Constructing Street Networks from GPS Trajectories[1]

Mahmuda Ahmed[1] Carola Wenk[2]

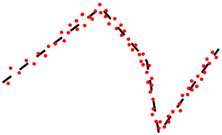
[1]The University of Texas at San Antonio, USA
[2]Tulane University, USA

European Symposium on Algorithms, 2012

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

Introduction

- **Shape reconstruction problem:** The problem is to reconstruct shapes from unorganized sample point-sets.



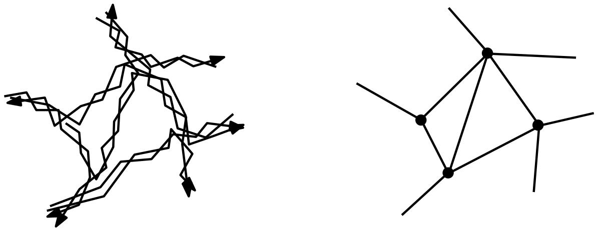
- We have ordered sequences of points, *trajectories*.
    - Each point is a time-stamped position sample.

Outline

Outline

Problem Statement

Given a set of *trajectories* in the plane, compute a *street map/graph* that *represents* all trajectories in the set.



Related Work

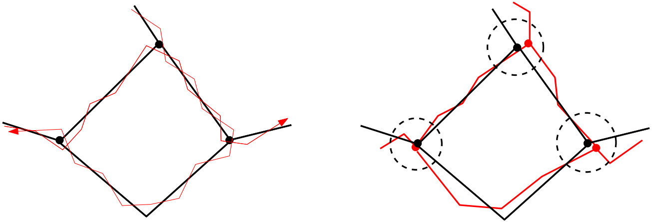
- Chen et al. [5]
  - $b\epsilon$-net
  - Small Hausdorff distance between original and reconstructed edges.
- Aanjaneya et al. [1]
  - *Rips-Vietoris graph*
  - Their reconstructed structure is homeomorphic to the original network.
- Ge et al. [6]
  - Model the graph as *Reebgraph*
  - Their reconstructed loops in skeleton graph has one-to-one mapping with original one.

They used unorganized point-sets and assumed the graph is very densely sampled.

Outline

Data Model

- We model the *original* graph and a *reconstructed* graph as embedded undirected graphs in $\mathbb{R}^2$.
- We model error associated with each trajectory by a precision parameter ϵ.



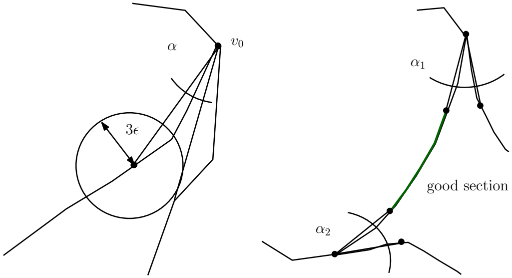
Assumptions

We have two sets of assumptions based on a precision parameter, ϵ.

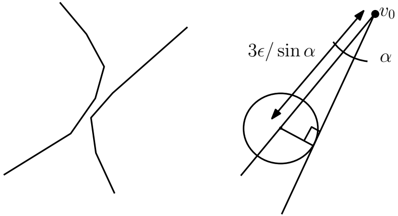
- Assumptions on original street-network, G_o.
- Assumptions on input data set, I.

Assumptions on Original Street-Network, G_o

Assumption 1: Each street will have a well-separated portion.
α**-Good[5]:** A point p on G is α-*good* if $B(p, \alpha\varepsilon) \cap G$ is a 1-ball that intersects the boundary of $B(p, \alpha\varepsilon)$ in two points. A point p is α-*bad* if it is not α-good. A curve β is α-good if all points on β are α-good.



Assumption 2: When two streets come closer than a specified distance, they must share a vertex within bounded distance.



Assumption 1: Each input curve is within Fréchet distance $\varepsilon/2$ of a street-path in original street-network, G_o.

Fréchet distance: $\delta_F(f,g) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \max_{t\in[0,1]} \|f(\alpha(t)) - g(\beta(t))\|$ where α, β range over continuous and non-decreasing reparametrizations.

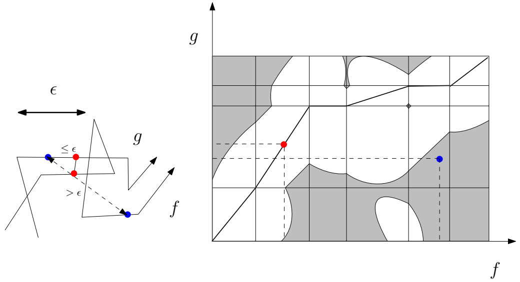


Figure : FD_ϵ for two polygonal curves f, g.

Decision variant of the problem is solved by finding a monotone path in the free space [3], from lower left to upper right corner.

Assumption 2: All input curves sample an acyclic path in G_o.

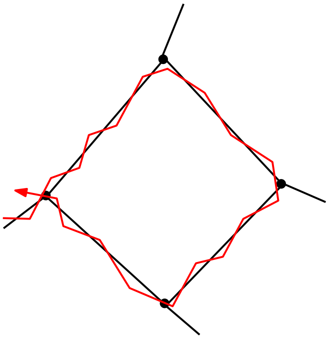
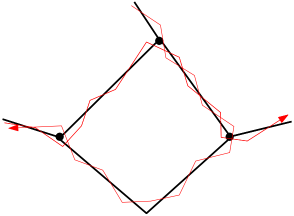


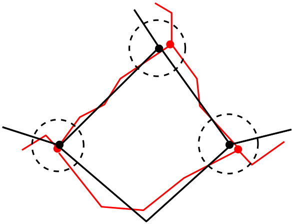
Figure : This is not allowed.

Outline

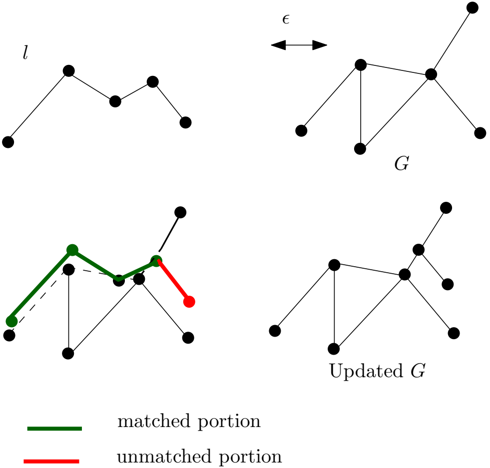
Given a set of input curves, I that samples an embedded graph (*original street-network*, $G_o = (V_o, E_o)$) and a precision parameter, ϵ we compute a *reconstructed graph*, $G = (V, E)$ that *represents* I.

- Good-sections are within ϵ Fréchet distance.
- Vertices would be created within bounded regions around original vertex.





Overview of Our Algorithm



Incremental Map Construction Algorithm

Steps

- **Step One:** Compute matched and unmatched portions of an input curve.
  - We combine the idea of *Partial Curve Matching* and *Map Matching* problem.
- **Step Two:** Add unmatched portion as an edge in the graph.
- **Step Three:** Compress the complexity of matched portion.
  - We compute a minimum-link representative curve.

Partial Curve Matching by Buchin et al. [4]:

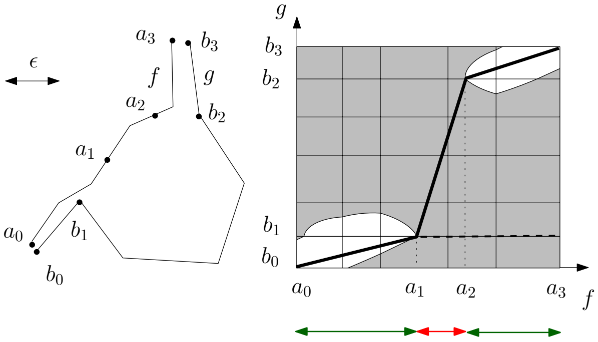


Figure : Partially similar f and g.

Find a monotone shortest path on weighted free-space diagram (black region weight 1, white region weight 0) from lower left to upper right end-point.

Map Matching Problem (uses weak Fréchet distance) by Alt et al. [2]
Given a graph, G, a curve, l and an ϵ, find a path, π in G such that, $\delta_F(l, \pi) \leq \epsilon$

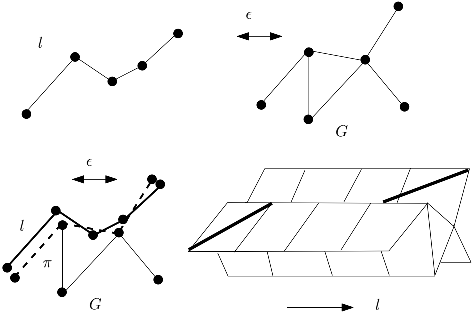
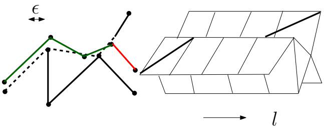


Figure : FS_ϵ for a graph G and a curve l. An example path π is shown in dashed line and an l-monotone path in FS_ε is highlighted in bold.

Find an l – *monotone* path from any left to any right end-point in the *free-space*.

Step One: Compute Matched and Unmatched Portions (contd.)

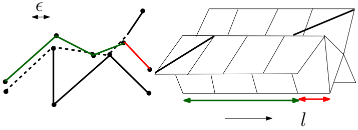
Combining the idea of Map Matching and Partial Curve Matching, Map Construction problem can be redefined as finding an $l-\textit{monotone}$ shortest path on free-space surface from any left end point to any right end point.



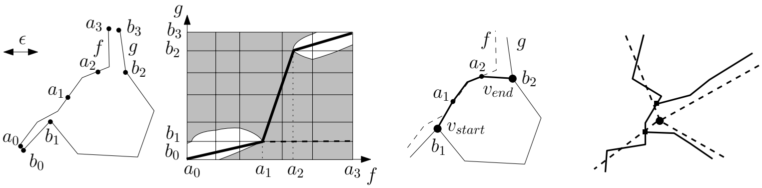
Finding such path is very hard as FS_ϵ is a *non-manifold* surface.

Our Observation

- We are interested to maximize the matched portion on curve.
  - That allows the desired path to go through the black region along the graph without adding additional cost.
  - And reduces the problem to computing total length of unmatched portions along the curve.
- Such matching can be computed by projecting *free-space* on to the curve as intervals.
  - Each white interval corresponds to a matched portion.
  - Each black interval corresponds to an unmatched portion.



Step Two: Create/Split Edges and Create Vertices

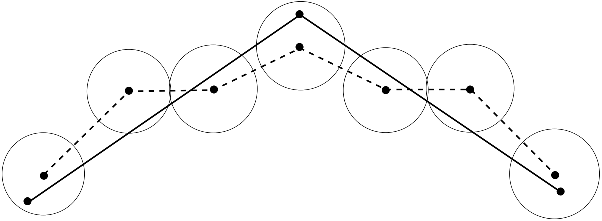


Multiple vertices would be created for an original vertex with degree ≥ 4.

Step Three: Compress the Complexity of Matched Portion

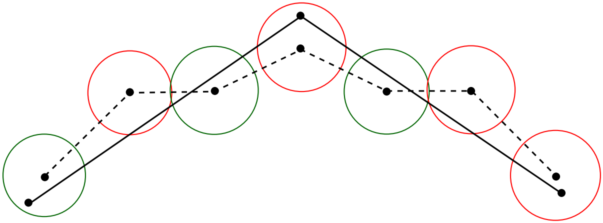
Minimum-Link Chain Stabbing Algorithm by Guibas et al. [7]

- Given an ordered sequence of ϵ-discs find the minimum link curve that stabs the sequence in order.
- If the vertices of min-link curve stays ϵ-fattening of the curve then they have Fréchet distance less than or equal to ϵ.



Step Three: Compress the Complexity of Matched Portion (contd.)

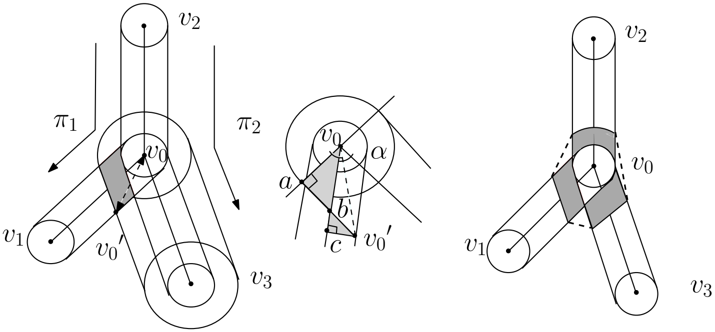
- We compute a sequence, combining vertices of the matched edge in the graph and corresponding matched portions of the curve.
- And apply min-link algorithm on that sequence.
- We prove that, our min-link reconstructed edge has complexity less than or equal to the original street.



Outline

Quality Guarantee

- We prove that, good-sections of our reconstructed edge is within ϵ Fréchet distance from original street and has complexity less than or equal to the original.
- We bound a region around original vertex, where the reconstructed vertices would be created.



Outline

Experiments

- We applied our algorithm on real tracking data obtained by sampling vehicle movements at a rate of 30 seconds.
- The dataset consists of $3,237$ vehicle trajectories consisting of a total of $57,109$ position samples.

Experimental Results

Figure : Portion of reconstructed graph of Berlin

Outline

Summary and Future Work

Summary:

- We presented an incremental algorithm which is very practical to use in real life.
- It assures reconstruction of good-sections and bounds the vertex region.

Future Work:

- Reduce the number of vertices.
- Probabilistic reconstruction of street-intersection.

Questions

Questions???

Thanks

Thank you...

Runtime

- **Step One:**Compute matched and unmatched portions of an input curve.
  - $O(n_i N_{i-1})$ where N_{i-1} is the complexity of G_{i-1} and n_i is the complexity of $i - th$ curve in I.
- **Step Two:**Add unmatched portion as an edge in the graph.
  - Constant time.
- **Step Three:**Compress the complexity of matched portion.
  - $O(n^2 \log n)$, where n is the number of vertices in the sequence.

Experimental results for synthetic data:

GPS error	s.r. in sec	ε m	# of curves	# of points	G edges	G g.s.(m)	G vertices	G_o edges	G_o g.s.(m)	G_o vertices
$\pm$ 5m	1	12	900	14515	805	105,062	56	560	76,248	37
	5	17	651	5934	731	66,755	46	560	67,959	32
	10	20	583	4318	681	56,379	32	560	65,270	28
	20	25	556	3323	733	62,522	28	560	58,810	26
	30	30	543	2793	696	65,784	22	560	53,200	20
$\pm$ 10m	1	22	900	14515	912	76,991	47	560	63,128	28
	5	27	651	5934	501	46,031	31	560	56,251	24
	10	30	583	4318	499	31,816	31	560	53,200	20
	20	33	556	3323	605	37,709	20	560	49,643	17
	30	35	543	2793	635	50,112	16	560	48,183	14

References

Mridul Aanjaneya, Frederic Chazal, Daniel Chen, Marc Glisse, Leonidas J. Guibas, and Dmitriy Morozov.
Metric graph reconstruction from noisy data.
In *Proc. ACM Symp.Computational Geometry*, pages 37–46, 2011.

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

Helmut Alt and Michael Godau.
Computing the Fréchet distance between two polygonal curves.
Int. J. of Computational Geometry and Applications, 5:75–91, 1995.

Kevin Buchin, Maike Buchin, and Yusu Wang.
Exact algorithm for partial curve matching via the Fréchet distance.
In *Proc. ACM-SIAM Symp. on Discrete Algo. (SODA09)*, pages 645–654, 2009.

Daniel Chen, Leonidas Guibas, John Hershberger, and Jian Sun.
Road network reconstruction for organizing paths.
In *Proc. ACM-SIAM Symp. on Discrete Algorithms*, 2010.

Xiaoyin Ge, Issam Safa, Mikhail Belkin, and Yusu Wang.
Data skeletonization via Reeb graphs.
In *25th Annual Conference on Neural Info. Processing Systems*, pages 837–845, 2011.

L J Guibas, J E Hershberger, J S B Mitchell, and J S Snoeyink.
Approximating polygons and subdivisions with minimum-link paths.