GRAPH SAMPLING FOR MAP COMPARISON

Jordi Aguilar
Universitat Politècnica de Catalunya
jordi.aguilar.larruy@estudiantat.upc.edu

Kevin Buchin
TU Eindhoven
k.a.buchin@tue.nl

Maike Buchin
Ruhr University Bochum
maike.buchin@rub.de

Erfan Hosseini Sereshgi *
Tulane University
shosseinisereshgi@tulane.edu

Rodrigo I. Silveira †
Universitat Politècnica de Catalunya
rodrigo.silveira@upc.edu

Carola Wenk *
Tulane University
cwenk@tulane.edu

ABSTRACT

Comparing two road maps is a basic operation that arises in a variety of situations. A map comparison method that is commonly used, mainly in the context of comparing reconstructed maps to ground truth maps, is based on *graph sampling*. The essential idea is to first compute a set of point samples on each map, and then to match pairs of samples—one from each map—in a one-to-one fashion. For deciding whether two samples can be matched, different criteria can be used. The total number of matched pairs gives a measure of how similar the maps are.

Since the work of Biagioni and Eriksson [1, 2], graph sampling methods have become widely used. However, there are different ways to implement each of the steps, which can lead to significant differences in the results. This means that conclusions drawn from different studies that seemingly use the same comparison method, cannot necessarily be compared.

In this work we present a unified approach to graph sampling for map comparison. In particular, we point out the importance of the sampling method (GEO vs. TOPO) and that of the matching definition, discussing the main options used precisely, and proposing better alternatives for both key steps in details. Furthermore, we provide a code base and an interactive visualization tool to set a standard for future evaluations in the field of map construction and map comparison.

1 Introduction

Many situations ask to compare different roadmaps, e.g., roadmaps reconstructed with different algorithms from the same data, or simplifications or generalizations of a given map. When comparing two roadmaps, one wants to take into account both the geometry and topology. Graph sampling was first introduced by Biagioni and Eriksson [1, 2] and Liu et al. [3] for comparing a reconstructed roadmap with a ground truth map. The basic idea is to first *sample* both roadmaps with points at a fixed distance, then *match* points on the two maps within a given distance threshold using a 1-to-1 matching, and finally use the number of matched and unmatched points to compute precision, recall, and F-scores.

These graph sampling scores have been used in many papers to evaluate map construction results [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. The method has proven useful, as it makes little assumptions on the roadmaps, and thus allows to compare a variety of immersed graphs, and is efficient to compute. However, the two key steps, sampling and matching, allow much freedom in their implementation, and the resulting scores vary greatly based on these. Indeed, Table 1 shows how two implementations of the graph sampling method, with the same settings, produce different values for precision, recall, and F-score[3] In the literature, the presented F-scores vary widely, as can be seen in Table 4 in Section 2.1. Hence we revisit the graph sampling method here.

*Partially supported by National Science Foundation grant CCF 1637576 and 2107434.
†Partially supported by projects PID2019-104129GB-I00/AEI/10.13039/501100011033 and Gen. Cat. 2017-SGR-1640.
[3]Sat2Graph's [11] TOPO code is available on Github and Biagioni's code [1, 2] was made available to us by James Biagioni.

Chicago	prec.	recall	F
Sat2Graph's TOPO [11]	0.947	0.353	0.514
Biagioni's [2, 1]	0.971	0.523	0.679

Table 1: Graph sampling scores computed by different implementations, with local sampling, 370 seeds, $r = 300m$, $d_{\max} = 15m$ and sampling interval $5m$ on *Biagioni*'s reconstructed map vs. *cropped Chicago (OSM)*.

1.1 Related work

There are several methods for comparing roadmaps. Many of them have been developed for determining the quality of map construction algorithms that construct maps from trajectory data or satellite imagery. And since roadmaps are immersed graphs, i.e., all vertices have associated locations and edges have associated curves in 2D or 3D, methods for comparing shapes and graphs are also available for comparing maps. See [12, 13, 14, 15] for surveys.

The path-based [16], shortest path-based [17], and traversal [18] distances represent each graph with paths and compare the paths, and thus measure connectivity to some extent. The Hausdorff distance [19] considers nearest neighbor assignments of points only, while the Fréchet distance requires establishing a homeomorphism between the graphs [20], however roadmaps are generally not homeomorphic. Less strict requirements on a roadmap between the two graphs are imposed by the weak and strong graph distances [21] and the contour tree distance [22, 14], but many variants are NP-complete. The local homology-based distance [23] compares the topological features in local neighborhoods by comparing locally computed persistence diagrams of the distance filtrations of the graphs. Edit distances, see e.g. [24], can also be defined, but are usually NP-complete. Methodology for locally evaluating map construction algorithms for hiking data trajectories has been provided in [25]. Graph sampling [1, 2, 3], the method discussed here, is -arguably- the most popular method for comparing two roadmaps.

2 Graph Sampling Methods

Graph sampling methods for map comparison typically have a simple structure. First, point samples are computed from each map, using some *sampling method*. Second, a matching between the point samples of each map is computed, according to a *matching rule*. Intuitively, the rule determines when two points should be identified as the same in both maps. Finally, the number of matched points is used to calculate one or more *scores*, typically precision and recall, which measure the proportion of points matched.

Hence the implementation of a graph sampling method involves two key decisions: a sampling method and a matching rule. Since there are multiple options for each, and they can have an important effect on the final scores, this section discusses each of them in detail. In the following, the two graphs to be compared are always denoted G and H.

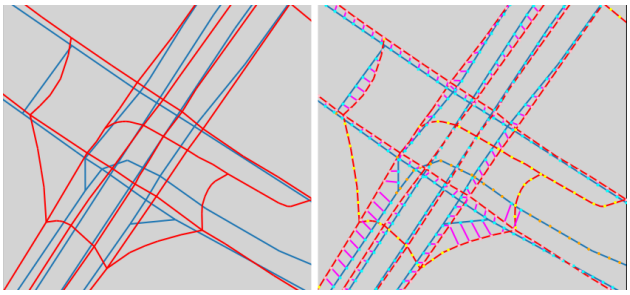


Figure 1: A highway intersection and its matching. Blue represents G and red is H

2.1 Sampling Method

The sampling method determines which points are sampled from each map. It is important that the sampling is dense enough to include all roads in the map, and that the number of samples along a road segment is proportional to its length. A simple way to achieve this is by sampling along each edge of the graph at a fixed distance between consecutive samples (as long as this distance is smaller than the minimum edge length). Some care must be taken at intersections, to ensure that the distance between consecutive samples is maintained across them as much as possible. Typically, the sampling is implemented using a graph traversal. This ensures that consecutive samples on paths from the root to the leaves are spaced at the fixed distance.

There are two major approaches to graph sampling:

(1) In *global* sampling, the roadmap G is sampled in its entirety with points at a fixed distance (typically 5m), resulting in a point set P_G sampled from G such that $|P_G|$ is proportional to $\text{len}(G)$. Here, $\text{len}(G)$ denotes the total length of all edges in G. The set P_G is a deterministic discretization of G. For the second graph H, the point set P_H is computed analogously.

(2) In *local* sampling, one proceeds in two phases. First, a set $S \subseteq \mathbb{R}^2$ of *seeds* is computed. Typically, S is chosen at random on G. Second, for each $s \in S$, the graphs $G \cap U_s$ and $H \cap U_s$ are sampled deterministically. Here, U_s is a neighborhood of s, usually a disk centered at s with a fixed radius r. Typically the sampling is performed using a graph traversal in $G \cap U_s$ starting at $s \in G$, and a graph traversal in $H \cap U_s$ starting at a nearest neighbor $s_H \in H$ to s, sampling points at a fixed distance.

Another important aspect of sampling is the *graph traversal*. The graph G can be interpreted as an undirected graph, or as a directed graph with edge directions and/or turn restrictions at vertices. (Not all roadmaps, in particular reconstructed ones, come equipped with edge directions or turn restrictions.) In addition, a traversal may traverse only a single connected component, or it may traverse every connected component. Actual roadmaps are of course (strongly) connected, but some reconstructed maps may not be connected. And in particular, local sampling may result in multiple connected components in smaller neighborhoods.

Global vs. Local Sampling

Global sampling is a deterministic sampling method, and for a fixed sampling distance and fixed graph traversal algorithm (in particular one that traverses all connected components), the sets P_G and P_H are uniquely determined. For a fixed matching rule (see Section 2.2), precision is $k/|P_G|$ and recall is $k/|P_H|$ (or vice versa), where k is the number of matched samples. The resulting graph comparison method, based on global sampling, has previously been termed GEO [1, 3].

Local sampling, on the other hand, introduces much more variability into the sampling process, and therefore the sample sets and the resulting scores are not well-defined. The choice and the number of the seeds pose the first problem. Table 2[4] shows an example where precision, recall, and F-scores vary widely for different numbers of random seeds. The precision values for the cropped ground truth, for example, vary between 0.702 and 0.938. If seeds are randomly chosen, some areas of the map may be oversampled, some undersampled; and it is not clear how many random seeds to choose. One way to alleviate this problem may be to choose seeds in a systematic way such that G or H or both are covered in a well-defined way; He et al. [7] for example compute seeds by sampling the ground truth map at a fixed distance of $50m$. One more caveat is how to tackle seeds in G that don't have a close enough sample $s_H \in H$. In this situation, seeds have been omitted from score calculation [1] or have been used for computing recall only [7]. Another source of variability in local sampling is the aggregation of the scores, see Section 2.3.

Biagioni [1]	Chicago			cropped Chicago		
# seeds	prec.	recall	F	prec.	recall	F
10,000	0.859	0.183	0.301	0.894	0.543	0.676
2,000	0.821	0.196	0.316	0.917	0.534	0.675
1,000	0.780	0.185	0.299	0.938	0.551	0.694
200	0.661	0.154	0.250	0.702	0.479	0.569
100	0.879	0.171	0.287	0.931	0.618	0.743

Table 2: Local evaluation with different number of seeds with $r = 300m$ and $d_{\max} = 15m$ on ***Biagioni***'s reconstructed map vs. OSM ground truth on ***Chicago*** data.

Biagioni	Chicago			cropped Chicago		
r	prec.	recall	F	prec.	recall	F
900	0.884	0.111	0.197	0.881	0.456	0.600
600	0.817	0.126	0.218	0.836	0.478	0.608
300	0.661	0.154	0.250	0.702	0.479	0.569
150	0.576	0.238	0.337	0.716	0.495	0.585
100	0.556	0.347	0.427	0.757	0.492	0.597
50	0.558	0.554	0.556	0.813	0.462	0.589

Table 3: Local evaluation with different radii r (in m), $d_{\max} = 15m$, and using 200 seeds on ***Biagioni*** vs. ***Chicago (OSM)***.

Local sampling was initially introduced [2] with the intent to measure topological differences between two roadmaps; Biagioni and Eriksson [1] called this method TOPO. For each seed $s \in S$, this graph comparison method only traverses one connected component in $G \cap U_s$ starting from s and one connected component in $H \cap U_s$ starting from s_H, and it uses edge directions and turn restrictions in G and H (as well as bearings and a greedy matching, see Section 2.2). So the only topological feature this method captures is local connectivity. It is, however, extremely sensitive to the definition of locality, i.e., the choice of the radius defining the local neighborhood U_s. See Table 3 for an example

[4] Using an adaptation of James Biagioni's graph sampling code to implement local sampling on undirected graph traversal, we achieve the results on Tables 2–4.

where precision, recall, and F-scores vary widely for different choices of radii. The precision numbers for the cropped ground truth, for example, vary between 0.702 and 0.881. It is not clear how this radius should be chosen in order to provide a useful comparison of local connectivity information. Intuitively the local neighborhood would need to be very small to even contain more than one connected component. In the literature, the choice of radii includes $100m$ [5], $300m$ [4, 1, 2, 7, 9, 12, 13][5], and a quite large value of 2,000m [8] which is 1/4 of the map diameter (for Chicago).

Due to the variability introduced by local sampling, and the limited (and not well-specified) benefit of comparing local connectivity, global sampling may be more beneficial to use in practice, since it is well-specified and reproducible.

	Global Sampling					Local Sampling				
	OSM		cropped OSM			OSM			cropped OSM	
Chicago	[8]	Ours	[10]	[1]	Ours	[8]	[12, 13]	Ours	[1]	Ours
Ahmed [26]		0.09	0.61		0.61		0.27	0.29		0.61
Biagioni [1]	0.24	0.07	0.78	0.78	0.64	0.58	0.35	0.25	0.78	0.57
Cao [27]	0.29	0.10		0.68	0.49	0.53	0.24	0.27	0.68	0.41
Edelkamp [28]	0.36	0.12		0.53	0.60	0.47	0.32	0.31	0.64	0.50
Karagiorgou [17]		0.08	0.82		0.70		0.27	0.28	0.27	0.71

Table 4: Varying F-scores comparing the same reconstructed maps in different papers for $d_{\max} = 15$; most values were visually transcribed from plots. All used $r = 300m$, except [8] used $r = 2,000m$. The number of seeds is 200 for [8] and ours, it is 100 for [1], and $1,000$ for [12, 13].

Graph Sampling Used in the Literature

Graph sampling scores have been used widely to evaluate map construction results [5, 6, 7, 8, 9, 10, 11, 12, 13]. Most use a $5m$ sampling interval and variants of local sampling. However, often not all parameters (e.g., r, number of seeds) or other choices (e.g., traversal, matching rule, score aggregation, map cropping method) are specified, affecting reproducibility, in particular for local sampling. Biagioni and Eriksson [1] use both global sampling (GEO [3]) and local sampling (TOPO [2] with directed road traversal), and they use a cropped ground truth. While the locality radius r and the number of seeds are not specified, in the code that James Biagioni made available to us the default values were $r = 300m$ and 100 random seeds, so we assume these parameter choices were made. Stanojevic et al. [8] also use both global sampling and local sampling (with $r = 2,000m$ and 200 seeds). Ahmed et al. [12, 13] use local sampling based on the code provided by James Biagioni (using $r = 300$) and do not crop the ground truth. They introduce the use of a fixed set of seeds for all comparisons in order to increase reproducibility; they use $1,000$ seeds. He et al. [7, 11] and Van Etten [9] use local sampling with $r = 300m$. Bastani et al. [6] also use local sampling; they present F-scores averaged over multiple cities, and they introduce a new score based on matching intersections. Chen et al. [5] use local sampling and take 1% of the GPS points of the input trajectories as seeds and $r = 100m$. Tang et al. [10] use a global approach to compute F-scores and manually cropped ground truth maps.

Even though graph sampling has been widely adopted as a method for comparing roadmaps, there is a large variability in the precision, recall, and F-scores in the literature. In Table 4 we show F-scores from different papers [8, 10, 1, 12, 13], including ours, that were computed on the same reconstructed maps[6] and OSM ground truth for Chicago. Most values were visually transcribed from plots in the papers, and may therefore contain some noise. The table includes F-scores for local and global sampling methods using (full) and cropped OSM ground truths and $d_{\max} = 15$. Our F-scores were computed using greedy matching, and local sampling parameters $r = 300m$ and 200 seeds. While all use OSM ground truth maps, only [10, 12, 13] and this article use the OSM maps from `mapconstruction.org`. The locality radius is $r = 300m$ for all, except for [8], it is $r = 2,000m$. The number of seeds is 200 for [8] and this paper, it is 100 for [1], $1,000$ for [12, 13]. It can be seen that the F-scores vary widely in each row. For example, for Biagioni's reconstructed map the local sampling scores vary between 0.25 and 0.58 for OSM, and between 0.57 and 0.78 for cropped OSM. The values for global sampling on cropped OSM are a bit more consistent – note that two approaches agree on 0.78 for Biagioni's map and two agree on 0.61 for Ahmed's map.

2.2 Matching Rule

The matching rule defines when a pair of points, one from each map, should be considered the same. Recall that a matching is a 1-to-1 correspondence (i.e., no point can be matched to two points). All matching rules include a distance

[5]This assumes [1, 2] used $r = 300m$ as in the code provided by James Biagioni.

[6]The trajectory data and reconstruction code are publicly available, e.g., at `mapconstruction.org`. However, reconstructed maps may still differ if parameters were set differently.

condition, establishing that only points that are closer than some maximum distance threshold $d_{\max}$ can be considered to match; this is the simplest possible rule. In principle, the more points that can be matched, the more similar the two maps will be considered.

Maximum Matching (MM) If the matching rule is only based on $d_{\max}$, the simplest approach is to match as many pairs of points as possible, as long as they are within distance $d_{\max}$. This is equivalent to finding a *maximum matching* in the bipartite graph whose vertices are the sampled points on each map, and whose edges are all pairs of points (from different maps) at distance at most $d_{\max}$.

Algorithm 1: Greedy Matching

Input : Set of samples $S_G \subseteq G$, and $S_H \subseteq H$, parameter k

Output : A 1-to-1 matching $M \subseteq S_G \times S_H$

$M_{init} = \emptyset$ *// Priority queue, sorted by matched distance*
// Create initial 1-to-many "matching"
for all $s_G \in S_G$ **:**
  s_H = closest among k-nearest neighbors of s_G that are within distance (and bearing) threshold
  Add (s_G, s_H) to M_{init}
// Convert to 1-to-1 matching, prioritizing shortest distances
while $M_{init} \neq \emptyset$ **:**
  (s_G, s_H) = M_{init}.*pop*() *// Pop closest pair*
  if s_H *not used*
    Add (s_G, s_H) to M; mark s_H as used
  else
    new_s_H = closest unused sample among k-nearest neighbors of s_G that are within distance (and bearing) threshold
    if new_s_H *found // If not found,* s_G *is discarded*
      Add pair (s_G, new_s_H) to M_{init}
return M

Greedy Matching While a maximum matching guarantees to match as many points as possible, it involves finding a global solution, which may be costly in large graphs. Also, all pairs within distance $d_{\max}$ are considered equivalent. Instead, one can find a locally maximal matching that is as large as possible, albeit possibly suboptimal, and gives priority to matching pairs of points that are close to each other. A *greedy matching* can be computed by choosing one point from one map, and matching it to the nearest point in the other map, if possible. If not, the second nearest point is tried, and so on, until the kth one (for a parameter k).

Unfortunately, the greedy matching is not clearly defined: there are multiple ways to implement it, leading to different methods. In particular, the order in which points are matched can result in very different matchings.

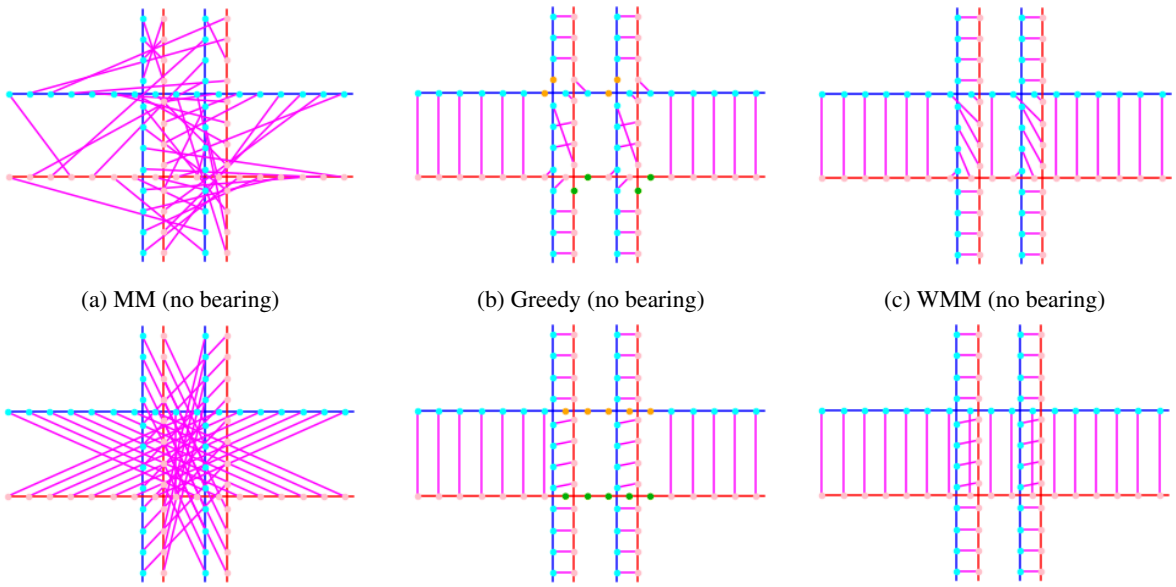
Algorithm 1 shows a greedy matching algorithm that follows the ideas in Biagioni's implementation of graph sampling as used in [1, 2]. It consists of two steps: First assign a nearest neighbor to each point. This produces an assignment that is not 1-to-1. In a second step, a 1-to-1 matching is greedily computed from this initial matching. Note that a point is only matched to one of its k nearest neighbors (typically, $k = 10$).

The greedy matching has two interesting properties: (i) it gives priority to matching points that are close to each other, as it tries to match closest pairs first. Moreover, (ii) it is more selective than the maximum matching: if none of the k nearest neighbors are available to match a point, the point is not matched. Thus one can expect fewer matched pairs with this method, but possibly *better* matched pairs.

Weighted Maximum Matching (WMM) We propose a new matching rule that combines the strongest points of the maximum and greedy matching. The idea is not only to try to match as many pairs as possible, but also to take the distance of each matched pair into account. We can formalize this as follows. We consider the same graph as in the maximum matching, but now each edge pq has a weight, defined as $d_{\max} - ||p - q||$, where $||p - q||$ is the Euclidean distance between p and q. The goal is now to compute a matching of maximum total weight, where the total weight of a matching is the sum of the weights of all edges in the matching.

The matching obtained may contain fewer edges than a maximum matching, but is expected to contain shorter edges. An important advantage of the weighted maximum matching is that it is unambiguously well-defined. Moreover, if no additional constraints are used, it produces matchings that are crossing-free (see Figure 2) which increases the accuracy. A disadvantage is that it requires a globally optimal solution, thus it can be computationally more expensive. Indeed, the best known methods to compute a weighted maximal matching have complexity $O(nm + n^2 \log n)$ [29], for n and m the number of vertices and edges, respectively. In our context, if $d_{\max}$ is small, one can expect m to be $o(n)$, or even constant.

Bearing Conditions. Matching rules can include other aspects in addition to distance. The most important one used in the literature is bearing. The idea is that two points should be matched only when they belong to edges with a similar orientation. The most common way to take it into account is to require that the angle between the two edges is at most



(a) MM (no bearing) (b) Greedy (no bearing) (c) WMM (no bearing)

(d) MM (with bearing) (e) Greedy (with bearing) (f) WMM (with bearing)

Figure 2: Example illustrating three different matching rules, without and with bearing. Two maps are compared, *map 1* with blue edges and *map 2* with red edges. A pair of matched samples is shown with a magenta segment between a sample in *map 1* (cyan) and a sample in *map 2* (pink). Unmatched samples in *map 1* are represented in orange, unmatched samples in *map 2* are represented in green. Sampling distance has been set to $5m$, and $d_{\max} = 50m$.

$45°$. A canonical example to motivate including bearing is to avoid matching two points that are very close to each other, but belong to edges that are perpendicular; in such a case, it is reasonable to argue that the points should not be considered the same, since their edges have opposite orientations.

Matching Rules Used in the Literature

All sampling based methods use some type of matching, but very few papers specify exactly how the matching is computed. In most cases, the description of the matching part only states that two points are matched whenever they are within the distance threshold (see, e.g., [5, 8, 10]), without explaining what is done when the nearest neighbor is already taken, which is often the case.

The exceptions that we are aware of are RoadRunner [7], that uses a maximum matching, and Biagioni and Eriksson [1, 2]—together with a few other papers that reused their code [12, 13, 4]—that implement greedy matching rules. The weighted maximum matching is proposed in this work for the first time. As for bearing, it is included in several papers [1, 2, 7, 8], although the exact bearing threshold used is not always mentioned (RoadRunner [7] uses $30°$).

Comparison of Matching Rules. Figure 2 presents a simple situation where each map has only three edges, shown in blue and red, respectively. Both maps are sampled in the same way (globally, using sampling distance $5m$). The resulting matchings are shown for the three matching rules (maximum matching MM, weighted maximum matching WMM, and Greedy) with two variations: with and without bearing.

Already in the first row, we can observe striking differences between the three matching rules. Maximum matching, as expected, matches at least as many points as the other rules, but at the cost of including pairs that visually do not seem to correspond to each other. In contrast, the two rules that give priority to shorter edges (WMM, Greedy) produce correspondences that are much more aligned with intuition. Note that the greedy matching fails to match some points around the intersections of the map edges. This can be explained by the fact that it is limited to matching among the 10-nearest neighbors. Using such a hard constraint can lead to being too selective in situations like the one shown.

The first row also shows that only taking distances into account can result in matching points that belong to clearly different edges. That is the case in the figure with matchings between horizontal and vertical edges. The second row, that restricts matching pairs to those with bearing difference of less than $45°$, solves this issue. This justifies the inclusion of bearing restrictions in the matching rules.

2.3 Score Calculation

Precision and recall are the two scores typically used to quantify the results of graph sampling methods. In this context, precision is the number of matched samples divided by the total number of samples on H (typically the reconstructed map). Recall is the number of matched samples divided by the total number of samples on G (typically the ground truth map). They are useful to measure the ratio of correct predictions and the ratio of covered ground truth, respectively. These two scores are often combined using the F-score, defined as the harmonic mean of precision and recall (i.e., $F = 2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall})$).

As mentioned in Section 2.1, in global sampling, precision, recall, and F-score are computed from matched samples taken over the entire graphs. In local sampling, however, there are different options for aggregation. The number of matched samples and total samples can be aggregated (summed) over all seeds, and precision, recall, and F-score computed using those total number of samples. Or, precision, recall, and F-score can be computed for each seed individually, and then aggregated in some way, e.g., by taking the mean. While it is reasonable to use such local aggregation in combination with local sampling, it does add extra variability to the computation, which should be clearly specified when presenting results. Moreover, unless exactly the same aggregation is used, results will not be comparable across different works.

Cropping the Ground Truth Map In the context of map reconstruction, the recall values can easily become distorted if the ground truth map used is not appropriate for the reconstructed map. Often, the ground truth map used is significantly larger than the reconstructed map, including roads that are not covered in the GPS dataset. This causes a dilution in the recall value, which also affects F-scores. One way to overcome this situation is to crop the ground truth such that it only contains the roads traversed by GPS trajectories. This can be done manually (see, e.g., [10]) or using map-matching algorithms (e.g., as in [1]). As it can be seen in Tables 2 and 3, the difference in recall between cropping the ground truth or not is significant. However, the use of a cropped ground truth adds an extra level of variability to the experiments, since there are various methods and settings to choose from, making the experiments unlikely to be reproducible if the method used is not specified in full detail (something that seldom occurs in the literature). It is also possible to overcome this problem by obtaining the number of matched samples. When working without a reliable ground truth, using the number of matched samples instead of the recall and the F-score avoids having to compare near-zero and unrealistic recalls.

2.4 Graph Sampling Toolkit

The Graph Sampling Toolkit consists of three components: the core is the graph sampling evaluation program. Additionally, there are tools for cropping maps and an interactive visualization program. The toolkit is available on Github: `https://github.com/Erfanh1995/GraphSamplingToolkit`

3 Discussion / Conclusion

Local sampling does not preserve topology, introduces many choices and parameters and thus the resulting scores are much less reproducible (see Table 4) than those computed with global sampling. Global distance on the other hand, is uniform and reproducible, hence would be a suitable choice for future evaluations. However, as has been done in [16, 23], local sampling can be used to visualize local differences by plotting heatmaps of all computed scores.

While graph sampling is an effective approach for map comparison, it is still a discrete method. A feasible continuous method might be the key to achieving more comprehensive results.

References

[1] J. Biagioni and J. Eriksson. Map inference in the face of noise and disparity. In *Proc. 20th ACM SIGSPATIAL GIS*, pages 79–88, 2012.

[2] J. Biagioni and J. Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, 2291:61–71, 2012.

[3] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu. Mining large-scale, sparse GPS traces for map inference: comparison of approaches. In *Proc. 18th ACM SIGKDD*, pages 669–677, 2012.

[4] K. Buchin, M. Buchin, J. Gudmundsson, J. Hendriks, E. Hosseini Sereshgi, V. Sacristan, R.I. Silveira, F. Staals, and C. Wenk. Improved map construction using subtrajectory clustering. In *Proc. 4th ACM SIGSPATIAL LocalRec Workshop*, pages 5:1–5:4, 2020.

[5] C. Chen, C. Lu, Q. Huang, Q. Yang, D. Gunopulos, and L. Guibas. City-scale map creation and updating using GPS collections. In *Proc. 22nd ACM SIGKDD*, page 1465–1474, 2016.

[6] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D.J. DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Salt Lake City, UT, USA, June 18-22*, pages 4720–4728. IEEE Computer Society, 2018.

[7] S. He, F. Bastani, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, and S. Madden. Roadrunner: improving the precision of road network inference from GPS trajectories. In *26th ACM SIGSPATIAL GIS*, pages 3–12, 2018.

[8] R. Stanojevic, S. Abbar, S. Thirumuruganathan, S. Chawla, F. Filali, and A. Aleimat. Robust road map inference through network alignment of trajectories. In *Proc. 2018 SIAM Int. Conf. Data Mining*, pages 135–143, 2018.

[9] A. Van Etten. City-scale road extraction from satellite imagery v2: Road speeds and travel times. In *IEEE Winter Conference on Applications of Computer Vision, WACV CO, USA, March 1-5*, pages 1775–1784. IEEE, 2020.

[10] J. Tang, M. Deng, J. Huang, H. Liu, and X. Chen. An automatic method for detection and update of additive changes in road network with GPS trajectory data. *ISPRS Int. J. Geo Inf.*, 8(9):411, 2019.

[11] S. He, F. Bastani, S. Jagwani, M. Alizadeh, H. Balakrishnan, S. Chawla, M.M. Elshrif, S. Madden, and M.A. Sadeghi. Sat2graph: Road graph extraction through graph-tensor encoding. In *Proc. 16th European Conference on Computer Vision*, volume 12369 of *Lecture Notes in Computer Science*, pages 51–67. Springer, 2020.

[12] M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk. *Map Construction Algorithms*. Springer, 2015.

[13] M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk. A comparison and evaluation of map construction algorithms. *Geoinformatica*, 19(3):601–632, 2015.

[14] M. Buchin, E. Chambers, P. Fang, B.T. Fasy, E. Gasparovic, E. Munch, and C. Wenk. Distances between immersed graphs: Metric properties, 2021. In preparation.

[15] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 18(3):265–298, 2004.

[16] M. Ahmed, B.T. Fasy, K.S. Hickmann, and C. Wenk. Path-based distance for street map comparison. *ACM Trans. Spatial Algorithms and Systems*, page 28 pages, 2015.

[17] S. Karagiorgou and D. Pfoser. On vehicle tracking data-based road network generation. In *Proc. 20th ACM SIGSPATIAL GIS*, pages 89–98, 2012.

[18] H. Alt, A. Efrat, G. Rote, and C. Wenk. Matching planar maps. *Journal of Algorithms*, pages 262–283, 2003.

[19] H. Alt and L.J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation - a survey. In *Handbook of Computational Geometry*, pages 121–154. Elsevier, 1999.

[20] P. Fang and C. Wenk. The Fréchet distance for plane graphs. In *European Workshop on Computational Geometry*, pages 62:1–62:5, 2021.

[21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, and C. Wenk. Distance measures for embedded graphs. *Computational Geometry: Theory and Applications*, page 101743, 2021.

[22] K. Buchin, T. Ophelders, and B. Speckmann. Computing the Fréchet distance between real-valued surfaces. In *Proc. 28th. Ann. ACM-SIAM on Discrete Algorithms*, pages 2443—-2455, 2017.

[23] M. Ahmed, B.T. Fasy, and C. Wenk. Local persistent homology based distance between maps. In *Proc. 22nd ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, pages 43–52. ACM, Nov. 2014.

[24] O. Cheong, J. Gudmundsson, H. Kim, D. Schymura, and F. Stehn. Measuring the similarity of geometric graphs. In *International Symposium on Experimental Algorithms*, pages 101–112. Springer, 2009.

[25] D. Duran, V. Sacristán, and R.I. Silveira. Map construction algorithms: a local evaluation through hiking data. *GeoInformatica*, 24:633–681, 2020.

[26] M. Ahmed and C. Wenk. Constructing street networks from GPS trajectories. In *Proc. 20th Ann. European Symp. on Algorithms*, pages 60–71, 2012.

[27] L. Cao and J. Krumm. From GPS traces to a routable road map. In *Proc. 17th ACM SIGSPATIAL GIS*, pages 3–12, 2009.

[28] S. Edelkamp and S. Schrödl. Route planning and map inference with global positioning traces. In *Computer Science in Perspective*, pages 128–151. Springer, 2003.

[29] H.N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of the First Annual ACM-SIAM, CA, USA*, pages 434–443. SIAM, 1990.