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

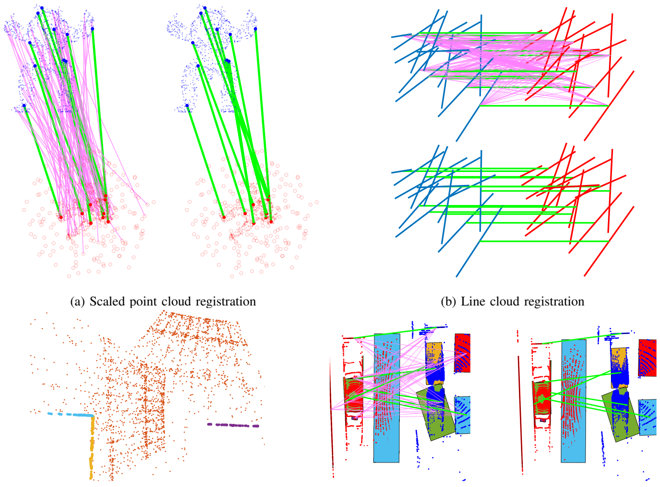
CLIPPER: A Graph-Theoretic Framework for Robust Data Association

Parker C. Lusk*, Kaveh Fathian*, Jonathan P. How

***Abstract*— We present CLIPPER (Consistent LInking, Pruning, and Pairwise Error Rectification), a framework for robust data association in the presence of noise and outliers. We formulate the problem in a graph-theoretic framework using the notion of geometric consistency. State-of-the-art techniques that use this framework utilize either combinatorial optimization techniques that do not scale well to large-sized problems, or use heuristic approximations that yield low accuracy in high-noise, high-outlier regimes. In contrast, CLIPPER uses a relaxation of the combinatorial problem and returns solutions that are guaranteed to correspond to the optima of the original problem. Low time complexity is achieved with an efficient projected gradient ascent approach. Experiments indicate that CLIPPER maintains a consistently low runtime of 15 ms where exact methods can require up to 24 s at their peak, even on small-sized problems with 200 associations. When evaluated on noisy point cloud registration problems, CLIPPER achieves 100% precision and 98% recall in 90% outlier regimes while competing algorithms begin degrading by 70% outliers. In an instance of associating noisy points of the Stanford Bunny with 990 outlier associations and only 10 inlier associations, CLIPPER successfully returns 8 inlier associations with 100% precision in 138 ms. Code is available at https://mit-acl.github.io/clipper.**

I. INTRODUCTION

Finding correct one-to-one correspondences between two sets of objects $\mathcal{A} \overset{\text{def}}{=} \{a_1, \dots, a_n\}$ and $\mathcal{A}' \overset{\text{def}}{=} \{a'_1, \dots, a'_m\}$ is a fundamental problem in robotics, arising in a wide range of perception and estimation pipelines. In practice, observations of objects are "noisy" and "partial", i.e., when an *unknown* number of objects in $\mathcal{A}$ do not correspond to any object in $\mathcal{A}'$ (outliers). The traditional linear assignment approach based on corresponding objects of high similarity, e.g., using the Hungarian [1] or auction [2] algorithms, is not robust to high-noise, high-outlier regimes, leading to incorrect correspondences. In this work, we propose the CLIPPER (Consistent LInking, Pruning, and Pairwise Error Rectification) framework which leverages the notion of *geometric consistency* between object pairs to find correct correspondences in these extreme regimes. Our primary motivation is robust perception with applications such as those shown in Fig. 1, SLAM/loop closure [3], [4], point cloud registration [5], [6], shape alignment [7], object detection [8], and multiple object tracking [9], [10]. However, our framework is general and applicable to other pairwise data association problems.


(a) Scaled point cloud registration

(b) Line cloud registration

(c) Plane registration (top-down view)

(d) Planar patch registration

Fig. 1. 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 [17]). 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.

When an attribute between objects in set $\mathcal{A}$ is the same as the attribute between their correctly associated objects in $\mathcal{A}'$, these objects are considered geometrically consistent (e.g., see Fig. 2). Incorporating geometric consistency in data association ultimately leads to a combinatorial optimization, such as maximum clique [11], maximum consensus [12], [13], or quadratic assignment [14] formulations. Relaxations of this NP-hard problem exist [15], [16], but exhibit poor performance in high-outlier regimes or for large problem size. In contrast, CLIPPER maintains high precision with low runtime across various outlier-regimes and problem sizes.

CLIPPER incorporates the concept of geometric consistency in a graph-theoretic framework, and proposes finding consistent association (inliers) by finding the densest subgraph. This formulation is particularly suited to weighted graphs and improves precision compared to maximum clique frameworks which are limited to binary graphs. CLIPPER uses the continuous relaxation technique introduced in [18] to guarantee that the recovered solution corresponds to a dense subgraph. Using projected gradient ascent with backtracking line search, CLIPPER achieves a consistently low runtime as compared to other algorithms. Further, our experiments show that CLIPPER is capable of attaining 100% precision even in high outlier regimes where other algorithms breakdown.

In summary, the contributions of this work include:

- An optimization formulation for selecting inlier associ-

P. C. Lusk, K. Fathian and J. P. How are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. {plusk, kavehf, jhow}@mit.edu. *Authors contributed equally.

This work is supported by Ford Motor Company and by ARL DCIST under Cooperative Agreement Number W911NF-17-2-0181.

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

ations suitable for both binary and weighted graphs.

- A relaxation of the resulting NP-hard optimization problem with optimality guarantees (see Sec III-A).
- A polynomial-time algorithm for solving the relaxed formulation based on projected gradient ascent, scalable to large-sized data association problems.

Additionally, we include a discussion on the application of geometric consistency to various observation types (e.g., points, lines, planes) commonly found in robotic perception. Finally, we benchmark CLIPPER against the state of the art when finding associations between two point clouds. We find that CLIPPER is able to achieve 100% precision in 99% outlier regimes, where the performance of competing algorithms begin to degrade.

Related works most pertinent to this work include Bailey et al. [11], where geometric consistency is leveraged for 2D LiDAR scan matching. Formulated as a maximum common subgraph problem, their technique resulted in a *binary* consistency graph for which the maximum clique indicated the correct data association. Leordeanu and Hebert [15] built on this graph-theoretic idea and instead constructed a *weighted* consistency graph, where edge weights represent the geometric consistency of associations. Enqvist et al. [19] noted the suboptimality of [15] and proposed a vertex covering formulation, essentially an alternative to the maximum clique formulation of [11]. Parra et al. [20] proposed a practical maximum clique algorithm for geometric consistency based on branch and bound and graph coloring. Recent algorithms leveraging these ideas for estimation tasks include PCM [4], where maximum cliques correspond to the largest set of pairwise-consistent loop closure measurements, and TEASER [6], where maximum cliques correspond to inlier associations for point cloud registration, further formalized in ROBIN [21]. CLIPPER advances these works with a continuous relaxation applicable to weighted graphs.

II. Graph-Theoretic Formulation

A standard approach for robust data association in the presence of noise and outliers is to find the largest set of *geometrically consistent* associations. This problem can be formulated in a graph-theoretic framework. In this section, we review the consistency graph framework and the construction of the affinity matrix.

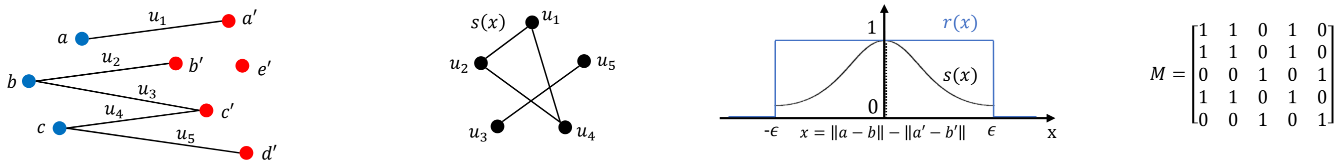
A. Consistency Graph

For simplicity, and without loss of generality, we introduce the concept of a consistency graph for the point cloud registration problem. In Section V, we will show how this general framework can be used for robust data association in other robotic perception domains. The point cloud registration problem consists of finding the rotation and translation that map coordinates of a set of points to their corresponding points in another set. The main challenge is finding the correct correspondences as the points are noisy and contain outliers. *Outlier points* are points with large noise, and ones that are partially observed (i.e., points in one set that do no have a counterpart in the other set). *Outlier associations* are correspondences that match (inlier or outlier) points in one set incorrectly to points in the other set. Fig. 2a gives an example of two point clouds identified by blue and red colors. Inlier points are denoted by a, b, c in the blue point cloud and their transformed counterparts by a', b', c' in the red point cloud. Points d', e' do not correspond to any blue point and hence are considered as outliers. A set of putative associations between the points, denoted by $u_1, \dots, u_5$, are given, where u_1, u_2, u_4 are inlier and u_3, u_5 are outlier associations. In the case that no putative associations are given, an all-to-all hypothesis can be generated.

The consistency of associations can be assessed and represented in the graph-theoretic framework of the *consistency graph.* The consistency graph $\mathcal{G}$ of n associations consists of n vertices, where each vertex represents an association. Edges between the vertices of $\mathcal{G}$ show that their corresponding associations are consistent. The example in Fig. 2b illustrates the consistency graph for the associations in Fig. 2a. Since rotation and translation are distance-preserving transformations, the distance between the points in one set should be identical (in the noiseless setting) to their counterparts in the other set when associations are correct. This attribute can be used to assess the *geometric consistency* of two associations, where an edge between two vertices of $\mathcal{G}$ indicates that the distances between the points matched by the associations are the same. The largest set of mutually consistent associations is given by the largest fully connected subgraph (maximum clique), which consists of vertices u_1, u_2, u_4 in the example of Fig. 2b.

B. Affinity Matrix

The *affinity matrix* M of a consistency graph with n vertices is an $n \times n$ *symmetric* matrix with entries in the interval $[0, 1]$. The diagonal entries $M(i, i)$ measure the *similarity* of the data points that association i matches, e.g., based on the similarity of point "descriptors" (for instance, the FPFH descriptors for point clouds [22]). Scores of 0 and 1 indicate the lowest and the highest similarity, respectively. The diagonal entries of M are set to 1 when similarity information is not available. In this case, $M = A + I$, where A is the (weighted) *adjacency matrix* of the consistency graph and I is the identity matrix. The off-diagonal entries $M(i, j)$ measure the geometric consistency of association pairs i and j, and similarly range from 0 to 1. For example, in the point cloud registration problem, the distance between two points a, b matched to a', b' by associations u_i and u_j can be used to measure the consistency as $M(i, j) \overset{\text{def}}{=} s\left(\|a - b\| - \|a' - b'\|\right) \in [0, 1]$. Here, $s : \mathbb{R} \to [0, 1]$ is a scoring function such that $s(0) = 1$ and $s(x) = 0$ for $|x| > \epsilon$, as illustrated in Fig 2c. The threshold ϵ is based on a bounded noise model with a noise range of $\epsilon/2$ on the point coordinates. Hence, if the distance between the points differs more than this threshold, the associations are considered inconsistent. Another source of inconsistency, referred to as the *distinctness constraint*, is when correct associations are expected to be one-to-one. Hence, for any two associations i and j that originate or terminate at the


u_1
a
a'
u_2
b'
e'
b
u_3
u_4
c'
c
u_5
d'
$s(x)$
u_1
u_2
u_5
u_3
u_4
1
$r(x)$
$s(x)$
0
$-\epsilon$
$x = \|a - b\| - \|a' - b'\|$
ϵ
x
$M = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$

(a) (b) (c) (d)

Fig. 2. An illustrative example of the consistency graph and its affinity matrix for point cloud registration. (a) Blue and red point clouds where putative associations are labeled as $u_1, \dots, u_5$. (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_1, u_2 mapping points a, b to a', b' are consistent if $\|a - b\| = \|a' - b'\|$. (c) Edges of the consistency graph are weighted to measure the consistency of associations using scalar functions $s(x)$ or $r(x)$. For example, the edge connecting vertices u_1, u_2 takes the weight $s(\|a - b\| - \|a' - b'\|)$. When the difference between distances is larger than the threshold ϵ or when two associations start and end at the same point, they are deemed as inconsistent. (d) The affinity matrix M of the consistency graph $\mathcal{G}$, computed for the point cloud example using the binary score $r(x)$.

same point, $M(i,j) \overset{\text{def}}{=} 0$ to indicate mutual inconsistency of associations. Lastly, we note that a binary scoring function, e.g., $r(x) : \mathbb{R} \to \{0,1\}$ in Fig. 2c, can be used. This will lead to a binary affinity matrix, shown in Fig. 2d.

III. Optimization Formulation of CLIPPER

Given the consistency graph $\mathcal{G}$ with n vertices representing associations (all-to-all or putative), and its $n \times n$, symmetric affinity matrix M, we propose the problem

$$\begin{array}{ll} \underset{u \in \{0,1\}^n}{\text{maximize}} & \dfrac{u^\top M\, u}{u^\top u} \\ \text{subject to} & u_i\, u_j = 0 \quad \text{if } M(i,j) = 0,\ \forall_{i,j}, \end{array} \tag{1}$$

for finding the *densest* subset of consistent associations. Here, the optimization variable u is a binary vector of size n, with 1 elements indicating associations that are selected as inliers, and 0's otherwise. Since u is binary, the constraint $u_i\, u_j = 0$ ensures that if $M(i,j) = 0$, then at most one of the associations u_i or u_j is selected in the answer.

When M is binary (e.g., obtained by using the scoring function $r(x)$ in Fig. 2c) and has one diagonal entries, it is straightforward to show that (1) simplifies to

$$\begin{array}{ll} \underset{u \in \{0,1\}^n}{\text{maximize}} & \sum_{i=1}^n u_i \\ \text{subject to} & u_i\, u_j = 0 \quad \text{if } M(i,j) = 0,\ \forall_{i,j}. \end{array} \tag{2}$$

Problem (2) is known as the maximum clique problem (MCP), and its solution gives the largest set of consistent associations. The MCP framework is used frequently in the literature [6], [4], [11] for data association in the presence of noise and a large number of outliers, where the vertices/associations in the maximum clique are considered as the correct/inlier correspondences. The justification for this consideration is based on the assumption that the noise and outlier points are random, unbiased, and unstructured, thus they are not expected to form a large clique in the consistency graph [15]. We note that MCP is a well-known NP-hard problem [23] in its full generality, hence, algorithms that rely on solving (2) become computationally intractable as the problem size grows.

The *density* of a graph is defined as the total sum of edge weights divided by the number of vertices. The *densest subgraph* is the subset of graph vertices and their corresponding edges that have the highest density. Given a graph $\mathcal{G}'$ with affinity matrix M' (with 1 diagonal entries), the densest subgraph of $\mathcal{G}'$ is found from

$$\underset{u \in \{0,1\}^n}{\text{maximize}} \quad \frac{u^\top M' u}{u^\top u}, \tag{3}$$

where elements $u_i = 1$ in the solution correspond to the vertices in the densest subgraph [24]. Noting the similarity of the objectives in (3) and (1), problem (1) can be interpreted as finding the densest, fully connected subgraph of $\mathcal{G}$. The full connectivity requirement is due to the constraints in (1), which prohibit the selection of vertices u_i, u_j that are not connected, i.e., $M(i,j) = 0$. The densest-subgraph objective in (1) is crucial in the weighted case, and sets it apart from the maximum edge weighted clique problem [25]. For example, consider a weighted matrix M and two solution candidates $u, \bar{u}$ as

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.2 & 0.2 \\ 0 & 0 & 0.2 & 1 & 0.2 \\ 0 & 0 & 0.2 & 0.2 & 1 \end{bmatrix},\ u = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ \bar{u} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{4}$$

The MCP objective in (2), or the unnormalized objective of $u^\top M\, u$, returns $\bar{u}$ as the optimum solution, whereas the block of M corresponding to $\bar{u}$ has low consistency scores of 0.2 between the vertices/associations. On the other hand, the normalized objective of (1) takes values of 2 and 1.4 for u and $\bar{u}$, respectively, leading to selection of the smaller, but more consistent subgraph. Example (4) further highlights the importance of choosing a weighted scheme over binary in the affinity matrix, since the maximum clique formulation returns $\bar{u}$ as a solution. While this problem can be avoided by choosing a smaller ϵ, a conservative threshold leads to classifying correct associations as outliers (i.e., lower output recall). Further, a weighted scheme can resolve symmetric cases where two binary cliques have the same size.

A. *Continuous Relaxation*

The main challenges in solving (1) are the combinatorial complexity of the problem due to its binary domain and the nonlinearity of the objective in u. This makes it intractable to solve the problem to global optimality in real time, even for small-sized data association problems. A standard workaround is to relax the domain and the constraints of (1) to obtain a continuous problem amenable to fast solution, followed by projecting this solution back to the domain and constraint manifold of the original problem. Among the relaxation strategies that can be considered, the main advantage of the following approach is that the solutions obtained from the relaxed problem correspond to optima of the original problem, as will be discussed shortly.

We propose the relaxation of (1) as

$$\begin{array}{ll} \underset{u\in\mathbb{R}^n_+}{\text{maximize}} & F(u) \overset{\text{def}}{=} u^\top M_d\, u \\ \text{subject to} & \|u\| \le 1 \end{array} \tag{5}$$

where $\mathbb{R}_+$ is the set of non-negative reals, $\|\cdot\|$ is the ℓ_2 vector norm, and

$$M_d(i,j) \overset{\text{def}}{=} \begin{cases} M(i,j) & \text{if } \; M(i,j) \neq 0 \\ -d & \text{if } \; M(i,j) = 0 \end{cases} \tag{6}$$

where $d > 0$ is a positive scalar. This approach is inspired by [18], which directly integrates the constraints of the original problem into the continuous formulation of (5) via the matrix M_d. Intuitively, when $M_d(i,j) = -d$, the scalar d penalizes joint selection of u_i, u_j in the objective by the amount $-2\,d\,u_i\,u_j$. Hence, as d increases the entries of solution u that violate the constraints are pushed to zero.

When $d \geq n$, (local or global) optima of (5) satisfy the constraints in the original problem, i.e., $u_i\,u_j = 0$ if $M(i,j) = 0$. This fact has been shown in [18] for the case when M is a binary matrix. While we have extended the proof to the weighted case, this discussion is beyond the space constraints of this paper and will be presented in a subsequent journal submission. We remark that since (1) is an NP-hard problem, depending on the initial condition an optimization algorithm used for solving (5) may converge to a local optima. To guarantee finding the global optima, it would be required to search the entire space of solutions.

Given a solution u of (5) with $d \geq n$, let $\mathcal{G}' \subseteq \mathcal{G}$ represent the subgraph that corresponds to the nonzero elements of u, and M' be the submatrix of M that corresponds to $\mathcal{G}'$. Noting that u, and therefore $\mathcal{G}'$, satisfies the constraints in (1), problem (1) reduces to binarizing u such that the objective is maximized. This is equivalent to the densest subgraph problem of (3), where the goal is to find $\mathcal{G}'' \subseteq \mathcal{G}'$ that has the maximum density. The densest subgraph problem can be solved in polynomial time using existing algorithms [26], but a good approximate answer can be obtained immediately by selecting the $\hat{\omega} \overset{\text{def}}{=} \text{round}(u^\top M\,u)$ largest elements of u as vertices of $\mathcal{G}''$. The justification follows from the well-known facts that $u^\top M\,u$, which is the spectral radius of $\mathcal{G}'$, is a tight upper bound for the graph's density [27] and nonzero elements of u, which form the principal eigenvector of M', represent centrality of their corresponding vertices, which is a measure of connectivity for a vertex in the graph [28].

IV. CLIPPER Algorithm

The CLIPPER algorithm consists of 1) obtaining a solution u of (5) via a projected gradient ascent approach with backtracking line search; and 2) estimating the densest cluster in u by selecting the $\hat{\omega}$ largest elements.

Algorithm 1 CLIPPER

```
1:  Input affinity matrix $M \in [0,1]^{n\times n}$ of $\mathcal{G}$
2:  Output $\mathcal{G}''$, densest component of feasible subgraph $\mathcal{G}' \subseteq \mathcal{G}$
3:  $u \leftarrow \text{rand}(n,1)$ % initialize with uniform random in $[0,1]$
4:  $d \leftarrow d_0$ % initialize d small
5:  while $d$ not large enough do
6:      calculate $M_d$ via eq. (6)
7:      while $u$ not converged do
8:          $\nabla F(u) = 2M_d u$
9:          $\nabla F_\perp(u) = (I - uu^\top)\nabla F(u)$ % orthogonal projection
10:         $\alpha = \min\{\alpha_i \overset{\text{def}}{=} |u_i/(\nabla F_\perp)_i| : (\nabla F_\perp)_i < 0, u_i > 0\}$
11:         $u \leftarrow u + \alpha\nabla F_\perp(u)$ % $\alpha$ via backtracking line search
12:         $u \leftarrow \max(u/\|u\|, 0)$ % project back onto $\mathbb{R}^n_+ \cap S^n$
13:     $d \leftarrow d + \Delta d$ % incrementally increase d
14: $\hat{\omega} \leftarrow \text{round}(u^\top M_d u)$ % estimate cluster size using max eig
15: $\mathcal{G}'' \leftarrow$ vertices corresponding to largest $\hat{\omega}$ elements of $u$
```

Algorithm 1 seeks a feasible subgraph by incrementally increasing the penalty parameter d (Line 13) and solving (5) via gradient ascent (Lines 7-12). Noting that any optimal solution u lies on the boundary of $\|u\| \leq 1$, the constraint manifold can be reduced to $\mathbb{R}^n_+ \cap S^n$, where S^n is the unit sphere. Instead of moving directly along the gradient $\nabla F(u) \overset{\text{def}}{=} 2M_d\,u$, we first project onto the tangent space of S^n at u (Line 9) and move according to this orthogonal projection $\nabla F_\perp(u)$. To move quickly in the search space, the step size α is chosen greedily so that if there is a u_i to be penalized (i.e., $\nabla F_\perp < 0$ and $u > 0$), a gradient step would cause u to hit the boundary of the positive orthant (Line 10). If no such u_i exists, then α is set such that the gradient update step gracefully degrades into a power iteration. In either case, if α is chosen too large, backtracking line search is used to find an appropriate step (Line 11). The solution is projected back onto the constraint manifold (Line 12) and gradient ascent continues. Once u has converged for the current value of M_d, d is incrementally increased (Line 13) and gradient ascent runs again with new objective $F(u) = u^\top M_d u$. This process continues until $d \geq n$ and u has converged. The convergence of the algorithm is guaranteed by the convergence property of the projected gradient approach [29].

The final step of CLIPPER selects the densest component of the subgraph $\mathcal{G}'$ (Lines 14-15), as explained in Section III-A. Since for all $M_d(i,j) = -d$ elements u_i, u_j in the solution u satisfy $u_i\,u_j = 0$, then $u^\top M\,u = u^\top M_d\,u$. The vertices of the densest component are then identified as the largest $\hat{\omega}$ elements of u.

A. Computational Considerations

Neglecting constant factors, the computational cost of Algorithm 1 is $\mathcal{O}(|E|)$ per iteration, where $|E|$ is the number of edges in the graph $\mathcal{G}$. Note that the orthogonal projection in Line 9 should be implemented as $\nabla F(u) - u\langle u, \nabla F(u)\rangle$, which costs $\mathcal{O}(|V|)$ operations, where $|V|$ is the number of vertices in the graph $\mathcal{G}$. Most of the time is spent computing the matrix-vector product $\nabla F(u) = M_d u$. All other operations run in at most $\mathcal{O}(|V|)$ operations.

V. 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. 1.

Point Clouds In Section II-A 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 [30].

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 $\{l_1, \dots, l_n\}$ and $\{l'_1, \dots, l'_m\}$, a simple invariant feature is the angle between pairs of lines, resulting in a geometric consistency score defined by $s(|\operatorname{acos} v_i^\top v_j - \operatorname{acos} v_i'^\top v_j'|)$. 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 [31], 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.

VI. EXPERIMENTS

We first present precision and runtime comparisons of CLIPPER as an approximate maximum clique solver on binary graphs. Then, we demonstrate CLIPPER's ability to perform data association in different outlier regimes and its strengths as a dense subgraph solver against the state of the art. Comparisons are performed against Leordeanu & Hebert [15] and Belachew & Gillis [18], which can also operate on weighted graphs. The exact parallel maximum clique (PMCx) solver [32] and its initial heuristic (PMCh) step are also compared against, as it has recently been used for data association in [6]. We include results for CLIPPER when given a binary graph (CLIPPERb). Experiments are run in MATLAB on an i9-7920, 64 GB RAM with a C++ interface to PMC. We allowed 12 threads for PMC.

A. Maximum Clique Finding in Synthetic Data

Given a data association problem and a scoring function $r(x)$ (see Fig. 2c), factors such as noise and outlier statistics result in consistency graphs ranging from dense to sparse. For example, in low outlier regimes, many associations are consistent and $\mathcal{G}$ is dense; conversely, in high outlier regimes with few consistent associations $\mathcal{G}$ is sparse. In this section, we evaluate CLIPPER against the state of the art as an approximate MC solver on graphs of varying sparsity. Graphs are generated from a complete graph that has increasingly more randomly selected edges removed according to the desired sparsity. Evaluations of precision and runtime are given across 50 Monte Carlo trials. We emphasize that in this synthetic analysis, we are disregarding the ability of CLIPPER to find dense subgraphs in weighted graphs.

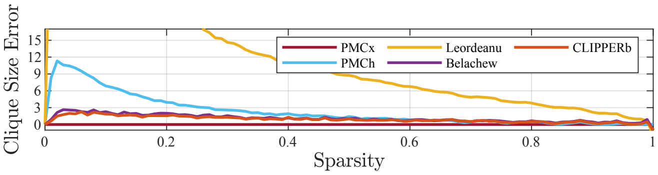


Fig. 3. Average clique size error of algorithms on binary graphs of size $n = 200$ across 50 Monte Carlo trials. PMC Exact recovers the true maximum clique at the cost of runtime. Leordeanu peaks in error with an underestimate of clique size by 57. CLIPPER and Belachew consistently achieve the lowest error.

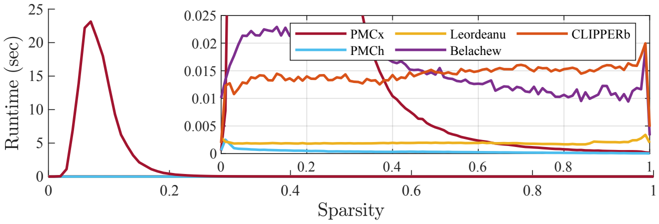


Fig. 4. Average algorithm runtime on binary graphs of size $n = 200$ across 50 Monte Carlo trials. When the input graph is not sparse, the exponential runtime of PMC Exact becomes clear. Compared to Belachew and PMC Exact, CLIPPER maintains a consistent runtime across all sparsity levels.

Fig. 3 shows the average clique size error of the approximate algorithms compared to the true clique size as provided by PMCx. Positive error indicates overestimation. In our evaluations, the graph size is limited to $n = 200$ due to the high runtime of PMCx (see Fig. 4). We observe that Leordeanu underestimates the clique size the most, while CLIPPERb and Belachew have the least average error across the entire sparsity range. An algorithm's ability to find the largest clique (or densest subgraph) in a consistency graph directly affects its precision and recall in data association.

Fig. 4 shows the corresponding runtime of each algorithm. While PMCx (using 12 threads) is fast for sparser graphs, its runtime peaks in the low sparsity range due to the NP-hardness of exactly recovering the maximum clique. As low graph sparsity corresponds to data association scenarios with few outliers, this runaway of runtime is problematic for a robot operating in varying conditions. PMCh and Leordeanu acheive the fastest overall runtime, but as observed in Fig. 3, underestimate the clique size the most on average. CLIPPERb and Belachew strike a balance of precision and runtime across the sparsity range of binary graphs, with CLIPPERb performing consistently around $15\,\mathrm{ms}$.

B. Data Association for Bunny Dataset

Using the Stanford Bunny [33], seen in Fig. 5, we evaluate CLIPPER as a data association algorithm in varying outlier regimes. The Bunny model is first scaled to fit in a $1\,\mathrm{m}$ cube

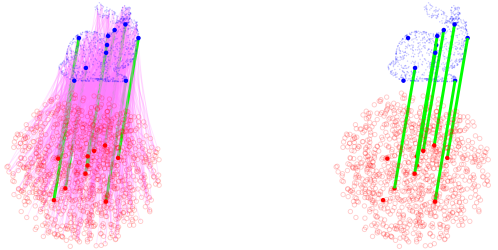


Fig. 5. Robust data association using CLIPPER on Stanford Bunny. Only 10 of 1000 blue points in view 1 were seen in view 2 (discs), simulating a partial view with 1000 additional outlier points (circles). (left) Putative associations between the two point clouds having 99% outlier associations, shown by magenta lines, and 1% inlier associations, shown by green lines. (right) CLIPPER removes all outlier associations with 80% recall in 138 ms.

and 1000 points are randomly sampled. These points are arbitrarily rotated and translated into a second view, where noise uniformly sampled from $[-1\,\mathrm{cm}, 1\,\mathrm{cm}]$ is added. For a model scale of $1\,\mathrm{m}$, this level of noise could reasonably be expected from a sensor. Additionally, 200 outlier points randomly drawn from a $1\,\mathrm{m}$ radius sphere are added in view 2 to simulate clutter. From the set of all-to-all associations, associations are randomly drawn according to the desired outlier ratio. This process creates putative associations that could have been generated from feature matching or, in the case of no prior information, an all-to-all hypothesis. Note that this is in stark contrast to ICP [34], [35], which performs poorly unless good initial information is provided.

To generate the consistency graph, pairwise consistency scores are computed as described in Section II-B. Weighted algorithms use the affinity matrix defined by $s(x) \overset{\text{def}}{=} \exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$ for $|x| \leq \epsilon$, and 0 otherwise. Binary algorithms use the affinity matrix defined by $r(x) \overset{\text{def}}{=} 1$ for $|x| \leq \epsilon$, and 0 otherwise. To study the effects of outliers, we choose $\epsilon = 8\,\mathrm{cm}$ and $\sigma = 3\,\mathrm{cm}$ small enough so that all algorithms have 100% precision in the 0% outlier regime.

By definition, precision $p \in [0, 1]$ is the ratio of correct associations to the total number of associations returned by an algorithm, and recall $r \in [0, 1]$ is the ratio of correct associations in an algorithm's output to the total number of associations in the ground truth. The best performance is achieved when both precision and recall are high. For many data association scenarios, precision is particularly important because a single outlier can have disastrous effects (e.g., loop closure in SLAM [4]). Precision, recall, and timing results vs outlier ratio are reported in Fig. 6, with precision/recall results called out in Table I. CLIPPER maintains 100% precision up to 90% outliers, leading in precision up to 99% outliers. We observe that CLIPPER and Leordeanu have a recall of 97% at 0% outliers, due to their use of the weighted scoring function $s(x)$. CLIPPER performs as expected in runtime, at about $150\,\mathrm{ms}$ on average. Belachew, however, takes significantly longer, requiring $10\,\mathrm{s}$ at 0% outliers and at least $1\,\mathrm{s}$ until a much sparser graph at 97% outliers.

A timing comparison is given in Fig. 7 with varying number of associations. The Bunny dataset is used with a randomly sampled outlier ratio in 0–10% and is averaged across 5 Monte Carlo iterations. While PMC quickly achieves the maximum clique for sparse problems (see Fig. 4), its runtime can increase without bound in large, dense problem domains.

TABLE I. Precision and recall of CLIPPER against the state of the art for various outlier ratios (OR). For each algorithm, results are reported as P/R. The highest precision for each OR is shown in bold. See Fig. 6.

OR	CLIPPER	CLIPPERb	Belachew	Leordeanu	PMCh	PMCx
0	**1.00**/0.96	**1.00**/1.00	**1.00**/1.00	**1.00**/0.96	**1.00**/1.00	**1.00**/1.00
0.70	**1.00**/0.97	0.99/1.00	0.99/1.00	0.99/1.00	0.99/1.00	0.99/1.00
0.80	**1.00**/0.97	0.99/1.00	0.99/1.00	0.96/1.00	0.99/1.00	0.99/1.00
0.90	**1.00**/0.98	0.97/0.99	0.97/1.00	0.81/1.00	0.97/1.00	0.97/1.00
0.95	**0.98**/0.99	0.91/1.00	0.91/1.00	0.53/1.00	0.79/0.47	0.91/0.99
0.97	**0.93**/1.00	0.83/0.99	0.84/1.00	0.32/0.97	0.51/0.34	0.82/0.99
0.99	**0.71**/0.98	0.44/0.76	0.55/0.98	0.04/0.38	0.16/0.22	0.55/0.98

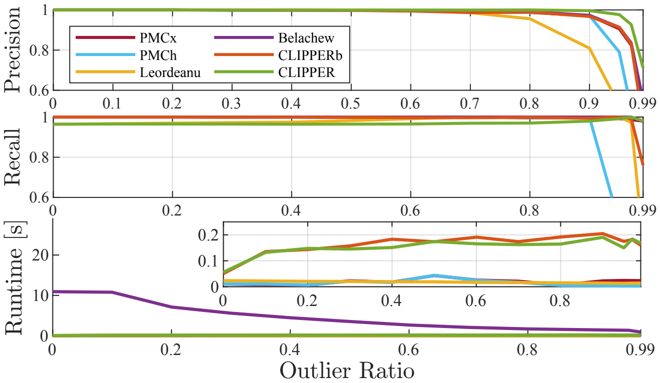


Fig. 6. Precision and recall for different outlier association ratios using $n = 1000$ randomly drawn associations from the Stanford Bunny dataset.

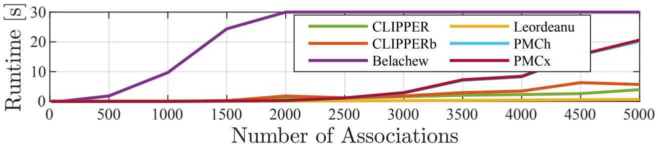


Fig. 7. Average runtime for increasing problem size with Bunny data in 0–10% outlier regime. As the size of the input graph increases, CLIPPER runtime remains below $4\,\mathrm{s}$ while PMC Exact and Heuristic both take up to $20\,\mathrm{s}$. Belachew was terminated early at $30\,\mathrm{s}$ if not finished.

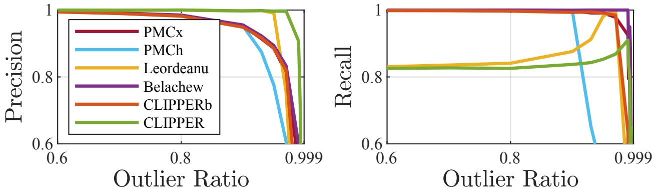


Fig. 8. Precision and recall of Bunny data association in high outlier regime. By leveraging CLIPPER's weighted graph formulation, the affinity scoring function can be shaped to trade off recall for precision.

Fig. 8 illustrates the benefit gained by having a weighted affinity matrix, which is the flexibility to shape the scoring function $s(x)$ as opposed to the rigidity of using $r(x)$. By varying the shape of $s(x)$, the user can trade off recall for precision in very high outlier regimes. Fig. 5 shows CLIPPER removing all outlier associations in this regime.

VII. CONCLUSION

We presented CLIPPER, a graph-theoretic framework for robust data association using the notion of geometric consistency. CLIPPER was shown to consistently execute with low runtime and to outperform the state of the art, achieving 100% precision, 80% recall in 99% outlier regimes. 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.

References

[1] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[2] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Annals of operations research*, vol. 14, no. 1, pp. 105–123, 1988.

[3] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "Door-slam: Distributed, online, and outlier resilient slam for robotic teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.

[4] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2916–2923.

[5] Á. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2868–2882, 2017.

[6] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, 2020.

[7] Y. Li, L. Gu, and T. Kanade, "Robustly aligning a shape model and its application to car alignment of unknown pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1860–1876, 2011.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[9] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3d multi-object tracking for autonomous driving," *arXiv preprint arXiv:2001.05673*, 2020.

[10] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4696–4704.

[11] T. Bailey, E. M. Nebot, J. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," in *IEEE International Conference on Robotics and Automation*, vol. 3, 2000, pp. 2512–2517.

[12] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone, "Outlier-robust estimation: Hardness, minimally-tuned algorithms, and applications," *arXiv preprint arXiv:2007.15109*, 2020.

[13] F. Wen, H. Wei, Y. Liu, and P. Liu, "Simultaneous consensus maximization and model fitting," *arXiv preprint arXiv:2008.01574*, 2020.

[14] F. Bernard, J. Thunberg, J. Goncalves, and C. Theobalt, "Synchronisation of partial multi-matchings via non-negative factorisations," *Pattern Recognition*, vol. 92, pp. 146–155, 2019.

[15] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1482–1489.

[16] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "SDRSAC: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 124–133.

[17] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *arXiv preprint arXiv:2003.07969*, 2020.

[18] M. T. Belachew and N. Gillis, "Solving the maximum clique problem with symmetric rank-one non-negative matrix approximation," *Journal of Optimization Theory and Applications*, vol. 173, no. 1, pp. 279–296, 2017.

[19] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *IEEE International Conference on Computer Vision*, 2009, pp. 1295–1302.

[20] A. P. Bustos, T.-J. Chin, F. Neumann, T. Friedrich, and M. Katzmann, "A practical maximum clique algorithm for matching with pairwise constraints," *arXiv preprint arXiv:1902.01534*, 2019.

[21] J. Shi, H. Yang, and L. Carlone, "Robin: a graph-theoretic approach to reject outliers in robust estimation using invariants," *arXiv preprint arXiv:2011.03659*, 2020.

[22] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *ICRA*. Kobe: IEEE, 2009, pp. 3212–3217.

[23] Q. Wu and J.-K. Hao, "A review on algorithms for maximum clique problems," *European Journal of Operational Research*, vol. 242, no. 3, pp. 693–709, 2015.

[24] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal, "A survey of algorithms for dense subgraph discovery," in *Managing and Mining Graph Data*. Boston, MA: Springer, 2010, pp. 303–336.

[25] S. Hosseinian, D. B. M. M. Fontes, S. Butenko, M. B. Nardelli, M. Fornari, and S. Curtarolo, *The Maximum Edge Weight Clique Problem: Formulations and Solution Approaches*. Cham: Springer International Publishing, 2017, pp. 217–237.

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

[27] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Application*. New York: Academic Press, 1980.

[28] G. Canright and K. Engø-Monsen, "Roles in networks," *Science of Computer Programming*, vol. 53, no. 2, pp. 195–214, 2004.

[29] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[30] S. Park, S.-K. Park, and M. Hebert, "Fast and scalable approximate spectral matching for higher order graph matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 479–492, 2013.

[31] A. M. Araújo and M. M. Oliveira, "A robust statistics approach for plane detection in unorganized point clouds," *Pattern Recognition*, vol. 100, p. 107115, 2020.

[32] R. A. Rossi, D. F. Gleich, and A. H. Gebremedhin, "Parallel maximum clique algorithms with applications to network analysis," *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. C589–C616, 2015.

[33] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[34] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[35] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.