Probabilistic Street-Intersection Reconstruction from GPS Trajectories: Approaches and Challenges*

Mahmuda Ahmed
Department of Computer Science
University of Texas at San Antonio
mahmed@cs.utsa.edu

Carola Wenk
Department of Computer Science
Tulane University
cwenk@tulane.edu

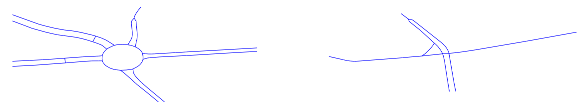
ABSTRACT
Analyzing and mining geo-referenced trajectory data has different aspects to researchers from different communities. For example, animal location data provides ecologists live points of contact between ecologies and the species. And by studying movements of individual animals, they have gained insight into population distributions, important resources, dispersal settings, social interaction or general patterns of how the space was used in an ecological system. Similarly, geologists and environmentalists use earthquake positional data for predicting the location of the next earthquake. Intelligent Transportation Systems and GIS communities use heuristic algorithms on vehicle trajectory data sets to construct or update digital street-maps that represent the data set. Recently, the Computational Geometry community started to give attention to the street-map construction problems as well, applying different approaches and providing quality guarantees. Although different communities use different types or aspects of the GPS data, they face one challenge in common: how to model or incorporate the impreciseness of the input data in their output. In this paper we discuss specifically the impact of spatial inaccuracy of GPS trajectory data on *street-map reconstruction* algorithms. In particular, we discuss approaches and challenges to associate that impreciseness with the reconstructed street-intersections.

Categories and Subject Descriptors
F.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms
Algorithms

Keywords
Uncertainty, Geometrical Problems and Computations.

*This work has been partially supported by the National Science Foundation grants CCF-0643597 and CCF-1216602.



(a) (b)

Figure 1: Multi-vertex intersections extracted from the original street-map of Berlin, Germany

1. INTRODUCTION
The problem of reconstructing street-maps from GPS trajectory data is defined as: Given an input set I of GPS trajectories in the plane and a precision parameter $\epsilon > 0$, the goal is to compute an undirected *reconstructed* graph that represents all curves in the set. Here, ϵ is a parameter that defines the quality of the data. Intelligent Transportation Systems and GIS communities mainly use heuristic algorithms [3, 10, 9, 5] while the Computational Geometry community applies different approaches with quality guarantees [6, 1, 8, 2] to solve this problem. Most of these approaches are based on decomposing each street of a street-map into *good portions* that are well-separated and other portions which are close to intersections or other streets. Each point on a good portion of a street is sufficiently far away from any point on any other street or intersection. Reconstructing good portions of a street is relatively easy using clustering approaches, where the main idea is to group sub-trajectories that sample the same street together based on ϵ and find a single representative curve for that group/cluster. But when it comes to areas around street-intersections this approach fails. As the streets incident to an intersection come close to each other before sharing the intersection-vertex, the clustering algorithm has a wide range of spatial uncertainty on where the vertex is actually located and it will likely detect the vertex where the trajectories become sufficiently close but not where the intersection is actually located. And when an intersection has a more complicated combinatorial structure, consisting of multiple vertices connected with very short streets, such algorithms fail to detect that as well. Hence, it is not possible to detect the *combinatorial structure* and the *spatial position* of a street-intersection with exact accuracy, given noisy input trajectories.

Fathi et al. [7] propose an algorithm that constructs a street-map from GPS data by initially finding street intersections. They find the intersections using a classifier learned from shape descriptors, then they connect the intersections with

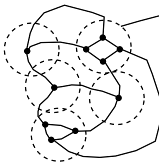


Figure 2: Original Street-map Model

roads and then optimize the estimated road network to best fit the connected roads. Aanjaneya et al. [1] are the first to provide guarantees on street-map reconstruction, although their focus is different. They view street networks as metric graphs and prove that their reconstructed structure is homeomorphic to the original street network. Their main focus is on computing an almost isometric space with lower complexity, therefore they focus on computing the combinatorial structure but they do not compute an explicit embedding of the edges or vertices. In our most recent work [2] we bound a region around an original vertex and prove that reconstructed vertices will be created in that region. We also proved that there is an one-to-one mapping between each such region and a vertex. However, the difficulty with our approach is that it generates multiple vertices for a single original vertex. In both [1] and [2], the homeomorphism claim is true only if each street-intersection consists of exactly one vertex and if there is a specified minimum distance (related to ϵ) between any two vertices. The algorithm of Fathi et al. [7] also assumes that vertices are well-separated. However, in practice street-intersections are not always well-separated and do contain clusters of vertices (See Figure 1).

In this paper, we discuss various aspects and challenges of the probabilistic street-intersection reconstruction problem. The aim of this problem is to identify possible combinatorial structures and spatial positions of vertices in a street-intersection, and associate a probability measure with each outcome. A natural extension to the street-map reconstruction problem can be achieved by combining the results of this paper with our previous work [2] to compute a class of possible graphs representing the input trajectory set I, each one having an associated probability measure.

2. PROBLEM FORMULATION

We model the *original* graph (street-map) $G_o = (V_o, E_o)$ as an embedded undirected graph in $\mathbb{R}^2$. We assume that V_o is a set of connected graphs (*vertex components*) and each edge is represented as a polygonal curve. Each vertex component $v \in V_o$ represents one street-intersection which could have one or more vertices each one having degree > 2 and connections between them (see Figure 2). Our Assumption 1b implies how such vertex components are defined. We refer to each edge of G_o as a *street*.

Each trajectory in the input curve set I is assumed to have sampled a connected sequence of edges in G_o (*street-path*). We model the error associated with each trajectory by a precision parameter ϵ. Given an input set I of polygonal curves in the plane, a precision parameter $\epsilon > 0$ and a vertex clustering parameter d, our goal is to compute probable structures and positions of street-intersections and assign a probability to each outcome.

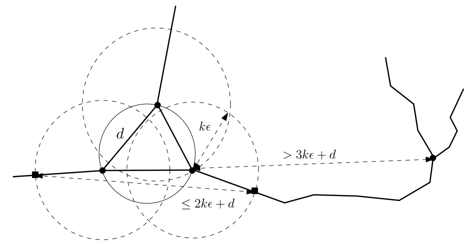


Figure 3: Illustration of Assumption 1 and 2

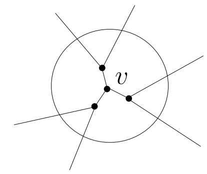


Figure 4: Intersection with an internal vertex

All of our assumptions in [2] will stay the same, except the one about street-intersections (Assumption 1b). In the previous work we assumed that each vertex is sufficiently far apart from other vertices, but since in practice vertices actually appear in clusters (see Figure 1) we redefine this assumption as follows:

1. If for two streets γ_1, γ_2 there are points $p_1 \in \gamma_1$ and $p_2 \in \gamma_2$ with distance $\leq 3\epsilon$, then either one of the following is true: *a*) γ_1 and γ_2 share a vertex v, and the sub-curves $\gamma_1[p_1, v]$ and $\gamma_2[p_2, v]$ have Fréchet distance $\leq 3\epsilon$ and they are fully contained in $B(v_0, 3\epsilon/\sin\alpha)$. Here, $\alpha = \angle p_1 v p_2$ and $B(v, \epsilon)$ is the ϵ-ball centered at v. *b*) γ_1 and γ_2 are connected to γ_s, and no point on γ_s is $(3\epsilon) - good$. γ_1 shares a vertex v_1 with γ_s, γ_2 shares a vertex v_2 with γ_s, and the sub-curves $\gamma_1[p_1, v_1]$ and $\gamma_2[p_2, v_2]$ have Fréchet distance $\leq 3\epsilon$ and they are fully contained in $B(v_1, 3\epsilon/\sin\alpha_1)$ and $B(v_2, 3\epsilon/\sin\alpha_2)$, respectively. Here, $\alpha_1 = \angle p_1 v_1 p_2$ and $\alpha_2 = \angle p_1 v_2 p_2$.

2. The vertex clustering parameter d defines the maximum distance between any two vertices of the same cluster. And the minimum distance between any two vertices of different clusters is $3k\epsilon + d$ (see Figure 3). Here, $k = 3/\sin\alpha$.

3. Vertices that are incident to at least one street $e \in E_o$ are called *terminal vertices*. We assume that the number of edges in a shortest path between any vertex and a terminal vertex is bounded by an integer c. For example, $c = 1$ in Figure 4 for vertex v.

The first assumption ensures that vertices are either within distance d from all vertices in an intersection or they have distance greater than $3k\epsilon + d$ from all the vertices that belong to different intersections. This assumption ensures unambiguous clustering. We use the first two assumptions to compute the vertex cluster of candidate vertices. The third assumption bounds the maximum possible number of vertices in an intersection.

3. MODELING UNCERTAINTY IN RECONSTRUCTED INTERSECTIONS

There are two main types of inaccuracy in time-stamped positional data [11]: *a*) *measurement error*, caused by GPS device error or signal propagation delay, and *b*) *sampling error*, reflecting how well the finite sequence of position samples represents the actual infinite trajectory of points. Although trajectories are often modeled as polygonal curves through the sequence of position samples, there are many other possible original curves that could have generated the same sequence of position samples. [1]

There are two approaches to model such spatial uncertainty in trajectories [12, 11]: 1. *Pdf-based models*, such as associating a two-dimensional probability density function with each time-stamped position. 2. *Shape-based models* bound the possible locations by geometric shapes.

Although, there are different models to handle uncertainty associated with the input, it is not yet clear how to use them to generate probable outputs and how to associate a probability measure to each one of them. So, the main problem is to model the uncertainty of the output. In our case, we observe two different kinds of uncertainty that could arise due to having uncertain input data:

1. *Spatial uncertainty* defines the probability of a specific position for an intersection-vertex. For example, in a $4-way$ intersection with one vertex, the reconstructed vertex could be in different positions with non-zero probabilities.
2. *Structural uncertainty* defines the probability of how the streets around a vertex cluster are connected to each other. For example, in a multi-vertex $4-way$ vertex-cluster, there are many different ways they could be connected (see Figure 6). A vertex cluster with just one vertex does not have structural uncertainty.

In this paper we discuss challenges and approaches of probabilistic street-intersection reconstruction for *single-vertex* intersections and *multi-vertex* intersections separately. When there is just one vertex, we will exploit the location of intersections of trajectories as witnesses to assign probabilities to different vertex positions.

Single-Vertex Intersections

The motivation behind this approach is to reduce the number of vertices generated by our previously proposed algorithm in [2]. In this section, we discuss our approach to identify probable positions of a single vertex based on the location of intersections of input trajectories. We compute a combinatorial structure C which, under reasonable conditions, has a one-to-one mapping between the intersections of its edges and the intersections of the corresponding trajectories. We also provide a guarantee for the distance between the original intersection and our reconstructed vertex.

[1]Note that we will neither model missing data in the form of failed location attempts because of adverse environmental conditions that make the device hard to connect to satellites, nor error caused by change of orientation of the device which is a common type of error in animal movement data [4] .

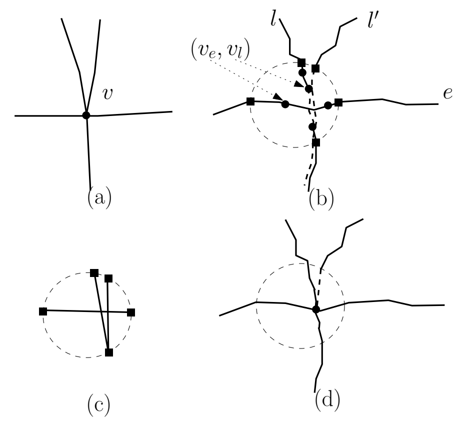


Figure 5: (a) Original street-map (b) Candidate vertices of intersection v (c) Combinatorial Representation C for v (d) Dashed portion is not part of the input trajectory

Our Approach

Our aim is to use the location of intersections between input trajectories as witnesses for positioning a vertex. Our approach consists of the following steps:

1. **Clustering Candidate Vertices**
   In this step we form clusters of candidate vertices which belong to the same intersection. A pair of *candidate vertices* (v_e, v_l) consists of the first or last two points on two different trajectories e and l, where $dist(v_e, v_l)$ is less than or equal to some specified distance, in our setting it is 1.5ϵ (see Figure 5(b)). Such pairs can be computed as terminal points of the partition in $M_{1.5\epsilon}(e, l)$ using our previously published algorithm in [2].
   After comparing v_l with endpoints of e, we determine if v_l and v_e belong to an existing cluster or to a new cluster. This is justified by Assumption 1b. Clustering of candidate vertices will be done within distance $2k\epsilon + d$. With each such cluster we identify and save the entry and exit candidate vertices for each trajectory that pass through the intersection. In this fashion we are storing traffic flow information.
2. **Creating Combinatorial Representation C**
   We compute a combinatorial representation for each cluster in the following steps: (a) Compute a disk D centered at a candidate vertex that contains all vertices of the cluster. (b) Compute a circular ordering S of intersections of D with all edges around the cluster (see Figure 5(b), the squares indicate such intersections). (c) Add all of the intersection points as vertices in C. And also add edges between two vertices if there is a trajectory that enters and exits the cluster through them (see Figure 5(c)). Exploiting the assumptions stated in Section 2, it can be proven that two edges e_1 and e_2 between non-consecutive vertices in S intersect in C iff their corresponding trajectories intersect.
3. **Probable Positions**
   Any intersection between edges of C indicates that corresponding trajectories have at least one intersection. We pick one pair of edges from C that intersect and

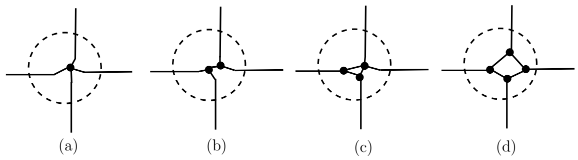


Figure 6: 4 way Intersection

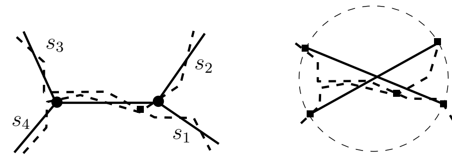


Figure 7: Intersections are not good witnesses.

compute and define one of the intersections of the corresponding trajectories as a probable position of the vertex. Using a similar explanation as in [2], it can be proven that the distance between the original and such probable vertex is $\leq \epsilon/2 \sin \alpha/2$.

Challenges

Unfortunately, this approach will not work when two or more one-way streets merge into a single one. Additional unsolved issues include: *a*) How to assign a probability measure to such probable positions, and *b*) how to assign a probability to the parts of the reconstructed streets computed by our algorithm which are not parts of the input trajectory (see Figure 5(d)).

Multi-Vertex Intersections

We consider two types of graphs in a vertex component: *a*) a DAG, similar to Figure 6(b), or *b*) a cyclic graph, similar to the intersections in Figure 6(c) and (d). We compute the cluster of candidate vertices for intersections using the same approach but, for multi-vertex intersections the trajectory intersections are not always good witnesses for existence or probable positions of a vertex. For example, in Figure 7 although the intersection has two vertices the trajectories intersect just once. How to compute probable vertex positions is an unsolved challenge for multi-vertex intersections.

Modeling Structural Uncertainty

Here, we propose a model for the structural uncertainty of a multi-vertex intersection. We assume we have the following input:

a) Let $S = s_1 s_2 s_3 \ldots s_n$ be the circular ordering of the streets around the intersection.

b) Each street has an associated probability that describes the probability of sharing a vertex or edges with incident edges: $Pr(s_i s_{i+1})$ denotes the probability of s_i and s_{i+1} sharing a vertex, and $Pr(\overline{s_i s_{i+1}})$ denotes the probability of s_i and s_{i+1} are being connected by an edge which does not have a good section (or, the probability of s_i and s_{i+1} not sharing a vertex). For example, s_1 and s_2 share a vertex while s_2 and s_3 are connected by an edge in Figure 7. Based on our assumptions, $Pr(s_i s_{i+1}) + Pr(\overline{s_i s_{i+1}}) = 1$.

And using our model we can answer the following queries: *a*) Find the most likely or least likely intersection structure for a given input. *b*) The probability of a given combinatorial structure of an intersection for a given input.

Challenges

There are many issues which need to be solved for multi-vertex intersections: *a*) How to compute the most likely spatial positions of vertices. *b*) How to assign probabilities to each street which will indicate if it shares a vertex or edges with incident streets. *c*) How to model the probability associated with streets that connect vertices of an intersection. Also, the existence of non-terminal vertices in intersections adds another whole new set of challenges.

4. CONCLUSION AND FUTURE WORK

This paper is an effort to identify different issues and research opportunities on the probabilistic modeling of the street-intersection reconstruction problem. And even though identifying *good-sections* is easy, it is still not clear how to combine the uncertainties of sub-trajectories that belong to one cluster to compute a representative curve with the lowest uncertainty. In our future work we hope to address the challenges presented here.

5. REFERENCES

[1] M. Aanjaneya, F. Chazal, D. Chen, M. Glisse, L. J. Guibas, and D. Morozov. Metric graph recons. from noisy data. In *Proc. ACM SoCG*, pages 37–46, 2011.

[2] M. Ahmed and C. Wenk. Constructing street networks from GPS trajectories. In *Proc ESA*, 2012.

[3] R. Bruntrup, S. Edelkamp, S. Jabbar, and B. Scholz. Incremental map generation with GPS traces. In *Proc. IEEE Intelligent Transp. Sys.*, pages 574 – 579, 2005.

[4] F. Cagnacci. *Animal ecology meets GPS-based radiotelemetry : a perfect storm of opportunities and challenges*. London : Royal Society, 2010.

[5] L. Cao and J. Krumm. From GPS traces to a routable road map. In *Proc. of the 17th ACM SIGSPATIAL Int. Conf. on Advances in GIS*, pages 3–12, 2009.

[6] D. Chen, L. Guibas, J. Hershberger, and J. Sun. Road network reconstruction for organizing paths. In *Proc. ACM-SIAM Symp. on Discrete Algorithms*, pages 1309–1320, 2010.

[7] A. Fathi and J. Krumm. Detecting road intersections from gps traces. In *Proceedings of the 6th inter. conf. on GIScience*, pages 56–69. Springer-Verlag, 2010.

[8] X. Ge, I. Safa, M. Belkin, and Y. Wang. Data skeletonization via Reeb graphs. In *25th Annual Conf. on Neural Info. Proc. Sys.*, pages 837–845, 2011.

[9] T. Guo, K. Iwamura, and M. Koga. Towards high accuracy road maps generation from massive GPS traces data. In *IEEE Int. Geoscience and Remote Sensing Symposium*, pages 667 –670, 2007.

[10] Z. Li, J.-G. Lee, X. Li, and J. Han. Incremental clustering for trajec. In *DASFAA*, pages 32–46, 2010.

[11] D. Pfoser and C. S. Jensen. Capturing the uncertainty of moving-object representations. In *Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pages 111–132, 1999.

[12] G. Trajcevski. Uncertainty in spatial trajectories. In *Computing with Spatial Trajec.*, pages 63–107. 2011.