Road Map Construction
And Comparison

Carola Wenk
Tulane University

In collaboration with:
Mahmuda Ahmed,
Brittany Fasy,
Kyle Hickman,
Sophia Karagiorgou,
Dieter Pfoser

Partially funded by: NSF CCF-1301911
GPS Trajectory Data
GPS Trajectory Data & Roadmap

⇒ Map Construction
Which is the Better Roadmap?
Which is the Better Roadmap?

⇒ Map Comparison
Map Construction
Map Construction

• Given a set of trajectories, compute the underlying road network

• Capturing constrained movement (explicit or implicit streets/routes, animal behavior)

• [mapconstruction.org], [openstreetmap.org]

• Related problem: Map updates
Map Construction

Geometric reconstruction problem:

• Given a set of movement-constrained trajectories, extract the underlying geometric graph structure

• Reconstruct a geometric domain that has been sampled with continuous curves that are subject to noise

⇒ Sampling with organized data (trajectories) instead of point clouds

⇒ Need to identify combinatorial information (edges, vertices), as well as geometric representation/embedding

⇒ Clustering & how to represent an edge/street
Input and Output Models

- **Trajectory**: A sequence of position samples $p_1, \ldots, p_n$. Each $p_i$ minimally consists of a position measurement (e.g., $(x,y)$-coordinate) and a time stamp, but may also include other information such as instantaneous speed.

- **Road network**: An embedded graph $G=(V,E)$.
Uncertainty and Error/Noise

- **Measurement error:** Usually modeled as Gaussian noise, or as an error-disk around each measurement point.

- **Sampling error:**
  - Amounts to modeling the transition between two measurements
  - Simple transition model: Linear interpolation.
  - Common transition models in ecology: Brownian bridges, Levy walks
  - Simple region-based model: Buffers of fixed radius around each trajectory

⇒ Need **input model:**
  - E.g., chain of beads model for trajectories

⇒ What is a good **output model**?

Map Construction: Some Results

- [DBH06]: Classical Kernel Density Estimation based method

![Density of tracks](image1)
![Contour](image2)
![Center lines](image3)

- [BE12]: Kernel Density Estimation based method; pipeline to first create scaffold then map-match trajectories.

Map Construction: More Results


- **[ACCGGM11]**: Reconstruct metric graph from point cloud. Compute almost isometric space with lower complexity. Focus on combinatorial info and not embedding. Quality guarantees assume dense sampling.

- **[GSBW11]**: Topological approach on neighborhood complex. Uses Reeb graph to model skeleton graph (branching structure)

---

Map Construction: Even More Results

- [FK10]: First identify intersections (vertices) using a shape descriptor, then fill in edges.
- [KP12]: Detect intersections from turns and speed change, then fill in edges.
- [AW12]: Use trajectory information. Incrementally add one trajectory after another. Use partial Fréchet distance to identify new and existing portions. Use min-link algorithm to compute representative curve/edge.

Map Construction [AW12]

Incrementally add one trajectory after another. For each trajectory:

1. Use partial Fréchet distance to identify new and existing portions by combining mapmatching with partial Fréchet distance:

2. Use min-link curve simplification algorithm to reconcile existing portions

Fréchet Distance for Curves

\[ \delta_F(f,g) = \inf_{\alpha,\beta:[0,1] \to [0,1]} \max_{t \in [0,1]} ||f(\alpha(t))-g(\beta(t))|| \]

where \( \alpha \) and \( \beta \) range over continuous monotone increasing reparameterizations only.

- Man and dog walk on one curve each
- They hold each other at a leash
- They are only allowed to go forward
- \( \delta_F \) is the minimal possible leash length

Let $\varepsilon > 0$ fixed (eventually solve decision problem)

$F_\varepsilon(f,g) = \{ (s,t) \in [0,1]^2 \mid \| f(s) - g(t) \| \leq \varepsilon \}$ white points free space of $f$ and $g$
Monotone path encodes reparametrizations of $f$ and $g$

$\delta_F(f,g) \leq \varepsilon$ iff there is a monotone path in the free space from $(0,0)$ to $(1,1)$

Such a path can be computed using DP in $O(mn)$ time
Partial Fréchet Distance

- For a given $\varepsilon>0$, compute a monotone path in the free space diagram
  - that is allowed to pass through both white and black regions and
  - that maximizes the portion of the path within the white regions.
- Apply DP approach as before, but on each cell boundary maintain a function (instead of an interval). This function measures the maximum length of any monotone path from the lower left corner to the point on the boundary.
- For technical reasons the $L_1$-distance is used to measure the Fréchet distance (hence the free space is polygonal)
- Runtime $O(n^3 \log n)$
- This partial distance identifies portions of the two curves that correspond to each other

[BBW09] K. Buchin, M. Buchin, Y. Wang, Exact Partial Curve Matching under the Fréchet Distance, SODA: 645-654, 2009
Map Matching

**Given:** A graph $G$, a curve $l$, and a distance parameter $\varepsilon$.

**Task:** Find a path $\pi$ in $G$ such that $\delta_F(l,\pi) \leq \varepsilon$

**Application:** GPS routing; use GPS data from vehicle fleets to build data base of current travel times
Map Matching

**Given:** A graph $G$, a curve $l$, and a distance parameter $\varepsilon$.

**Task:** Find a path $\pi$ in $G$ such that $\delta_F(l, \pi) \leq \varepsilon$

Compute free space surface.
And find path $\pi'$ in it.
Map Construction [AW12]

We model the original map and the reconstructed map as embedded undirected graphs in the plane.

We model error associated with each trajectory by a precision parameter $\varepsilon$.

1. We assume each input curve is within Frechet distance $\varepsilon/2$ of a street-path in the original map.

2. (We assume all input curves sample acyclic paths.)

3. Two additional assumptions on original map help us to provide quality guarantees.

Map Construction [AW12]

Incrementally add one trajectory after another.
For each trajectory:

1. Use partial Fréchet distance to identify new and existing portions by combining mapmatching with partial Fréchet distance:

2. Use min-link curve simplification algorithm to reconcile existing portions

Map Construction [AW12]

Incrementally add one trajectory after another. For each trajectory:

1. Use partial Fréchet distance to identify new and existing portions by combining mapmatching with partial Fréchet distance:
   - Compute free space surface
   - Find path that maximizes matched portion on the curve.

   \[ \Rightarrow \text{Project free space onto curve:} \]
   \[ \text{white interval} = \text{matched portion, black interval} = \text{unmatched portion} \]

2. Use min-link curve simplification algorithm to reconcile existing portions

Assumptions

Assumptions on unknown graph:

1. Road fragments are "good".
   "good": Every small circle intersects in just two points

2. Close fragments must have an intersection point

\[ p_1 \leq 3\epsilon \leq 3\epsilon \leq 3\epsilon \]

\[ \gamma_1 \quad \gamma_2 \]

\[ p_1 \quad p_2 \]

\[ \Rightarrow \text{Projection approach is justified, because free space has special structure. Trajectory can only sample one good section in original network.} \]

Give quality guarantees

- **Good regions:** We prove the quality guarantee that there is a 1-to-1 correspondence with bounded description complexity between well-separable good portions of original network and reconstructed graph.

- **Bad regions:** We give the first description and analysis of vertex regions.

⇒ It is relatively easy to handle well-sampled clean data. Deal with noisy data that is not well-sampled and give quality guarantees.

Map Construction [AW12]

Incrementally add one trajectory after another. For each trajectory:

1. Use partial Fréchet distance to identify new and existing portions by combining mapmatching with partial Fréchet distance:
   - Compute free space surface
   - Find path that maximizes matched portion on the curve.

   \[ \Rightarrow \text{Project free space onto curve:} \]
   \[ \text{white interval} = \text{matched portion, black interval} = \text{unmatched portion} \]

2. Use min-link curve simplification algorithm to reconcile existing portions

Different Reconstructions

GPS Trajectory Data
Reconstructed Maps
Different Reconstructions
Map Comparison
Compare Constructed Maps

• How can one measure the quality of constructed maps?
• Surprisingly, there is no applicable benchmark map:
  – Professional maps
  – Do not cover the same area and the same details as a given input set of trajectories

• Given two geometric (planar...) graphs embedded in the same plane. How similar are they?
• What if one of the graphs is reconstructed?
Compare Constructed Maps

- How can one measure the quality of constructed maps?
- Surprisingly, there is no applicable benchmark map:
  - Professional maps
  - Do not cover the same area and the same details as a given input set of trajectories

1. Given two geometric (planar...) graphs embedded in the same plane. How similar are they?
2. What if one of the graphs is reconstructed?
Graph Comparison

• Subgraph-isomorphism:
  – Enforces 1-to-1 correspondence, and works on abstract graphs without embedding

• Graph edit distances:
  – Either hard, or only for special graph classes (trees!)
  – Does not incorporate common embedding

⇒ Map comparison is different:
  – We have a common embedding
  – We need to incorporate partial matching
  – 1-to-many assignments may be allowed
  – Graphs are planar
  – Connectivity should be similar
Distance Measures for Map Comparison

• [BE12], [AKPW14]: Overview / benchmark papers

• [BE12b]: “Haensel and Gretel distance”: Graph sampling-based distance measure in local neighborhood. Maximize number of breadcrumbs that match 1-to-1.

• [KP12]: Compare shortest paths in both maps, with nearby start and end positions. Ensures similar connectivity/routing properties.

• [AFHW15]: Considers maps as sets of paths, and compares path sets.

• [AFW14]: Compares local topology of graphs using persistent homology


[BE12b] Haensel and Gretel Distance

- Sample each graph within a common neighborhood by traversing the graphs and dropping equally spaced “bread crumbs”.

- Find a maximum bipartite matching of the bread crumbs.

- Define $u_i = \# \text{unmatched bread crumbs of graph } G_i$, and $m_i = \# \text{matched bread crumbs of graph } G_i$.

- Compute
  
  $F$-score $= 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

  $\text{precision} = \frac{m_1}{u_1 + m_1}$

  $\text{recall} = \frac{m_2}{u_2 + m_2}$

[KP12] Shortest Path-Based Distance

• [KP12]: Compare shortest paths in both maps, with nearby start and end positions. Ensures similar connectivity/routing properties.

[AFHW15] Path-Based Distance

• Directed Hausdorff distance on path-sets:

\[ \overrightarrow{d}_{G,H}(\pi_G, \pi_H) = \max_{p_G \in \pi_G} \min_{p_H \in \pi_H} \delta_F(p_G, p_H) \]

• \( \pi_G \) path-set in G, and \( \pi_H \) path-set in H

• We prove that using the set of paths of link-length three approximates the overall distance, if vertices in G are well-separated and have degree \( \neq 3 \).

• Asymmetry of distance definition is desirable, if G is a reconstructed map and H a ground-truth map.

[AFHW15] Path-Based Distance

- Define local signature to visualize differences in the graphs:
  - As the path set in $G$ consider all paths of a fixed link-length that go through a fixed edge, and compute the path-based distance to $H$. => Heat map on $G$

![Heat map on $G$](link-length 1)

![Heat map on $G$](link-length 2)

Consider a common local neighborhood of both maps.

Consider the cycles of each graph inside this neighborhood.

Now thicken each graph and track changes in the cycle structure using persistent homology.

Use distance between persistence diagrams to compare changing local cycle structure.

Local “signature” that captures local topological similarity of graphs.

Dataset: Athens

- 129 GPS trajectories from school buses
- 2.6km x 6km


Dataset: Athens

• 129 GPS trajectories from school buses
• 2.6km x 6km


Dataset: Athens

• 129 GPS trajectories from school buses
• 2.6km x 6km


[AFW14] Local Homology Based Distance

• Compared two reconstructed maps: [BE12], [KP12]

• Local signature captures different topology (missing intersections) well.

Conclusion

• Map construction and map comparison are recent data-driven problems

• Related to geometric reconstruction, trajectory clustering, shape comparison

• There is a lot of potential for theoretical modeling and algorithms that provide quality guarantees

• Open problems / future work:
  – Map updates
  – More complicated/realistic input and noise models for trajectories
  – More complicated/realistic output models for the maps (vertex regions; directed graphs, with turn information, road categories, etc.)