Comparing Embedded and Immersed Graphs

Carola Wenk

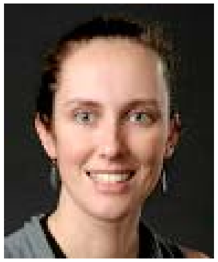
Department of Computer Science

Tulane University

Collaborators:

(Highly incomplete list)

Maike Buchin
U. Bochum



Brittany Fasy
Montana State U.

Erin Chambers
Saint Louis U.

Erfan Hosseini
Tulane U.

Pan Fan
Tulane U.

Majid Mirzanezhad
Tulane U.

Kevin Buchin
TU Eindhoven

Ellen Gasparovic
Union College

Dieter Pfoser
George Mason U.

Mahmuda Ahmed
Google

Rodrigo Silveira
U. Poli. de Catalunia

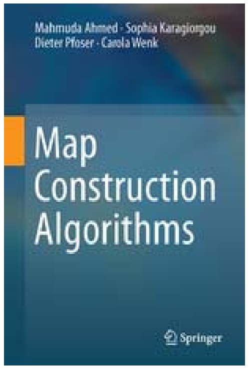
Liz Munch
Michigan State U.

Sophia Karagiorgou
Ubitech

Websites:

fv1wxodqh1hgx2£fdurod
p dsfrqvwuxfwlrq1ruj

Book:



Mahmuda Ahmed · Sophia Karagiorgou
Dieter Pfoser · Carola Wenk

Map Construction Algorithms

Springer

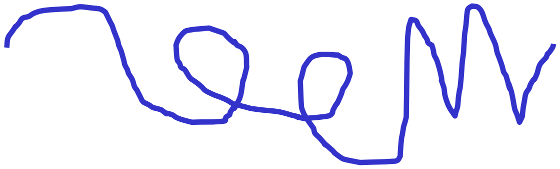
Grant support:

NSF CCF-1618469
and CCF-1637576

Outline

1. 1D embedded data: Curves and embedded & immersed graphs
2. Hausdorff and Fréchet-like distances:
    - Hausdorff distance
    - Fréchet distance
    - Path-based distance
    - Traversal distance
    - Strong/weak graph distance
    - Contour tree distance
3. Local persistent homology distance and local signatures
4. Other distances
    - Edit distance for geometric graphs
    - Shortest path sampling distance
    - Point sampling distance

1. 1D Embedded Data

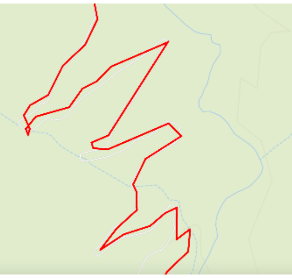


1D Embedded Data

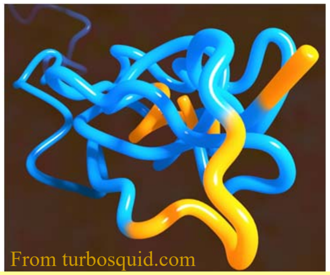
embedded in abient (usually Euclidean) space

C
u
r
v
e
s

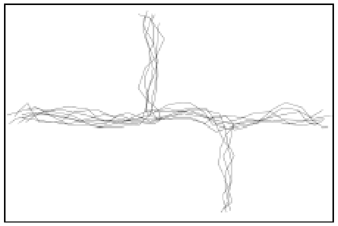
GPS trajectories



Protein chains

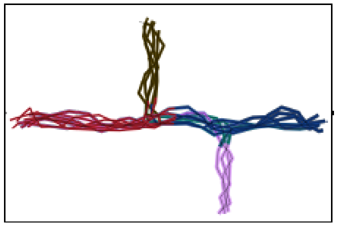

From turbosquid.com

Set of trajectories



⇒

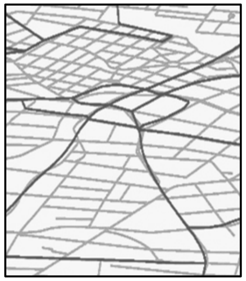
Sub-trajectory clusters



⇒Want to
order to fi

- Want to compare such 1D embedded data ⇒ Geometric shapes
- There are lots of distance measures and algorithms for comparing curves, and some for trees. But not so many for embedded (geometric) graphs.
- Graphs are the most general 1D shapes.

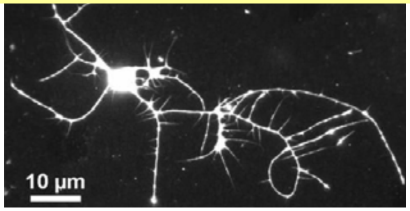
⇒



Constructed roadmap

T
r
e
e
s

Plant mor

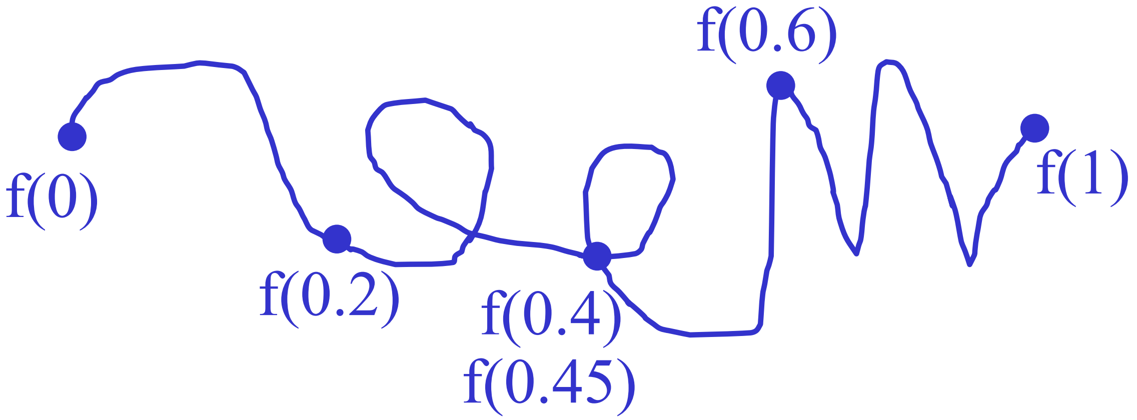


r
a
p
h
s

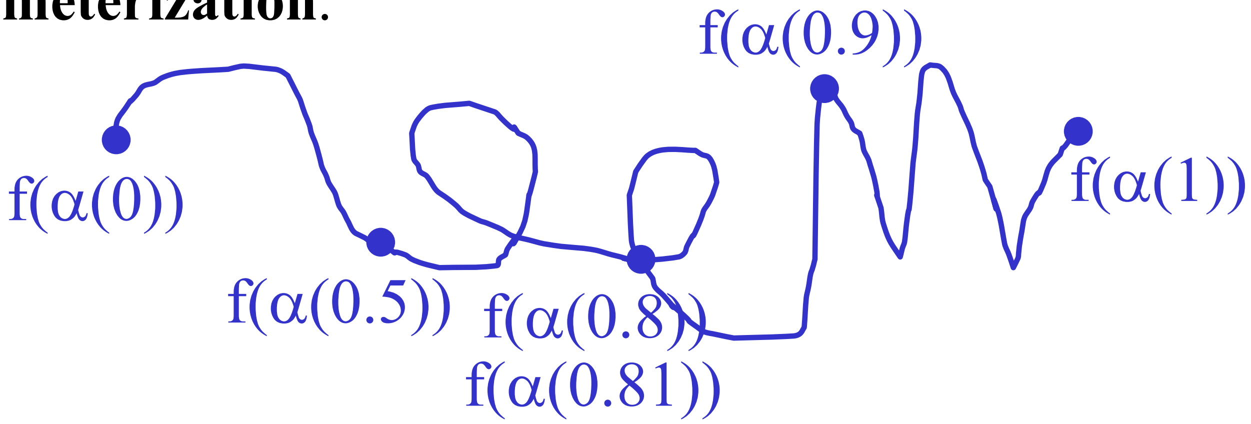
Roadmap comparison

Curves

- A **curve** is a continuous map $f:[0,1] \to \mathbb{R}^d$

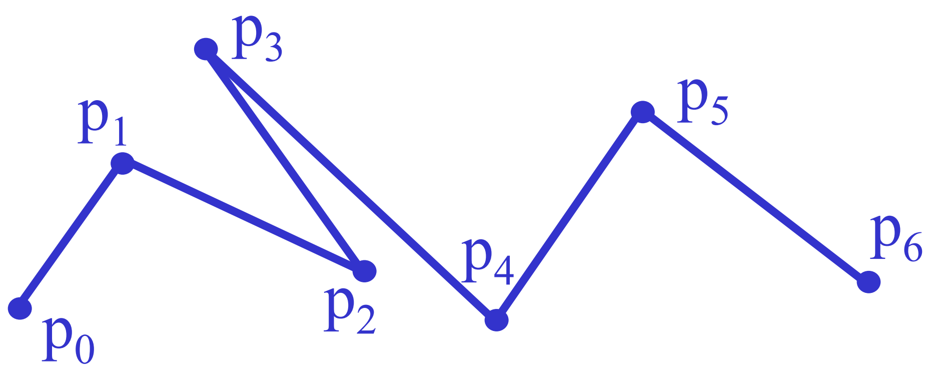


- Many different curves can have the same image.
- We can reparameterize curves: $f \circ \alpha: [0,1] \to \mathbb{R}^d$, where $\alpha: [0,1] \to [0,1]$ is a **reparameterization**.



Polygonal Curves & Trajectories

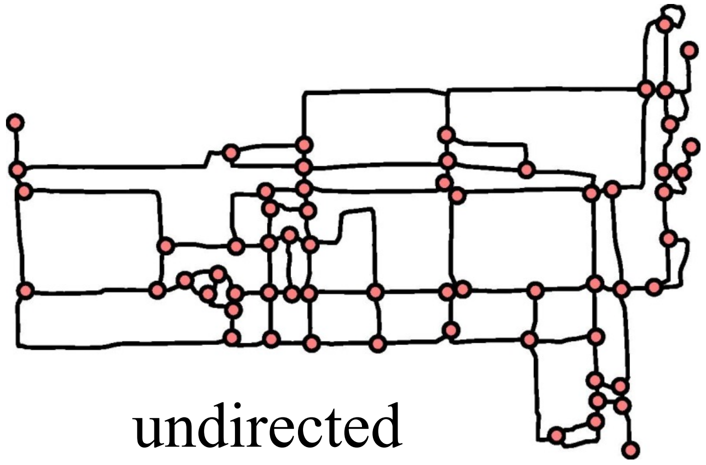
- Polygonal curves consist of a finite number of line segments and vertices. They can be specified by a sequence of points $<p_0,\ldots, p_{n-1}>$



- We typically endow a polygonal curve with its arc-length parameterization $f:[0,1] \rightarrow \mathbb{R}^d$. On each edge $p_i p_{i+1}$ this is a linear function, hence a piecewise linear function overall.
- A (geospatial) trajectory is a sequence of time-stamped position samples.

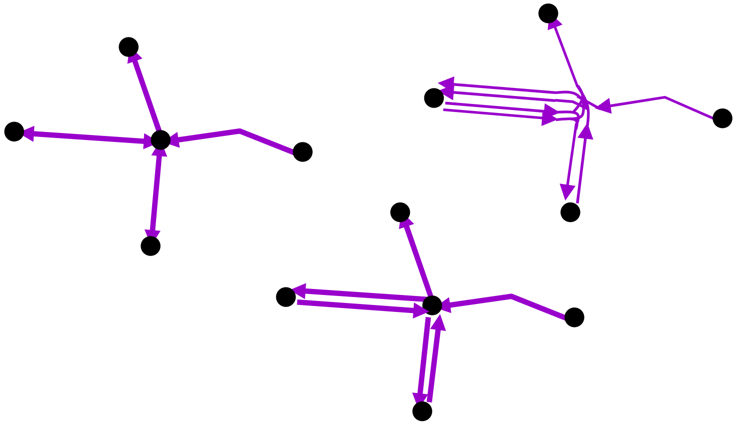
Embedded/Immersed Graphs

- Graph $G = (V, E)$ with a set of vertices V and edges E.
- Road network: Planar embedded



undirected

directed



- Can consider G as a topological space (e.g., 1D simplicial complex)
- **Embedded graph**: Have a continuous function $\phi: G \to \mathbb{R}^d$, $d \geq 2$, that is homeomorphic onto its image.
- **Immersed graph**: $\phi: G \to \mathbb{R}^d$ is only **locally** homeomorphic onto its image.

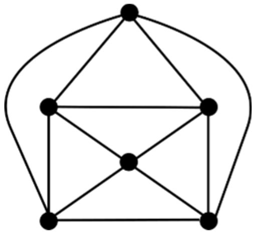
Embedded/Immersed Graphs

- **Embedded graph**: Have a continuous function $\phi: G \rightarrow \mathbb{R}^d$, $d \geq 2$, that is homeomorphic onto its image.
- **Immersed graph**: $\phi: G \rightarrow \mathbb{R}^d$ is only **locally** homeomorphic onto its image.

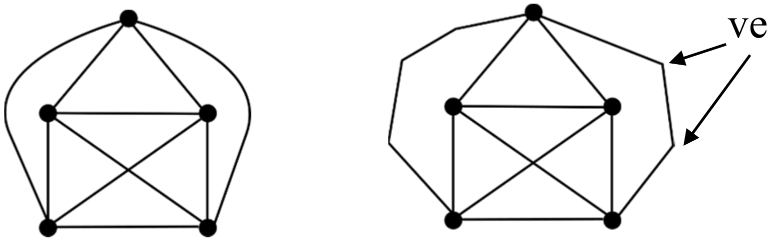
=> Each vertex is mapped to a point and edges are mapped to curves in $\mathbb{R}^d$ in such a way that the graph structure is maintained.

- Homeomorphism: A continuous, bijective map whose inverse is continuous.

Embedding:
all edge-curves are non-crossing
(every crossing is a vertex)



Immersion:
"Bridges" are allowed

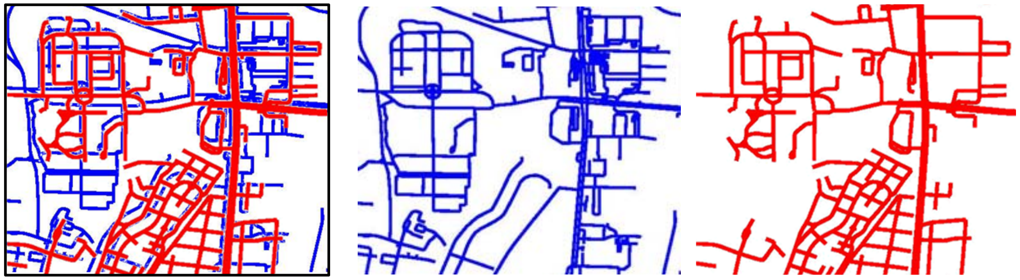


- $\mathbb{R}^2$: planar graphs vs. plane (= planar embedded) graphs
- Assume edge curves are piecewise linear, and may ignore deg-2 vertices

Immersed Graph Comparison

Given two immersed graphs $G = (V_G, E_G, \phi_G)$ and H $= (V_H, E_H, \phi_H)$, we want to compare them.

- How similar / different are they?
- What does it mean to be similar?
  - Depends on the application.
  - Graph isomorphism?



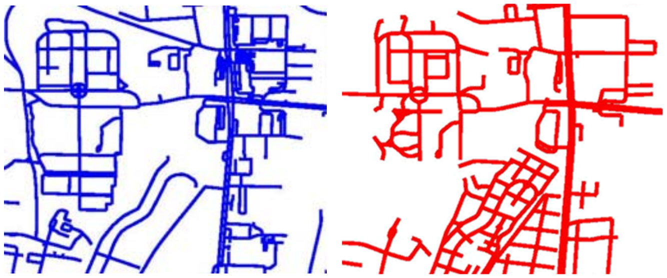
- Here: Assume G and H are embedded in the same space and aligned.

1. Define different distances between G and H, and study their properties and computational complexities.
2. Compute correspondences between portions of G and H.
3. Consider local distance signatures (heatmaps).

Graph Isomorphism

- An isomorphism of $G = (V_G, E_G)$ and H $= (V_H, E_H)$ is a
  - bijective map $f: V_G \to V_H$ for which holds
  - $\{u, v\} \in E_G \Leftrightarrow \{f(u), f(v)\} \in E_H$

  Can be computed in linear time for planar graphs [HW74]
- Subgraph isomorphism: An isomorphism between G and a subgraph of H
  - NP-complete
  - Can be computed in linear time if G and H are planar and G has constant complexity [E95]



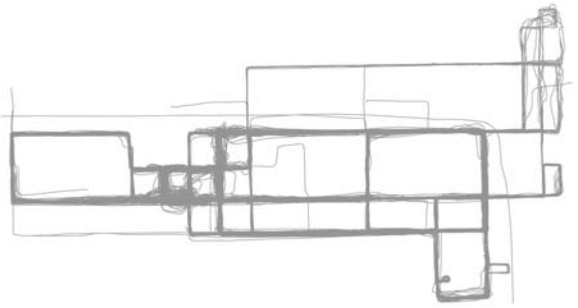
- Isomorphisms are bijective (1-to-1). However, we may want to allow 1-to-many assignments.
- We may also want to allow partial matchings.
- Isomorphisms are combinatorial in nature and don't take the embeddings/immersions into account.

[E95] D. Eppstein, Subgraph isomorphism in planar graphs and related problems, SODA: 632–640, 1995.

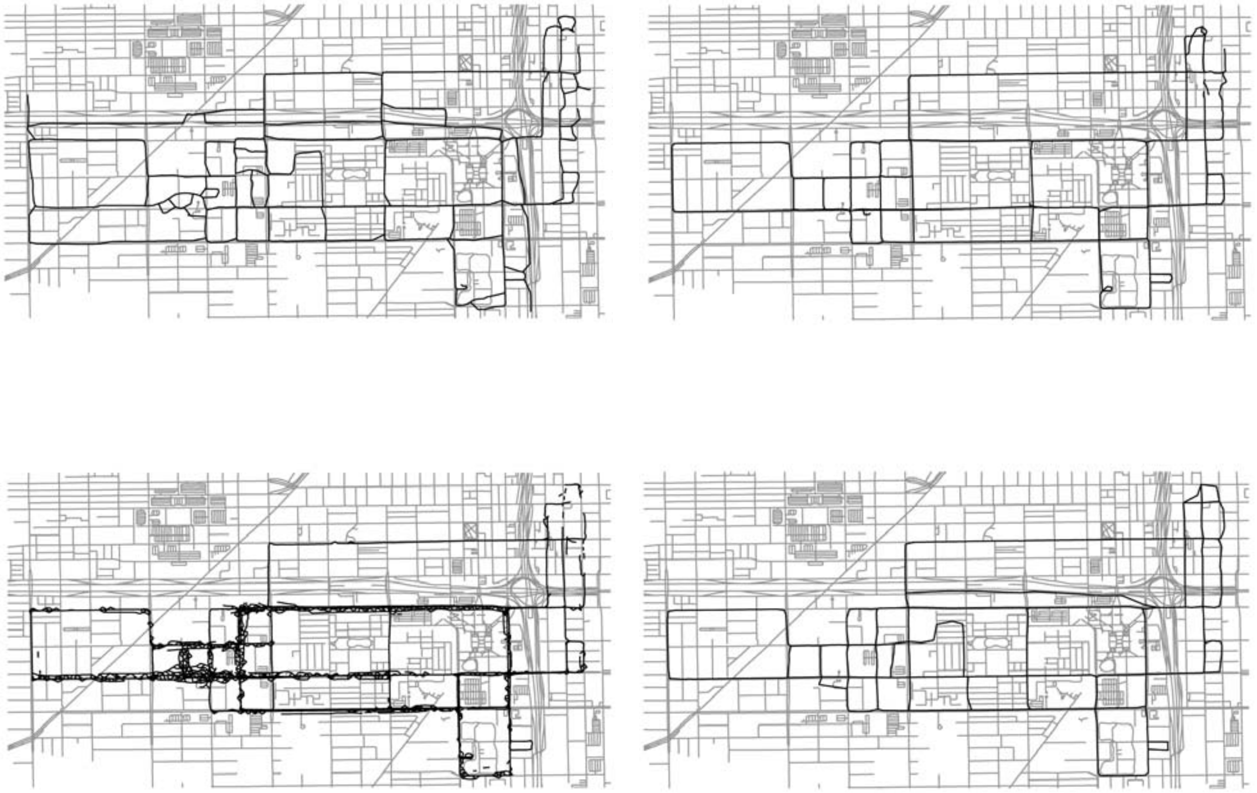
[HW74] J. Hopcroft, J. Wong, Linear time algorithm for isomorphism of planar graphs, STOC: 172–184, 1974.

Compare Reconstructed Roadmaps

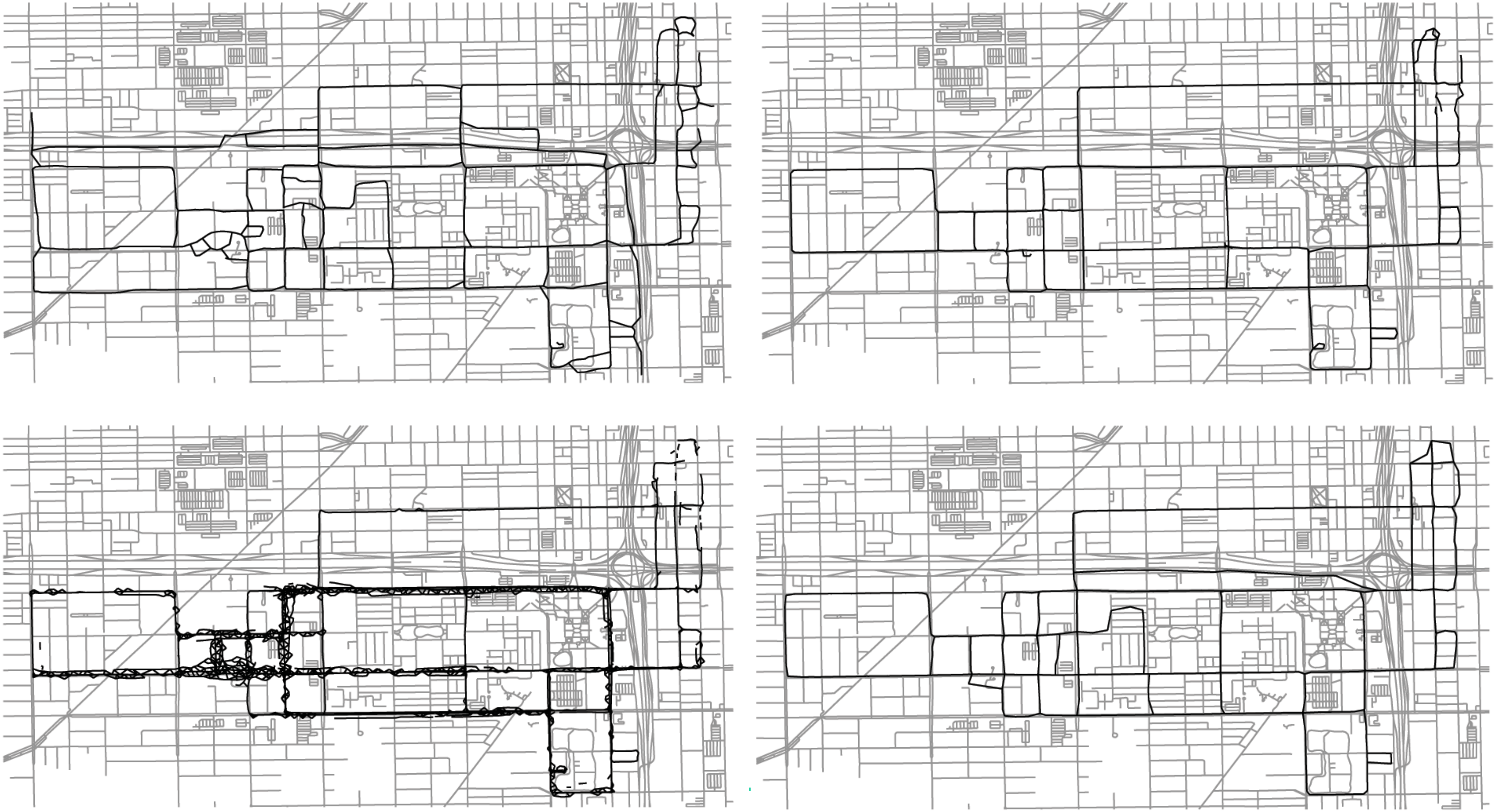
GPS Trajectory Data



Reconstructed Roadmaps



Compare Reconstructed Roadmaps

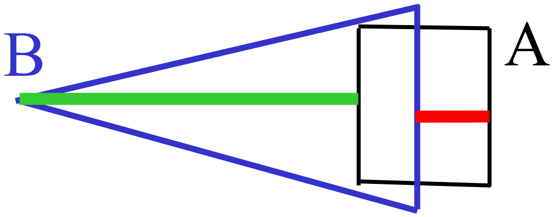


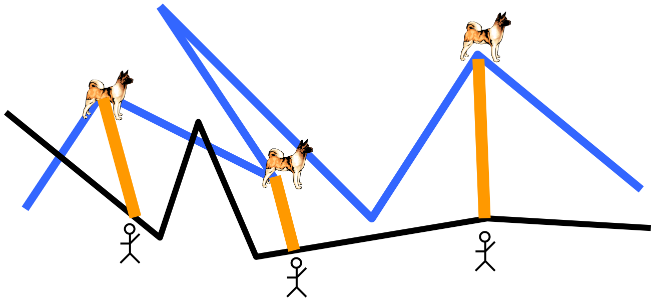
Compare Reconstructed Roadmaps

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

⇒ Compare two immersed graphs

2. Hausdorff and Fréchet-Like Distances

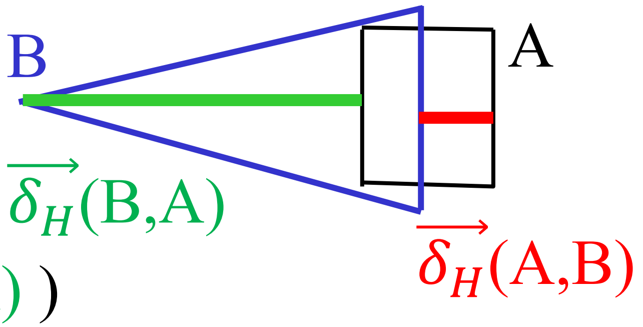




Hausdorff Distance

- Directed Hausdorff distance

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

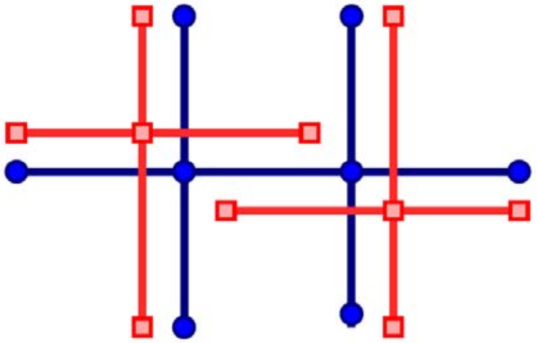

B
A
$\overrightarrow{\delta_H}(B,A)$
$\overrightarrow{\delta_H}(A,B)$

- Undirected Hausdorff-distance

$$\delta_H(A,B) = \max\left(\overrightarrow{\delta_H}(A,B)\ ,\ \overrightarrow{\delta_H}(B,A)\right)$$

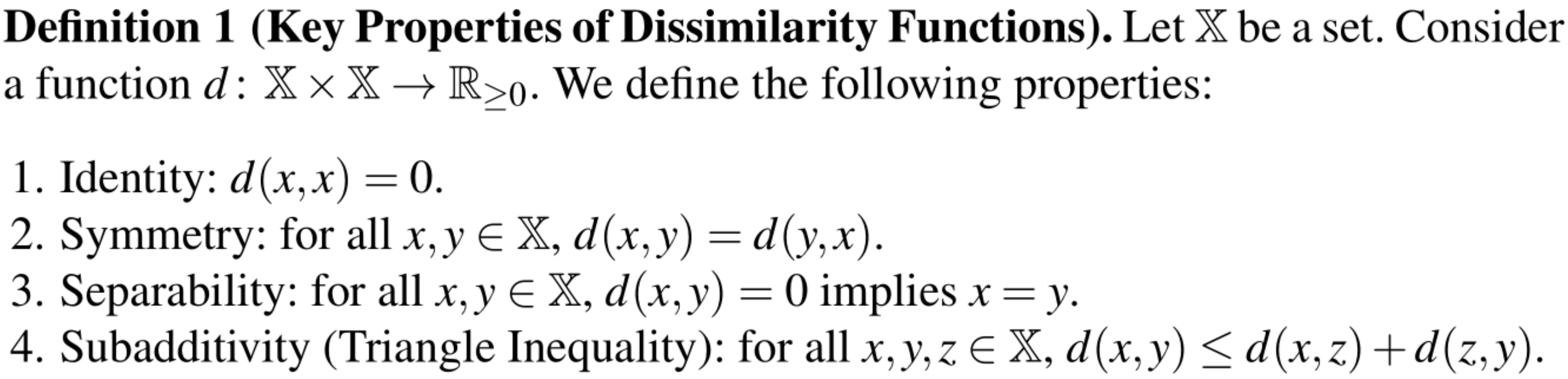
- Can be computed in polynomial time; O(N log N) in the plane

- **Con:** When applied to graph comparison, δ_H only compares the geometry but not the topology
- **Pro:** $\overrightarrow{\delta_H}$ allows for partial comparison of one graph



- δ_H is a metric on the set of compact subsets of $\mathbb{R}^d$

Metric Properties



Definition 1 (Key Properties of Dissimilarity Functions). Let $\mathbb{X}$ be a set. Consider a function $d\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$. We define the following properties:

1. Identity: $d(x,x) = 0$.
2. Symmetry: for all $x,y \in \mathbb{X}$, $d(x,y) = d(y,x)$.
3. Separability: for all $x,y \in \mathbb{X}$, $d(x,y) = 0$ implies $x = y$.
4. Subadditivity (Triangle Inequality): for all $x,y,z \in \mathbb{X}$, $d(x,y) \leq d(x,z) + d(z,y)$.

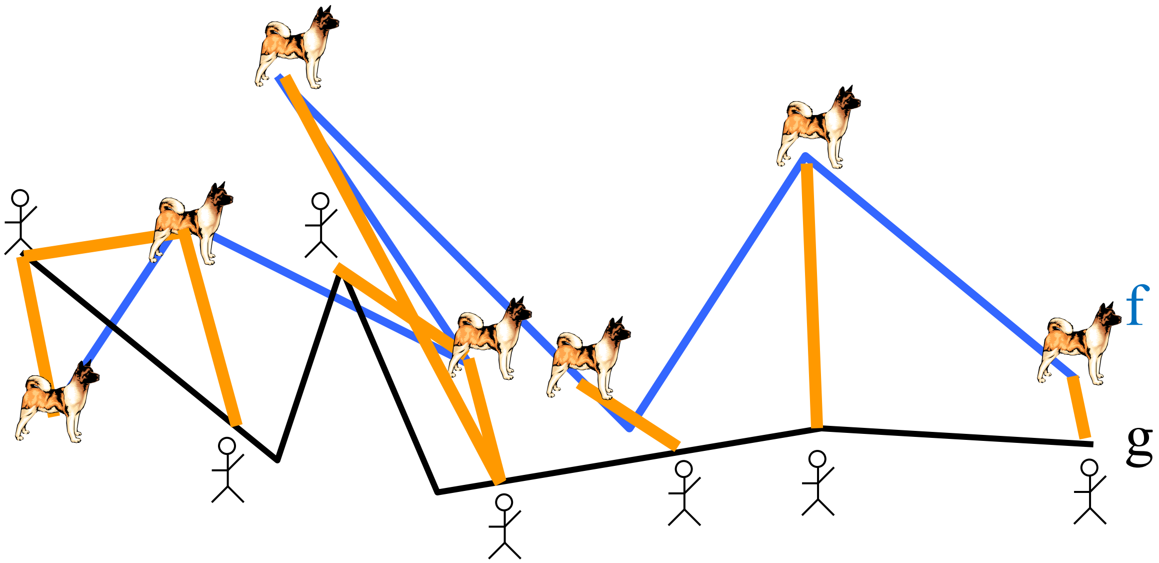
- **Metric:** Fulfills 1.-4.
- **Directed:** Does not fulfill 2.
- **Pseudo-metric:** 1., 2., 4.
- **Semi-metric:** 1., 2., 3.
- **Quasi-metric:** 1., 3., 4.

$\Rightarrow \overrightarrow{\delta_H}$ is a directed pseudo-metric

Fréchet Distance for Curves

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

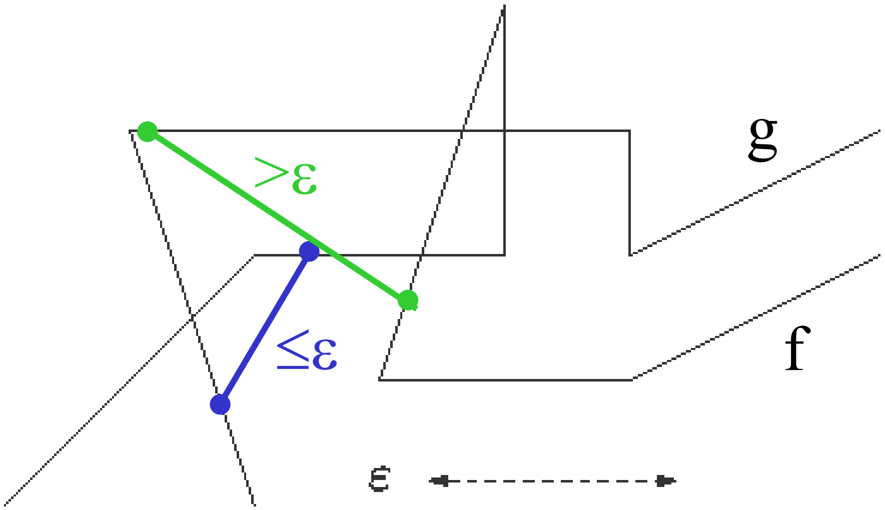
where α and β range over continuous monotone increasing reparameterizations only.

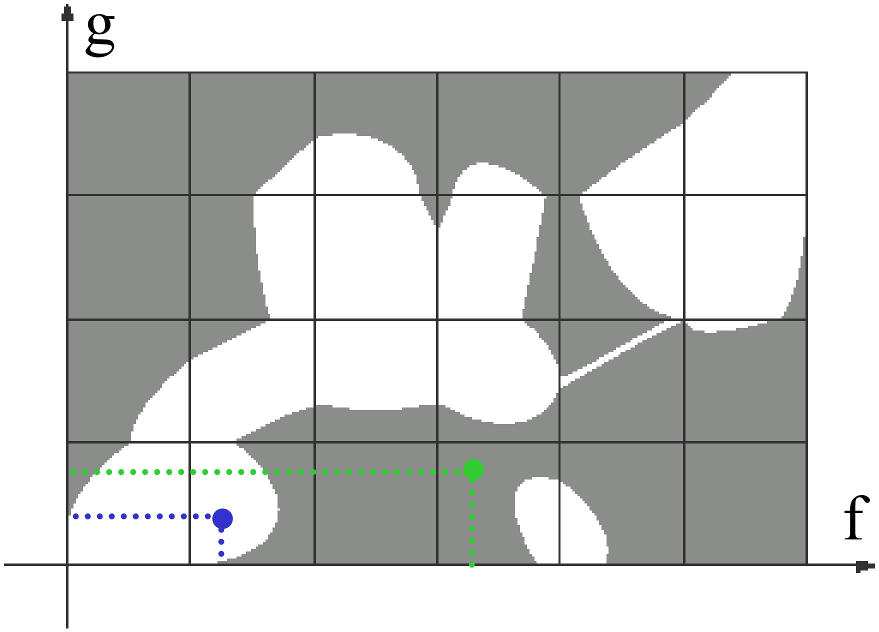


- Man and dog walk on one curve each
- They hold each other at a **leash**
- They are only allowed to go forward
- δ_F is the minimal possible leash length

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

Free Space Diagram

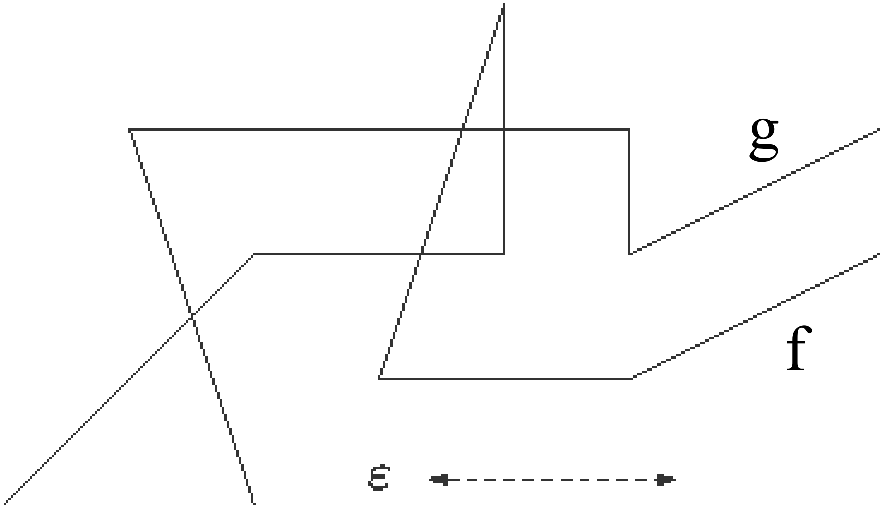


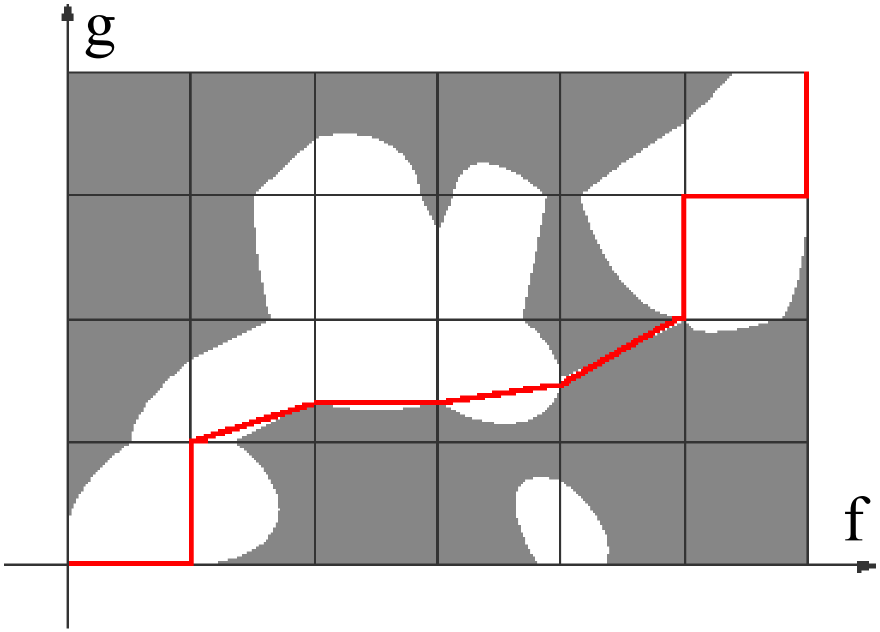


- Let ε > 0 fixed (eventually solve decision problem)
- $F_\varepsilon(f,g) = \{ (s,t) \in [0,1]^2 \mid \| f(s) - g(t) \| \leq \varepsilon \}$ *white points*

  free space of f and g
- The free space in one cell is an ellipse.

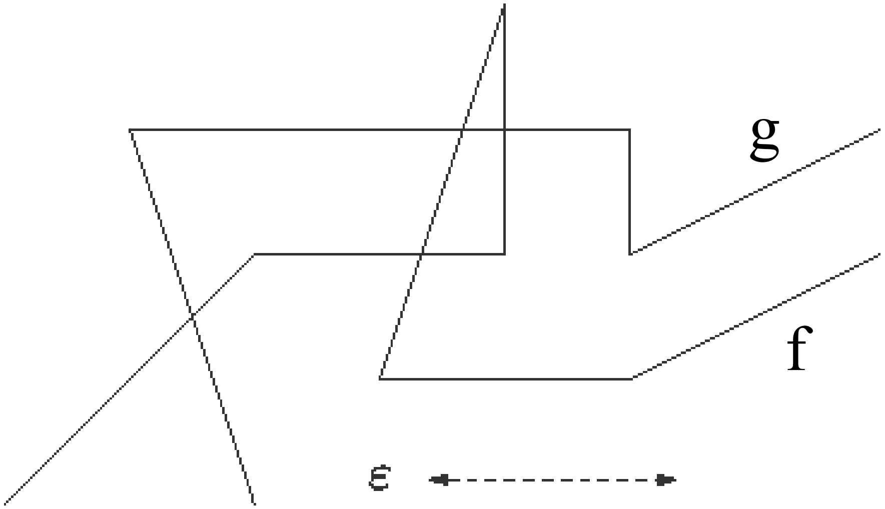
Free Space Diagram

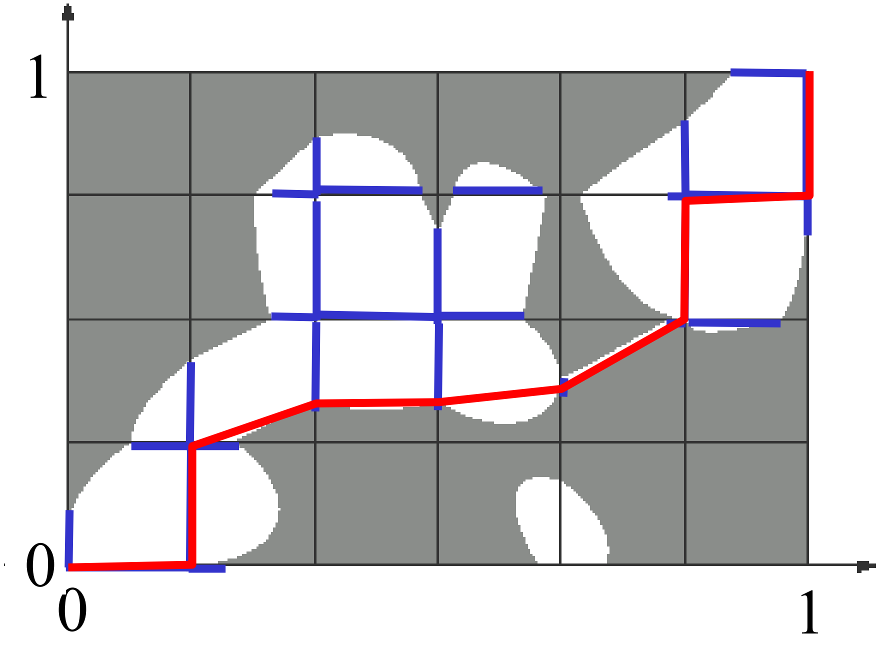




- Monotone path encodes reparametrizations of f and g
- $\delta_F(f,g) \leq \varepsilon$ iff there is a monotone path in the free space from (0,0) to (1,1)
- Such a path can be computed using DP in O(*mn*) time

Free Space Diagram



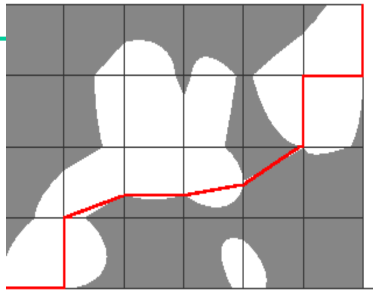


- Monotone path encodes reparametrizations of f and g
- $\delta_F(f,g) \leq \varepsilon$ iff there is a monotone path in the free space from (0,0) to (1,1)
- Such a path can be computed using DP in O(*mn*) time

Weak Fréchet Distance

- Weak Fréchet distance $\delta_{wF}(f,g)$: Allow any continuous reparameterizations α and β
$\Rightarrow$ Any continuous path in free space (not necessarily monotone)
- $\delta_H(f,g) \leq \delta_{wF}(f,g) \leq \delta_F(f,g)$

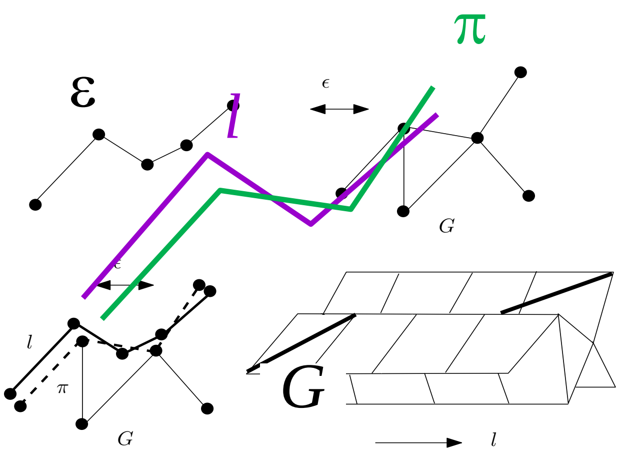
Map-Matching

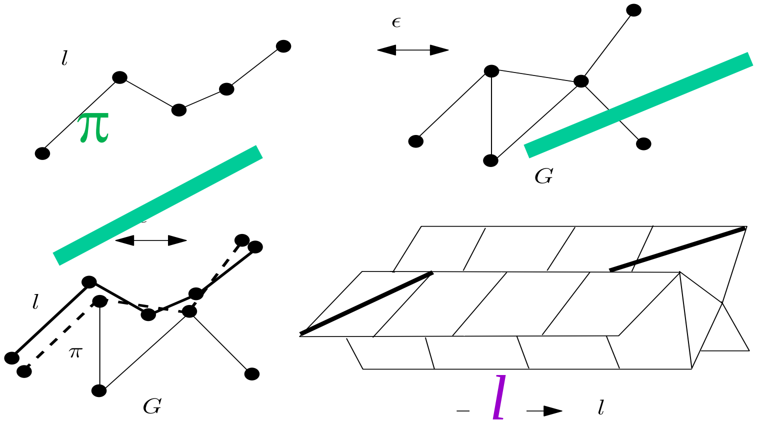


Given: A graph *G*, a curve *l*, and a distance parameter ε.

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

Compute free space surface.
And find path π' in it





Such a path can be computed using DP in O(*mn*) time

Fréchet Distance, General

Let $A, B \subseteq \mathbb{R}^k$ be two oriented manifolds. And let $f: A \rightarrow \mathbb{R}^d$ and g: $B \rightarrow \mathbb{R}^d$ be two immersions. Then

$$\delta_F(f, g) = \inf_\alpha \max_{t \in A} \|f(t) - g(\alpha(t))\|,$$

where $\alpha: A \rightarrow B$ ranges over all orientation-preserving homeomorphisms.

- The Fréchet distance is a pseudo-metric (separability is not fulfilled, since shapes with different parameterizations can have distance 0).
- Originally defined for oriented manifolds, but can be generalized even further.

Fréchet Distance, Immersed Graphs

Let $G = (V_G, E_G, \phi_G)$ and H $= (V_H, E_H, \phi_H)$ be two immersed graphs.

- We can apply the Fréchet distance definition in principle on the maps ϕ_G and ϕ_G.
- Drop the „orientation-preserving“ requirement.
- Equivalent definition:

$$\delta_F(G, H) = \inf_{\alpha} \max_{e \in E_G} \delta_F(e, \alpha(e)),$$

where α ranges over all edge mappings corresponding to isomorphisms of G and H.

- Is graph-isomorphism hard. Can be computed in poly time for trees and for graphs of bounded tree-width. [BKN20]
- For planar graphs, can enumerate orientation-preserving isomorphisms in polynomial time. [FW21]

[BKN20] M. Buchin, A. Krivosija, A. Neuhaus. Computing the Fréchet distance of trees and graphs of bounded tree width. EuroCG. 2020

[FW21] P. Fang, C. Wenk. The Fréchet distance for plane graphs. EuroCG 21.

Path-Based Distance

- Directed Hausdorff distance on path-sets:

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

Fréchet distance

- π_G path-set in G, and π_H path-set in H
- 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.

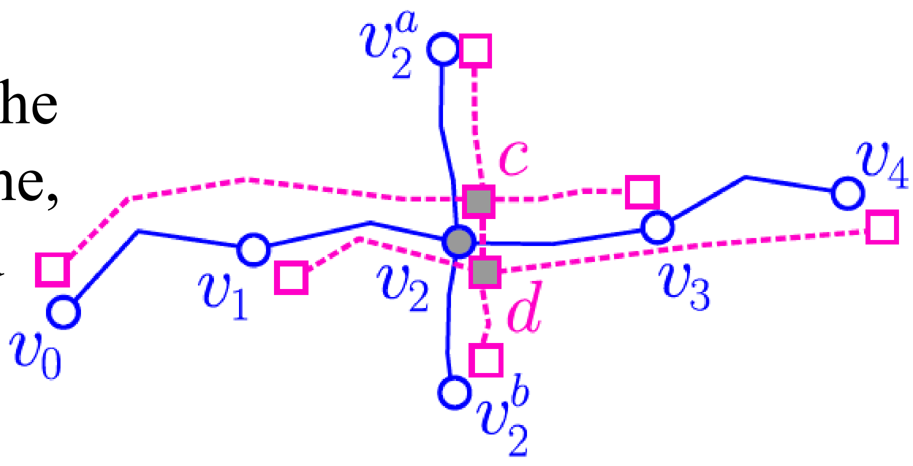
Path-Based Distance

- Ideally, π_G and π_H are the set of all paths in G and H

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

- It is a directed pseudo-metric.

- One can use the set of paths of link-length three to approximate the overall distance in polynomial time, if vertices in G are well-separated and have degree ≠ 3.
→ Stitch link-length three paths together to form longer paths



[AFHW14] M. Ahmed, B. Fasy, K. Hickmann, **C. Wenk**, Path-based distance for street map comparison, TSAS 1(1): article 3, 28 pages, 2015.

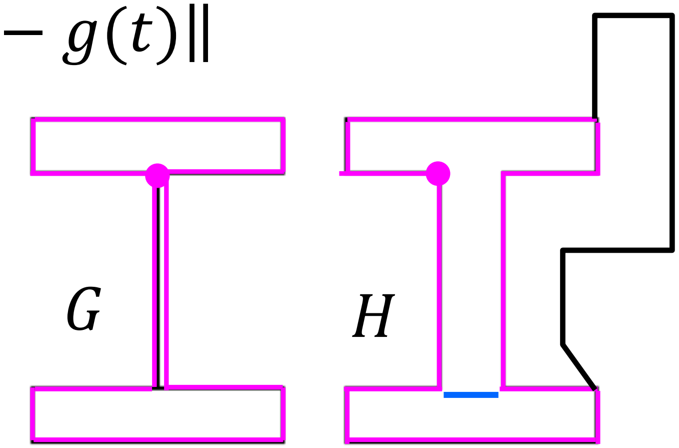
Traversal Distance

Let $G = (V_G, E_G, \phi_G)$ and H $= (V_H, E_H, \phi_H)$ be two immersed graphs.

- Represent G by traversals $f\colon [0,1] \to G$ (continuous, surjective) and H by partial traversals g: $[0,1] \to H$:

$$\overrightarrow{d_T}(G,H) = \inf_{f,g} \max_{t\in[0,1]} \|f(t) - g(t)\|$$

- Can be computed in O(mn log mn) time using free space diagram.
- Is a directed distance, but fulfills neither separability nor triangle inequality.
- Concides with the weak Fréchet distance when G and H are polygonal curves.



Small traversal distance

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

Traversal Distance

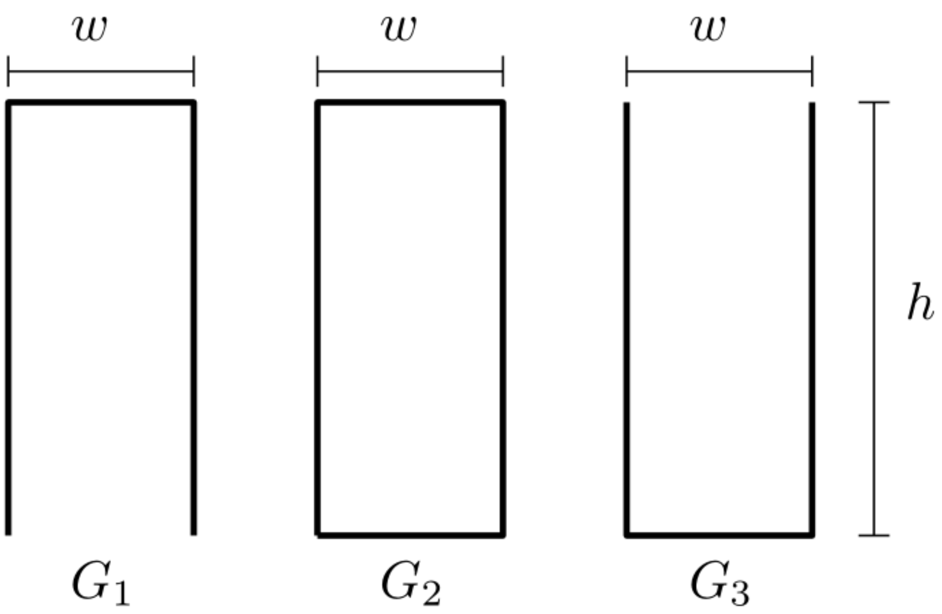


Fig. 1: Example showing that the traversal distance violates the separability and triangle inequality. Assume all graphs lie on top of each other, i.e., G_1 and G_3 are subgraphs of G_2. Then $\overrightarrow{d_T}(G_1,G_2) = 0, \overrightarrow{d_T}(G_2,G_3) = w/2$, but $G_1 \neq G_2$ and $\overrightarrow{d_T}(G_1,G_3) = w > w/2$.

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

Strong and Weak Graph Distances

Let $G = (V_G, E_G, \phi_G)$ and H $= (V_H, E_H, \phi_H)$ be two immersed graphs.

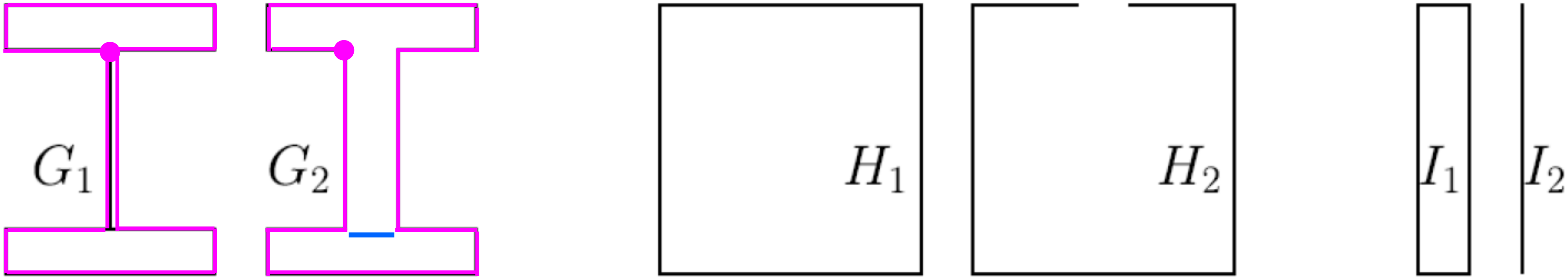
- Define a graph mapping $s: G \to H$ as follows:
  - s sends each $v \in V_G$ to a point $s(v) \in H$
  - s sends each e $\in E_G$ to a simple path from $s(u)$ to $s(v)$ in H.
- Then the strong graph distance is

$$\vec{\delta}(G,H) = \inf_{s:G\to H} \max_{e\in E_G} \delta_F(e, s(e))$$

- The weak graph distance $\overrightarrow{\delta_w}$ uses δ_{wF} instead of δ_F.
- We have $\overrightarrow{d_T}(G,H) \leq \overrightarrow{\delta_w}(G,H) \leq \vec{\delta}(G,H)$
- NP-hard to decide, but can be computed in poly time for trees, and the weak graph distance can be computed in poly time for planar embedded graphs.

[ABKS**W**21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, **C. Wenk**, Distance Measures for Embedded Graphs, CGTA 95, 101743, 2021.

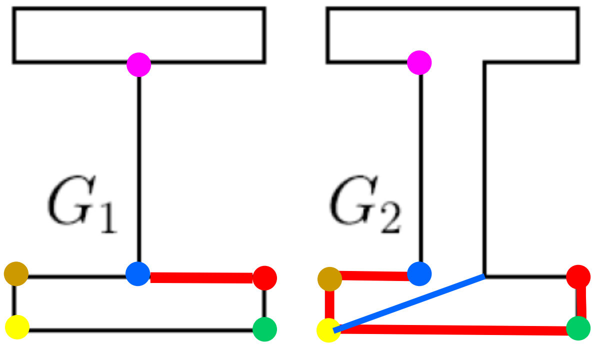
Traversal and Graph Distance



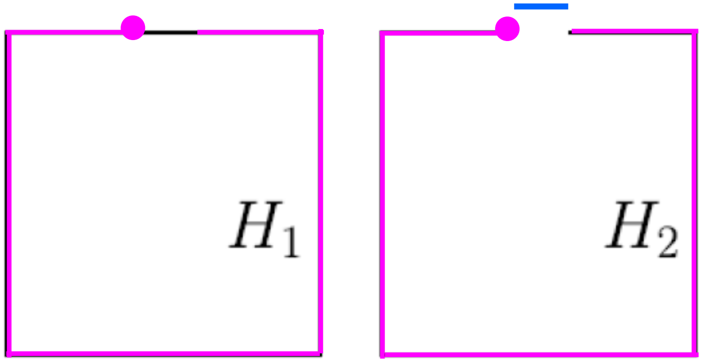
- Small traversal distance

[ABKS**W**21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, **C. Wenk**, Distance Measures for Embedded Graphs, CGTA 95, 101743, 2021.

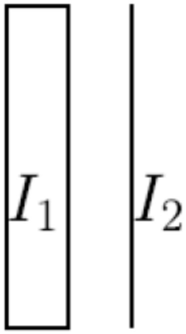
Traversal and Graph Distance



- Small traversal distance
- Large graph distance

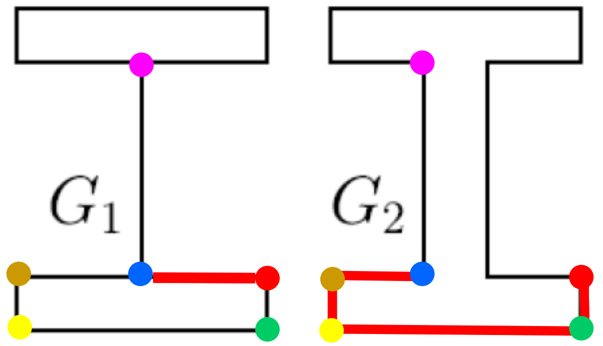


- Small traversal distance

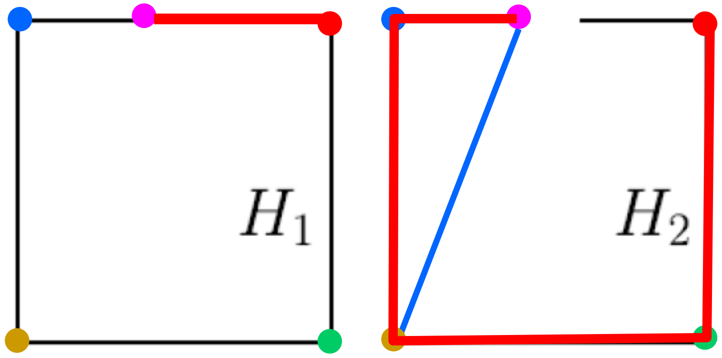


[ABKSW21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, C. Wenk, Distance Measures for Embedded Graphs, CGTA 95, 101743, 2021.

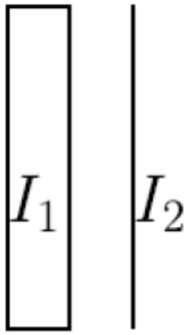
Traversal and Graph Distance



- Small traversal distance
- Large graph distance



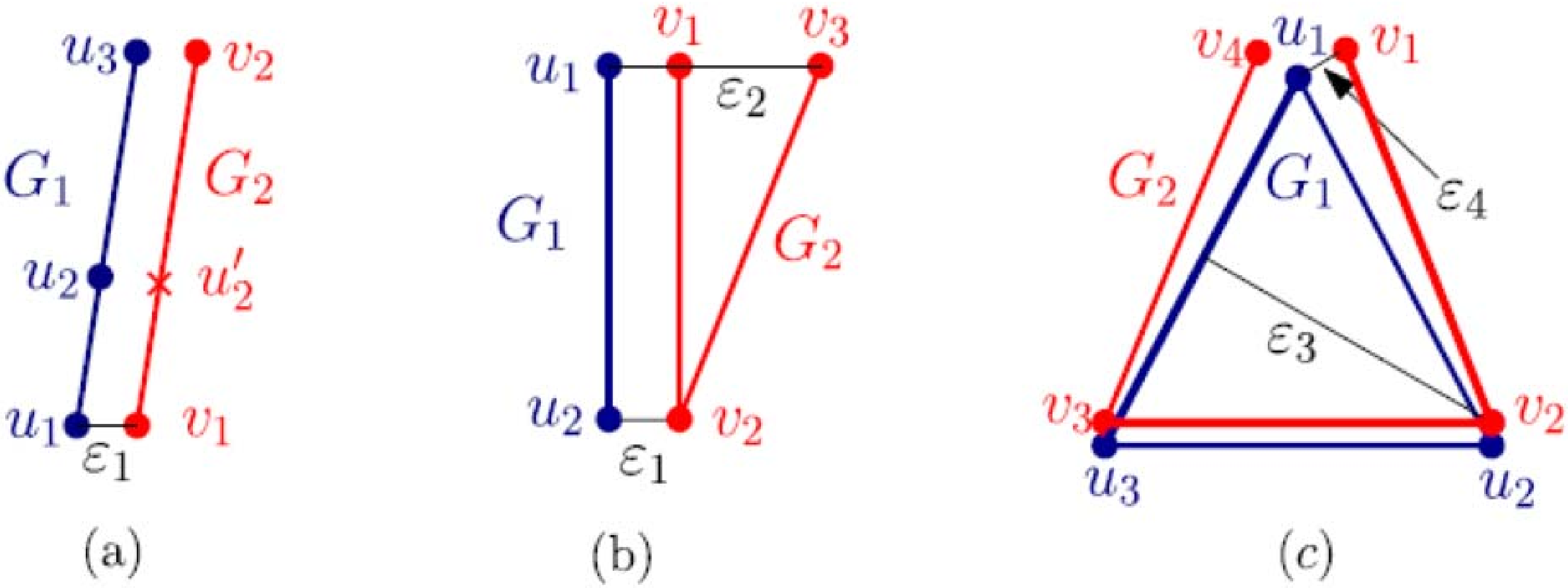
- Small traversal distance
- Large graph distance



- Both small

[ABKSW21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, **C. Wenk**, Distance Measures for Embedded Graphs, CGTA 95, 101743, 2021.

Strong and Weak Graph Distances



[ABKSW21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, **C. Wenk**, Distance Measures for Embedded Graphs, CGTA 95, 101743, 2021.

Contour Tree Distance

Let $G = (V_G, E_G, \phi_G)$ and H $= (V_H, E_H, \phi_H)$ be two connected immersed graphs.

- The contour tree distance is

$$d_C(G, H) = \inf_{\tau} \sup_{(x,y)\in\tau} \|x - y\|,$$

where G ranges over all correspondences τ between G and H such that

1. $\tau \subseteq G \times H$ is connected
2. For each $x \in G$: The set $\tau \cap (\{x\} \times H)$ is non-empty and connected
3. For each y $\in H$: The set $\tau \cap (G \times \{y\})$ is non-empty and connected

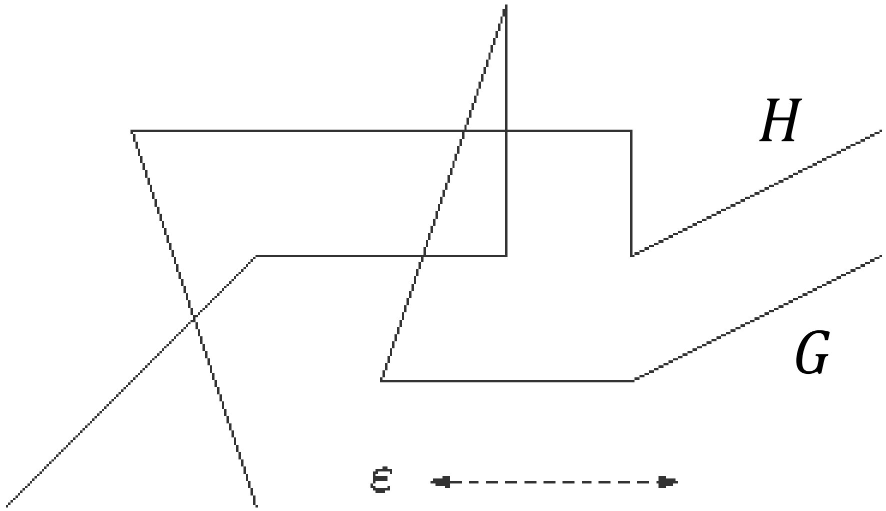
[BOS17] K. Buchin, T. Ophelders, B. Speckmann. Computing the Frechet distance between real-valued surfaces. SODA: 2443–2455, 2017.

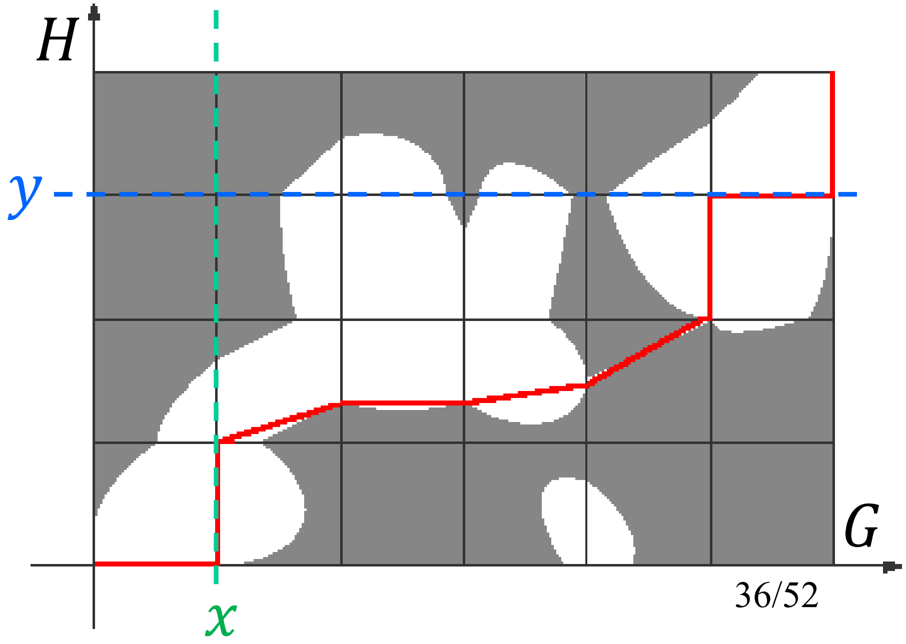
Contour Tree Distance

$$d_C(G,H) = \inf_{\tau} \sup_{(x,y)\in\tau} \|x-y\|,$$

where G ranges over all correspondences τ between G and H such that

1. $\tau \subseteq G \times H$ is connected
2. For each $x \in G$: The set $\tau \cap (\{x\} \times H)$ is non-empty and connected
3. For each y $\in H$: The set $\tau \cap (G \times \{y\})$ is non-empty and connected





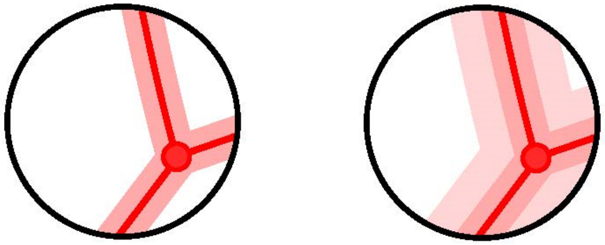
Contour Tree Distance

- The contour tree distance is a metric.
- But it is NP-complete, already for trees.
- This distance seems to correspond to a symmetric version of the (strong or weak) graph distances.

[BOS17] K. Buchin, T. Ophelders, B. Speckmann. Computing the Frechet distance between real-valued surfaces. SODA: 2443–2455, 2017.

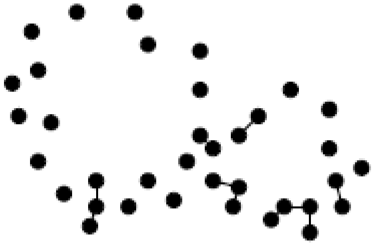
[BK**W**21] M. Buchin, B. Kilgus, **C. Wenk**. Ongoing work.

3. Local Persistent Homology Distance and Local Signatures

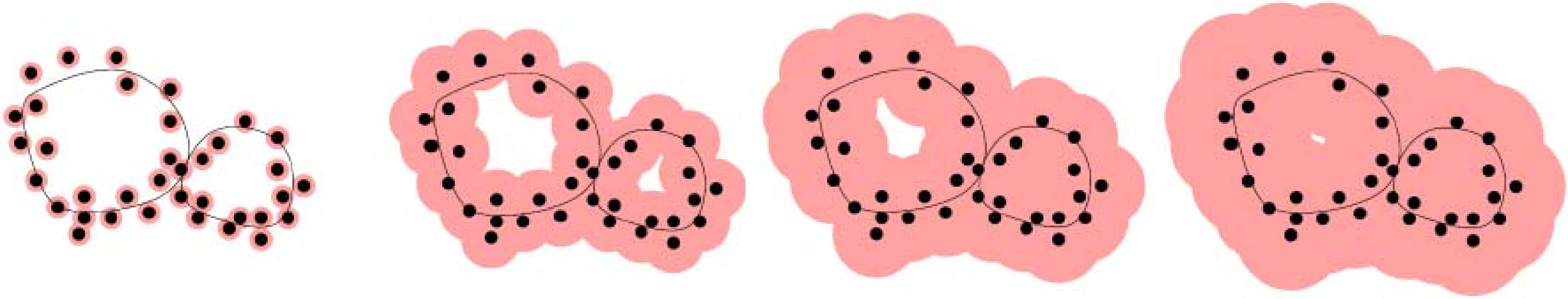


Excursion into Computational Topology: Persistent Homology

- Develop topological descriptors to analyze point set shapes

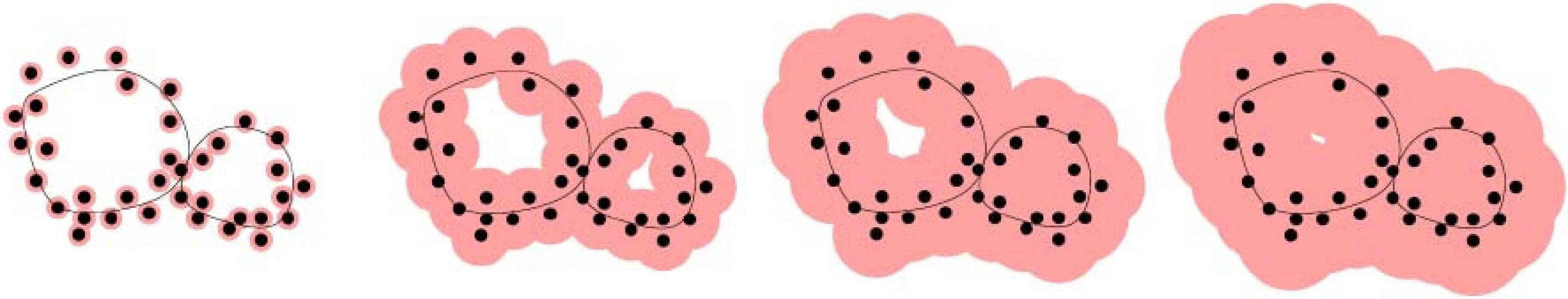


- It looks like this shape contains two cycles. But how do we know?
- Let's make the points thicker:



Adapted from Tamal Dey's slides http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf

Persistent Homology



- $\mathrm{f}(x) = d(x, P)$: distance to point cloud P
- Sublevel sets $f^{-1}[0, r]$ are union of balls
- Evolution of the sublevel sets with increasing radius r
  $\Rightarrow$ The left hole persists longer
- Growing union of balls are nested topological spaces
  $\Rightarrow$ a filtration

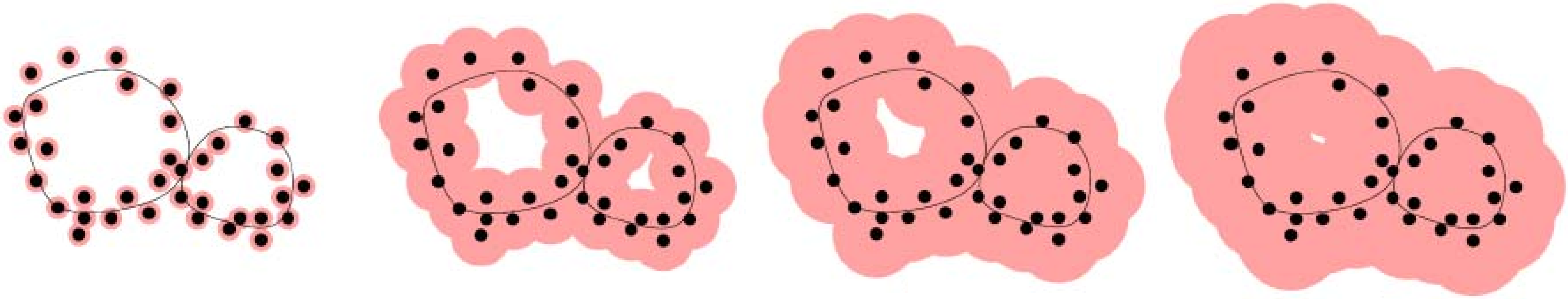
$\downarrow$ nested $\quad$ $\downarrow$ homology classes (groups)

nested homology classes (groups)

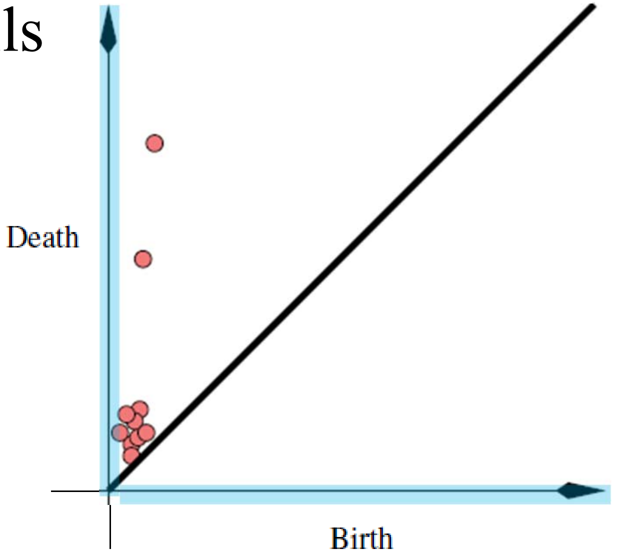
$\Rightarrow$ persistent homology classes (groups)

Adapted from Tamal Dey's slides http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf

Persistence Diagram

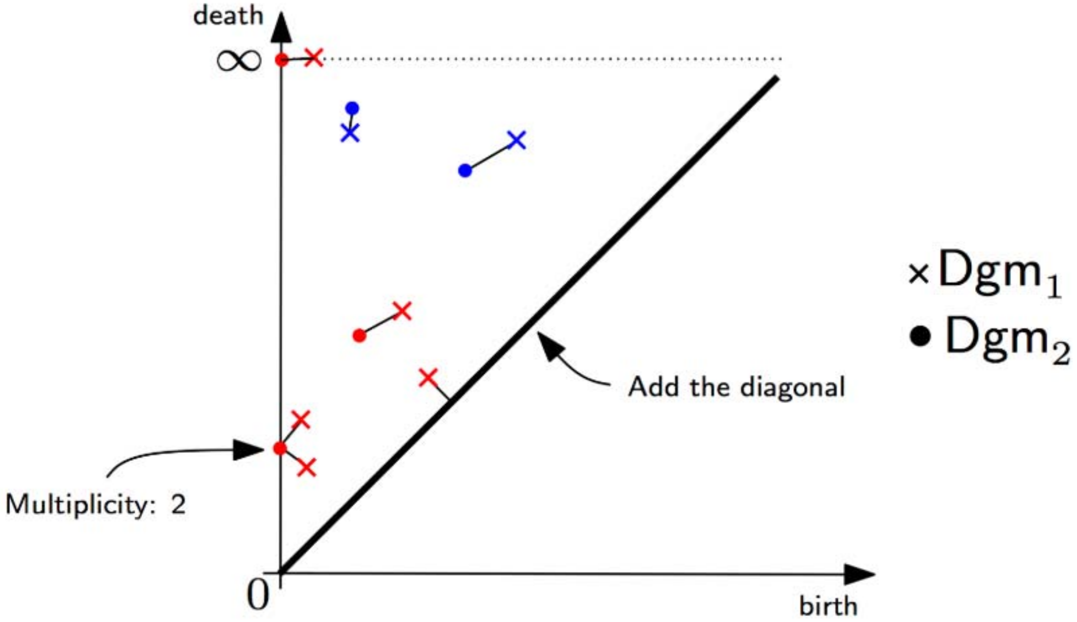


- $\mathrm{f}(x) = d(x, P)$: distance to point cloud P
- Sublevel sets $f^{-1}[0, r]$ are union of balls
- $Dgm(f, P)$ is the persistence diagram of P
- Each point in $Dgm(f, P)$ is a pair of r-values: (birth, death)
- ⇒ **Topological descriptor of P**



Adapted from Tamal Dey's slides http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf

Bottleneck Distance



The bottleneck distance between two diagrams Dgm_1 and Dgm_2 is

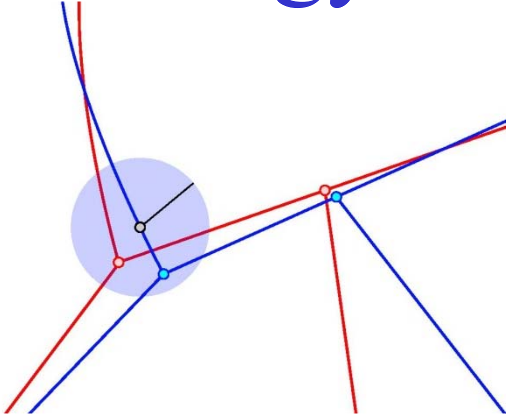
$$\mathrm{d_b}(\mathsf{Dgm}_1, \mathsf{Dgm}_2) = \inf_{\gamma \in \Gamma} \sup_{p \in \mathsf{Dgm}_1} \|p - \gamma(p)\|_\infty$$

where Γ is the set of all the bijections between Dgm_1 and Dgm_2 and

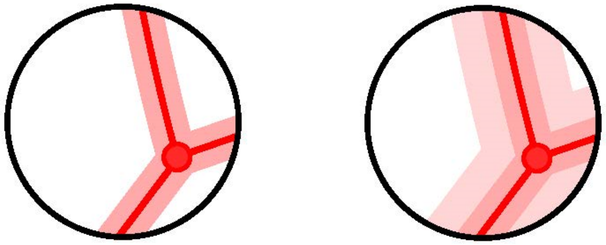
$$\|p - q\|_\infty = \max(|x_p - x_q|, |y_p - y_q|).$$

From Gudhi tutorial: http://bertrand.michel.perso.math.cnrs.fr/Enseignements/TDA/Tuto-Part2.html

Local Persistent Homology 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

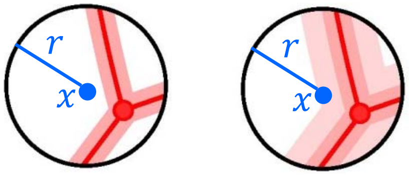


⇒ Use (bottleneck) distance between persistence diagrams to compare changing local cycle structure

[AFW14] M. Ahmed, B. Fasy, **C. Wenk**, Local persistent homology based distance between maps, ACM SIGSPATIAL, 10 pages, 2014.

Local Persistent Homology Distance

- **Local "signature"** that captures local topological similarity of graphs: $\psi_r(x) = d(\mathcal{P}_{1,x,r}, \mathcal{P}_{2,x,r})$,



where d is the bottleneck distance between the two persistence diagrams

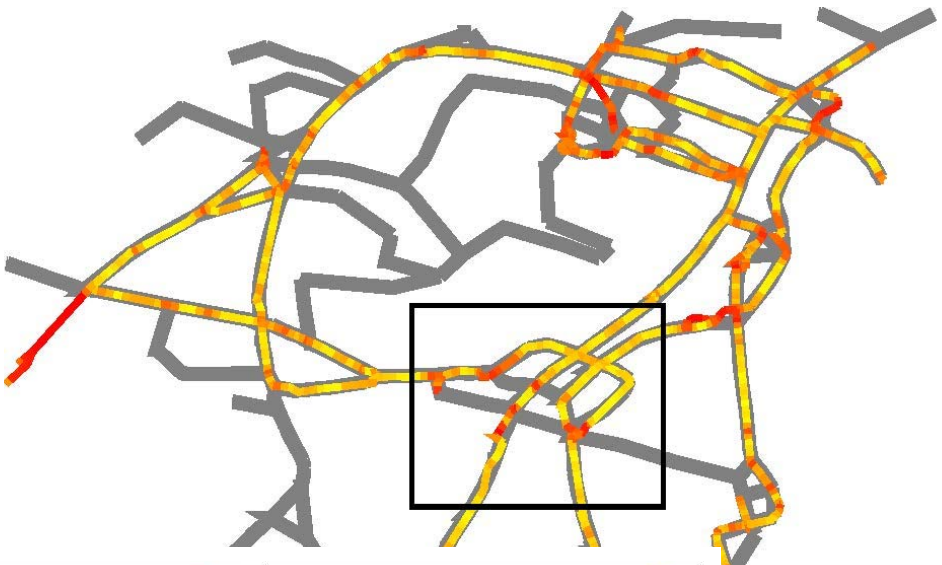
- Fixed radius: $$d_r^{LH}(G_1, G_2) = \frac{1}{|\mathbb{X}|} \int_{\mathbb{X}} \psi_r(x)\, dx,$$

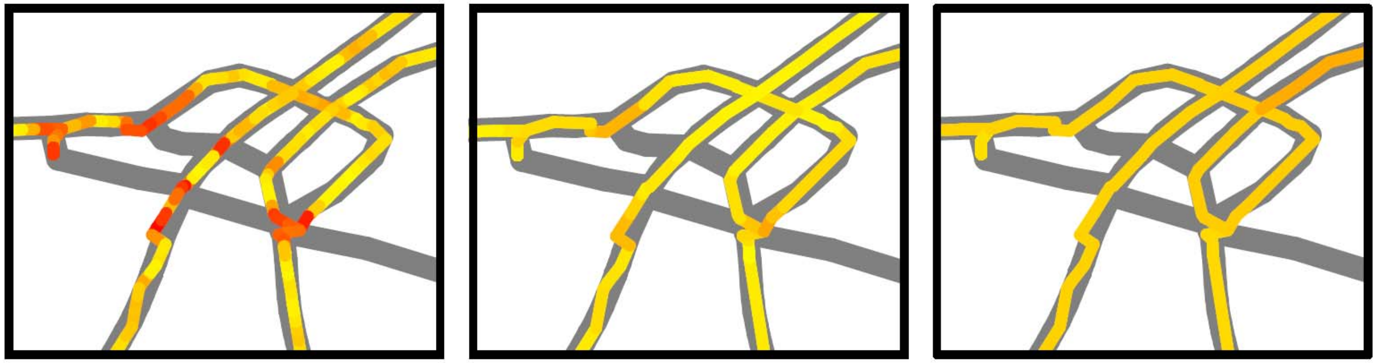
- Local homology metric: $$d^{LH}(G_1, G_2) = \frac{1}{r_1 |\mathbb{X}|} \int_0^{r_1} \omega(r) \int_{\mathbb{X}} \psi_r(x)\, dx\, dr$$

[AF**W**14] M. Ahmed, B. Fasy, **C. Wenk**, Local persistent homology based distance between maps, ACM SIGSPATIAL, 10 pages, 2014.

Local Persistent Homology Distance

- Compared two reconstructed maps.
- Disk centers sampled 5m; disk radius 25m
- Local signature captures different topology (missing intersections) well

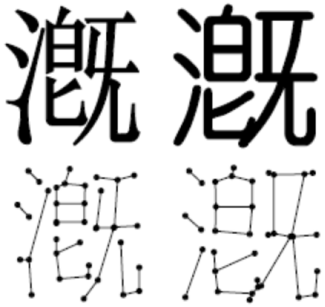


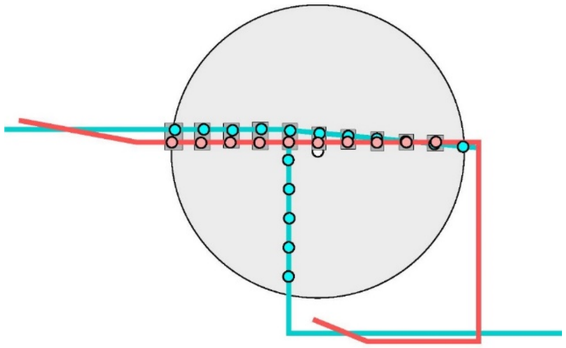


Local homology Hausdorff Path-based (Fréchet)

[AFW14] M. Ahmed, B. Fasy, C. Wenk, Local persistent homology based distance between maps, ACM SIGSPATIAL, 10 pages, 2014.

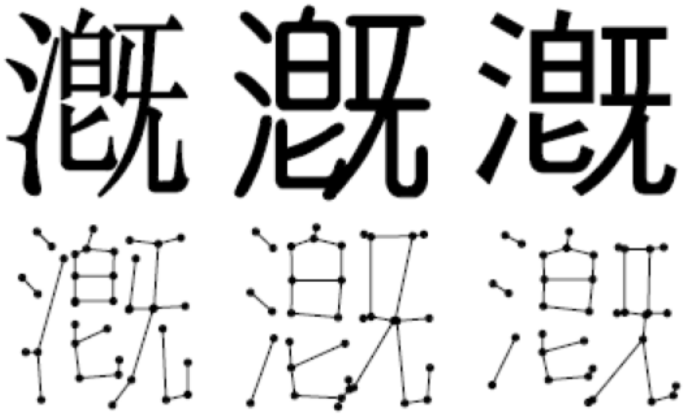
4. Other Distances





Geometric Edit Distance

- Geometric Edit Distance [CGKSS09]
  - Defined for straight-line embedded graphs.
  - Motivated by Chinese character comparison
  - Perform the following edit operations in this order:
    Edge deletion, vertex deletion,
    vertex translation,
    vertex insertion, edge insertion
  - Costs are proportional to edge lengths and to the distance a vertex has been translated.
  - Is a metric. But NP-hard.



[CGKSS09] O. Cheong, J. Gudmundsson, H.-S. Kim, D. Schymura, F. Stehn, Measuring the similarity of geometric graphs, SEA: 101–112, 2009.

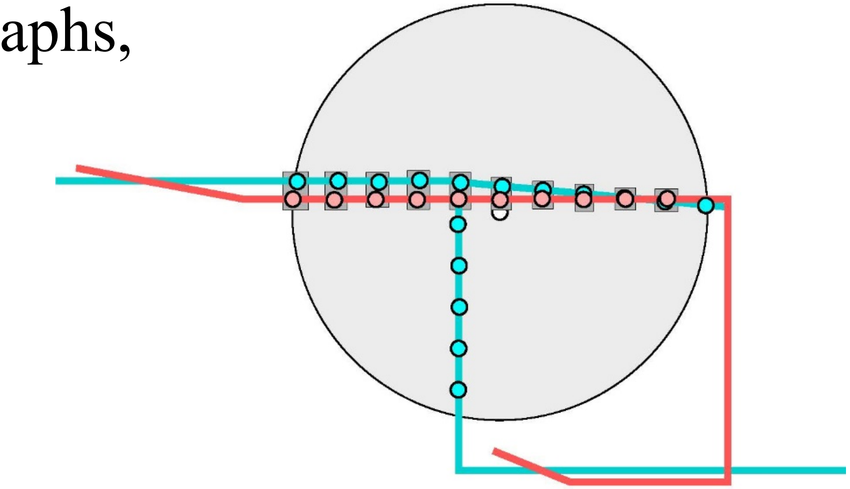
Shortest Path Sampling Distance

- Shortest Path Sampling Distance [KP12] in $\mathbb{R}^2$:
  - Randomly sample $\mathrm{x}, \mathrm{y} \in \mathbb{R}^2$
  - Find nearest neighbors x_G, y_G on G and compute a shortest path π_G from x_G to y_G in G.
  - Similarly, compute a shortest π_H from x_H to y_H in H.
  - Compute $\delta_F(\pi_G, \pi_H)$.
  - Repeat for several random samples, and compare sets of resulting distances

[KP12] S. Karagiorgou, D. Pfoser, On vehicle-tracking data-based road network generation, 20th ACM SIGPATIAL: 89-98, 2012.

Point Sampling Distance

- In a local neighborhood of both graphs, traverse the graphs (from random seeds) and place point samples. (Only graph edges of length ≤ τ.)



- $\boldsymbol{\tau}$: match_distance threshold
  m $=m(\tau)$: #samples in G
  n $=n(\tau)$: #samples in H
  k $=k(\tau)$ = #matched samples (1-1) within distance τ
- **Precision:** $p = k/n$ **Recall:** $r = k/m$ **F-score:** $2pr/(p+r) = 2k/(n+m)$

[BE12b] J. Biagioni, J. Eriksson, Inferring road maps from global… TRR: J. of the Transportation Research Board 2291, 61-71, 2012.

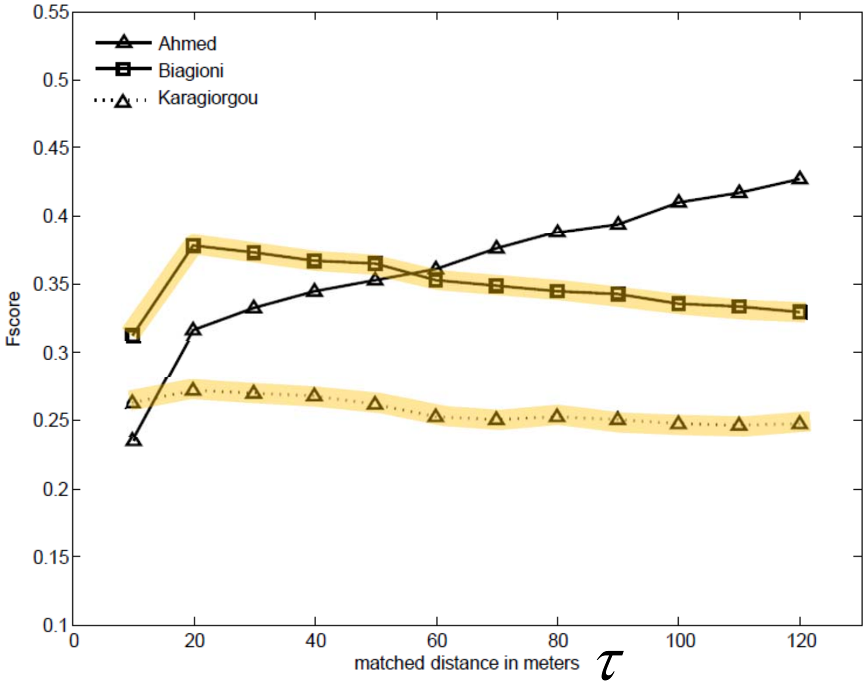
Point Sampling Distance

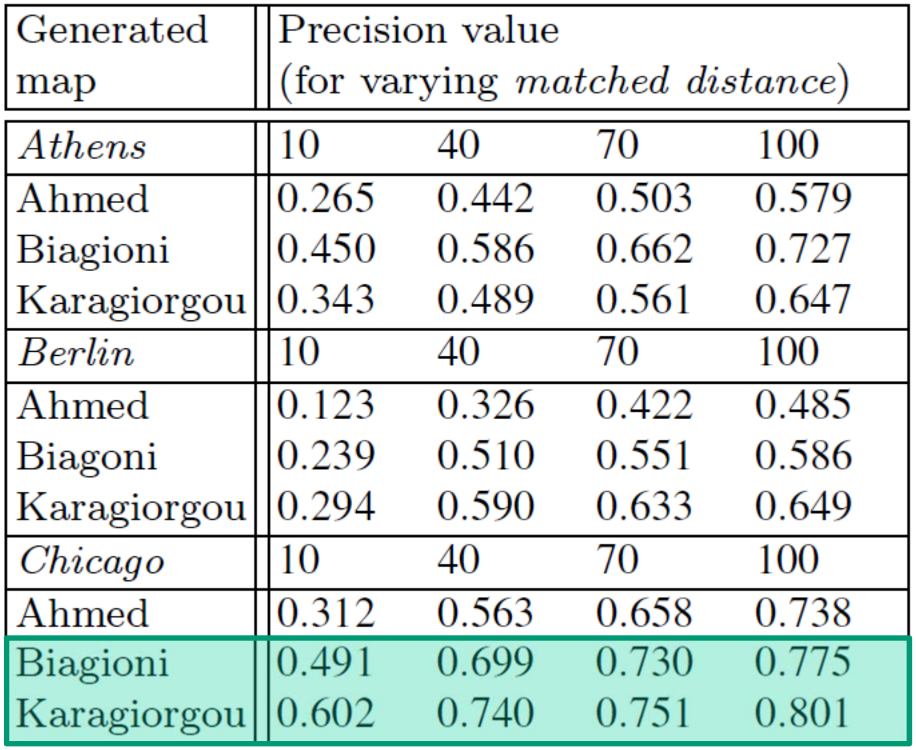
G = OSM ground-truth: m samples; H = constructed map: n samples

$2k/(n+m)$ $p = k/n$

Biagioni and Karagiorgou: F-score decreases, precision increases
→ More matched samples (k), more (unmatched) ground-truth samples (m)

Chicago

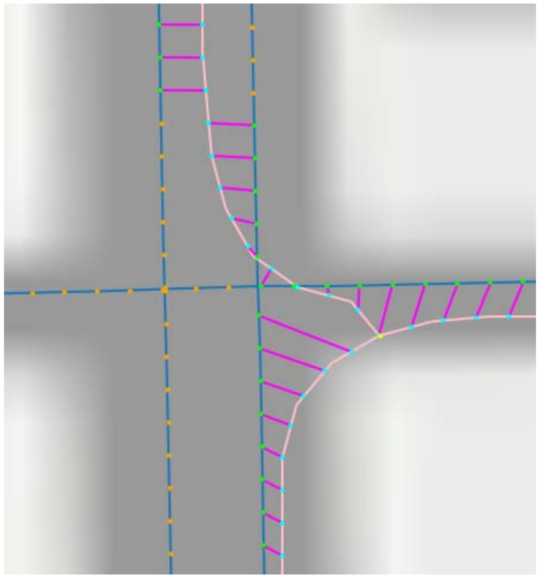




Generated map	Precision value (for varying *matched distance*)			
Athens	10	40	70	100
Ahmed	0.265	0.442	0.503	0.579
Biagioni	0.450	0.586	0.662	0.727
Karagiorgou	0.343	0.489	0.561	0.647
Berlin	10	40	70	100
Ahmed	0.123	0.326	0.422	0.485
Biagoni	0.239	0.510	0.551	0.586
Karagiorgou	0.294	0.590	0.633	0.649
Chicago	10	40	70	100
Ahmed	0.312	0.563	0.658	0.738
Biagioni	0.491	0.699	0.730	0.775
Karagiorgou	0.602	0.740	0.751	0.801

Point Sampling Distance

- Can also be used as a **local distance signature**.
- Lacks theoretical foundation but is practical.
- Does not work well if the reconstructed graph is compared with more a detailed ground-truth graph (e.g., OSM).
- Provides a matching (1-to-1) between a subset of points in G and H



- What is a good matching?
- Can one define this continuously (and compute/approximate efficiently)?

[BE12b] J. Biagioni, J. Eriksson, Inferring road maps from global… TRR: J. of the Transportation Research Board 2291, 61-71, 2012.

Conclusion & Discussion

1. We've seen a lot of distances for immersed graphs.
   - Are they useful in practice? (Noisy input, runtimes)
   - What are their mathematical properties? (Metric, topological)
2. Would like to compute a correspondence / mapping between the two graphs efficiently.
   - An application: Merge multiple road networks
3. Optimize under transformations
4. Local signatures:
   - Useful to identify local differences
   - Compute global correspondence from local correspondences?