

# Inexact Loops in Robotics Problems

Erik Nelson  
Nuro, Inc.

**Abstract**—Loops are pervasive in robotics problems, appearing in mapping and localization, where one is interested in finding loop closure constraints to better approximate robot poses or other estimated quantities, as well as planning and prediction, where one is interested in the homotopy classes of the space through which a robot is moving. We generalize the standard topological definition of a loop to cases where a trajectory passes close to itself, but doesn’t necessarily touch, giving a definition that is more practical for real robotics problems. This relaxation leads to new and useful properties of inexact loops, such as their ability to be partitioned into topologically connected sets closely matching the concept of a “loop closure”, and the existence of simple and nonsimple loops. Building from these ideas, we introduce several ways to measure properties and quantities of inexact loops on a trajectory, such as the trajectory’s “loop area” and “loop density”, and use them to compare strategies for sampling representative inexact loops to build constraints in mapping and localization problems.

## I. INTRODUCTION

The study of time-parameterized trajectories is common across many subfields of robotics. Nearly all robotics applications require estimating or forecasting the state of some system through its state space, whether that state includes the joint angles of a manipulator, the location and orientation of a mobile robot, a vector of weights in a machine learning model during training, or the pose of a nearby agent. Fundamental to the study of trajectories is the concept of a loop - a trajectory whose start and end points are the same.

While the standard topological definition of a loop is extremely useful in developing algorithms and theory, it fails to suit our needs in real robotics problems for several reasons. In practice, it is rare to have certainty that a path’s start and end points are identical. Sensor noise, tracking error, and sometimes deliberately added stochasticity lead to an inability to know a system’s state exactly, therefore making it impossible to tell whether that state is equal to its former self. Furthermore, trajectories with loops are simply not common on many of the high dimensional state spaces considered in robotics. For example, while in 2D a discrete random walk will return to its starting point with probability one [41], in 3D this probability is only 34% [31], leading to the expression “a drunk [person] will find their way home, but a drunk bird may get lost forever”. Is it intuitive that a random continuous 1D path embedded in an  $nD$  manifold will self intersect less frequently with larger  $n$ .

Due to a definitional void, we seek to describe a relaxed type of loop that admits a non-zero distance between its start and end points. The idea of an inexact loop is not novel; it is widely understood that a loop closure [19] in the context of Simultaneous Localization and Mapping (SLAM) [8] refers

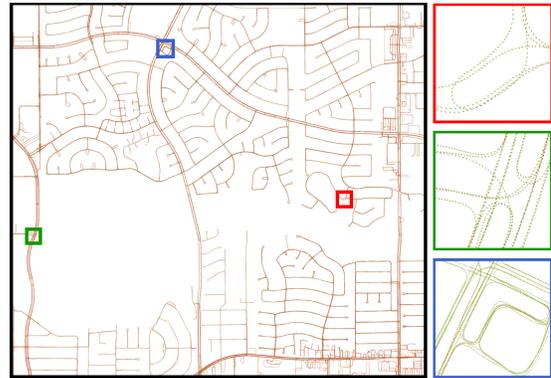


Fig. 1: Characterizing loops on trajectories in the real world can be challenging. A set of trajectories spanning road surfaces in an urban area are shown. Notably, no two points on the trajectories meet exactly in  $\mathbb{SE}(3)$ . Which parts are “loops”?

to this idea already. However, inexact loops have not been rigorously defined in a general way that captures trajectories over arbitrary smooth manifolds, and their properties have not been thoroughly studied. An exploration of the functions built en route to a general definition uncovers many new concepts that do not apply to exact loops.

A primary motivator for this research is the desire to better understand how “dense” a trajectory is with potential loop closure constraints when solving large scale SLAM problems on billions of variables, thereby identifying locations where constraints might be over or under sampled. Improving sampling algorithms that choose which loops to build constraints out of may allow us to achieve the same quality of reconstructions in less time, which is important when the number of possible pairs of discretely sampled points on a large set of trajectories is in the trillions to quadrillions. We therefore focus our discussion and examples on the mapping and localization setting, but note that the concepts introduced here are broadly applicable to planning, prediction, and other settings as well. For example, the machinery for clustering inexact loops may be useful in optimal exploration and path coverage problems, or in sampling based planning, where one wants to know if two trajectories that do not begin or end at exactly the same location belong to the same homotopy class [5].

An exemplary set of reconstructed trajectories covering road surfaces in a city area are shown in Fig. 1. While it is easy to identify pairs of points that are near one another, answers to the following are less clear:

- Which pairs of points constitute a single “loop”?
- How often is a particular point near the rest of the trajectory or a particular segment?

- How do we identify false negative detections of loops?
- How can we build a sample of point pairs that describes cases where the trajectory leaves and returns to itself?
- When is such a sample representative?

Our contributions are as follows. In Sec III, we provide a formal definition of an inexact loop and discuss its topological properties. In Sec. IV we introduce the concepts of loop duration, loop area, and loop density, which are measurements that can be used to analyze inexact loops on a trajectory. In Sec. V we compare methods for sampling inexact loops from a trajectory, showcasing their effects on reconstructed trajectories driven by robots on urban streets.

## II. RELATED WORK

Loops have been studied in a broad variety of topics, appearing generally in the study of algebraic topology, and specifically in any situation involving tracking the state of a system through its state space, e.g. in orbital mechanics, optimization, signal processing, quantum physics, and robotics. We split discussion of related work into the study of loops in mathematics and other fields, trajectory similarity in robotics contexts, and loops as they pertain to SLAM.

### A. Loops in Mathematics and Other Fields

The formalization of loops in mathematics dates back to at least the early 19th century with the introduction of Cauchy’s contour integrals [10]. In 1895, Poincaré established the idea of the fundamental group of a topological space by studying classes of loops that can be continuously deformed to one another [40]. This foundational idea aided in the classification of 2D closed surfaces, and ultimately nucleated the ongoing abstract study of homology [28]. While loops can be defined over any topological space [1], inexact loops require the addition of a metric on that space, pushing them into the realm of differential geometry where they have not been studied.

Outside of pure mathematics, loops in  $\mathbb{R}$  and  $\mathbb{Z}$  appear in the context of periodic functions within signal processing and quantitative finance [37, 47], and notions of distance on curved manifolds appear prevalently in geodesy [50], satellite tracking [12, 38], and physics [13].

### B. Trajectory Similarity in Robotics

There are many existing methods for comparing two trajectories, comparing a trajectory to itself, or clustering trajectories by similarity [29, 2, 4, 26]. Famous examples of these include Frechét distance [16], and the dynamic time warping algorithm [33]. Computing pairwise distances between trajectory points [46], a technique expanded upon here, is general enough to exist as a simple function call in some software packages [39]. Within planning, loops and distance on manifolds arise in the study of homotopy constraint satisfaction [5, 21].

### C. Loop Closing in SLAM

Loops and trajectories on curved manifolds are both heavily explored in the topic of SLAM. The identification [11, 18, 23] and long-term management [30, 9, 49] of loop closure constraints remain some of SLAM’s hardest problems [8]. Most

introductory SLAM literature does not define the concept of a loop rigorously, and typical explanatory diagrams feature simple exact loops in  $\mathbb{R}^2$ , leaning on practitioners to interpret what constitutes a loop in the wild. Several works investigate the problem of clustering and sampling loop closures [35, 22, 56], which we formalize here. Most closely aligned to this work is a recent investigation into proving the existence of loops on a given trajectory [43], which makes headway into some of the concepts introduced below restricted to exact loops on trajectories in  $\mathbb{R}^2$ .

## III. ANALYSIS AND PROPERTIES OF LOOPS

In the following section we introduce the concept of a loop from topology, and relax it in order to define inexact loops which are more suitable for robotics problems. This definition adheres to the standard mental model of a “loop closure”, which should in some way convey the fact that two separate parts of a trajectory were in close enough proximity to observe similar parts of the environment. However, the definition additionally unveils some interesting properties of inexact loops, such as the concept of a loop component, and the existence of simple and nonsimple inexact loops. Furthermore, this definition leads to several functions that can be used as intuitive tools for reasoning about inexact loops.

### A. Definitions

We begin by considering a continuous scalar-parameterized trajectory,  $\xi : [0, 1] \rightarrow \mathcal{M}$ , embedding a closed unit interval, which one usually interprets as an interval of time (shifted and scaled from some  $[t_0, t_f]$ ), into a smooth manifold  $\mathcal{M}$ . Common examples of manifolds that one might encounter in robotics problems include spheres ( $\mathcal{M} \simeq S^n$ ), tori ( $\mathcal{M} \simeq S^1 \times S^1 \times \dots$ ), real projective spaces ( $\mathcal{M} \simeq \mathbb{RP}^n$ ), and the translations ( $\mathcal{M} \simeq \mathbb{R}^n$ ), rotations ( $\mathcal{M} \simeq \text{SO}(n)$ ), and rigid body transformations ( $\mathcal{M} \simeq \text{SE}(n)$ ) of space in  $n = 2$  and  $n = 3$  dimensions. We use  $\mathcal{E}$  to refer to the trajectory’s corresponding *path* - the points on  $\mathcal{M}$  that  $\xi$  passes through.

$$\mathcal{E} = \{\xi(t) \mid t \in [0, 1]\} \subset \mathcal{M}. \quad (1)$$

Furthermore, we use  $\xi([t, t']) \subseteq \mathcal{E}$  to refer to the subset of the path covered from  $t$  to  $t'$ .

**Definition 1** ((Exact) Loop). Given a trajectory  $\xi$ , a *loop* is a continuous function,  $e : [t, t'] \subseteq [0, 1] \rightarrow \mathcal{E}$ , such that:

- $\forall x \in [t, t'], e(x) = \xi(x)$
- $e(t) = e(t')$

We will sometimes refer to these as *exact loops*. It is typical in topology for loops to be defined freely, in absence of a particular trajectory [1]. We instead use Definition 1 to identify the loops along a provided trajectory, noting that the definitions are identical when allowed arbitrary choice of  $\xi$ . The topological definition of a loop enforces that the trajectory exactly meets itself at some pair of times, which is relatively rare in real robotics problems. We therefore seek to relax this definition to consider cases in which the trajectory merely comes within close proximity of itself.

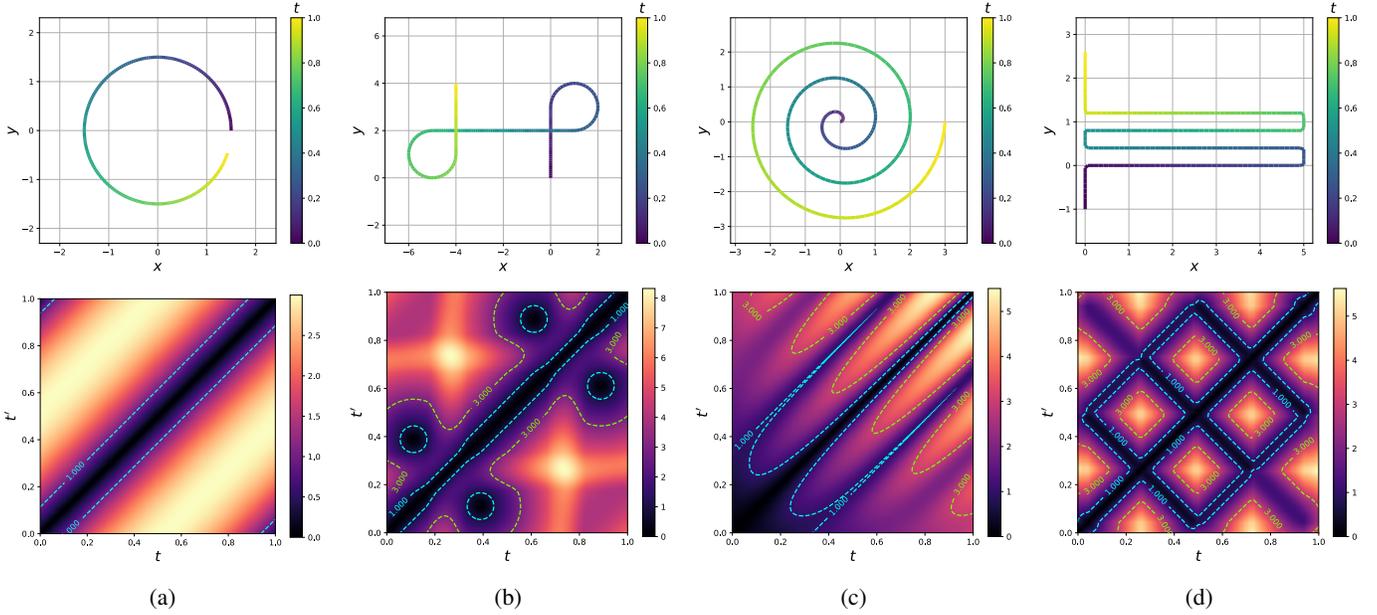


Fig. 2: Example trajectories  $\xi : [0, 1] \rightarrow \mathbb{R}^2$  (top), and their corresponding pairwise distance functions  $\pi_\xi(t, t')$  (bottom) under the  $\mathcal{L}_2$  metric, with  $\gamma = 1$  and  $\gamma = 3$  isocontours highlighted. Loop components correspond to the regions below these isocontours. Fig. 2a displays a 95% complete circle, highlighting the existence of the trivial loop component along the diagonal of  $\pi_\xi(t, t')$ , with one additional loop component near  $(0, 1)$  (mod symmetry). Fig. 2b demonstrates that inexact loops can merge into the trivial loop component as  $\gamma$  increases. Fig. 2c depicts an ambiguous case, where without a rigorous definition of an inexact loop it might be difficult to identify them. Fig. 2d displays case where the trivial loop component is multiply connected.

Given a manifold  $\mathcal{M}$ , one may introduce a *metric*, or distance function,  $\delta : \mathcal{M}^2 \rightarrow \mathbb{R}_+$ , which satisfies:

- $\delta(\mathbf{a}, \mathbf{b}) = 0 \iff \mathbf{a} = \mathbf{b}$  (identity)
- $\delta(\mathbf{a}, \mathbf{b}) = \delta(\mathbf{b}, \mathbf{a})$  (symmetry)
- $\delta(\mathbf{a}, \mathbf{b}) \leq \delta(\mathbf{a}, \mathbf{c}) + \delta(\mathbf{b}, \mathbf{c})$  (triangle inequality)

A metric  $\delta$  on  $\mathcal{M}$  allows one to compute distances between pairs of points. Importantly,  $\delta$  need not be a metric that evaluates distance between points directly, but might instead measure distance or quality of match between sensor measurements at two points [14, 52]. For a particular trajectory, we define the pairwise distance function  $\pi_\xi : [0, 1]^2 \rightarrow \mathbb{R}_+$ , which evaluates the metric at a given pair of times:

$$\pi_\xi(t, t') = \delta(\xi(t), \xi(t')). \quad (2)$$

$\pi_\xi$  inherits the symmetry property of  $\delta$ , hence  $\pi_\xi(t, t') = \pi_\xi(t', t)$ . Examples of trajectories on  $\mathbb{R}^2$  and their pairwise distance functions under the  $\mathcal{L}_2$  metric are shown in Fig. 2. We define the  $\gamma$ -sublevel set of a real-valued function  $f$  as

$$\mathcal{S}_\gamma(f) = \{\mathbf{x} \mid f(\mathbf{x}) \leq \gamma, \gamma \in \mathbb{R}\}, \quad (3)$$

and consider the set  $\mathcal{S}_\gamma(\pi_\xi) \subseteq [0, 1]^2$ , which contains all pairs of times for which the trajectory came within  $\gamma$  distance of itself. For brevity we will refer to this set as  $\mathcal{S}_\gamma$ , remembering that it references a particular metric and trajectory. Using these pieces, we are ready to define inexact loops.

**Definition 2 (Inexact Loop).** Given a trajectory  $\xi$ , metric  $\delta$ , and distance threshold  $\gamma \in \mathbb{R}_+$ , an *inexact loop* is a continuous function,  $l : [t, t'] \subseteq [0, 1] \rightarrow \mathcal{E}$ , such that:

- $\forall x \in [t, t'], l(x) = \xi(x)$ .
- $(t, t') \in \mathcal{S}_\gamma$ .

Inexact loops can be thought of as segments of the trajectory whose start and end points are within  $\gamma$  distance from one another. Definition 2 is nearly identical to Definition 1, except that instead of requiring  $\pi_\xi(t, t') = 0$ , we require  $\pi_\xi(t, t') \leq \gamma$ . This grants the additional flexibility to consider cases where a trajectory came close to itself, yet dissolves to the standard loop definition when  $\gamma = 0$ . For a given trajectory, inexact loops (which are functions), can be bijectively mapped to  $\mathcal{S}_\gamma$  via  $l \mapsto (\inf(\text{dom}(l)), \sup(\text{dom}(l))) = (t, t')$ , so for notational convenience, we refer to  $l$  as the 2D point  $(t, t')$ .

Next, we consider the partition of  $\mathcal{S}_\gamma$  into a disjoint union of  $k$  connected components:

$$\mathcal{S}_\gamma = \bigsqcup_{i=1}^k \Omega_i, \quad (4)$$

where two inexact loops,  $l$  and  $l'$ , belong to the same component  $\Omega$  if their time pairs are path-connected in  $\mathcal{S}_\gamma$ . In this case we say  $l \sim l'$ , and can describe the component as the equivalence class  $\Omega = [l] = [l']$ .

Each component is a compact set of pairs of times in which the trajectory was within  $\gamma$  distance of itself. For exact loops (i.e. when  $\gamma = 0$ ), the component  $[e]$  is almost always a singleton composed of one point - the loop  $e$  itself, and is therefore not interesting ( $[e]$  and  $e$  are not conceptually distinct). In rare cases the components can be 1D or 2D compact subsets

of  $[0, 1]^2$ , e.g. when the trajectory stops at exactly the point of intersection before continuing, or in the special case of  $[(t, t)]$ . By contrast, for inexact loops, the components are only singletons when the trajectory glances itself at a distance of exactly  $\gamma$  without stopping. Unlike exact loops, inexact loops are conceptually distinct from the components that they belong to, and so we give these components a name.

**Definition 3 (Loop Component).** Given a trajectory  $\xi$ , metric  $\delta$ , and distance threshold  $\gamma \in \mathbb{R}_+$ , a *loop component*  $\Omega \subseteq [0, 1]^2$  is a path-connected component of  $\mathcal{S}_\gamma$ .

In other words, a loop component is a compact path-connected set of inexact loops. Its path-connectedness can be pictured by imagining two points on a trajectory sliding independently. Two inexact loops belong to the same loop component if there exists a way of continuously sliding the endpoints from one configuration to the other without the pair of points separating by a distance more than  $\gamma$  (under the metric  $\delta$ ).

Loop components are useful because they give us a way to enumerate the distinct locations where a trajectory has inexact loops. The concept of a loop component is more in line with intuition and loose terminology used in mapping and localization settings. We are typically more interested in finding the set of loop components for a given trajectory, than in finding the set of all loops, or samples thereof. The former contains more information than the latter, since the set of loop components is a partition of the set of all inexact loops.

### B. Properties of Inexact Loops and Loop Components

We now discuss some properties of inexact loops.

Due to the symmetry of  $\pi_\xi$  and  $\delta$ , we should expect to find inexact loops in pairs. The simple mental picture of sliding two independent points along a trajectory makes this symmetry clear: fixing endpoint  $\xi(t)$  at a location and sliding endpoint  $\xi(t')$  leads to the same distance values as fixing endpoint  $\xi(t')$  at the same location and sliding endpoint  $\xi(t)$ . In other words, for a given inexact loop, one can form a different loop by flipping its direction. This can be observed in the plots of  $\pi_\xi$  in Fig. 2.

The identity property of  $\delta$  implies that for any choice of  $\gamma$  on any trajectory, the pair  $(t, t)$  is always an exact loop, and therefore belongs to some loop component. We refer to the loop component  $[(t, t)]$  as the *trivial loop component*, and refer to loops within it as *simple loops* (so as not to be confused with trivial loops in topology). The trivial loop component contains both exact and inexact loops whose endpoints can be continuously slid along the trajectory until they match (in both  $\mathcal{M}$  and  $[0, 1]$ ), without moving further than  $\gamma$  from one another. Fig. 2c displays a case in which for both  $\gamma = 1$  and  $\gamma = 3$ , all inexact loops are simple. Indeed, one can imagine sliding the points  $\xi(t)$  and  $\xi(t')$  from any starting configuration in which  $\pi_\xi(t, t') \leq 1$  to meet at a single point without ever moving 1 unit away from one another.

The sizes, shapes, and number of loop components are controlled by several variables. The parameter  $\gamma$  clearly plays

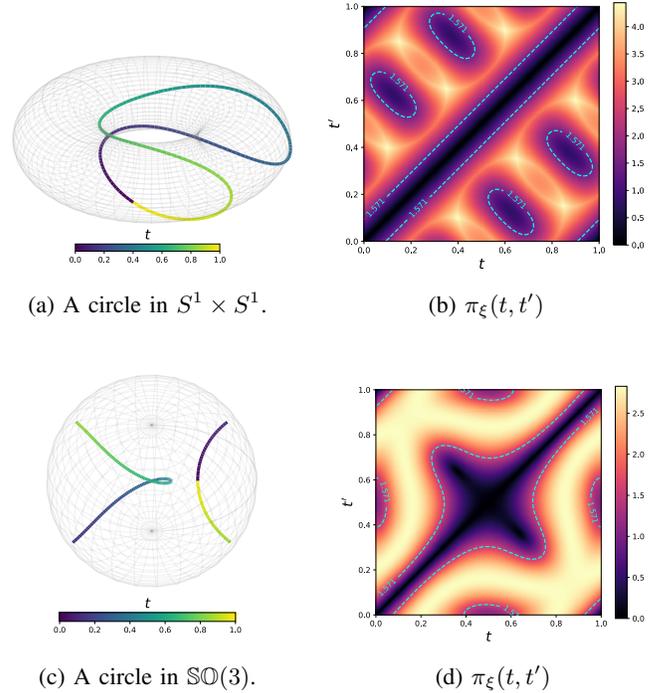
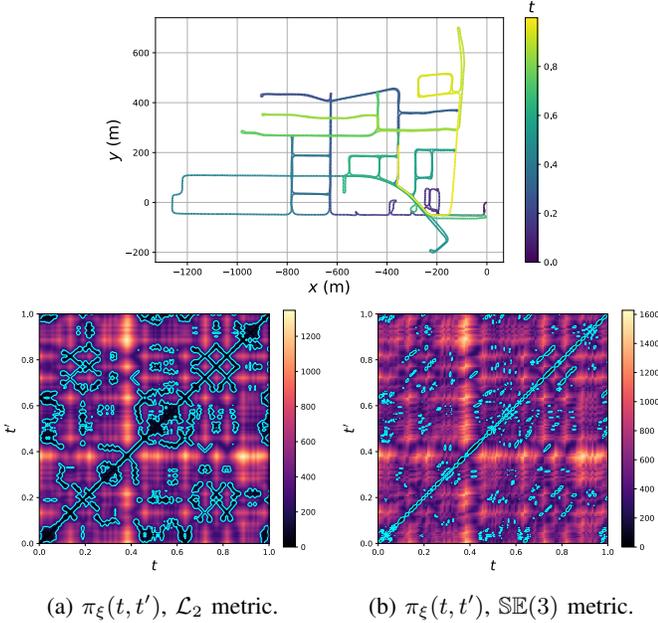


Fig. 3: Examples of circular trajectories with radius  $\frac{7}{8}\pi$  on two curved manifolds, which exhibit different loop components (a third circular trajectory in  $\mathbb{R}^2$  is shown in Fig. 2a for comparison). In  $S^1 \times S^1$ , we use the metric  $\delta((\theta, \phi), (\theta', \phi')) = \|\Delta(\theta, \theta'), \Delta(\phi, \phi')\|_2$ , where  $\Delta$  returns least angular distance. In  $\mathbb{S}\mathbb{O}(3)$  we use the metric  $\delta(\mathbf{p}, \mathbf{q}) = \|I - \exp(\mathbf{p}^\wedge) \exp(\mathbf{q}^\wedge)^\top\|_F$ , where points on the path are axis-angle vectors of magnitude  $\leq \pi$ .

a role in deciding these quantities. For any metric, as  $\gamma \rightarrow \infty$ , the trivial component  $[(t, t)]$  approaches  $[0, 1]^2$ , effectively merging with any other components. Fig. 2b displays an example in which increasing  $\gamma$  from  $1 \rightarrow 3$  causes all loop components to merge with the trivial component. These quantities additionally depend both on the manifold that the path is embedded in, as well as the path's location when the manifold is curved. Fig. 3 displays two trajectories whose paths carve out circles in the parameter space of two curved manifolds. On the torus  $S^1 \times S^1$ , the circle exactly meets itself at its start and end points, and in the process comes within close vicinity of itself two other times. On  $\mathbb{S}\mathbb{O}(3)$ , the circle approaches its own start and end points at their antipodal point.

Surprisingly, loop components need not be simply connected. This property is exhibited by the trivial loop component for the trajectory depicted in Fig. 2d (note that some exact loops on the space  $\mathcal{S}_\gamma$  itself cannot be shrunk to a point). This observation brings up the confusing discussion of the possible exact loops within the set of a trajectory's inexact loops. Because loop components are closed path-connected topological subspaces of  $[0, 1]^2$ , they are homeomorphic to the 2D disk punctured at  $n$  points. This implies that the fundamental group of a loop component itself can only be the free group on  $n$  generators for some  $n$ , which is rather uninteresting. Anecdotaly, data from real trajectories implies



(a)  $\pi_{\xi}(t, t')$ ,  $\mathcal{L}_2$  metric. (b)  $\pi_{\xi}(t, t')$ ,  $\mathbb{SE}(3)$  metric.

Fig. 4: A trajectory containing numerous inexact loops, with  $\gamma = 200$  isocontours highlighted for two metrics.

that it is common to encounter cases with  $n > 0$ . Fig. 4 shows a real trajectory taken on  $\mathbb{SE}(3)$ , where some loop components are multiply connected under the  $\mathcal{L}_2$  metric on translations, but not under an  $\mathbb{SE}(3)$  metric that incorporates rotation.

Given a loop component  $\Omega$ , we can consider the set of 2D points along its boundary  $\partial\Omega$ . This boundary traces out paths taken by pairs of points on the trajectory that stay a fixed distance of  $\gamma$  from one another. To visualize this, imagine a rigid bar of length  $\gamma$  connecting two points on the trajectory as they slide together. Similarly, for any closed subset  $\Lambda \subset [0, 1]^2$ , we can consider the boundary  $\partial\Lambda$ . Previous work has shown that one can identify exact loops by finding points in  $\mathcal{S}_0$  about which such a boundary has a non-zero winding number [43]. The same can be said for inexact loops in  $\mathcal{S}_{\gamma}$ .

#### IV. MEASURING INEXACT LOOPS ON TRAJECTORIES

It would be useful to have measurements that describe various loop-related properties of a trajectory, such as how many inexact loops begin or terminate within a particular segment, or how many inexact loops are connected to the average point. For example, in SLAM problems in which an approximate absolute initial estimate is available (e.g. from GNSS, or gravity readings), these measurements would help identify cases when loop closure constraints that should have been detected are missed, simply by examining the trajectory. False negatives of this nature can lead to large errors and should be avoided at all costs. Alternatively, these measurements could be used to determine when a particular loop component has too many loop closure constraints, leading to overconfidence in an estimated reconstruction of the trajectory.

To build towards these measurements, we consider measuring the total duration for which other parts of the trajectory

are within a distance  $\gamma$  of a point fixed at time  $t$ .

$$\tau_{\mathcal{X}}(t) = \int_0^1 \mathbf{1}_{\mathcal{X}}(t, t') dt'. \quad (5)$$

Here,  $\mathbf{1}_{\mathcal{X}} : X \rightarrow \{0, 1\}$  is the indicator function, returning 1 when its argument is inside the set  $\mathcal{X}$ , and 0 otherwise. Depending on the subset  $\mathcal{X} \subseteq [0, 1]^2$  in consideration,  $\tau_{\mathcal{X}}$  can be used to measure the duration of all nearby points on the trajectory ( $\mathcal{X} = \mathcal{S}_{\gamma}$ ), or just those falling within particular loop components ( $\mathcal{X} = \Omega_1 \cup \Omega_2 \cup \dots$ ). For example,

$$\tau_{\mathcal{S}_{\gamma}}(t) = \sum_{i=1}^k \tau_{\Omega_i}(t). \quad (6)$$

The quantity  $\tau_{\mathcal{X}}(t) \in [0, 1]$  has the same physical units as the domain of the trajectory (e.g. of time, or perhaps length), hence we refer to the measurement  $\tau_{\mathcal{S}_{\gamma}}$  as the *loop duration* of the trajectory at time  $t$ . Setting the lower limit of integration on the integral in Eq. (5) to  $t$  includes only inexact loops that begin at  $\xi(t)$ , while setting the upper limit of integration to  $t$  includes only loops that end there. Considered on the whole set  $\mathcal{X} = \mathcal{S}_{\gamma}$ , a loop duration  $\tau_{\mathcal{S}_{\gamma}}$  near zero implies that no other points on  $\xi([0, 1])$  are nearby  $\xi(t)$ , while a loop duration of one implies that all of  $\xi([0, 1])$  is in the vicinity of  $\xi(t)$ .

Integrating loop duration once again over a segment of a trajectory starting from a time  $a$  and ending at a time  $b$  gives us a measure of the total number of inexact loops that either begin or end along that segment, restricted to the set  $\mathcal{X}$ .

$$\begin{aligned} \alpha_{\mathcal{X}}(a, b) &= \int_a^b \tau_{\mathcal{X}}(t) dt = \int_a^b \int_0^1 \mathbf{1}_{\mathcal{X}}(t, t') dt' dt \\ &= \iint_{\mathcal{A}} dt' dt, \text{ with } \mathcal{A} = \mathcal{X} \cap [a, b] \times [0, 1]. \end{aligned} \quad (7)$$

This surface integral leaves us with a simple mental picture for computing the quantity  $\alpha_{\mathcal{X}}(a, b) \in [0, 1]$ : it is the area of the set  $\mathcal{X}$  between  $t = a$  and  $t = b$  in a trajectory's pairwise distance plot. Like loop duration, setting  $\mathcal{X} = \mathcal{S}_{\gamma}$  gives a measurement of the area of all loop components, while  $\mathcal{X} = \Omega$  measures only the area of a single loop component. Since the physical quantity  $\text{time}^2$  has no name, we refer to  $\alpha_{\mathcal{X}}(a, b)$  as the *loop area* of the trajectory along the segment  $\xi([a, b])$ . The same argument of setting limits of integration to  $t$  can be made for loop area as well, restricting the measurement to include only inexact loops that either begin or end on the segment.

Finally, we can define a trajectory segment's *loop density* by averaging its loop area over the segment duration:

$$\rho_{\mathcal{X}}(a, b) = \frac{\alpha_{\mathcal{X}}(a, b)}{b - a}. \quad (8)$$

Loosely speaking, for  $\mathcal{X} = \mathcal{S}_{\gamma}$ , this final measurement tells us the average number of inexact loops attached to each point on the segment. Put more concretely, it measures the average loop duration of points on the segment. Loop area and loop density can be measured over the entire trajectory, in which case they are equivalent to one another in value, but have different units (since  $\forall \mathcal{X} \in 2^{[0, 1]^2}$ ,  $\rho_{\mathcal{X}}(0, 1) = \alpha_{\mathcal{X}}(0, 1)$ ).

## V. SAMPLING INEXACT LOOPS

Armed with a definition for inexact loops and several tools for measuring them, we turn our attention to the more practical problem of building a representative sample of inexact loops from a provided trajectory. This problem finds relevance in SLAM and state estimation, where the goal is to reconstruct the trajectory taken by a system through its state space, given a set of noisy measurements constraining parts of the state. Sampling is important in this context because the number of inexact loops in a trajectory grows quadratically with time, i.e. in proportion to the loop area  $\alpha_{\mathcal{S}_\gamma}(0, 1)$ . A two hour long trajectory sampled at 10 Hz has close to three billion unordered pairs of sample points. Typical matching algorithms for generating constraints from a pair of sensor measurements, such as PnP [20], point cloud registration [45, 54], or specialized neural networks [11, 57], run on the order of tens to hundreds of milliseconds. Optimistically, densely matching all pairs would require close to one year on one CPU core.

### A. Sampling Complexity Classes

For the moment we sidestep discussion of *detection* of inexact loops, and consider sampling a representative subset as if we have the entire set  $\mathcal{S}_\gamma$  available. It is possible to imagine three distinct complexity classes of inexact loop sampling algorithms. These correspond to whether the number of inexact loops sampled is proportional to (a) loop area, a measure of time<sup>2</sup>, (b) loop density, a measure of time, or (c) the discrete number of loop components, a dimensionless measure (Fig. 5). Let  $\Psi = \{l_1, l_2, \dots\} \subset \mathcal{S}_\gamma$  be a finite sample of inexact loops beginning or ending on the segment  $\xi([a, b])$ . Let  $\kappa \leq k$  be the number of loop components intersecting  $[a, b] \times [0, 1]$ . These complexity classes yield the following numbers of samples:

$$\#\Psi \propto \sum_{i=1}^{\kappa} \alpha_{\Omega_i}(a, b) = \alpha_{\mathcal{S}_\gamma}(a, b) \quad (\text{quadratic})$$

$$\#\Psi \propto \sum_{i=1}^{\kappa} \rho_{\Omega_i}(a, b) = \rho_{\mathcal{S}_\gamma}(a, b) \quad (\text{linear})$$

$$\#\Psi \propto \sum_{i=1}^{\kappa} 1 = \kappa. \quad (\text{constant})$$

If  $\sigma : \mathcal{S}_\gamma \mapsto \Psi$  is a sampling algorithm, we say  $\sigma \in \mathcal{O}(\alpha)$ ,  $\sigma \in \mathcal{O}(\rho)$ , or  $\sigma \in \mathcal{O}(1)$  and refer to these as quadratic, linear, and constant complexity classes, respectively. It can be simpler to think of the classes  $\mathcal{O}(\alpha)$  and  $\mathcal{O}(\rho)$  in terms of the number of inexact loops that they produce per loop component  $\Omega$ , per sample time  $t$ . With this restriction,  $\sigma \in \mathcal{O}(\alpha)$  generates samples in proportion to the loop duration  $\tau_\Omega(t)$ , while  $\sigma \in \mathcal{O}(\rho)$  generates samples in proportion to a constant factor.

We remark that most existing loop closing strategies in SLAM literature belong to  $\mathcal{O}(\alpha)$ , including intelligent strategies that cluster detections into sets resembling loop components [35, 22], which still keep a number of constraints proportional to the loop area of each component.

### B. Representative Samples

What properties should a good inexact loop sampling algorithm exhibit? When is a sample  $\Psi$  representative of the

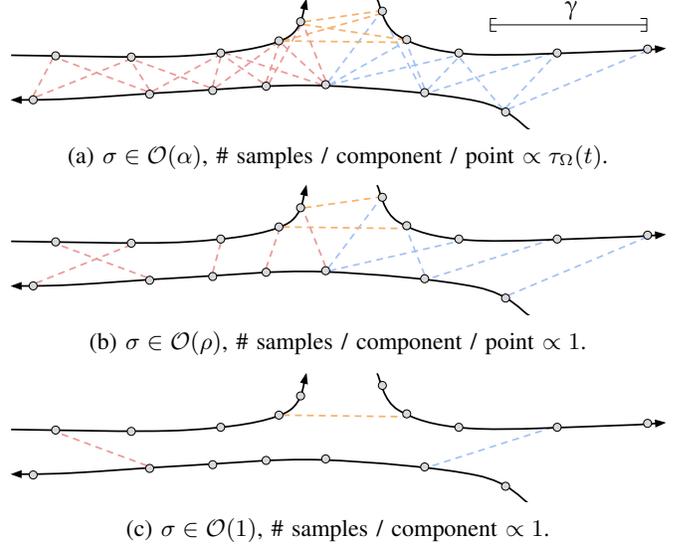


Fig. 5: Possible complexity classes for a loop sampling algorithm  $\sigma$ . Samples from three non-trivial loop components of a trajectory are displayed (red, orange, blue).

inexact loops on a trajectory? We propose the following desirable properties for a sampling algorithm  $\sigma$ :

- P1** Each unique loop component should be represented.
- P2** Similar inexact loops should be sampled infrequently.
- P3** Larger loop components should be sampled more often.

**P1** is useful because we would prefer not to miss any locations where the trajectory returns to itself. This property is nearly a requirement in SLAM problems, because as previously mentioned, false negative loop closure detections can lead to arbitrarily large errors in a reconstructed trajectory. It is beneficial to err on the side of high recall than high precision [17], since most modern SLAM back ends are robust against the presence of large fractions of outlier constraints [8, 51, 36]. **P2** (i.e. sample dispersion) is desirable of any spatial sampling algorithm [42]. Again in the context of SLAM, building two loop constraints that convey the same information violates the fundamental assumption of independence of measurements [48]. **P3** posits that large loop components generated when a trajectory remains near itself for a long period should be sampled more frequently than small loop components, where a trajectory nears itself only for a brief period.

Relating these desirable properties to the three sampling complexity classes, we see that **P3** diminishes the usefulness of algorithms in  $\mathcal{O}(1)$ , where inexact loops are sampled independently of the size of the component they belong to. Algorithms in  $\mathcal{O}(1)$  leave open the possibility of omitting important inexact loops in large loop components. These algorithms may still be useful in the case where one expects loop components to have similar areas. We also see that while all three complexity classes satisfy **P1**, common strategies such as random downsampling can violate this property. **P1** asserts that regardless of algorithm, one must always cluster detected inexact loops into loop components and keep at least one sample per component, in order to avoid false negatives.

This key algorithmic step is not present in many existing loop closing strategies. Finally, we see that **P2** and **P3** offer a trade off: if larger loop components are sampled more often, we obtain many inexact loops that convey similar information. Algorithms in  $\mathcal{O}(\rho)$  correspond to a preference for **P2**, because although they sample more in larger loop components, this growth is sublinear in loop area. Algorithms in  $\mathcal{O}(\alpha)$  sample in linear proportion to loop area, preferring **P3**.

### C. Sampling From Detections

To avoid false negatives, a sampling algorithm must necessarily identify loop components from its input. In the case that the input to  $\sigma$  is the full set  $\mathcal{S}_\gamma$ , detecting loop components amounts to finding connected components on a regular 2D lattice, where vertices represent inexact loops of the form  $(t, t')$ , and edges are built in a 4-neighborhood of each vertex. However, in many of the problem settings mentioned above, we are given a set of detections  $\mathcal{D} = \{l_1, l_2, \dots\} \subset \mathcal{S}_\gamma$  to sample from, rather than  $\mathcal{S}_\gamma$  itself (i.e.  $\Psi \subset \mathcal{D} \subset \mathcal{S}_\gamma$ ). In this case, we build a graph whose vertices are detected inexact loops  $l = (t, t')$ , and build edges between vertices whose  $\mathcal{L}_2$  distance is less than a threshold duration  $\epsilon$ . The connected components on this graph form a suitable approximation of the trajectory’s loop components. When  $\mathcal{D}$  is a sparse subset of  $\mathcal{S}_\gamma$ , a larger  $\epsilon$  will group more detections.

### D. Detecting Inexact Loops

We briefly return to the topic of detecting inexact loops on a trajectory in absence of a priori knowledge of  $\mathcal{S}_\gamma$ , surveying existing solutions in order of hardness. Given a perfectly estimated trajectory (e.g. when planning simulated trajectories), detecting inexact loops simply involves caching and querying for neighboring points within  $\gamma$ . These operations are supported in most  $O(1)$  [24, 34] or  $O(n \log n)$  [32] spatial indexes, which sometimes support metrics on curved manifolds [44, 6]. In this problem setting it is possible to build a discrete sample of the set  $\mathcal{S}_\gamma$  with no approximation.

In offline state estimation tasks where sensor measurements that constrain part of the state in an absolute sense are available, detecting inexact loops involves first building an initial erroneous reconstruction of the trajectory, and then performing the same steps as above, perhaps with an inflated  $\gamma$  to account for error. This case is applicable to settings such as warehouse robotics [3] or mapping for autonomous vehicles [7], where measurements from GNSS, an IMU sensing gravity, sensors that match their measurements against an existing map [25], or detections of predeployed environment markers [53] are available. Inexact loops detected in this manner are typically used afterwards to improve the estimate with additional constraints. In offline state estimation tasks where absolute measurements are not available, one must either increase  $\gamma$  with time to account for expanding odometry error [43], or find perceptually similar sensor measurements via techniques like visual hashing [55] and place recognition [27]. The latter category amounts to using a pseudometric  $\delta$  that measures distance in measurement space [14, 18, 52].

Inexact loop detection methods for online state estimation mirror their offline counterparts, except that their computation must be performed in real-time, fixing previous mistakes when new loops are detected so that the trajectory itself can be used for perception and planning. An additional tactic introduced in the online setting involves clustering detections [15] (sometimes into sets resembling loop components [35, 22]), to reduce computational overhead.

### E. Sampling Experiments

We conducted experiments using real trajectory data collected by a sensor-equipped robot on urban streets to compare benefits and drawbacks of sampling algorithms in the three separate complexity classes. Specifically, we wish to understand the behavior of algorithms in  $\mathcal{O}(\alpha)$  in comparison to the other two classes.  $\mathcal{O}(\alpha)$  contains most existing loop closing strategies, as well as the naïve strategy of randomly sampling inexact loops from a set of detections with no concern for which loop component they fall in, which we hypothesize has the potential to both miss loop components, and over sample large loop components (violating **P1** and **P2**).

Experiments were conducted in the offline state estimation setting with access to an approximately correct (up to 0.3 meters of absolute translation error) SLAM reconstruction of a 2.1 hour long trajectory in  $\mathbb{SE}(3)$ , with poses sampled at 10 Hz. This led to approximately  $7.6e4$  discrete timestamps from which inexact loops could begin or end. We began with implicit access to the entire  $\mathcal{S}_\gamma$  by indexing and radius searching for poses in a KD-tree. Since this set is unwieldy (up to a possible  $5.8e9$  elements,  $\sim 86$  GB using 64 bit timestamps), we randomly sampled to 1% to produce a set of detections  $\mathcal{D}$ , ensuring we retained enough that no loop components were omitted, and making the connected components algorithm tractable. We used the  $\mathcal{L}_2$  metric on  $\mathbb{R}^3$  pose translations,  $\gamma = 40$  m, and  $\epsilon = 30$  s. With these parameters the trajectory contained 490 loop components of varying loop areas. For sampling algorithms we used  $\sigma_1 \in \mathcal{O}(1)$ , which chose  $c$  samples randomly from each loop component,  $\sigma_\rho \in \mathcal{O}(\rho)$ , which kept  $r$  samples randomly per loop component per trajectory point, and  $\sigma_\alpha \in \mathcal{O}(\alpha)$ , which randomly sampled directly from  $\mathcal{D}$ , producing a number of samples proportional to  $\alpha_{\mathcal{S}_\gamma}(0, 1)$ .  $\sigma_1$  and  $\sigma_\rho$  intentionally kept at least one sample per loop component, while  $\sigma_\alpha$  did not run connected components at all, and was therefore unable to.

We allowed each algorithm a budget of exactly  $1e4$  inexact loop samples, forcing them to favor some loop components over others in relation to their complexity class. Fig. 6 shows the trajectory, as well as the inexact loops sampled by each algorithm. In regions containing loop components with large loop areas (Fig. 6a),  $\sigma_1$  under samples and omits relevant inexact loops. In regions containing loop components with small loop areas (Fig. 6b),  $\sigma_\alpha$  categorically fails to sample some components, leading to missed loop closures.  $\sigma_\rho$  adequately samples inexact loops in both cases.

The trajectory’s pairwise distance plot is depicted in Fig. 7, with loop components for  $\gamma = 40$  m outlined. A zoom in dis-

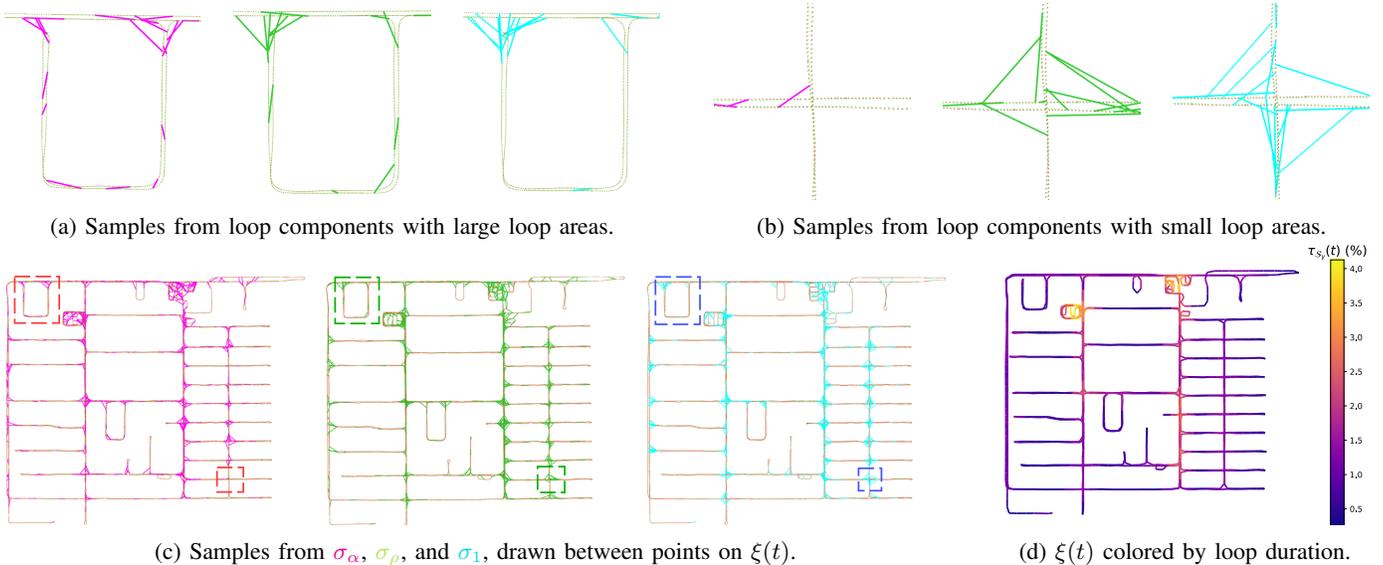


Fig. 6: A 2.1 hour trajectory through an urban area ( $\sim 1 \text{ km}^2$ ), with inexact loops sampled by algorithms from three complexity classes. Boxes highlight regions containing loop components with relatively high (Fig. 6a) and low (Fig. 6b) loop areas.

plays samples chosen by each algorithm in loop components of varying size, again revealing that  $\sigma_\alpha$  omits small components and  $\sigma_1$  under samples large ones (Fig. 7b). Fig. 8 sorts loop components by their area, then shows the percentage (top) and cumulative number (bottom) of possible detections in  $\mathcal{D}$  sampled from each component, showcasing the relationship between loop area and number of samples for each algorithm.

From these results, we readily suggest that other practitioners cluster inexact loops into loop components and retain at least one sample from each component, in order to avoid false negative loop detections. Furthermore, we suggest preferring sampling algorithms in the class  $\mathcal{O}(\rho)$  that retain a fixed number of inexact loops per point, per component, as they sample more from loop components with larger areas, while avoiding over sampling in some high density areas.

## VI. CONCLUSION

In this work we formally introduced the concept of an inexact loop by relaxing the standard mathematical definition of a loop to allow for non-zero distance between its start and end points, which is of general use in robotics problems. We explored the topological consequences of this relaxed definition, including the existence and properties of path-connected clusters of inexact loops we called loop components. We then introduced several tools to measure inexact loops on a given trajectory. Finally, we characterized the complexity classes of inexact loop sampling algorithms, and compared several representative algorithms on long trajectories collected by a robotic vehicle on urban streets.

We hope to see inexact loops and their related machinery find use broadly in robotics. Future work includes the application of inexact loops to the topic of motion planning, as well as a focused discussion on inexact loop sampling algorithms restricted to the difficult online SLAM setting.

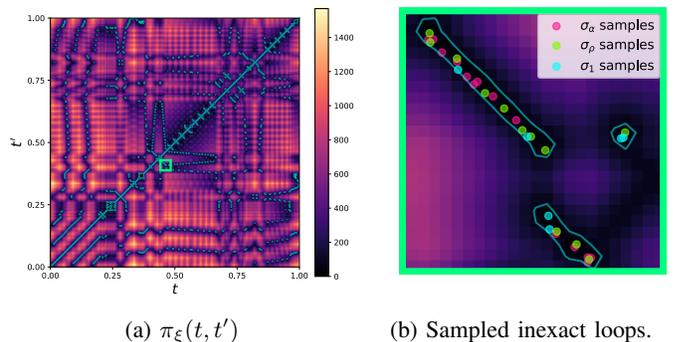


Fig. 7: Pairwise distance plot of experimental trajectory, and samples from  $\sigma_\alpha$ ,  $\sigma_\rho$ , and  $\sigma_1$  on three loop components with different areas ( $\gamma = 40 \text{ m}$  isocontours shown). Fig. 7b corresponds to the boxed region in Fig. 7a.

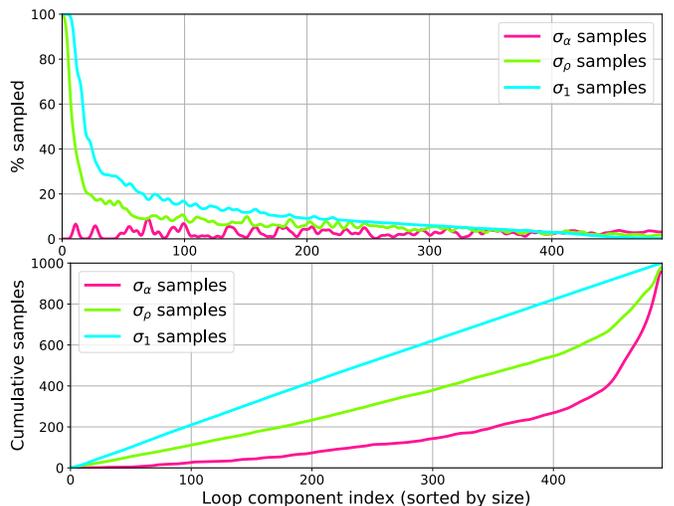


Fig. 8: Samples per loop component by algorithm.

## REFERENCES

- [1] John Frank Adams. *Infinite loop spaces*. Princeton University Press, 1978.
- [2] Aris Anagnostopoulos, Michail Vlachos, Marios Hadjieleftheriou, Eamonn Keogh, and Philip S Yu. Global distance-based segmentation of trajectories. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.
- [3] Patric Beinschob and Christoph Reinke. Graph slam based mapping for agv localization in large-scale warehouses. In *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 245–248. IEEE, 2015.
- [4] Philippe Besse, Brendan Guillouet, Jean-Michel Loubes, and Royer François. Review and perspective for distance based trajectory clustering. *arXiv preprint arXiv:1508.04904*, 2015.
- [5] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Topological constraints in search-based robot path planning. *Autonomous Robots*, 33(3):273–290, 2012.
- [6] Jose Luis Blanco and Pranjal Kumar Rai. nanoflann: a C++ header-only fork of FLANN, a library for nearest neighbor (NN) with kd-trees. <https://github.com/jlblancoc/nanoflann>, 2014.
- [7] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.
- [8] 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.
- [9] Nicholas Carlevaris-Bianco, Michael Kaess, and Ryan M Eustice. Generic node removal for factor-graph slam. *IEEE Transactions on Robotics*, 30(6):1371–1385, 2014.
- [10] Augustin Louis Baron Cauchy. *Cours d’analyse de l’École Royale Polytechnique*, volume 1. Imprimerie royale, 1821.
- [11] Xieyuanli Chen, Thomas Labe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, Cyrill Stachniss, and FKIE Fraunhofer. Overlapnet: Loop closing for lidar-based slam. In *Robotics: Science and Systems*, 2020.
- [12] Vladimir A Chobotov. *Orbital mechanics*. American Institute of Aeronautics and Astronautics, 2002.
- [13] Wendell D Curtis and Forrest R Miller. *Differential manifolds and theoretical physics*. Academic Press, 1985.
- [14] Vito Di Gesu and Valery Starovoitov. Distance-based functions for image comparison. *Pattern Recognition Letters*, 20(2):207–214, 1999.
- [15] Jing Dong, Erik Nelson, Vadim Indelman, Nathan Michael, and Frank Dellaert. Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5807–5814. IEEE, 2015.
- [16] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical report, Citeseer, 1994.
- [17] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [18] Arren Glover, William Maddern, Michael Warren, Stephanie Reid, Michael Milford, and Gordon Wyeth. Openfabmap: An open source toolbox for appearance-based loop closure detection. In *2012 IEEE International Conference on Robotics and Automation*, pages 4730–4735. IEEE, 2012.
- [19] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.
- [20] Radu Horaud, Bernard Conio, Olivier Le Boulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, 47(1):33–44, 1989.
- [21] James J Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, volume 4, pages 3993–3998. IEEE, 2004.
- [22] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph slam. *The International Journal of Robotics Research*, 32(14):1611–1626, 2013.
- [23] Yasir Latif, Guoquan Huang, John J Leonard, and José Neira. An online sparsity-cognizant loop-closure algorithm for visual navigation. In *Robotics: Science and Systems*, 2014.
- [24] Sylvain Lefebvre and Hugues Hoppe. Perfect spatial hashing. *ACM Transactions on Graphics (TOG)*, 25(3):579–588, 2006.
- [25] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. Map-based precision vehicle localization in urban environments. In *Robotics: Science and Systems*, volume 4, page 1. Citeseer, 2007.
- [26] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S Jensen, and Wei Wei. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 617–628. IEEE, 2018.
- [27] 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.
- [28] Saunders MacLane. *Homology*. Springer Science & Business Media, 2012.
- [29] Nehal Magdy, Mahmoud A Sakr, Tamer Mostafa, and Khaled El-Bahnasy. Review on trajectory similarity measures. In *2015 IEEE seventh international conference on Intelligent Computing and Information Systems (ICICIS)*, pages 613–619. IEEE, 2015.

- [30] Mladen Mazuran, Wolfram Burgard, and Gian Diego Tipaldi. Nonlinear factor recovery for long-term slam. *The International Journal of Robotics Research*, 35(1-3): 50–72, 2016.
- [31] WH McCrea and FJW Whipple. Random paths in two and three dimensions. *Proceedings of the Royal Society of Edinburgh*, 60(3):281–298, 1940.
- [32] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.
- [33] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
- [34] Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. Openvdb: an open-source data structure and toolkit for high-resolution volumes. *ACM SIGGRAPH 2013 Courses*, 2013.
- [35] Edwin Olson. Recognizing places using spectrally clustered local matches. *Robotics and Autonomous Systems*, 57(12):1157–1172, 2009.
- [36] 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.
- [37] Alan V Oppenheim, John R Buck, and Ronald W Schafer. *Discrete-time signal processing. Vol. 2*. Upper Saddle River, NJ: Prentice Hall, 2001.
- [38] Russell P Patera. General method for calculating satellite collision probability. *Journal of Guidance, Control, and Dynamics*, 24(4):716–722, 2001.
- [39] Edzer Pebesma, Benedikt Klus, Benedikt Graeler, Nikolai Gorte, Mehdi Moradi, and Pebesma. ‘trajectories’. *R Software Package*, 2020.
- [40] Henri Poincaré. *Analysis situs*. Gauthier-Villars Paris, France, 1895.
- [41] Georg Pólya. Über eine aufgabe der wahrscheinlichkeit-srechnung betreffend die irrfahrt im straßennetz. *Mathematische Annalen*, 84(1):149–160, 1921.
- [42] Brian D Ripley. Tests of randomness for spatial point patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(3):368–374, 1979.
- [43] Simon Rohou, Peter Franek, Clément Aubry, and Luc Jaulin. Proving the existence of loops in robot trajectories. *The International Journal of Robotics Research*, 37(12):1500–1516, 2018.
- [44] John T Sample and Elias Ioup. *Tile-based geospatial information systems: principles and practices*. Springer Science & Business Media, 2010.
- [45] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.
- [46] Zhanpeng Shao, YF Li, and Yao Guo. Motion trajectory recognition using local temporal self-similarities. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 102–107. IEEE, 2015.
- [47] Didier Sornette and Anders Johansen. Significance of log-periodic precursors to financial crashes. *Quantitative Finance*, 1:452–471, 2001.
- [48] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.
- [49] Yulun Tian, Kasra Khosoussi, Matthew Giamou, Jonathan P How, and Jonathan Kelly. Near-optimal budgeted data exchange for distributed loop closure detection. *arXiv preprint arXiv:1806.00188*, 2018.
- [50] Wolfgang Torge and Jürgen Müller. *Geodesy*. Walter de Gruyter, 2012.
- [51] Vasileios Tzoumas, Pasquale Antonante, and Luca Carlone. Outlier-robust spatial perception: Hardness, general-purpose algorithms, and guarantees. *arXiv preprint arXiv:1903.11683*, 2019.
- [52] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2593–2601, 2017.
- [53] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.
- [54] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 2020.
- [55] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.
- [56] Hao Zhang, Fei Han, and Hua Wang. Robust multimodal sequence-based loop closure detection via structured sparsity. In *Robotics: Science and Systems*, 2016.
- [57] Xiwu Zhang, Yan Su, and Xinhua Zhu. Loop closure detection for visual slam systems using convolutional neural network. In *2017 23rd International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE, 2017.