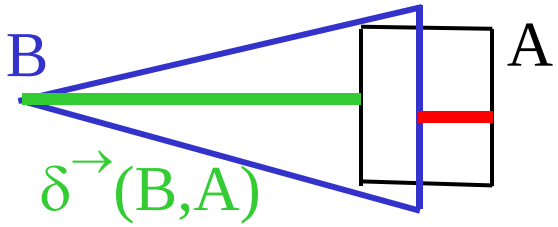
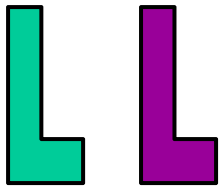
CMPS 3130/6130 Computational Geometry
Spring 2015

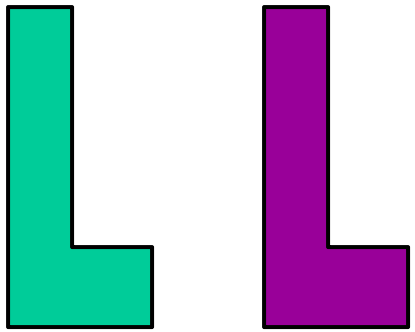


Shape Matching

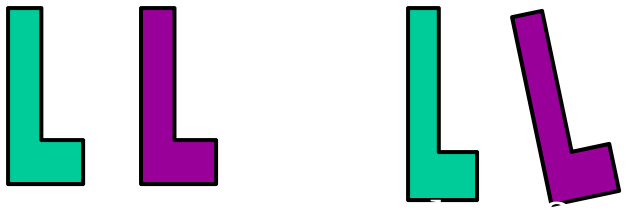
Carola Wenk

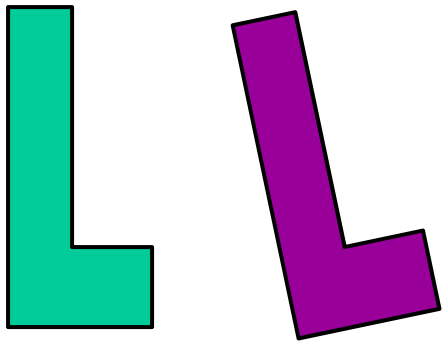
When are two shapes similar?



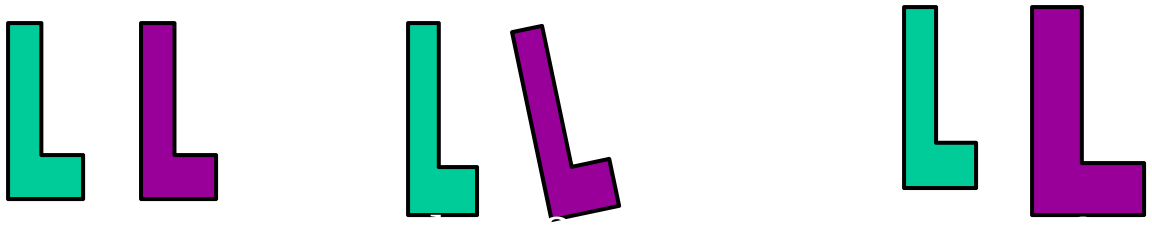


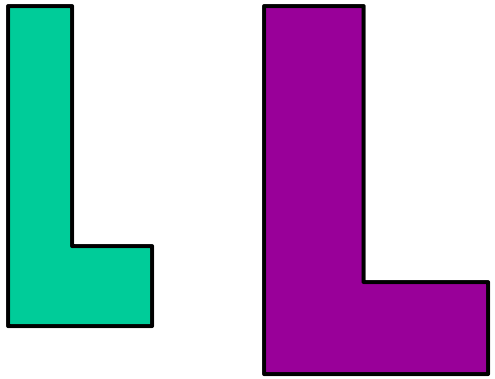
When are two shapes similar?



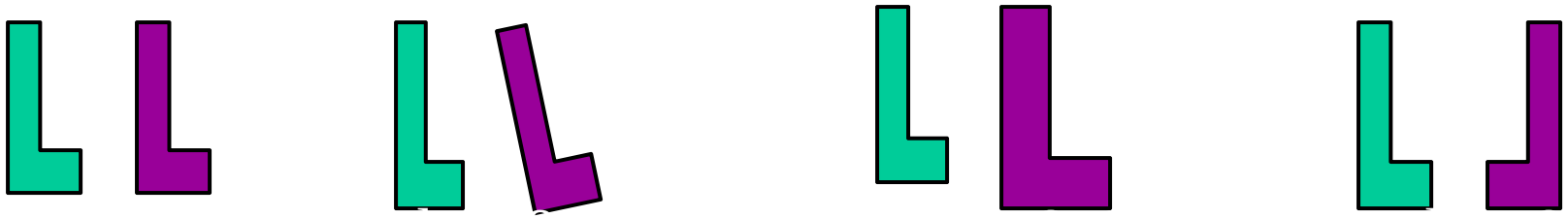


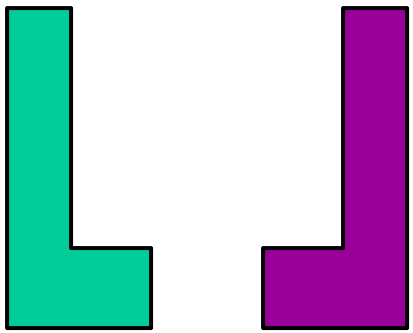
When are two shapes similar?



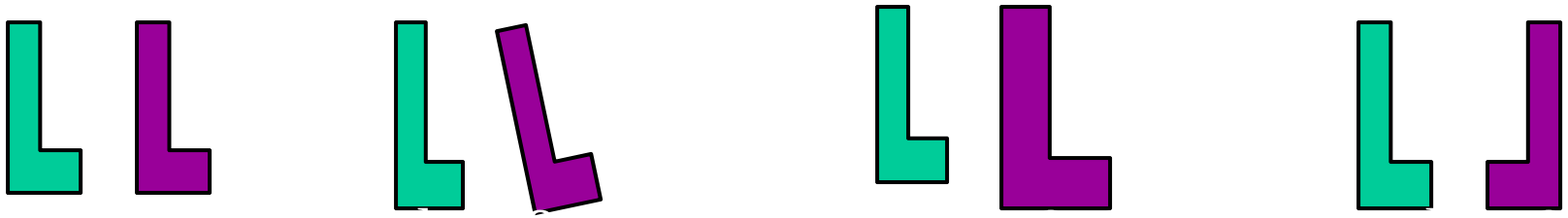


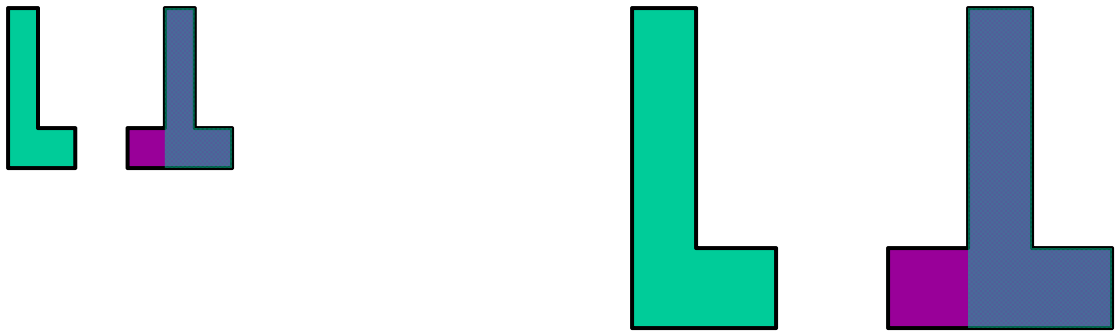
When are two shapes similar?



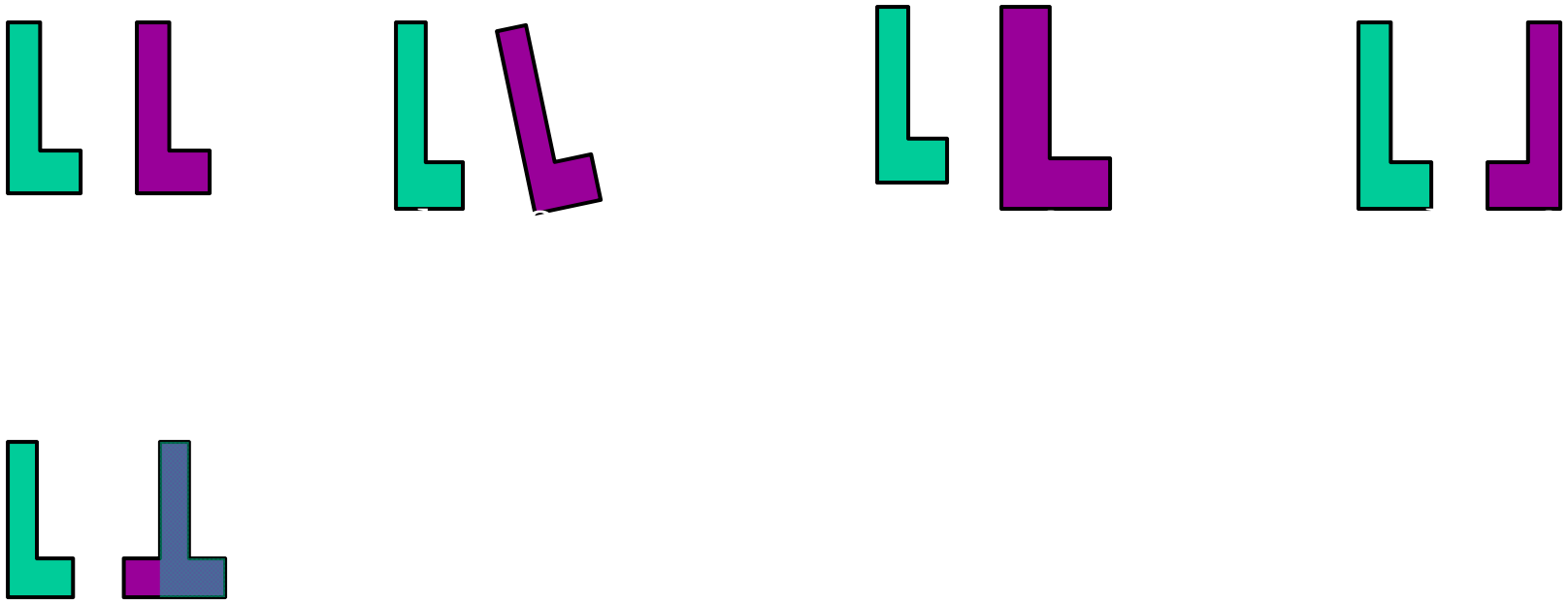


When are two shapes similar?





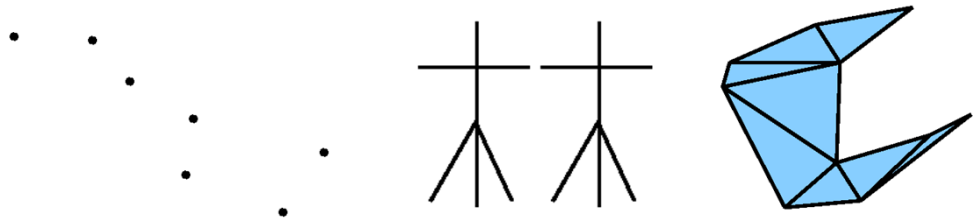
When are two shapes similar?



Geometric Shape Matching

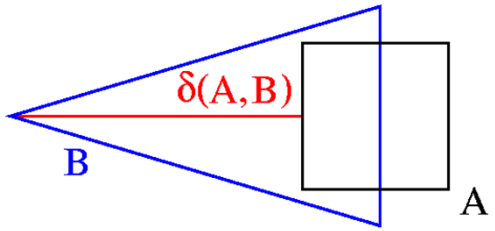
Given:

Two geometric shapes, each composed of a number of basic objects such as



points line segments triangles

An application dependant distance measure δ, e.g., the Hausdorff distance



A set of transformations $\boldsymbol{T}$, e.g., translations, rigid motions, or none

Matching Task: Compute $\min_{T \in \boldsymbol{T}} \delta(T(A),B)$

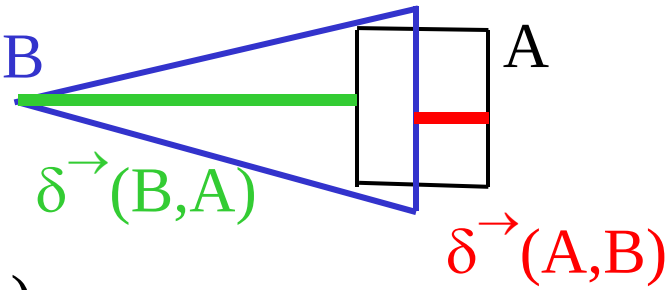
Hausdorff distance

- Directed Hausdorff distance

$$\vec{\delta}(A,B) = \max_{a \in A} \min_{b \in B} \| a-b \|$$

- Undirected Hausdorff-distance

$$\delta(A,B) = \max (\vec{\delta}(A,B) , \vec{\delta}(B,A))$$



- If A, B are discrete point sets, |A|=*m*, |A|=*n*
  - In d dimensions: Compute in O(*mn*) time brute-force
  - In 2 dimensions: Compute in O((*m*+*n*) log (*m*+*n*) time
    - Compute $\vec{\delta}(A,B)$ by computing VD(B) and performing nearest neighbor search for every point in A

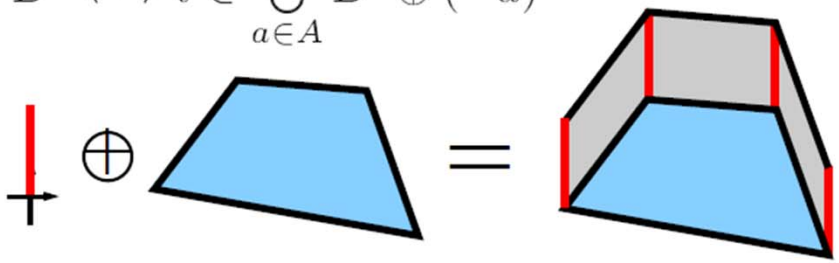
Hausdorff distance under translations

Decision problem, transformation space approach:

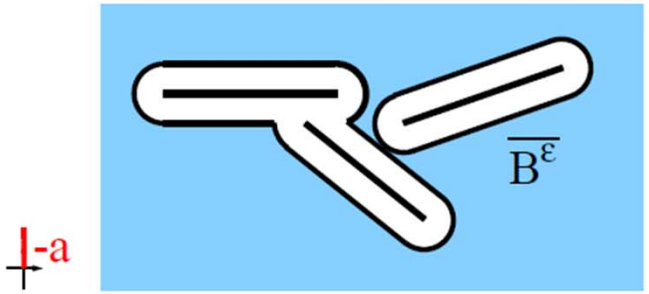
$$\vec{\delta}_H(A+t,B) \leq \epsilon \iff A+t \subseteq B^\epsilon \iff t \in \bigcup_{a \in A} \overline{B^\epsilon} \oplus (-a)$$

- Minkowski sum

  $$a \oplus b = \bigcup_{x \in a} \bigcup_{y \in b} x+y$$



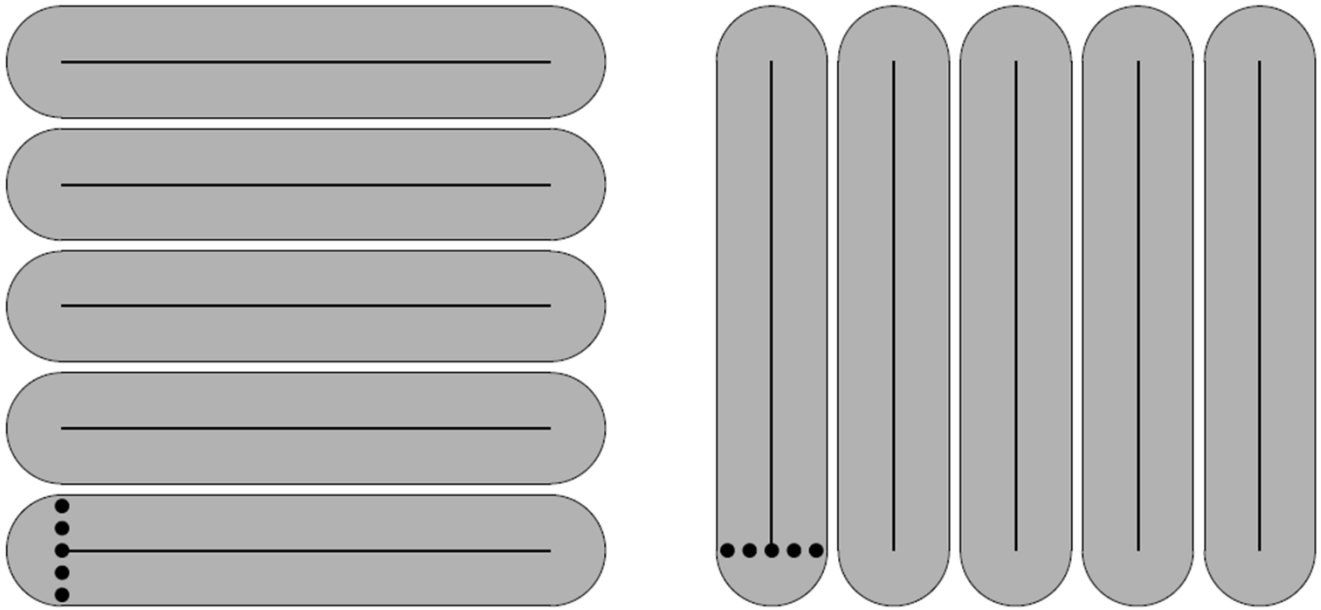
- Arrangement construction



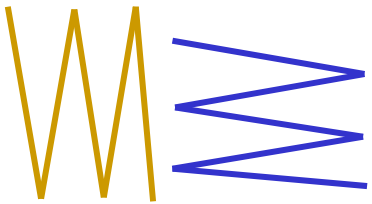
- Segments, $\mathbb{R}^2$: $O((n \cdot n)^2)$ arrangement complexity (AST94)
- Simplices, $\mathbb{R}^d$, L_∞: $O((\text{Complexity}(B^\epsilon))^d n^d) \leq O(n^{d^2+d})$
- Triangles, $\mathbb{R}^3$, L_2 : $O(n^{6+\delta})$ arrangement complexity

Hausdorff distance

• $\Omega(m^2n^2)$ lower bound construction for directed Hausdorff distance of line segments under translations

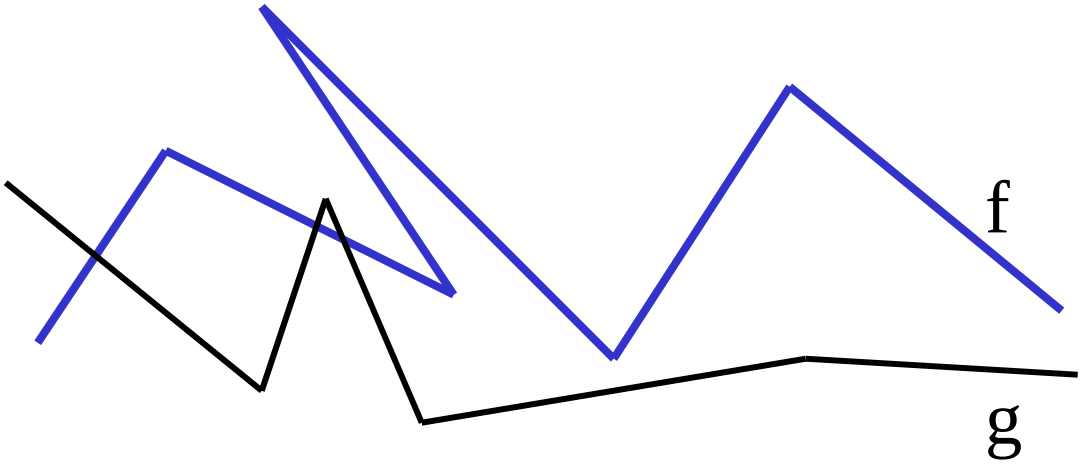


Comparing Curves and Trajectories



Polygonal Curves

- Let $f,g:[0,1] \rightarrow R^d$ be two **polygonal curves** (i.e., piecewise linear curves)

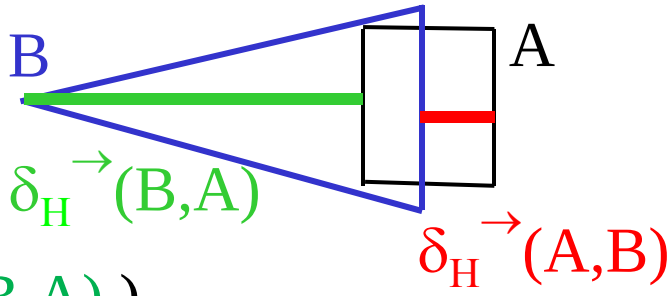


- What are good distance measures for curves?
  - Hausdorff distance?
  - Fréchet distance?

When Are Two Curves „Similar“?

- Directed Hausdorff distance

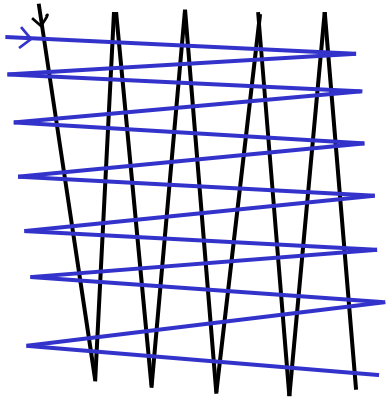
$$\delta_H^{\rightarrow}(A,B) = \max_{a \in A} \min_{b \in B} \| a-b \|$$



- Undirected Hausdorff-distance

$$\delta_H(A,B) = \max(\delta_H^{\rightarrow}(A,B), \delta_H^{\rightarrow}(B,A))$$

But:

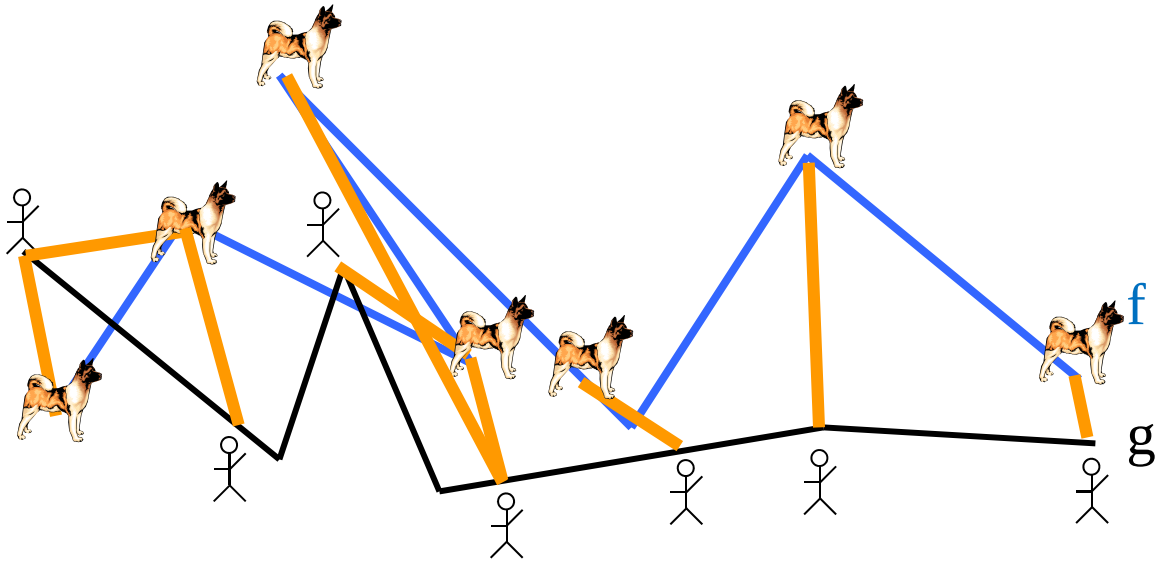


- Small Hausdorff distance
- When considered as curves the distance should be large
- The Fréchet distance takes the continuity of the curves into account

Fréchet Distance for Curves

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

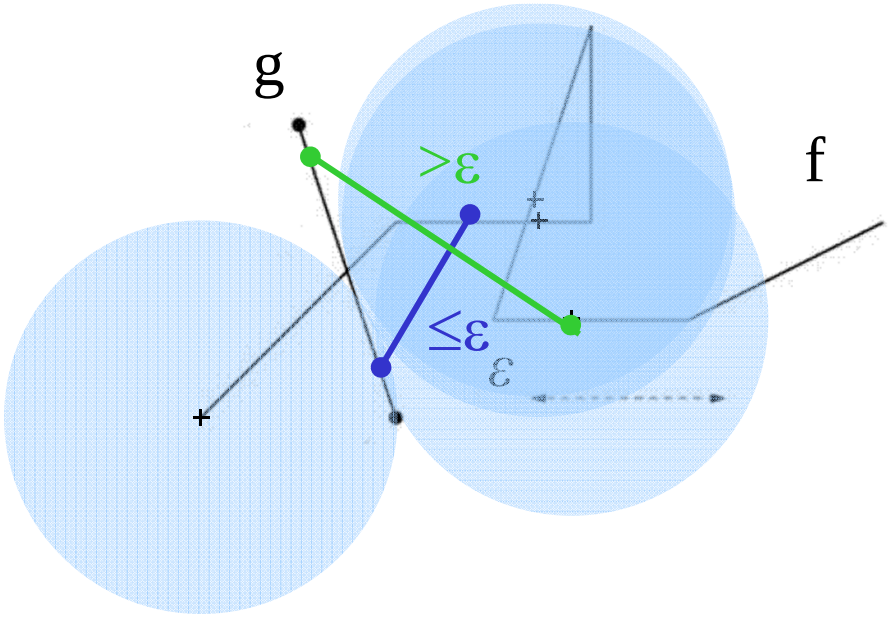
where α and β range over continuous monotone increasing reparameterizations only.

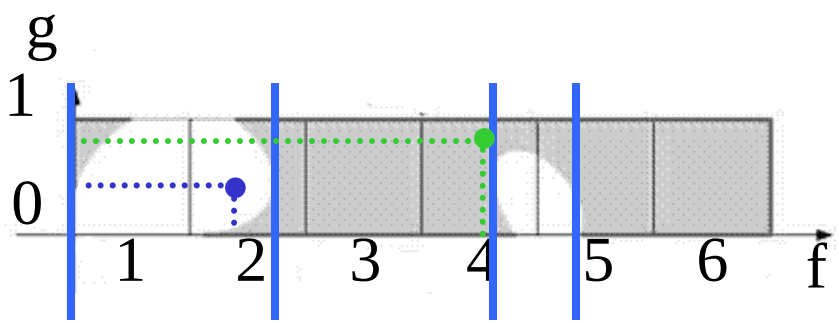


- Man and dog walk on one curve each
- They hold each other at a **leash**
- They are only allowed to go forward
- δ_F is the minimal possible leash length

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

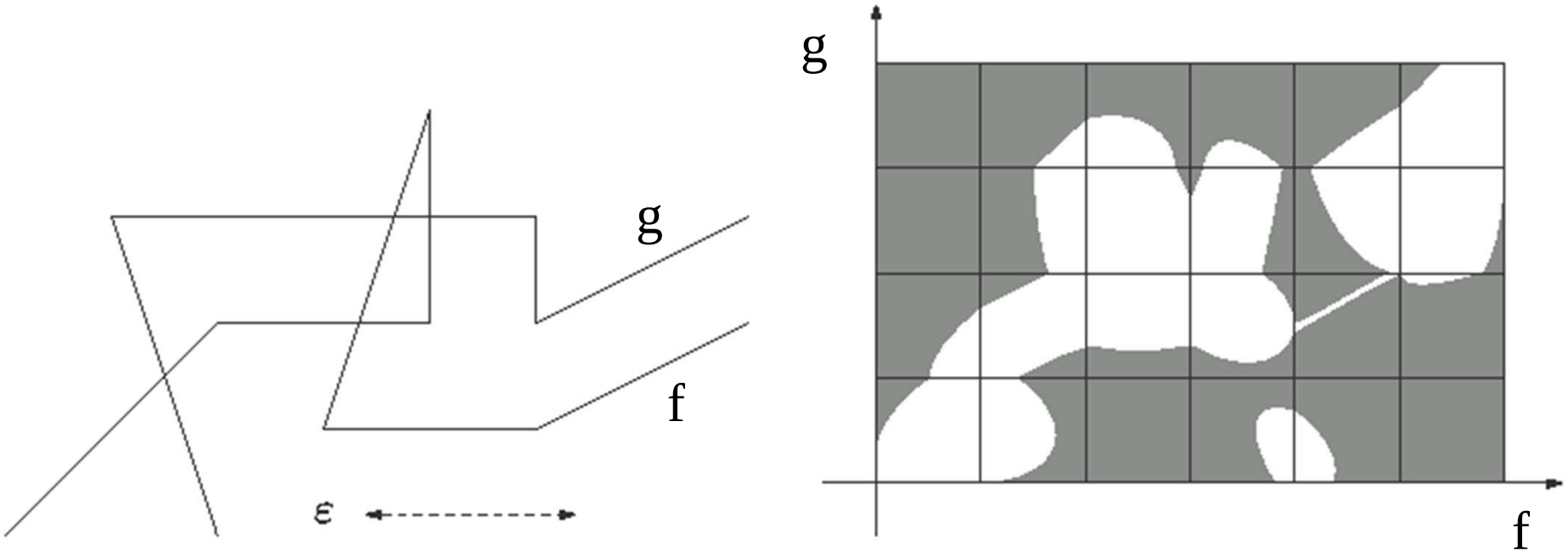
Free Space Diagram





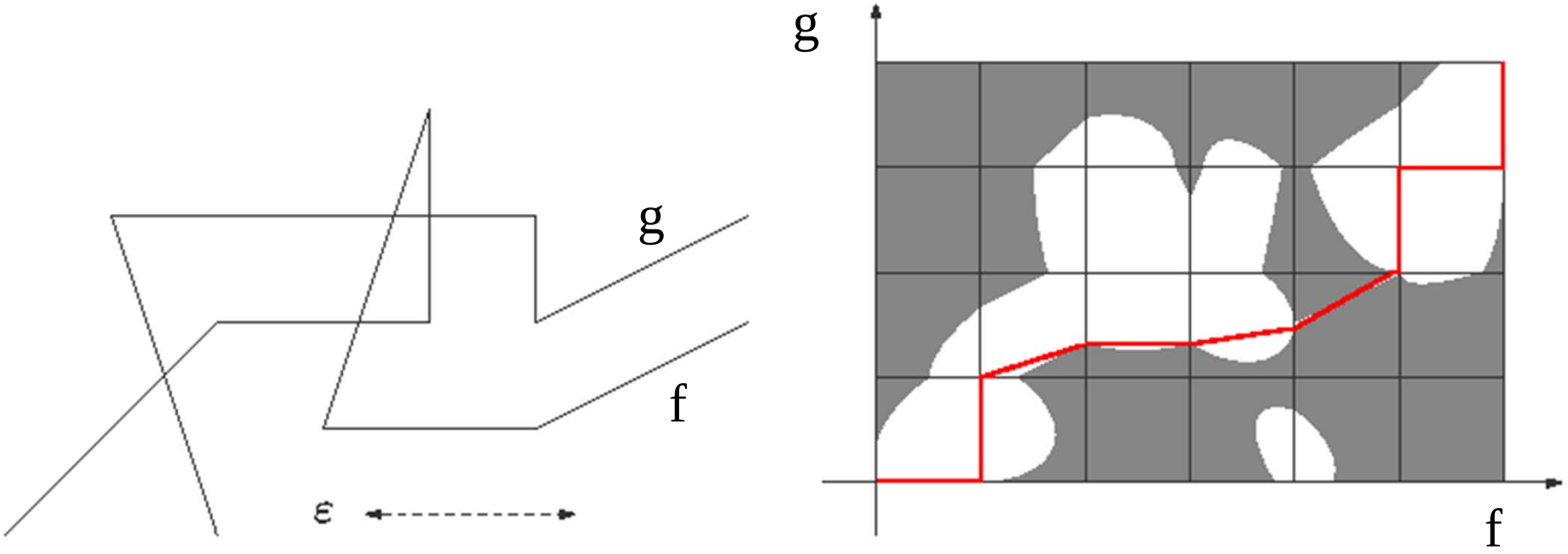
- Let ε > 0 fixed (eventually solve decision problem)
- $F_\varepsilon(f,g) = \{ (s,t) \in [0,1]^2 \mid \| f(s) - g(t) \| \leq \varepsilon \}$ *white points* **free space** of f and g
- The free space in one cell is an ellipse.

Free Space Diagram



- 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
- The free space in one cell is an ellipse.

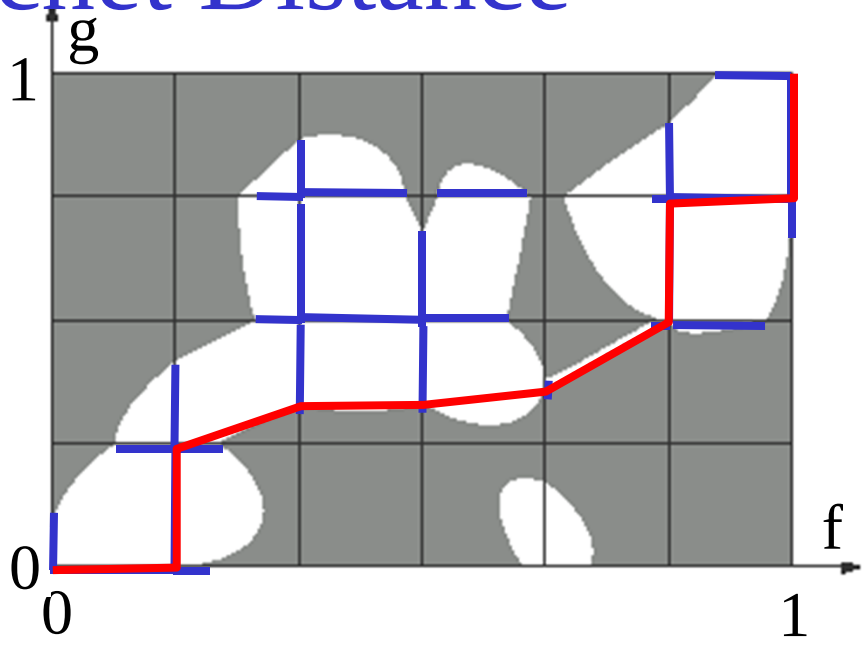
Free Space Diagram



- 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)

Compute the Fréchet Distance

- **Solve the decision problem**
  $\delta_F(f,g) \leq \varepsilon$ 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

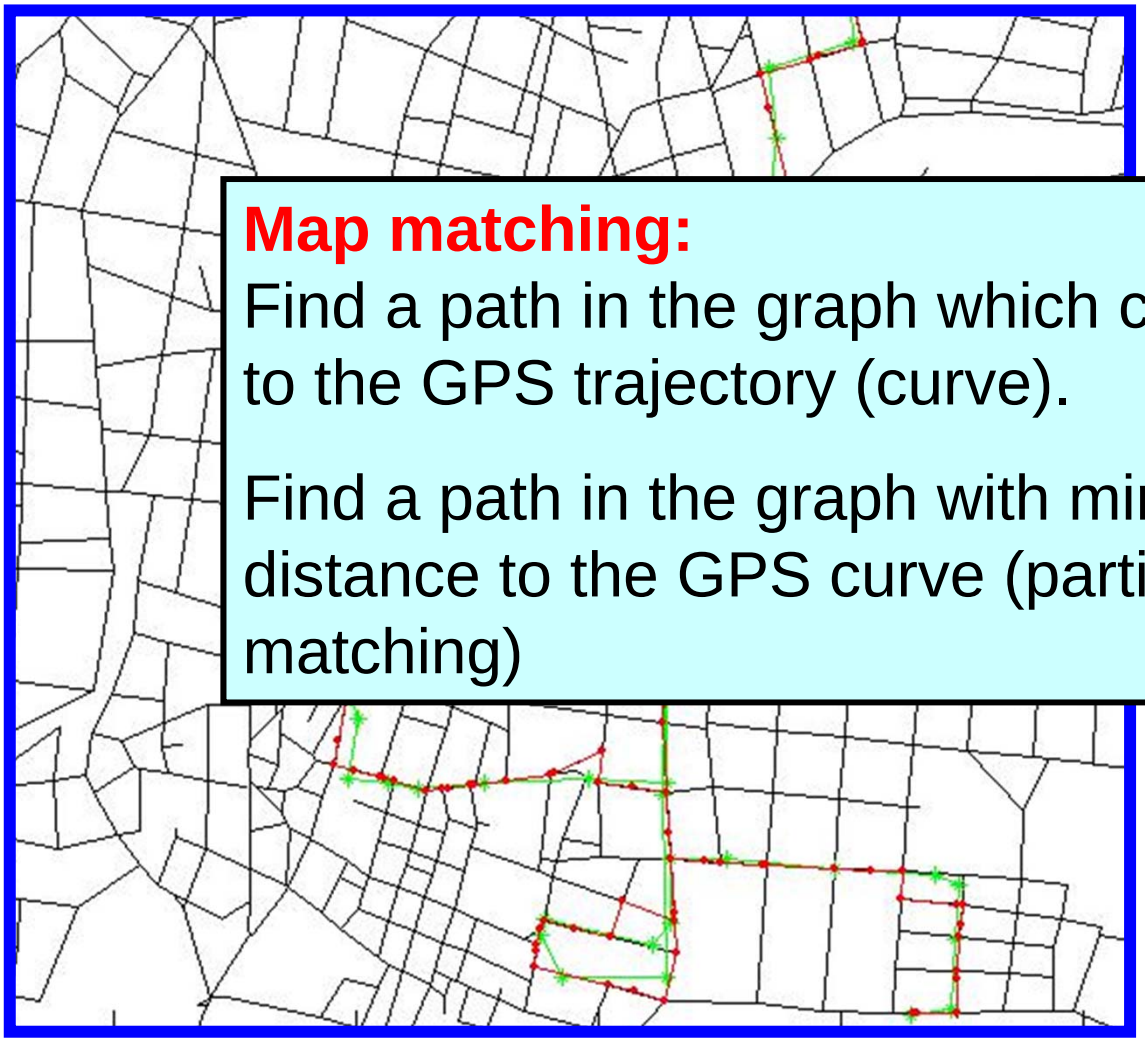
[AG95] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, *IJCGA* 5: 75-91, 1995.
[CW10] A.F. Cook IV, **C. Wenk**, Geodesic Fréchet Distance Inside a Simple Polygon, ACM TALG 7(1), 19 pages, 2010.

GPS Trajectories for Dynamic Routing

- Navigation systems answer shortest path queries based on travel times on road segments
- Usually based on static travel-time weights derived from speed limits
- **Goal:** Develop a navigation system which answers routing queries **based on the current traffic situation**

- **Current traffic situation:**
  - Database of current travel times per road segment
  - Use GPS trajectory data from vehicle fleets

GPS Trajectories from Vehicles



Map matching:
Find a path in the graph which corresponds to the GPS trajectory (curve).

Find a path in the graph with minimal distance to the GPS curve (partial matching)

1) Measurement error: GPS points do not lie d map

error: trajectory oduct and sampled every 30s

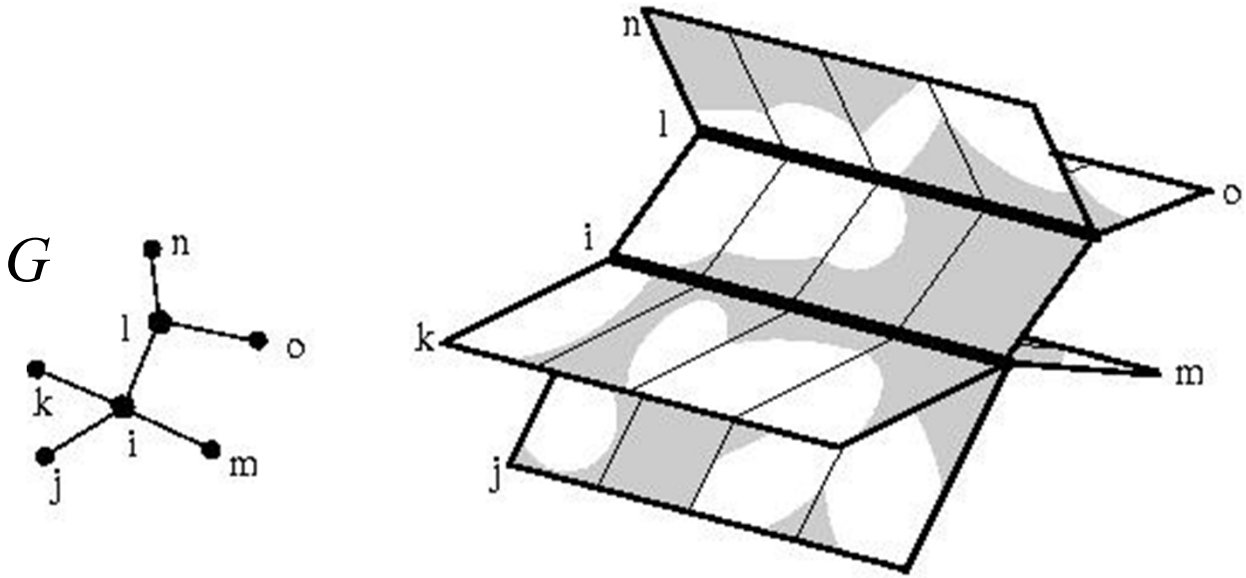
⇒ The GPS trajectory does not lie on the road map

Road map of Athens

GPS trajectory

Corresponding path in the road map

Free Space Surface

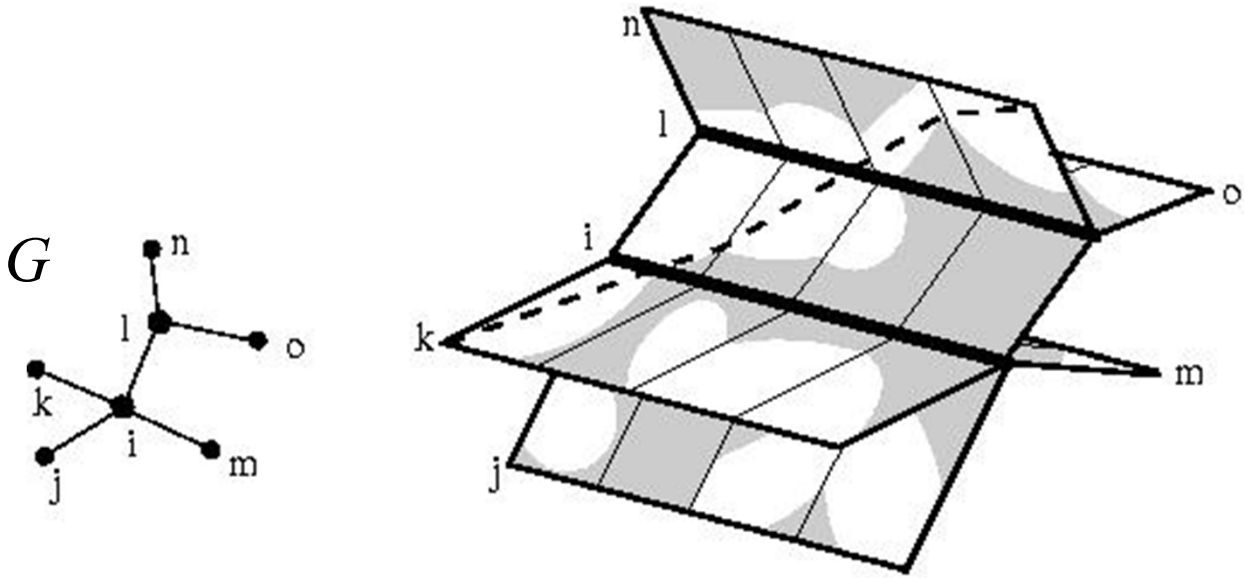


- Glue the free space diagrams $FD_{i,j}$ together according to adjacency information in *G*

→ **Free space surface of f and *G***

[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.

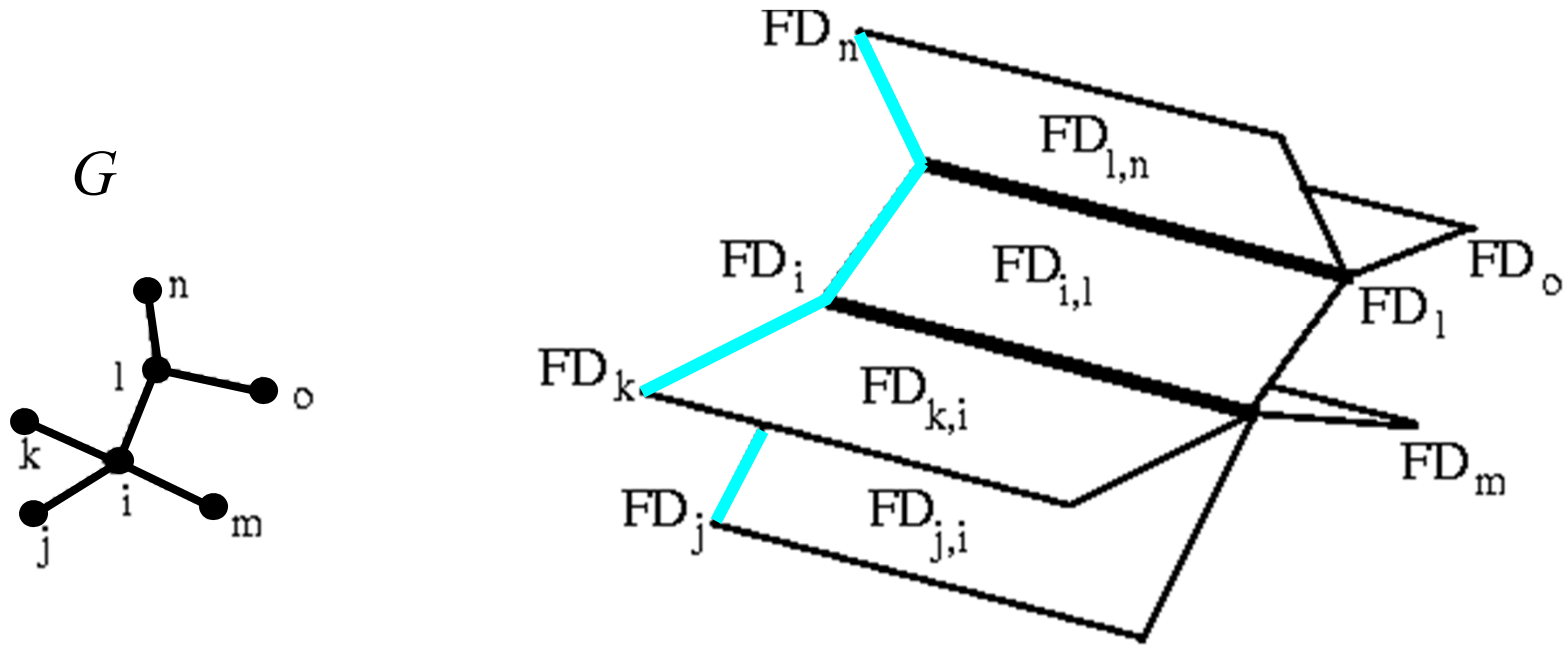
Free Space Surface



- **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.

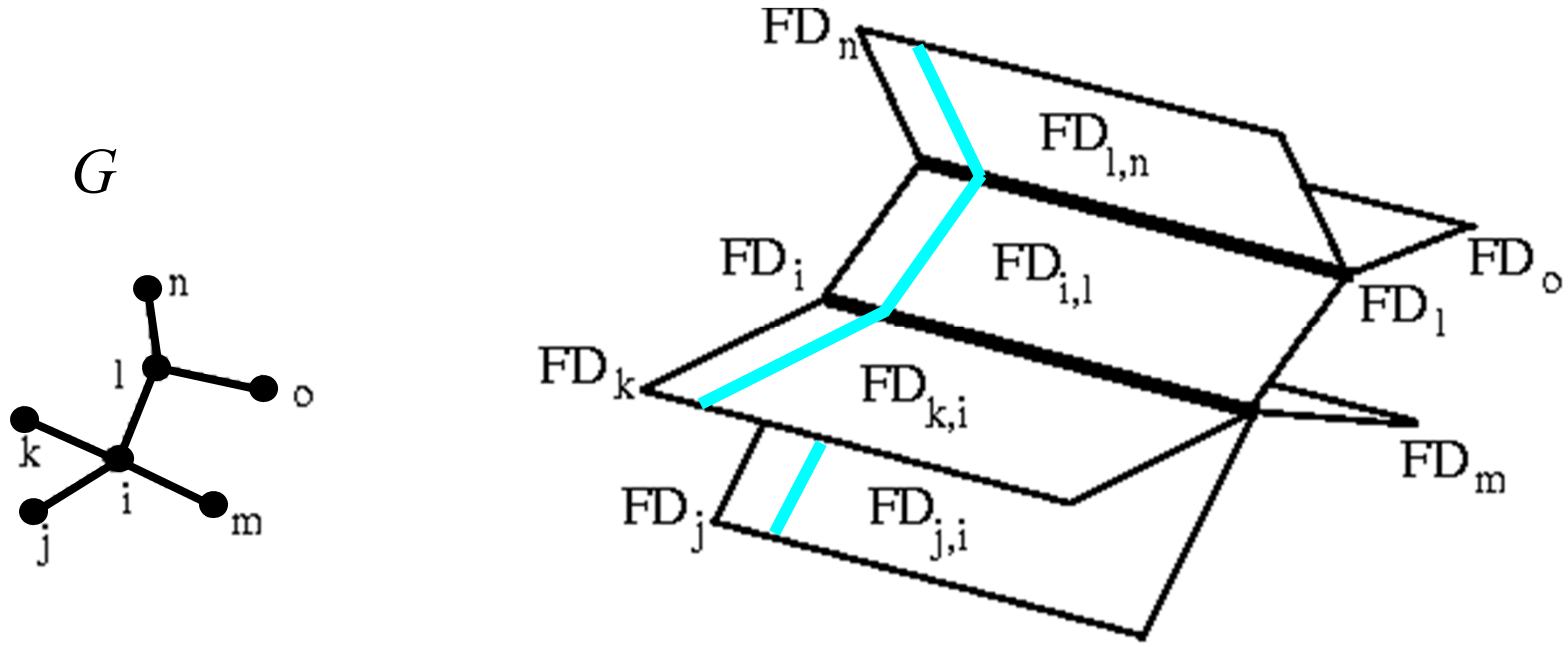
Sweep



- 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.

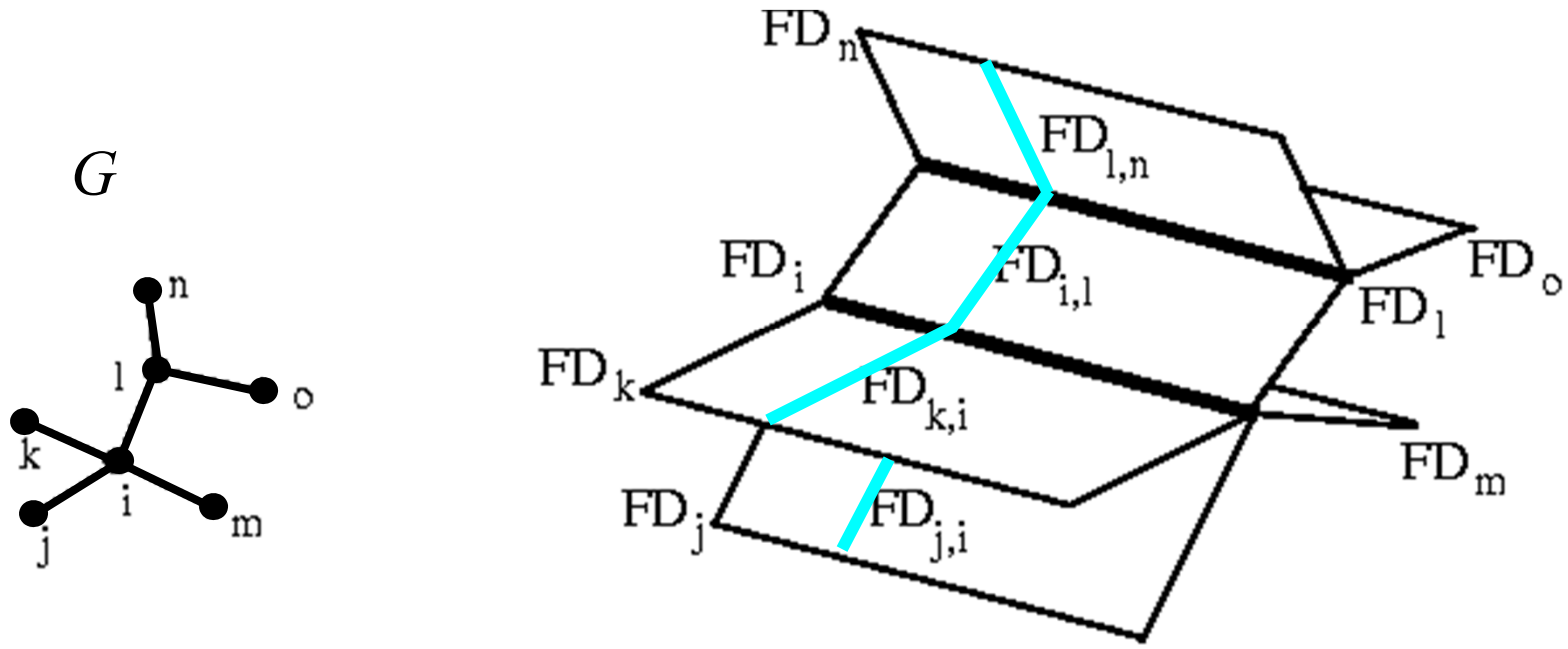
Sweep



- 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.

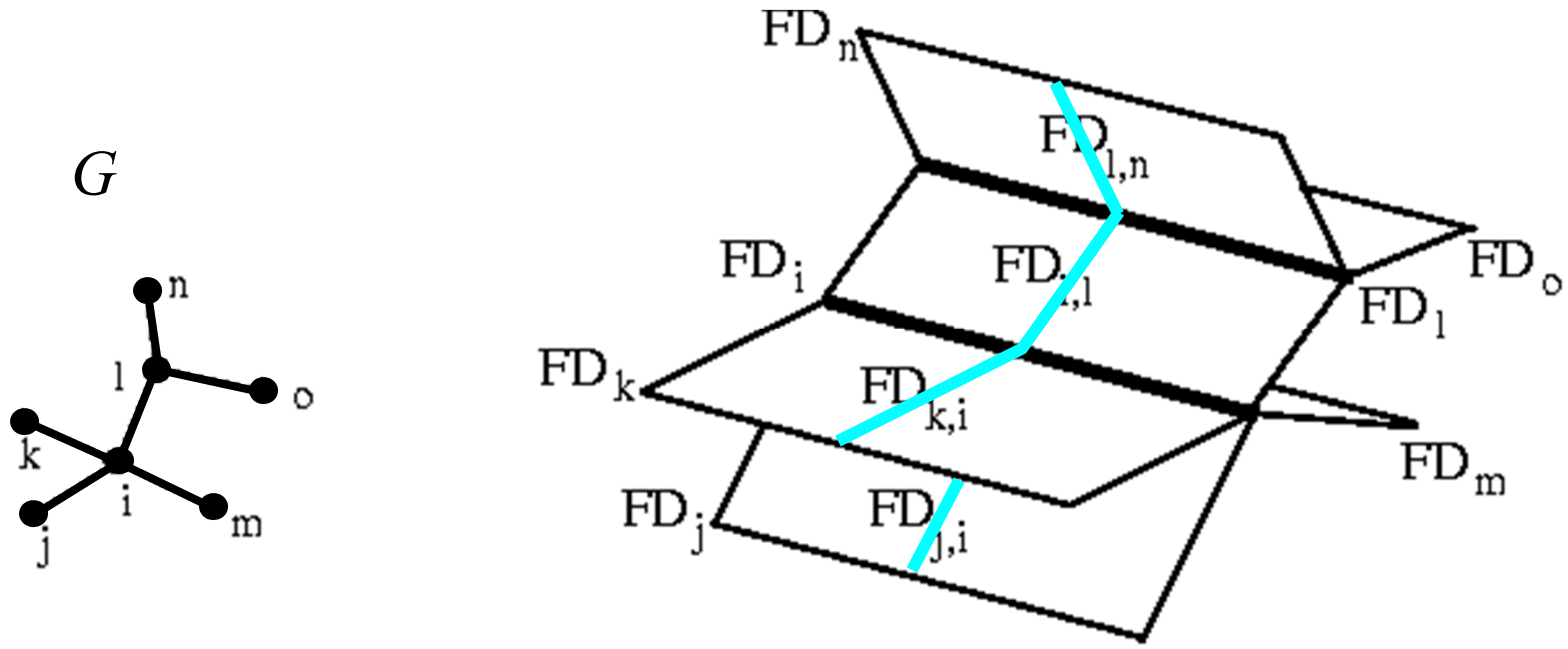
Sweep



- 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.

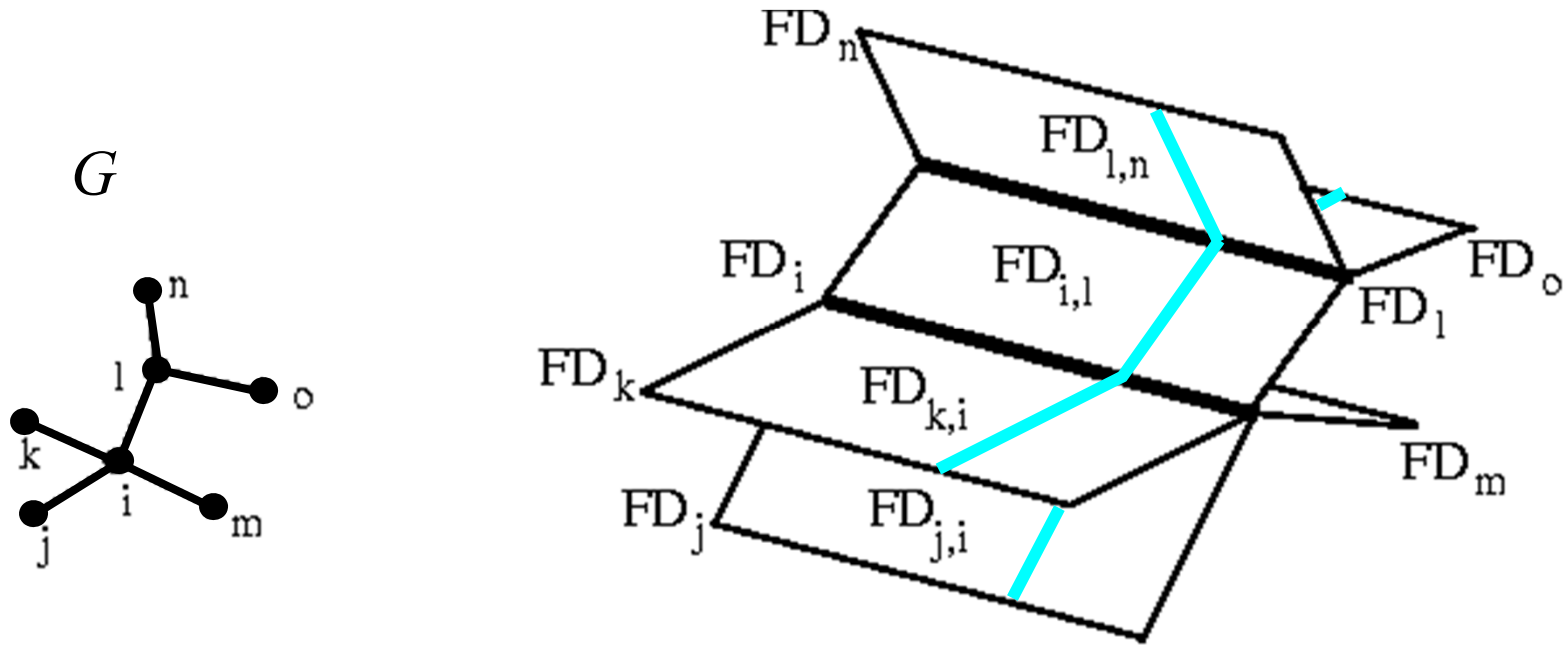
Sweep



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

[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.

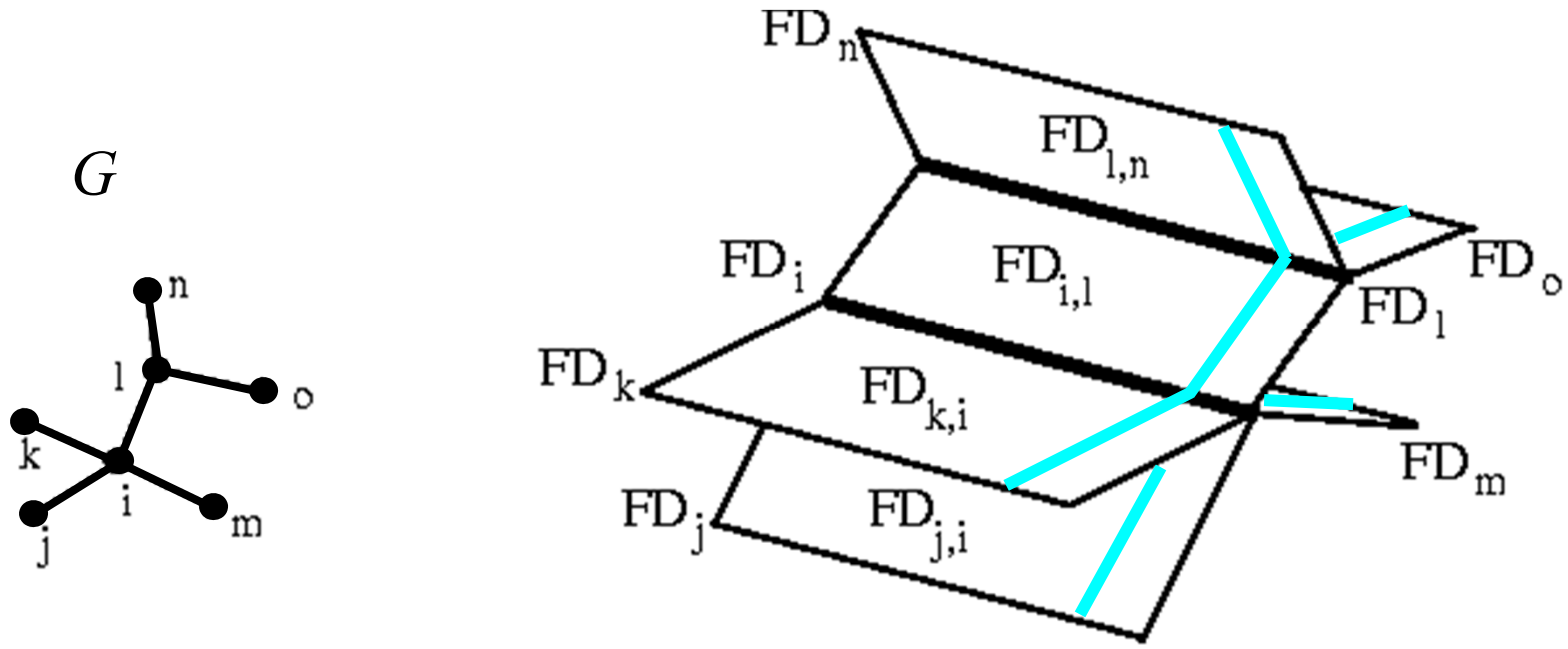
Sweep



- 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.

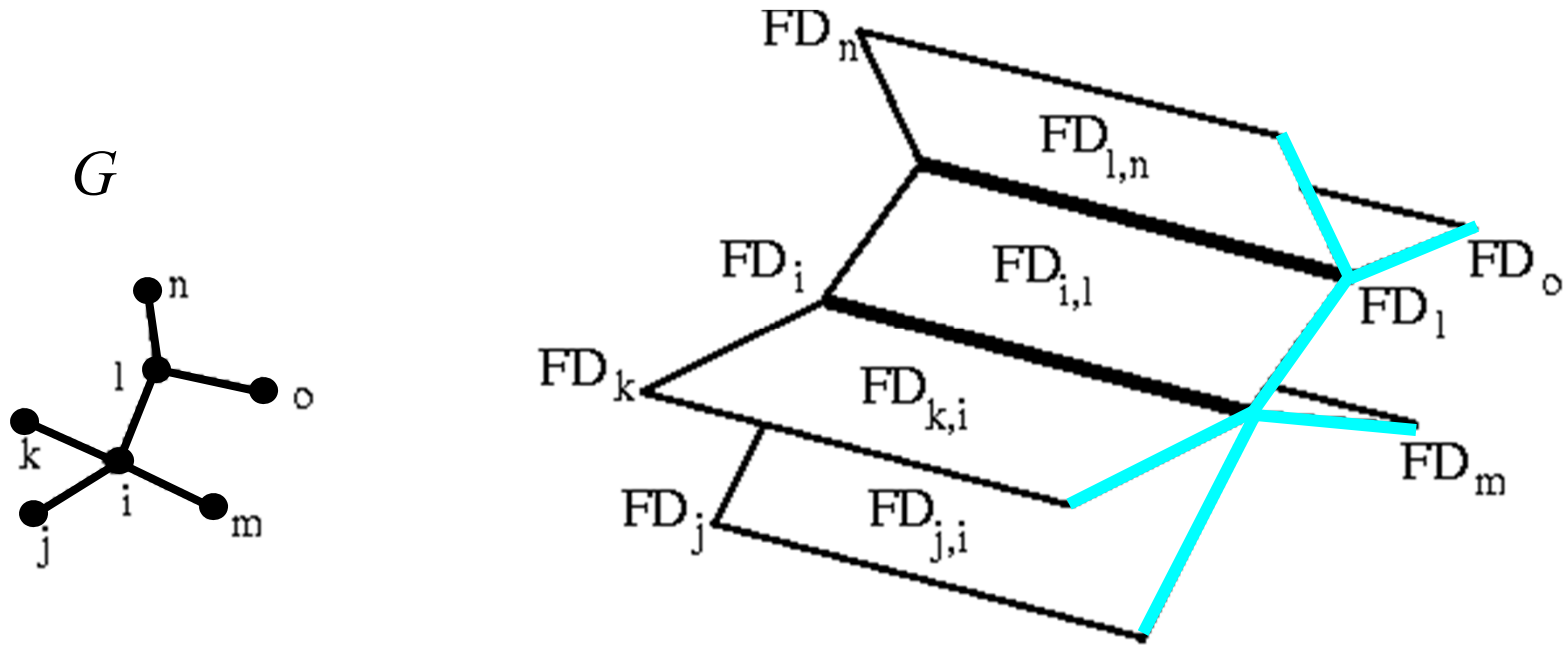
Sweep



- 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.

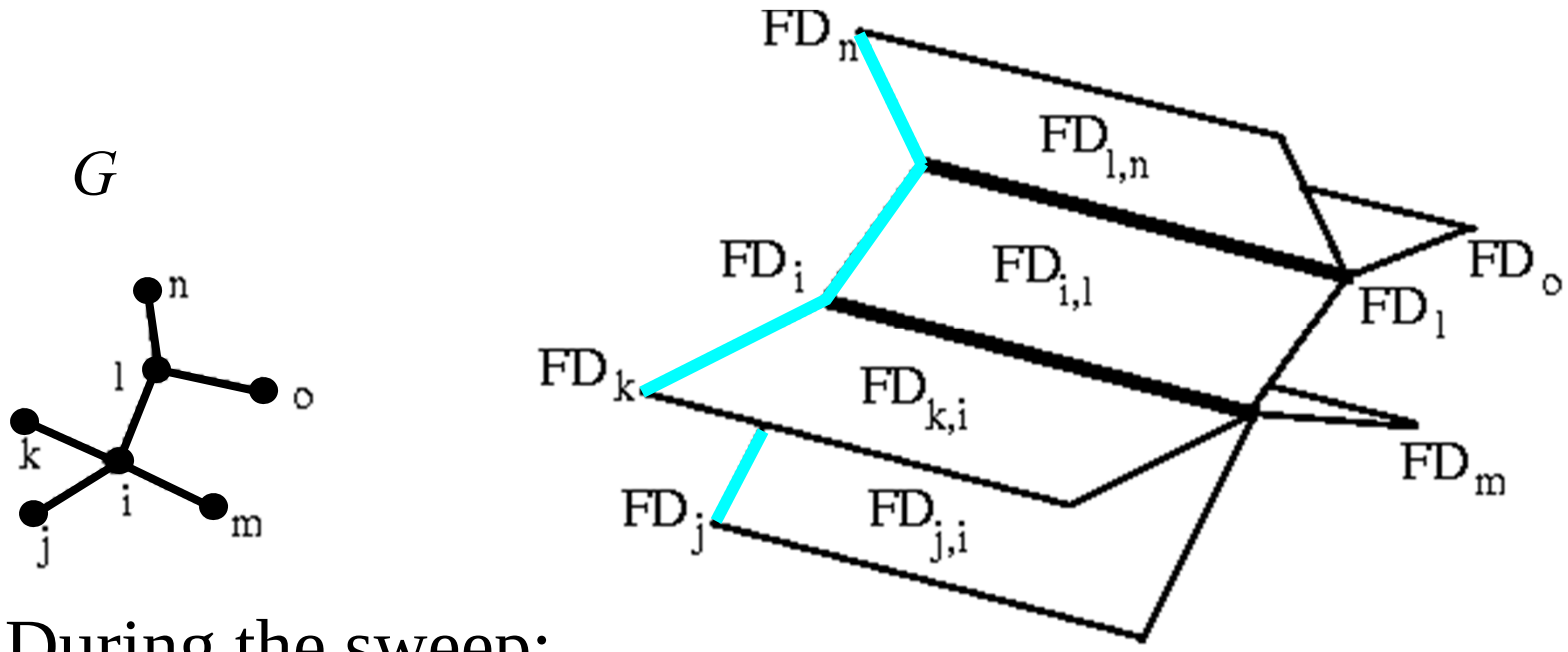
Sweep



- 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.

Compute Reachable Points



- During the sweep:

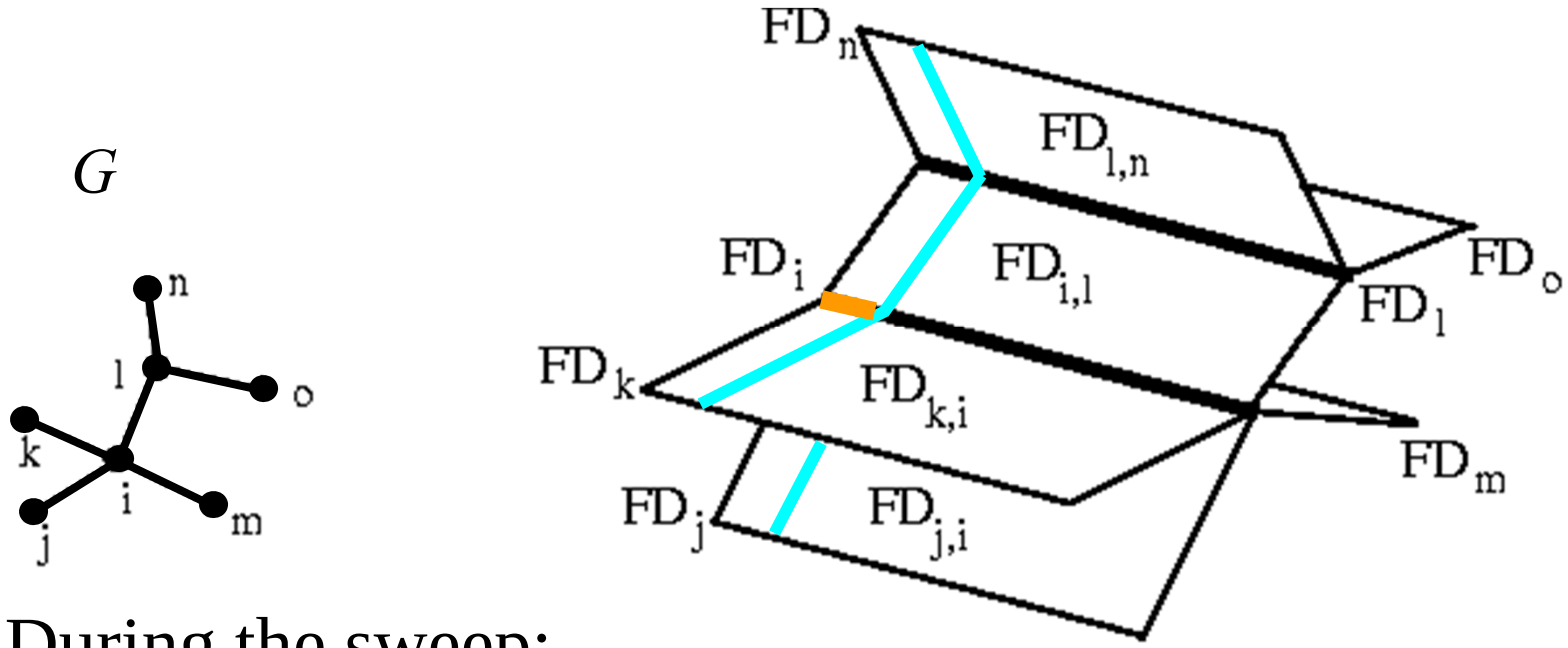
  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[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:

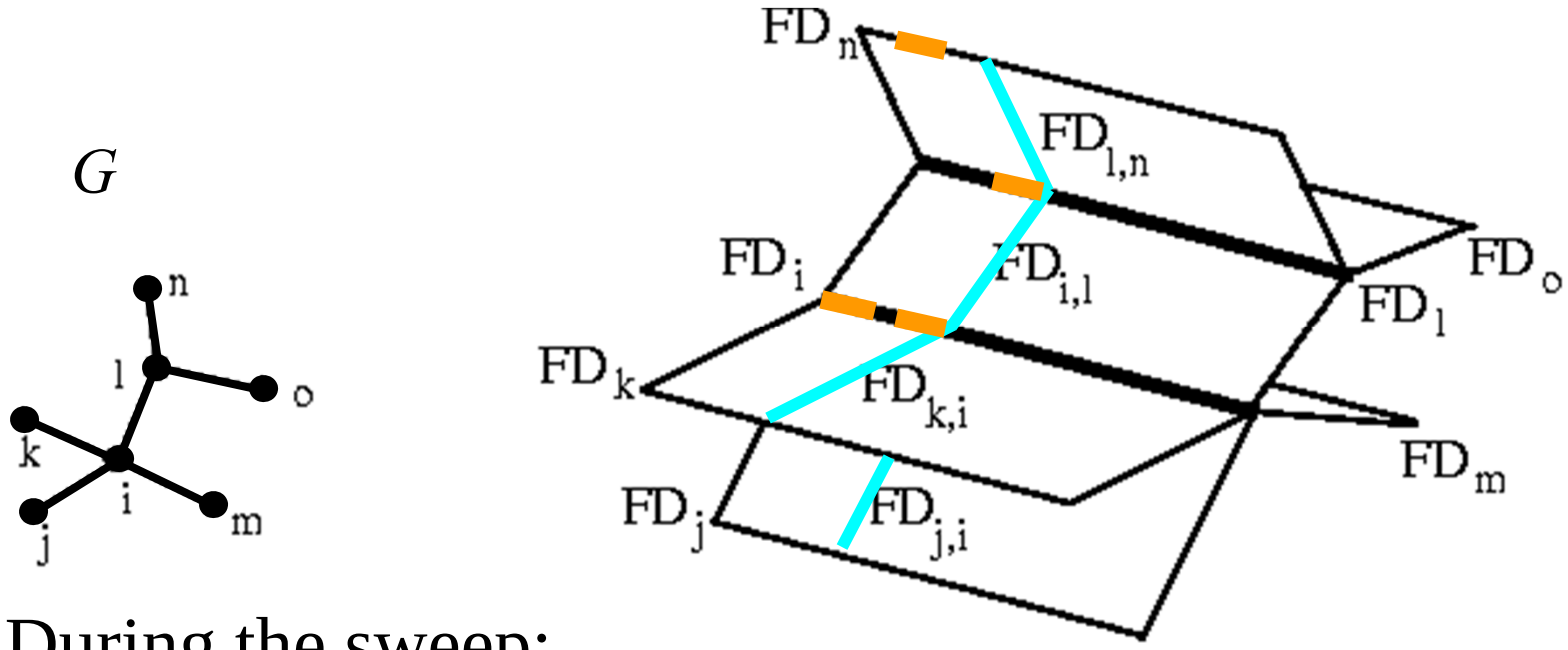
  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[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:

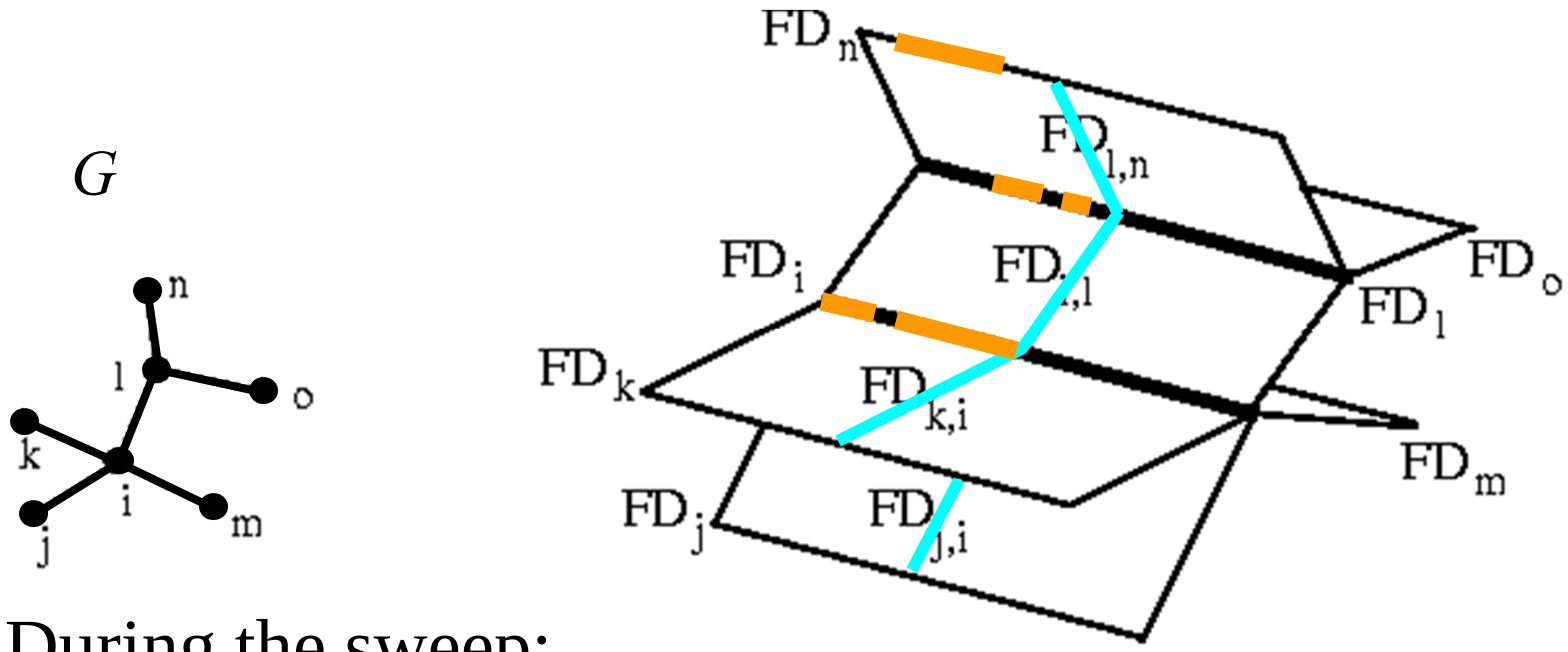
  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[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:

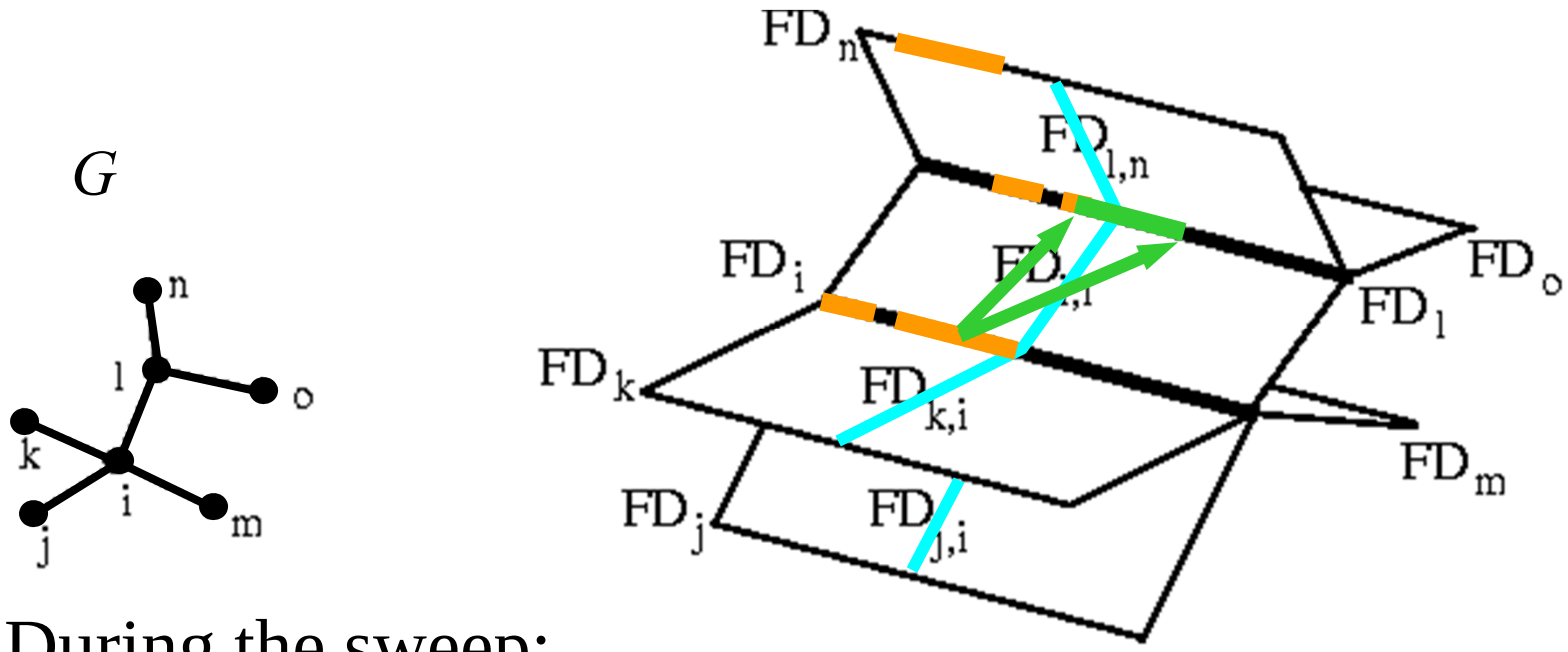
  Compute points on the free space surface, to the left of the sweep line, which are reachable by a monotone path from a lower left corner.

[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.

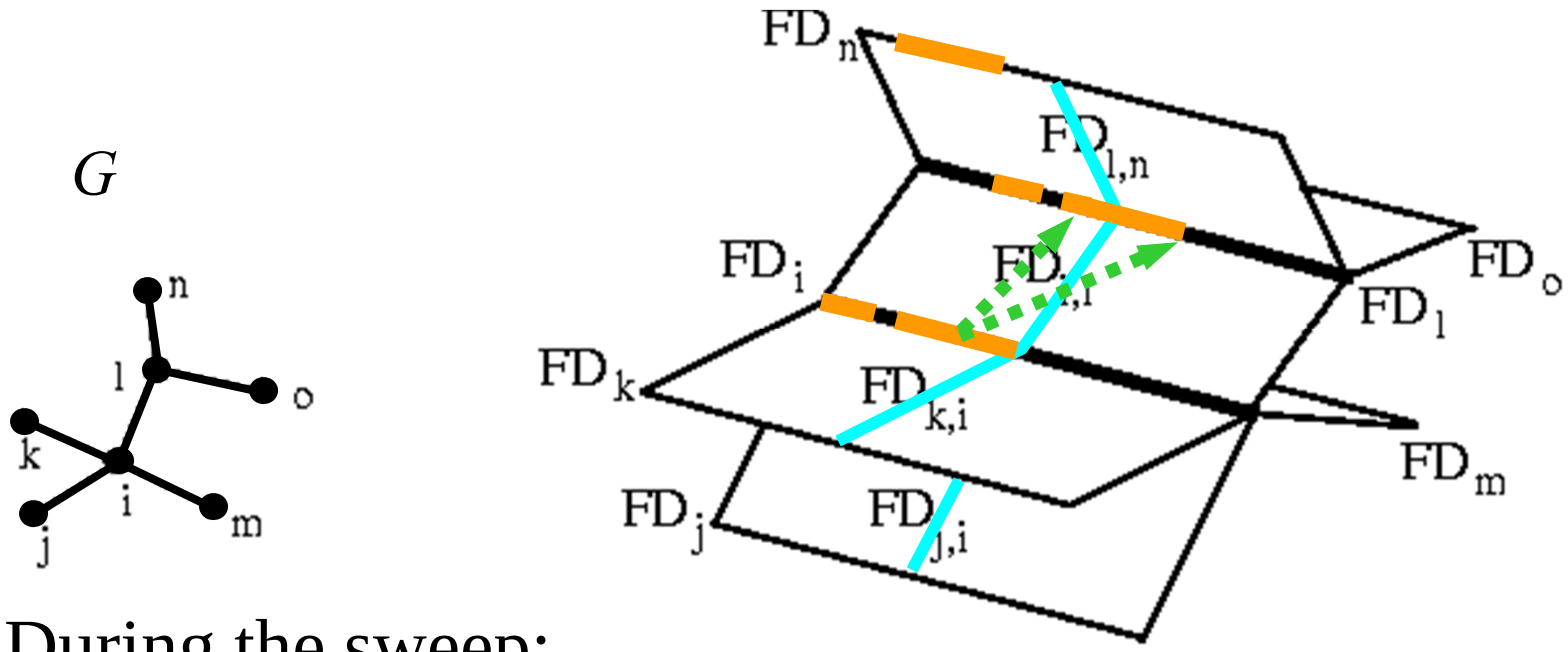
Update Reachable Points



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

[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



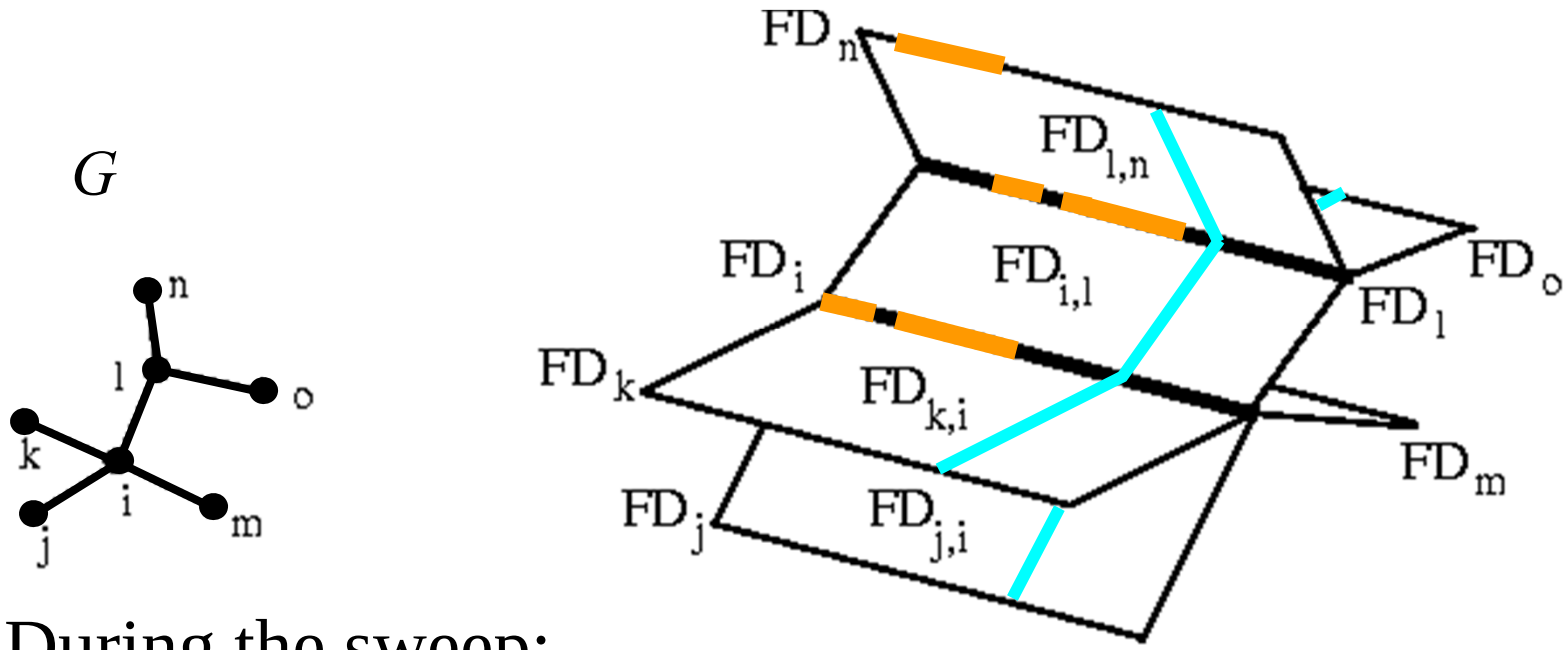
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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



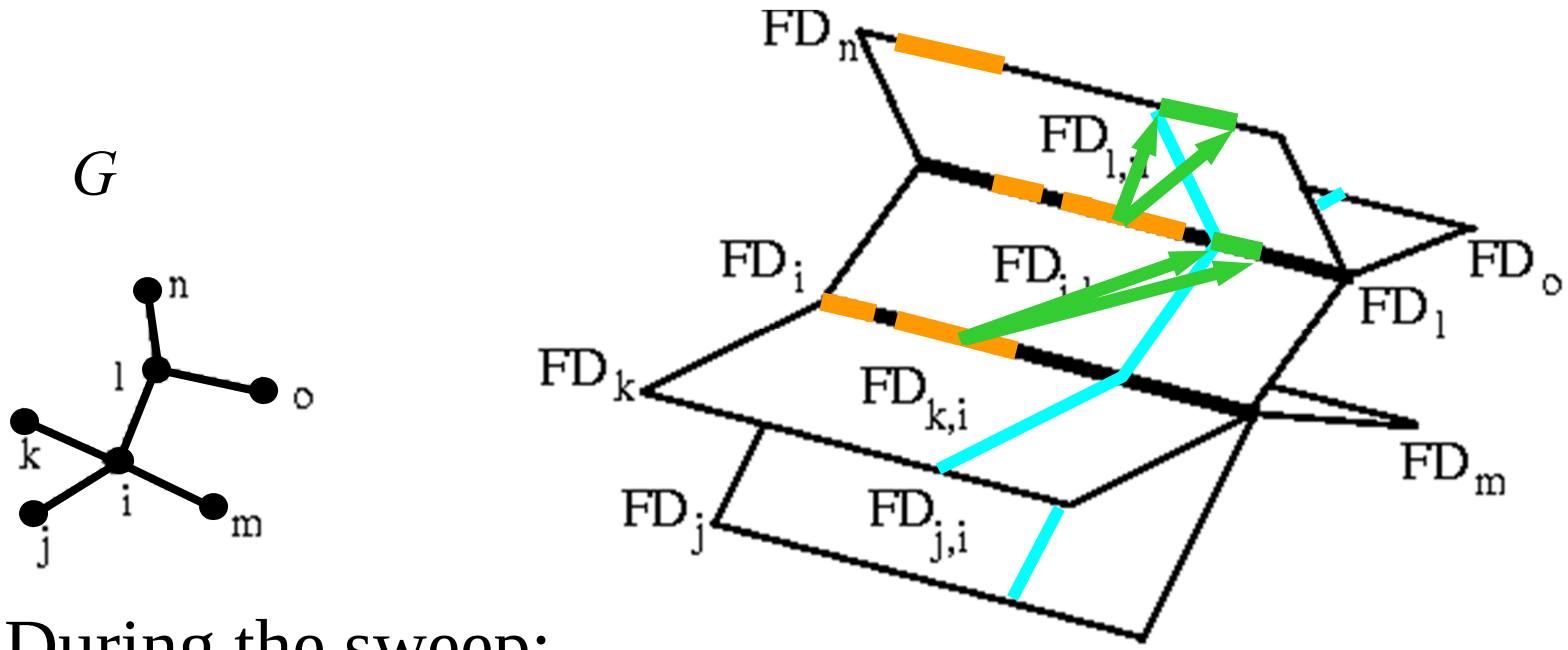
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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.

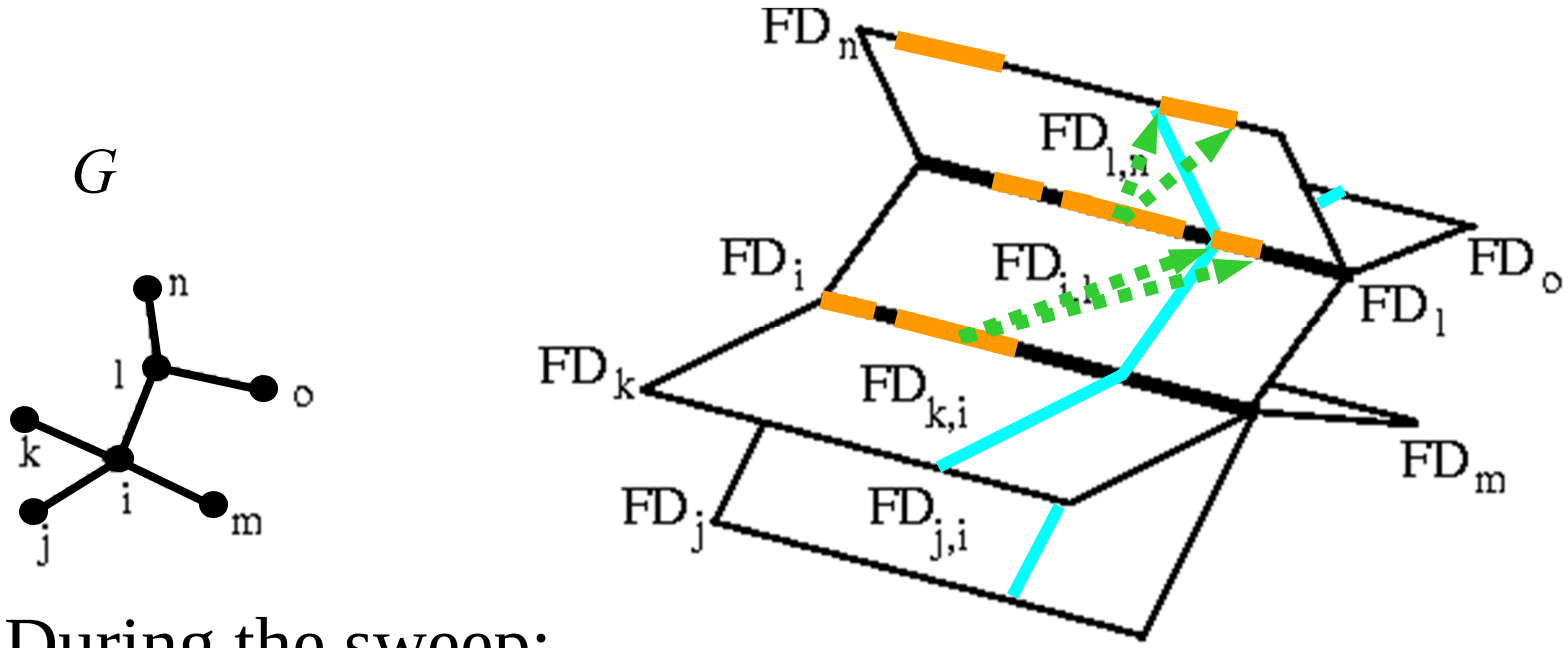
Update Reachable Points



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

[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



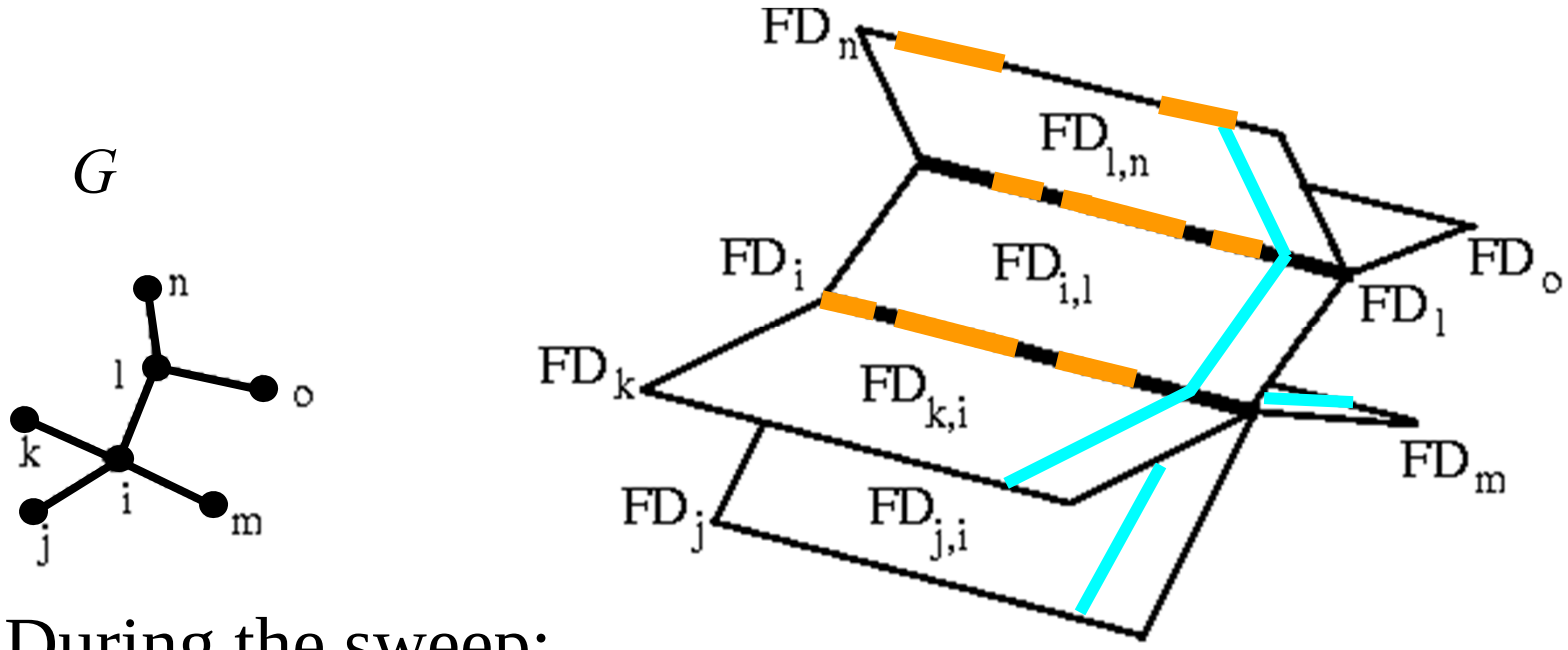
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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



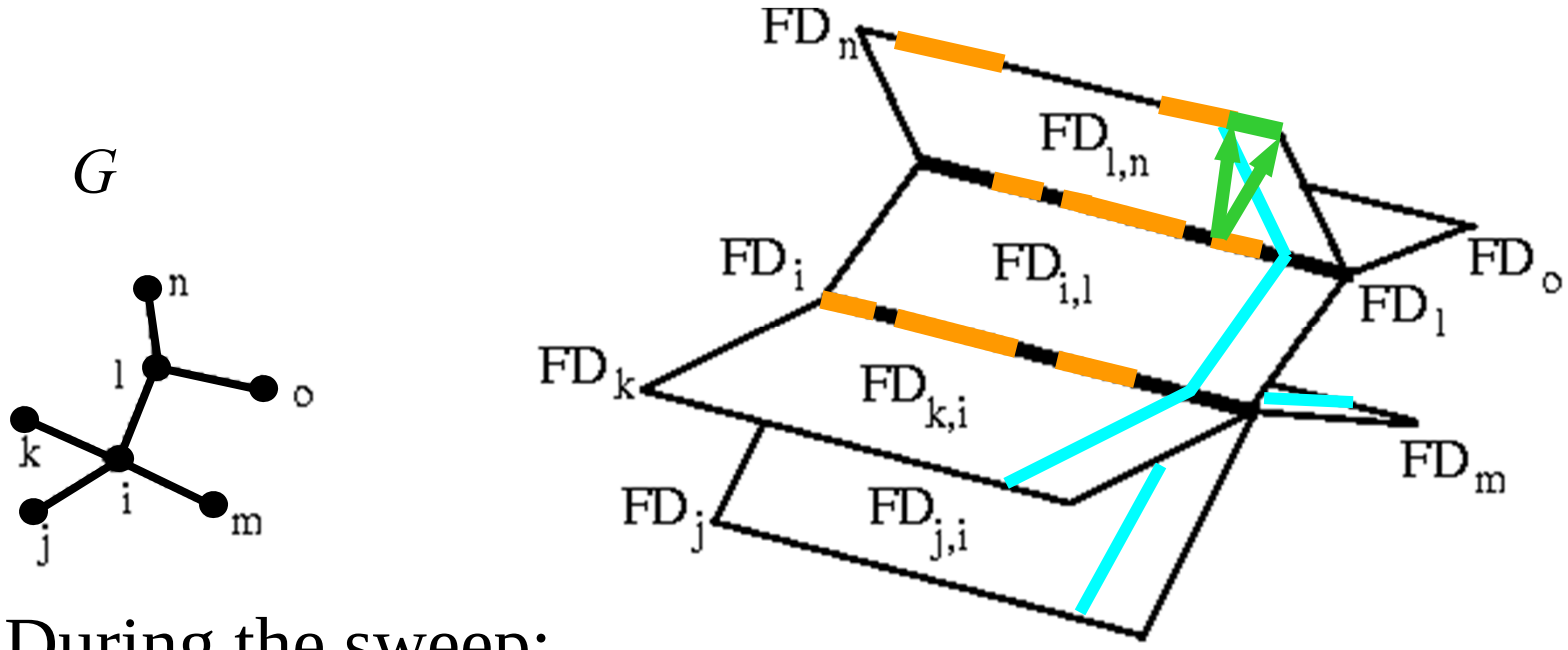
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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



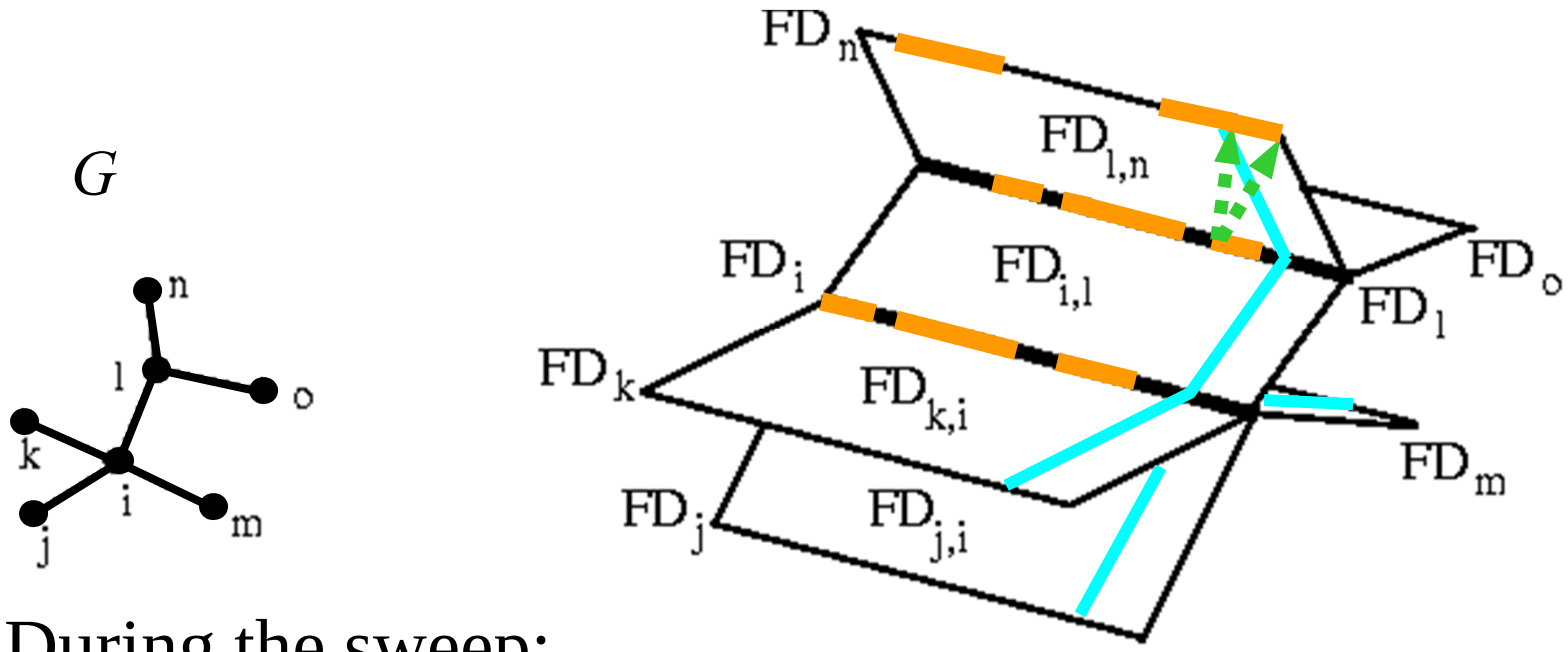
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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



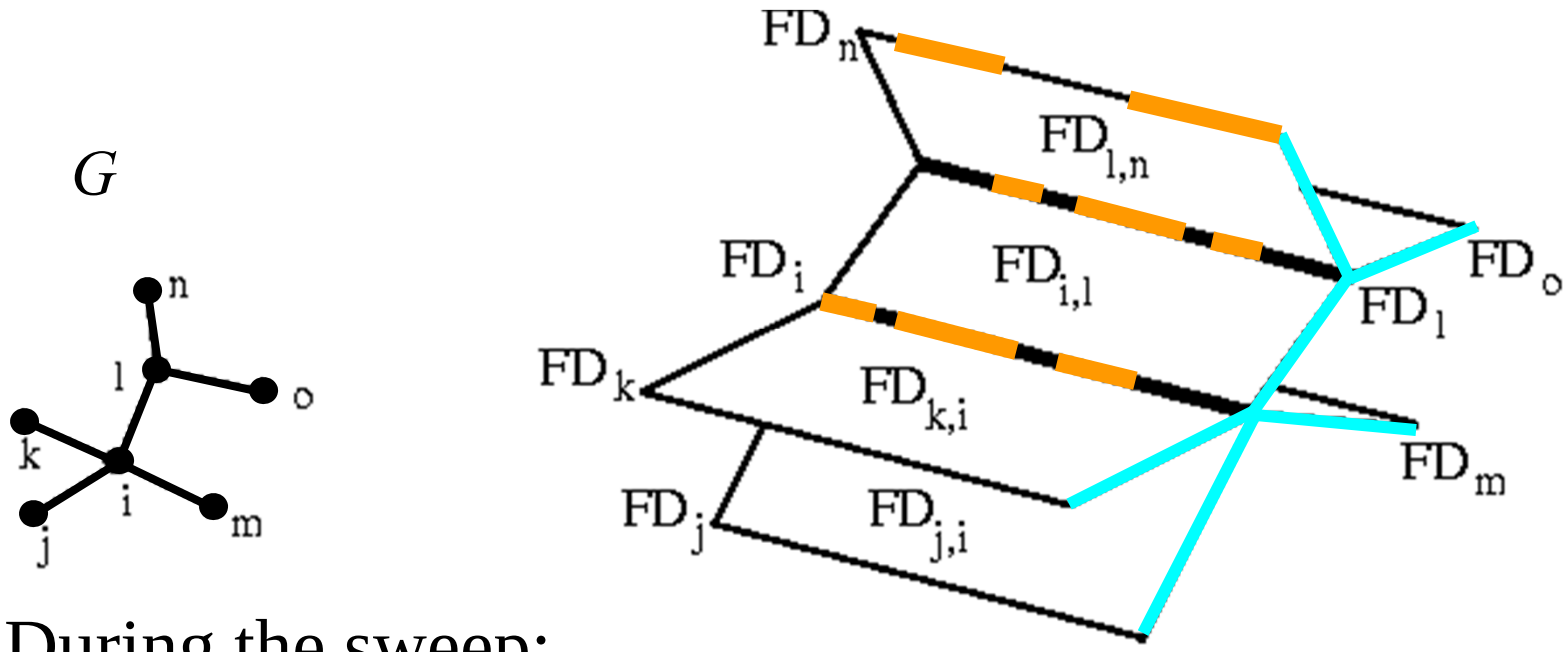
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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



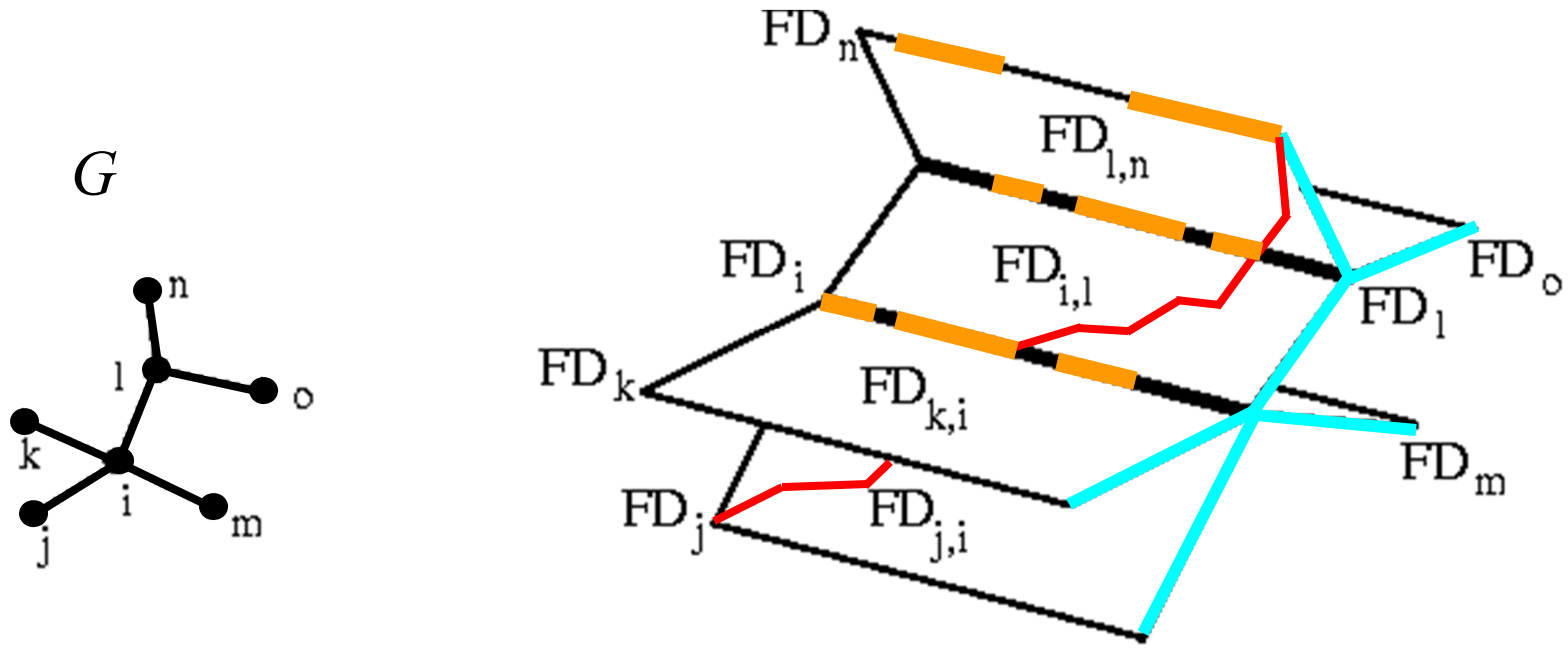
- During the sweep:
  - Update reachable points Dijkstra-style
  - Use a data structure which supports reachability queries in the free space surface

[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

[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.

Map-Matching

- Algorithm for decision problem takes O(mn log(mn)) time and O(mn) space.
- Optimization problem with parametric search: O(mn $\log^2$(mn)) time

[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.