On Length-Sensitive Fréchet Similarity

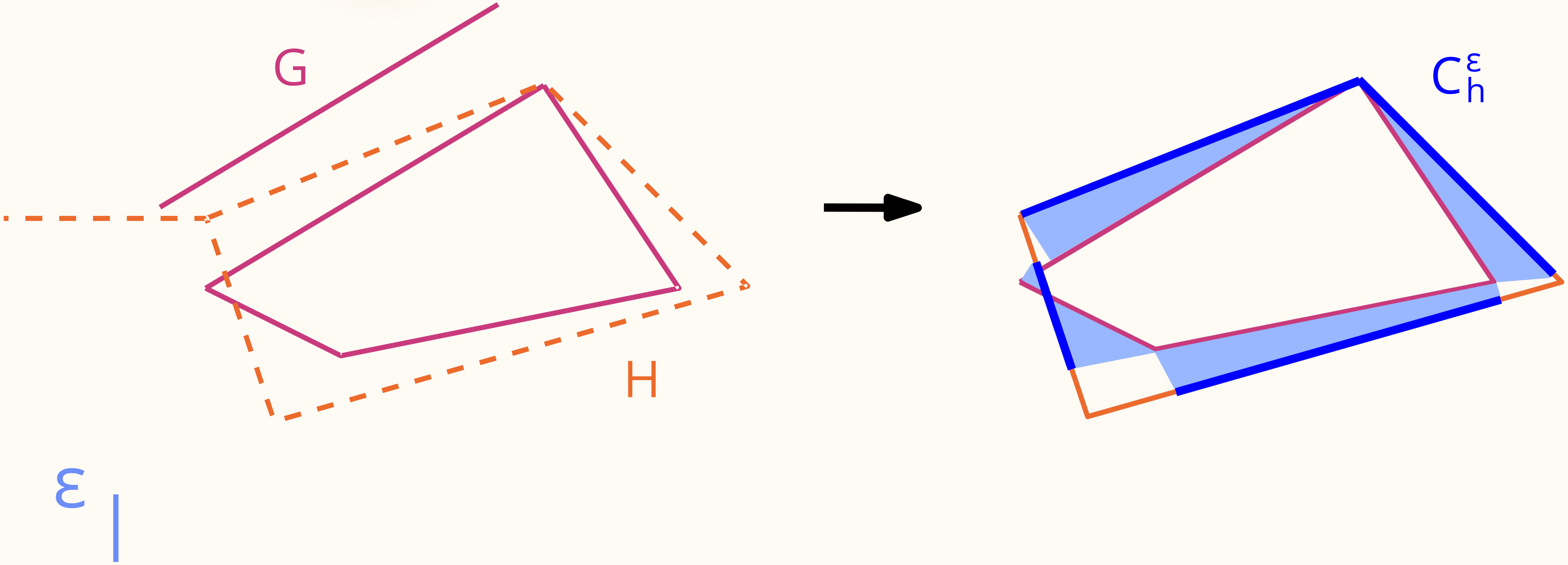
Kevin Buchin, Brittany Terese Fasy, Erfan Hosseini Sereshgi and Carola Wenk

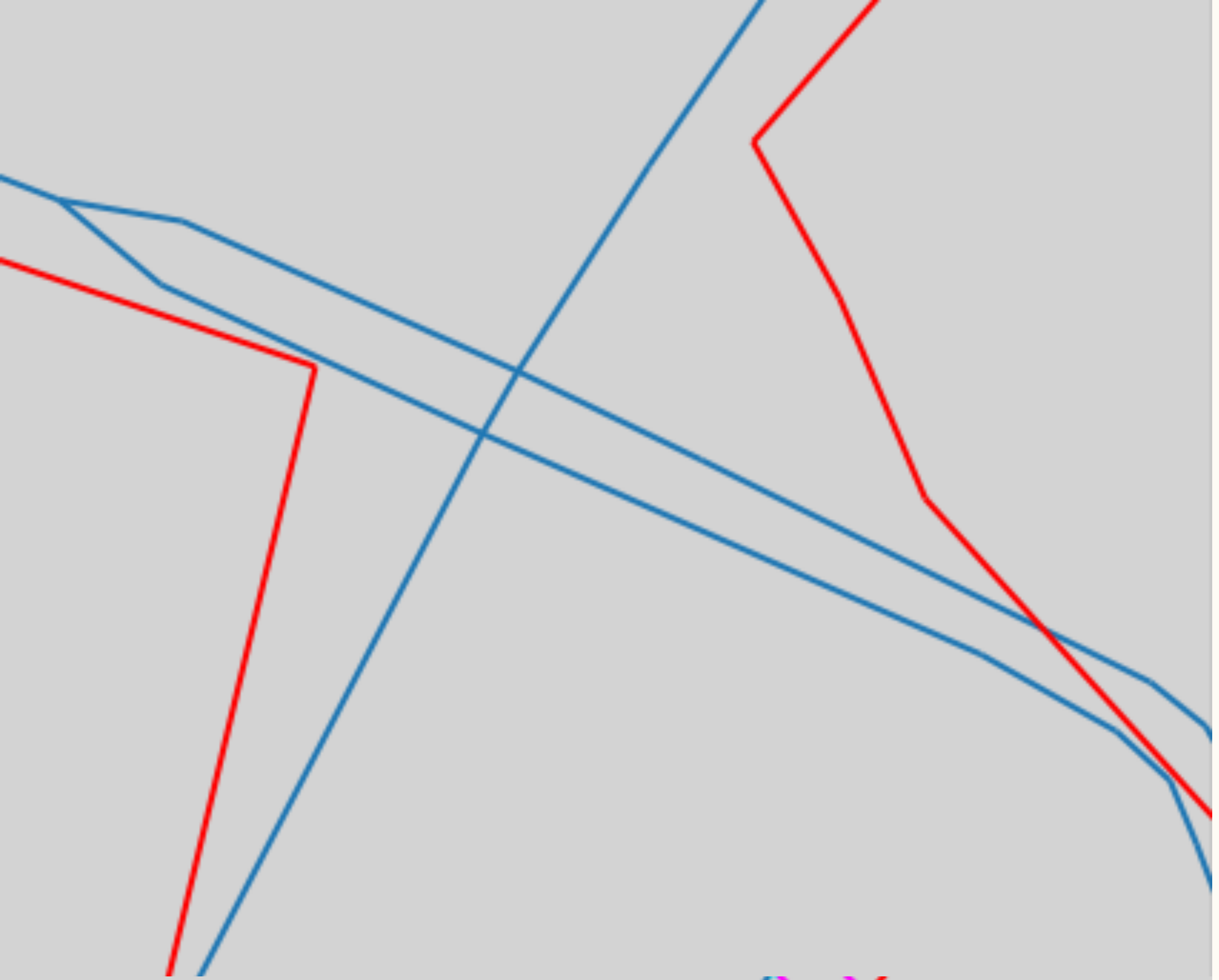
Overview

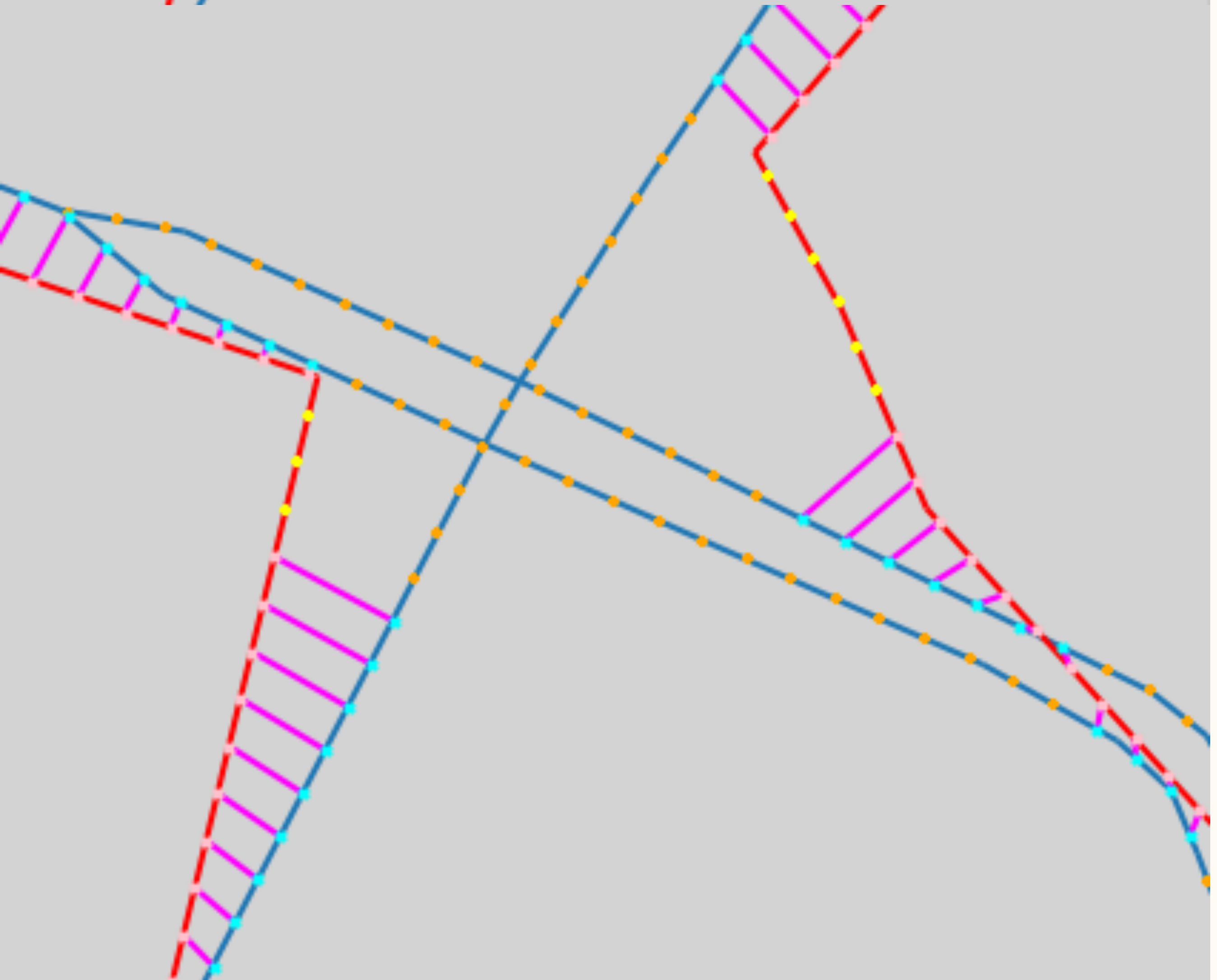
Statement of the Problem

Objectives

- Match segments of two graphs in a certain proximity one-to-one
- One-to-one = The lengths of the matched portions must be equal
- A long segment *A* can be matched to two smaller segments *B* and *C* as long as: $len(A) = len(B)+len(C)$


G
H
ε
$C_h^ε$



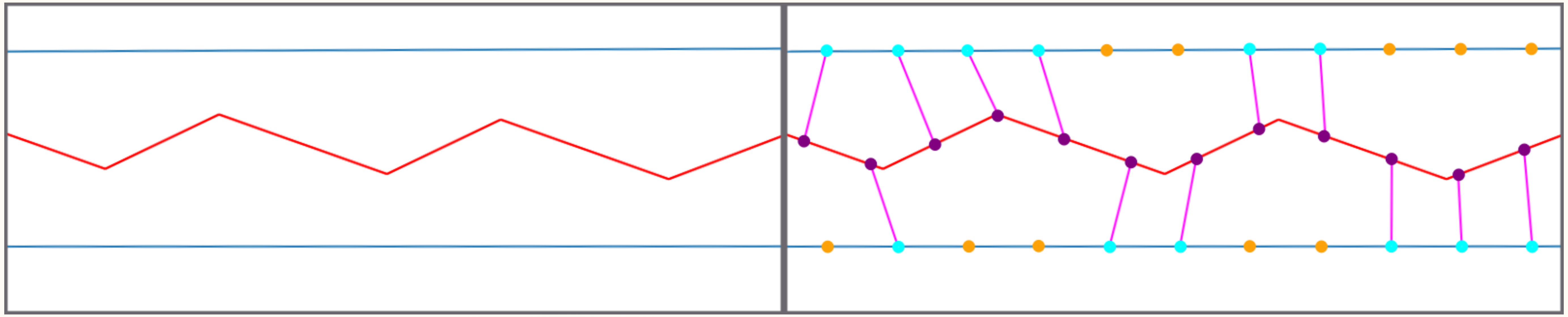


Why do we need this?

- For comparing road networks and GPS trajectories
- For comparing some biomedical structures
- For clustering any dataset of geometric graphs
- A continuous version of Graph Sampling*

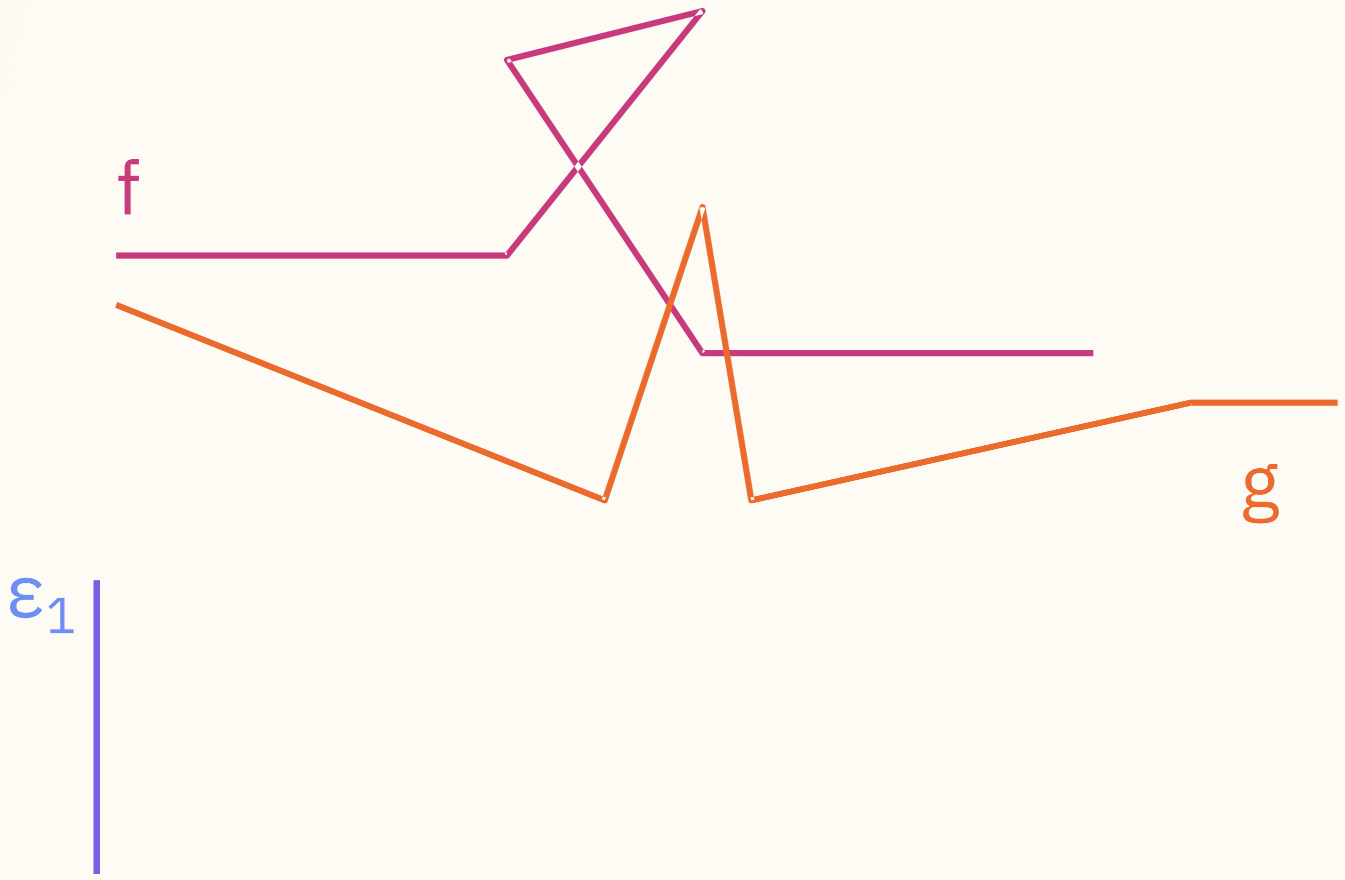
**Aguilar, J., Buchin, K., Buchin, M., Hosseini Sereshgi, E., Silveira, R.I., Wenk, C.: Graph sampling for map comparison. In: 3rd ACM SIGSPATIAL International Workshop on Spatial Gems (2021)*

Why discrete matching is not ideal:

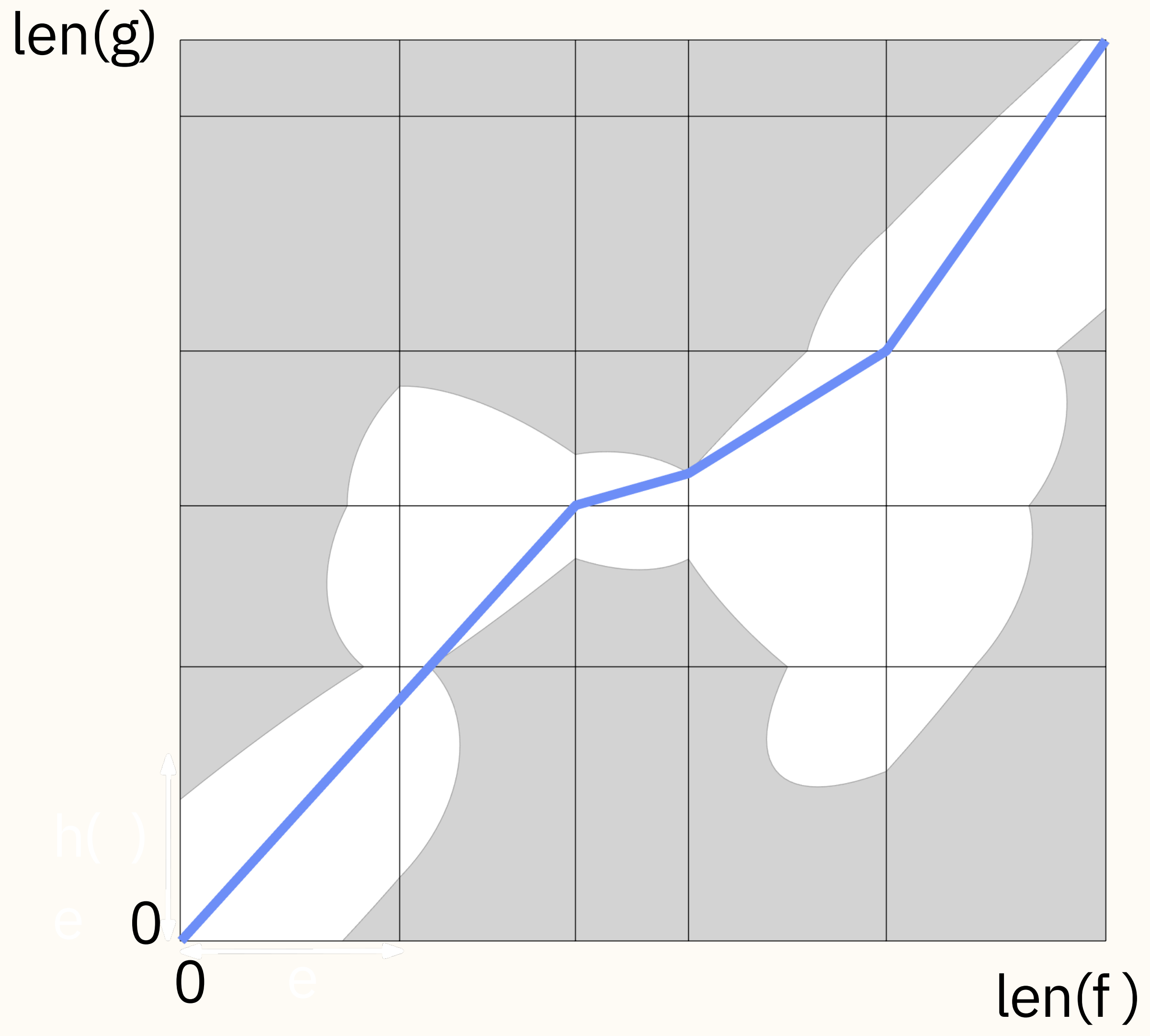


Back to Overview

Fréchet for Curves



Free-space diagram: a binary function



$$d_F(f, g) = \inf_{h:[0,L_f]\to[0,L_g]} \max_{t\in[0,L_f]} ||f\,(t) - g\,(h(t))\,||_p$$

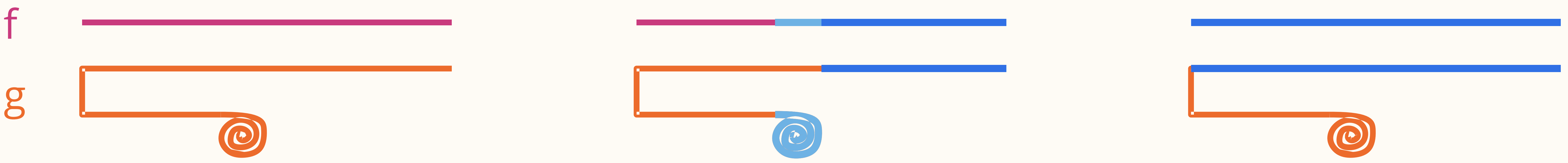
Partial Fréchet Matching (Similarity)

[Back to Overview](#)

Curve
s

Partial Fréchet Similarity*

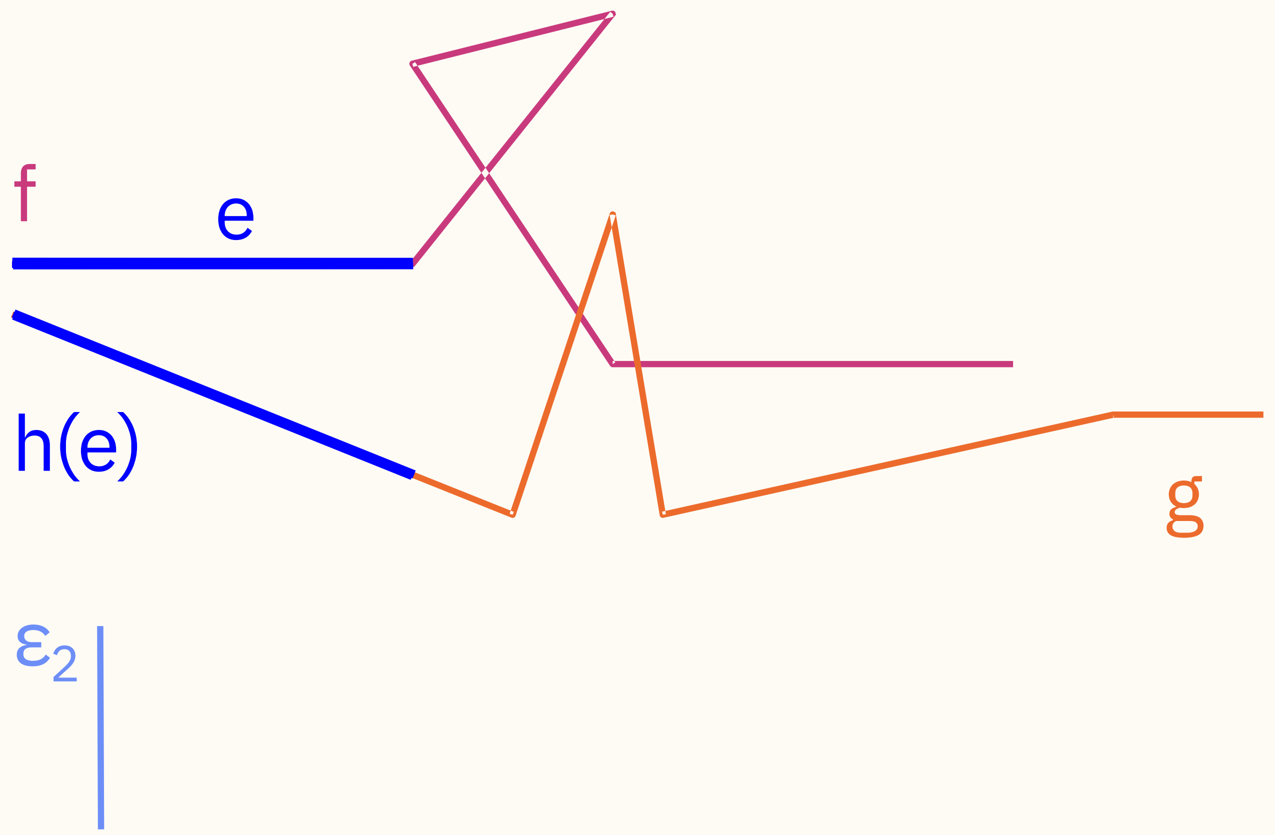
Length-Sensitive Fréchet Similarity

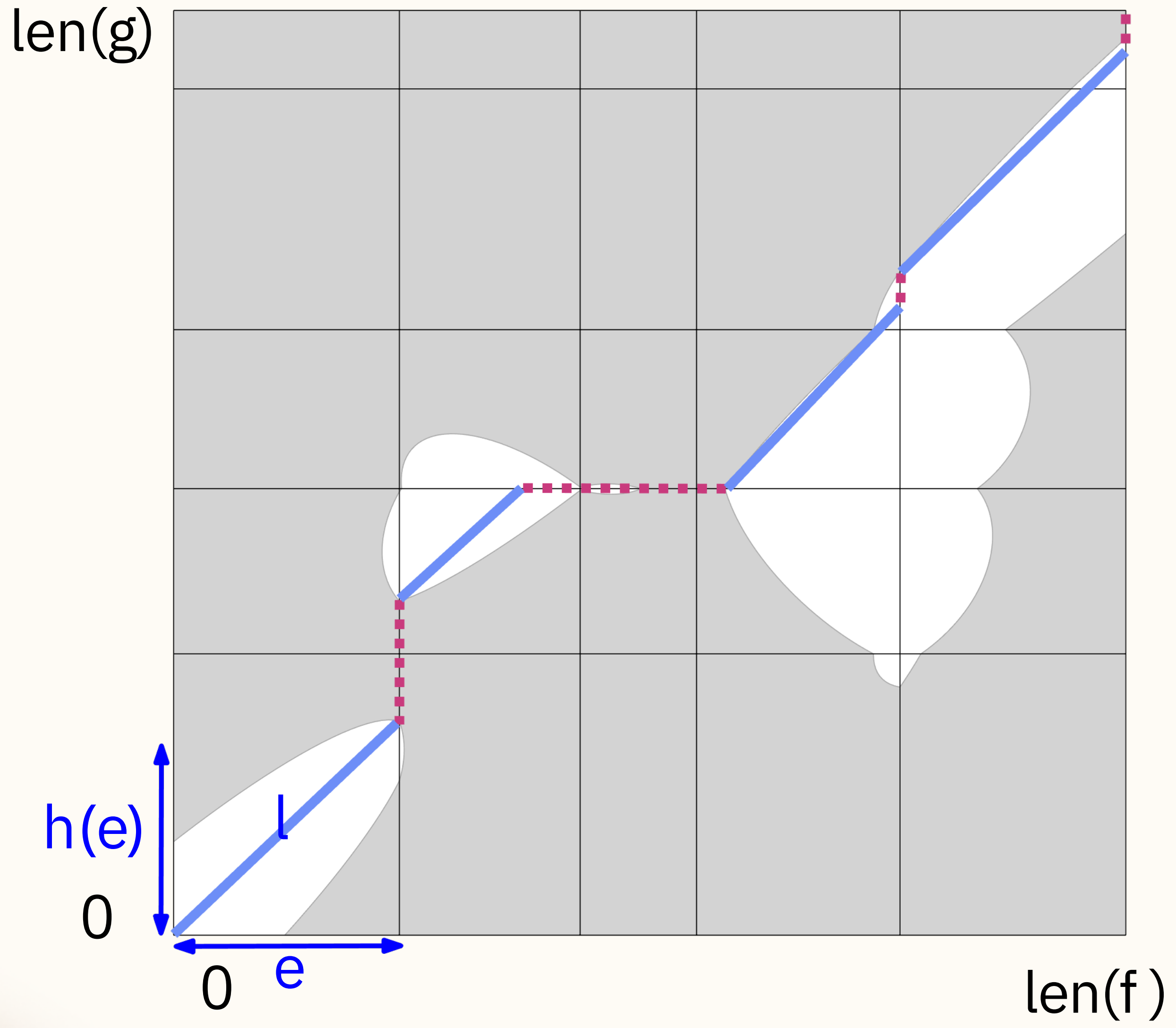


---

*Buchin, K., Buchin, M., Wang, Y.: Exact algorithms for partial curve matching via the Fréchet distance. In: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, pp. 645–654. Society for Industrial and Applied Mathematics, USA (2009)

LSFS for Curves





LSFS for Curves

Back to Overview

Suppose we have a function h: $f \rightarrow g$ that is homeomorphic onto its image

$$I_h^{\varepsilon} = \{x \in [0, L_f] \mid (x, h(x)) \text{ is free and } \exists \delta > 0\text{: } h|_{(x-\delta, x+\delta)} \text{ is length-preserving.}\}$$

We define the Length-Sensitive Fréchet Similarity (LSFS) of *(f,g)* as:

$$\mathbb{F}_{\varepsilon}(f, g) = \sup_{h:[0,L_f]\rightarrow[0,L_g]} \text{len}(I_h^{\varepsilon})$$

Computing LSFS for Curves

Refining the free-space diagram

- Initialization
- Propagation
- Concatenation

traceback (optional)

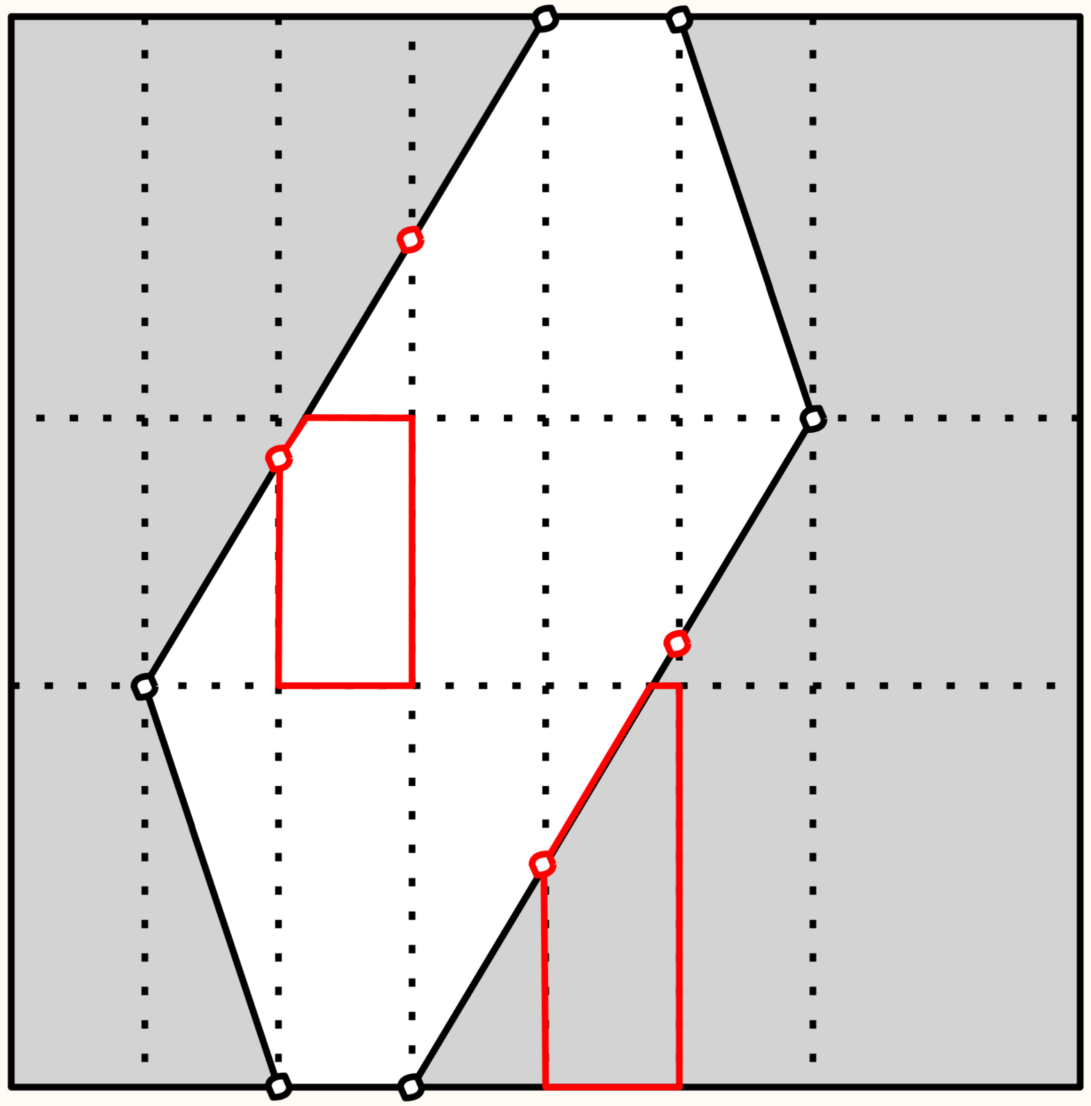
A polynomial time dynamic programming algorithm for two curves in $\mathbb{R}^2$ under L_1 and L_∞ norms (*)

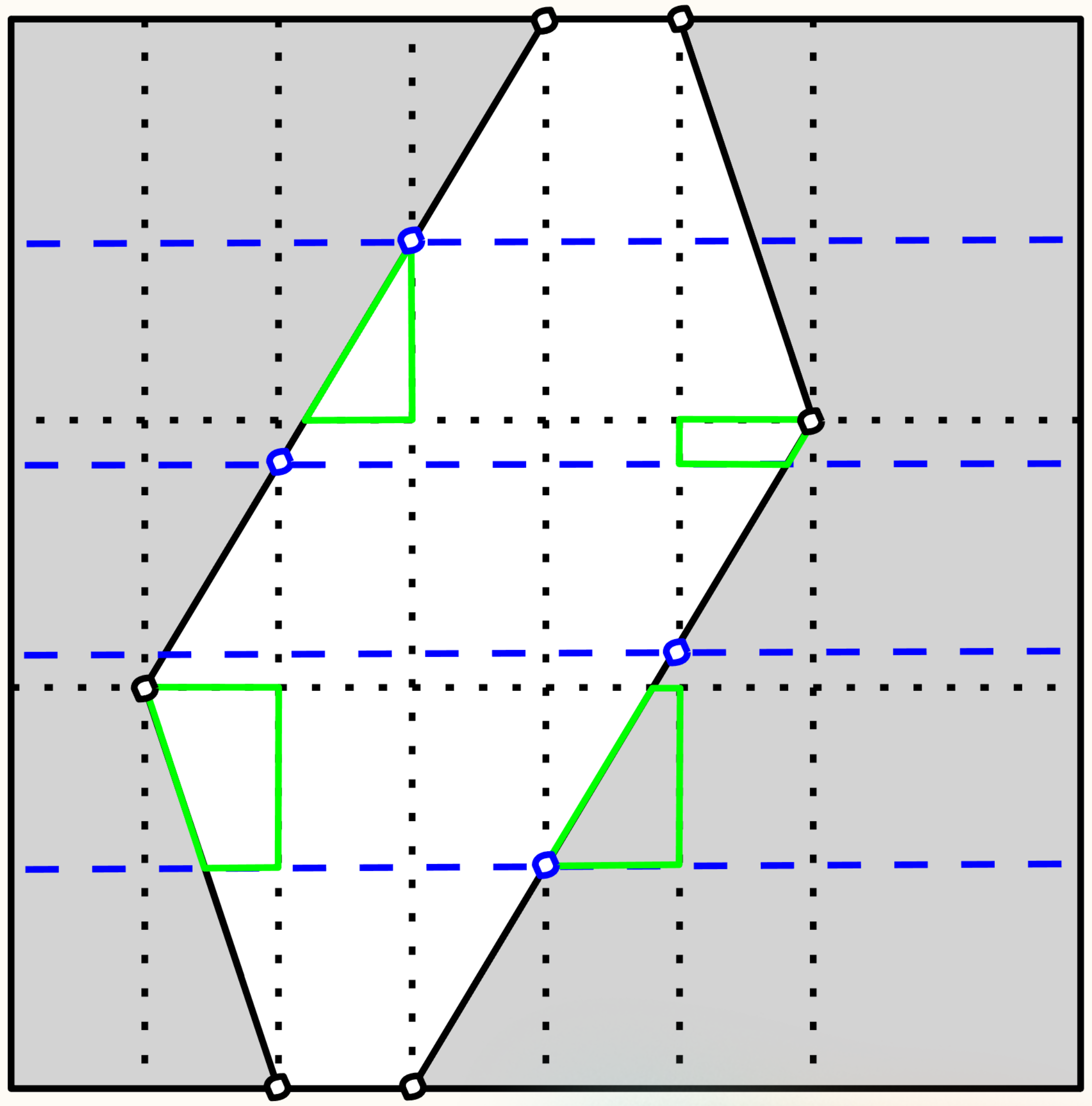
$$\mathcal{S}(x,y) = \mathbb{F}_{\varepsilon}(f|_{[0,x]}, g|_{[0,y]}) = \sup_{h:[0,x]\to[0,y]} \mathrm{len}(I_h^{\varepsilon})$$

---

*De Carufel, J.L., Gheibi, A., Maheshwari, A., Sack, J.R., Scheffer, C.: Similarity of polygonal curves in the presence of outliers. Comput. Geom. 47(5), 625–641 (2014). https://doi.org/10.1016/j.comgeo.2014.01.002

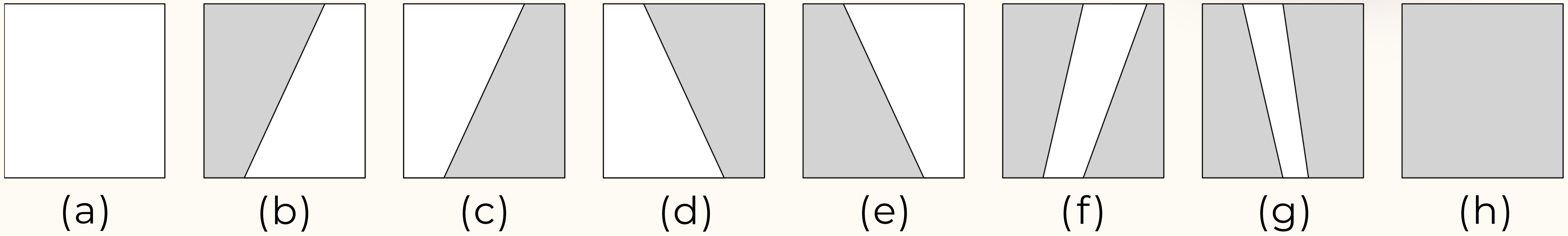
Refining the free-space diagram





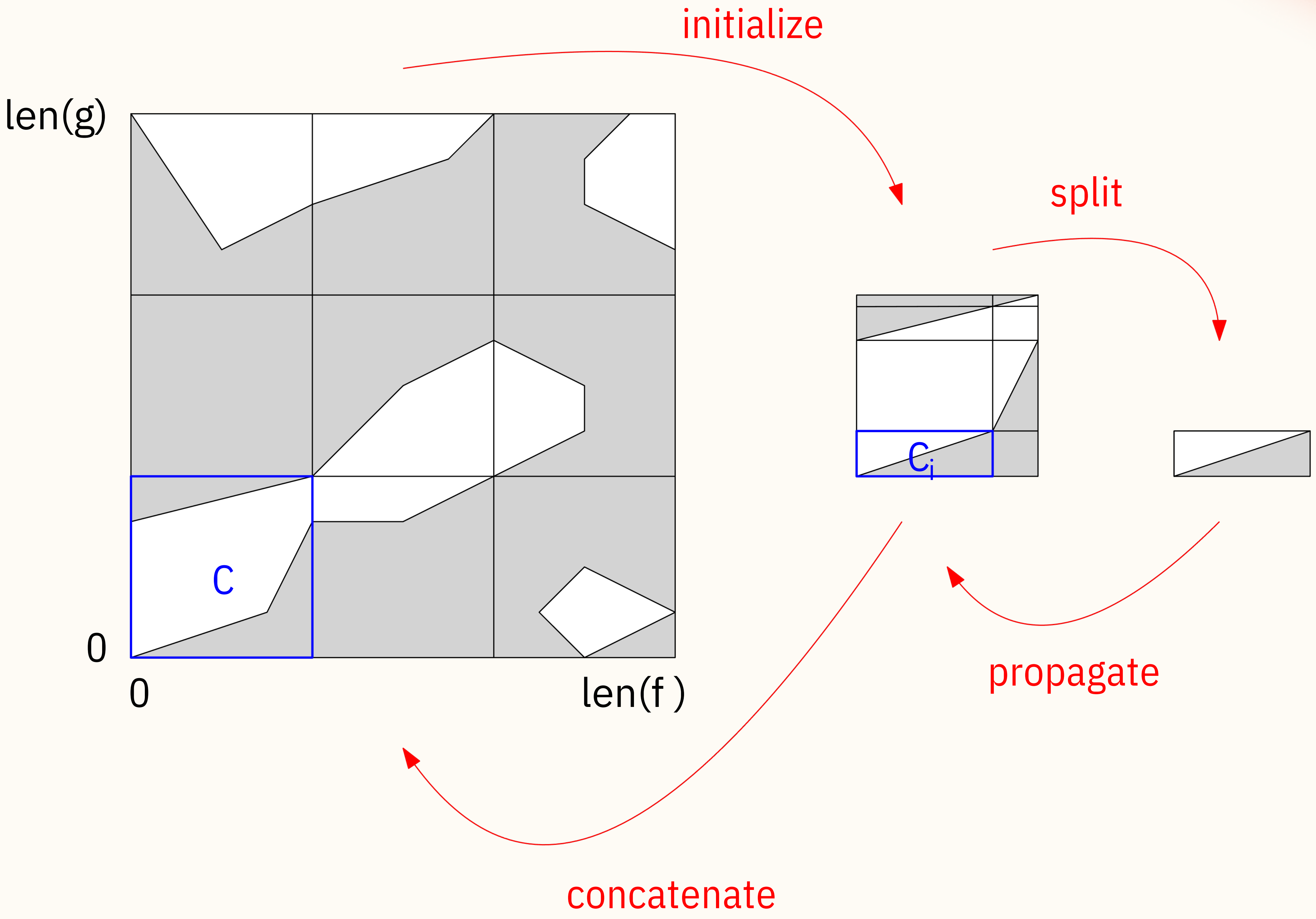
Refining the free-space diagram

Observation 1:



Lemma 1: If *f* and *g* consist of *m* and *n* segments respectively, the total number of refined cells is $\Theta(nm)$.

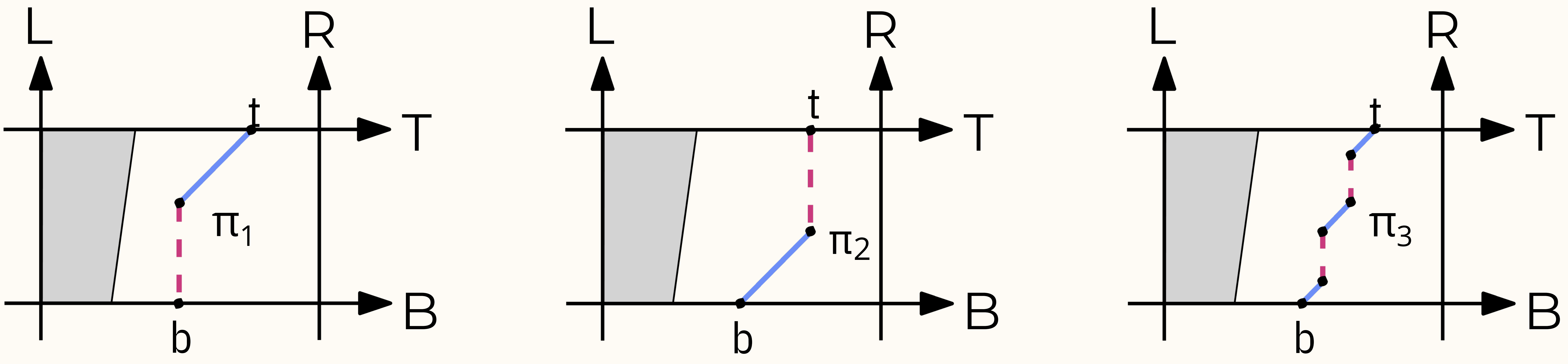
The Algorithm



Score Function within a cell

Observation 2:

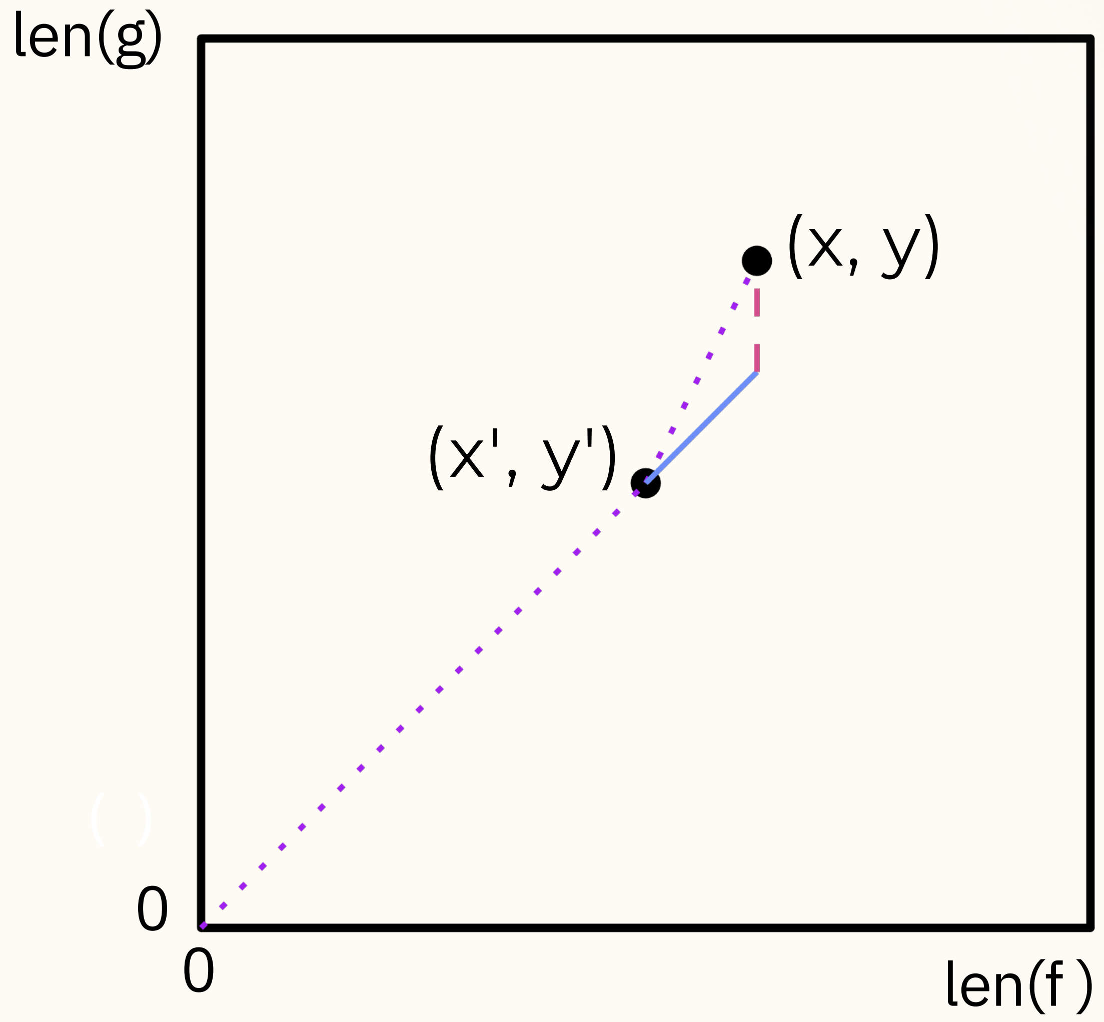
$$\mathcal{L}((x,y),(x+\Delta x, y+\Delta y)) \leq \min(\Delta x, \Delta y)$$



Properties of the score function

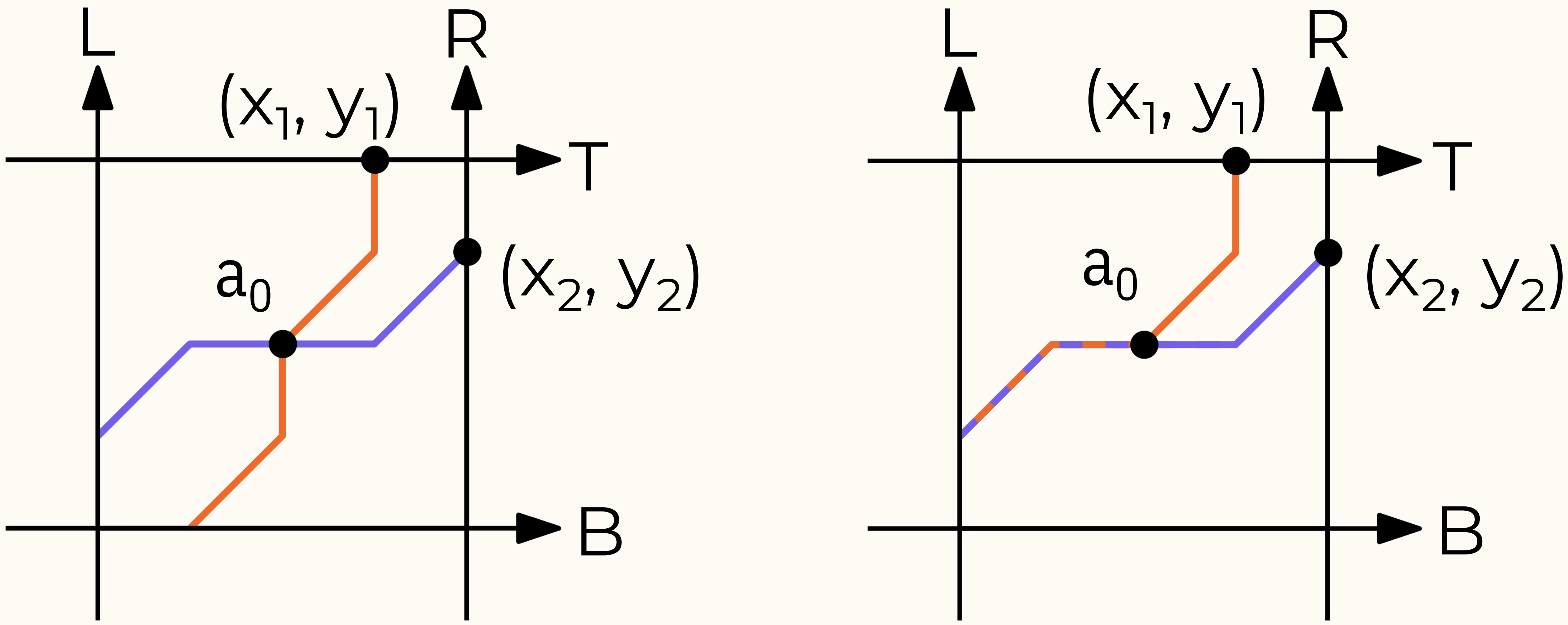
Observation 3 (Optimal Substructure):

$$\mathcal{S}(x,y) \leq \mathcal{S}(x',y') + \min(y-y', x-x')$$



Properties of the score function

Lemma 2 (Single Breakpoint):

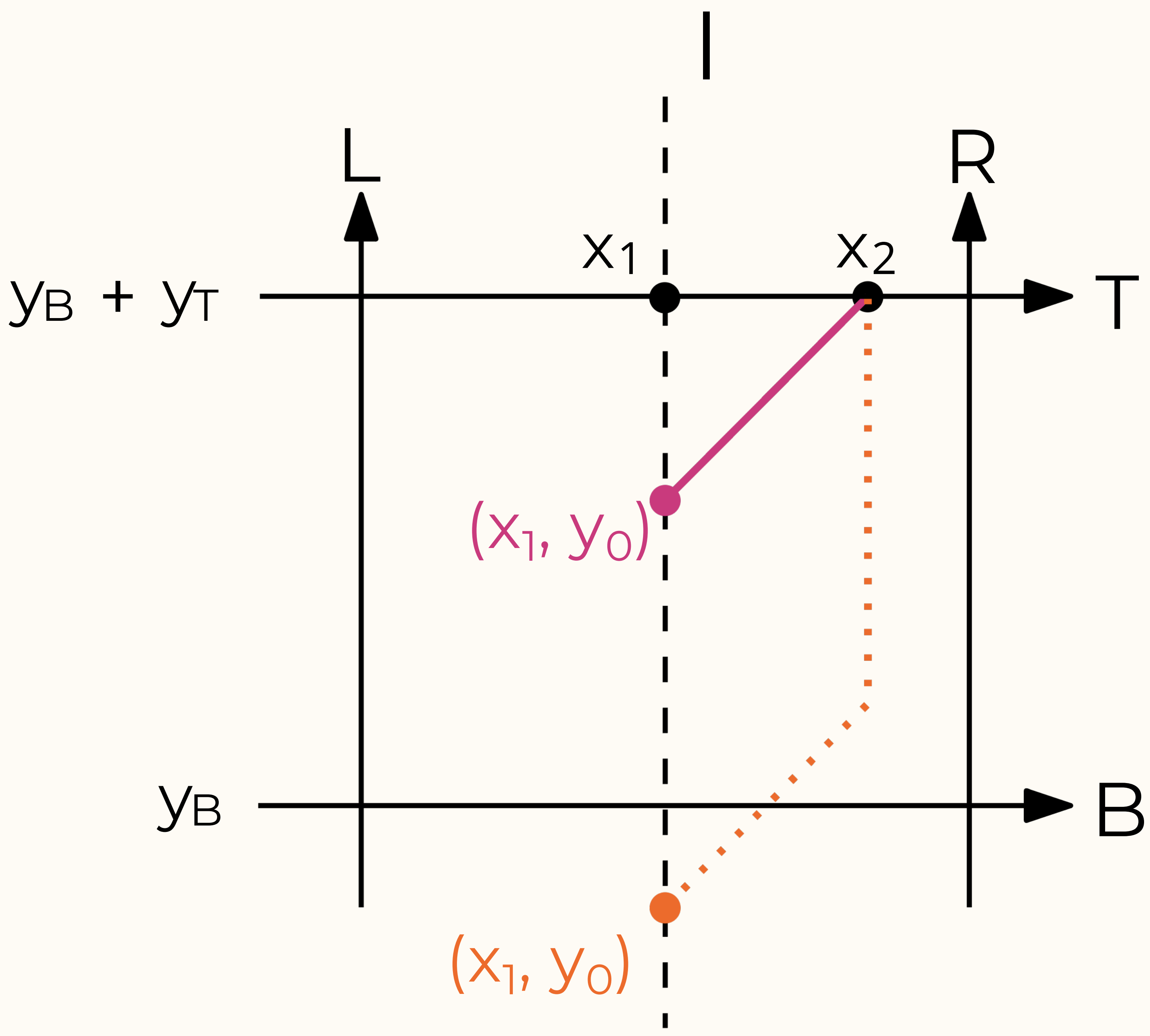


Properties of the score function

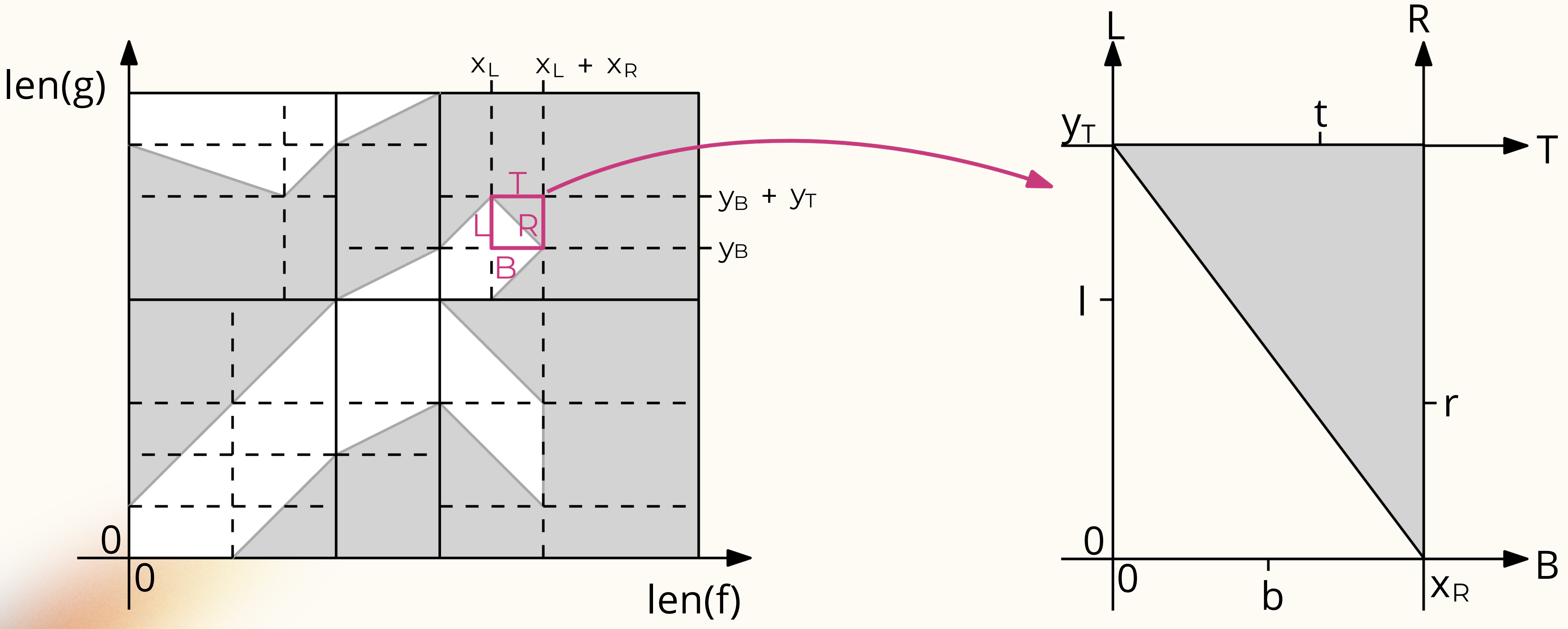
Lemma 3 (Slope Upper-Bounded):

The score function on a refined cell boundary is:

- Piecewise linear
- Each piece has slope less than or equal to 1



Propagation



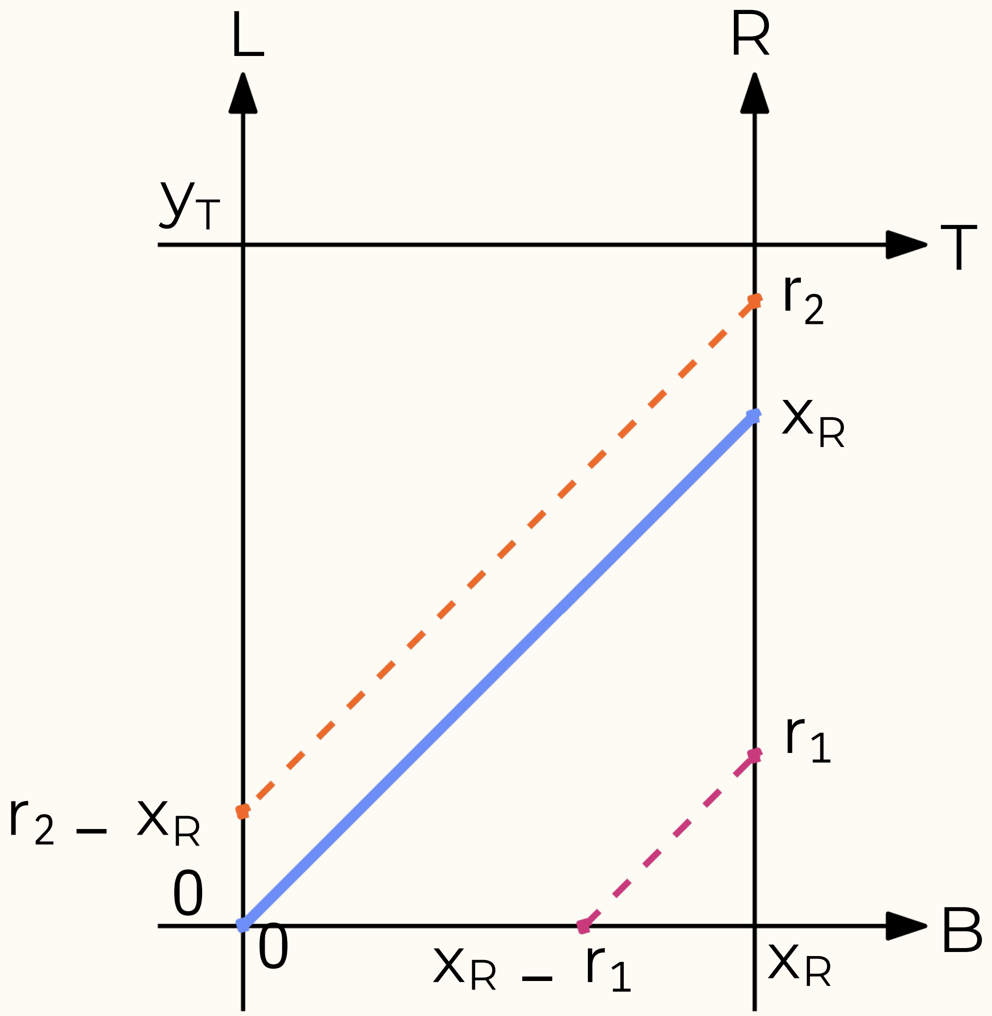
Propagation

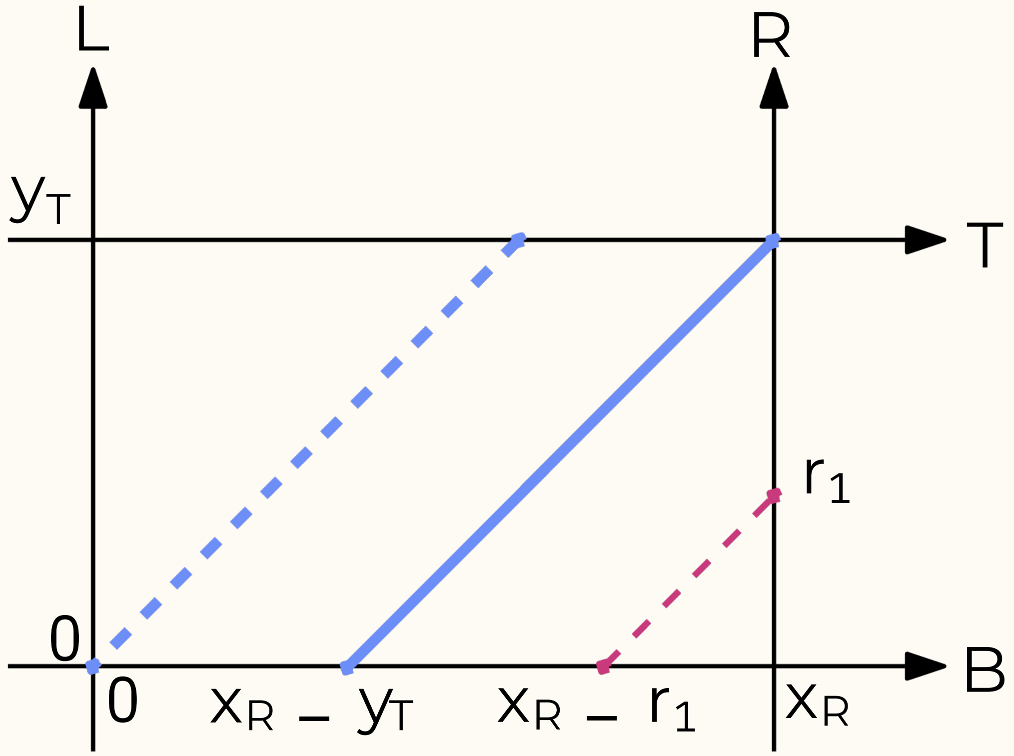
Case (a):

$$\mathcal{S}_{L\to R}(r) = \begin{cases} \mathcal{S}_L(0) + r & \text{for } r \le x_R \\ \mathcal{S}_L(r - x_R) + x_R & \text{for } r \ge x_R \end{cases}$$

$$\mathcal{S}_{B\to R}(r) = \begin{cases} \mathcal{S}_B(x_R - r) + r & \text{for } r \le x_R \\ \mathcal{S}_B(0) + x_R & \text{for } r \ge x_R \end{cases}$$

$$\mathcal{S}_R(r) = \max(\mathcal{S}_{L\to R}(r), \mathcal{S}_{B\to R}(r))$$





Analysis

Back to Overview

Total number of refined cells is in *O(mn)* (from Lemma 1)

The initialization and concatenation steps take time linear in the complexity of the involved score functions.

Propagation within a refined cell and computing the score functions on top and right boundaries add *O(1)* complexity (from Lemma 2)

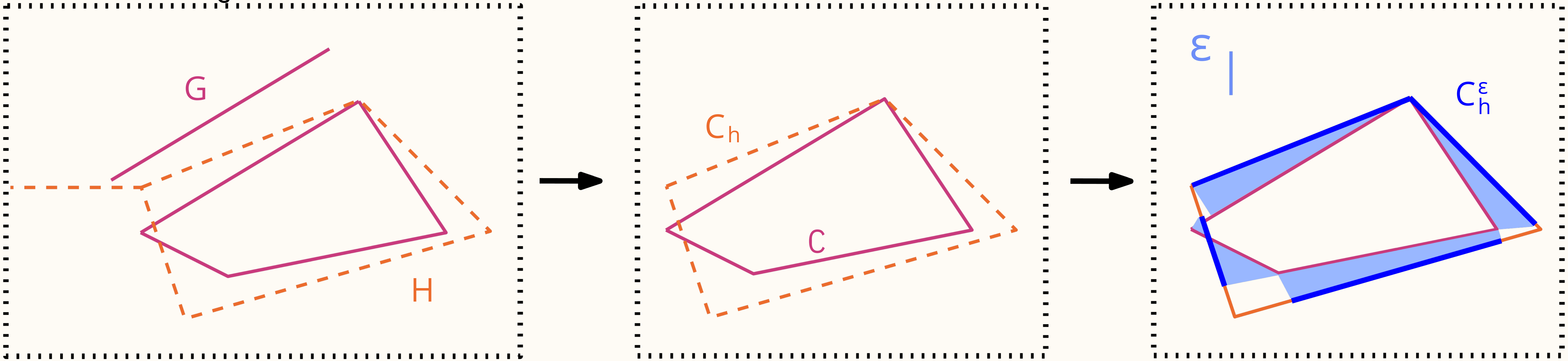
So the number of breakpoints in the cell *D[i][j]* is in *O(ij).* the complexity of the score functions on the top right cell is *O(mn)* and the total runtime of the DP is $O(m^2n^2)$.

LSFS for Graphs

Input
s

Homeomorphic subgraph

Output



LSFS for Graphs

Back to Overview

$$\boldsymbol{C_h^\varepsilon} = \{x \in C \mid ||\phi_G(x) - \phi_H(h(x))||_p \leq \varepsilon \text{ and } \exists \delta > 0 \colon h|_{\mathbb{B}_p(x,\delta)} \text{ is length-preserving}\}$$

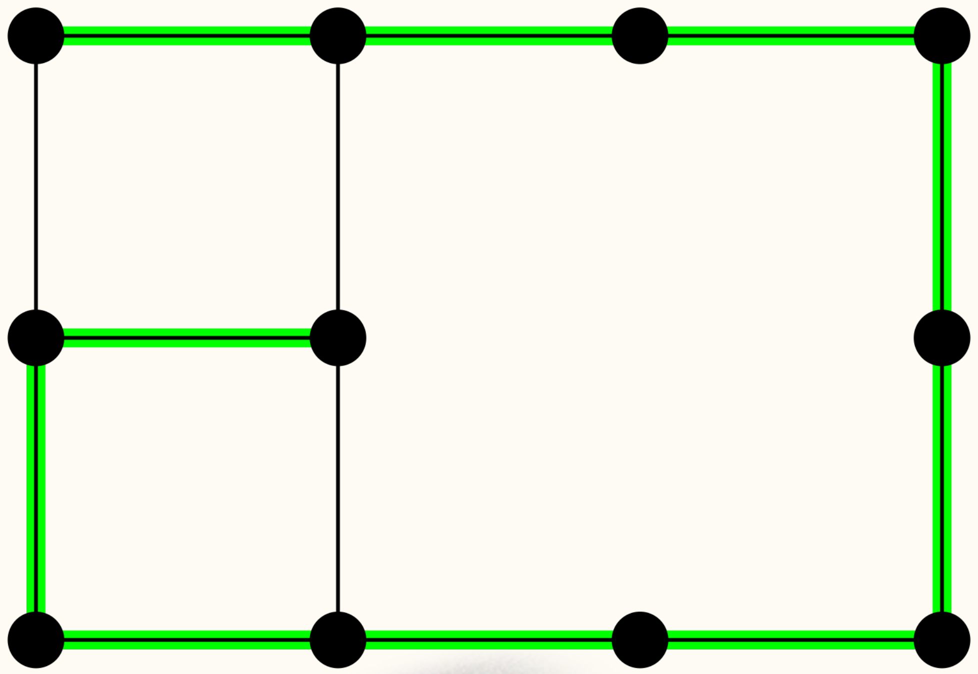
$$\mathbb{F}_\varepsilon(\boldsymbol{G}, \boldsymbol{H}) = \sup_{C \subset G} \sup_{h : C \to H} \operatorname{len}(C_h^\varepsilon)$$

LSFS for Graphs is NP-hard

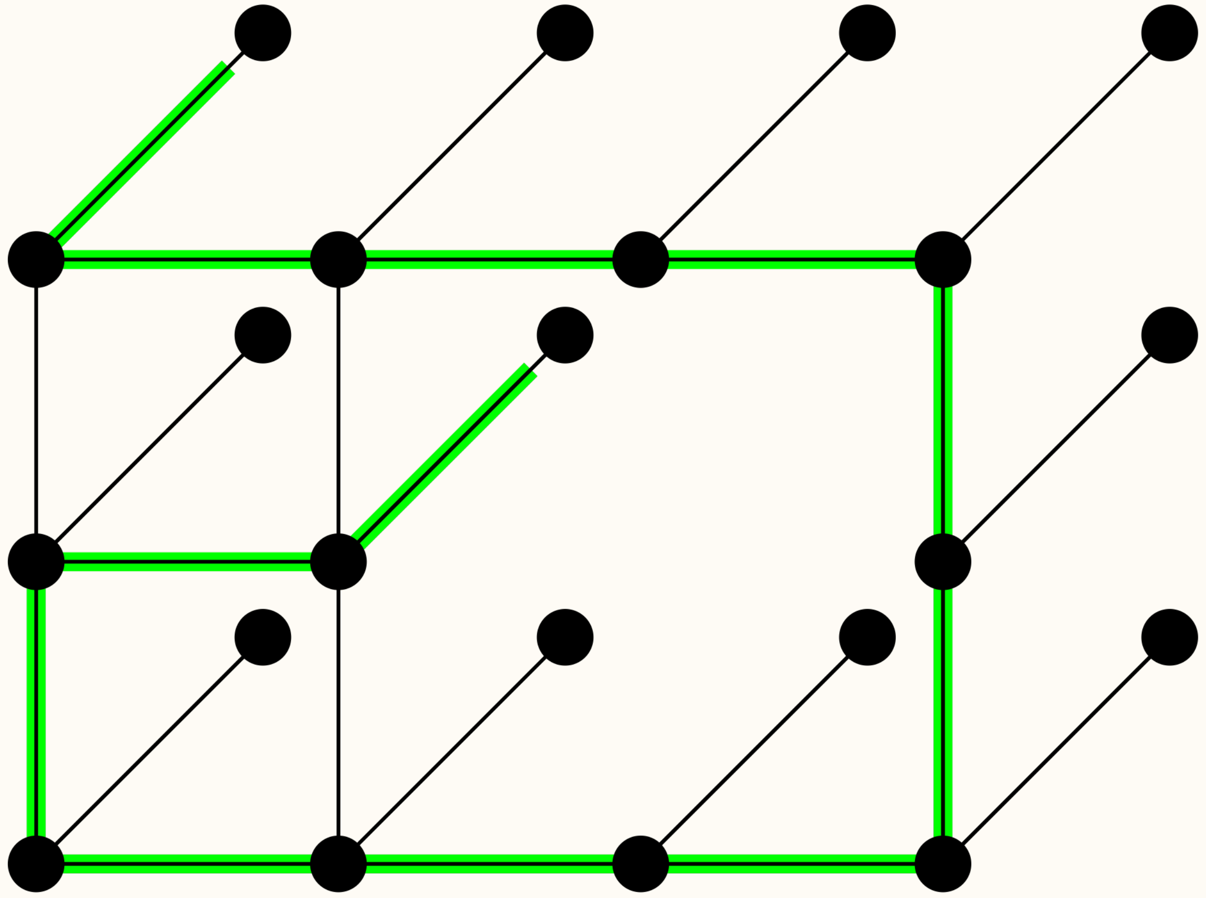
Back to Overview

We choose ε = n+1

V′ + (3/4, 3/4)



H′



H

We claim that if H′ has a Hamiltonian path then $F_\varepsilon(G, H) = n + 1$, and otherwise $F_\varepsilon(G, H) < n + 1/5$

Contributions and Open problems

Back to Overview

- Defined Length-sensitive Fréchet Similarity for Curves and Graphs in $\mathbb{R}^d$
- Presented an efficient algorithm for computing LSFS under L_1 and L_∞ norms for curves
- Showed NP-hardness for Graphs

- Finding an approximation algorithm for graphs
- Is there a poly time algorithm for trees?
- Is there a faster algorithm for curves in higher dimensions?

Thank You

I am nearing graduation and actively seeking opportunities in industry, national and university laboratories. If my skills align with your projects, please contact me on **erfanhosseini.com**