1902.01534*, 2, 2019.

[131] Jingnan Shi, Heng Yang, and Luca Carlone. Robin: a graph-theoretic approach to reject outliers in robust estimation using invariants. *arXiv preprint arXiv:2011.03659*, 2020.

[132] Edwin Olson, Matthew R Walter, Seth J Teller, and John J Leonard. Single-cluster spectral graph partitioning for robotics applications. In *Robotics: Science and Systems*, pages 265–272, 2005.

[133] Andrew V Goldberg. *Finding a maximum density subgraph.* University of California Berkeley, 1984.

[134] Yuhang Ming, Xingrui Yang, and Andrew Calway. Object-augmented rgb-d slam for wide-disparity relocalisation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2203–2209. IEEE, 2021.

[135] Sarah H Cen and Paul Newman. Radar-only ego-motion estimation in difficult settings via graph matching. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 298–304. IEEE, 2019.

[136] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433, 2010.

[137] Andy Nguyen, Mirela Ben-Chen, Katarzyna Welnicka, Yinyu Ye, and Leonidas Guibas. An optimization approach to improving collections of shape maps. In *Computer Graphics Forum*, volume 30, pages 1481–1491. Wiley Online Library, 2011.

[138] Stephen Phillips and Kostas Daniilidis. All graphs lead to rome: Learning geometric and cycle-consistent representations with graph convolutional networks. *arXiv preprint arXiv:1901.02078*, 2019.

[139] Yunpeng Shi, Shaohan Li, and Gilad Lerman. Robust multi-object matching via iterative reweighting of the graph connection laplacian. *Advances in Neural Information Processing Systems*, 33:15243–15253, 2020.

[140] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[141] Nan Hu, Qixing Huang, Boris Thibert, and Leonidas J Guibas. Distributable consistent multi-object matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2463–2471, 2018.

[142] Jin-Gang Yu, Gui-Song Xia, Ashok Samal, and Jinwen Tian. Globally consistent correspondence of multiple feature sets using proximal gauss–seidel relaxation. *Pattern Recognition*, 51:255–267, 2016.

[143] Antonio De Rosa and Aida Khajavirad. Efficient joint object matching via linear programming. *arXiv preprint arXiv:2108.11911*, 2021.

[144] Florian Bernard, Johan Thunberg, Jorge Goncalves, and Christian Theobalt. Synchronisation of partial multi-matchings via non-negative factorisations. *arXiv preprint arXiv:1803.06320*, 2018.

[145] Junchi Yan, Zhe Ren, Hongyuan Zha, and Stephen Chu. A constrained clustering based approach for matching a collection of feature sets. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3832–3837. IEEE, 2016.

[146] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV 10*, pages 705–718. Springer, 2008.

[147] Zachary Serlin, Guang Yang, Brandon Sookraj, Calin Belta, and Roberto Tron. Distributed and consistent multi-image feature matching via quickmatch. *The International Journal of Robotics Research*, 39(10-11):1222–1238, 2020.

[148] Junchi Yan, Yu Tian, Hongyuan Zha, Xiaokang Yang, Ya Zhang, and Stephen M Chu. Joint optimization for consistent multiple graph matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1649–1656, 2013.

[149] Xinchu Shi, Haibin Ling, Weiming Hu, Junliang Xing, and Yanning Zhang. Tensor power iteration for multi-graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5062–5070, 2016.

[150] Junchi Yan, Minsu Cho, Hongyuan Zha, Xiaokang Yang, and Stephen M Chu. Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1228–1242, 2015.

[151] Paul Swoboda, Dagmar Kainmuller, Ashkan Mokarian, Christian Theobalt, and Florian Bernard. A convex relaxation for multi-graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[152] François-Xavier Dupé, Rohit Yadav, Guillaume Auzias, and S Takerkart. Kernelized multi-graph matching. In *14th Asian Conference on Machine Learning*, pages 136–144, 2022.

[153] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE ICRA*, pages 3212–3217, 2009.

[154] Alexander Schaefer, Daniel Büscher, Johan Vertens, Lukas Luft, and Wolfram Burgard. Long-term urban vehicle localization using pole landmarks extracted from 3-d lidar scans. In *IEEE ECMR*, pages 1–7, 2019.

[155] Julius Kümmerle, Marc Sons, Fabian Poggenhans, Tilman Kühner, Martin Lauer, and Christoph Stiller. Accurate and efficient self-localization on roads using basic geometric primitives. In *IEEE ICRA*, pages 5965–5971, 2019.

[156] Lipu Zhou, Shengze Wang, and Michael Kaess. π-lsam: Lidar smoothing and mapping with planes. In *IEEE ICRA*, pages 5751–5757, 2021.

[157] Yulin Yang and Guoquan Huang. Observability analysis of aided ins with heterogeneous features of points, lines, and planes. *IEEE Transactions on Robotics*, 35(6):1399–1418, 2019.

[158] Daniel Wilbers, Christian Merfels, and Cyrill Stachniss. Localization with sliding window factor graphs on third-party maps for automated driving. In *IEEE ICRA*, pages 5951–5957, 2019.

[159] Kaustubh Pathak, Andreas Birk, Narunas Vaškevičius, and Jann Poppinga. Fast registration based on noisy planes with unknown correspondences for 3-d mapping. *IEEE T-RO*, 26(3):424–441, 2010.

[160] Jianwen Jiang, Jikai Wang, Peng Wang, Peng Bao, and Zonghai Chen. Lipmatch: Lidar point cloud plane based loop-closure. *IEEE Robotics and Automation Letters*, 5(4):6861–6868, 2020.

[161] Eduardo Fernández-Moral, Walterio Mayol-Cuevas, Vicente Arevalo, and Javier Gonzalez-Jimenez. Fast place recognition with plane-based maps. In *IEEE ICRA*, pages 2719–2724, 2013.

[162] Bingyi Cao, Ricardo Carrillo Mendoza, Andreas Philipp, and Daniel Göhring. Lidar-based object-level slam for autonomous vehicles. In *IEEE/RSJ IROS*, 2021.

[163] Peter J. Huber. Robust Estimation of a Location Parameter. volume 35, pages 73–101. Institute of Mathematical Statistics, 1964.

[164] Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1):59–76, 1997.

[165] Michael Bosse, Gabriel Agamennoni, and Igor Gilitschenski. Robust estimation and applications in robotics. *Foundations and Trends in Robotics*, 4:225–269, 01 2016.

[166] Daniel M Dunlavy and Dianne P O'Leary. Homotopy optimization methods for global optimization. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA ..., 2005.

[167] Melisew Tefera Belachew and Nicolas Gillis. Solving the maximum clique problem with symmetric rank-one non-negative matrix approximation. *Journal of Optimization Theory and Applications*, 173(1):279–296, 2017.

[168] Dragoš M. Cvetković, Michael Doob, and Horst Sachs. *Spectra of Graphs: Theory and Application*. Academic Press, New York, 1980.

[169] Geoffrey Canright and Kenth Engø-Monsen. Roles in networks. *Science of Computer Programming*, 53(2):195–214, 2004.

[170] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870*, 2017.

[171] Siddharth Agarwal, Ankit Vora, Gaurav Pandey, Wayne Williams, Helen Kourous, and James McBride. Ford multi-av seasonal dataset. *The International Journal of Robotics Research*, 39(12):1367–1376, 2020.

[172] Soonyong Park, Sung-Kee Park, and Martial Hebert. Fast and scalable approximate spectral matching for higher order graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):479–492, 2013.

[173] Abner MC Araújo and Manuel M Oliveira. A robust statistics approach for plane detection in unorganized point clouds. *Pattern Recognition*, 100:107115, 2020.

[174] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016.

[175] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[176] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *IEEE TPAMI*, (5):698–700, 1987.

[177] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[178] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2015.

[179] Sourav Garg, Tobias Fischer, and Michael Milford. Where is your place, visual place recognition? In *IJCAI*, pages 4416–4425, 8 2021.

[180] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[181] Giseop Kim, Sunwook Choi, and Ayoung Kim. Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 38(3):1856–1874, 2022.

[182] Hriday Bavle, Jose Luis Sanchez-Lopez, Muhammad Shaheer, Javier Civera, and Holger Voos. Situational graphs for robot navigation in structured indoor environments. *arXiv preprint arXiv:2202.12197*, 2022.

[183] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, Berkeley, CA, 2014.

[184] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[185] Ake Björck and Gene H Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 1973.

[186] Lek-Heng Lim, Ken Sze-Wai Wong, and Ke Ye. The grassmannian of affine subspaces. *Foundations of Computational Mathematics*, 2021.

[187] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE CVPR*, 2012.

[188] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE TPAMI*, 2022.

[189] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *IJRR*, 35(9):1023–1035, 2016.

[190] Hao Dong, Xieyuanli Chen, and Cyrill Stachniss. Online range image-based pole extractor for long-term lidar localization in urban environments. In *IEEE ECMR*, 2021.

[191] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF ICCV*, 2019.

[192] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[193] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[194] Abner MC Araújo and Manuel M Oliveira. A robust statistics approach for plane detection in unorganized point clouds. *Pattern Recognition*, 2020.

[195] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPF-FoldNet: Unsupervised learning of rotation invariant 3d local descriptors. In *ECCV*, 2018.

[196] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[197] Steven S Skiena. *The algorithm design manual*, volume 1. Springer Science & Business Media, 1998.

[198] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[199] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[200] Helmut Lutkepohl. *Handbook of matrices*, volume 1. Wiley Chichester, 1996.

[201] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[202] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[203] Jean Gallier. Notes on elementary spectral graph theory. applications to graph clustering using normalized cuts. *arXiv preprint arXiv:1311.2492*, 2013.

[204] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[205] Andrea Vedaldi and Brian Fulkerson. VLFeat: An open and portable library of computer vision algorithms. In *ACM international conference on Multimedia*, pages 1469–1472, 2010.

[206] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-D vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

[207] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J.P. How. Search and rescue under the forestcanopy using multiple uas. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2018.

[208] Rainer Kuemmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[209] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *IEEE ICIP*, pages 3464–3468, 2016.

[210] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[211] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE TPAMI*, 2019.

[212] Aryan Mokhtari, Asuman Ozdaglar, and Ali Jadbabaie. Escaping saddle points in constrained optimization. *NeurIPS*, 31, 2018.

[213] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[214] Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[215] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Kimera-multi: robust, distributed, dense metric-semantic slam for multi-robot systems. *IEEE Transactions on Robotics*, 2022.

[216] Hsu-kuang Chiu, Jie Li, Rareş Ambruş, and Jeannette Bohg. Probabilistic 3d multi-modal, multi-object tracking for autonomous driving. In *IEEE ICRA*, pages 14227–14233. IEEE, 2021.

[217] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv*, 2018.

[218] Yi-Ting Chen, Jinghao Shi, Zelin Ye, Christoph Mertz, Deva Ramanan, and Shu Kong. Multimodal object detection via probabilistic ensembling. In *ECCV*, pages 139–158. Springer, 2022.

[219] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. GNN3DMOT: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *IEEE/CVF CVPR*, pages 6499–6508, 2020.

[220] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *IEEE/CVF ICCV*, pages 2365–2374, 2019.

[221] Quentin Rentmeesters, Pierre-Antoine Absil, and Paul Van Dooren. A filtering technique on the grassmann manifold. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*, volume 5, 2010.

[222] Min Yang, Yuwei Wu, and Yunde Jia. A hybrid data association framework for robust online multi-object tracking. *IEEE Transactions on Image Processing*, 26(12):5667–5679, 2017.