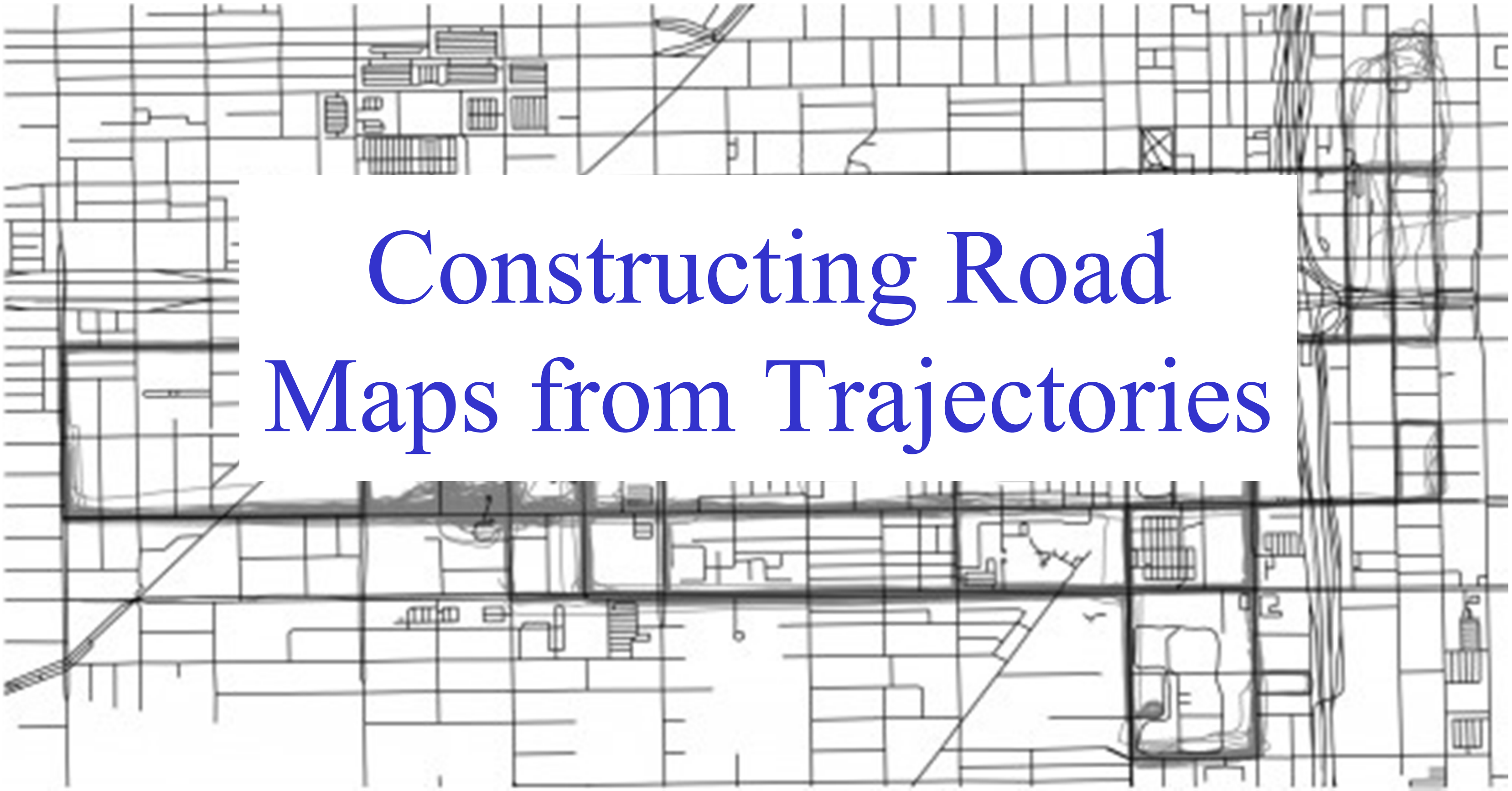
Constructing Road Maps from Trajectories

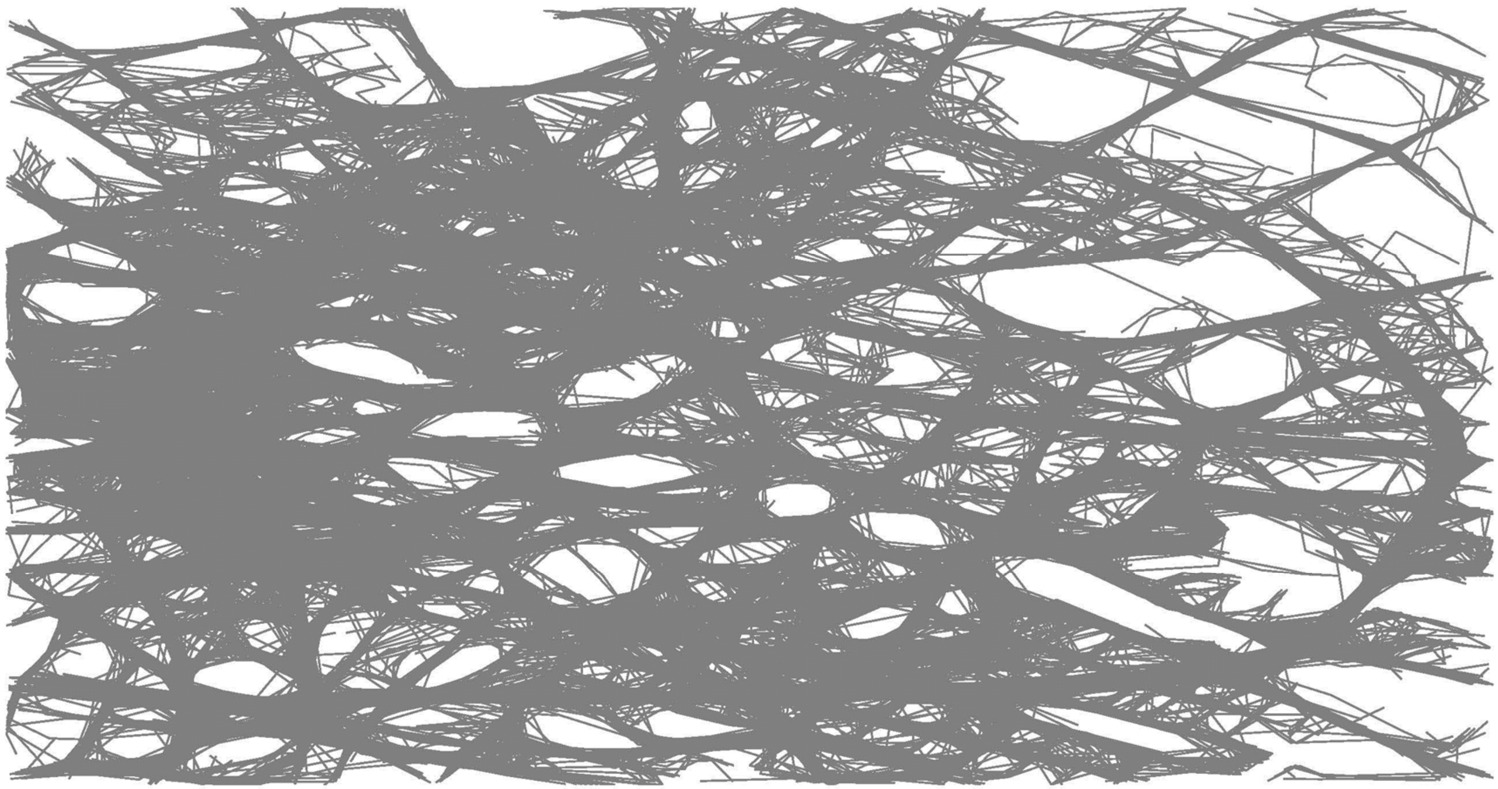


Carola Wenk

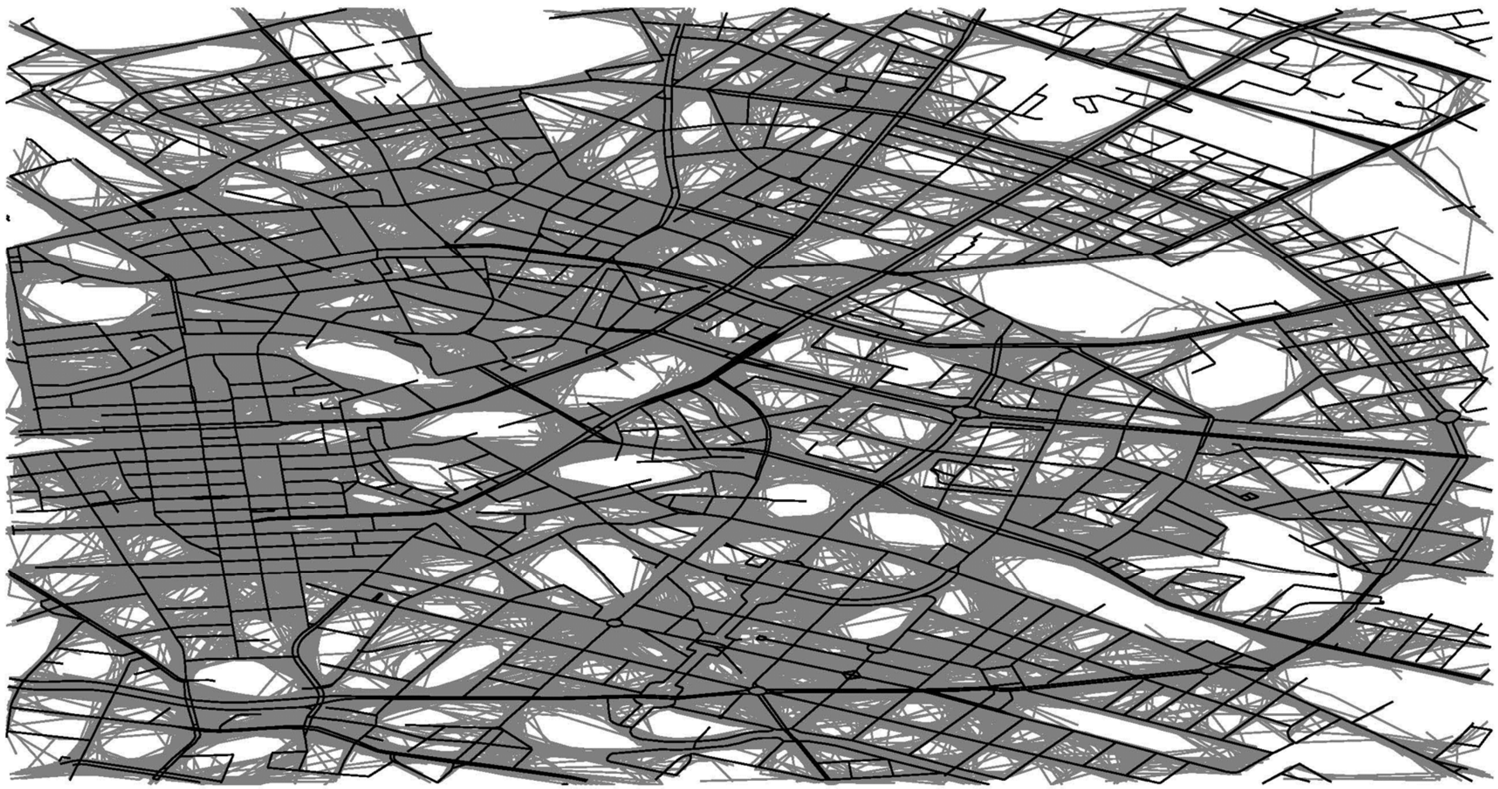
Department of Computer Science

Tulane University

GPS Trajectory Data



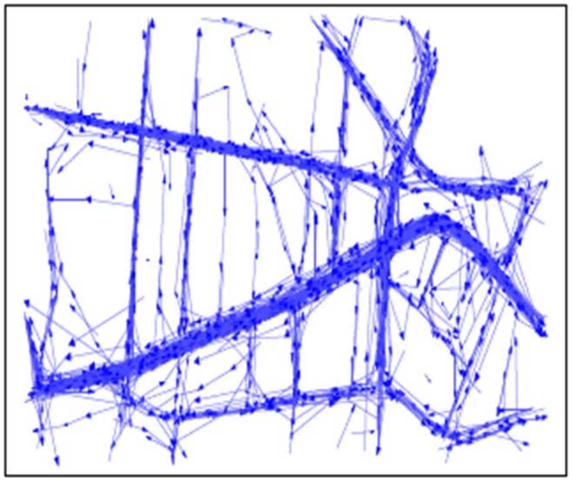
GPS Trajectory Data & Roadmap



⇒ Map Construction

Map Construction

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

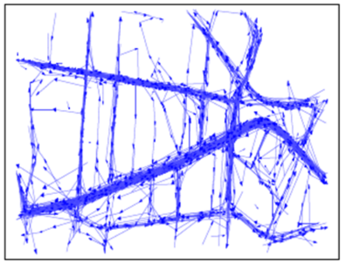


- Capturing constrained movement (explicit or implicit streets/routes, animal behavior)

- mapconstruction.org , openstreetmap.org

- Related problem: Map updates

Map Construction



Geometric reconstruction problem:

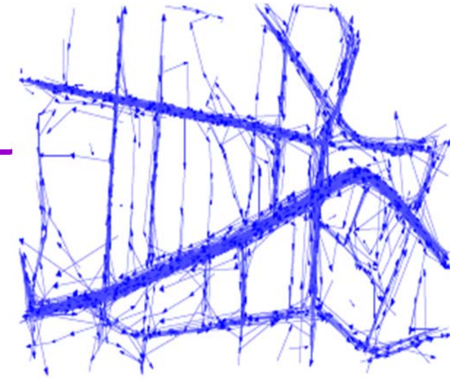
- Given a set of movement-constrained trajectories, extract the underlying geometric graph structure
- Reconstruct a geometric domain that has been sampled with continuous curves that are subject to noise

$\Rightarrow$ Sampling with organized data (trajectories) instead of point clouds

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

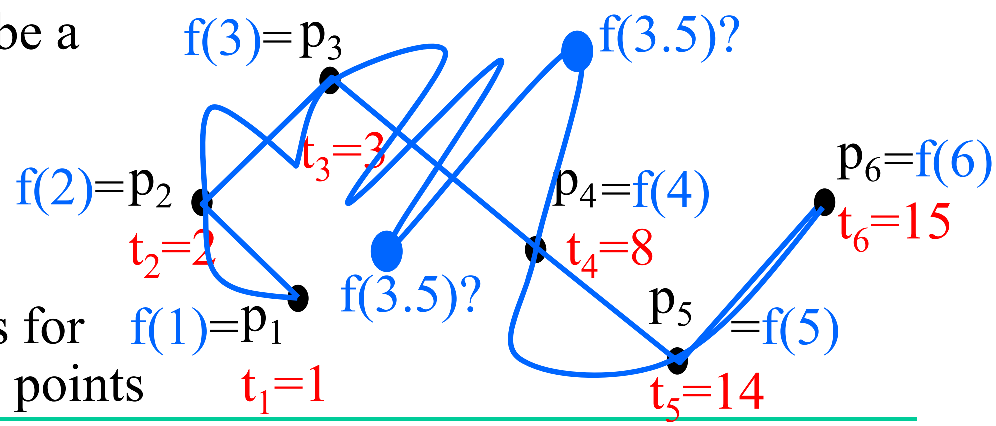
$\Rightarrow$ Clustering & how to represent an edge/street

Trajectories



- A **trajectory** is a sequence of position samples: $p_1,\ldots,p_n$
- Each p_i minimally consists of:
  - position measurement (e.g., (x,y)-coordinate)
  - time stamp
  - $\Rightarrow$ e.g., $p_i = (x_i, y_i, t_i)$

- Such a trajectory is a finite sample of a continuous curve $f:[t_1, t_n] \rightarrow R^2$
- For simplicity, f is often assumed to be a piecewise linear interpolation.
- But clearly there are many possible choices for f.
- There are also many possible choices for parameterizations in between sample points

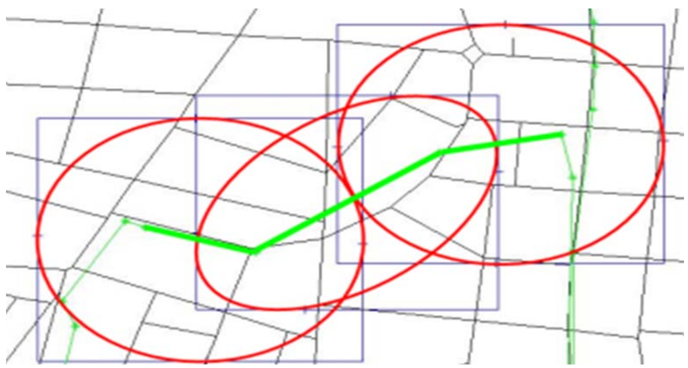


Uncertainty and Error/Noise

- **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**?

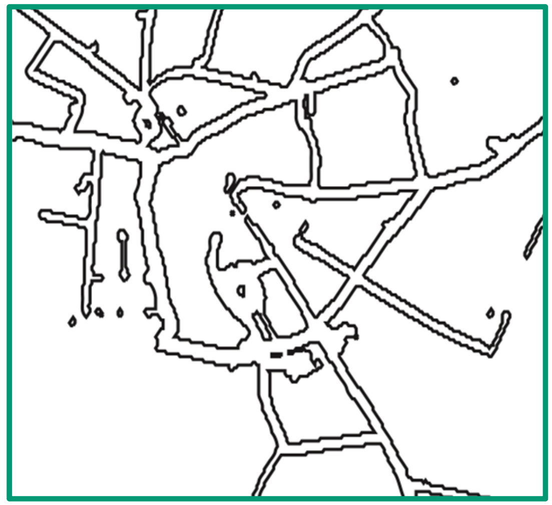


[T11] G. Trajcevski, Uncertainty in spatial trajectories, in Y. Zheng, X. Zhou (eds), Computing with Spatial Trajectories: 63-107, 2011.

Map Construction: Some Results

- [DBH06]: Classical Kernel Density Estimation based method

Density of tracks



Contour

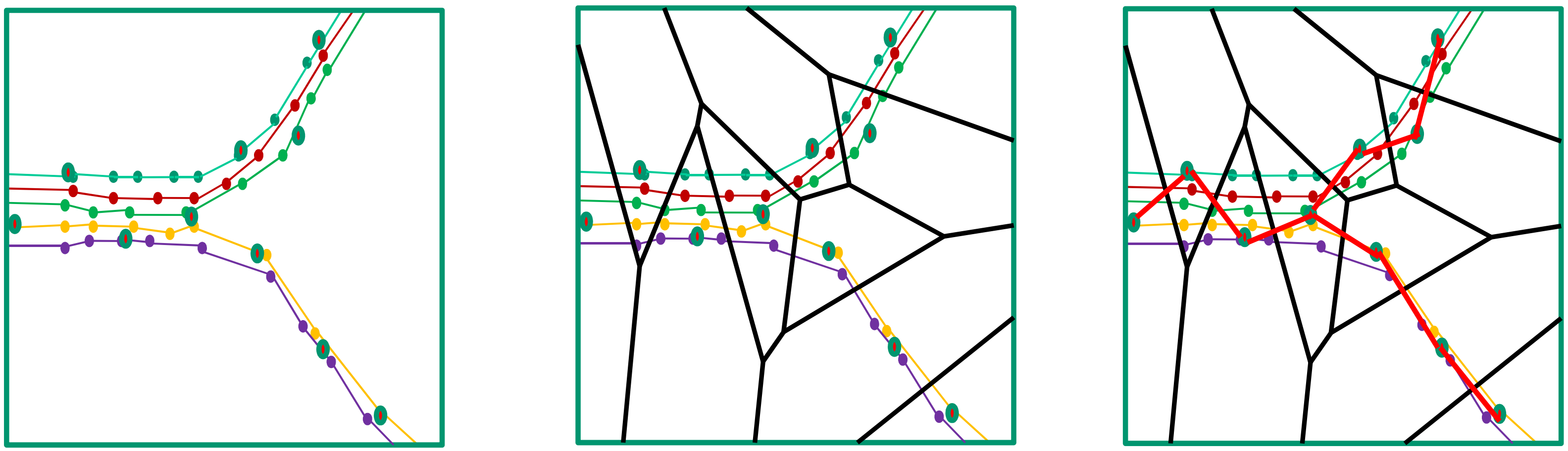
Center lines

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

[DBH06] J. Davies, A. Beresford, A. Hopper: Scalable, distributed, real-time map generation. IEEE Pervasive Comp. 5(4), 47-54, 2006.
[BE12] J. Biagioni, J. Eriksson, Map inference in the face of noise and disparity, 20th ACM SIGSPATIAL: 79-88, 2012

Map Construction: More Results

- [CGHS10]: First algorithm with quality guarantees. Subsamples trajectories, yielding a point cloud. Uses local neighborhood simplicial complexes. Reconstructs "good" portions of edges.



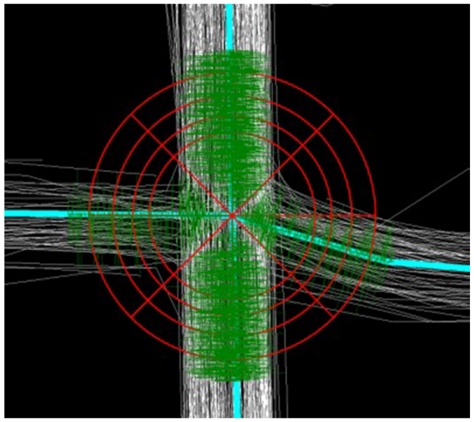
- [ACCGGM11]: Reconstruct metric graph from point cloud. Compute almost isometric space with lower complexity. Focus on combinatorial info and not embedding. Quality guarantees assume dense sampling.

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

[ACCGGM11] M. Aanjaneya, F. Chazal, D. Chen, M. Glisse, L. Guibas, D. Morozov. Metric graph reconstruction…, SoCG, 2011.
[CGHS10] D. Chen, L. Guibas, J. Hershberger, J. Sun, Road network reconstruction for organizing paths, SODA, 2010.
[GSBW11] X. Ge, I. Safa, M. Belkin, Y. Wang, Data skeletonization via Reeb graphs, Conf. Neural Inf. Proc. Systems: 837-845, 2011.

Map Construction: Even More Results

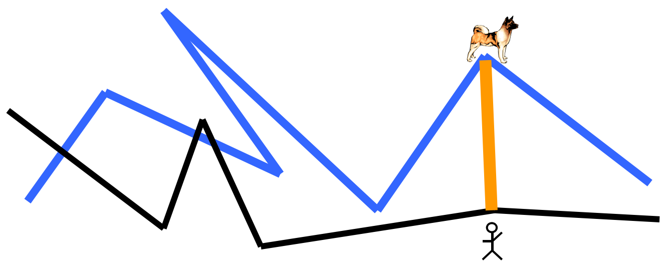
- [FK10]: First identify intersections (vertices) using a shape descriptor, then fill in edges.
- [KP12]: Detect intersections from turns and speed change, then fill in edges.



- [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.

[FK10] A. Fathi, J. Krumm, Detecting road intersections from GPS traces, Geographic Information Science, LNCS 6292: 56-69, 2010.
[KP12] S. Karagiorgou, D. Pfoser, On vehicle-tracking data-based road network generation, 20th ACM SIGPATIAL: 89-98, 2012.
[BE12] J. Biagioni, J. Eriksson, Map inference in the face of noise and disparity, 20th ACM SIGSPATIAL: 79-88, 2012
[AW12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

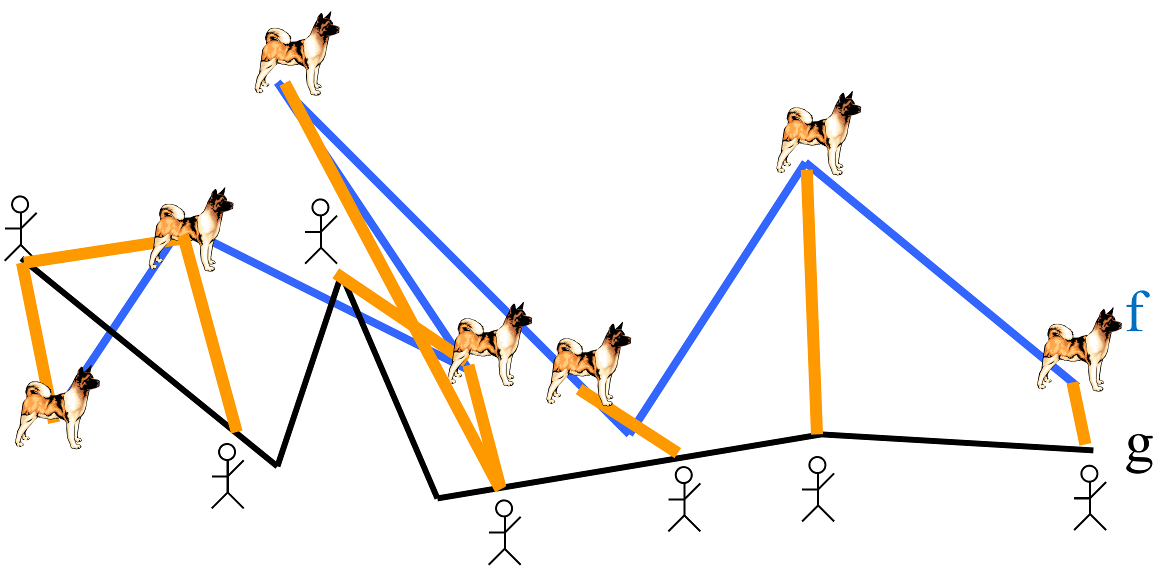
Fréchet Distance



Fréchet Distance for Curves

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

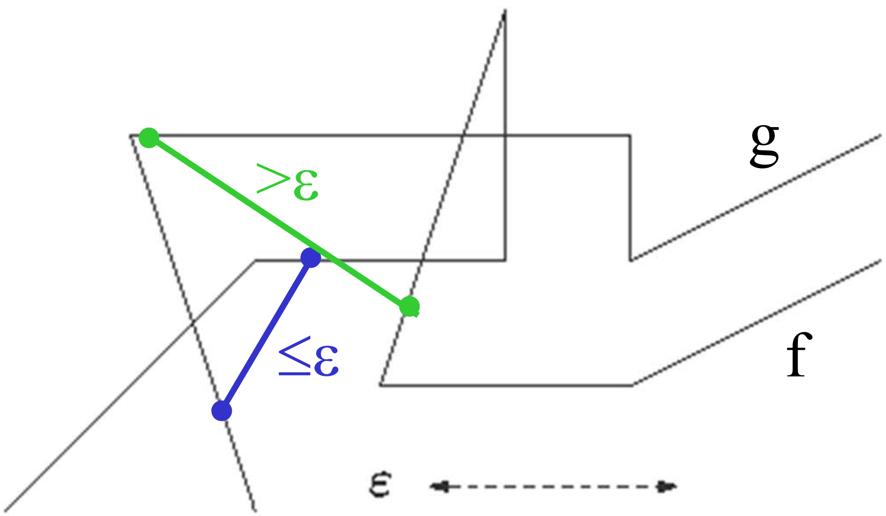
where α and β range over continuous monotone increasing reparameterizations only.

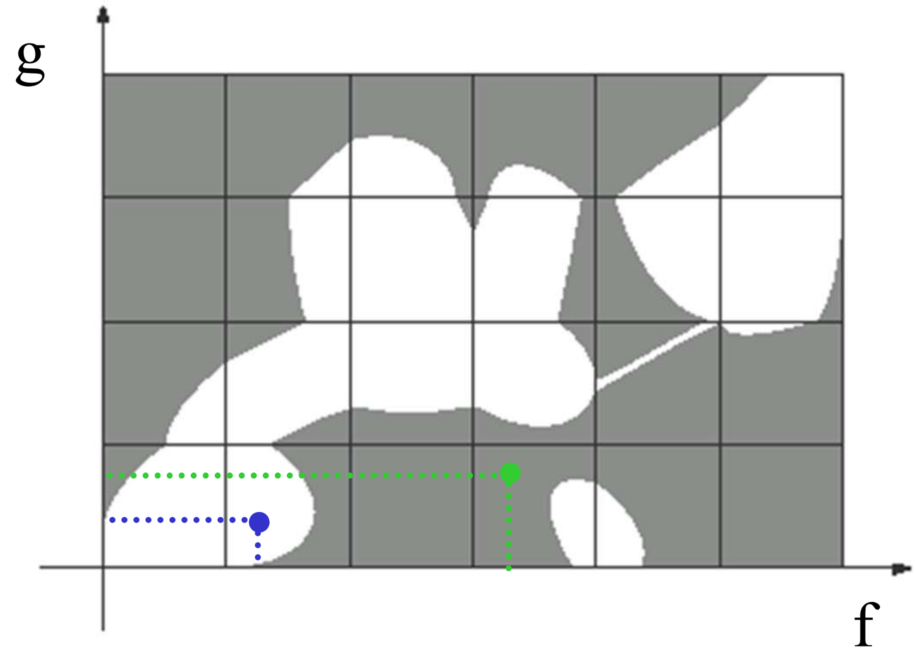


- Man and dog walk on one curve each
- They hold each other at a **leash**
- They are only allowed to go forward
- δ_F is the minimal possible leash length

[F06] M. Fréchet, Sur quelques points de calcul fonctionel, *Rendiconti del Circolo Mathematico di Palermo* 22: 1-74, 1906.

Free Space Diagram

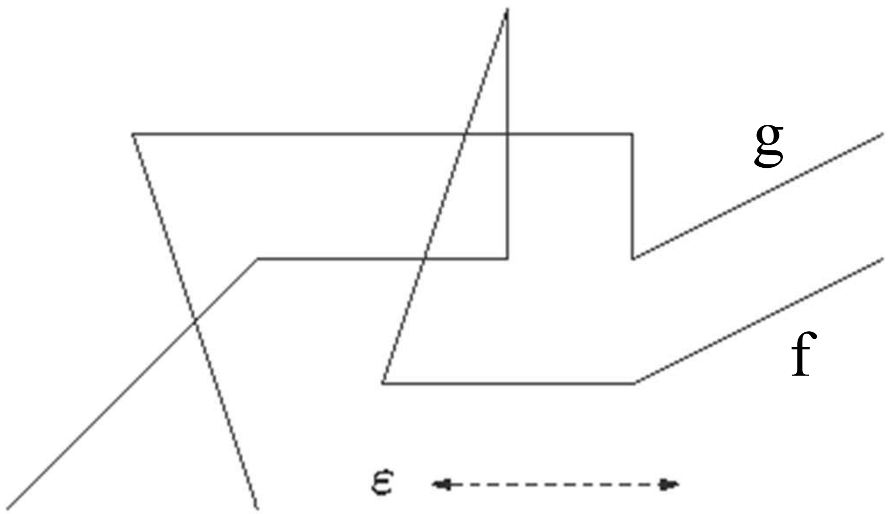


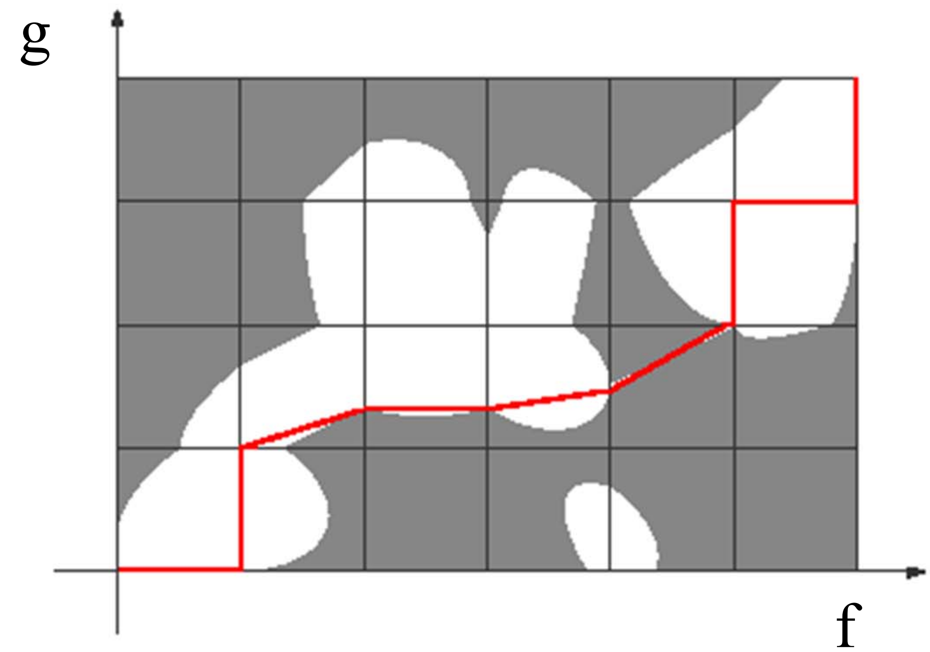


- Let ε > 0 fixed (eventually solve decision problem)
- $F_\varepsilon(f,g) = \{ (s,t) \in [0,1]^2 \mid \| f(s) - g(t) \| \leq \varepsilon \}$ *white points*

  free space of f and g

Free Space Diagram

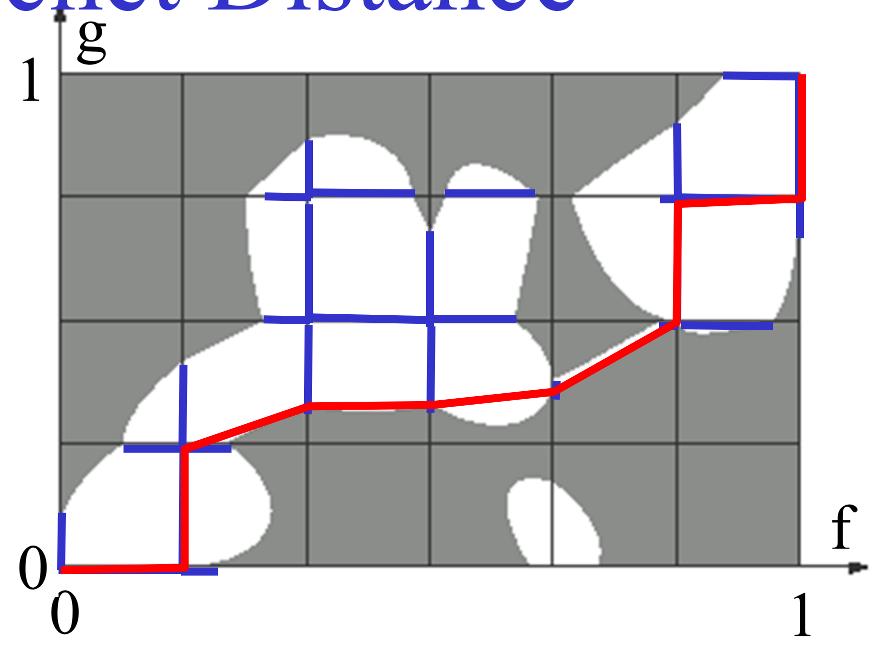




- Monotone path encodes reparametrizations of f and g
- $\delta_F(f,g) \leq \varepsilon$ iff there is a monotone path in the free space from (0,0) to (1,1)
- Such a path can be computed using DP in O(*mn*) time

Compute the Fréchet Distance

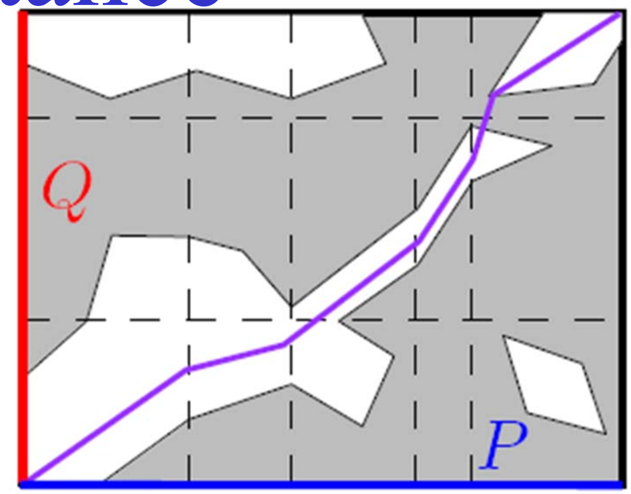
- **Solve the decision problem**
  $\delta_F(f,g) \le \varepsilon$ in O(mn) time:
  - Find monotone path using DP:
  - On each cell boundary compute the interval of all points that are reachable by a monotone path from (0,0)
  - Compute a monotone path by backtracking



- **Solve the optimization problem**
  - In practice in O(mn log b) time with binary search and b-bit precision
  - In O(mn log mn) time [AG95] using parametric search (using Cole‘s sorting trick)
  - In O(mn $\log^2$mn) expected time [CW09] with randomized red/blue intersections

[AG95] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, *IJCGA* 5: 75-91, 1995.
[C**W**10] A.F. Cook IV, **C. Wenk**, Geodesic Fréchet Distance Inside a Simple Polygon, ACM TALG 7(1), 19 pages, 2010.

Partial Fréchet Distance

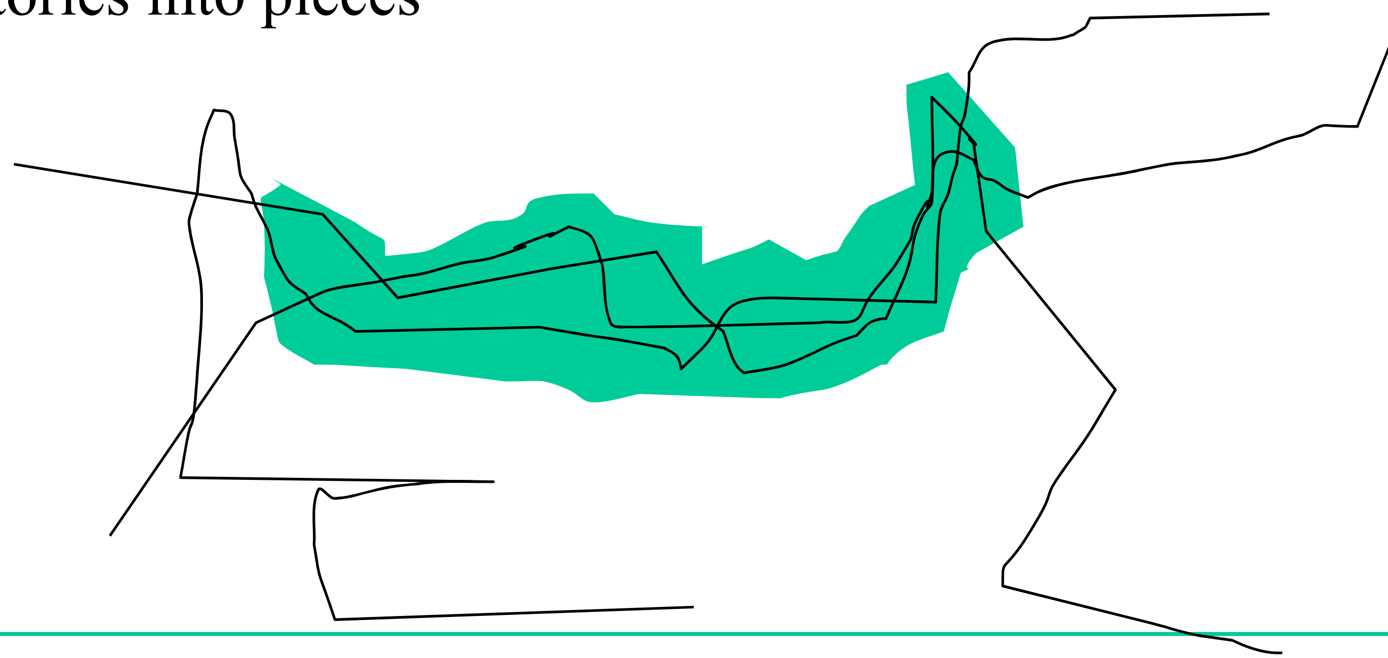


- For a given ε>0, compute a monotone path in the free space diagram
  - that is allowed to pass through both white and black regions and
  - that maximizes the portion of the path within the white regions.
- Apply DP approach as before, but on each cell boundary maintain a function (instead of an interval). 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 $\Rightarrow$ helpful for subtrajectory clustering

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

Sub-Trajectory Clustering

- Find similar portions in trajectories
- Lots of algorithms for finding clusters in point sets
- Harder for trajectories since you need to figure out where to break the trajectories into pieces



Sub-Trajectory Clustering

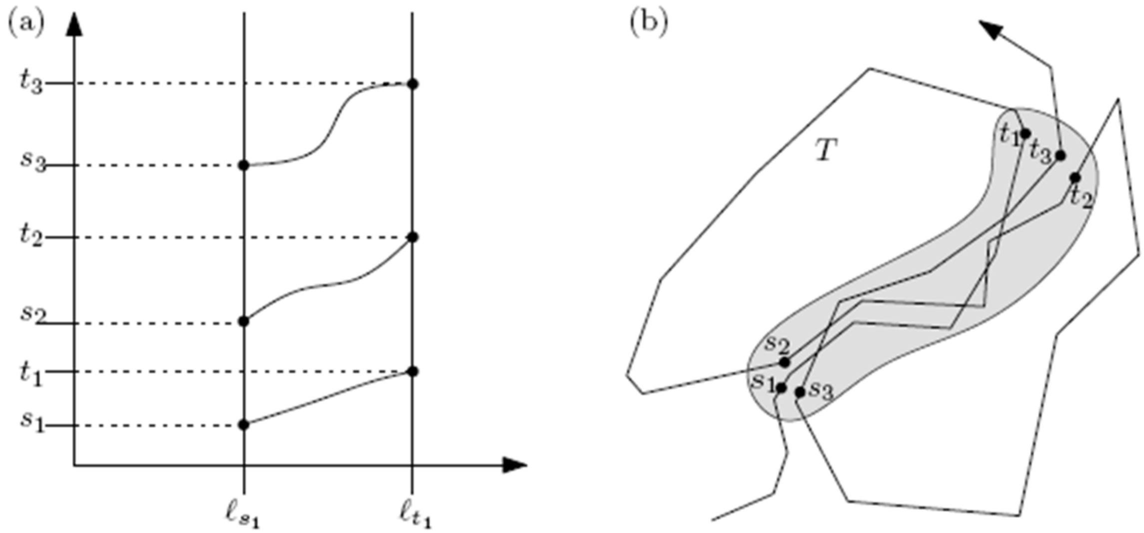
- A. Asahara, A. Sato, and K. Maruyama. Evaluation of trajectory clustering based on information criteria for human activity analysis. In 10th Int. Conf. on Mobile Data Management: Systems, Services and Middleware (MDM), pages 329-337, 2009.
- K. Buchin, M. Buchin, J. Gudmundsson, M. L öffler, and J. Luo. Detecting commuting patterns by clustering subtrajectories. International Journal of Computational Geometry and Applications, special issue on 19th International Symposium on Algorithms and Computation (ISAAC), 2010.
- K. Buchin, M. Buchin, M. van Kreveld, and J. Luo. Finding long and similar parts of trajectories. In 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), pages 296-305, 2010.
- A. Dahlbom and L. Niklasson. Trajectory clustering for coastal surveillance. In 10th Int. Conf. on Information Fusion, pages 1-8, 2007.
- J. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: A partition-and-group framework. In Proc. ACM SIGMOD International Conference on Management of Data, pages 593-604, 2007.

Sub-Trajectory Clustering

- X. Li, W. Hu, and W. Hu. A coarse-to-ne strategy for vehicle motion trajectory clustering. In 18th Int. Conf. on Pattern Recognition (ICPR), volume 1, pages 591-594, 2006.
- Z. Li. Incremental clustering for trajectories. Master's thesis, University of Illinois at Urbana-Champaign, 2010.
- Z. Li, J.-G. Lee, X. Li, and J. Han. Incremental clustering for trajectories. In Proc. 15th Int. Conf. Database Systems for Advanced Applications (DASFAA), pages 32-46, 2010.
- T.W. Liao. Clustering of time series data - a survey. Pattern Recognition, 38:1857-1874, 2005.
- Y. Zhang and D. Pi. A trajectory clustering algorithm based on symmetric neighborhood. In WRI World Congrees on Computer Science and Information Engineering, volume 3, pages 640-645, 2009.

Fréchet-Based Clustering

- Given an input set of trajectories, append them all to form a single trajectory f
- Compute the free space diagram of f with itself (comparing f with f)



- Find clusters of monotone curve pieces in the (white) free space

⇒ Sweep free space from left to right, maintain data structure

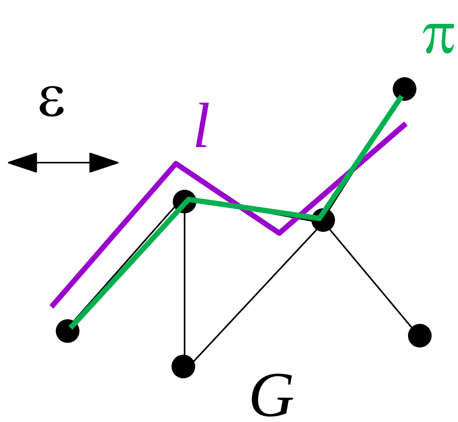
[BBGLL10] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler, J. Luo. Detecting commuting patterns…, IJCGA, 2010.

Map-Matching

Map Matching

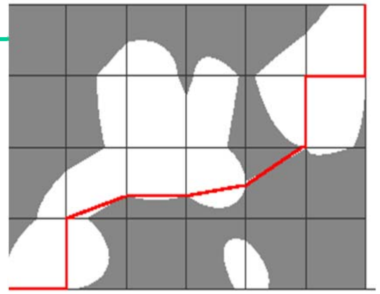
Given: A graph G, a curve l, and a distance parameter ε.

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



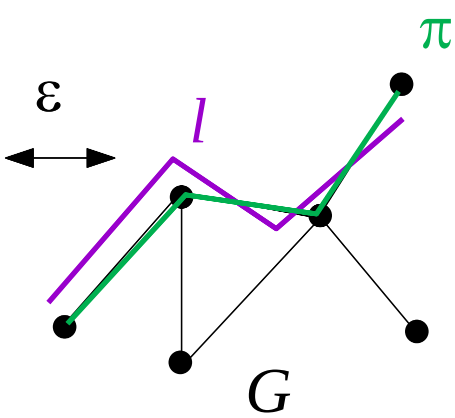
Application: GPS routing; use GPS data from vehicle fleets to build data base of current travel times

Subtask: Map Matching



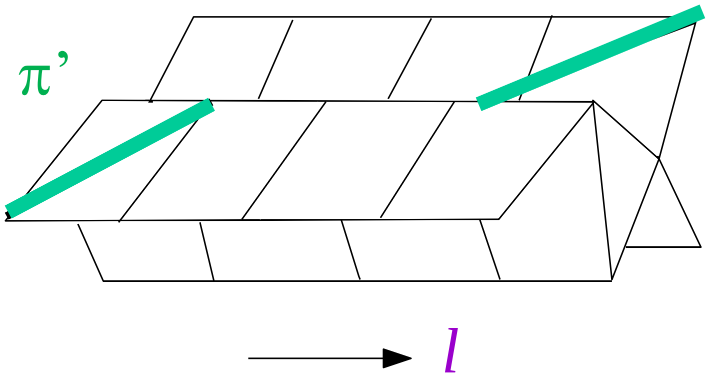
Given: A graph G, a curve l, and a distance parameter ε.

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

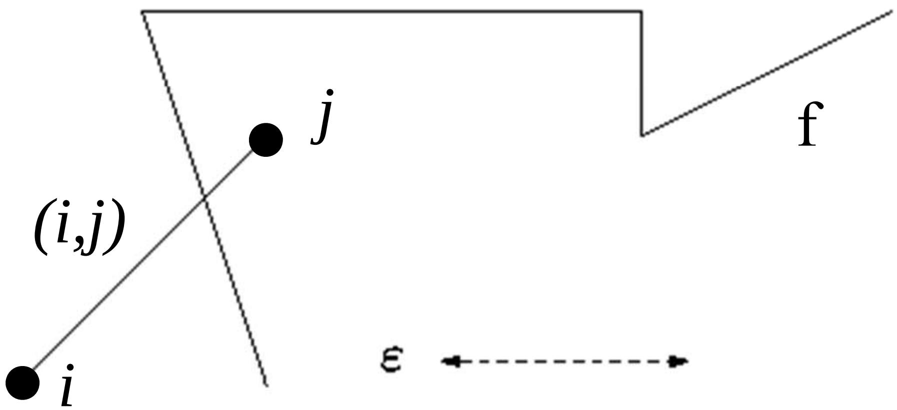


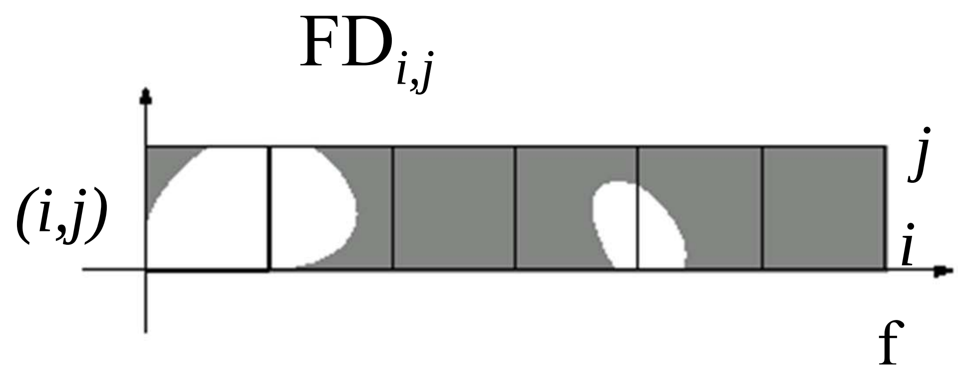
Compute free space surface.

And find path π' in it



Map Matching: Free Space Diagram

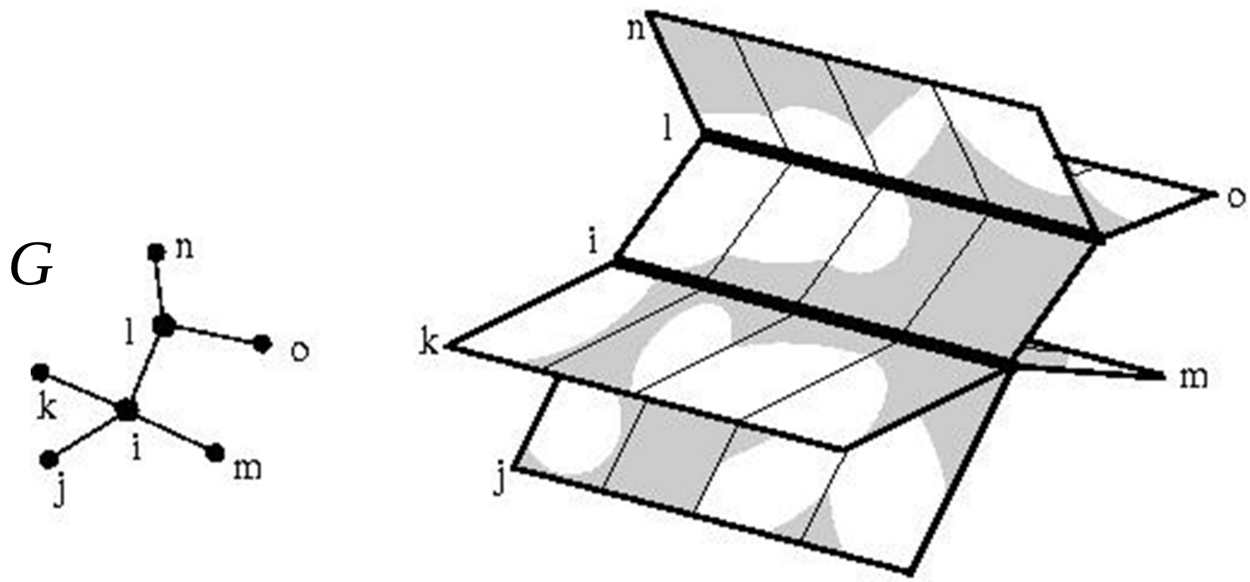




- For every edge (i,j) in G: $FD_{i,j} = FD(f, (i,j))$
- For every vertex i in G: sei $FD_i = FD(f, i)$ *1-dimensional*

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Free Space Surface

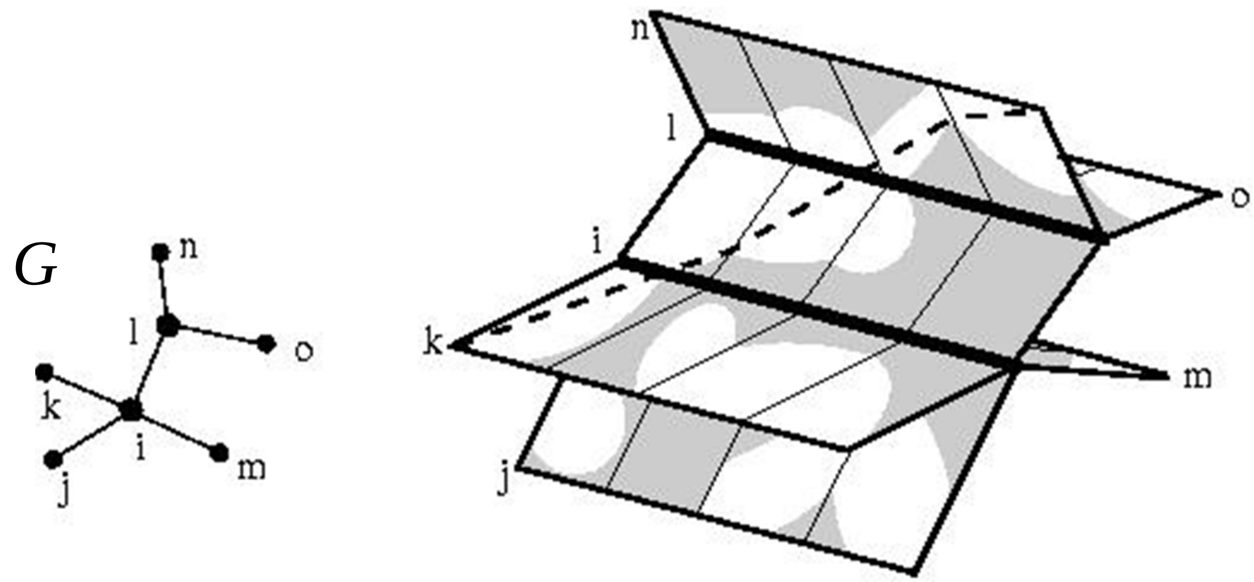


- Glue the free space diagrams $FD_{i,j}$ together according to adjacency information in *G*

→ **Free space surface of f and *G***

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

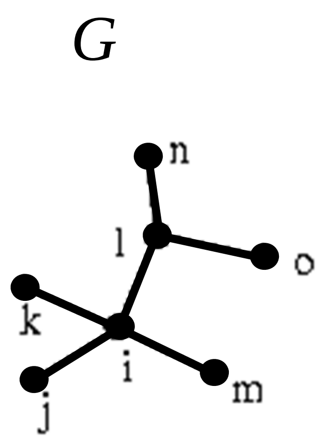
Free Space Surface

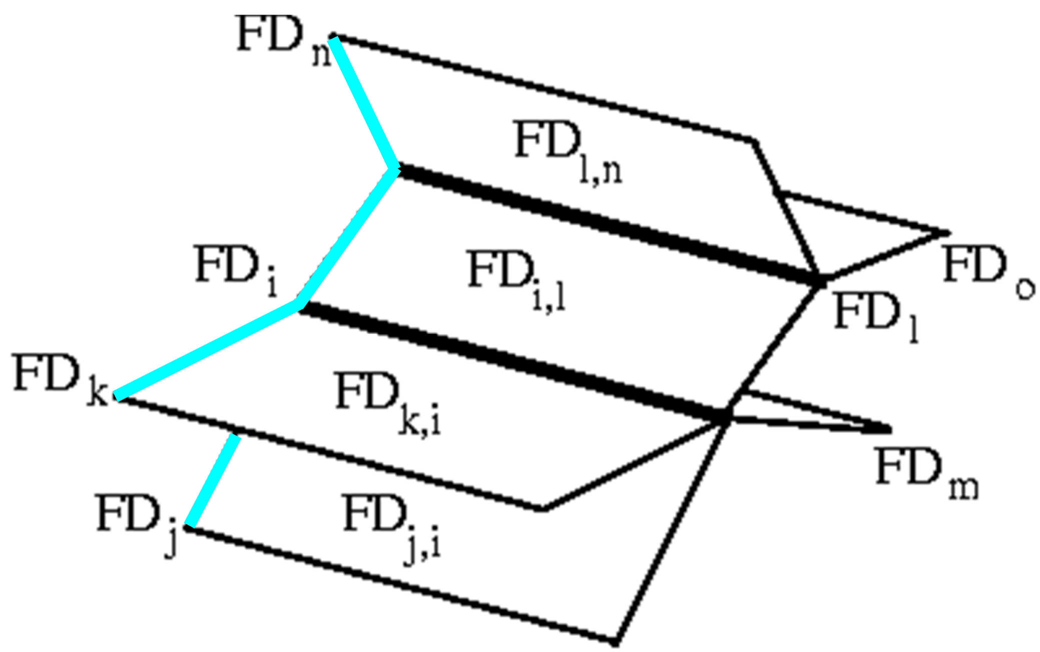


- **Task:** Find a **monotone path** in the free space surface that
  - starts in a lower left corner
  - and ends in an upper right corner

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep

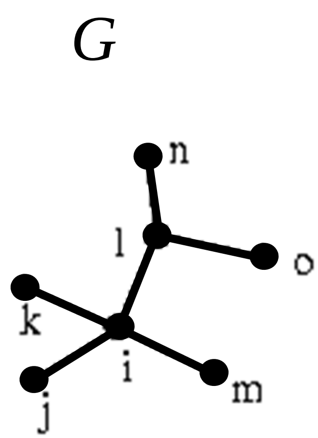


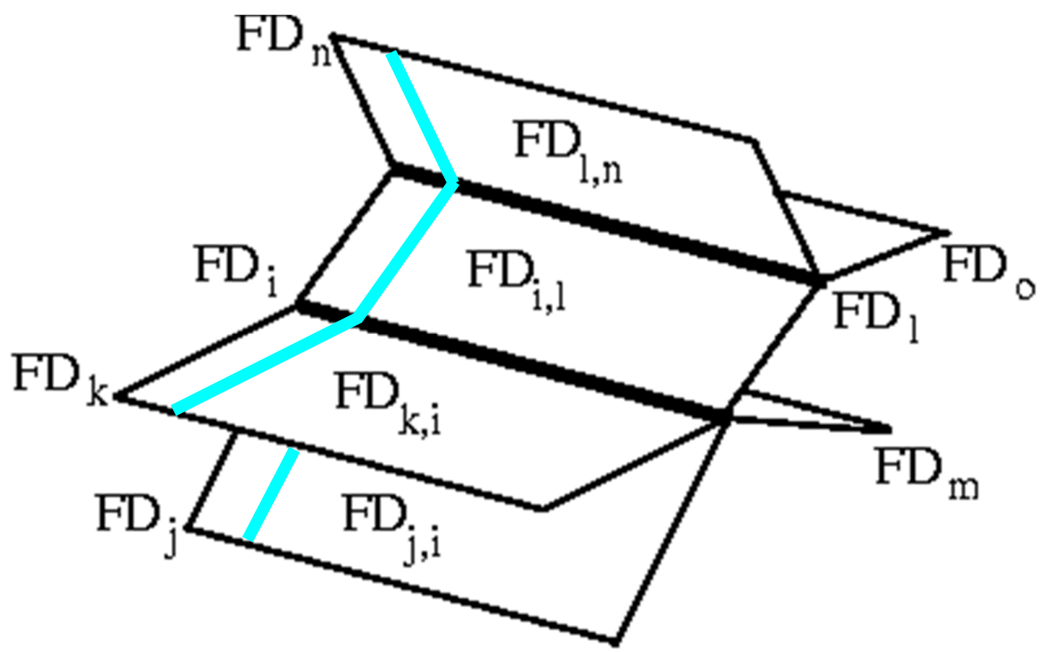


- Sweep all $FD_{i,j}$ with a sweep line from left to right

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep

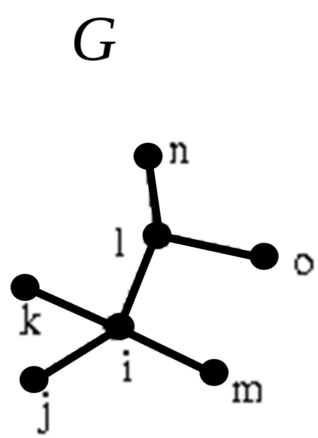


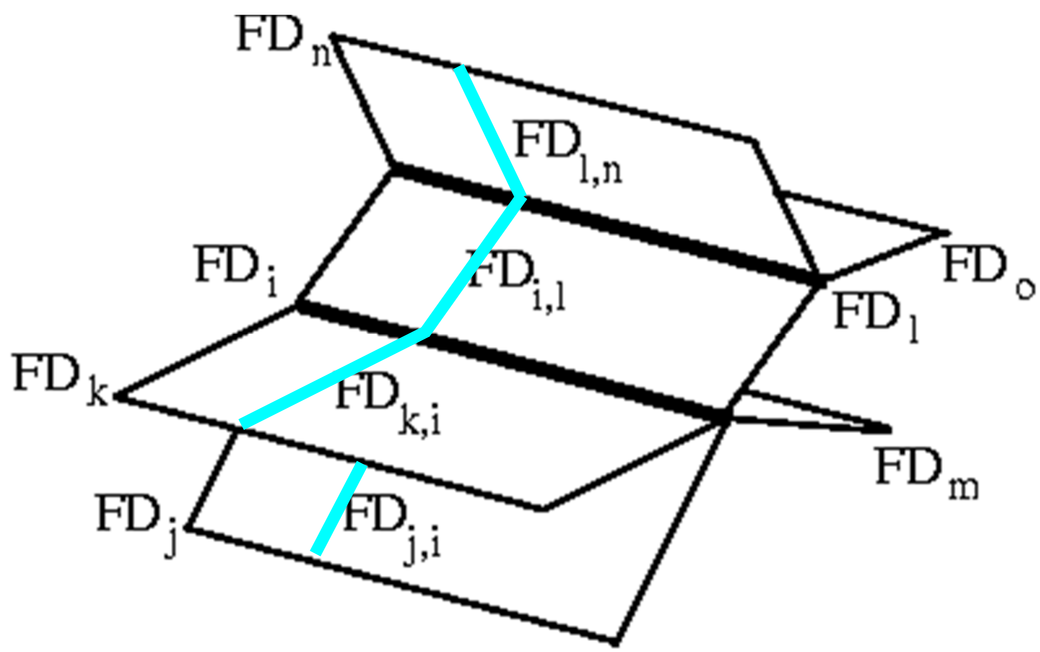


- Sweep all $FD_{i,j}$ with a sweep line from left to right

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep

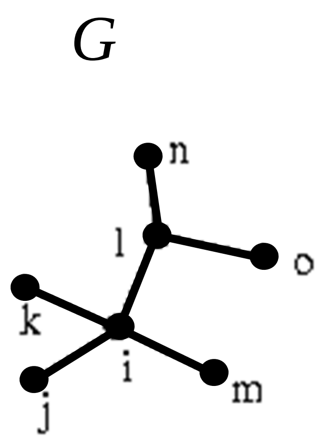


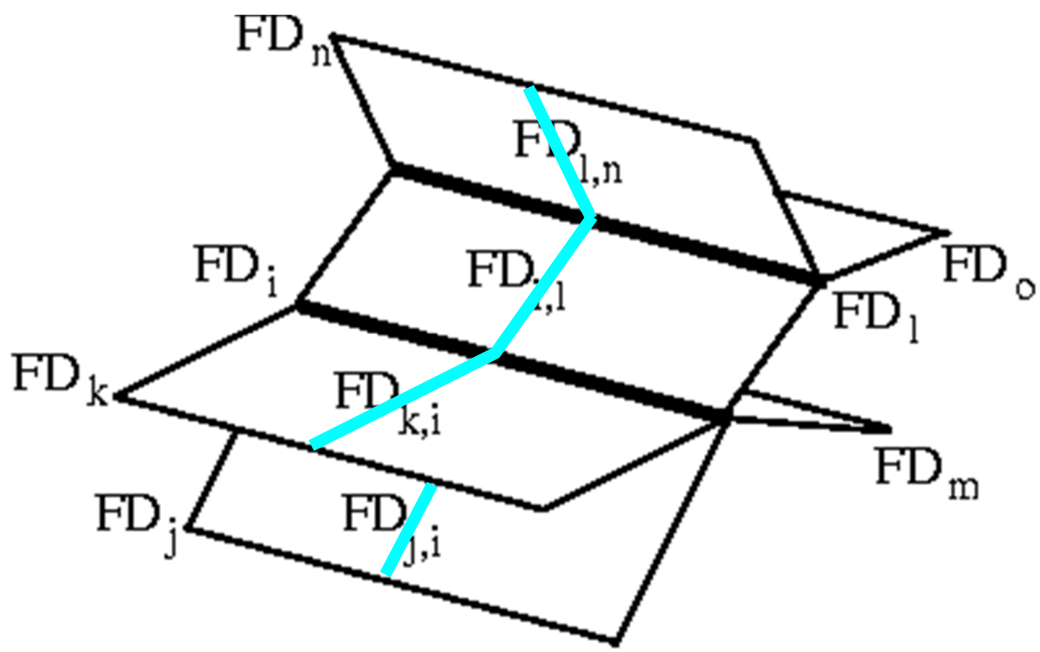


- Sweep all $FD_{i,j}$ with a sweep line from left to right

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep

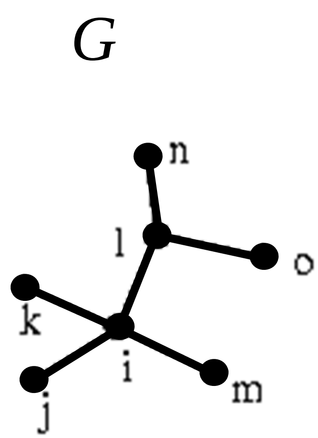


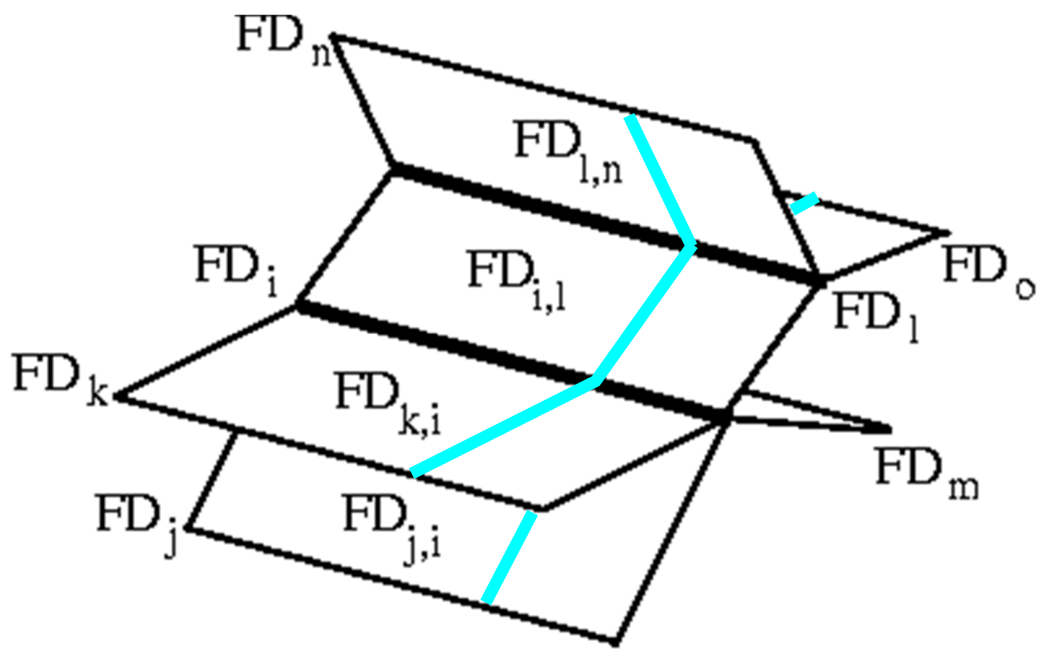


- Sweep all $FD_{i,j}$ with a sweep line from left to right

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep

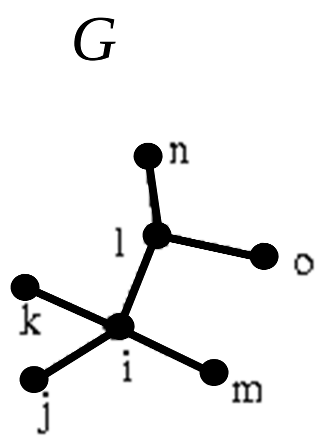


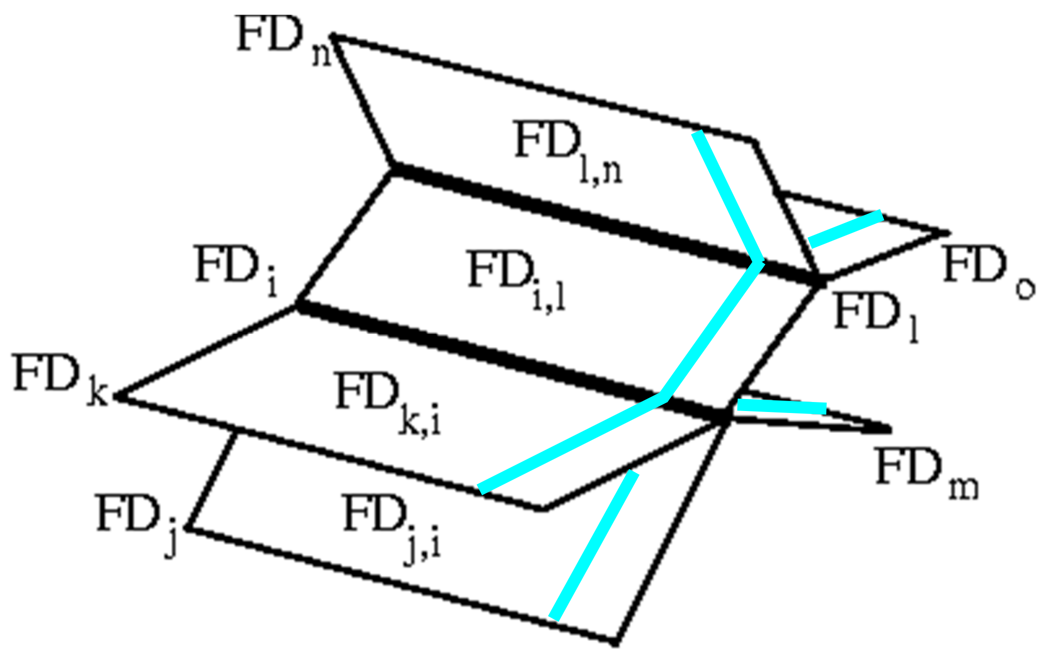


- Sweep all $FD_{i,j}$ with a sweep line from left to right

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep





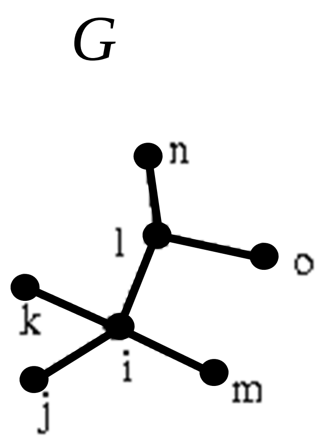
- Sweep all $FD_{i,j}$ with a sweep line from left to right

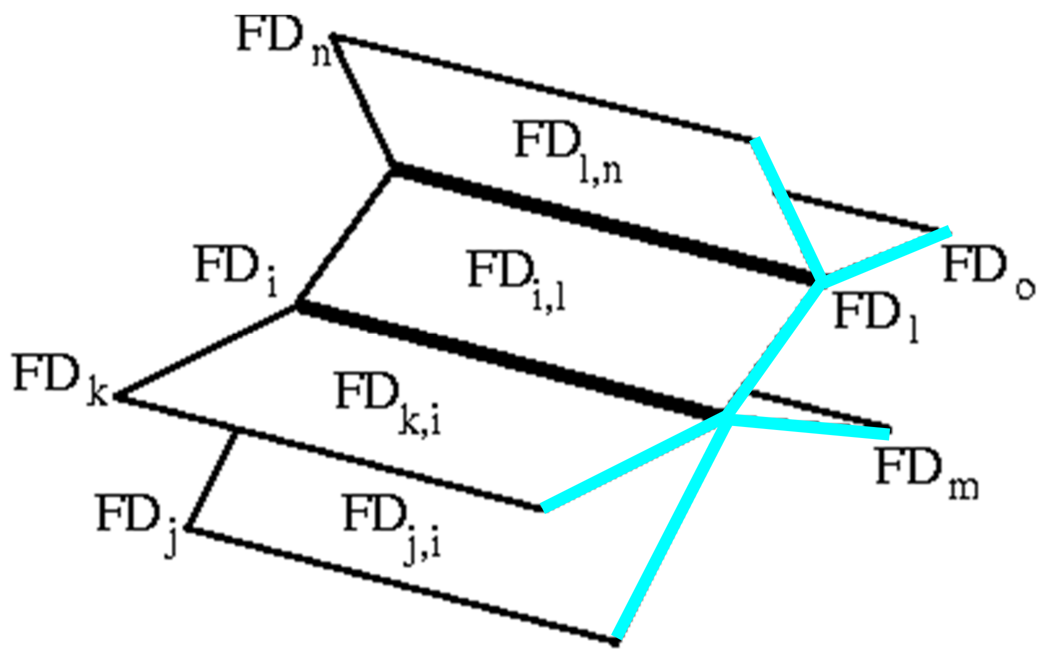
[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.

[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.

[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Sweep

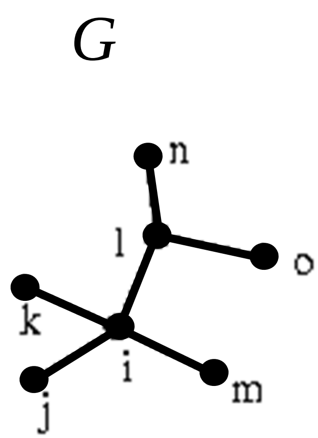


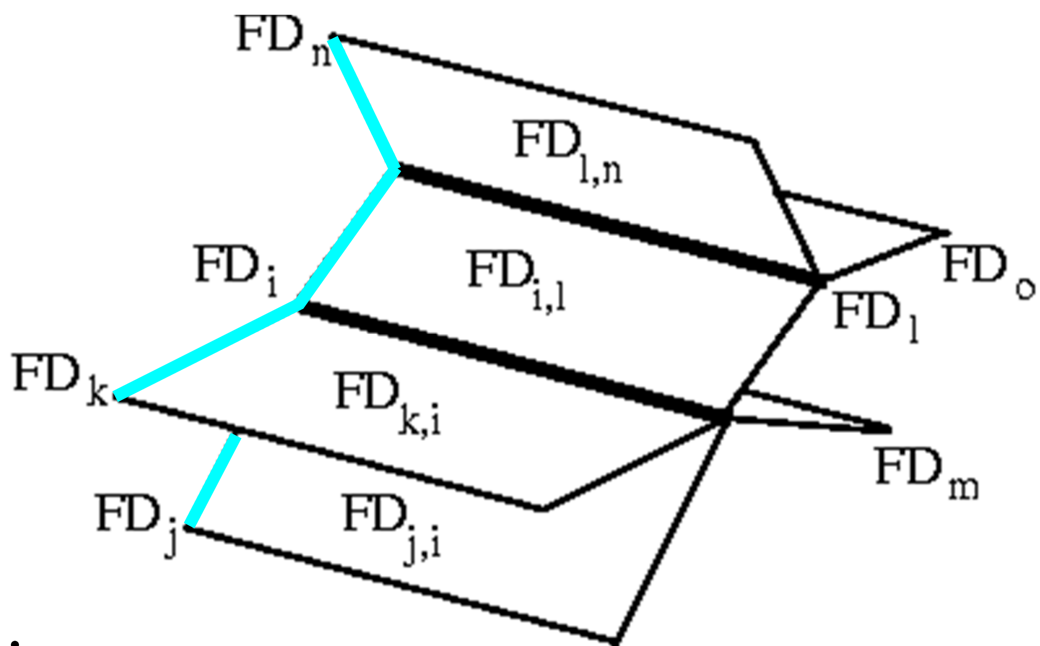


- Sweep all $FD_{i,j}$ with a sweep line from left to right

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Compute Reachable Points





- During the sweep:

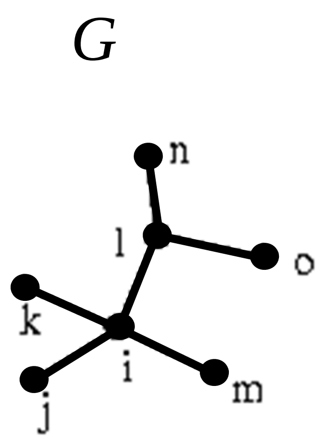
  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

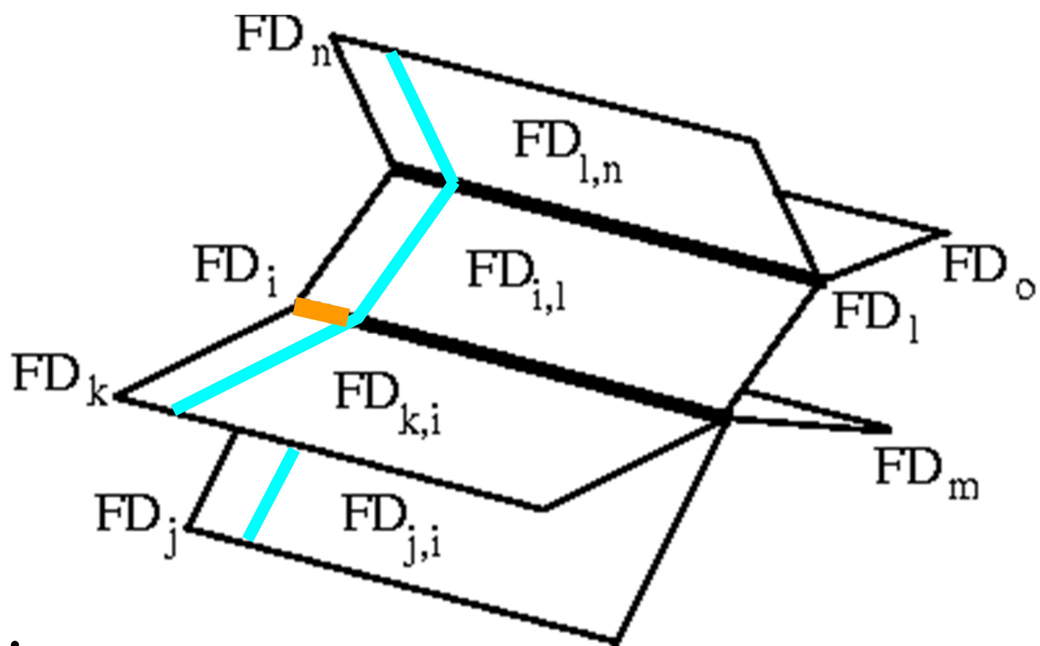
[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.

[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.

[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Compute Reachable Points



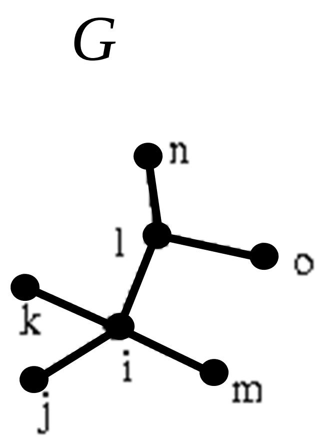


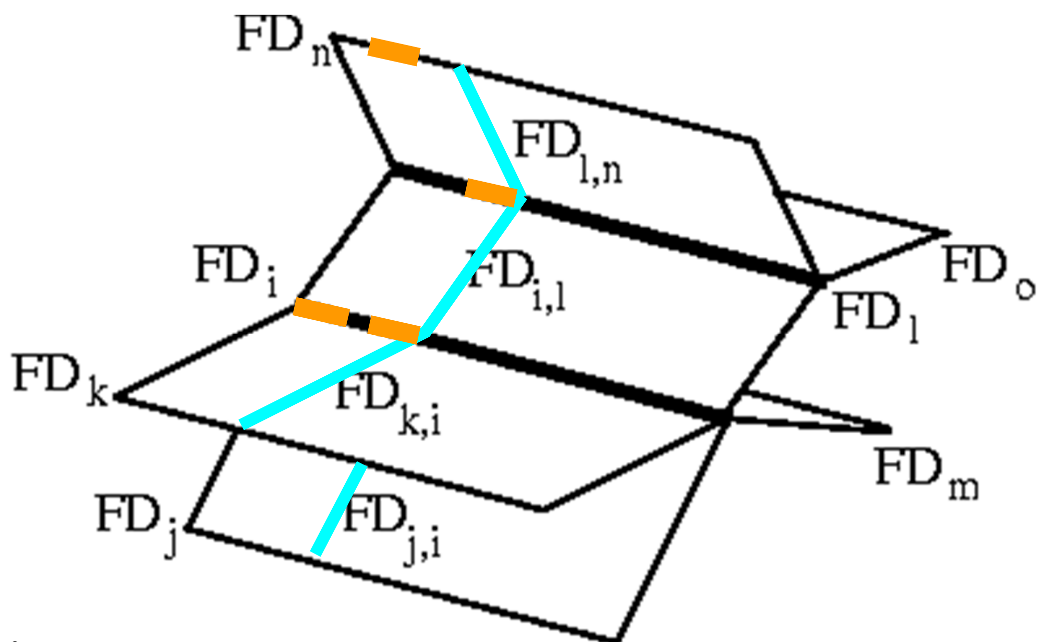
- During the sweep:

  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Compute Reachable Points



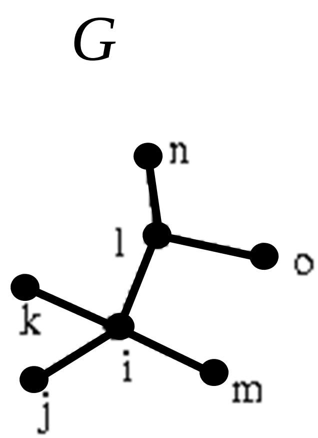


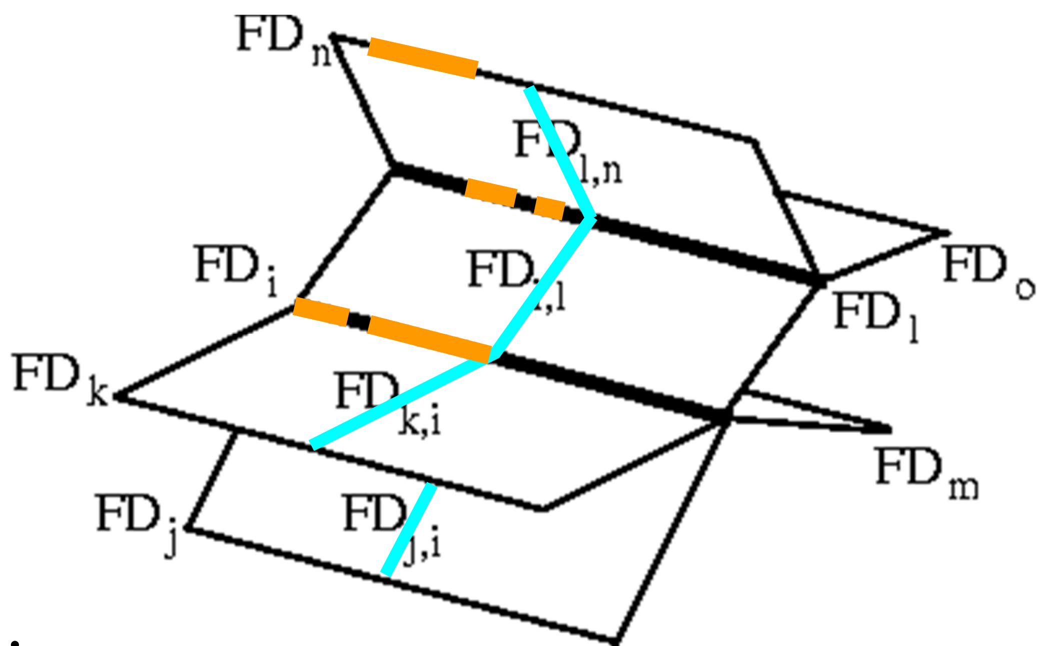
- During the sweep:

  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Compute Reachable Points



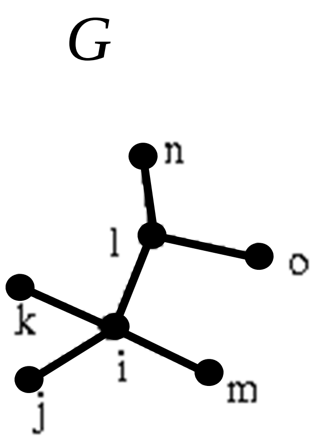


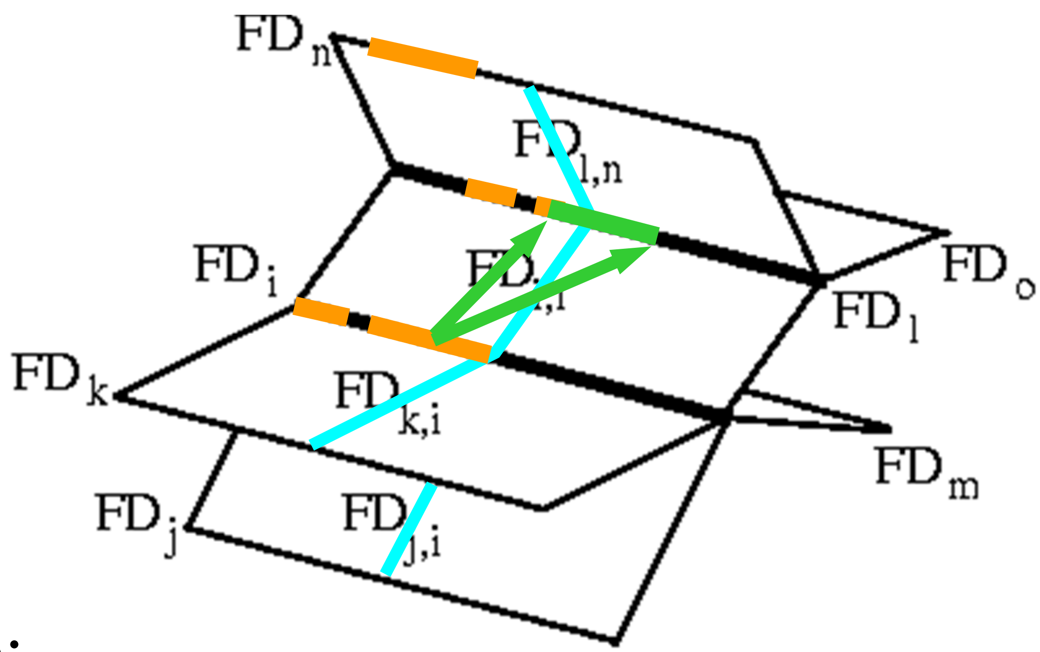
- During the sweep:

  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

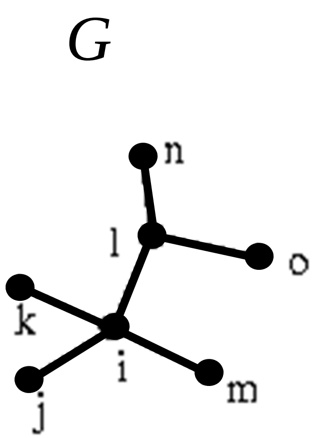


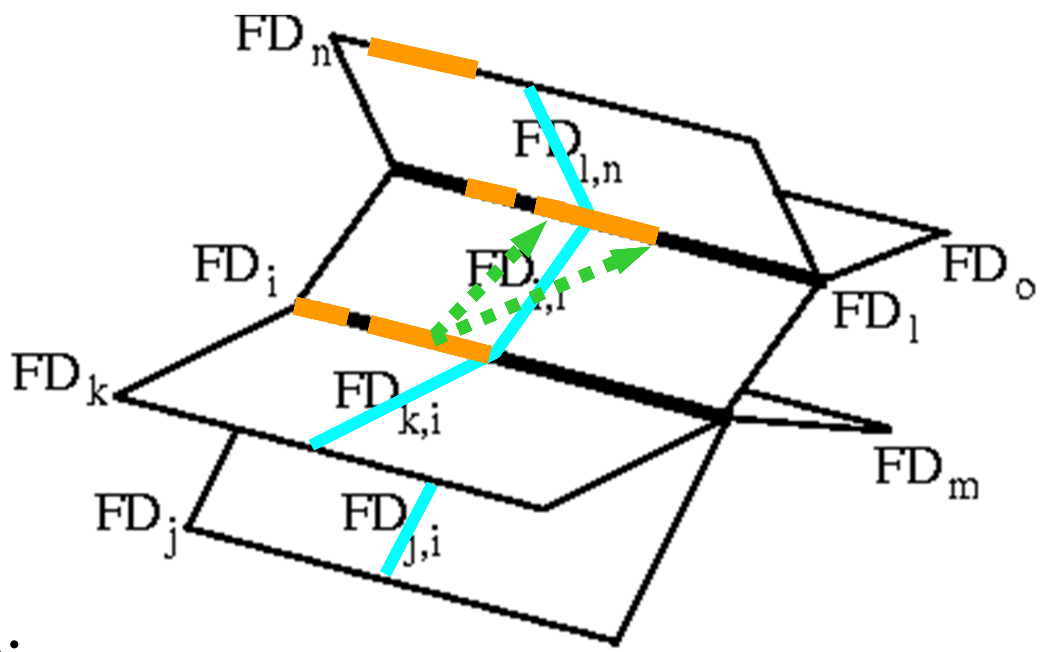


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

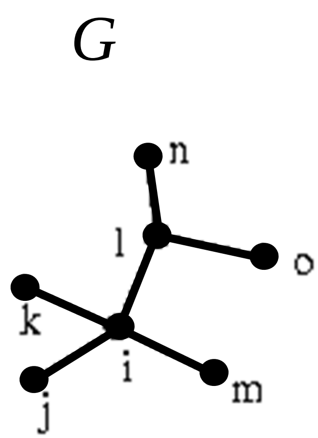


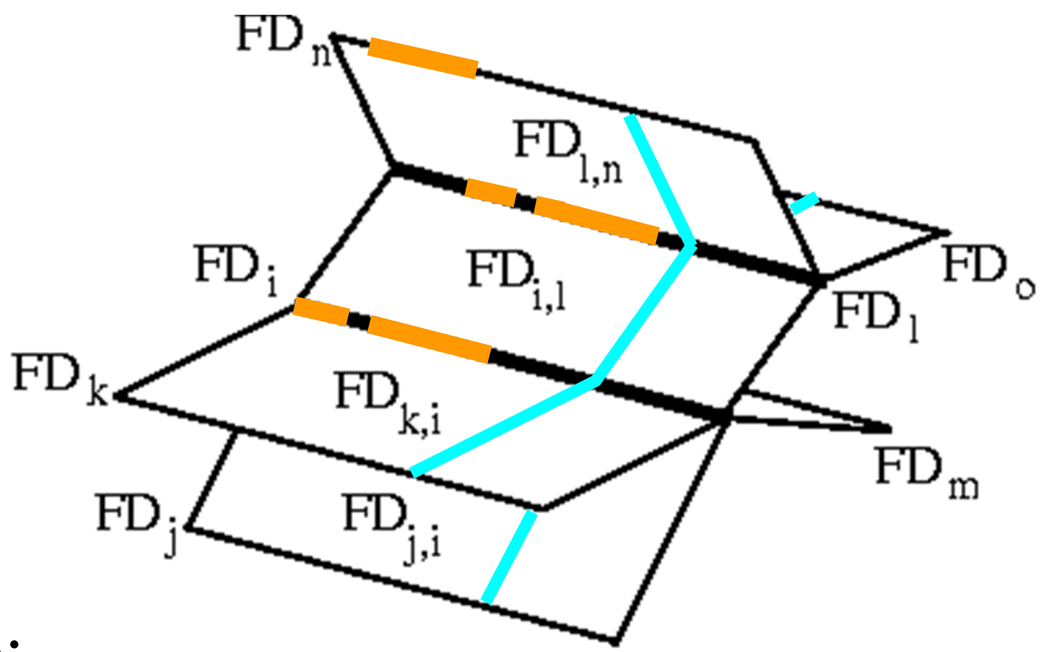


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

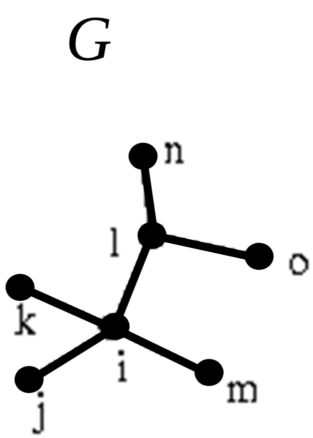


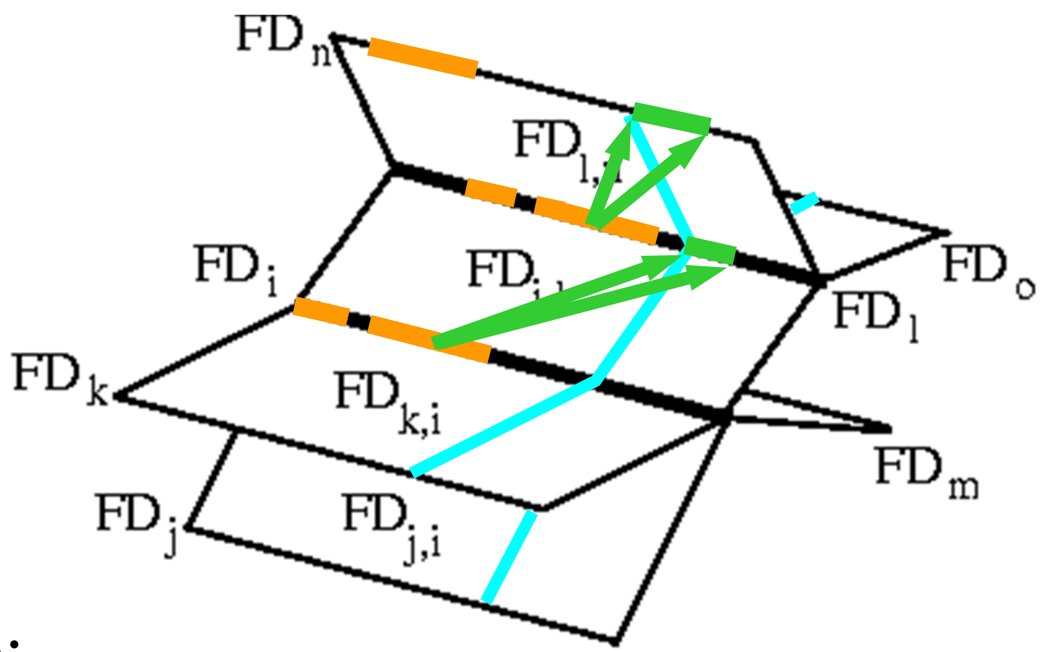


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

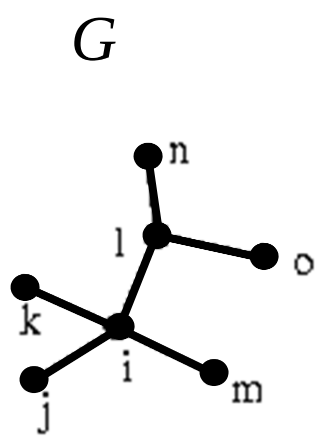


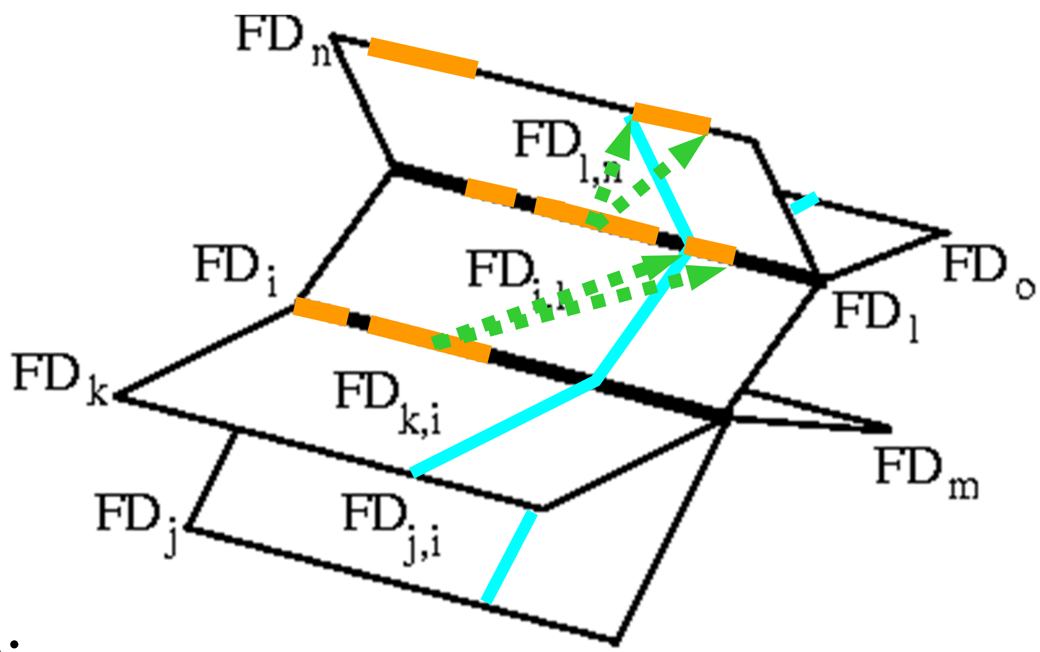


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

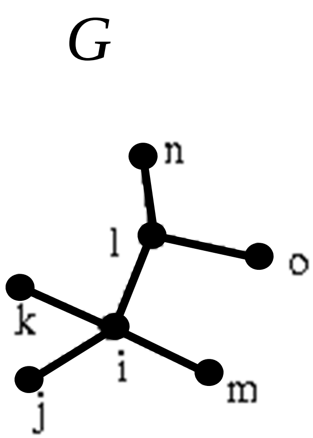


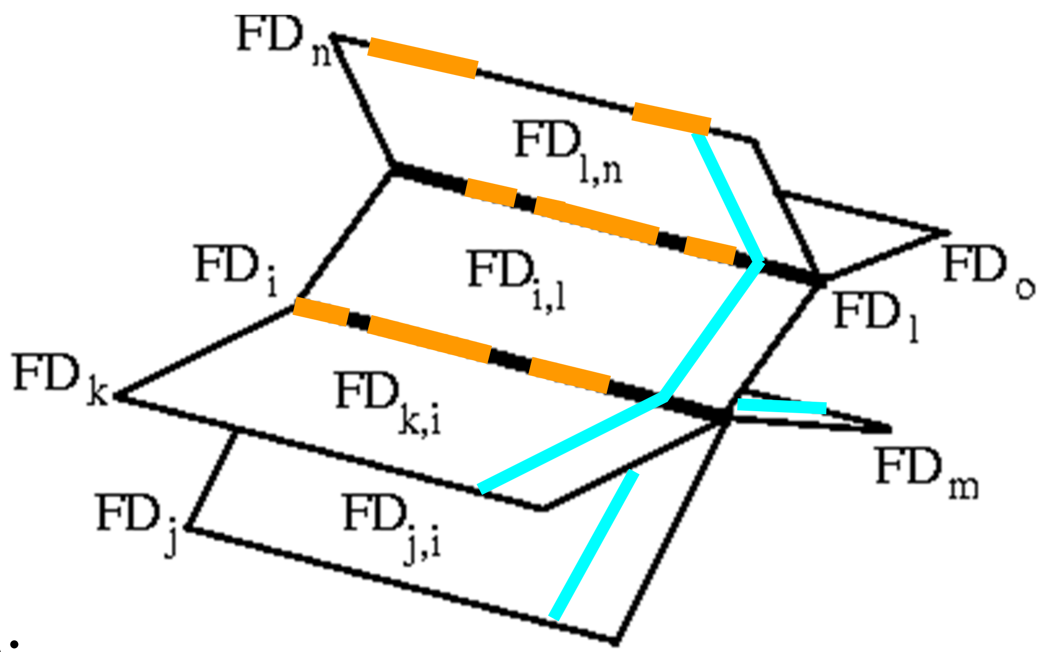


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

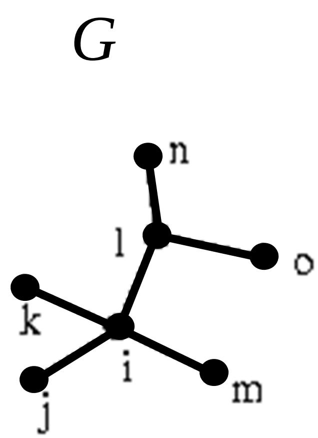


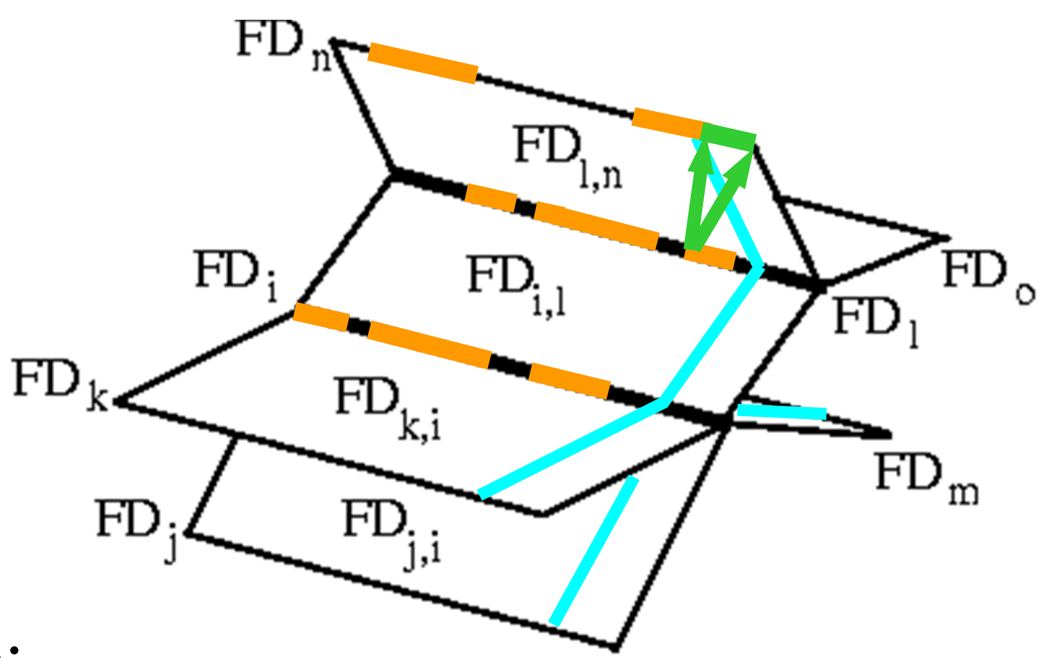


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

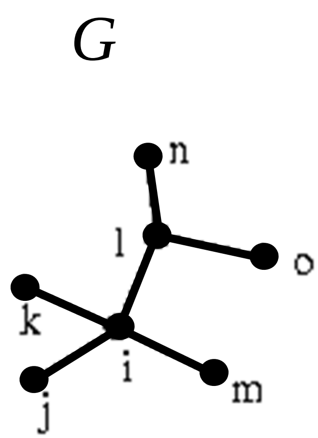


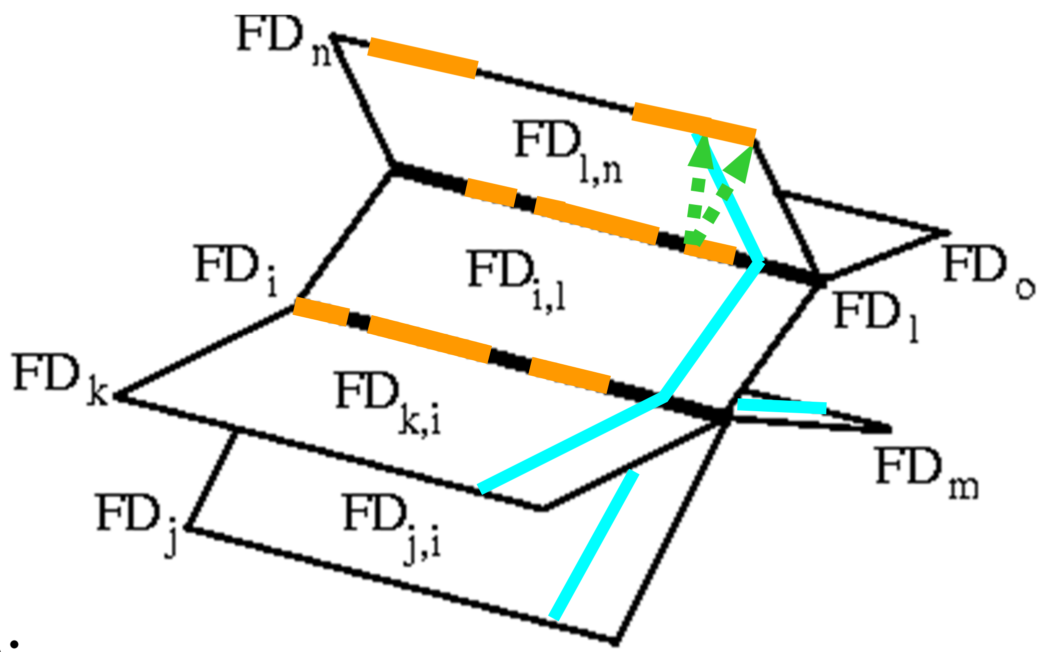


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

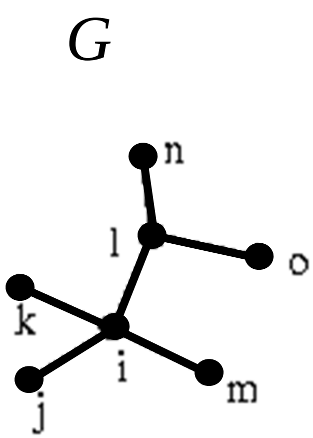


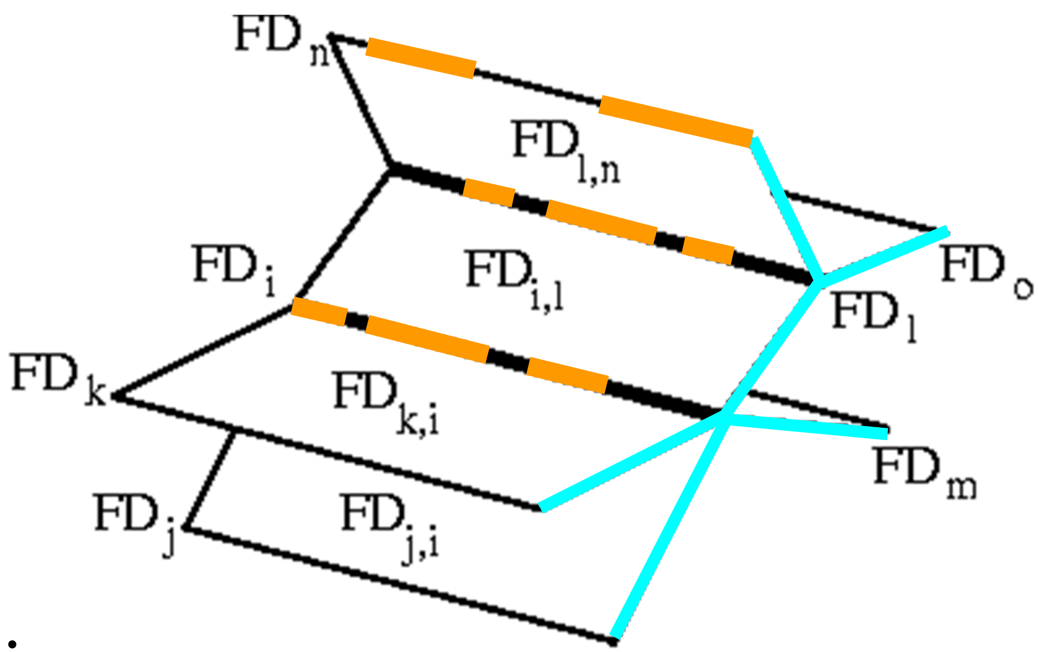


- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Update Reachable Points

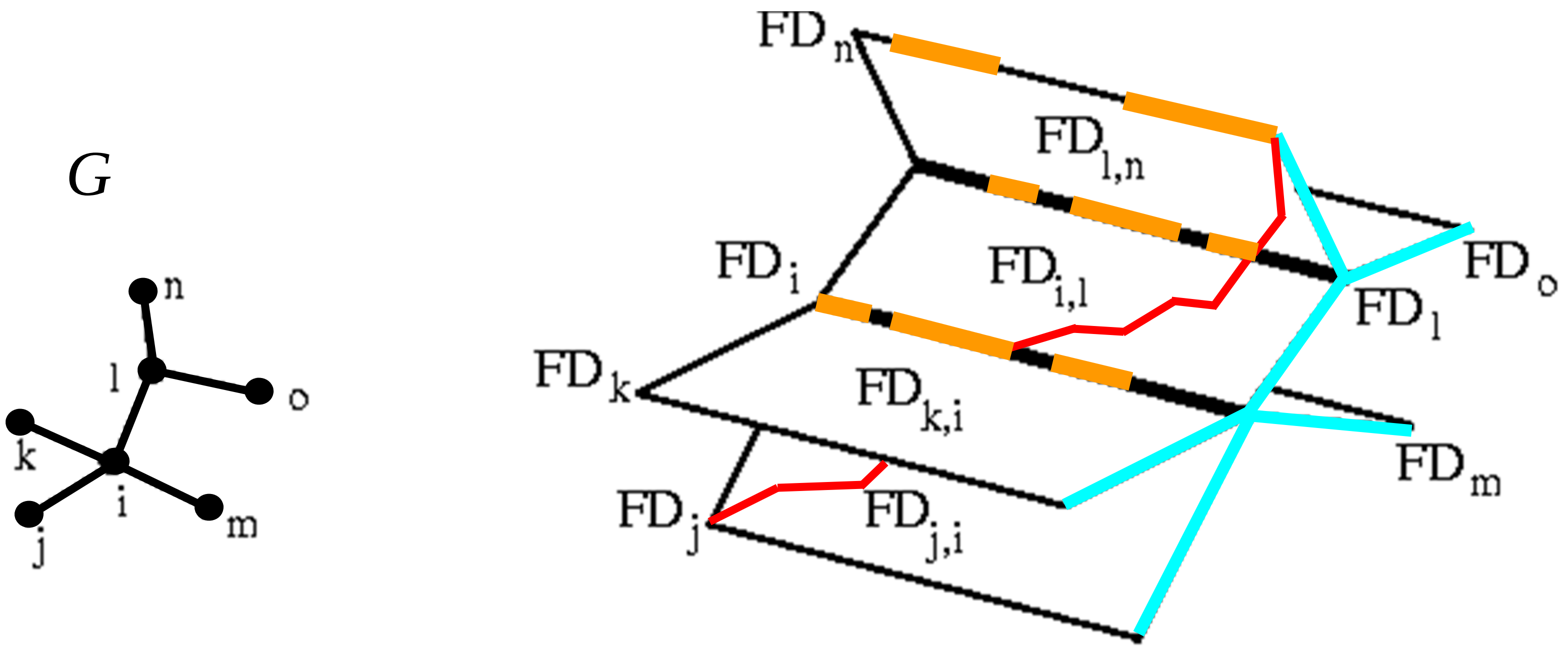




- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[AERW03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.
[BPSW05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.
[WSP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Backtracking



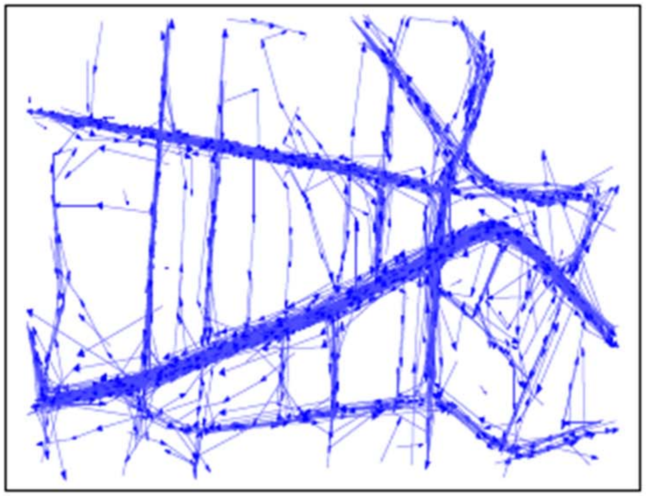
- After the sweep:
  - Construct a monotone path via backtracking
- Runtime:
  - O(mn log(mn)) time for decision problem; optimize using parametric search

[AER**W**03] H. Alt, A. Efrat, G. Rote, **C. Wenk**, Matching Planar Maps, *J. of Algorithms* 49: 262-283, 2003.

[BPS**W**05] S. Brakatsoulas, D. Pfoser, R. Salas, **C. Wenk**, On Map-Matching Vehicle Tracking Data , VLDB 853-864 , 2005.

[**W**SP06] **C. Wenk**, R. Salas, D. Pfoser, Adressing the Need for Map-Matching Speed…, SSDBM: 379-388, 2006.

Map Construction



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 ε.

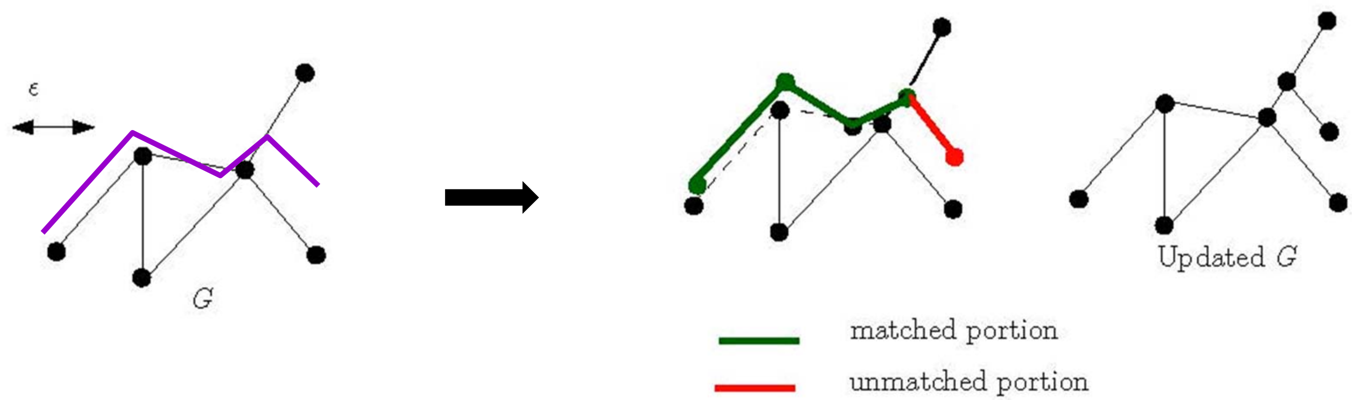
1. We assume each input curve is within Frechet distance ε/2 of a street-path in the original map.
2. (We assume all input curves sample acyclic paths.)
3. Two additional assumptions on original map help us to provide quality guarantees.

[A**W**12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

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:



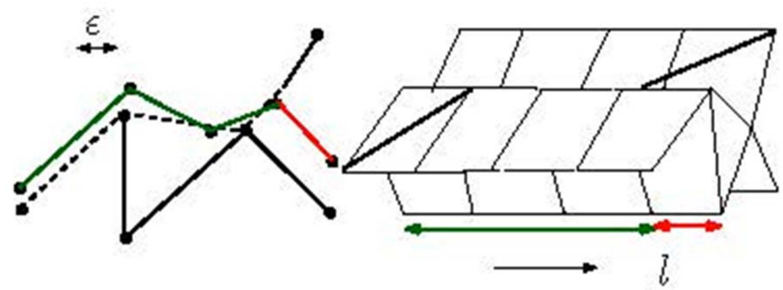
2. Use min-link curve simplification algorithm to reconcile existing portions

[AW12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

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.



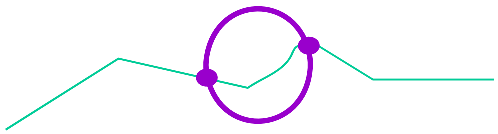
   ⇒ Project free space onto curve:
   white interval = matched portion, black interval = unmatched portion

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

[AW12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

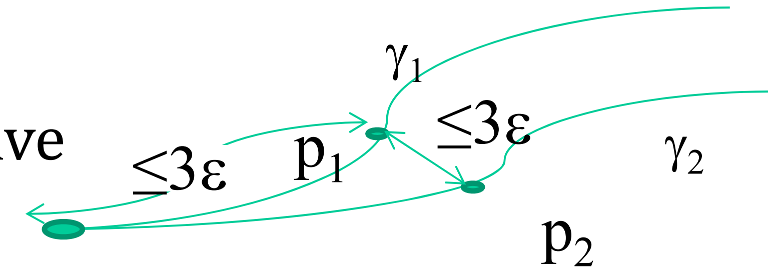
Assumptions

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

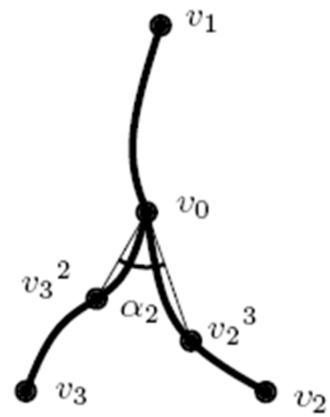


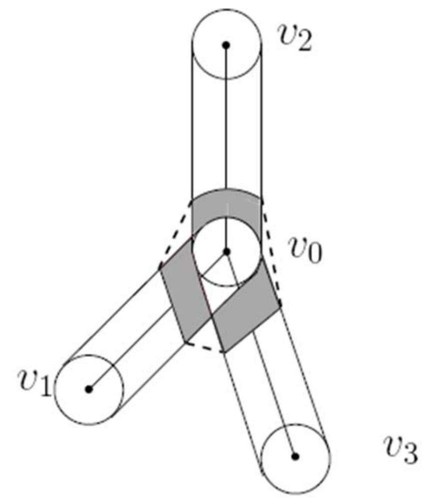
⇒ Projection approach is justified, because free space has special structure. Trajectory can only sample one good section in original network.

[AW12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

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.





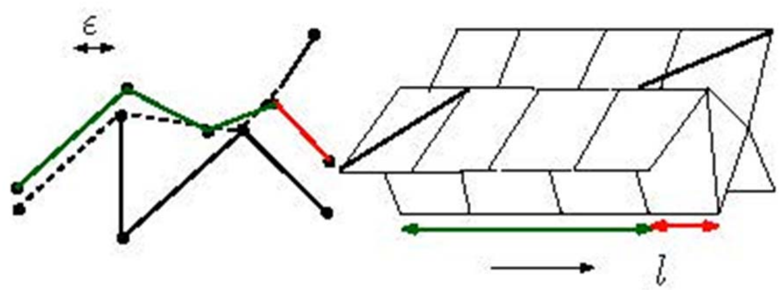
$\Rightarrow$ It is relatively easy to handle well-sampled clean data.
Deal with noisy data that is not well-sampled and give quality guarantees.

[A**W**12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

Map Construction [AW12]

Incrementally add one trajectory after another.
For each trajectory:

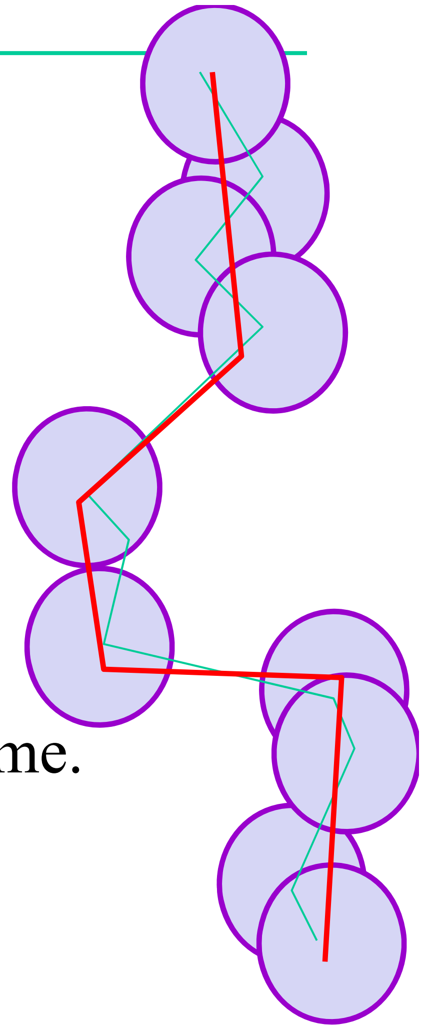
1. Use partial Fréchet distance to identify new and existing portions by combining mapmatching with partial Fréchet distance:
   - Compute free space surface
   - Find path that maximizes matched portion on the curve.

   ⇒ Project free space onto curve:
   white interval = matched portion, black interval = unmatched portion
2. Use min-link curve simplification algorithm to reconcile existing portions

[AW12] M. Ahmed, **C. Wenk**, Constructing Street Networks from GPS Trajectories, ESA: 60-71, 2012.

Curve Simplification

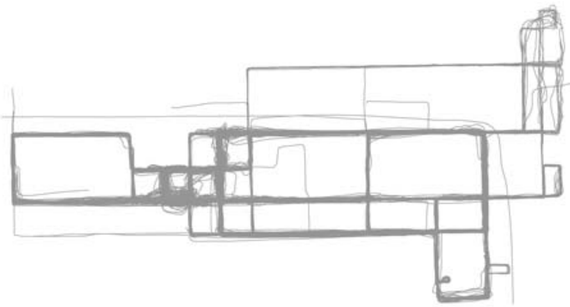


- **Arbitrary vertices:**
  - **min-#:** Compute a minimum link path that stabs ε-neighborhoods around vertices of f in the correct order. O(n) time.
  - **min-ε:** Binary search on critical values. O(n^2) time.
- **Subsequence of original vertices:**
  - Douglas-Peucker algorithm, O(n log n) time.
  - **min-#:** Add shortcut-edges between vertices if shortcut is within error ε. Then find a path with min-# edges. O(n^2) time, and O($n^{4/3+\delta}$) for x-monotone chains.

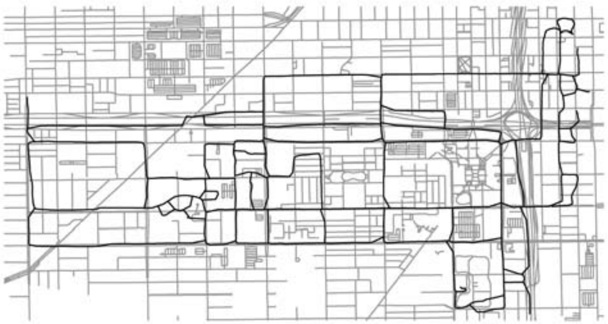
[AG99] H. Alt, L. Guibas, Discrete Geometric Shapes…, Handbook of Computational Geometry: 121-153, 1999.
[GHMS93] L. Guibas, J. Hershberger, J. Mitchell, J. Snoeyink, Approximating …, minimum link paths, IJCGA 3(4): 383-415, 1993
[II88] H. Imai, M. Iri, Polygonal approximations of a curve…, in Computational Morphology, Toussaint (eds): 71-86, 1988

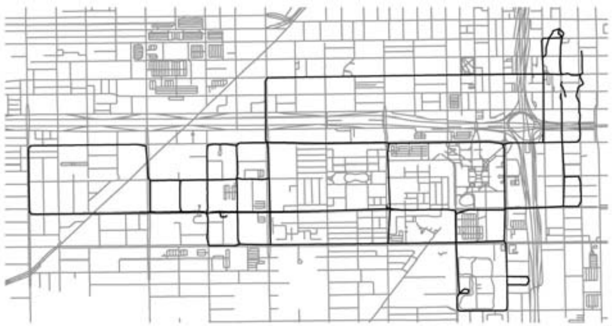
Different Reconstructions

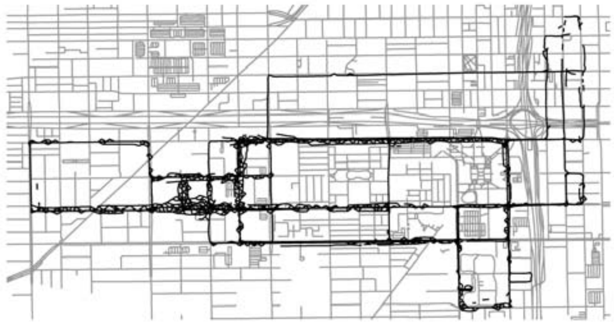
GPS Trajectory Data



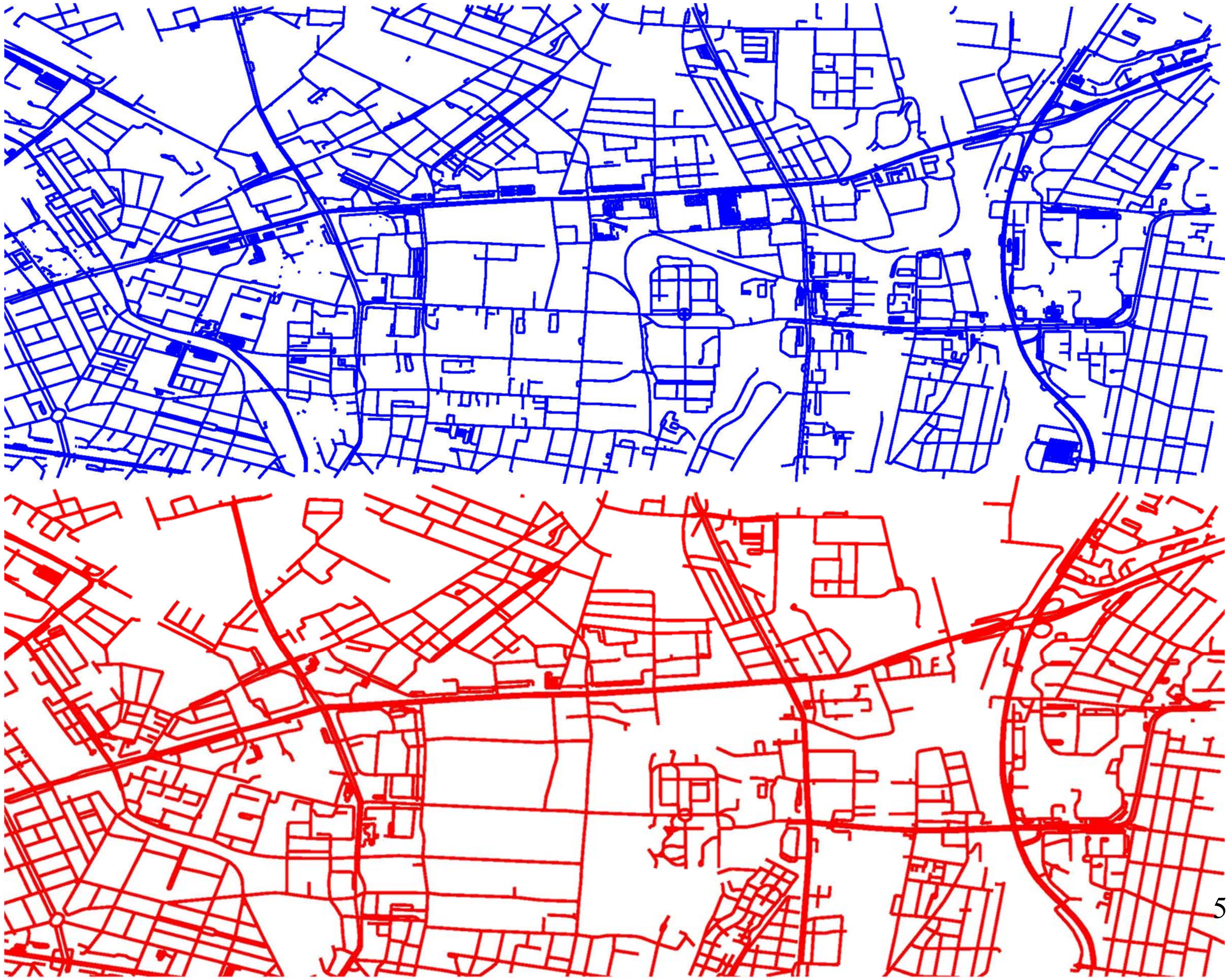
Reconstructed Maps







Which is the Better Roadmap ?



Which is the Better Roadmap ?

⇒ Map Comparison

Distance Measures for Map Comparison

- [BE12b]: Graph sampling-based distance measure in local neighborhood.
  Maximize number of marbles and holes that match 1-to-1.

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

- [BE12], [AKP**W**14]: Overview / benchmark papers

[AFH**W**14]: Considers maps as sets of paths, and compares path sets.

[AF**W**13]: Compares local topology of graphs using persistent homology

[AFH**W**14] M. Ahmed, B. Fasy, K. Hickmann, **C. Wenk**, Path-based distance for street map comparison, arXiv:1309.6131, 2014.
[AF**W**13] M. Ahmed, B. Fasy, **C. Wenk**, Local homology based distance between maps, submitted to SoCG, 2013.

[KP12] S. Karagiorgou, D. Pfoser, On vehicle-tracking data-based road network generation, 20th ACM SIGPATIAL: 89-98, 2012.
[BE12] J. Biagioni, J. Eriksson, Map inference in the face of noise and disparity, 20th ACM SIGSPATIAL: 79-88, 2012
[BE12b] J. Biagioni, J. Eriksson, Inferring road maps from global… TRR: J. of the Transportation Research Board 2291, 61-71, 2012.
[AKP**W**14] M. Ahmed, S. Karagiorgou, D. Pfoser, **C. Wenk**, A Comparison and Evaluation of …, submitted to Geoinformatica, 2014.

[AFHW14] Path-Based Distance

- Directed Hausdorff distance on path-sets:

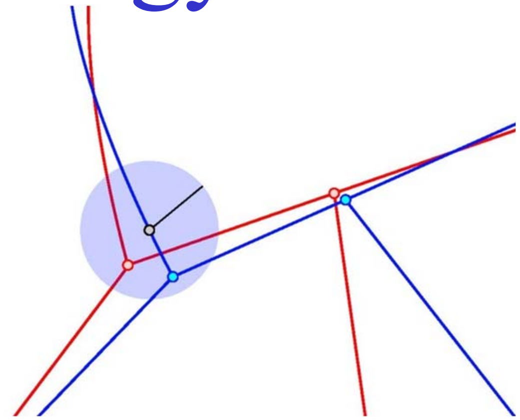
$$\overrightarrow{d}_{G,H}(\pi_G, \pi_H) = \max_{p_G \in \pi_G} \min_{p_H \in \pi_H} \delta_F(p_G, p_H)$$

- π_G path-set in G, and π_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.

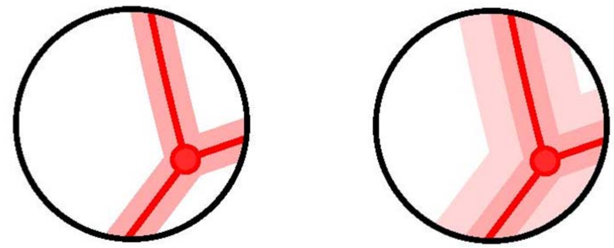
[AFH**W**14] M. Ahmed, B. Fasy, K. Hickmann, **C. Wenk**, Path-based distance for street map comparison, arXiv:1309.6131, 2014.
[AF**W**13] M. Ahmed, B. Fasy, **C. Wenk**, Local homology based distance between maps, submitted to SoCG, 2013.

[KP12] S. Karagiorgou, D. Pfoser, On vehicle-tracking data-based road network generation, 20th ACM SIGPATIAL: 89-98, 2012.
[BE12] J. Biagioni, J. Eriksson, Map inference in the face of noise and disparity, 20th ACM SIGSPATIAL: 79-88, 2012
[BE12b] J. Biagioni, J. Eriksson, Inferring road maps from global… TRR: J. of the Transportation Research Board 2291, 61-71, 2012.
[AKP**W**14] M. Ahmed, S. Karagiorgou, D. Pfoser, **C. Wenk**, A Comparison and Evaluation of …, submitted to Geoinformatica, 2014.

[AFW13] Local Homology Based Distance



- 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



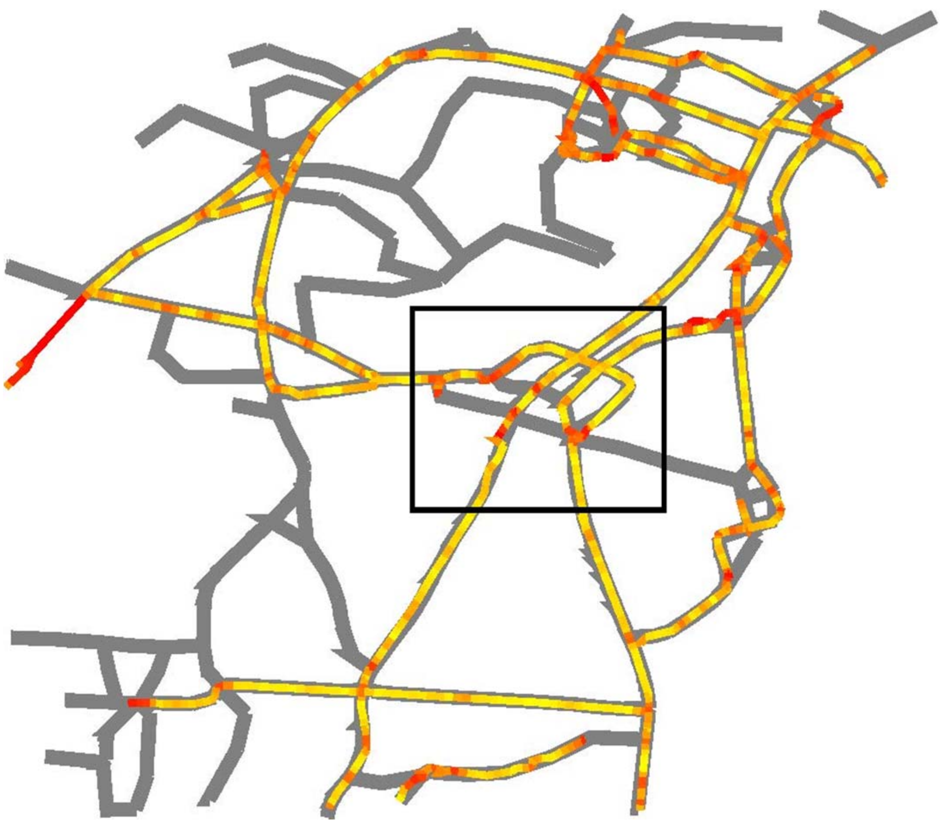
⟹ Use distance between persistence diagrams to compare changing local cycle structure

⟹ Local "signature" that captures local topological similarity of graphs

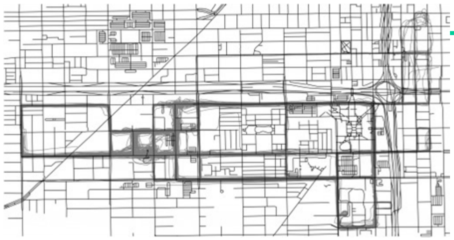
[AFW13] M. Ahmed, B. Fasy, **C. Wenk**, Local homology based distance between maps, submitted to SoCG, 2013.

[AFW13] Local Homology Based Distance

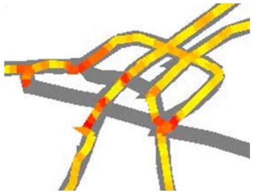
- Compared two reconstructed maps.
- Local signature captures different topology (missing intersections) well



[AFW13] M. Ahmed, B. Fasy, **C. Wenk**, Local homology based distance between maps, submitted to SoCG, 2013.



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.)