Utrecht University

Quantitative evaluation of trajectory based road map construction algorithms

Author:
Ari Saadon (5495741)
ari._s@hotmail.com

Supervisor:
Dr. Frank Staals
f.staals@uu.nl

Second supervisor:
Dr. Maarten löffler
m.loffler@uu.nl

A thesis submitted for the Master of Science degree

Department of Information and Computing Science
May 10, 2021

Abstract

In recent years there is a positive trend in the use of quantitative evaluation to assess performance of trajectory based road map construction algorithms. But little effort has been made in to how such an evaluation should be performed. This thesis addresses this problem with the aim of determining how to perform a proper quantitative evaluation. We discern two problems with the current evaluation methodology. The first problem being that the same few public test cases are being used throughout the literature. These test cases only give insight of algorithm performance under very specific circumstances and are provided with ground truth maps that contain roads impossible to be inferred. The second problem involves the evaluation measure with which a similarity is determined between the ground truth and a constructed map. The graph sampling similarity measure introduced by Biagioni and Eriksson [1] is mostly used in the literature. However, little effort has been made in assessing its evaluation performance. To address these problems, we propose a new evaluation framework using synthetically generated trajectories, with which we can investigate algorithm performance under a variety of circumstances. We additionally provide a working public implementation and use it to evaluate two state-of-the-art map construction algorithms. Finally, for the graph sampling measure, we perform a thorough analysis using multiple experiments. By doing so we provide insight in its strengths, shortcomings and proper usage.

Acknowledgement

First and foremost I want to thank Frank for his time and guidance during the many meetings we had. Through the long discussions, his insight helped me time and time again. And I have no doubt his teachings will continue to help me in the future for any direction I take.

I want to thank Maarten for his time and supervision as well. His generous flexibility in the final phase, gave me the much needed time to finish up the thesis.

Lastly, I want to thank my family and friends for their support during the research and corona crisis. Their help made the many days I had to work in isolation much more bearable.

Contents

1 Introduction

Road maps are an integral part of our lives, with many people and industries relying on them daily. A challenge lies not only in creating road maps for uncharted areas, but also in keeping our current maps up to date. Numbers indicate that each year up to 15% of all roads change [2]. While quality maps are predominantly made and updated using field survey data and manual input, new advancements in geographic technologies have provided possibilities to generate road maps automatically. Not only can this improve the quality of our current maps. But it would also diminish the need for labor intensive and cost heavy surveying, which can make creating road maps economically viable for yet uncharted areas.

As a result, we have seen in recent years a strong influx of research aiming to construct road maps automatically. Two popular methodologies have emerged. The first approach makes use of high quality satellite imagery and tries to infer roads from visual data. These efforts go hand in hand with the recent advancements in deep neural networks. Though results are promising [3, 4], the technique is limited by processing times, impossible road speed inference, view obstructions and weather conditions [5]. The second approach, the one we are focusing on, makes use of the strong increase in GPS capable mobile devices and tries to infer roads from collected trajectory data. Although not limited by the same problems of the first approach, GPS interference errors in the trajectories are known to make this method challenging. Numerous algorithms and strategies have been devised for this purpose [6, 1, 7, 8, 9, 10].

While there is still room for improvement of these trajectory based algorithms, a more pressing issue is the lack of proper evaluation of their performance. Without proper evaluation, no consensus can be formed about the current top performance. Proposed algorithms in the literature were initially only accompanied by qualitative evaluation. This typically involved visually comparing a constructed map with ground truth of the area in question. Ground truth usually being some satellite imagery or existing maps. Only in recent years a positive trend has occurred, where presented algorithms are also accompanied by a brief quantitative analysis. This involves constructing a map on one or two well known trajectory sets, followed by calculating a similarity measure between the constructed road map and a ground truth road map. The resulting similarity is then compared against a few other algorithms to assess performance. We follow with a brief history of evaluation in the field.

The importance of quantitative evaluation was first addressed in Liu et al. [11]. The authors argued it necessary for (a) better measurement of progress, (b) comparison of many algorithms in a single plot and (c) the use of machine learning techniques for parameter tuning. A graph sampling based similarity measure was introduced with which the authors evaluated a novel map construction algorithm.

Biagioni and Eriksson [1] then provided the first survey in which eleven algorithms were discussed. They performed a quantitive evaluation on three of them using a modified version of the graph sampling similarity measure. This version is to our knowledge

the one predominantly used as quantitative evaluation measure in current road map construction literature [12, 8, 9, 7].

Ahmed et al. [13] complemented and expanded the work by evaluating more algorithms. The evaluation was performed using four instead of one data-set and made use of additional similarity measures.

More recently, Duran et al. [14] focused on qualitative evaluation of five algorithms using four hiking data-sets instead of urban data-sets. They also introduced similarity measures capable of quantifying the extent of typical local errors.

Hashemi [15] provided a testbed for evaluation. A qualitative evaluation of a number of measures in the literature was also provided. The testbed provided a collection of three new measures to quantitatively evaluate different errors associated with map construction.

And finally, Chao et al. [5] performed another survey and comparison of algorithms. Additionally, a quantitative analysis of evaluation measures was performed.

Our research builds on the results and efforts by this literature, but differentiates itself clearly. Our main aim is to determine how to accurately evaluate road map construction algorithms. We believe that current methods of evaluation are insufficient in two parts. The first problem being that the same small amount of limited trajectory sets are being used. These sets only exhibit very specific GPS interference errors. As a result, performance of algorithms is only compared on those unique instances, telling us little about their general performance. The second problem lies in the similarity measures being used to compare the constructed map to ground truth. Although multiple evaluation measures have been presented, the graph sampling based similarity measure introduced by Biagioni and Eriksson [1] is by far most used in the literature. However, as far as we know, only limited attempts have been made to analyse its effectiveness and use as a global road map similarity measure. Based on our aim and these two problems, the contributions of this thesis are the following:

- Provide a thorough comparison of trajectory based road map construction algorithms.
- Use synthetic trajectories to evaluate performance of the algorithms under a variety of GPS interference circumstances.
- Provide a publicly available evaluation framework in c# to facilitate more robust evaluation of road map construction algorithms in the field.
- Provide an in depth analysis of the effectiveness and limitations of the graph sampling similarity measure in global road map comparison.

In the next section we will look at background based on previous work in the field. In the methods we will discuss our framework, including the trajectory generation, and look at the setup for the experiments concerning the analysis of the graph sampling similarity measure. The results will highlight the different behaviour of our algorithms on both public data-set test cases and synthetic test cases. We will also look at findings

of the similarity measure analysis experiments, and see how the measure works on local and global level. In the conclusion we will share the key takeaways of our findings in relation to our aim and contributions. Finally, in appendix A and B we include tables and figures containing additional results of test cases run on the algorithms.

2 Background

We will start by looking at trajectory based road map construction in general. An overview will be given of algorithm methodologies and where the specific algorithms that will be used in our evaluation place themselves. Then a switch will be made to road map evaluation, where we will give an overview of some evaluation measures. The graph sampling measure we use in our evaluation will be discussed in more detail.

2.1 Road map construction from GPS trajectories.

In trajectory based road map construction we are given a set of trajectories T. Each trajectory is a sequence of measurements taken from an entity moving on a geometric ground truth graph $G = (V, E)$. V being the set of vertices and E the set of edges. Different versions exist for which GT is directed/undirected and planar/non-planar. A taken measurement in a trajectory consists of at least a timestamp and a position, but can also contain additional data like heading direction and speed. Position is given in either a projected coordinate system, like Universal Transverse Mercator (UTM), or in a spherical coordinate system, as longitude and latitude. For undirected road graphs we define intersections I as the subset of vertices with a degree different from two. For directed road graphs we consider vertices of degree two, with either two outgoing or two incoming edges, also as intersections. Finally we consider roads R as the set of polylines, consisting of a pair of intersections that are start and end, connected by a sequence of non-intersection vertices. Figure 1 contains a simple overview of the structures.

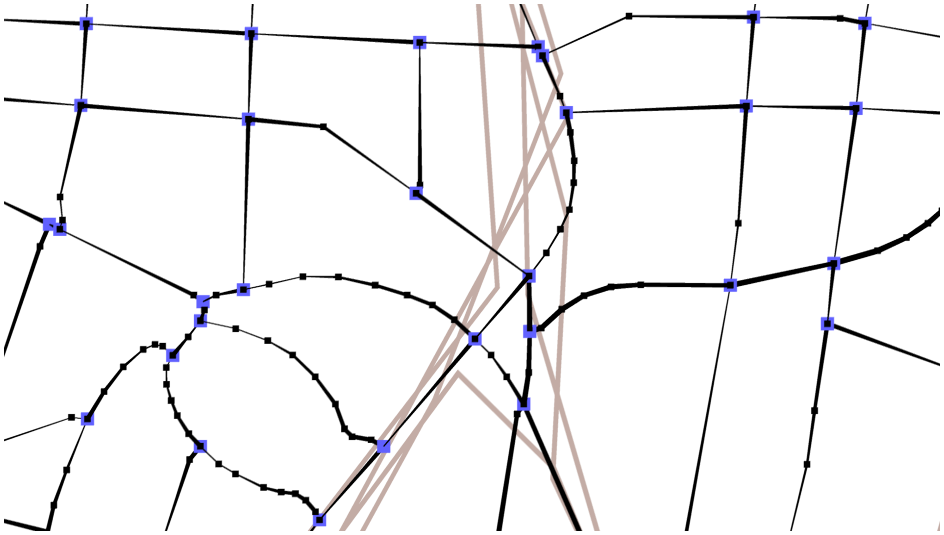

Figure 1: Snippet of map data. Vertices that are intersections colored blue. Roads shown as polylines between intersections, where directedness is visualized by increasing thickness. The thin side being start and the thick side end. Undirected edges have uniform thickness. Polylines of trajectories shown in grey.

Using only the given trajectories T, an underlying constructed road graph $CM = (V^*, E^*)$ has to be inferred. This map CM should represent the ground truth road map GT as close as possible.

An important thing to note is that there can not be any assumptions made on the coverage of the trajectories. T might consist of trajectories that never traverse certain roads in our ground truth map GT, making it impossible to infer those parts of the graph.

2.1.1 Map construction algorithm categories

Within trajectory based map construction, many different algorithms have been devised in recent years [1, 8, 16, 17, 9]. Survey and evaluation literature has often separated them into categories that follow the same algorithmic methodology. These categorisations are slightly different across sources, but for the most part share the same high-level overview. We will handle the same categorisation given in [14], namely:

- Point clustering algorithms
- Density based algorithms
- Incremental track insertion algorithms
- Intersection linking algorithms

The biggest difference that can be observed in comparison with other categorisations, is that point clustering algorithms and density based algorithms each have their own category instead of having a merged one. We find that the algorithms in both categories are sufficiently distinct in their methodologies, and thus deserve their own category. In the next subsections we will give a high-level overview of each category. Visual examples will be provided along, inspired by [1]. These figures do not reflect actual performances of categories in any way. The trajectories were chosen in a way to additionally highlight some limitations.

2.1.2 Point clustering algorithms

In general, point clustering algorithms work by mapping the trajectory set T to a representative point set P, which is then connected to form the desired constructed graph CM. An example of a simple approach follows and is highlighted in Figure 2. Starting with raw trajectories, representative clustering points P are sampled evenly along the trajectory data as an initial guess. A clustering algorithm, like k-means, is then run to slowly converge the cluster points to positions that best describe the underlying trajectory data. Here the Euclidian distance for projected coördinates, or Vincenty distance for longitude and latitude coördinates is used. Once converged, the points of P are connected by looking at the trajectory data to form CM. Edelkamp and Schrödl [18] kick-started this approach, after which many point clustering based algorithms have been presented. A more recent and optimized algorithm is given by Kharita [9], which

will be used in our evaluation.

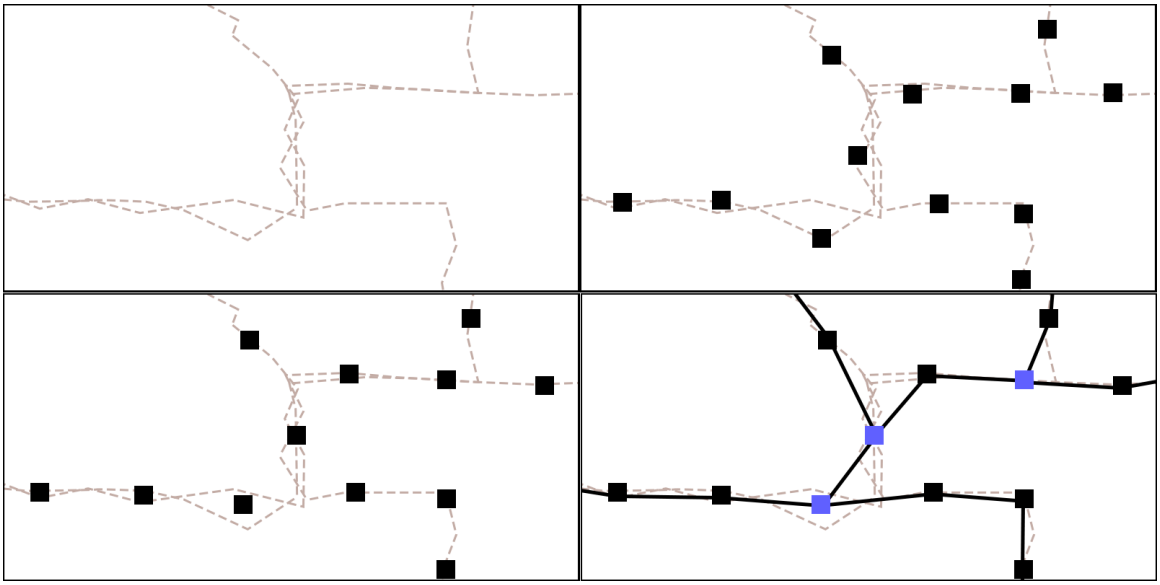

Figure 2: Sketch of basic point clustering strategy. Top-left trajectories. Top-right showing a possible initial representative point set. Clustering algorithm is run to fit point set, shown in bottom-left. Bottom-right showing final map with intersections after linking vertices.

Kharita differentiates itself from the sketch of point clustering algorithms we have given by incorporating angle information in the process. Each point is not only labeled with its position, but also with an angle. This angle corresponds to the clockwise orientation of the edge, of a trajectory, that contains the point. Instead of only using the Vincenty distance, a distance for the angles of two points is used as well. For this purpose the unit circle metric is used, defined as $d_{angle}(\alpha_1, \alpha_2) = min(|\alpha_1 - \alpha_2|, 360 - |\alpha_1 - \alpha_2|)$. By incorporating this information in to the clustering, points of different roads that are geometrically close to eachother can still have a big distance between them if their heading is very different. In this way they can converge to different cluster points, preventing them from merging to the same road in the constructed map. However it does not allow for close parallel roads with the same heading to be separable. For a more detailed overview of the exact algorithm we refer to the paper [9].

2.1.3 Density based algorithms

Since density based strategies share a lot of similarities with point clustering strategies, they are sometimes merged into one category [5]. However, both performance and lower procedures differ widely from strict point clustering algorithms. Density based algorithms approach inference by first determining a probability density function of the trajectories. Road graphs can then be extracted from this function based on the probabilities. A very popular density based approach is using kernel density estimation (KDE),

also highlighted in Figure 3. This works by first overlaying a 2-dimensional grid over the area spanning our trajectories. A histogram can then be produced by calculating how many trajectories roughly go through a cell of the grid. Here the resolution of the grid can play a big role in accuracy and performance. Once we have a weighted 2D mapping, we apply a threshold to the cells to prune away possible outliers. After pruning, the road outlines and center lines can be computed in various ways, for example using Voronoi diagrams. Once we have the raw road outlines, intersections can be extracted from it, producing the final graph CM. A popular KDE algorithm used in many cross comparisons in the literature is the algorithm by Biagioni and Eriksson [12].

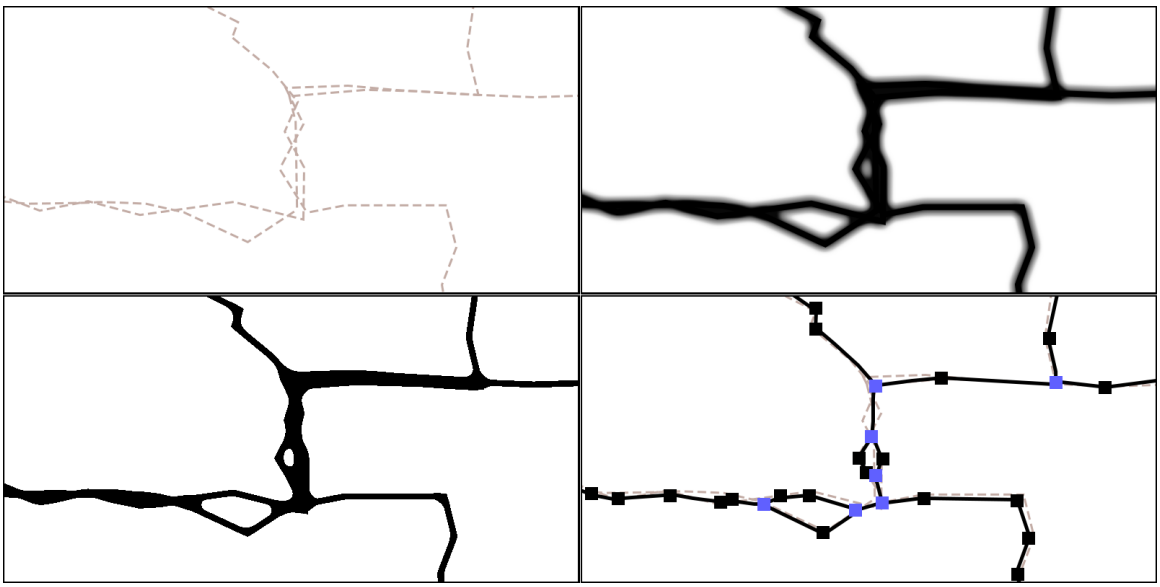

Figure 3: Sketch of basic density based strategy. Top-left trajectories. Top-right kernel density estimation based on trajectories. Threshold is applied to discretize estimation, shown in bottom-left. Bottom-right showing final map with center lines determined and intersection extracted.

2.1.4 Incremental track insertion algorithms

Incremental track insertion is a category of algorithms present in all reviewed surveys. The process is fundamentally different from the first two strategies that we discussed, in that it constructs the graph CM iteratively. Here we start with an empty weighted graph, and add a given trajectory from T in the first iteration. The main idea then is to iteratively expand the current graph on the basis of a not yet added trajectory. If the chosen trajectory does not overlap with the current map, we insert it fully into the map. If it does overlap fully or partly, the part that overlaps is determined. The weights of the edges in CM that overlap with the trajectory are increased by one. The part of the trajectory that does not overlap with any part of the graph is added as a new segment to it. Once iterated through all of the trajectories, we end up with a map of weighted

edges. Edges with low weight can be pruned away to remove possible outliers, after which the graph CM is returned. Figure 4 shows an overview of the procedure. Cao et al. [19] presented an initial incremental track insertion algorithm that has seen a lot of popularity throughout the years in evaluations and surveys. Ahmed et al. [6] presented a more advanced method of figuring out which part of trajectories overlap with the current map. A more recent algorithm is Roadrunner [8]. This algorithm does not strictly follow the incremental track insertion steps we defined, since it handles map expansion at a smaller scale. High-level the same strategy applies though to constructing the map. Another recent algorithm is by Buchin et al. [7]. This algorithm will be used in our evaluation.

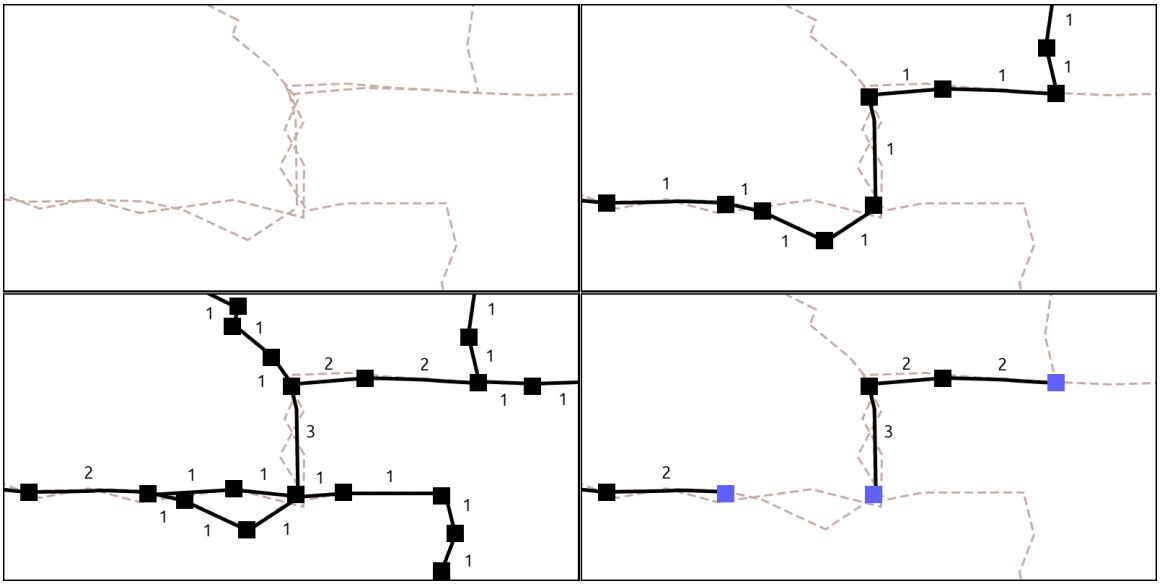

Figure 4: Sketch of incremental track insertion strategy. Top-left trajectories. Top-right trajectory added in first iteration. Bottom-left weighted road graph after all trajectories added. Although detrimental in this example, pruning can be used to get final road map in bottom-right.

The algorithm by Buchin et al. differs from our basic sketch of incremental track insertion algorithm in what trajectories it uses to expand the map. Instead of using the raw given trajectories, bundles of sub-trajectories are determined and used to build the map. These bundles are determined by looking at segments of different trajectories that overlap. Looking at bundles instead of raw trajectories provides a way in which outlying parts of trajectories can be disregarded. Additionally it allows road segments that are geometrically close to remain separated in different bundles if prior or following parts of the segment are substantially different. In this way close parallel roads with the same heading can for example still be captured. For a more detailed overview of the exact algorithm we refer to the paper [7].

2.1.5 Intersection linking algorithms

Lastly we will discuss the intersection linking category, which works in two steps. The first step has some overlap with the point clustering algorithms. Instead of trying to represent the whole of trajectories T in a representative point set, it focuses on exclusively identifying the intersection points of graph CM first. While there are different ways to do this, changes in directions and speed of trajectories are known to be used. Turn points in trajectories with such abrupt changes can be determined. If multiple different turn points are close to eachother, this indicates the presence of an intersection. We thus find clusters of turnpoints based on proximity and use their centroid as intersection. In the second step the found intersection points are connected by looking at trajectories that go along identified intersections. Figure 5 shows an overview of all the steps. Karagiorgou et al. [20] presented the specific algorithm described, with recently a more advanced version [10].

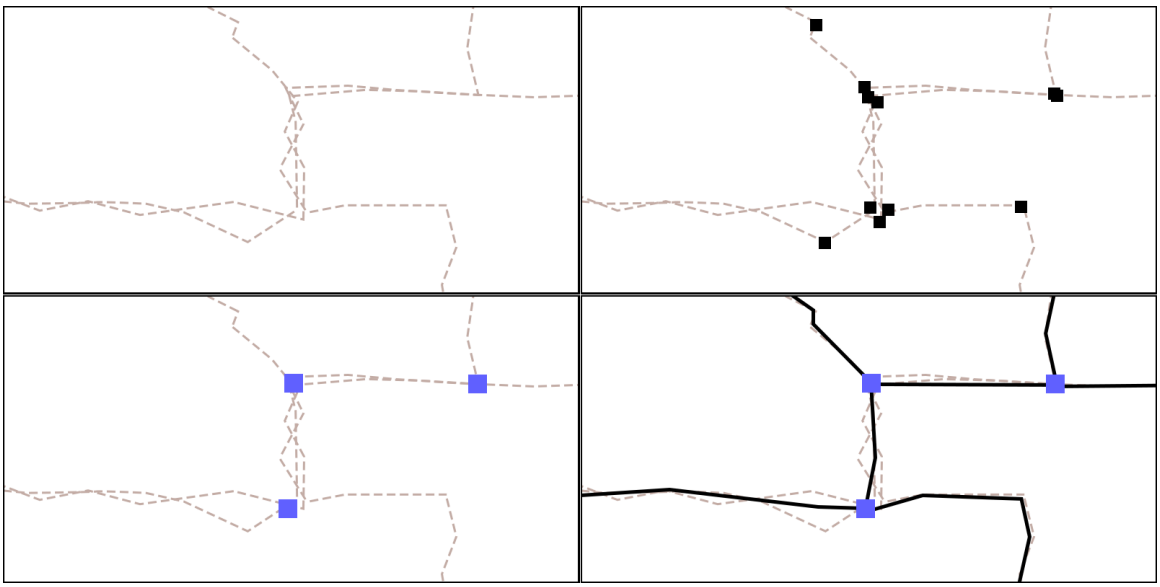

Figure 5: Sketch of intersection linking strategy. Top-left trajectories. Top-right extracted turn points from trajectories. Bottom-left identified cluster centroids as intersections. Bottom-right shows final map by linking intersections using the trajectories.

2.2 Trajectories and map data

As was stated in the introduction, part of the developments in the map construction field are the result of the massive increase in the amount of mobile devices that can track trajectories. Predominantly those working on the GPS network. Other networks like DGPS, WiFi and cellular are also capable of capturing trajectories. However the popularity of the GPS network for navigation purposes has made it the assumed network

on which trajectories are collected. This has consequences for map construction, since we are in a way constrained by the errors of a network that has been specifically designed for navigation purposes. We see this limitation in two types of errors that can be assigned to positioning networks. A precision error and a sampling error. We discuss both of them below.

2.2.1 Precision error

The precision error concerns itself with the erroneous distance between the position of a measure point and the actual position of the moving entity at time of measuring. When the precision error is high, trajectories have more noise, which makes the path harder to follow. Precision error especially becomes problematic if the distance of the error becomes larger than the distance that separates roads in the ground truth. In that case a measurement point of a trajectory might be closer to an untraveled road, than the actual road on which that measurement point was taken. GPS is known to have a precision error resembling a normal distribution with a mean of about 5-20m [21, 22, 5] depending on device used and environment. This is well above the distance between a lot of parallel roads, which prove to be difficult cases for many map construction algorithms [14]. DGPS for example only has an average precision error of smaller than a meter [23].

2.2.2 Sampling error

The sampling error concerns itself with the time between two subsequent measurement points in a trajectory. When the sampling rate is high, trajectories have a higher resolution, making the path followed much more clear. Especially parts of a path that have abrupt changes in direction, like corners or very unstraight roads, suffer harshly from low sampling rates. In those cases the momentary change might not be captured well and has to be inferred from the previous and next measurement point. GPS is known to have a wide sampling rate of around 1 to 300 seconds [24], depending again on device and environment when measuring. Additionally, trajectories might get simplified for storage purposes, resulting in a differing sampling rate than was captured. A sampling rate under 30 seconds is generally considered as a high sampling rate.

Finally, it should be emphasized that the existence and scale of these two errors in the data is what makes the trajectory based map construction problem challenging. The absence of the errors or the usage of a much more accurate network would make the map construction problem much easier to solve. Where trajectories would directly correspond to sequences of edges in our ground truth map, making road geometry inference almost trivial. This highlights the importance of an evaluation setup where the extent of the two errors are heavily taken into consideration. Another important factor is the heterogeneity of the errors. GPS error experiences variability as a result of satellite geometry, signal blockage, atmospheric conditions, and receiver design quality [21]. Quality is often reasonable for map construction purposes when the trajectories are taken during

good weather conditions on open roads where interference is minimal. But starts to deteriorate when those conditions are not met, highlighted in Figure 6.

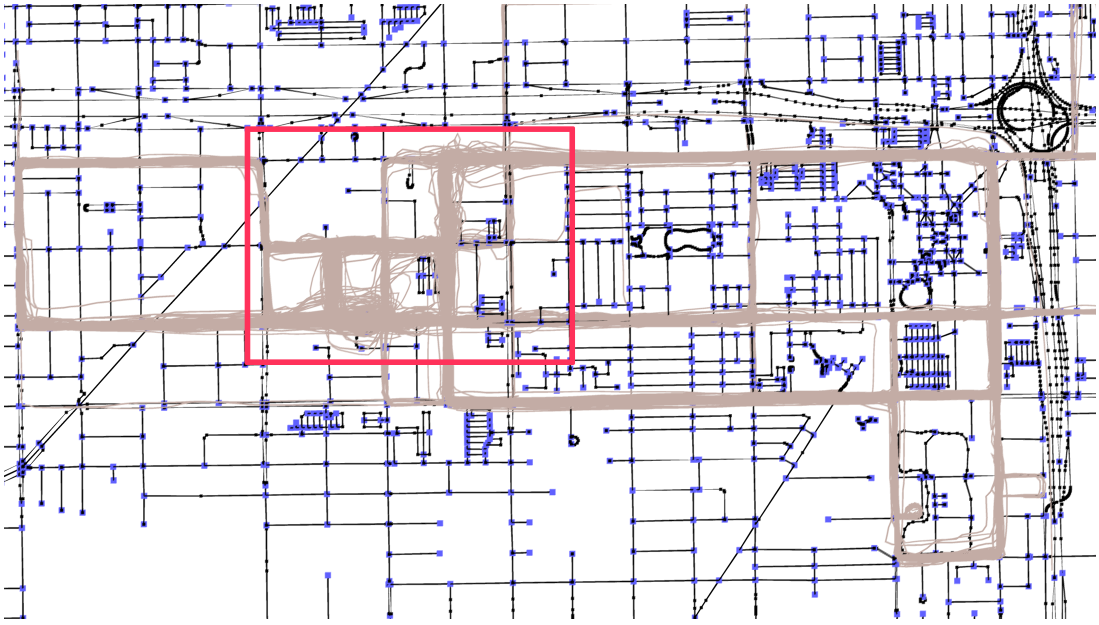

Figure 6: An overview of the Chicago data-set. Trajectories in grey, laid on top of the ground truth map. In red we see a district with high GPS interference environment, because of its many high-rise buildings.

2.2.3 Data-sets

We end this subsection with a quick description of the data currently being used in the map construction literature. Although GPS trajectories are individually gathered on sites like https://wikiloc.com/, known bigger test cases of areas are preferred in the literature. A couple of test cases are especially popular in the field, most of them publicly available on http://mapconstruction.org/. The Chicago shuttle data set introduced by Biagioni and Eriksson [12] is an example. These are mostly urban data-sets containing ground truth and trajectories, which have been captured by vehicles for map-construction-like purposes. Table 1 shows an overview of properties of some well-known sets. Something to note here is that often these trajectory sets have been captured using the same vehicles and capturing devices. Making the data more homogeneous than natural user collected data. An interesting departure from typical maps was given by Duran et al. [14], where instead of evaluation on urban maps, rural ones were chosen with data gathered from volunteered geographic information.

Name	$\|V\|$	$\|E\|$	$\|I\|$	$\|R\|$	$\|T\|$	Avg. sampling rate
Athens-small	2694	3436	1253	1998	128	fixed 30s
Athens-large	32212	39699	13479	20969	119 (+ 128)	fixed 30s
Chicago	9429	11801	4257	6668	888	3.62s up to 29s
Berlin	5894	6839	1784	2729	27189	41.66s up to 120s

Table 1: An overview of several popular publicly available urban data-sets. Showing numbers of vertices, edges, intersections, roads and trajectories. An indication of the sampling rates of trajectories within a set is also given.

A final remark is that for all of these data-sets, except the Berlin set, a large amount of roads are not covered by any trajectories. An example of this is visible for Chicago in figure 6. We will see later on that this has consequences for evaluation.

2.3 Road graph evaluation

Ideally the performance of an algorithm would be assured by its design, however in the map construction problem the specification and input of the problem has not allowed for it. Except for specific cases where additional properties can be assumed [25]. Thus we are left in assessing performance by looking at the resulting maps that algorithms output. Comparing them to some ground truth.

2.3.1 Graph matching

The comparison of two road graphs has much similarities with the graph matching problem. In the most typical case of graph matching, called graph isomorphism, we are trying to find some one-to-one correspondence between the nodes of two graphs that is edge-preserving. Here edge-preserving means that if an edge is present between two vertices of the first graph, an edge has to be present between the corresponding matched vertices of the second graph, and vice-versa. Graph matching, however, has grown to be a field encompassing a much bigger set of problems. Conte et al. [26] provides an excellent survey with detailed taxonomy of the field. For road network comparison we are mostly interested in the inexact version of graph matching. So that an optimal matching can be determined with nodes that do not have to strictly correspond to eachother. Some measure of similarity is then returned indicating the confidence of the matching. This relaxation allows nodes that are mostly similar but have some inference error, like a single missing edge, to still be able to match. Additionally, since we are dealing with maps instead of graphs, we want to take the geometric embedding in to account when matching. A known method that takes this in to consideration is the graph edit distance between two graphs. Which is defined as finding an optimal sequence of weighted operations that can transform one graph into the other. No polynomial algorithm has been found for calculating the graph edit distance. Several versions of the problem have been proven to be NP-complete, with 3D-Matching and Hamiltonian

path problems being reducible to them [27]. This dead end in inexact geometric graph matching, has as consequence that we are left to heuristic measures to calculate the similarity between our road graphs.

2.3.2 Road graph sampling similarity measure

The literature has presented multiple global road map similarity measures. We refer to Ahmed et al. [13] and Hashemi [15] for an overview of proposed measures. Although the usage of several measures would be preferred, we have seen in algorithm literature that evaluation in practice is performed using only one or two measures. The measure that is mostly being used and which we will be focusing on is the TOPO version of the graph sampling similarity measure by Biagioni and Eriksson [1]. A detailed overview of the measure follows.

We take x_G as a random point on a road in G, for which x_C is the closest point on a road in CM to x_G. We require that the Euclidian distance $dist(x_G, x_C) < k$, otherwise we resample the points. This additional distance requirement has been used in various papers to ensure that we are only evaluating parts of GT that are within reach of CM. This is to our knowledge done when GT is substantially larger than CM. Presumably since GT contains a significant amount of roads that are not covered by the trajectory set T, and which have not been pruned away. We will see later on that such an unpruned ground truth has consequences for the outcome of the measure.

For both points x_G and x_C we now determine the respective point sets N_G and N_C. These two point sets represent local neighbourhoods of the two points. To obtain N_G, we explore GT starting from x_G in a depth first manner. Exploration is fully constrained by the topology of the neighbourhood, including directed/undirected nature of edges. Additionally, traversing the same edge multiple times is not allowed. Along the exploration we sample points. A point is sampled if we have explored a distance of l meters from the previous sampled point. Here we consider x_G to be our initial sample point. We keep exploring until a max radius/distance of r is reached to x_G. The exact same method is used to obtain N_C by exploring starting from x_C.

Now we compare the two local neighbourhood representative point sets by matching the points in N_C to N_G. We define a_i as point $i = 0..|N_C|$ in N_C. We define b_j as point $j = 0..|N_G|$ in N_G. We determine a maximum bipartite matching between the point sets where two points can only be matched if $dist(a_i, b_j) < d$.

Finally we obtain the amount of matched points z and calculate $precision = \frac{z}{|N_C|}$ and $recall = \frac{z}{|N_G|}$. With which we can determine a f1-score for the specific neighbourhood. The procedure of evaluating a local neighbourhood is highlighted in Figure 7. To evaluate the whole graph then, the process is repeated many times for other random neighbourhoods. An average is taken over all the precision, recall and f1-scores of each neighbourhood which can function as our desired global similarity measure.

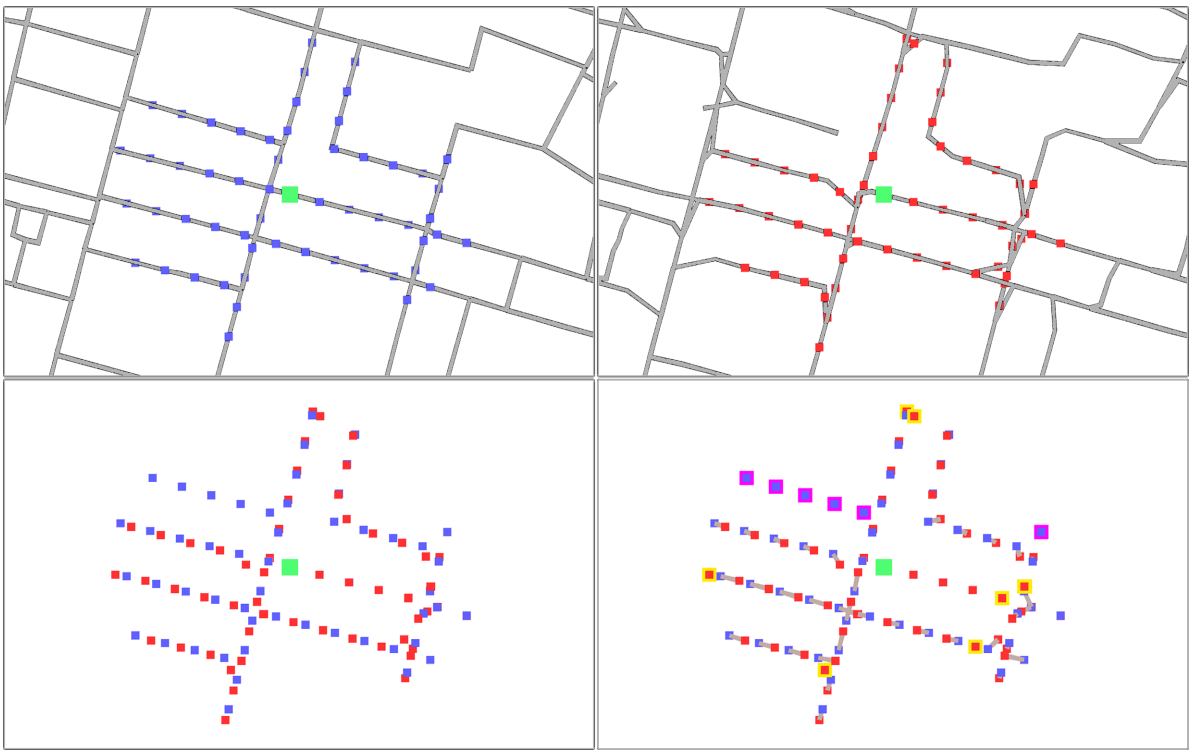

Figure 7: Overview of evaluating an example neighbourhood. Top-left showing GT, X_G in green and N_G in blue. Top-right showing CM, X_C in green and N_C in red. Bottom-left showing N_G and N_C together. Bottom-right showing result after matching. Matches shown in grey, unmatched points in N_G indicated with magenta and unmatched points in N_C with yellow.

2.3.3 Analysis of evaluation measures

Analysis of measures has been previously performed in literature, looking to what extent important errors are captured. We will focus only on the shortcomings identified for the graph sampling measure. The following four types of errors have often been discussed:

- Geometric error
- Topological error
- Spurious roads
- Road losses

Hashemi [15] gave an insightful qualitative evaluation of many quantitative measure used. Each measure was highlighted for its general shortcomings and limitations in regards to capturing the errors above. For the graph sampling evaluation measure three main shortcomings were identified. The first one being that it has difficulties in reflecting the completeness of the graph. The second one being a combination of problems arising from the lack of description in how its parameters should be tuned. And the

final one being related to estimating the global similarity by averaging random local neighbourhood similarities.

Chao et al. [5] performed a quantitative analysis on some evaluation measures. The main idea behind this evaluation was to synthetically create maps that have in varying degrees only one isolated type of error. This was done by transforming the ground truth at local points. In the case of analyzing the road loss error for example, some specified amount of random roads in the ground truth were removed. Not only the four errors mentioned above were analysed, but also some specific road misconfigurations. The synthetically erroneous map is then compared to the ground truth map using the evaluation measure that needs to be assessed. By increasing the amount of errors that are present and observing how well the evaluation measure reflects this increase, its extent to capture that type of error can be investigated. In general the results showed that each measure still had shortcomings in identifying some types of errors. Graph sampling performed well in comparison to the other analyzed measures. But still showed problems in capturing road shape errors, i.e. small geometric errors, and to some extent spurious roads. Additionally none of the measures took the importance of certain roads in to consideration. A final remark, is that this analysis was based on a ground truth map for which all of its road were able to be present in the constructed map. As far as we have seen, this does not reflect how graph sampling evaluation is usually performed in the literature. Where it is the case that GT has not been pruned for roads that no trajectories cover.

3 Methodology

Our methods will be split in two distinct parts according to the goals mentioned in the introduction. In the first part we will be focusing on the first three goals and discuss the complete framework used for the evaluation of our algorithms. In the second part we will focus on the final goal and look at experiments for the analysis of the graph sampling evaluation measure.

3.1 Map construction algorithm evaluation

In this subsection we will present our improved setup for map construction algorithm evaluation. Implementation is available on https://github.com/AriSaadon/RoadMapConstructionEvaluation. We seperate this discussion in to four parts:

1. The data used for evaluation and the preprocessing involved on it. We make a distinction between two types of data pipelines. A standard one involving direct publicly available data-sets and a new one, based on map extraction from Open Street Map (OSM), made possible with synthetic trajectory generation.
2. The synthetic trajectory generation, which will allow us to investigate the performance of algorithms under a variety of GPS interference circumstances.
3. The map construction algorithms and our setup for them in our performance comparison.
4. The setup of the road graph sampling evaluation measure we use to quantify performance of the algorithms.

Figure 8 shows how these parts relate to the pipeline of our framework. We will next discuss each of them in more detail.

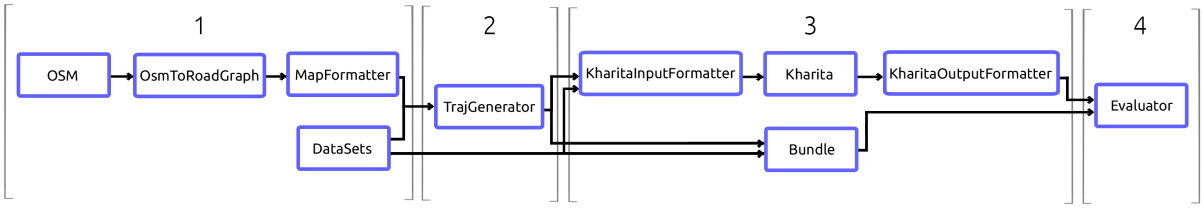

Figure 8: Simplified overview of evaluation framework pipeline.

3.1.1 Data and preprocessing

Both map data and trajectory data are needed for our evaluation. Map data functions as ground truth and allows us to synthetically generate trajectories. Trajectory data functions as input for the algorithms. Although we will be mainly synthetically generating

trajectories, we will also use the trajectory data included with data-sets for contextual result to compare the synthetic trajectory results to.

Three maps are being used in our evaluation. A subsection of the known Athens-small and Chicago maps, and a section of Utrecht. Figure 9 shows an overview of all of them. We will first discuss the usage of the Athens-small and Chicago data-sets, after which we will look at the additionally used Utrecht test cases.

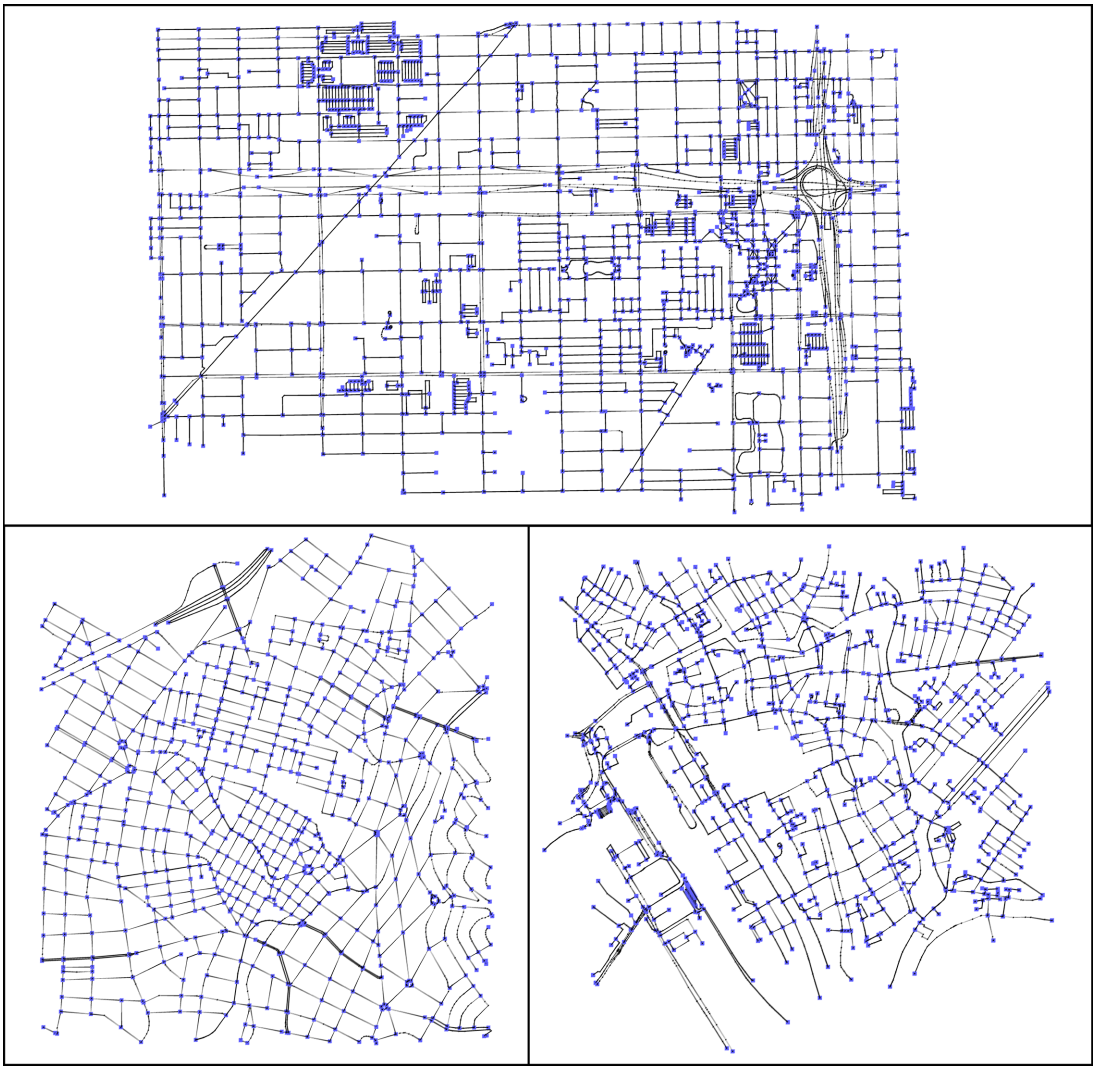

Figure 9: Overview of the three used maps. Chicago on top, Athens bottom-left and Utrecht bottom-right.

Public datasets - For Chicago and Athens-small, data-sets were used available on http://mapconstruction.org/. We used full ground truth maps for test cases involving trajectory data from these data-sets. However, for synthetic trajectory test cases, smaller sections of ground truth map were used to keep our experiments more perfor-

mant. The smaller section of Athens-small we used, is defined by reducing the original tight bounding box by $(left\_padding, top\_padding, right\_padding, bottom\_padding) = (300, 600, 0, 1500)$. For Chicago we used a reduction of $(2000, 1000, 1100, 1000)$, distances are given in meters.

For the publicly available data-sets, data comes in the following format: A *Vertices* file containing on each line

$$vertex\_id, projected\_x, projected\_y$$

An *Edges* file containing on each line

$$edge\_id, vertex\_id\_1, vertex\_id\_2, directedness$$

And finally many $Trip\_x$ $(x = 0, .., |T| - 1)$ files, containing on each line

$$projected\_x \; projected\_y \; time$$

The id's being integers, $projected\_x$ and $projected\_y$ being coordinates of a projected coordinate system, $directedness$ being a boolean indicating whether the edge is directed or undirected and $time$ being an integer indicating seconds. Chicago uses UTM zone 16N and Athens-small uses the GGRS87 system. The Chicago set contains 888 trajectories and Athens-small 128. However, for Chicago only the first 100 trajectories were used, since the trajectory bundling algorithm implementation ran out of memory using more. Kharita could use the full set, but for the sake of a fair comparison we used the same number of trajectories.

OSM data - Since we can generate trajectories, we are not bound by data-sets specifically available for map construction. We only need road graph data for the synthetic trajectory pipeline. This data is available and can be extracted from OpenStreetMap (OSM). Allowing us to evaluate map construction for every charted area of the world available in OSM. For Utrecht we chose a suitable section with diverse geometry and topology. The OSM export functionality was used on the area defined by the coordinates $(5.1044, 52.0994, 5.1350, 52.0836)$.

OSM exports a custom XML file containing all the information available in the defined area. This includes data we are not interested in like street names and attractions. Additionally, not only the road graph is exported, but also sidewalks and bicycle roads. Although these are also interesting test cases, we are solely interested in vehicle roads. Extracting this road graph from the XML file is not trivial. However, the excellent public *OsmToRoadGraph* library alleviates most of this task. Using it we convert the original OSM XML file to an intermediate custom road graph format. A parser was then written to convert the custom format to the earlier described *Vertex* and *Edge* files. Here we use UTM zone 31N as projected coördinate system.

Additional preprocessing - Lastly, due to the maps containing both directed and undirected edges, we had to perform a transformation in some cases. The ground truth

maps will be compared against either a fully directed or undirected map outputted by the algorithms. The trajectory bundling algorithm outputs an undirected map, and Kharita a directed map. For this reason the hybrid ground truth maps were made undirected by changing the directed roads to undirected ones, and directed by taking an undirected road and splitting it into two directed roads that go in reverse directions. This transformation is performed before synthetically generating trajectories on the map.

3.1.2 Trajectory generation

Our approach generates trajectories on a given graph to investigate the performance of algorithms under a variety of circumstances. We especially want control over the extent of precision and sampling errors of our trajectories. The generated trajectories should look visually similar to the trajectories present in the public data-sets. And preferably we want to simulate the normal distributed nature of the precision error.

We approach generating trajectories in the following way. First two random points in the road graph are sampled. This can be done in multiple ways. We chose that every point encompassing the graph should have an equal probability of being sampled. In this way the probability of having a point generated on a certain road is given as a function of its length. By using such a weighted probability for every edge, we can pick one at random and generate a point uniform randomly on it.

After sampling two points, we assign one as the start point and the other as end point. We then calculate a shortest path from start to end. The A^* algorithm was used. If a shortest path could not be found, we resampled the points. This can be the case when two points are sampled for which only paths exist that also contain roads outside of our map.

We take a precision error parameter π and a sampling error parameter σ. Given a found path, we traverse it and sample points after every σ meters. Corresponding to an entity moving at constant speed along the trajectory, for which measurements are taken at a constant rate. Although sampling rates and movement speed are not constant in practice, fixing both variables will allow us to evaluate for different levels of error without having to take heterogeneity of both variables into consideration. After traversing the whole path we will have a sequence of points.

We then apply a precision error to the sequence S. Initially we used a multivariate normal distribution to generate an error with mean $\sqrt{\pi}$. This resulted in unnatural jagged trajectories for higher values of π combined with a low value for σ, also visible in Figure 10. To counter this, smoothing was applied by averaging the error of a measurement with the errors of the q previous measurements. A Central limit theorem (CLT) exists for lineair combination of independent identically distributed variables. But by smoothing, subsequent measures are no longer independently distributed, since they average for a part the same errors.

However, if we have X_i being the i^{th} averaged measure, X_{i+q} is the first subsequent measure that averages a set of non-overlapping errors, making it independent from X_i.

What follows, is that S is m-dependent, since we have that for $j > m \geq 0$ the two sets

$$(X_1, ..., X_i) \quad and \quad (X_{i+j}, X_{i+j+1}, ...)$$

are independent. In our case $m = q - 1$. CLT is proven to exist for m-dependent sequences, a proof is given in section 2.8 of Lehmann [28]. Thus our averaged measurements remain normally distributed.

By visual comparison with the Chicago trajectory set, we found $q = 20$ to give realistic looking trajectories for all used values of π and σ. In practice we used all combinations of $\pi = \{0, 50, 100\}$ and $\sigma = \{50, 100, 150\}$. Trajectory sets of size $n = \{100, 200\}$ were generated for each combination to also investigate performance of algorithms under different input size. Directed and undirected versions of each trajectory set were produced. This eventually leaves us with thirty-six distinct trajectory sets for each map.

Figure 10: Overview of trajectory generation. Top-left showing start point in red and end point in green. Top-right shortest path between points. Bottom-left sampled measures with independent precision error. Bottom-right smoothed errors.

A final important step that we perform after generating a trajectory set, is pruning the ground-truth graph. While generating trajectories, we can keep track of which edges are used. Edges that eventually are not covered by any of the generated trajectories are deleted. On average, pruned maps kept 70.28% of the original ground truth road length with the amount of trajectories we used. Evaluation using these pruned ground-truth

graph is more fair, since it is impossible for algorithms to infer a map with edges not present in the trajectory input. We will see the consequences of pruning later on in our results.

We can only prune accurately for synthetic trajectory sets. For public data-sets we do not prune the graph, since it is not provided what edges a trajectory actually has used. Although not often done in the literature, inferring these edges using a map-matching algorithm is a possibility, but inevitably introduces bias. Especially for trajectory sets that have large interference errors.

3.1.3 Algorithm setup

Algorithms take trajectories as input and return a road graph as output. Initially a choice was made to evaluate the performance of three algorithms. Kharita by Stanojevic et al. [9], Roadrunners by He et al. [8] and the trajectory bundling algorithm by Buchin et al. [7]. These were mainly chosen because they are state-of-the-art in the literature.

A basic setup was used to test the three algorithms before our actual evaluation. This setup consisted of providing the algorithm with the Athens-small and Chicago trajectory sets. Since these data-set have been used in the literature many times, results can be compared to validate an algorithm working correctly. While Kharita and the trajectory bundling algorithm showed expected results, Roadrunners produced an extremely small graph with often only one or two roads. Different parameters were tried, but results remained disappointing.

We suspect the reason for this behaviour being the sparsity of the trajectory set in combination with a high dependence on its parameter input. Roadrunners mostly falls in the incremental track insertion category. However, it iteratively expands roads locally at the borders of its current road graph. If in its parameters the initial points given to the algorithm lie in an area with low trajectory coverage, the algorithm might immediately stop expanding the map. For Chicago this seemed the case. After specific parameters were kindly provided by Buchin et al. [17], more complete results could be produced. But for Athens-small no parameters could be found, because in all likelihood most of its areas are too sparse for Roadrunners.

Although in theory this does not provide a problem, since we control the amount of synthetically generated trajectories. In practice, depending on the map, a lot of trajectories would have to be generated for Roadrunners to output a sensible road graph. This would not be a problem were the running time of the public implementation manageable. However, on our system (i7-4771, 8GB work memory, GTX 1660 super) running times for sets with a hundred trajectories already came close to multiple hours. Where we estimated for our evaluation this would result in several days per experiment. Excluding the additional time the increase in trajectories would have on the running time of the other two algorithms. A data preprocessing step for the purpose of creating an index for the trajectories, was the main bottleneck. We suspect this has something to do with the implementation, since very low CPU usage was observed. Because of this, after careful

consideration, we refrained from incorporating Roadrunners in our evaluation. Thus we will look at the steps involved in the evaluation of only Kharita and the trajectory bundling algorithm.

Bundling algorithm - The trajectory bundling algorithm is made with the discussed trajectory and map formats in mind. No processing of the input and output was needed. The algorithms also uses no parameters that can change the output (it does have some calculation performance parameters). Making it very easy to use.

Kharita - Kharita requires additional information and handles its own input and output format. Speed and angle must be provided at each measurement and coordinates have to be given in latitude and longitude. The *Pyproj* library was used for translating coordinates. When we were using the provided trajectories instead of generating ones, speed and angle were estimated using subsequent measurement points. Input was transformed to the correct format, and after running the algorithm output was transformed to an *Edges* and *Vertices* file. In many cases the directed map that Kharita outputs is made undirected for comparison against the trajectory bundling algorithm. This will be mentioned along the individual results.

Finally, Kharita takes two main parameters alongside the trajectory input. A *seed_radius* effecting the clustering distance. And an *heading_angle_threshold* effecting the extent angle differences are weighted during clustering. We used a separately generated trajectory set of Chicago with moderate precision and sampling errors ($\pi = 50$ and $\sigma = 100$) as tuning set. On it we tuned the parameters according to Stanojevic et al. [9]. An $heading\_angle\_threshold = 160$ twice the size of the $seed\_radius = 80$ was determined. These parameters were then used for all the other trajectory sets.

3.1.4 Algorithm evaluation

Lastly we need to evaluate the results produced by the algorithms. For this purpose we compare the outputted constructed map to the ground-truth graph. After all the trajectory sets are run on each algorithm. An implementation of the evaluation measure discussed in section 2.3.2 is used. Here neighbourhoods origins are generated randomly with the same weighted road length probabilities we used for the trajectory generation. For synthetic trajectories, where the ground-truth map is pruned, we always use the random neighbourhood origins as they are sampled (i.e. $k = \infty$). For public data-set trajectories, where no pruning is performed, the ground-truth origin and the constructed map origin are resampled if they are more than $k = 100$ away from eachother. Neighbourhood points are sampled using a depth first search. And a matching is determined using the Edmond-Karp algorithm. For the parameters we use a neighbourhood radius of $r = 500$, a sample distance of $l = 30$ and a maximum matching threshold of $d = 20$. The parameters were decided based on visual inspection of the sampling neighbourhoods and on what we have seen in the literature.

We average the errors of two hundred local neighbourhoods for the synthetic trajectory test cases and a thousand for the public data-set cases. We see in practice that pre-

cision, recall and f-1 values converge on those numbers for the maps. We prefer using precision and recall however for most of our results, as they give a better overview of inference shortcomings. Additionally we calculate the total sum of all the road length of both graphs and calculate the ratio of length between CM and GT, as $\frac{|CM|}{|GT|}$. Preferably we would like to see a value of one for this ratio. Lower scores indicating the presence of missing roads, and higher scores indicating spurious roads. We also visualize each ground truth map along with the constructed graph and the generated trajectories for qualitative comparison. These additional outputs will help us interpret the main numeric results.

This marks the end of our discussion of the algorithm evaluation framework used. Next we will discuss the second part of our methods concerning the experiments done to analyze the graph sampling evaluation measure being used.

3.2 Evaluation measure analysis setup

The graph sampling evaluation measure introduced by Biagioni and Eriksson [1] is frequently used in road map construction literature. However, little is known about its effectiveness in measuring similarity of two road graphs. For this reason we performed various analyses on its performance. With the aim of providing a better overview of its strengths and shortcomings.

Our analysis of the measure is separated in two parts, aligning intuitively with the two phases of its workings:

- In the first part we focus solely on the local neighbourhood evaluation. We do so in a qualitative manner. This will give us insight in how errors can locally be captured by the measure.
- In the second part we focus solely on the aggregation of multiple neighbourhood evaluations. This will give us insight in how the locally captured errors are combined to evaluate the road graph on a global level.

3.2.1 Local neighbourhood matching analysis

A prime reason for using a quantitative evaluation in the first place, is that it can give us a more accurate measure of performance. A good measure has to be able to capture and reflect many types of errors present in the map. This includes errors that can not easily be found or accounted for in a high-level visual comparison of road graphs.

To investigate the reflection of errors in the graph sampling measure, we perform a qualitative evaluation of local neighbourhood matchings. Here we are interested in what types of errors contribute in what way to the precision and recall returned by the matching. For a given precision and recall of a neighbourhood, we can identify errors that have effected the score, and that fail to be captured. By doing this for

many neighbourhood we can map out strengths and shortcomings of the graph sampling evaluation measure in relation to different errors.

Additionally from an algorithm point of view, a neighbourhood analysis can be particularly interesting and beneficial when unexpected scores are given back that do not align with what is expected from visual comparison between GT and CM. For example, a low recall is expected when large parts of the graph are shown missing, but unexpected when most roads seem to be inferred correctly. For unexpected results, inspection of neighbourhoods can shed light on output and teach us less obvious shortcomings of an algorithm.

We stick throughout the analysis to the same parameters for the graph sampling measure that are used in our algorithm evaluation, with the exception of the neighbourhood radius r. Instead of $r = 500$ we use $r = 150$ to narrow reflection to single errors and make the visualisations more clear. We use the artifacts and errors defined in section 2.3.3 from both Chao et al. [5] and Duran et al. [14] as inspiration in identifying local errors, but do not constraint ourselves to their exact classifications. Since in practice we see for example a varying extent of geometric errors, that are reflected differently by the local neighbourhood similarity.

A total of 50 neighbourhoods were investigated. Figure 17 and 18 in section 4.2.1 of the results visualize the local errors we identified. Investigation was done by visualising unmatched GT and CM neighbourhood points. These points were determined by looking at unused edges in the residual graph used for calculating the maximum bipartite matching between the CM and GT neighbourhood. Unmatched GT neighbourhood points indicate the presence of an error being reflected in recall, and unmatched CM neighbourhood points in precision. These points, together with their respective neighbourhood origins, are then overlayed on visualisations of both GT and CM for evaluation.

3.2.2 Global neighbourhood aggregation analysis

For each point in GT a local neighbourhood can be sampled. Matching is used to measure a local similarity between the neighbourhood and its closest corresponding neighbourhood in CM. However, eventually we are interested in the global similarity between both road maps. For this purpose we aggregate the similarity measure of different local neighbourhoods. If no bias exists towards any points, we would ideally want to aggregate over the neighbourhoods of all points defined by GT. This aggregation would result in an exact global measure of similarity between CM and GT. But, regretfully no algorithm is known to exist that can calculate this measure. Leaving the literature to estimate an exact value by averaging a discrete amount of random neighbourhoods.

Two aspects have to be considered for this purpose. The first aspect relates to how the random neighbourhoods are exactly sampled. This can play a role in the relative importance of certain sections of our map in the global evaluation. For example, (a) in the case with no bias towards any points, an edge that is k times longer than another edge, will be k times as likely to have a point be sampled on it. This can result in an

heightened contribution from longer edges to our aggregate global measure. Another possibility (b) is to make the probability of a point to be sampled on an edge, uniform over all edges. Here we introduce positive bias towards points defined by relatively small edges of our map. Literature using the graph sampling measure makes no mention of this aspect. We assume a random sampling is used with no bias towards any points (a) and have thus used a same approach in our algorithm evaluation. For the purpose of investigating the influence of a different neighbourhood sampling approach on the global aggregate, we compare results of (a) to (b). Note that the use of a different sampling approach changes the exact value we are trying to estimate.

The second aspects then relates to how many neighbourhoods need to be aggregated to effectively estimate the exact value we desire. For this two experiments are performed.

In the first, we investigate the change of local similarity that occurs on a small scale. Neighbourhoods that are close to eachother are likely to have many similarities. But it is not clear whether this also translates to rather similar local evaluation scores. If these indeed differ only by a small degree when sample points are close to eachother, estimation of our global measure can possibly be approached in a more efficient way. Since if little variance in precision and recall occurs for a section of the map, we can benefit from only using a single weighted representative sample from the total set of points defined by it.

To look at this change we evaluate neighbourhoods for multiple origin points lying on a single road. We select origin points evenly spaced over the road in question. For investigating m neighbourhoods on road q, we begin from one of the endpoints and sample a point every $\frac{|q|}{m-1}$ meter. For each of these points we then determine its GT neighbourhood and the closest CM neighbourhood, after which we evaluate. A higher m beneficially provides more resolution in the change of the score along the road, but increases running time. To keep running time feasible, we chose $m = 500$ for our first experiment on a single road. And $m = 20$ for our second, involving multiple roads. The only test case used is Athens with 200 trajectories and $(\pi, \sigma) = (0, 50)$. This was done to keep the experiments in scope. The preference for smaller trajectory errors, was to keep variation in the type of inference errors. Since we see that for larger trajectory errors, specific inference errors start to dominate the outputted maps.

In the second experiment we will look at convergence of the global average measure as more neighbourhood are sampled. By comparing this convergence between different test cases, we can obtain insight to how many samples are sufficient under specific circumstances. This aspects has also never been addressed in the literature before, as far as we know.

The same test case of Athens is used for this analysis, unless mentioned otherwise in the results. Evaluation is the same as we used for our algorithm evaluation. Except that not only the final average over all 200 neighbourhoods is calculated, but also the 199 previous averages, each obtained by the addition of a newly evaluated local neighbourhood.

4 Results

Like our methods we separate the results in a section for the algorithm comparison and a section for the experiments of the graph sampling evaluation measure. We start with the algorithm evaluation.

4.1 Algorithm Evaluation

Various test cases were run on both Kharita and the trajectory bundling (Bundle) algorithm. We first begin with an overview of a few standard test cases also used in the literature. These make use of the trajectories present in the data-sets. We will then focus on the bulk of our test cases, which involve the synthetically generated trajectory sets.

4.1.1 Data-set trajectory test cases

Using the framework and methodology described in 3.1, the public data-set test cases were run. These test cases correspond to those that are mostly used in the literature. Table 2 shows the f1-scores, precision and recall for both Kharita and Bundle. Running time was about a minute for Kharita, while an hour for the Bundle algorithm.

Name	**Athens-small**	**Chicago**
Kharita	0.29, (0.53, 0.20)	0.23, (0.90, 0.13)
Bundle	0.27, (0.55, 0.17)	0.22, (0.88, 0.13)

Table 2: Road graph sampling evaluation results for public data-sets. Values given as f1-score, (precision, recall).

Performance seems fairly similar for both Kharita and the Bundle algorithm. In general, precision is relatively high, while f1-scores and recall are low. The latter is a direct result from having an unpruned ground-truth map which has a lot of roads that no trajectories cover. Table 3 and Figure 11 highlight this shortcoming of these test cases.

A consequence is that recall is not an absolute measure reflecting the extent to which an algorithm can capture the ground-truth roads. Instead it signifies some aggregate of an impossibility for algorithms to infer roads and an inability to infer roads. For algorithm evaluation we are obviously only interested in the inability to infer roads. Another consequence is that the precision measure also looses a lot of its significance. When recall no longer reflects the decrease of an ability to capture roads, it becomes unfairly advantageous for algorithms that focus on very high precision and low recall. Since our scores no longer penalize such a low recall as much as they should. Most of the quantitative evaluations that have been performed in state-of-the-art algorithm literature [8, 7, 9] have this shortcoming.

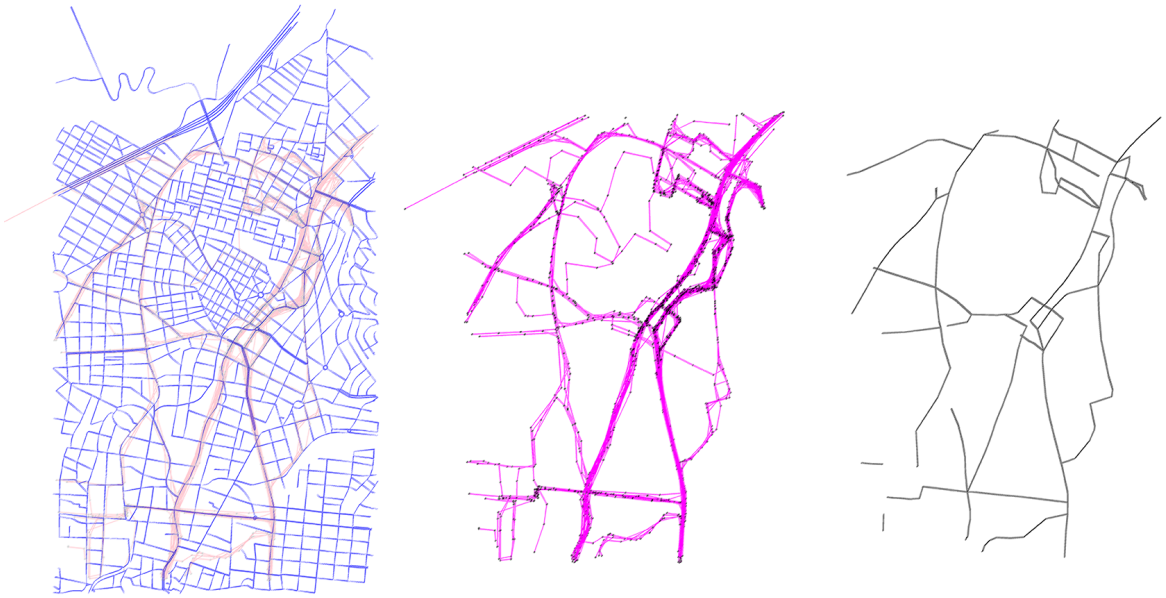

Figure 11: Overview of the Bundle algorithm for the public Athens-small data-set. Left the unpruned ground-truth, middle the provided trajectories and right the inferred map.

Name	**Athens-small**	**Chicago**
Kharita	0.59	0.05
Bundle	0.13	0.03

Table 3: Road length difference constructed map and ground truth map. Calculated as factor $\frac{|CM|}{|GT|}$.

The larger road length of the maps outputted by Kharita are mostly the result of spurious roads. This is part an undesired symptom of the algorithm when GPS interference errors increase, as we will see in the next section. But also part unfairly the consequence of the directed output by Kharita, which is made undirected before being evaluated. Here, an undirected ground truth road can be correctly inferred by Kharita as two directed roads in opposing directions. But after transforming the map to an undirected one, the two now undirected roads will represent the same ground-truth one. Making one of them duplicate. To investigate the unfair effect of this on the $\frac{|CM|}{|GT|}$, we calculated the road length difference also in a directed setting. Here we determined the undirected roads of GT, that were within the tight bounding box of CM, and transformed each of them in to two directed roads. Even in this most favorable case we see $\frac{|CM|}{|GT|} = 0.04$ for Chicago and $\frac{|CM|}{|GT|} = 0.53$ for Athens.

4.1.2 Synthetic generated trajectory test cases

The bulk of our evaluation is performed on synthetic test cases. Figure 12 highlights Kharita evaluated on a single synthetic test case involving the Utrecht map extracted from OSM. The case in question contains $n = 200$ trajectories with precision and sampling error, $\pi = 50$ and $\sigma = 100$. The directed output map was evaluated without transformation in this test. Note that the ground-truth has been perfectly pruned according to the generated trajectories.

The low recall in this case solely reflects the inability of the algorithm to infer the graph correctly. Although the ground truth visually seems to have inferred most roads of the map, presumably a lot of topological errors have been made.

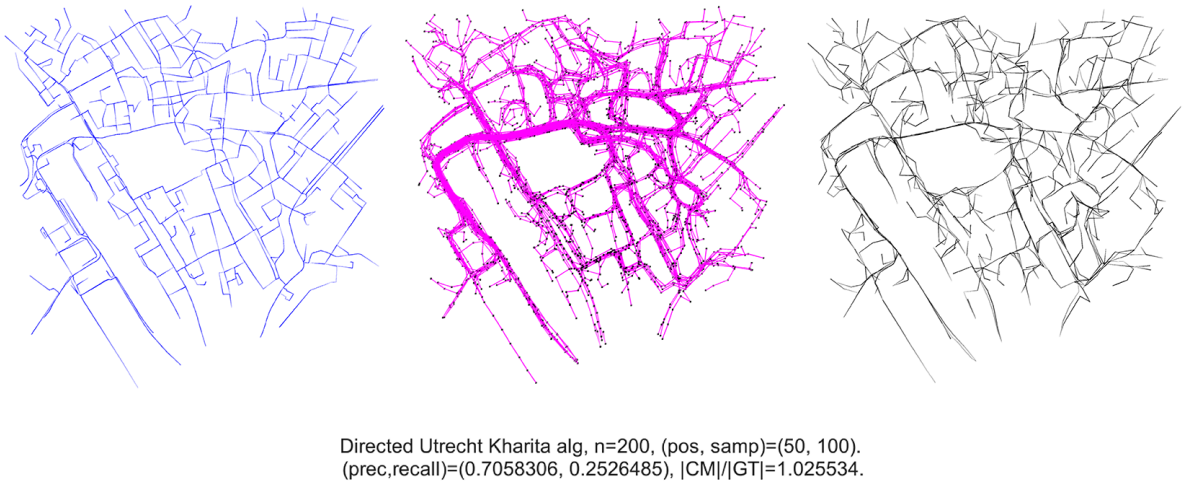

Figure 12: Overview of the Kharita algorithm for trajectories with a moderate interference error on Utrecht. Left the pruned ground-truth, middle the generated trajectories and right the inferred map.

Inferring correct topology is especially hard and error prone for directed road graphs. A significantly lower precision but higher recall of $(precision, recall) = (0.42, 0.83)$ is achieved using Kharita on the transformed undirected version of this test case. In this case incorrect directedness can no longer cause topological errors. We will see the effect that incorrect topology has on the precision and recall more closely in the evaluation measure analysis in 4.2.

Figure 13 shows a complete overview of the performance of Bundle and Kharita on all the Athens test cases using 200 trajectories. Kharita output was transformed undirected in all of them. We refer to the link in appendix A for a visualisation of the trajectories used in these test cases.

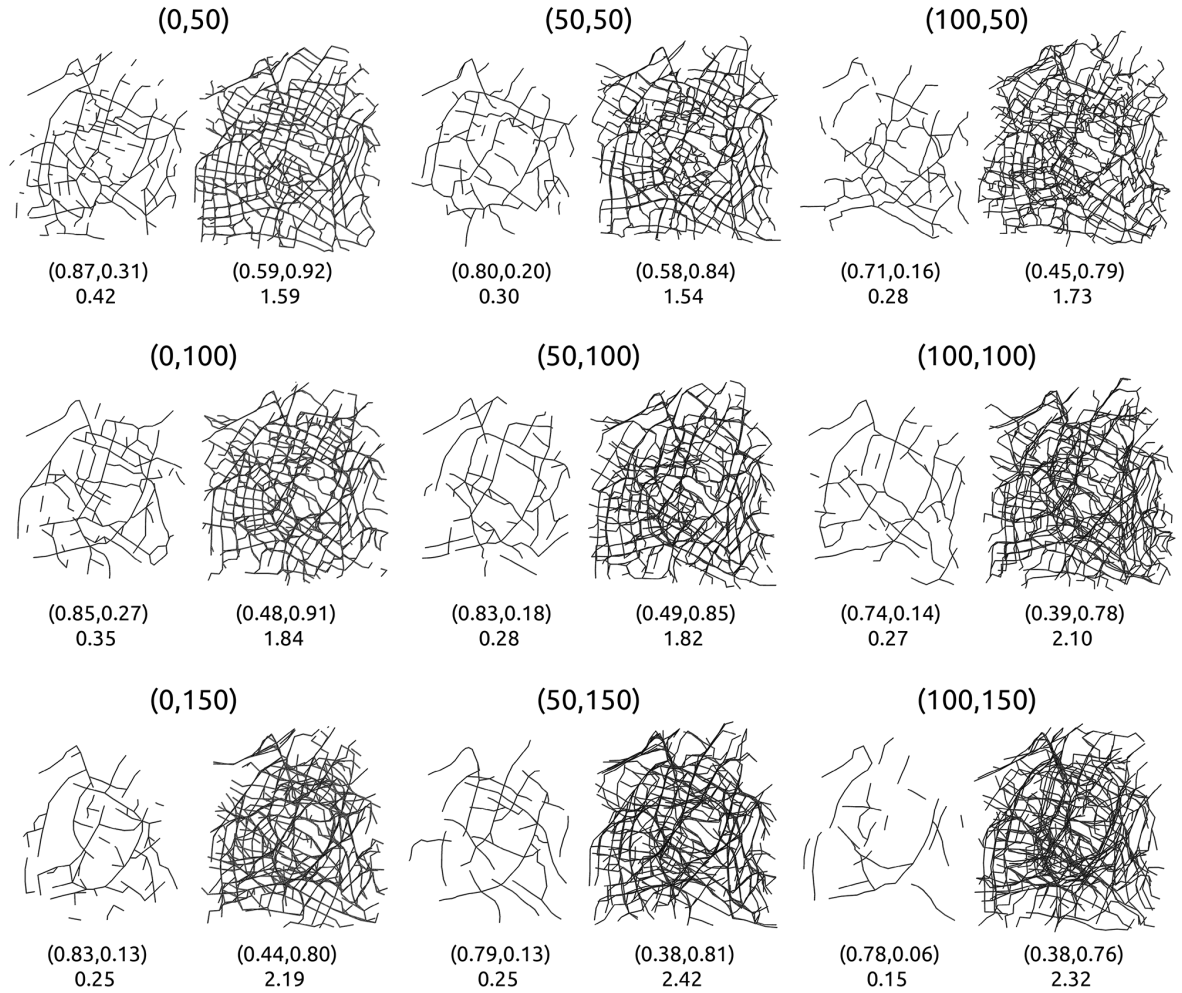

Figure 13: Comparison between Bundle and Kharita output for nine test cases using the Athens map. For each test case, (π, σ) is notated above, Bundle output is left and Kharita output right. Below each output map (precision, recall) is given followed by the factor $\frac{|CM|}{|GT|}$.

Generally, performance deteriorates for both algorithms as precision and sampling error increase. But behaviour of the algorithms is vastly different. Bundle overall prefers precision over recall. Even with no precision error $\pi = 0$ and low sampling error $\sigma = 50$, only a recall of 0.31 is achieved. The small output size in comparison to GT seems to explain most of the low recall. This can be caused by the algorithm pruning away trajectory bundles with a cardinality below a set threshold. When trajectory errors increase, we see a relatively small loss occurring to the high precision. But recall lowers significantly, with only on average 6% of sampled GT points being matched to a CM point in the test case with highest errors. Loss of recall can for a large part be appointed to a smaller ground-truth output. But fragmentation of the result seems to play an additional part, as can be seen in the drawn road networks. No difference can be found

between test cases with high precision error and low sampling error, and cases with low precision error and high sampling error.

For Kharita almost the reverse seems to occur. Even with minimal trajectory errors a precision of only $\pi = 0.59$ is achieved. The high $\frac{|CM|}{|GT|}$ factor indicates spurious roads as possible cause. This can again unfairly be the consequence of the map transformation we perform on the output. When trajectory errors increase, precision score decreases. While the relatively high recall is not effected as much. We see a strong increase in the amount of (spurious) roads outputted when trajectory errors increase. Which can partly explain the diminishing precision. No difference can be found between test cases with high precision error and low sampling error, and cases with low precision error and high sampling error on precision and recall. However, higher sampling errors seem to more strongly cause spurious road in comparison to high precision errors. More interesting is a closer visual comparison of the road graph in these two circumstances, given in figure 14.

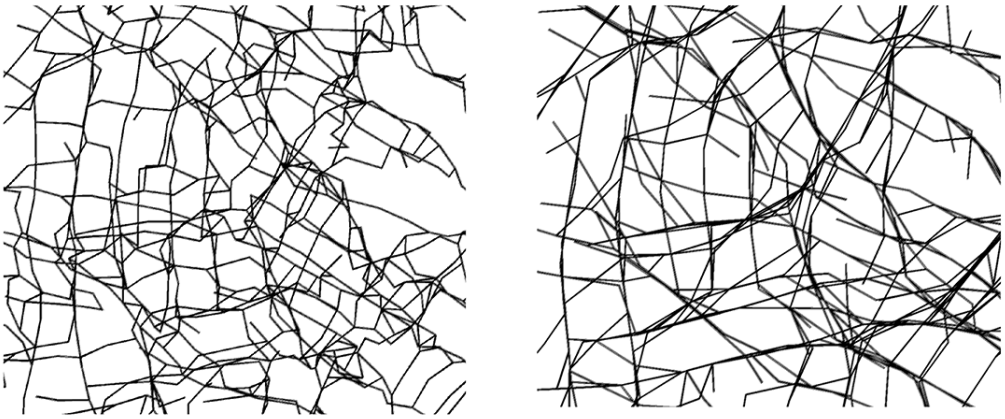

Figure 14: On the left, zoomed snippet of undirected Kharita output map with $(\pi, \sigma) = (100, 50)$. Spurious roads seem to be mostly duplicate roads. On the right, $(\pi, \sigma) = (0, 150)$. Spurious roads seem to be mostly shortcuts at intersections.

A noticeable difference can be seen in the nature of the spurious roads present under these trajectory errors. For high precision errors whole road segments seem to be incorrectly duplicated, resulting in many parallel roads. For high sampling errors, problem occur mostly at intersections, where shortcuts get formed. Especially at complex intersections like roundabouts we see almost clique-like structures. This difference in spurious road manifestation and, which may be a consequence of it, the difference in road map size is poorly reflected by the graph sampling measure output.

Table 4 shows scores on all the Chicago test cases using 200 trajectories.

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.90, 0.26), 0.38	(0.90, 0.21), 0.31	(0.78, 0.17), 0.30
$\sigma = 100$	(0.85, 0.20), 0.30	(0.88, 0.20), 0.31	(0.76, 0.19), 0.28
$\sigma = 150$	(0.82, 0.22), 0.30	(0.80, 0.11), 0.24	(0.71, 0.12), 0.22

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.67, 0.88), 1.33	(0.65, 0.83), 1.30	(0.49, 0.77), 1.50
$\sigma = 100$	(0.61, 0.81), 1.45	(0.56, 0.79), 1.59	(0.45, 0.70), 1.87
$\sigma = 150$	(0.52, 0.76), 1.87	(0.47, 0.73), 1.99	(0.40, 0.70), 2.20

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.84, 0.44), 0.91	(0.78, 0.41), 0.91	(0.59, 0.25), 1.06
$\sigma = 100$	(0.79, 0.30), 1.03	(0.73, 0.36), 1.10	(0.60, 0.31), 1.23
$\sigma = 150$	(0.73, 0.22), 1.23	(0.67, 0.21), 1.37	(0.55, 0.16), 1.55

Table 4: Chicago-200 test case results Bundle top, undirected Kharita middle and directed Kharita bottom. Values given as $(precision, recall), \frac{|CM|}{|GT|}$

Results are similar to that of Athens for the undirected cases. However, the directed test cases of Kharita show different behaviour in comparison to the undirected ones. Substantially less roads are present in the directed maps. The lower amount of spurious roads has a positive effect on the precision. Recall lowers significantly though compared to the undirected map, and even in the case with lowest trajectory errors is twice as low. Although the smaller output may play a part in this, we suspect an increase in topological errors playing a bigger role. We will look at this in more detail in the next section. The result and decrease in performance is not unexpected. The directed version of map inference is a harder problem, since directedness of edges has to be additionally inferred. We refer to appendix A for similar results of all the test cases run on both Utrecht and Athens (Since figure 13 only contains the undirected Athens test case results).

Finally, Figure 15 shows a scatterplot containing the results of all the undirected Utrecht test cases. Both with 100 and 200 trajectories.

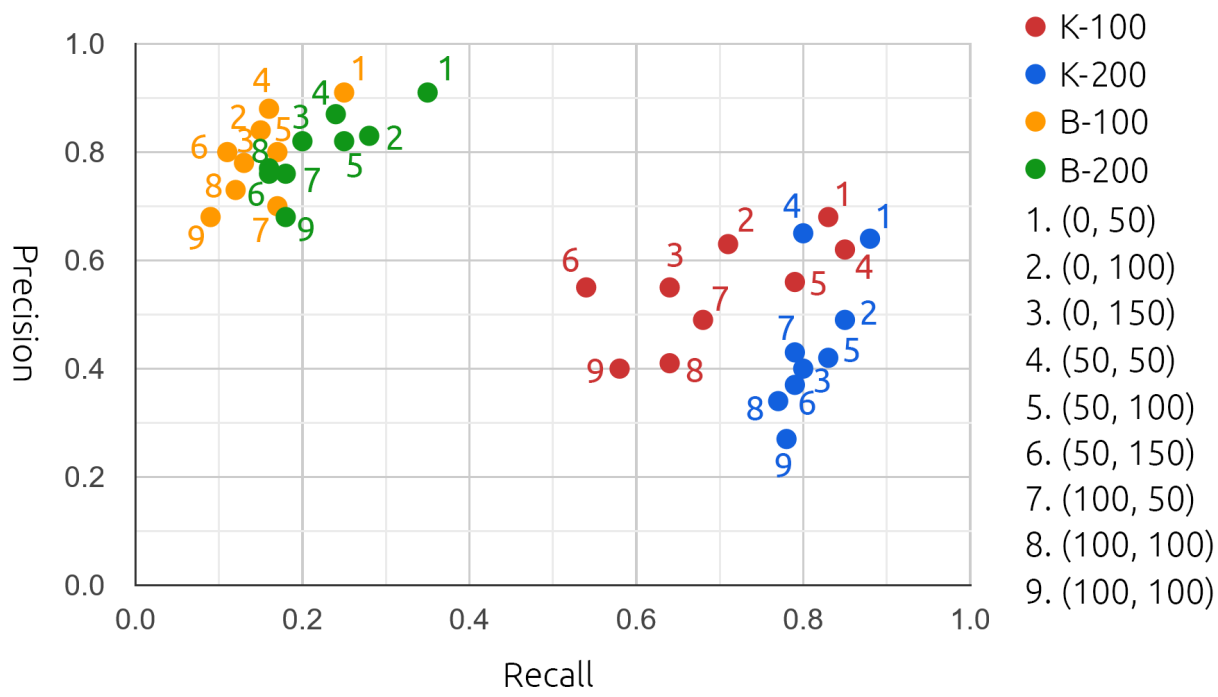

Figure 15: Scatterplot of undirected Utrecht test cases. K-100 indicating Kharita results with 100 trajectories, K-200 Kharita results with 200 trajectories. The same for Bundle with B-100 and B-200. Number alongside each point indicating (π, σ) of the test case according to legend.

Appendix B contains similar plots for Athens and Chicago. Behaviour of both algorithms under different trajectory errors are again similar to those of Athens and Chicago. There is little visible difference in this regard between the 100 and 200 trajectory sets. Only the bigger horizontal spread of the K-100 points being noteworthy, indicating a less stationary recall if Kharita is provided with less trajectories. Performance does increase for most cases with more trajectories. For Kharita we see better recall, but a decrease in precision, resulting in slightly better f1-scores. This growth can likely not be sustained when trajectory sizes increase further, since recall is already high. For Bundle we see in most cases both an increase in recall and a smaller increase in precision. We believe recall can keep increasing if more trajectories are provided. However, the running time will quickly become a limiting factor. Even for only 200 trajectories the running time was an hour, while performance does not scale linearly in the case of Bundle.

4.2 Evaluation measure analyses

For the purpose of investigating the performance of the graph sampling evaluation measure, a couple of analyses were performed. We will first look qualitatively at how well local map errors are captured in sampled neighbourhoods. Then we will look at how local neighbourhood scores vary along roads of the map. Finally, we will see how the global aggregate of neighbourhood scores converges as more neighbourhoods are sampled.

4.2.1 Local neighbourhood analysis

Based on the results of our algorithm evaluation, we chose to focus on the directed Kharita output for our local neighbourhood analysis. These road graphs are especially interesting since precision and recall value on first glance do not correspond with expectations based on the visual output. Thus along with investigating the strengths and shortcomings of our evaluation measure, we can get a better idea what caused the unexpected values in those cases. Figure 16 shows the directed Kharita test case we performed our analysis on. Figure 17 shows various identified neighbourhood errors that were successfully captured by the neighbourhood matching. Precision and recall per sample are additionally provided.

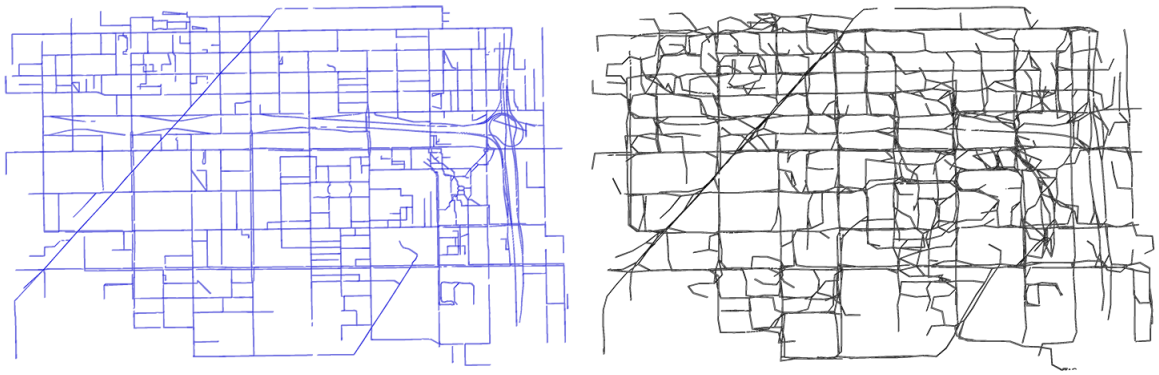

Figure 16: Overview of the directed Chicago test case on which the neighbourhood analysis was performed. CM being Kharita output from 200 trajectories with $(\pi, \sigma) = (50, 100)$. On the left GT and on the right CM. Note a recall being evaluated of only 0.356514, despite CM capturing GT well on a high-level and $\frac{|CM|}{|GT|}$ being close to one.

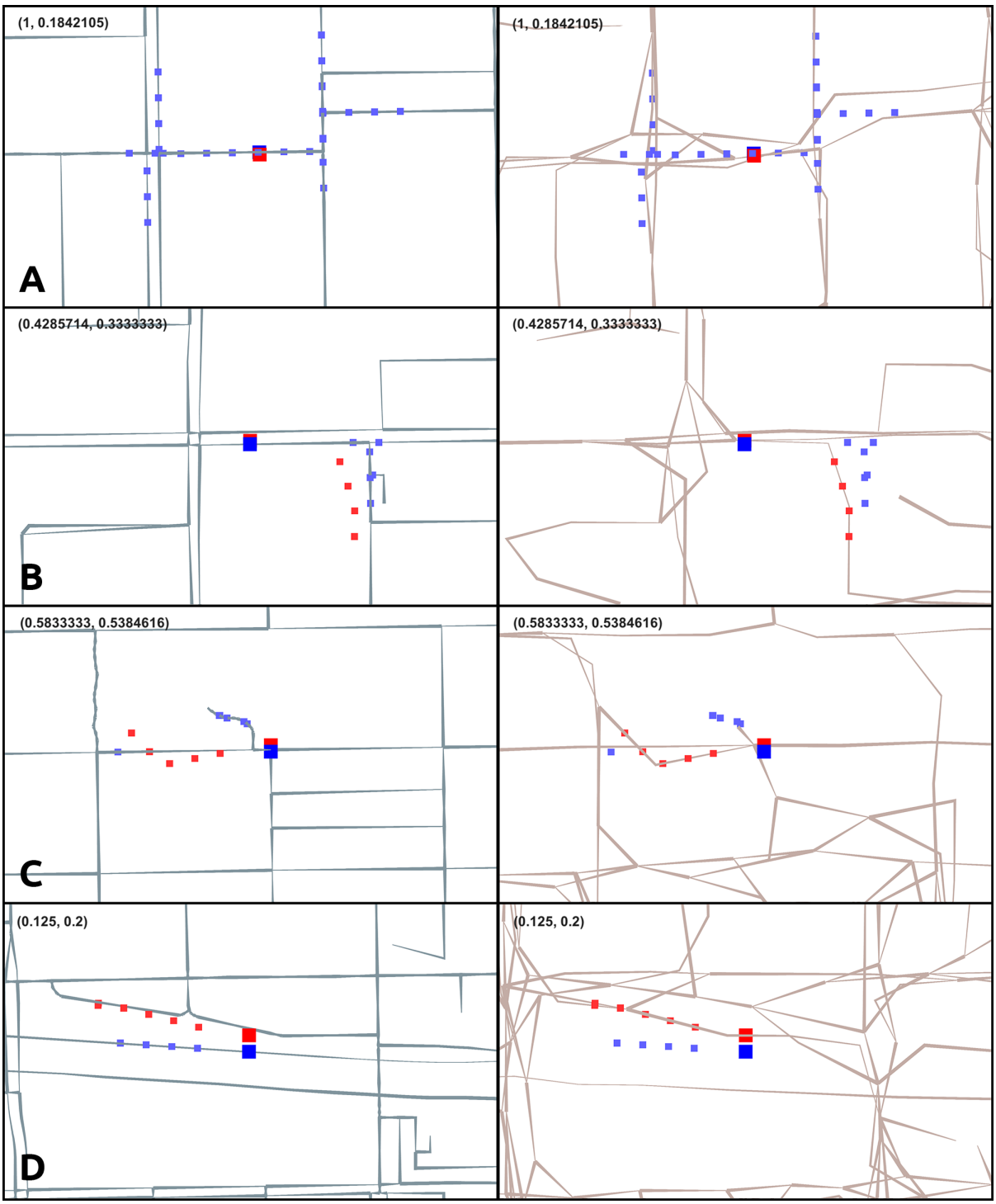

Figure 17: Overview of various sampled local neighbourhoods containing errors. For each row, blue points show unmatched GT points that contribute to recall, red points show unmatched CM points that contribute to precision. Origins are indicated by bigger points. On the left points are overlayed on GT, right on CM. Precision and recall score of the sample in question is displayed in top-left corners. A highlights incorrect intersection errors, B geometric error, C both a missing road and spurious road and D merged parallel roads.

Of the errors Figure 17 shows, A is especially prevalent. 54% of all the neighbourhoods we evaluated in the chosen map contained to some degree incorrectly inferred intersections in the sampled region. The main cause of this is that multiple intersections are inferred for what is actually a single intersection in the GT. This results in lower degree intersections, between which the connectivity of the original intersection is distributed. Because of this lower local connectivity, exploration from the origin diverges to a smaller extent. The result is that CM neighbourhoods tend to have less points, do not capture some near roads and are more centered around the origin compared to the GT neighbourhoods. Because only a small amount of points are close to the origin of the CM neighbourhood, precision is more likely to be high. Since errors between CM and GT neighbourhoods tend to accumulate the further away from the origin we go. At the same time, the points of the GT neighbourhood on roads that have not been captured due to the splintered intersections, can not be matched to any CM point. This results in a lower recall. The frequency of this error combined with the resulting high precision and low recall score, explains the results we got for directed Kharita test cases in section 4.1.2.

The other errors shown were less frequent in our chosen test case, but showed to be successfully captured by the measure to some degree. While incorrect intersections mostly resulted in only lower recall, geometric errors are reflected in both lower precision and recall. However, the extent of the geometric error plays an important factor in its capture. Errors that are smaller than the matching distance threshold d are left completely uncaptured. This corresponds with findings of the "road shape error" in Chao et al. [5]. Missing roads and spurious roads cause as expected lower recall and lower precision respectively. While the merged parallel road in D shows a reflection in both precision and recall, this was not always the case. In some samples the ground truth origin did not lie on the incorrectly merged edge. In these cases all CM points can often be matched, causing only unmatched GT points. This is reflected in solely a lower recall.

Lastly, Figure 18 shows a situation that is not captured well by the matched neighbourhood.

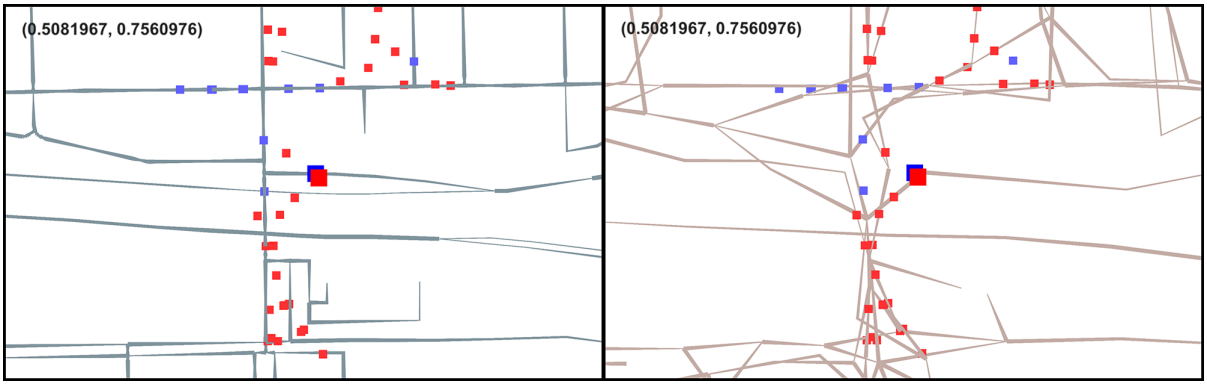

Figure 18: Overview of a sample for which the neighbourhood sampling matching shows shortcoming reflection.

In this sample, both geometric and topological errors are present to a large degree. But the close proximity of a lot of roads causes large amounts of points to be sampled for both GT an CM in a relatively small region. Even though CM points are sampled erratically throughout this region, a lot of matchings can still be made. This results in a precision and recall that is higher than should be expected based on the visual extent of the errors. Occurrence of such similar errors was infrequent, with only 4% of the investigated neighbourhoods containing it to some degree.

4.2.2 Global neighbourhood aggregation analysis

We start the global neighbourhood aggregation analysis with the investigation of neighbourhoods on a single road and how their scores change over the span of it. Figure 19 shows the precision and recall from 500 evenly distanced neighbourhoods over the entire span of a single example road.

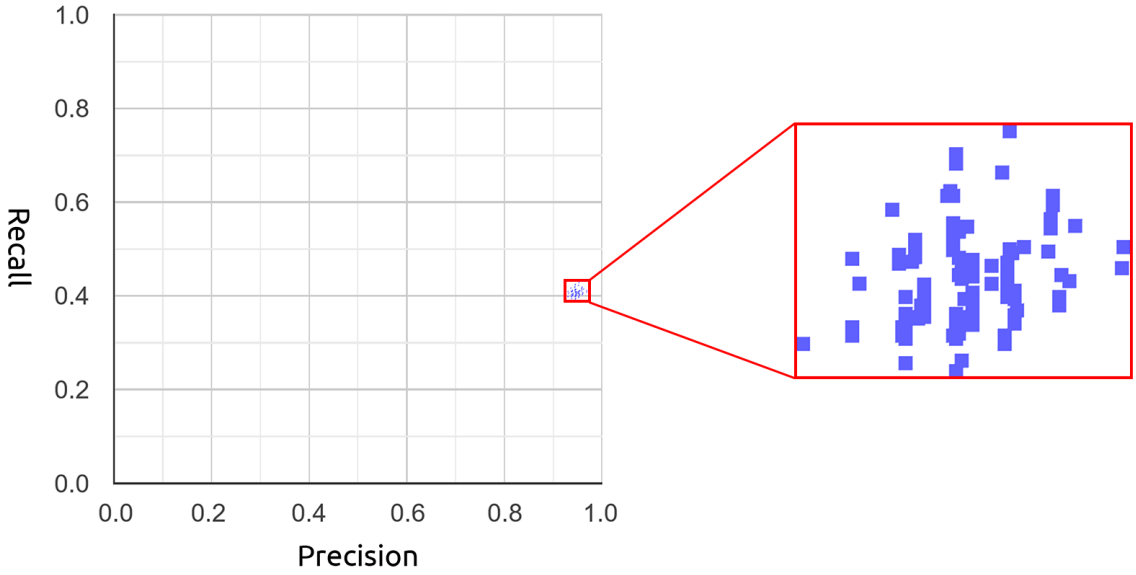

Figure 19: Overview of precision and recall scores for neighbourhoods sampled over a single road. On the left, scatter plot containing the various scores of evaluated neighbourhoods. Due to very low variance, small points had to be used to distinguish values. On the right a zoomed view of the area containing the scores.

For most roads a similar result with low variance was found, where different scores from the same road at most differ 0.05 from eachother on both precision and recall. On first glance the low variance does indicate the possibility of using only one representative sample for an entire road when estimating. However, roads with high variance scoring between neighbourhoods were also present. These roads tended to be evaluated with

significantly lower running time. Indicating the possibility of neighbourhood size having to do with it. To investigate, we experimented with changing the radius parameter r in the same setup. Figure 20 looks at the scoring of 20 evenly spaced neighbourhoods over the entire span of a road. Multiple roads were evaluated for four test cases with changing r.

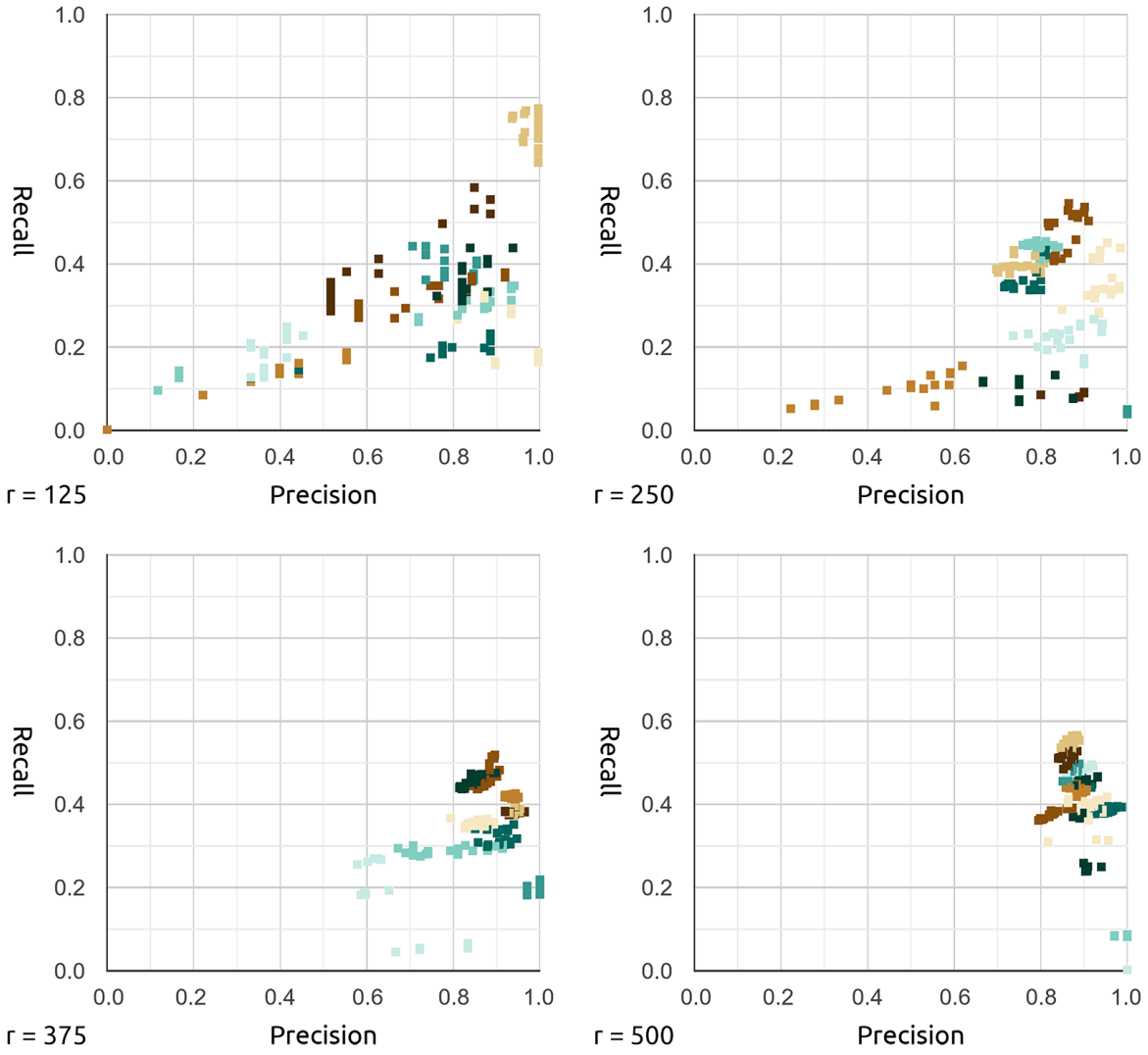

Figure 20: Overview of different test cases where precision and recall scores are plotted for neighbourhoods over the span of multiple roads. Colors indicating neighbourhoods sampled over the same road. Radius r varies for the four test cases, with exact value shown in bottom-left corner of the scatter plots.

For test cases with a low r we mostly see unstable changes in precision and recall for neighbourhoods along a road. Very few stable segments are present in most roads. A visual observation showed that when stable, only slight translations occurred to points

in neighbourhoods of adjacent origins. The biggest observed differences were the result of significant changes occurring to the position of the CM origin. This happens when the GT origin moves to a new point for which the closest CM origin lies on a different edge than for the previous point. The radius r however, has a major influence on this. For higher r we can explore more road graph before reaching the radius threshold. This makes the amount of representative points of a neighbourhood larger. A smaller percentage of points are near the origin, as more points are sampled further away. This makes the same translation of the CM origin have less impact on score. Additionally, when neighbourhood are defined by many points, the addition or deletion of a point, due to it moving in or out of the radius for example, has a much smaller effect on the score. Because of this we see that for higher r values, both the variance of neighbourhoods on a single road, as well as the variance of neighbourhoods between different roads, become significantly smaller.

Lastly we investigate convergence of our estimate as we sample and aggregate more neighbourhood similarities. Figure 21 to 24 show convergence of precision over amount of sampled points for various test cases.

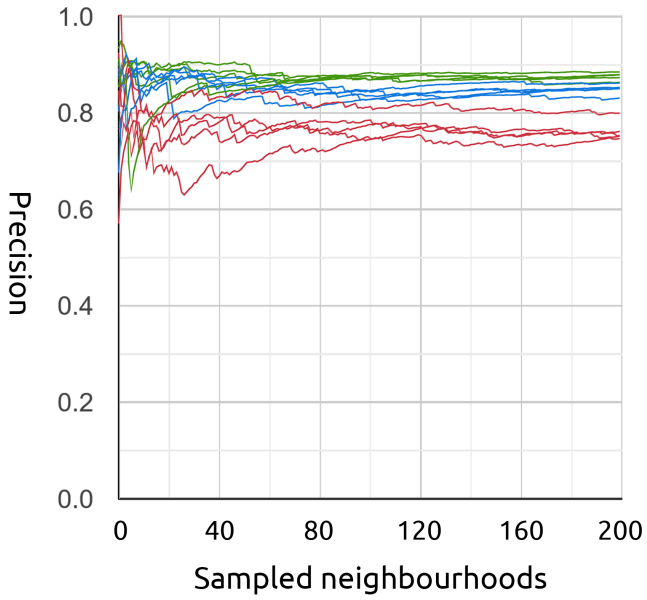

Figure 21: Precision estimate convergence over 200 neighbourhoods for test cases with varying radius r. Red indicating $r = 150$, blue $r = 375$ and green $r = 500$.

In Figure 21 we see a convergence of the precision for test cases with varying radius r corresponding to our previous finding. For test cases with larger r our aggregate converges faster. Different random road sampling strategies plotted in Figure 22 do not show significant differences, even though the exact measure we are trying to estimate changes. The uniform random sampling does seem to converge slightly faster, with less

variance at 200 aggregated samples.

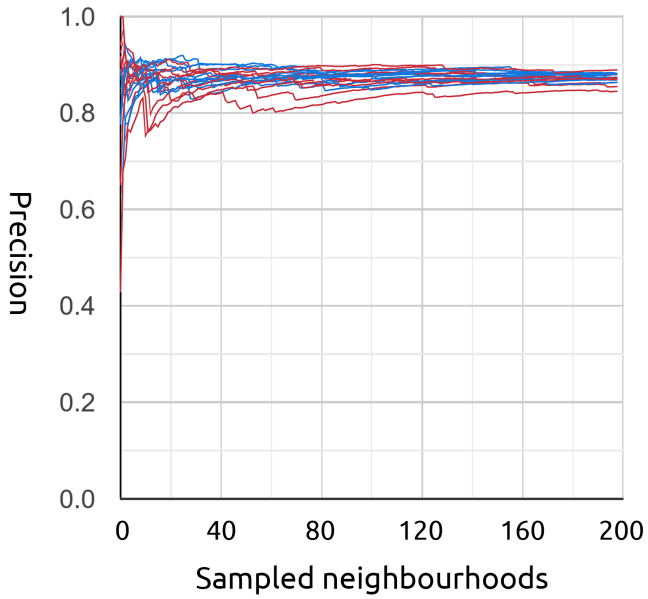

Figure 22: Precision estimate convergence over 200 neighbourhoods for test cases with different random sampling strategies. Red indicating a weighted random road sampling strategy and blue a uniform random sampling strategy.

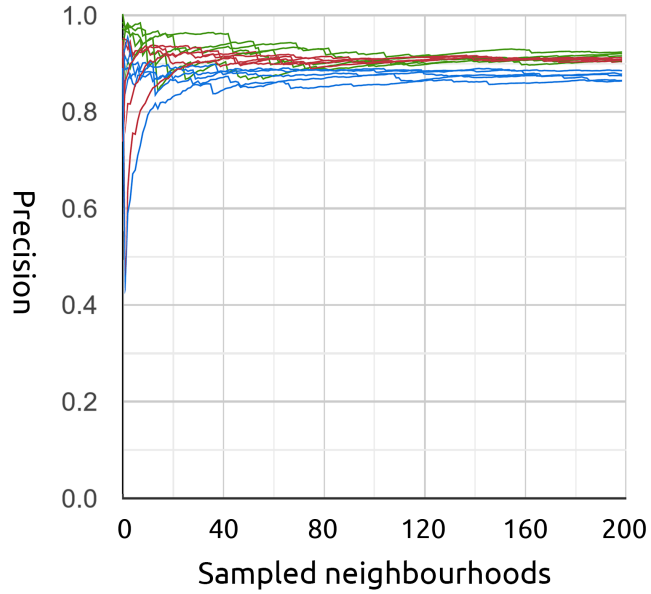

Figure 23: Precision estimate convergence over 200 neighbourhoods for test cases with different maps. Blue indicating Utrecht, red Athens and green Chicago.

Figure 23 shows the difference in convergence for test cases using varying maps with different sizes. No significant differences are visible. The blue Utrecht test cases do show slightly more variance. Which is unexpected considering the smaller map size.

Lastly, Running test cases with different trajectory errors, shown in Figure 24, only produces significantly slower convergence for $(\pi, \sigma) = (100, 150)$. For the trajectory bundling algorithms the output for this test case contained very few roads. Again indicating map size is possibly playing a role. A guess might be that neighbourhood sizes for these small maps remain relatively small. We have seen in the previous experiments that when neighbourhood sizes are small, even neighbourhoods sampled on the same road exhibit a lot of variance. This can play a factor in the slower convergence of these test cases.

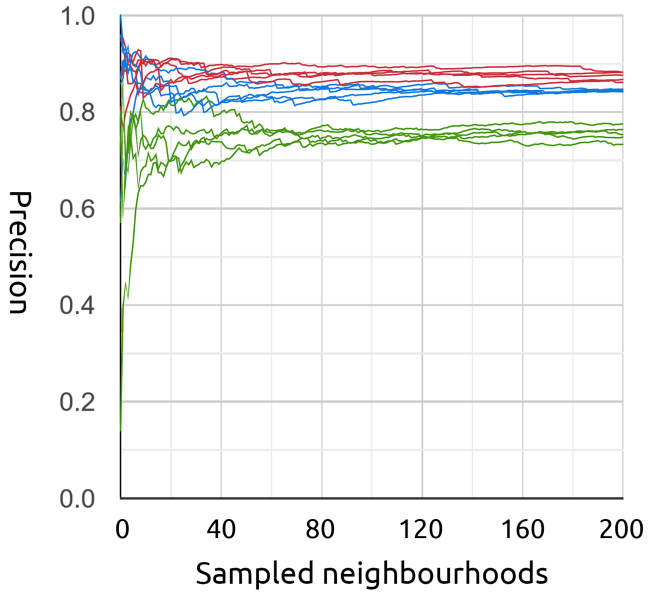

Figure 24: Precision estimate convergence over 200 neighbourhoods for test cases with different trajectory errors. Red indicating $(\pi, \sigma) = (0, 50)$, blue $(\pi, \sigma) = (50, 100)$ and green $(\pi, \sigma) = (100, 150)$.

5 Conclusion

Coming back to our original aim of determining how to accurately evaluate road map construction algorithms. We stated that two problems were present with previous methods of evaluation. One related to variety and quality of used test cases and the other related to the quality and understanding of the graph sampling similarity measure being used. To address these two problems we made several contributions.

For the test cases, we presented our own framework. This new framework, which has been made publicly available, made it possible to investigate the performance of our algorithms to a much better degree.

Part of our novel framework was an approach for synthetic trajectory generation. We have shown that the usage of synthetic trajectory generation resulted in multiple significant benefits in comparison to the traditional evaluation method currently being used in the literature. The first being the possibility of creating diverse test cases to evaluate algorithms under different circumstances. The second being the possibility of using any section of road graph that is publicly available. This not only opens up the ability of investigating more diverse topology from different regions of the world, but also gives control over the size of the trajectory set and the size of the map on which we generate them. And the third being the possibility of pruning our ground truth map perfectly, to only include edges that are used by at least some trajectories. Current public data-sets do not provide us with this information. Strictly speaking, it is not impossible to make data-sets that contain this information. But making such new sets is expensive and would take time, disregarding even the public availability aspect. Another approach can be to infer the usage of edges by map matching the trajectories. But this creates additional bias that is undesired. We strongly discourage qualitative evaluation using the graph sampling measure without a pruned ground truth map. As we saw that recall suffers immeasurably from it, making it impossible to determine performance of map construction algorithms accurately. Additionally, we strongly recommend to measure and provide ground truth and constructed map road length differences and visualisations. As we have seen, these help provide insight to untangle the possible causes of evaluation scores.

The test cases we then presented, showed substantially changing behaviour for algorithms when trajectory errors increased. This part of algorithm assessment is missing in previous evaluations performed in the literature. But is important for understanding general performance of algorithms. Kharita outperformed the bundle algorithm significantly in recall, and did not perform much worse in precision for cases with low trajectory errors. When trajectory errors increased however, the output of Kharita exhibited a lot of spurious roads. Which strongly lowered the precision of the map. The trajectory bundling algorithms outputted very precise sections of the map, but recall remained low. However, test cases with more trajectories showed a positive trend, where both

precision and recall improved. This is opposed to Kharita, where only recall improved and precision lowered. Investigating this trend further would have been interesting, but because of the high running time of the Bundling algorithm, this was out of scope for our research. Surprisingly, the usage of different maps did not provide much difference in behaviour and provided no additional insight about performance.

A problem with the graph sampling similarity measure is mainly the lack of research in the literature about its strengths, shortcomings and usage, while at the same time being the mostly used quantitative evaluation. For analysing the strengths and shortcomings of it, we performed a number of experiments. We separated our analysis in to a part investigating local neighbourhood similarity measurement, and a part investigating the aggregation of local scores in to a global road map similarity measure.

By looking at the neighbourhoods capturing a particular kind of error with its evaluation, we showed how it reflects in precision and recall. We also encourage this type of analysis for cases where unexpected global similarity scores occur that do not correspond with visual output. This helped in our case shed light on the evaluation of Kharita on some directed test cases, where we saw that the splintering of many intersections was the prime cause of high precision and low recall scores. For the neighbourhood evaluation, shortcomings of capturing certain errors lined up with previous observations from literature pertaining inability of capturing small geometric errors. However, our evaluation did show in some cases shortcoming in more extreme cases of geometric error and topological error that have not been discussed before in literature. Where the neighbourhood was evaluated well, while visuals showed poor erratic inference of the map. For capturing topological errors good performance was shown. However, when exploration was hampered at intersections, precision scores became overly high. And while reflection of many topological errors is generally good, untangling their individual effects becomes hard at global level. Highlighting again the importance of additional data like road length and visuals to make sense of scores.

Finally, our aggregation analysis shed light on usage of the evaluation measure that has not been addressed before. Throughout all the analyses, neighbourhood radius r showed to play the biggest factor for variance between neighbourhoods and convergence of our estimate to the exact similarity value. Where an increase in radius size, shows faster convergence. On a single road we found the movement of the CM origin to cause the biggest shift in scores. Different maps, random sampling strategies and trajectory errors all surprisingly did not have a significant effect on convergence of our global average. Even though using a different random sampling strategy changes the exact measure we were trying to estimate.

We think these insights in the graph sampling evaluation measure, our recommendations for its use and our frameworks which strongly enhances the mapping of algorithms performance, can already greatly increase the quality of quantitative evaluation in the field. Even if smaller recommendations, like the use of a pruned ground truths, becomes

universal in the field. Research for better map construction algorithms will have a stronger foundation with which targeted improvement can take place.

6 Future works

Many directions can be taken for future work. Evaluating more algorithms using our novel framework would be an obvious one. Especially older well-known algorithms and algorithms from different categories than the two we covered, would be a first choice in our opinion. Another important direction is to look at multiple instances of the same test cases. Right now we only used one instance for a certain test case, corresponding to how evaluation is performed with public data-set test cases. Synthetic trajectory generation opens up the possibility to create multiple trajectory sets of the same test case. Investigating the variance across these trajectory sets would provide interesting insight.

Although we did not gain much from using different maps in our evaluation, we think a non-urban map might still be an interesting new test case. Especially since we have the possibility of generating many trajectories in remote places. Another possibility would be in improving our synthetic trajectory generation. Even though output already looks natural, the addition of road speed data can add more heterogeneity to the trajectory set, which at the moment is missing. These road speeds are easily obtainable via the OSM data pipeline. However, we do feel more advanced trajectory generation can deter the clarity of current test cases. It can also make results overly complicated, since already over a hundred trajectory sets and corresponding pruned road maps were used in our algorithm evaluation. For the algorithms themselves we would especially like to see an improved implementation for the trajectory bundling algorithm. So that the positive trend in performance that occurs with more input trajectories can be fully investigated.

For road graph similarity we think there is much that can be done as well. We only touched the radius parameter of the graph sampling similarity measure in our analyses. The influence of the other parameters is still not well known. We think there is also importance and a possibility in defining an exact measure using graph sampling. Defining such an exact measure would allow the pursuit of an exact method of calculating similarity. But even if performance of an exact method of calculation is unreasonable, having it can give more insight in to estimating it efficiently. Additionally, an exact definition would take away the different possible variations of the graph sampling measure that are being used in the literature, and would force consistency between different papers in their evaluation methodology. We think an approach where identifying all events where a local neighbourhood matching can change, might prove to be successful.

Finally, the possibility of devising a novel road graph similarity measure remains. Previous attempts have not proven to be much successful. But the graph sampling measure has, like we have shown, significant shortcomings that a new measure can improve on. We think a research proposing a novel measure should consist of three parts for successful adoption by the field. The first part involving the design of a set of well defined quantitative and qualitative experiments with which preference over the graph sampling evaluation measure can be shown and justified. We think starting with this part is a beneficial first step. The second part simply involves the design of such a new measure.

Finally, the third part should involve providing an easy-to-use performant public implementation of the novel measure. We think all of these parts are equally important, since not only designing a good similarity measure, but also convincing the field of it use over the former measures will be a necessity for success.

References

[1] James Biagioni and Jakob Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation research record*, 2291(1):61–71, 2012.

[2] Tomtom. Coronavirus on the map: how tomtom digitizes temporary road changes, https://www.tomtom.com/blog/moving-world/covid-19-road-closures/. 04-01-2021.

[3] Pankaj Pratap Singh and RD Garg. Automatic road extraction from high resolution satellite image using adaptive global thresholding and morphological operations. *Journal of the Indian Society of Remote Sensing*, 41(3):631–640, 2013.

[4] Chao Tao, Ji Qi, Yansheng Li, Hao Wang, and Haifeng Li. Spatial information inference net: Road extraction using road-specific contextual information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:155–166, 2019.

[5] Pingfu Chao, Wen Hua, Rui Mao, Jiajie Xu, and Xiaofang Zhou. A survey and quantitative study on map inference algorithms from gps trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[6] Mahmuda Ahmed and Carola Wenk. Constructing street networks from gps trajectories. In *European Symposium on Algorithms*, pages 60–71. Springer, 2012.

[7] Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Jorren Hendriks, Erfan Hosseini Sereshgi, Vera Sacristán, Rodrigo I Silveira, Jorrick Sleijster, Frank Staals, and Carola Wenk. Improved map construction using subtrajectory clustering. In *Proceedings of the 4th ACM SIGSPATIAL Workshop on Location-Based Recommendations, Geosocial Networks, and Geoadvertising*, pages 1–4, 2020.

[8] Songtao He, Favyen Bastani, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, and Sam Madden. Roadrunner: improving the precision of road network inference from gps trajectories. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 3–12, 2018.

[9] Rade Stanojevic, Sofiane Abbar, Saravanan Thirumuruganathan, Sanjay Chawla, Fethi Filali, and Ahid Aleimat. Kharita: Robust map inference using graph spanners. *arXiv preprint arXiv:1702.06025*, 2017.

[10] Sophia Karagiorgou, Dieter Pfoser, and Dimitrios Skoutas. A layered approach for more robust generation of road network maps from vehicle tracking data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 3(1):1–21, 2017.

[11] Xuemei Liu, James Biagioni, Jakob Eriksson, Yin Wang, George Forman, and Yanmin Zhu. Mining large-scale, sparse gps traces for map inference: comparison of

approaches. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 669–677, 2012.

[12] James Biagioni and Jakob Eriksson. Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 79–88, 2012.

[13] Mahmuda Ahmed, Sophia Karagiorgou, Dieter Pfoser, and Carola Wenk. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, 19(3):601–632, 2015.

[14] David Duran, Vera Sacristán, and Rodrigo I Silveira. Map construction algorithms: a local evaluation through hiking data. *GeoInformatica*, pages 1–49, 2020.

[15] Mahdi Hashemi. A testbed for evaluating network construction algorithms from gps traces. *Computers, Environment and Urban Systems*, 66:96–109, 2017.

[16] Mahmuda Ahmed, Sophia Karagiorgou, Dieter Pfoser, and Carola Wenk. Map construction algorithms. In *Map Construction Algorithms*, pages 1–14. Springer, 2015.

[17] Kevin Buchin, Maike Buchin, David Duran, Brittany Terese Fasy, Roel Jacobs, Vera Sacristan, Rodrigo I Silveira, Frank Staals, and Carola Wenk. Clustering trajectories for map construction. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10, 2017.

[18] Stefan Edelkamp and Stefan Schrödl. Route planning and map inference with global positioning traces. In *Computer science in perspective*, pages 128–151. Springer, 2003.

[19] Lili Cao and John Krumm. From gps traces to a routable road map. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 3–12, 2009.

[20] Sophia Karagiorgou and Dieter Pfoser. On vehicle tracking data-based road network generation. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 89–98, 2012.

[21] Navigation National Coordination Office for Space-Based Positioning and Timing. Gps accuracy, https://www.gps.gov/governance/excom/nco/. 04-01-2021.

[22] Krista Merry and Pete Bettinger. Smartphone gps accuracy study in an urban environment. *PloS one*, 14(7):e0219890, 2019.

[23] Marek Dziewicki and Cezary Specht. Position accuracy evaluation of the modernized polish dgps. *Polish Maritime Research*, 16(4):57–61, 2009.

[24] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 352–361, 2009.

[25] Daniel Chen, Leonidas J Guibas, John Hershberger, and Jian Sun. Road network reconstruction for organizing paths. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1309–1320. SIAM, 2010.

[26] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

[27] Otfried Cheong, Joachim Gudmundsson, Hyo-Sil Kim, Daria Schymura, and Fabian Stehn. Measuring the similarity of geometric graphs. In *International Symposium on Experimental Algorithms*, pages 101–112. Springer, 2009.

[28] Erich Leo Lehmann. *Elements of large-sample theory*. Springer Science & Business Media, 2004.

7 Appendix A: Evaluation tables Athens and Utrecht

Results for Athens and Utrecht test cases are provided in table 5 and 6. Visualisations of individual test cases are additionally available on:
github.com/AriSaadon/RoadMapConstructionEvaluation/tree/main/ExperimentPlots

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.87, 0.31), 0.42	(0.80, 0.20), 0.30	(0.71, 0.16), 0.28
$\sigma = 100$	(0.85, 0.27), 0.35	(0.83, 0.18), 0.28	(0.74, 0.14), 0.27
$\sigma = 150$	(0.83, 0.13), 0.25	(0.79, 0.13), 0.25	(0.78, 0.06), 0.15
Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.59, 0.92), 1.59	(0.58, 0.84), 1.54	(0.45, 0.79), 1.73
$\sigma = 100$	(0.48, 0.91), 1.84	(0.49, 0.85), 1.82	(0.39, 0.78), 2.10
$\sigma = 150$	(0.44, 0.80), 2.19	(0.38, 0.81), 2.42	(0.38, 0.76), 2.32
Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.84, 0.53), 0.87	(0.77, 0.47), 0.93	(0.54, 0.45), 1.25
$\sigma = 100$	(0.74, 0.48), 1.10	(0.68, 0.44), 1.17	(0.52, 0.32), 1.50
$\sigma = 150$	(0.67, 0.29), 1.50	(0.63, 0.31), 1.63	(0.42, 0.38), 1.98

Table 5: Athens-200 test case results Bundle top, undirected Kharita middle and directed Kharita bottom. Values given as $(precision, recall)$, $\frac{|CM|}{|GT|}$

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.91, 0.35), 0.40	(0.87, 0.24), 0.31	(0.76, 0.18), 0.29
$\sigma = 100$	(0.83, 0.28), 0.38	(0.82, 0.25), 0.31	(0.76, 0.16), 0.24
$\sigma = 150$	(0.82, 0.20), 0.28	(0.77, 0.16), 0.27	(0.68, 0.18), 0.27

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.64, 0.88), 1.38	(0.65, 0.80), 1.34	(0.43, 0.79), 1.81
$\sigma = 100$	(0.49, 0.85), 1.70	(0.42, 0.83), 1.93	(0.34, 0.77), 2.30
$\sigma = 150$	(0.40, 0.80), 2.21	(0.37, 0.79), 2.25	(0.27, 0.78), 2.96

Error	$\pi = 0$	$\pi = 50$	$\pi = 100$
$\sigma = 50$	(0.82, 0.49), 0.73	(0.78, 0.38), 0.74	(0.61, 0.22), 1.98
$\sigma = 100$	(0.71, 0.36), 0.90	(0.71, 0.25), 1.03	(0.54, 0.29), 1.32
$\sigma = 150$	(0.66, 0.30), 1.25	(0.62, 0.25), 1.39	(0.52, 0.28), 1.64

Table 6: Utrecht-200 test case results Bundle top, undirected Kharita middle and directed Kharita bottom. Values given as $(precision, recall), \frac{|CM|}{|GT|}$

8 Appendix B: Evaluation plots Athens and Chicago

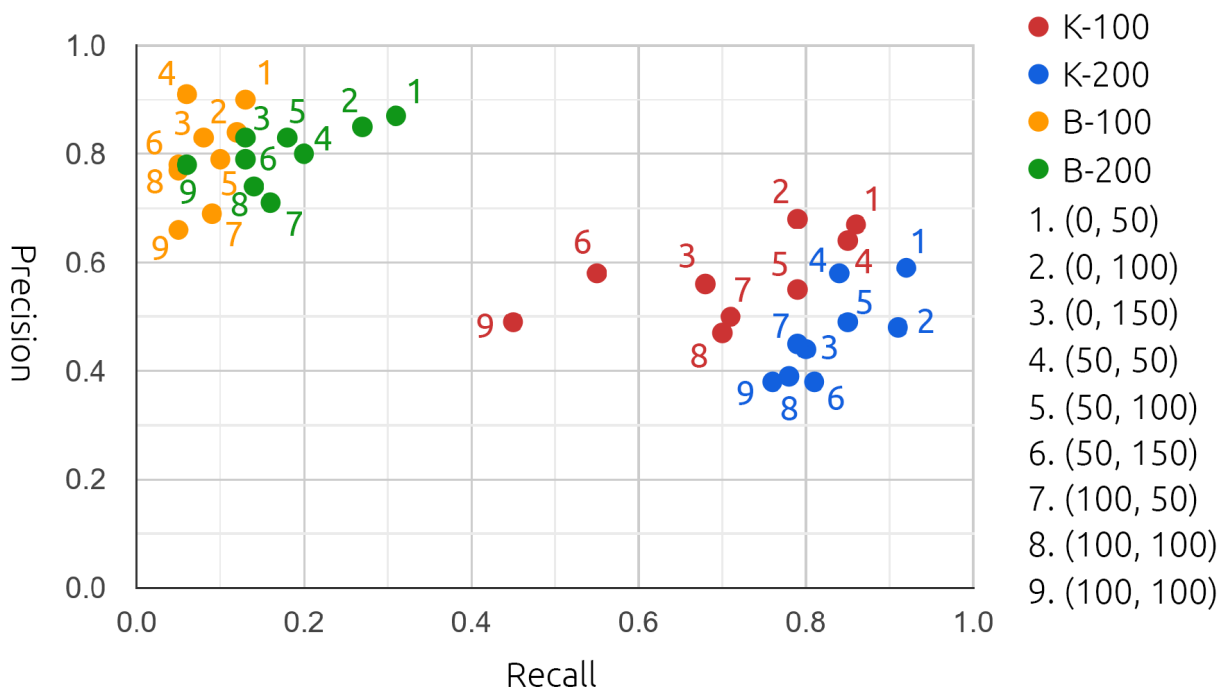

Figure 25: Scatterplot of undirected Athens test cases. K-100 indicating Kharita results with 100 trajectories, K-200 Kharita results with 200 trajectories. The same for Bundle with B-100 and B-200. Number alongside each point indicating (π, σ) of the test case according to legend.

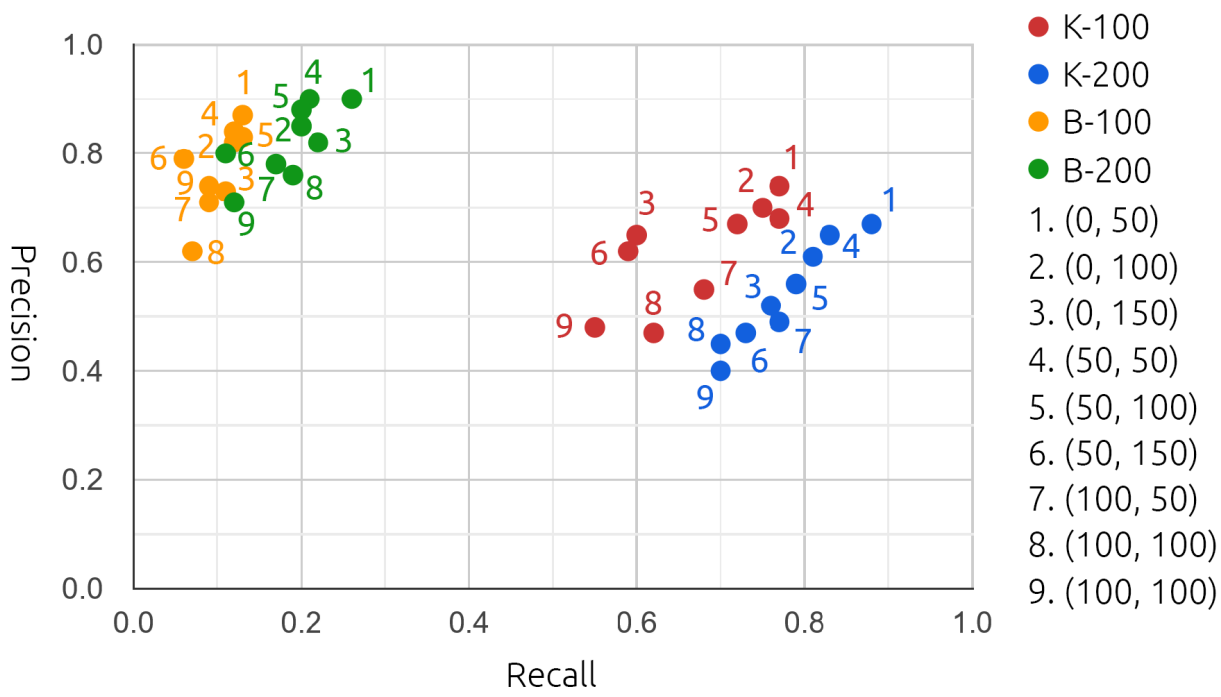

Figure 26: Scatterplot of undirected Chicago test cases. K-100 indicating Kharita results with 100 trajectories, K-200 Kharita results with 200 trajectories. The same for Bundle with B-100 and B-200. Number alongside each point indicating (π, σ) of the test case according to legend.