Mobile computing

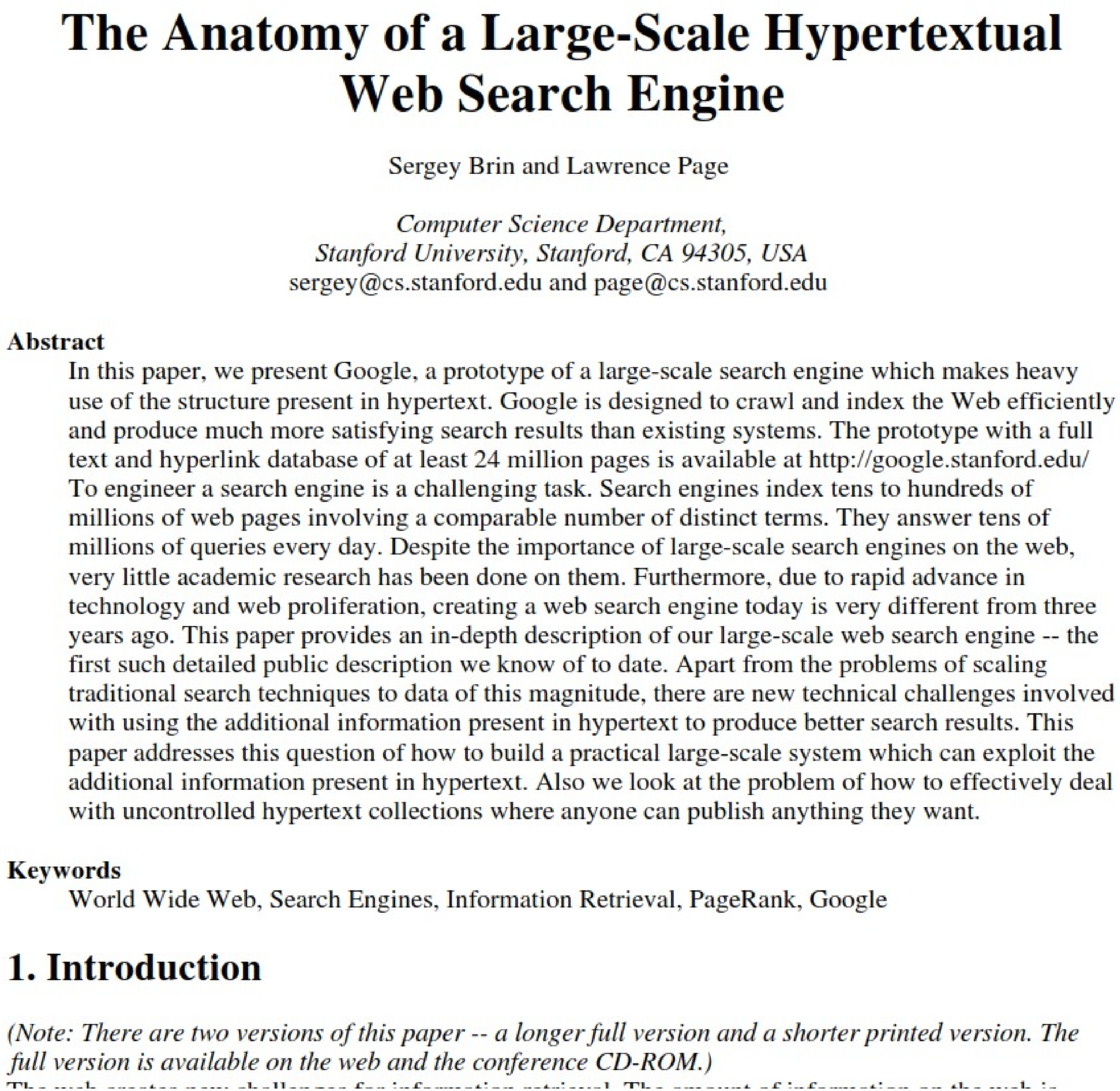
Hľadanie a porovnávanie trajektórií

Martin Drozda

Hľadanie a porovnávanie

2D: trajektórie ľudí a zariadení

3D: trajektórie dronov, riadenie letovej prevádzky drónov, účtovanie poplatkov za prelet dronov



The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/
To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago. This paper provides an in-depth description of our large-scale web search engine -- the first such detailed public description we know of to date. Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. This paper addresses this question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want.

Keywords

World Wide Web, Search Engines, Information Retrieval, PageRank, Google

1. Introduction

(Note: There are two versions of this paper -- a longer full version and a shorter printed version. The full version is available on the web and the conference CD-ROM.)
The web creates new challenges for information retrieval. The amount of information on the web is

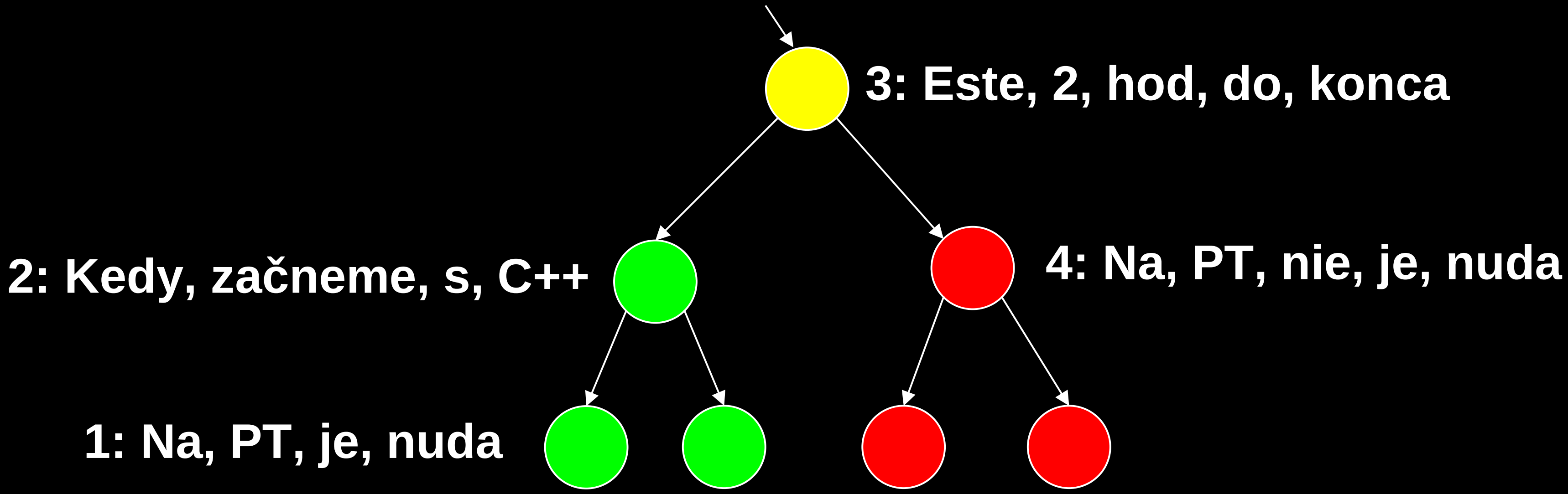
Inverzný zoznam

Doc id	Zoznam slov
1	Na, PT, je, nuda
2	Kedy, zacneme, s, C++
3	Este, 2, hod, do, konca
4	Na, PT, nie, je, nuda

Inverzný zoznam (index ako na konci knihy):

Slovo	Doc id
Na	1, 4
PT	1, 4
nie	4
je	1, 4
nuda	1, 4
…	

Zoznam



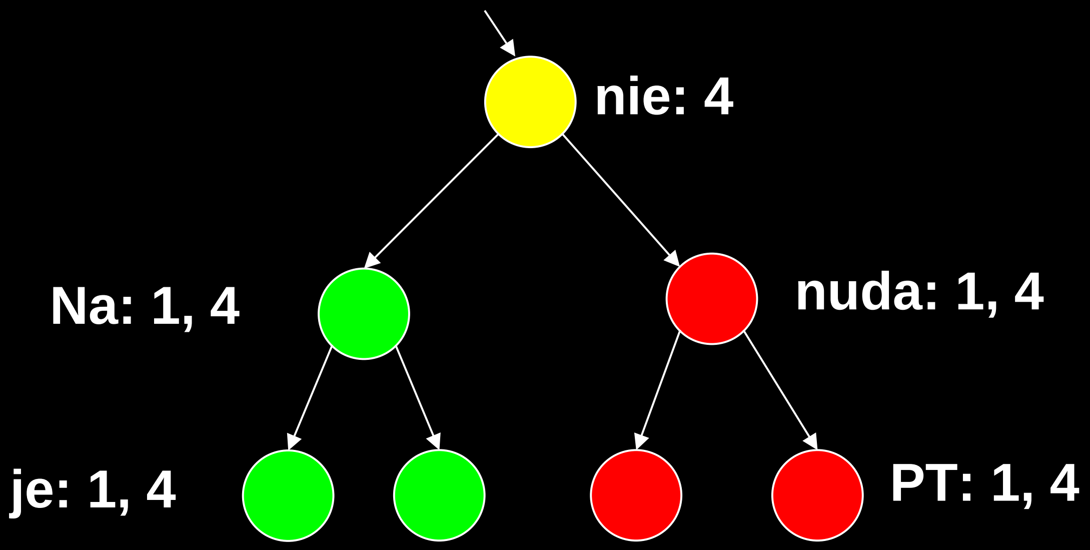
3: Este, 2, hod, do, konca

2: Kedy, začneme, s, C++

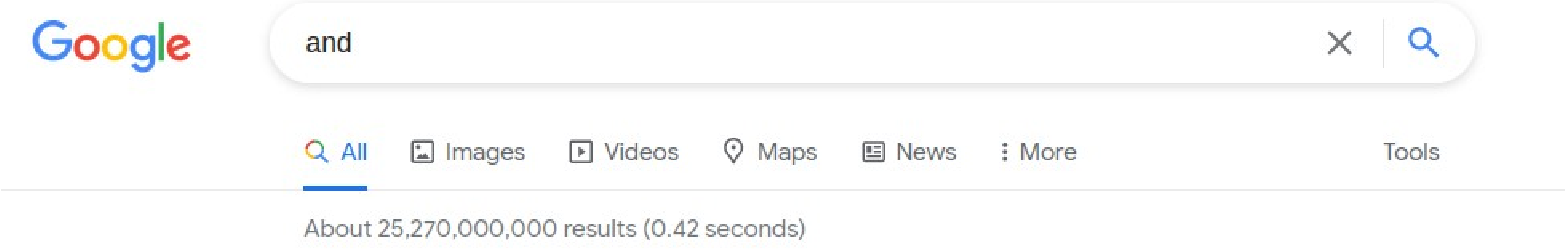
4: Na, PT, nie, je, nuda

1: Na, PT, je, nuda

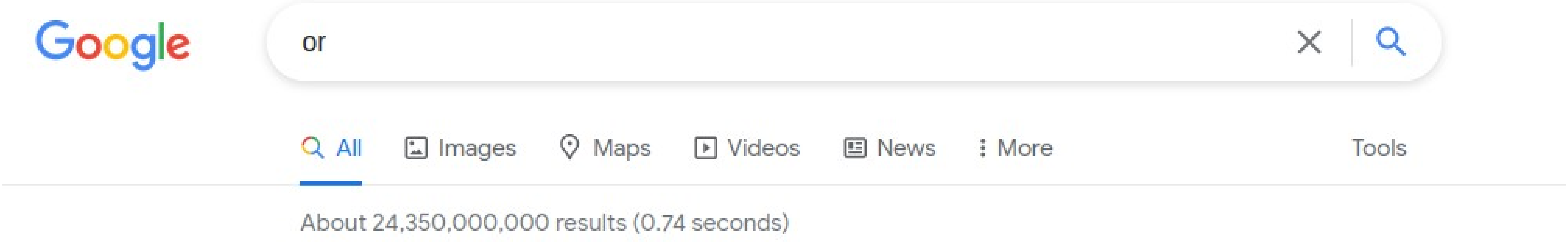
Inverzný zoznam



Hľadanie



About 25,270,000,000 results (0.42 seconds)



About 24,350,000,000 results (0.74 seconds)

Relácia podobnosti

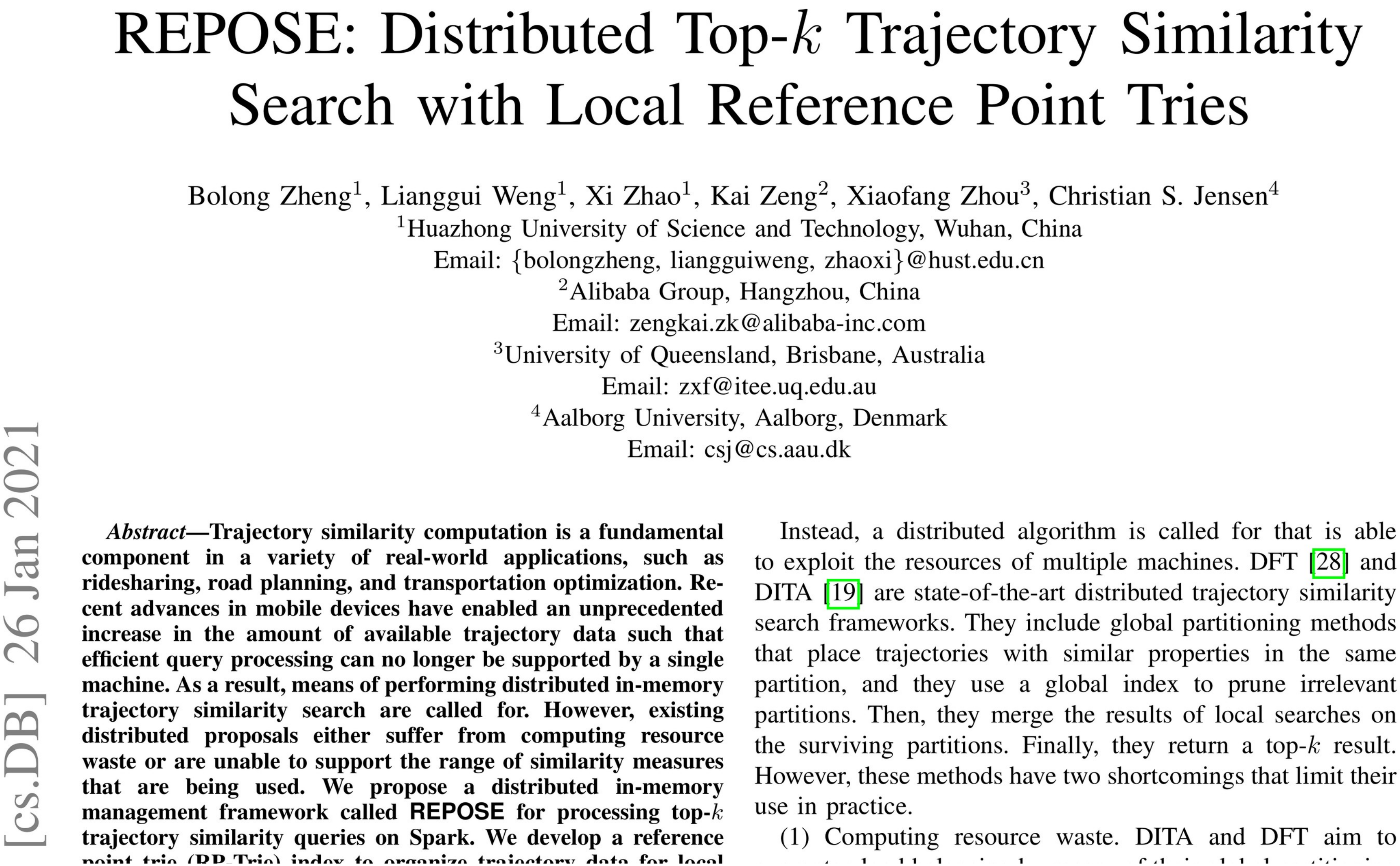
Hľadanie v texte: dokumenty sú podobné, ak obsahujú hľadané slovné frázy

- Inverzný zoznam

Hľadanie trajektórií: trajektórie sú podobné, ak ich vzdialenosť je menšia ako γ

- Hausdorffova vzdialenosť
- Vzdialenosť viem vypočítať, ak preiterujem cez všetky pozície trajektórie

REPOSE



REPOSE: Distributed Top-k Trajectory Similarity Search with Local Reference Point Tries

Bolong Zheng[1], Lianggui Weng[1], Xi Zhao[1], Kai Zeng[2], Xiaofang Zhou[3], Christian S. Jensen[4]
[1]Huazhong University of Science and Technology, Wuhan, China
Email: {bolongzheng, liangguiweng, zhaoxi}@hust.edu.cn
[2]Alibaba Group, Hangzhou, China
Email: zengkai.zk@alibaba-inc.com
[3]University of Queensland, Brisbane, Australia
Email: zxf@itee.uq.edu.au
[4]Aalborg University, Aalborg, Denmark
Email: csj@cs.aau.dk

***Abstract*—Trajectory similarity computation is a fundamental component in a variety of real-world applications, such as ridesharing, road planning, and transportation optimization. Recent advances in mobile devices have enabled an unprecedented increase in the amount of available trajectory data such that efficient query processing can no longer be supported by a single machine. As a result, means of performing distributed in-memory trajectory similarity search are called for. However, existing distributed proposals either suffer from computing resource waste or are unable to support the range of similarity measures that are being used. We propose a distributed in-memory management framework called REPOSE for processing top-k trajectory similarity queries on Spark. We develop a reference point trie (RP-Trie) index to organize trajectory data for local**

Instead, a distributed algorithm is called for that is able to exploit the resources of multiple machines. DFT [28] and DITA [19] are state-of-the-art distributed trajectory similarity search frameworks. They include global partitioning methods that place trajectories with similar properties in the same partition, and they use a global index to prune irrelevant partitions. Then, they merge the results of local searches on the surviving partitions. Finally, they return a top-k result. However, these methods have two shortcomings that limit their use in practice.

(1) Computing resource waste. DITA and DFT aim to

Hausdorffova vzdialenosť

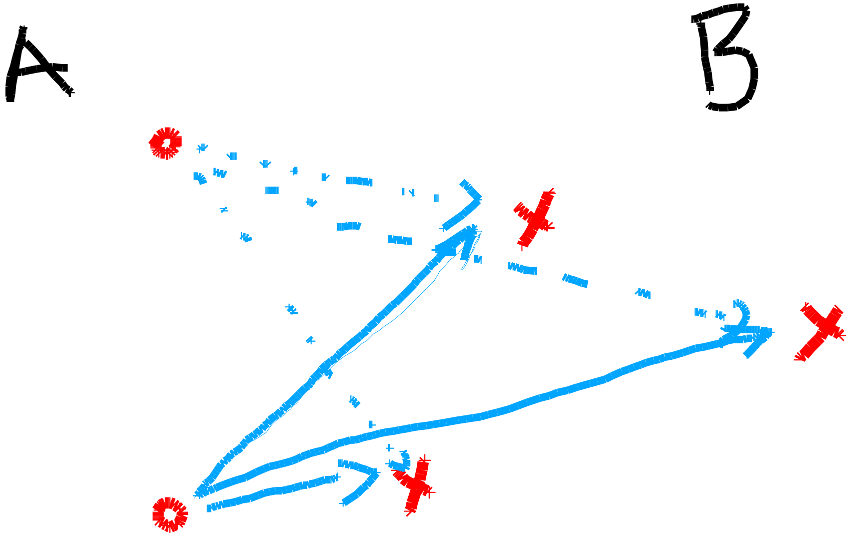
Hausdorffova vzdialenosť H(A, B) = maximálna vzdialenosť množiny bodov A k najbližšiemu bodu množiny B, maximálna vzdialenosť množiny bodov B k najbližšiemu bodu množiny A, maximum týchto dvoch vzdialeností

$$h(A,B) = \max_{a \in A} \min_{b \in B} d(a,b)$$

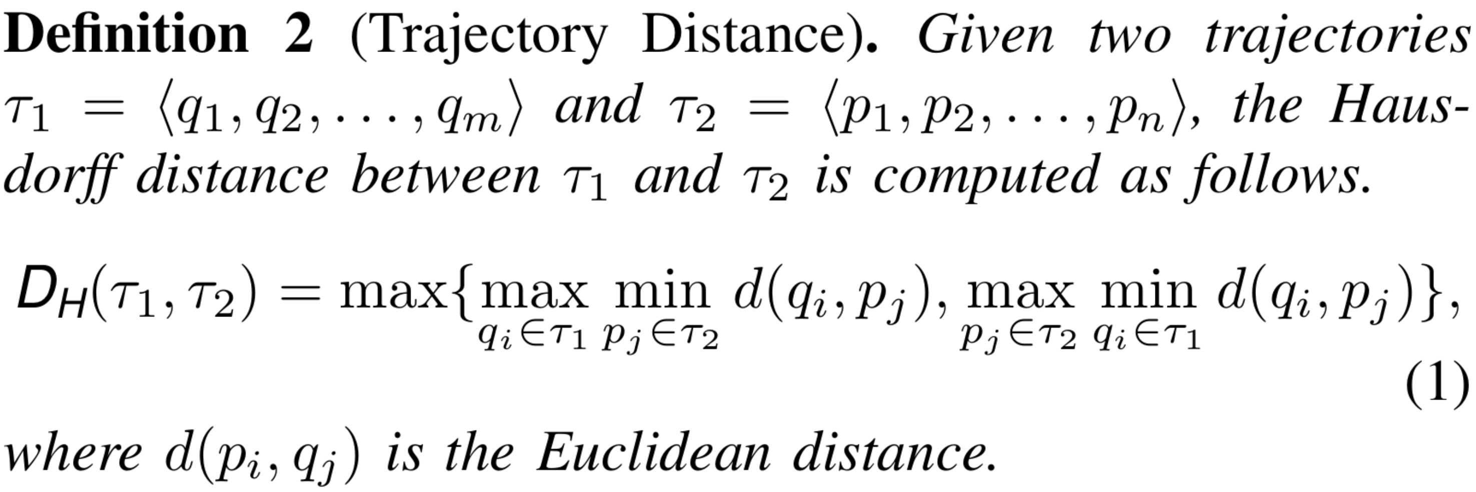
$$h(B,A) = \max_{b \in B} \min_{a \in A} d(a,b)$$

$$H(A,B) = \max\{h(A,B), h(B,A)\}$$

Hausdorffova vzdialenosť h(A, B)



REPOSE: podobnosť



Definition 2 (Trajectory Distance). *Given two trajectories* $\tau_1 = \langle q_1, q_2, \ldots, q_m \rangle$ *and* $\tau_2 = \langle p_1, p_2, \ldots, p_n \rangle$, *the Hausdorff distance between* τ_1 *and* τ_2 *is computed as follows.*

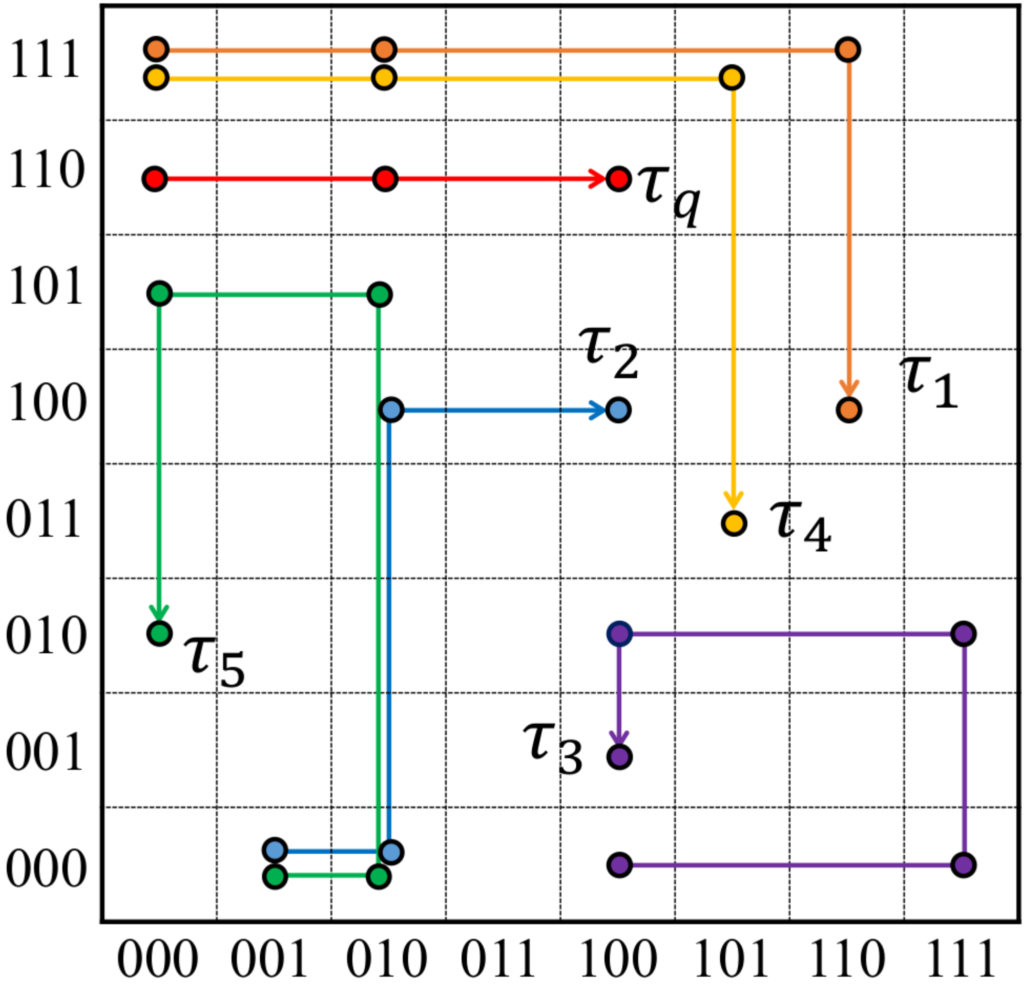
$$D_H(\tau_1, \tau_2) = \max\{\max_{q_i \in \tau_1} \min_{p_j \in \tau_2} d(q_i, p_j), \max_{p_j \in \tau_2} \min_{q_i \in \tau_1} d(q_i, p_j)\}, \tag{1}$$

where $d(p_i, q_j)$ *is the Euclidean distance.*

Obr. zdroj: ZHENG, Bolong, et al. REPOSE: Distributed Top-k Trajectory Similarity Search with Local Reference Point Tries. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021. p. 708-719.

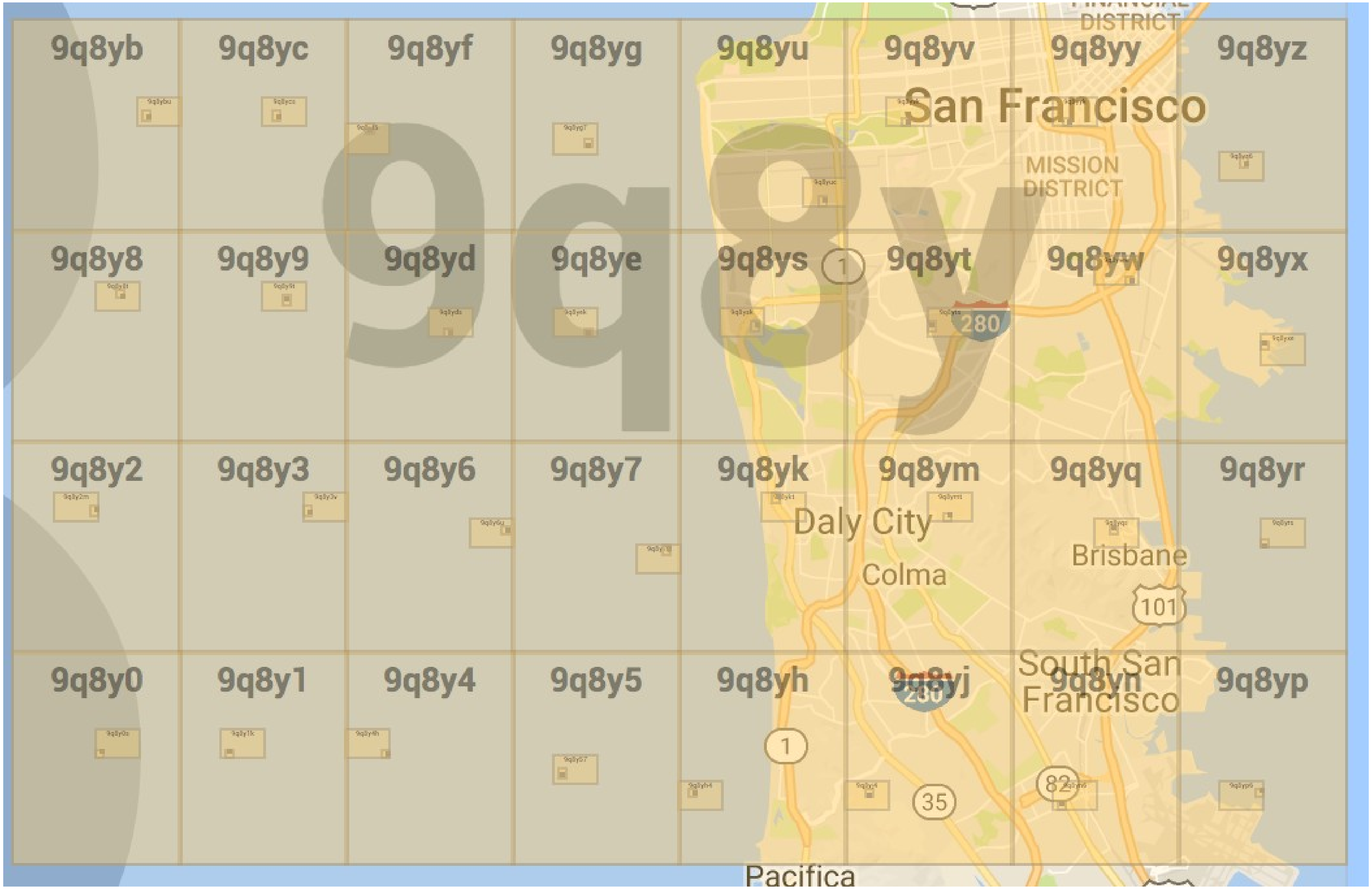
Z-order

Vytvorenie referenčnej trajektórie pomocou Z-usporiadania



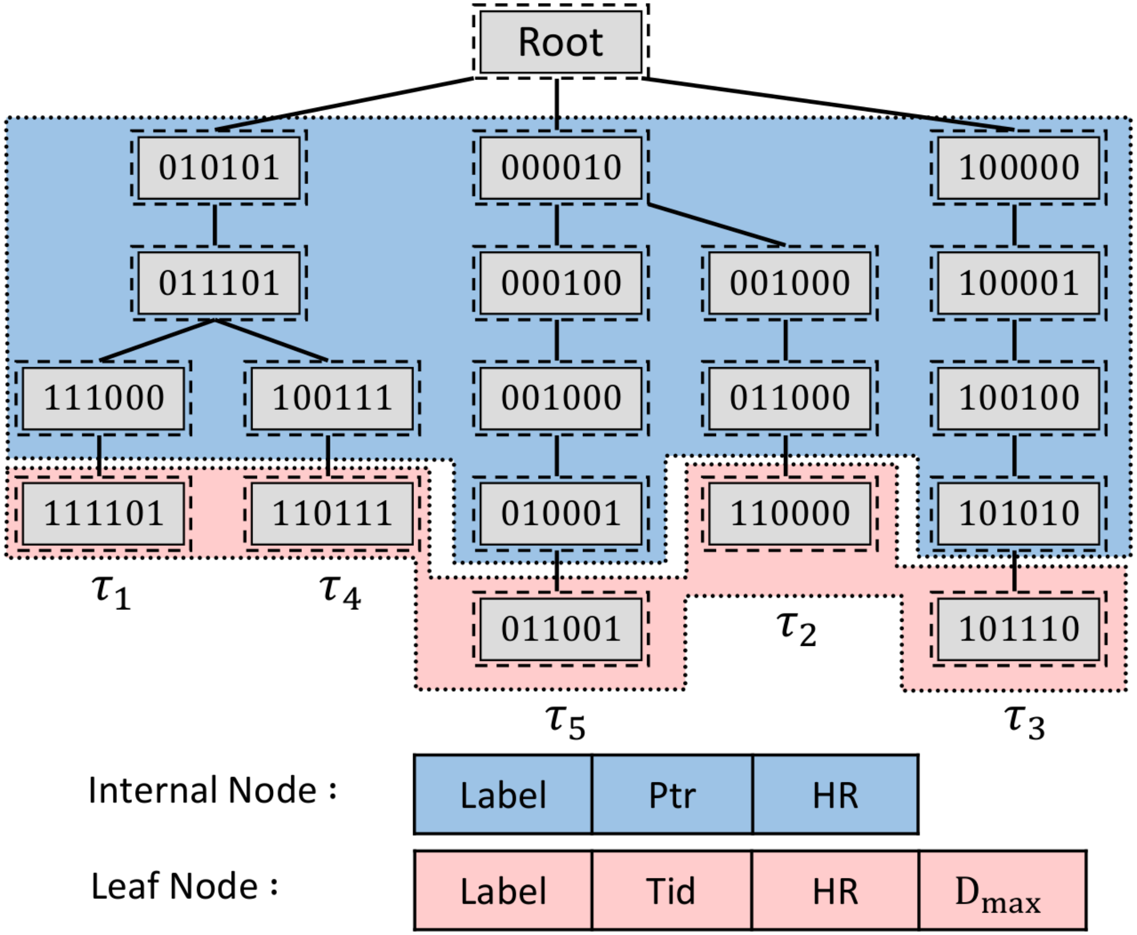
Obr. zdroj: ZHENG, Bolong, et al. REPOSE: Distributed Top-k Trajectory Similarity Search with Local Reference Point Tries. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021. p. 708-719.

Geohashing



Obr. zdroj: https://medium.com/@bkawk/geohashing-20b282fc9655

REPOSE algoritmus



Obr. zdroj: ZHENG, Bolong, et al. REPOSE: Distributed Top-k Trajectory Similarity Search with Local Reference Point Tries. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021. p. 708-719.

REPOSE algoritmus (sketch)

1. Vypočítaj referenčné trajektórie.

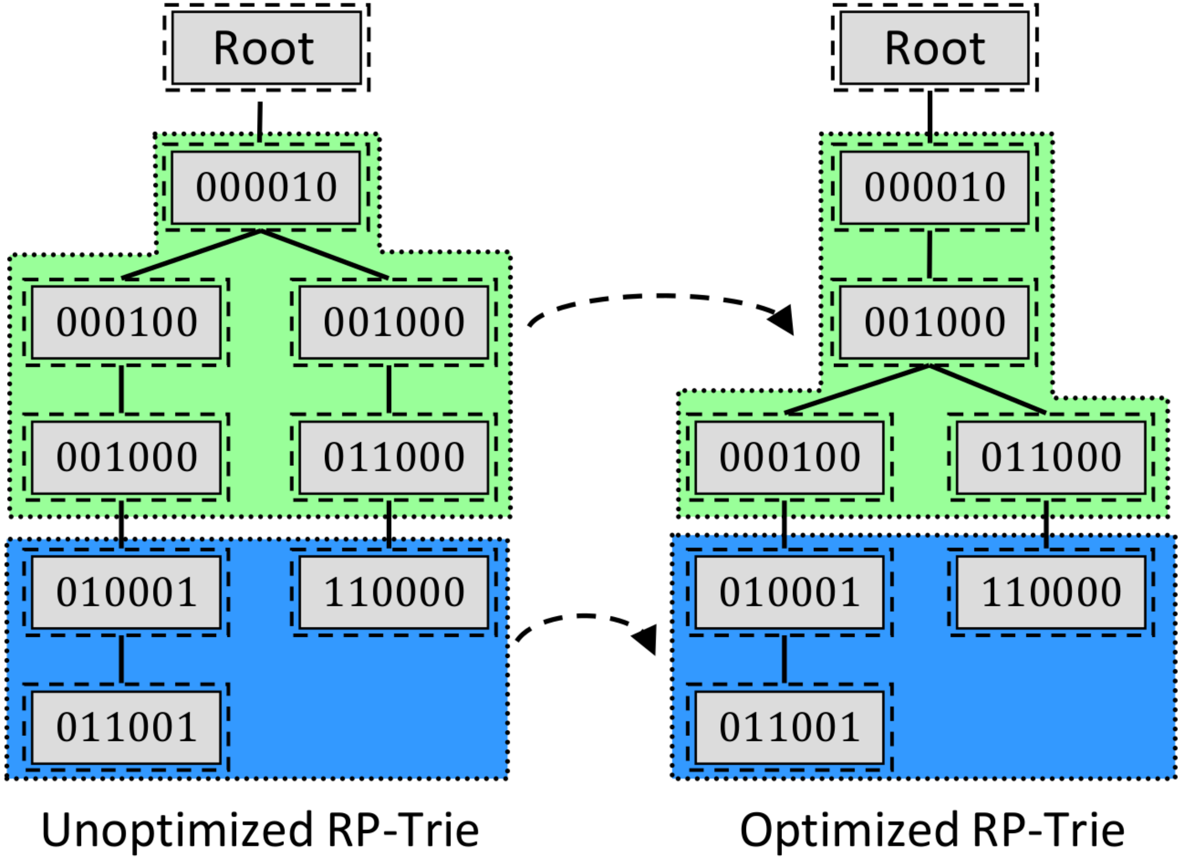
2. Z množiny referenčných trajektórií náhodne vyber nízky počet „pivot“ trajektórií (ktoré sú nepodobné).

3. Vlož pivot trajektórie do stromu: každá pivot trajektória má priradené trajektórie podobné tejto pivot trajektórii.

4. Prechádzaj cez strom a porovnávaj pivot trajektórie s hľadanou trajektóriou.

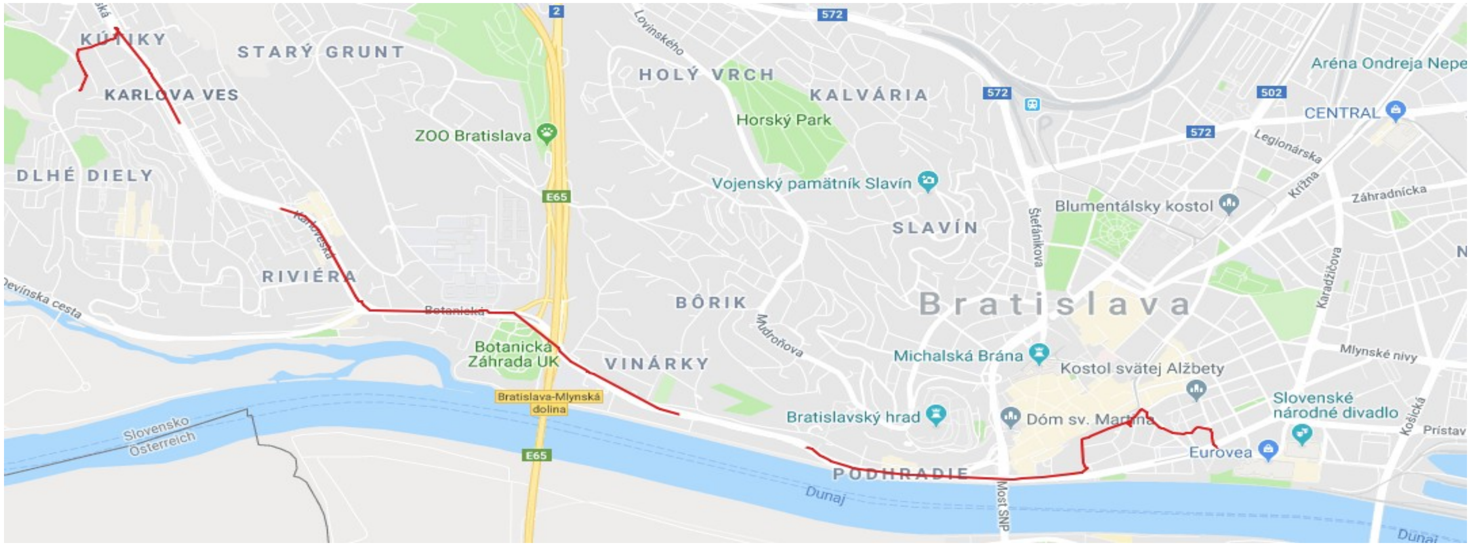
Obr. zdroj: ZHENG, Bolong, et al. REPOSE: Distributed Top-k Trajectory Similarity Search with Local Reference Point Tries. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021. p. 708-719.

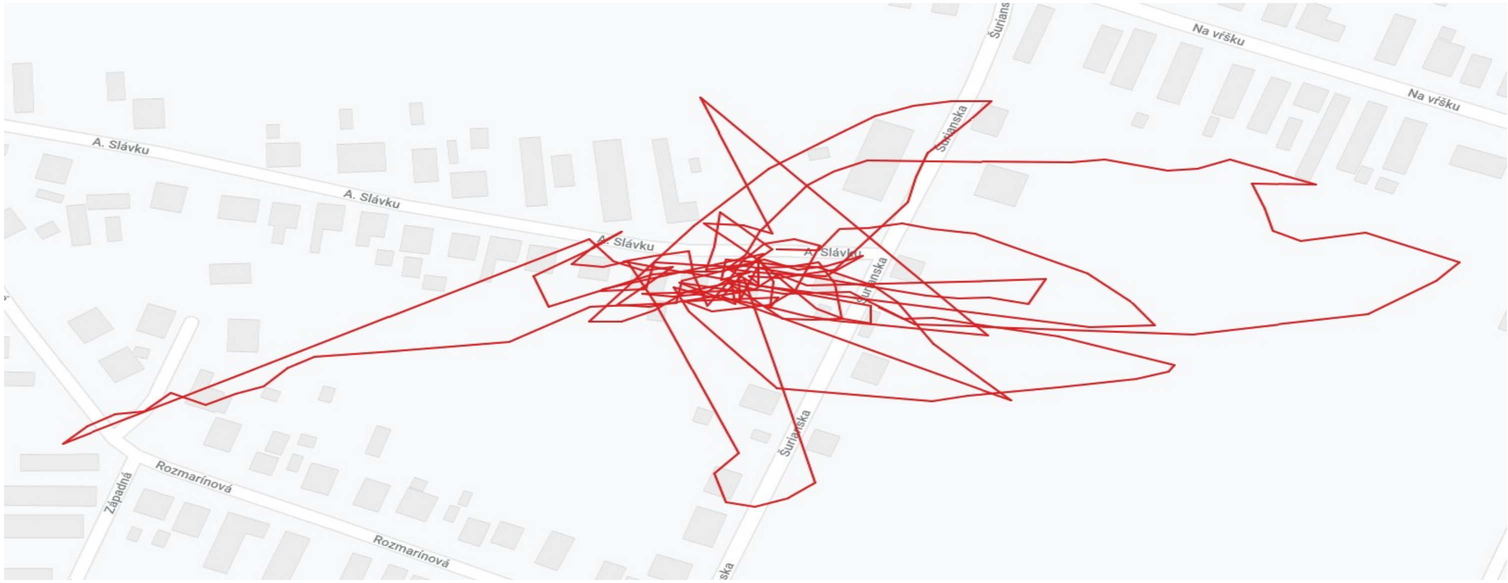
REPOSE algoritmus



Obr. zdroj: ZHENG, Bolong, et al. REPOSE: Distributed Top-k Trajectory Similarity Search with Local Reference Point Tries. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021. p. 708-719.

Gap vs nest


KÚTIKY
KARLOVA VES
STARÝ GRUNT
DLHÉ DIELY
RIVIÉRA
ZOO Bratislava
Botanická Záhrada UK
Bratislava-Mlynská dolina
VINÁRKY
BÔRIK
HOLÝ VRCH
KALVÁRIA
Horský Park
Vojenský pamätník Slavín
SLAVÍN
Bratislava
Michalská Brána
Kostol svätej Alžbety
Bratislavský hrad
Dóm sv. Martina
PODHRADIE
Dunaj
Most SNP
Eurovea
Slovenské národné divadlo
Blumentálsky kostol
CENTRAL
Aréna Ondreja Nepe
Slovensko
Österreich
Devínska cesta
E65


A. Slávku
Šurianska
Na vŕšku
Rozmarínová
Západná

Dijkstra, best first search

Zarovnanie

Sequence 1:	G	T	C	G	A	C	G
Sequence 2:	G	A	T	T	A	C	A

Sequence 1:	G	–	T	–	–	C	G	A	C	G
Sequence 2:	G	A	T	T	A	C	–	A	–	–

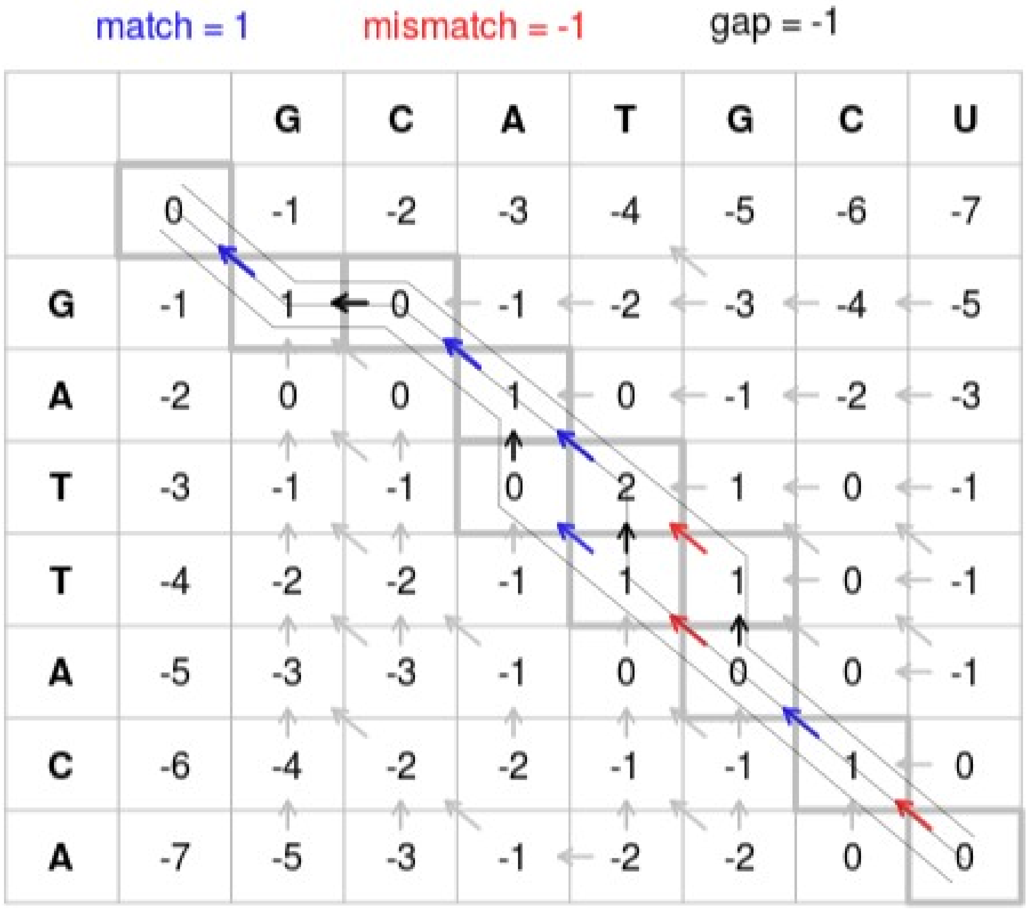
Sequence 1:	G	–	T	C	G	A	C	G
Sequence 2:	G	A	T	T	–	A	C	A

mismatch

Needleman-Wunsch algorithm

$$M(i,j) = max \begin{cases} M(i-1, j-1) + score(i,j), \\ M(i-1, j) + score(i, \_), \\ M(i, j-1) + score(\_, j), \end{cases}$$

Needleman-Wunsch algorithm

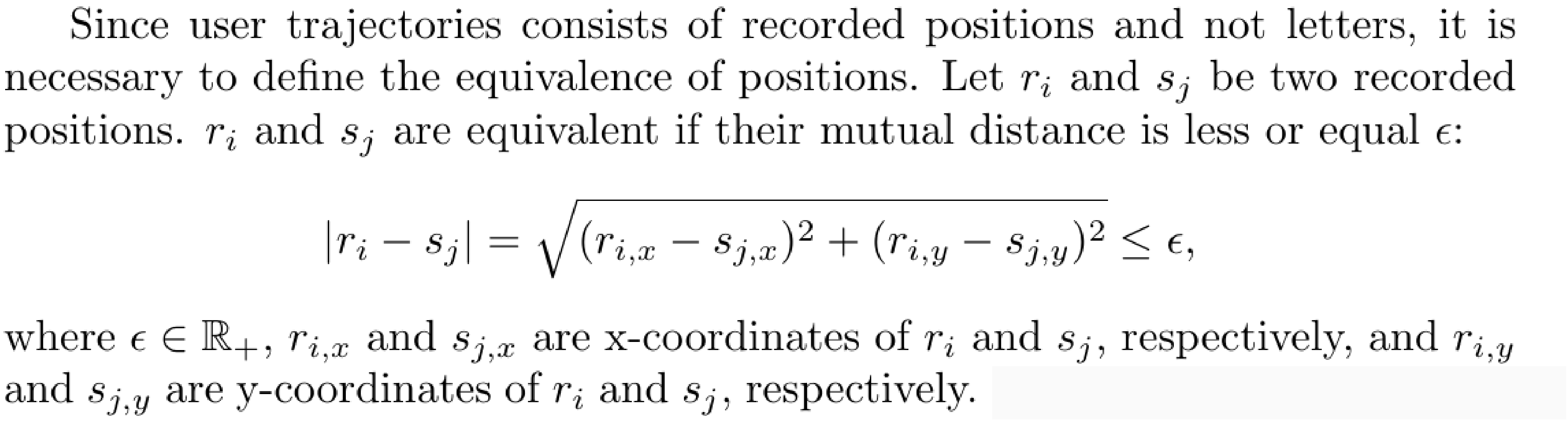


match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Zdroj: Slowkow - Own work, CC0, https://commons.wikimedia.org/w/index.php?curid=31963972

Konverzia

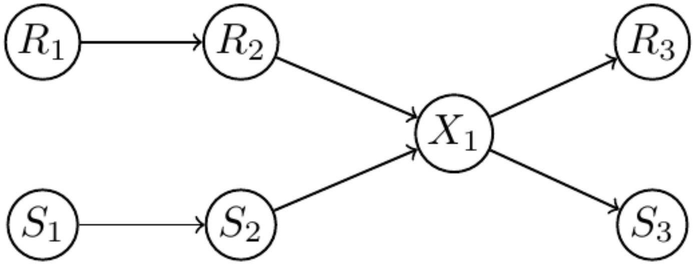


Since user trajectories consists of recorded positions and not letters, it is necessary to define the equivalence of positions. Let r_i and s_j be two recorded positions. r_i and s_j are equivalent if their mutual distance is less or equal ϵ:

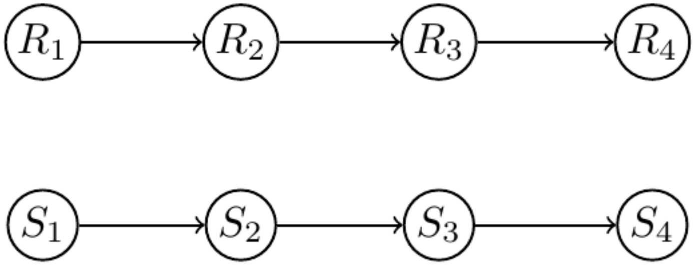
$$|r_i - s_j| = \sqrt{(r_{i,x} - s_{j,x})^2 + (r_{i,y} - s_{j,y})^2} \leq \epsilon,$$

where $\epsilon \in \mathbb{R}_+$, $r_{i,x}$ and $s_{j,x}$ are x-coordinates of r_i and s_j, respectively, and $r_{i,y}$ and $s_{j,y}$ are y-coordinates of r_i and s_j, respectively.

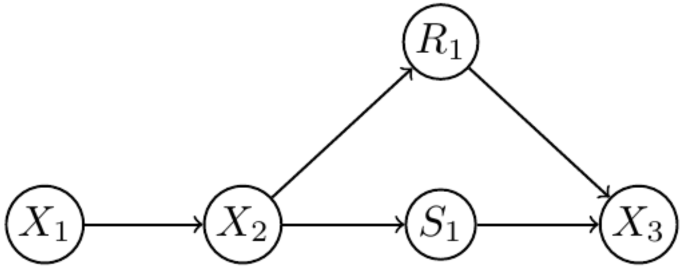
Porovnávanie



(a) Intersecting trajectories



(b) Non-intersecting trajectories



(c) Similar trajectories

Zarovnávanie

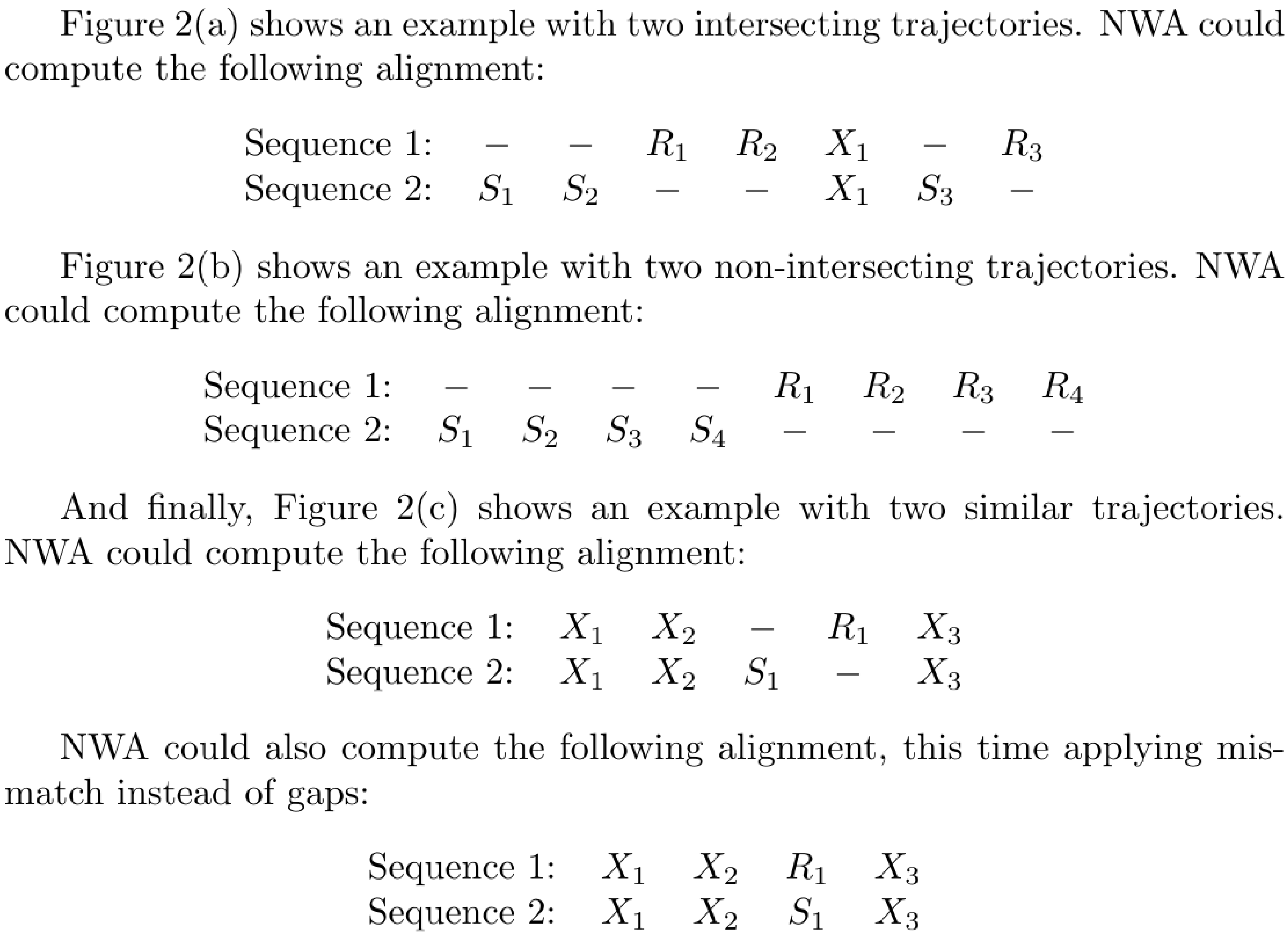


Figure 2(a) shows an example with two intersecting trajectories. NWA could compute the following alignment:

Sequence 1:	–	–	R_1	R_2	X_1	–	R_3
Sequence 2:	S_1	S_2	–	–	X_1	S_3	–

Figure 2(b) shows an example with two non-intersecting trajectories. NWA could compute the following alignment:

Sequence 1:	–	–	–	–	R_1	R_2	R_3	R_4
Sequence 2:	S_1	S_2	S_3	S_4	–	–	–	–

And finally, Figure 2(c) shows an example with two similar trajectories. NWA could compute the following alignment:

Sequence 1:	X_1	X_2	–	R_1	X_3
Sequence 2:	X_1	X_2	S_1	–	X_3

NWA could also compute the following alignment, this time applying mismatch instead of gaps:

Sequence 1:	X_1	X_2	R_1	X_3
Sequence 2:	X_1	X_2	S_1	X_3

Match, mismatch, gap

$$\phi(|r_i - s_j|) = \frac{1}{\epsilon\sqrt{2\pi}} e^{-\frac{1}{2\epsilon^2}|r_i - s_j|^2},$$

$$match = \phi(|r_i - s_j|)^{-1} = \phi(0)^{-1} = \epsilon\sqrt{2\pi},$$

$$mismatch = -\phi(|r_i - s_j|)^{-1} = -\epsilon\sqrt{2\pi} e^{\frac{1}{2\epsilon^2}|r_i - s_j|^2},$$

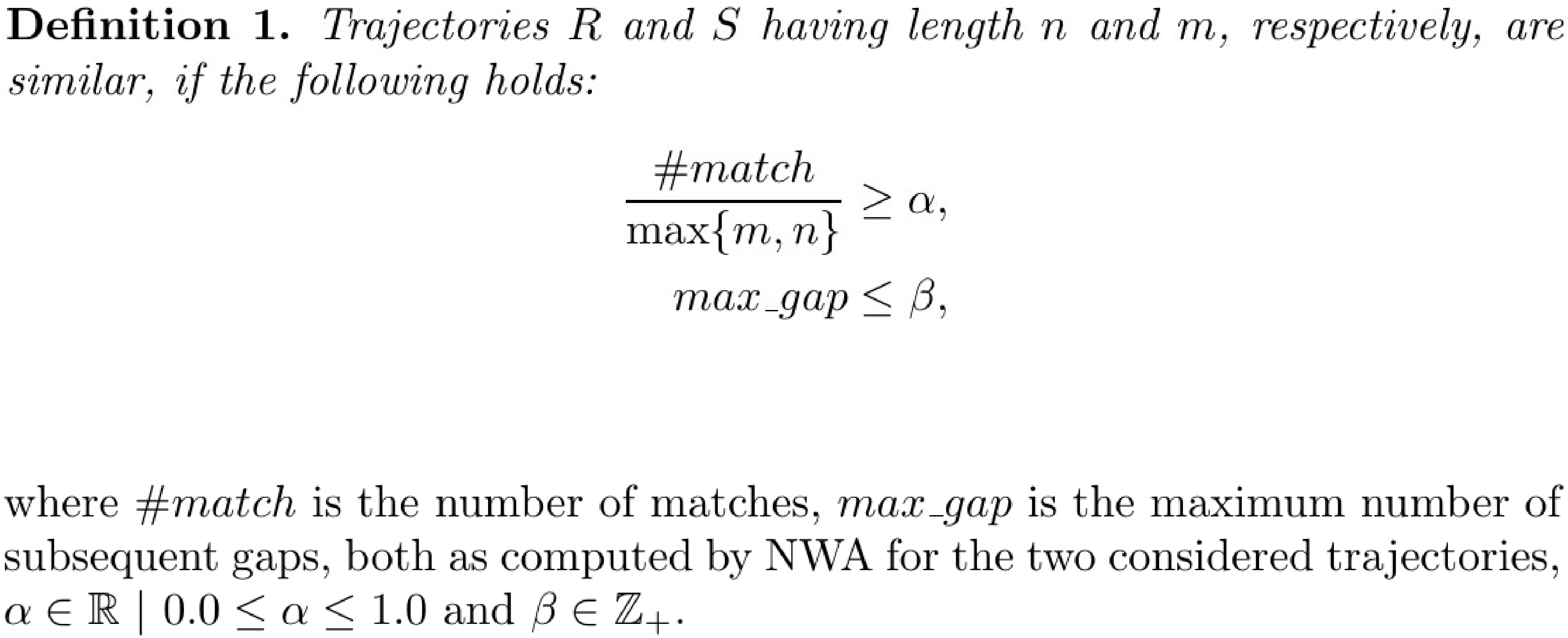
$$match = 1,$$

$$mismatch = \left\lceil -e^{\frac{1}{2\epsilon^2}|r_i - s_j|^2} \right\rceil.$$

$$\begin{aligned} match &= 1, \quad \text{if } |r_i - s_j| \leq \epsilon, \\ mismatch &= \left\lceil -e^{\frac{1}{2\epsilon^2}|r_i - s_j|^2} \right\rceil \leq -1, \quad \text{if } |r_i - s_j| > \epsilon. \end{aligned}$$

$$(match = 1) > (gp = 0) > mismatch = \{-1, \cdots, -k\},$$

Podobnosť



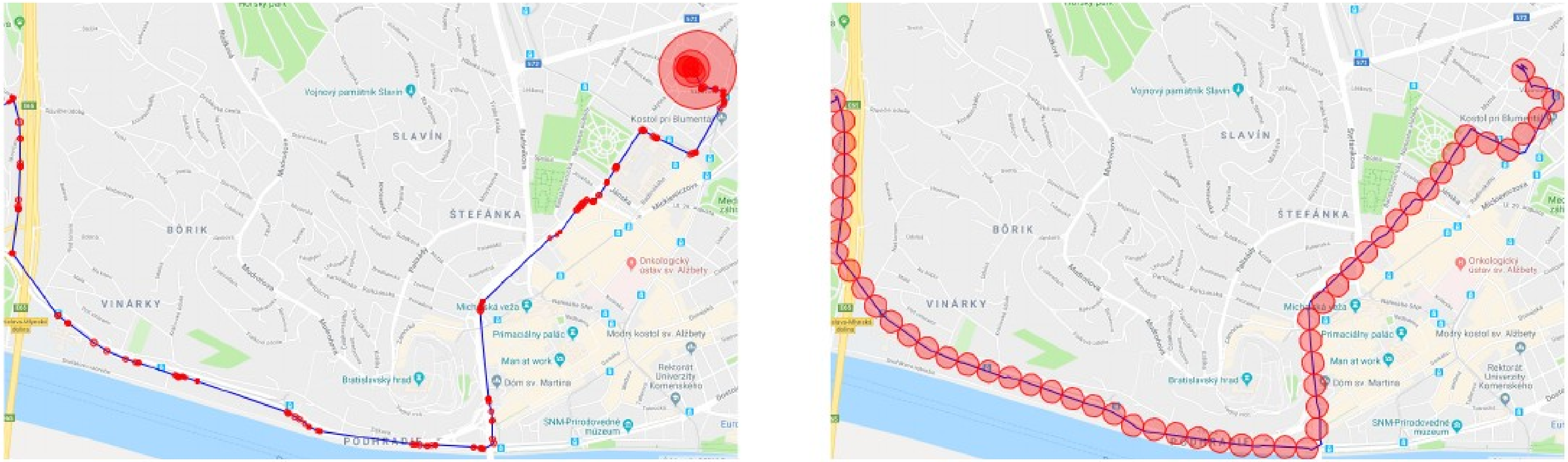
Definition 1. *Trajectories R and S having length n and m, respectively, are similar, if the following holds:*

$$\frac{\#match}{\max\{m, n\}} \geq \alpha,$$

$$max\_gap \leq \beta,$$

where $\#match$ is the number of matches, max_gap is the maximum number of subsequent gaps, both as computed by NWA for the two considered trajectories, $\alpha \in \mathbb{R} \mid 0.0 \leq \alpha \leq 1.0$ and $\beta \in \mathbb{Z}_+$.

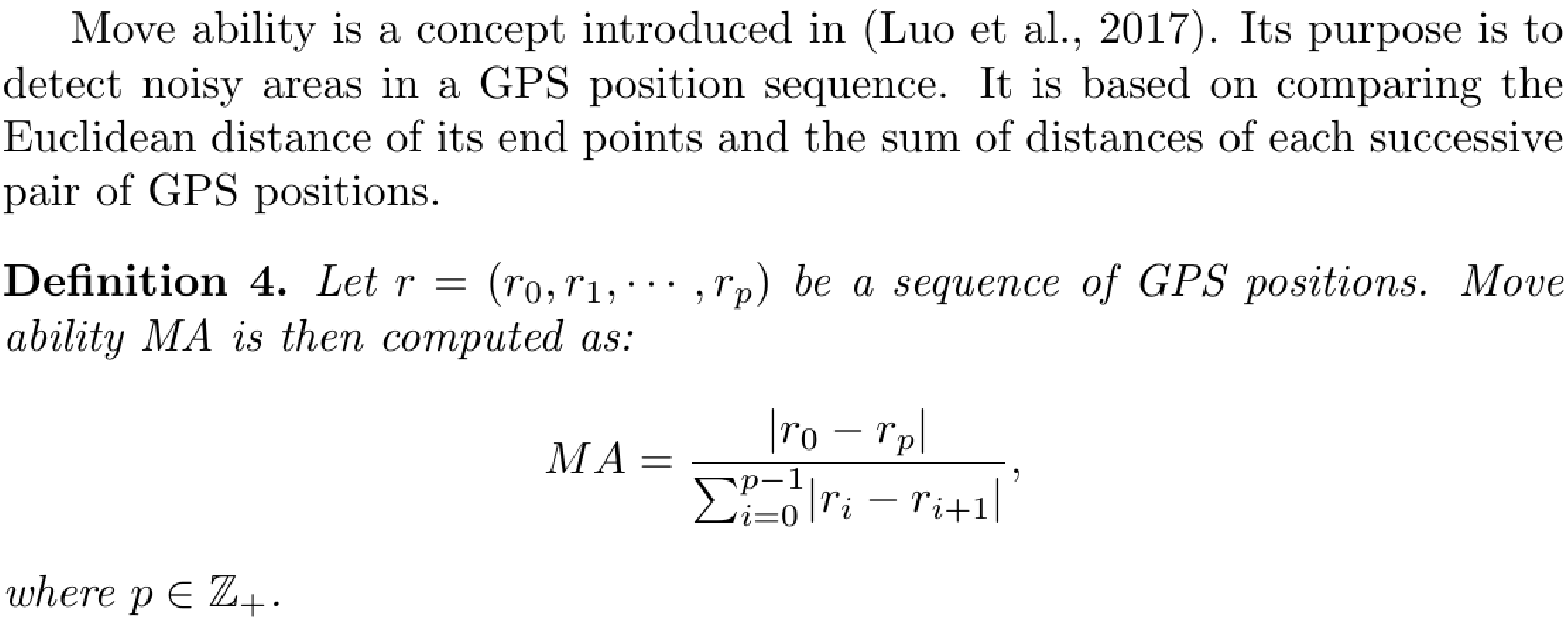
Interpolácia



(a) Recorded trajectory.

(b) Interpolated trajectory.

Move ability



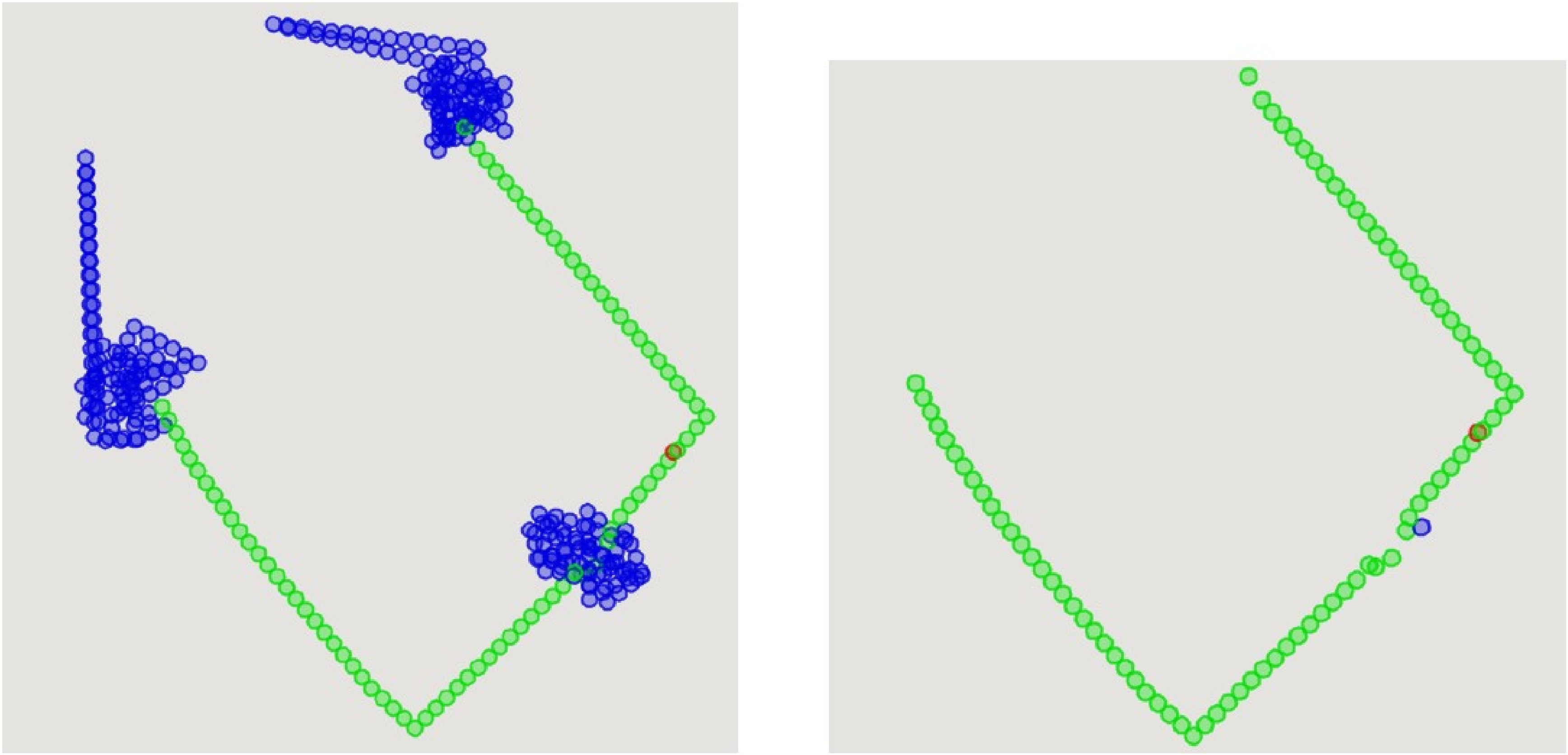
Move ability is a concept introduced in (Luo et al., 2017). Its purpose is to detect noisy areas in a GPS position sequence. It is based on comparing the Euclidean distance of its end points and the sum of distances of each successive pair of GPS positions.

Definition 4. *Let $r = (r_0, r_1, \cdots, r_p)$ be a sequence of GPS positions. Move ability MA is then computed as:*

$$MA = \frac{|r_0 - r_p|}{\sum_{i=0}^{p-1} |r_i - r_{i+1}|},$$

where $p \in \mathbb{Z}_+$.

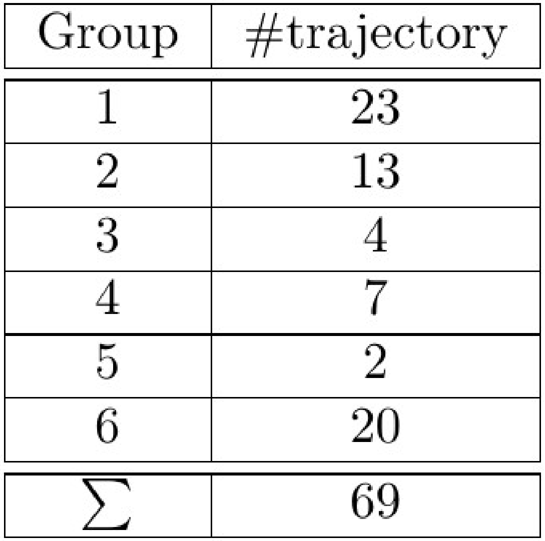
Move ability



(a) Without move ability.

(b) With move ability.

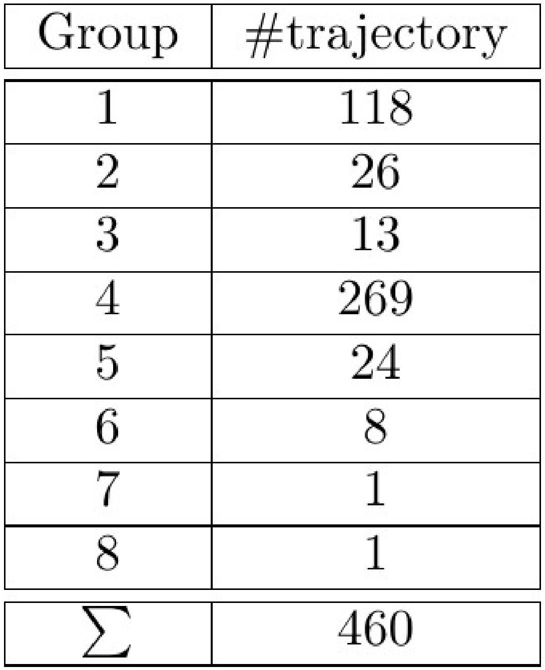
Cohave data set



Group	#trajectory
1	23
2	13
3	4
4	7
5	2
6	20
$\sum$	69

Table 1: COhave: the number of groups and trajectories.

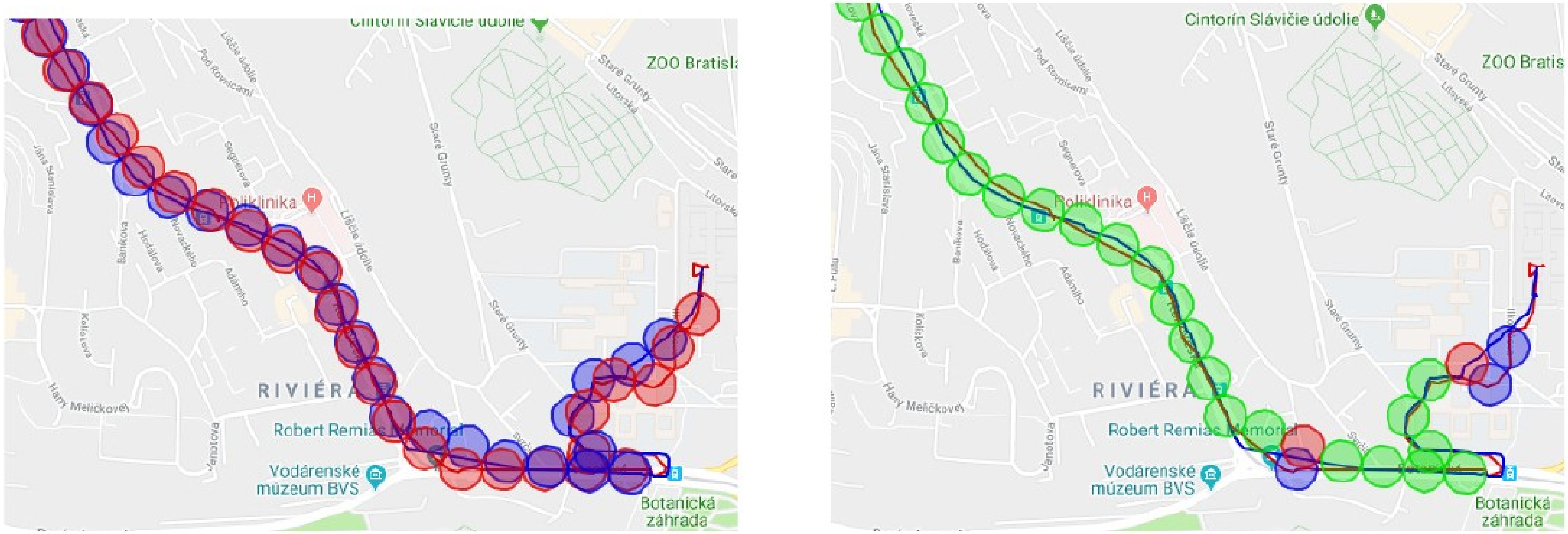
Geolife data set



Group	#trajectory
1	118
2	26
3	13
4	269
5	24
6	8
7	1
8	1
$\sum$	460

Table 2: Geolife: the number of groups and trajectories.

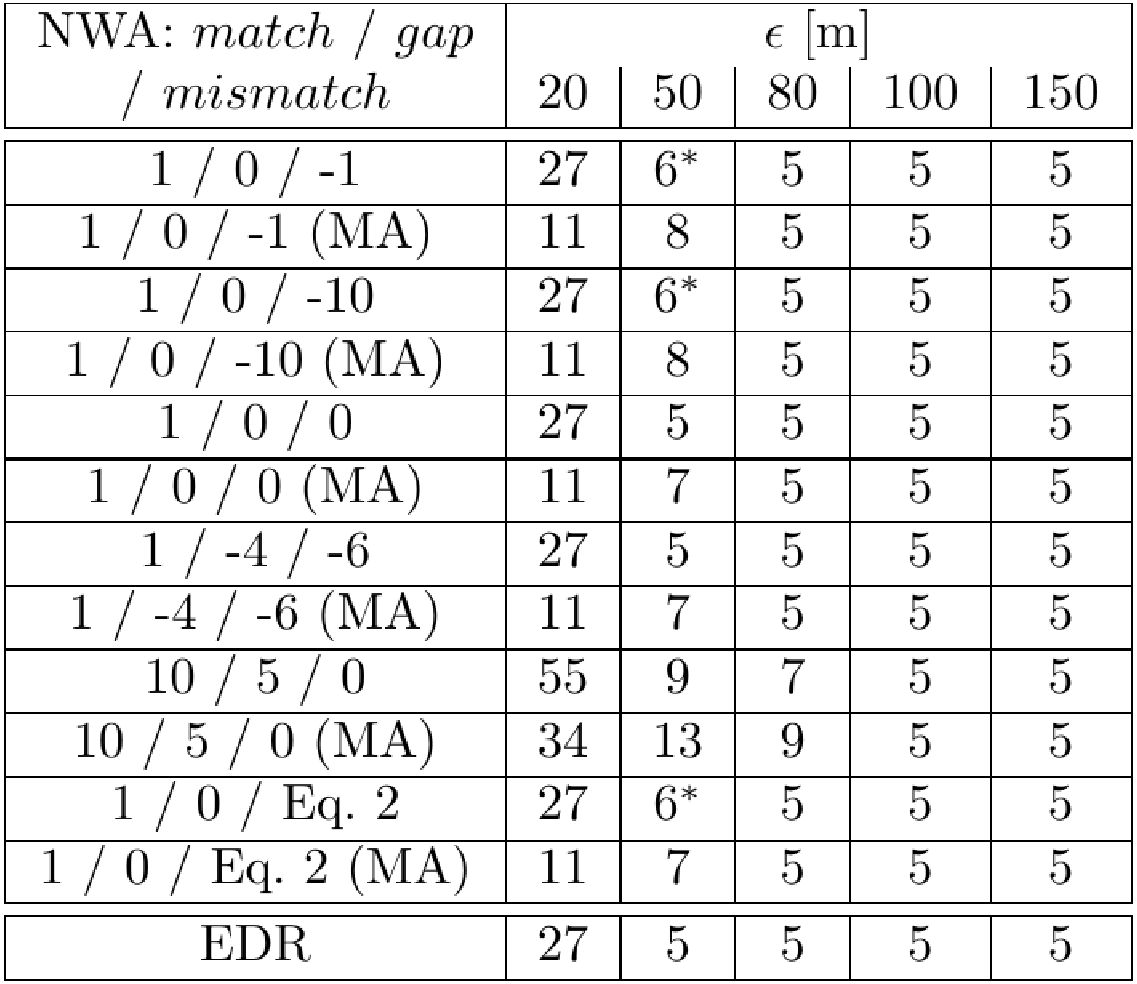
Výsledky



(a) Two trajectories with linear interpolation.

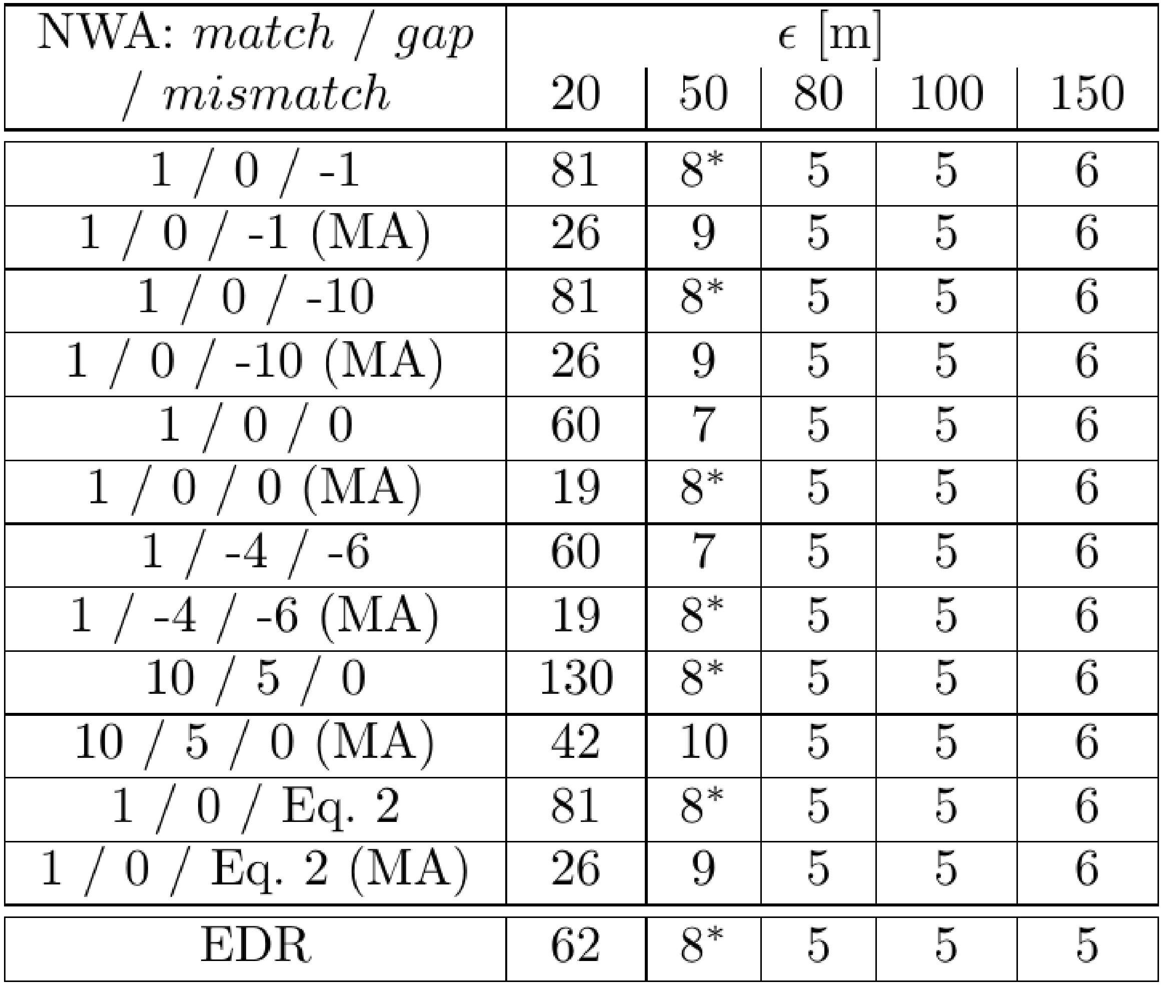
(b) Two aligned trajectories, where green color indicates matches as computed by NWA.

COhave: výsledky



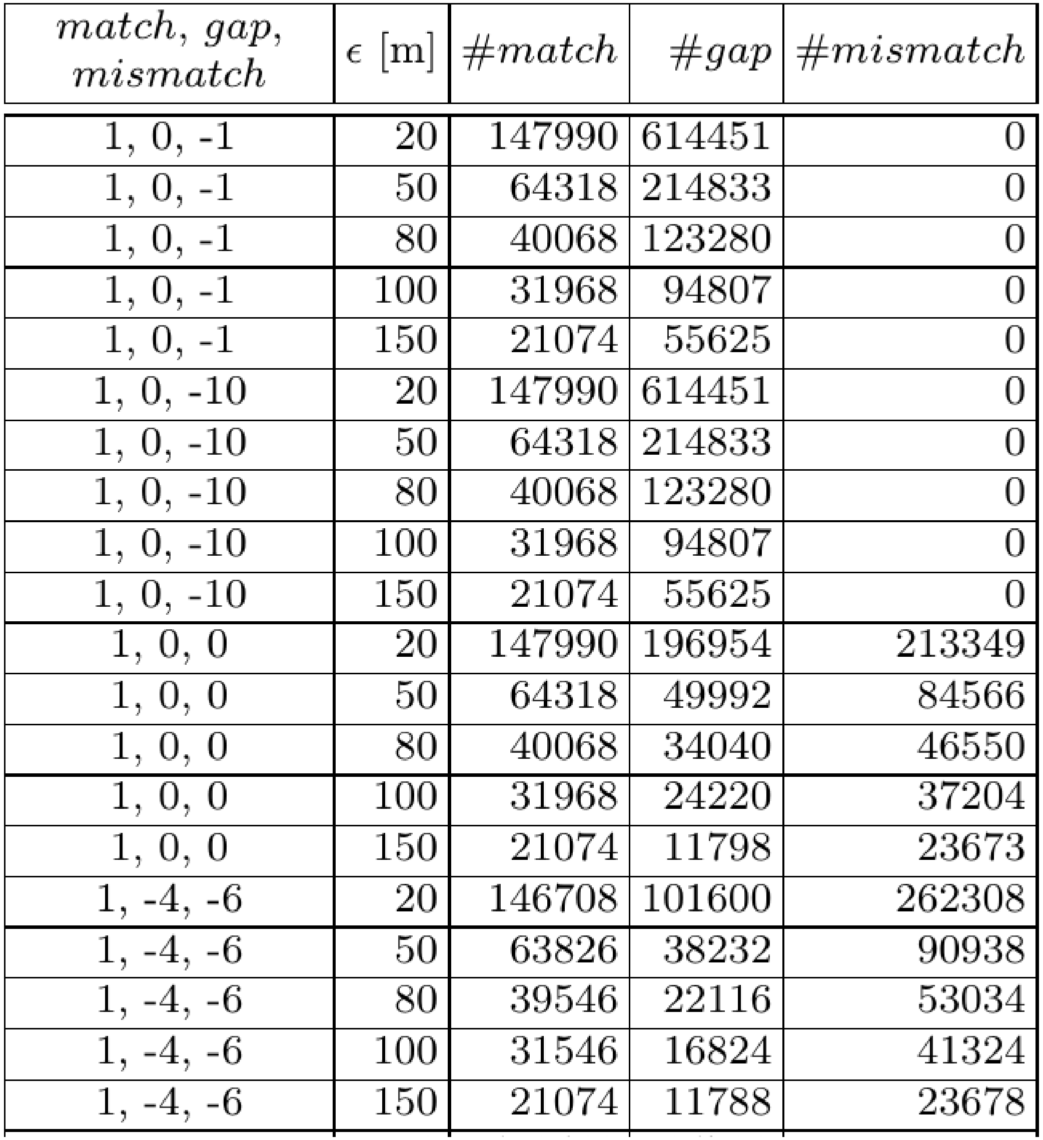
NWA: *match* / *gap* / *mismatch*	ϵ [m]				
	20	50	80	100	150
1 / 0 / -1	27	6*	5	5	5
1 / 0 / -1 (MA)	11	8	5	5	5
1 / 0 / -10	27	6*	5	5	5
1 / 0 / -10 (MA)	11	8	5	5	5
1 / 0 / 0	27	5	5	5	5
1 / 0 / 0 (MA)	11	7	5	5	5
1 / -4 / -6	27	5	5	5	5
1 / -4 / -6 (MA)	11	7	5	5	5
10 / 5 / 0	55	9	7	5	5
10 / 5 / 0 (MA)	34	13	9	5	5
1 / 0 / Eq. 2	27	6*	5	5	5
1 / 0 / Eq. 2 (MA)	11	7	5	5	5
EDR	27	5	5	5	5

Geolife: výsledky



NWA: *match* / *gap* / *mismatch*	ϵ [m]				
	20	50	80	100	150
1 / 0 / -1	81	8*	5	5	6
1 / 0 / -1 (MA)	26	9	5	5	6
1 / 0 / -10	81	8*	5	5	6
1 / 0 / -10 (MA)	26	9	5	5	6
1 / 0 / 0	60	7	5	5	6
1 / 0 / 0 (MA)	19	8*	5	5	6
1 / -4 / -6	60	7	5	5	6
1 / -4 / -6 (MA)	19	8*	5	5	6
10 / 5 / 0	130	8*	5	5	6
10 / 5 / 0 (MA)	42	10	5	5	6
1 / 0 / Eq. 2	81	8*	5	5	6
1 / 0 / Eq. 2 (MA)	26	9	5	5	6
EDR	62	8*	5	5	5

COhave: výsledky



match, gap, mismatch	ϵ [m]	#*match*	#*gap*	#*mismatch*
1, 0, -1	20	147990	614451	0
1, 0, -1	50	64318	214833	0
1, 0, -1	80	40068	123280	0
1, 0, -1	100	31968	94807	0
1, 0, -1	150	21074	55625	0
1, 0, -10	20	147990	614451	0
1, 0, -10	50	64318	214833	0
1, 0, -10	80	40068	123280	0
1, 0, -10	100	31968	94807	0
1, 0, -10	150	21074	55625	0
1, 0, 0	20	147990	196954	213349
1, 0, 0	50	64318	49992	84566
1, 0, 0	80	40068	34040	46550
1, 0, 0	100	31968	24220	37204
1, 0, 0	150	21074	11798	23673
1, -4, -6	20	146708	101600	262308
1, -4, -6	50	63826	38232	90938
1, -4, -6	80	39546	22116	53034
1, -4, -6	100	31546	16824	41324
1, -4, -6	150	21074	11788	23678

Ďakujem za pozornosť