



SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

Ing. Maroš Čavojský

Autoreferát dizertačnej práce

Metodológia zaznamenávania pohybu používateľa

na získanie: vedecko-akademickej hodnosti
philosophiae doctor, PhD.

v doktorandskom študijnom programe: Aplikovaná informatika
v študijnom odbore: 9.2.9 aplikovaná informatika

Miesto a dátum: Bratislava, 28. 5. 2019

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Ing. Maroš Čavojský

Autoreferát dizertačnej práce

Metodológia zaznamenávania pohybu používateľa

na získanie: vedecko-akademickej hodnosti
philosophiae doctor, PhD.

v doktorandskom študijnom programe: Aplikovaná informatika

Miesto a dátum: Bratislava, 28. 5. 2019

Dizertačná práca bola vypracovaná: v dennej forme doktorandského štúdia na Ústave informatiky a matematiky FEI STU v Bratislave.

Predkladateľ: Ing. Maroš Čavojský
ÚIM FEI STU
Ilkovičova 3
812 19 Bratislava

Školiteľ: Dr. rer. nat. Martin Drozda
ÚIM FEI STU
Ilkovičova 3
812 19 Bratislava

Oponenti: Dr. Tomáš Piatrik
Queen Mary, University of London
Mile End Road
E1 4NS London, UK

doc. Ing. Zoltán Balogh, PhD.
Univerzita Konštantína Filozofa v Nitre
Tr. A. Hlinku 1
949 74 Nitra

Autoreferát bol rozoslaný dňa:

Obhajoba dizertačnej práce sa koná:

Na: Fakulte elektrotechniky a informatiky STU,
Ilkovičova 3, 812 19 Bratislava, v miestnosti C602.

prof. Dr. Ing. Miloš Oravec
dekan FEI STU

Obsah

1	Úvod	2
2	Ciele dizertačnej práce	3
3	Dosiahnuté výsledky dizertačnej práce	3
3.1	Zaznamenávanie pohybu - Locator, FunFi	4
3.2	Metodológia vyhodnotenia metód - COhaveCompare	11
4	Splnenie cieľov dizertačnej práce	15
5	Riešiteľ projektov	17
6	Zoznam prác dizertanta	17
	Literatúra	18
7	Summary	20

1 Úvod

Pri používaní GPS navigácie v aute, jednotliví vodiči odosielajú svoju polohu a rýchlosť, ktoré sa využívajú aj na vizualizáciu stavu aktuálnej premávky, alebo upozornenie o dopravných zápchach. Výhodu vyhnúť sa zápcham alebo iným obmedzeniam v cestnej premávke [22, 21], týmto spôsobom poskytujú aj vodičom, ktorí údaje neodosielajú. Návrh výstavby pozemných komunikácií sa v niektorých prípadoch riadi aj z pozbieraných údajov a to nie len z informácií o pohybe ale aj na základe úrovni hlučnosti v jednotlivých častiach mesta [16]. Komerčné lokalaizačné služby zaoberajúce sa navigáciou vo vnútri budov sa spoliehajú na WiFi mapy vytvorené z údajov pozbieraných miliónmi používateľov mobilných zariadení [18]. Mobilné systémy na zber dát (mobile crowdsourcing) sa stali neoddeliteľnou súčasťou nášho života.

Tento rastúci trend mobilných systémov na zber dát predovšetkým záleží od zaznamenávania a odosielania týchto dát veľkým množstvom používateľov z rôznych senzorov mobilných zariadení. Minimalizácia zátaže asistencie používateľov pri zaznamenávaní údajov je klúčovým faktorom pre akceptovateľnosť týchto systémov. V súčasnosti sa väčšina existujúcich systémov zameriava na jednoduché a neefektívne stratégie získavania údajov [17, 3]. Prvá stratégia sa zameriava na časté aktívne zapájanie používateľa pri zaznamenaní, ktorého výsledkom sú vysoko kvalitné dáta. Stratégie nevyžadujúce aktívne zapájanie používateľa sú zaznamenávanie údajov v pravidelných (alebo náhodných) intervaloch prípadne zaznamenávanie na základe geograficky definovaných oblastí.

Každá z uvedených stratégii má svoje výrazné nežiadúce efekty v podobe nízkej kvality údajov, vysokej spotreby batérie prípadne zapojenia používateľa do procesu zaznamenávania. Ideálny systém pre zaznamenávanie údajov je súčasne energeticky efektívny a nevyžaduje interakciu používateľa pričom zaznamenáva pohyb používateľa v dostatočnej kvalite na dodatočnú analýzu. V práci testujeme základné dostupné možnosti Android API pre zaznamenávanie pohybu používateľa. Následne navrhujeme, analyzujeme a vyhodnocujeme vlastné navrhnuté metódy Locator a FunFi na zaznamenávanie pohybu používateľa.

2 Ciele dizertačnej práce

Hlavným cieľom, ktorý sme na začiatku doktorandského štúdia stanovili je návrh metodológie zaznamenávania pohybu používateľa pomocou mobilných zariadení s platformou Android OS. Jedným z cieľov je implementácia jednotlivých návrhov zaznamenávania pohybu používateľa na základe analýzy využitia rozhrania operačného systému Android pre zaznamenávanie pohybu. Ďalším cieľom je zhodnotenie kvality zaznamenaných údajov pre jednotlivé navrhnuté spôsoby zaznamenávania pohybu používateľov. Vedeckými cieľmi, resp. tézami tejto dizertačnej práce sú príspevky do oblasti kontextovo-uvedomelých mobilných aplikácií (context-aware mobile applications) a to:

1. analýza možností pre zaznamenávanie pohybu používateľa,
2. výskum ich výhod/nevýhod s prehľadom obmedzení a chýb,
3. návrh metód pre zaznamenávanie pohybu používateľa a
4. vyhodnotenie metód pre zaznamenávanie pohybu používateľa.

Pri návrhu a vyhodnotení metódy zaznamenávania sa uvažuje pomer medzi kvalitou zaznamenania pohybu a spotreby energie batérie pri zaznamenávaní.

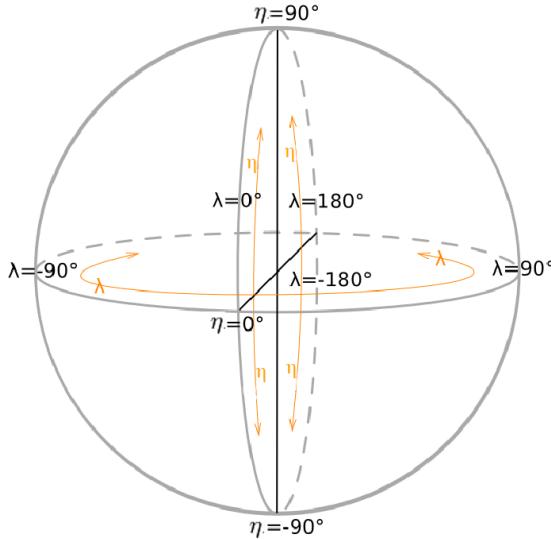
3 Dosiahnuté výsledky dizertačnej práce

V prvej kapitole s názvom *Teoretický úvod* popisujeme dostupné možnosti určovania polohy zariadenia (1. téza) a uvádzame informácie ohľadom získavania polohy pomocou lokalizačných systémov, senzorov zariadenia, základňových rádiových staníc alebo pomocou WiFi sietí. V druhej a tretej kapitole s názvami *Zaznamenávanie pohybu pomocou GPS* (Globálny lokalizačný systém) a *Zaznamenávanie pohybu pomocou GPS a WiFi sietí* uvádzame zhrnutie aktuálnych možností pre zaznamenávanie pohybu pomocou Android SDK spolu s analýzou ich výhod/nevýhod a prehľadom obmedzení a chýb (2. téza). Uvedená analýza vychádza z preštudovaných vedeckých článkov a výskumu v spolupráci so študentmi bakalárskych prác, počas ktorej navrhujeme a implementujeme **dve natívne metódy** (GPS, Network). Počas testovania uvedených natívnych metód identifikujeme niekoľko obmedzení a chýb: vonkajšia chyba statickej polohy, vnútorná chyba statickej polohy (hniezdo), diery, dlhý TTFF (Time to first fix), neúspešný fix či prekážky v okolí. Tieto obmedzenia a chýby riešime prípadne minimalizujeme v návrhoch dvoch vlastných metód (**Locator**, **FunFi**) (3. téza). Vo štvrtej kapitole predstavujeme vlastný spôsob pre vyhodnotenie navrhnutých metód (4. téza) na základe inšpirácie z oblasti bioinformatiky (zarovnávanie sekvencií DNA algoritmom Needleman-Wunsch). Náš spôsob vyhodnotenia metód nazývame **COhaveCompare**. Použitím vyhodnotenia COhaveCompare porovnávame a vyhodnocujeme jednotlivé testované metódy.

Za jednotlivé prínosy respektívne výsledky dizertačnej práce považujeme návrh metódy Locator, neskôr z nej vylepšenú verziu FunFi a návrh spôsobu vyhodnotenia COhaveCompare inšpirovaného bioinformatikou.

3.1 Zaznamenávanie pohybu - Locator, FunFi

Pre pohyb zariadenia (používateľa), ktorý chceme zaznamenávať potrebujeme jednotlivé umiestnenia na povrchu Zeme, kde sa zariadenie nachádzalo. Z tohto dôvodu v práci definujeme *pozíciu* ako umiestnenie zariadenia pomocou zemepisných súradníč.



Obr. 1: Zemepisná šírka a dlžka

Definícia 1. *Pozícia \mathcal{P} je dvojica (lat, lon) , kde*

- *lat je zemepisná šírka (η) vyjadrená v desatinných číslach (napr. 48.1518568),*
- *lon je zemepisná dlžka (λ) vyjadrená v desatinných číslach (napr. 17.0711559).*

Získavanie polohy použitím Android aplikáčného rozhrania nie je presné, nakoľko každé meranie zariadením obsahuje určitú chybu. Získanie polohy je možné v Android OS tromi rôznymi spôsobmi: pomocou GPS; mobilných vysielačov (BTS) a WiFi sietí (Network); alebo hybridným spôsobom (Fused). Preto okrem pozície zavádzame pri meraní pojem poloha. Poloha obsahuje okrem pozície aj iné potrebné údaje ako čas, presnosť či spôsob, akým bola poloha získaná.

Definícia 2. *Poloha \mathcal{L} je štvorica $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{V})$, kde*

- \mathcal{P} je pozícia v desatinných stupňoch,
- \mathcal{T} je čas v sekundách počítaný od 1.1.1970 (UTC),
- \mathcal{A} je horizontálna presnosť v metroch,
- \mathcal{V} je spôsob získania pozície, $\mathcal{V} \in \{GPS, Network, Fused\}$.

Postupnosť nameraných polôh zariadením určuje pohyb používateľa. Pohyb používateľa však obsahuje celú história polôh používateľa (od začiatku experimentu). Pre účely analýzy pohybu využívame aj čiastkové pohyby ako napríklad cesta do práce, z práce či iné významné časti pohybu. Jednotlivé časti pohybu v rámci celkového pohybu označujeme ako *stopy*.

Definícia 3. Pohyb \mathcal{M} je postupnosť štvoríc $(\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n)$, kde pre každé $i = 1 \dots n$ platí, že \mathcal{T}_i je neklesajúca postupnosť časov meraní polôh. Každú $\mathcal{S} \subseteq M$ nazývame stopou.

Android OS umožňuje získavanie polohy rôznymi spôsobmi. Pri určení polohy pomocou GPS sa využíva GPS čip zariadenia, ktorý určuje polohu na základe GPS satelitov na zemskej orbite v dosahu. Zaznamenávanie pomocou sietí (Network) uskutočňuje Android OS na základe mobilných vysielačov (BTS) a WiFi sietí v okolí zariadenia. Pri kombinovanom spôsobe (Fused) určuje Android API samostatne, ktorý z dvoch predchádzajúcich sa použije na získanie polohy. Pri jednotlivých meraniach môže API použiť rôzne spôsoby. Pre zaregistrovanie požiadavky o automatické získavanie polohy je potrebné určiť minimálny rozdiel času a vzdialenosť, po ktorom aplikácia informuje o novej polohe. V prípade nastavenia obidvoch hodnôt na nulu systém poskytuje polohy v najrýchlejšej možnej intenzite. Zaznamenávanie stopy môžeme pomocou uvedených parametrov definovať nasledovnou funkciou.

Definícia 4. *Zaznamenávanie polohy je funkcia,*

$$\mathcal{Z}(\mathcal{V}, \min(\Delta t), \min(\Delta s)) = \mathcal{S},$$

ktorej výstupom je stopa $\mathcal{S} = \{\mathcal{L}_1, \dots, \mathcal{L}_n\}$, kde pre každú polohu $\mathcal{L}_i \in \mathcal{S}$, $i = 1, \dots, n$ platí, že:

$$(|\mathcal{T}_{i+1} - \mathcal{T}_i| \geq \min(\Delta t)) \wedge (\mathcal{P}_{i+1} - \mathcal{P}_i \geq \min(\Delta s)) \wedge \mathcal{V}_i \in \{GPS, Network, Fused\}.$$

Pre zistenie aktuálnych možností pre zaznamenávanie pohybu pomocou Android SDK, ich výhod/nevýhod a výskum obmedzení a chýb sme implementovali a otestovali jednotlivé spôsoby získavania polohy.

Zaznamenávanie polohy pomocou GPS:

$$\mathcal{Z}_{gps}(\{GPS\}, \min(\Delta t)_{gps}, \min(\Delta s)_{gps}).$$

Zaznamenávanie polohy pomocou siete:

$$\mathcal{Z}_{network}(\{Network\}, \min(\Delta t)_{network}, \min(\Delta s)_{network}).$$

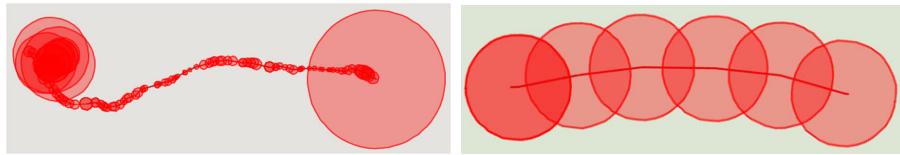
Zaznamenávanie polohy pasívnym spôsobom:

$$\mathcal{Z}_p(\{GPS, Network\}, \min(\Delta t)_p, \min(\Delta s)_p).$$

Zaznamenávanie polohy hybridným spôsobom:

$$\mathcal{Z}_h(\{Fused\}, \min(\Delta t)_h, \min(\Delta s)_h).$$

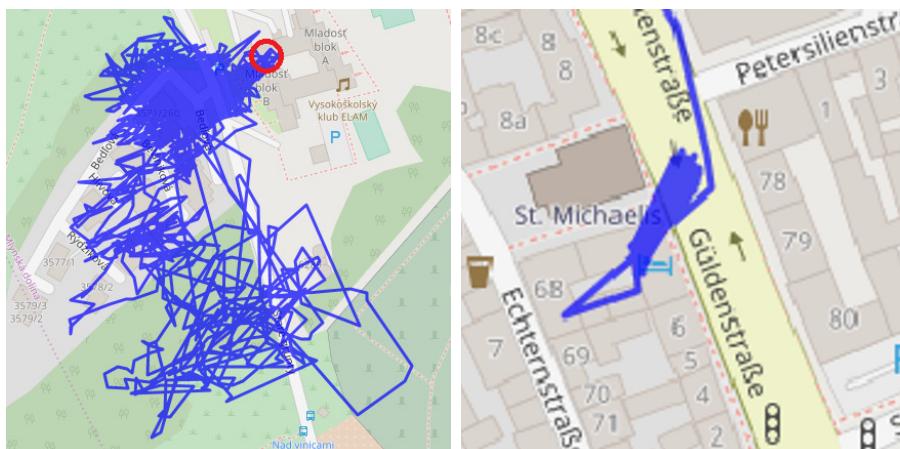
Už po prvých testoch metódy GPS na zaznamenávanie pohybu vieme povedať, že predstava o zaznamenaných stopách sa veľmi odlišuje od skutočne získaných stôp zariadením. Na obrázku 2 prestavujú kružnice získané polohy, pričom polomer kružnice je ich horizontálna presnosť (\mathcal{A}). Pri začiatku merania väčšinou zariadenie začína s nižšou presnosťou, ktorá sa neskôr zvyšuje. V priebehu pohybu si zariadenie udržiava v ustálenom intervale ale pri ukončení zaznamenávania opäť klesá často kvôli ukončeniu v vnútorných priestoroch.



(a) Zaznamenaná stopa - nameraná (b) Zaznamenaná stopa - predstava

Obr. 2: Porovnanie získanej stopy oproti predstave o stope

Po vykonaní testov uvedených štyroch spôsobov pre zaznamenávanie polohy uvádzame jednotlivé chyby a nedostatky, ktoré riešime v navrhnutých metódach Locator a FunFi. Ak sa používateľ nepohybuje, tak určenie jeho polohy môže byť v priebehu času rôzne (nepresné). Dôvodom nepresnosti sú hlavne budovy v jeho okolí kde dochádza k viac-cestnému šíreniu signálu medzi satelitom a GPS prijímačom v mobilnom zariadení. Uvedenú chybu merania označujeme ako **vonkajšia chyba statickej polohy** (obr. 3a). Keď je používateľ vo vnútorných priestoroch a nepohybuje sa pri zaznamenávaní získavame falošné pohyby medzi rôznymi vzdialenosťmi miestami. Pri spôsobe GPS sú dôvodom viacnásobné odrazy signálu. Pri zaznamenávaní pomocou siete je dôvodom hlavne kolísanie prijímaného signálu či množina dostupných WiFi sietí a mobilných vysielačov. V určitých prípadoch vo vnútri budov nie je zaznamenaná vôbec žiadna poloha. Uvedenú chybu označujeme ako **vnútorná chyba statickej polohy** alebo skrátene **hniezdo** (obr. 3b).



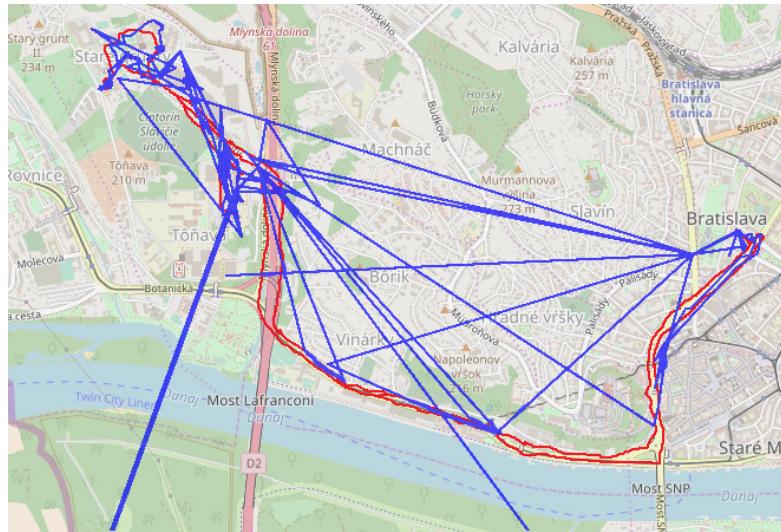
(a) Vnútorná

(b) Vonkajšia

Obr. 3: Ukážka chýb statickej polohy

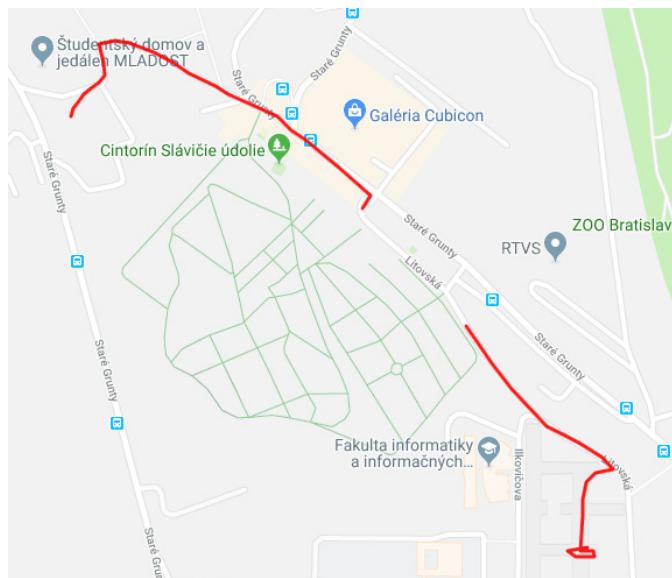
Ak pre zaznamenávanie používame sieť a používateľ sa pripája na internet len cez WiFi, tak môžeme zaznamenať jeho pohyb len v miestach kde je pripojený na WiFi s internetovým pripojením. Akýkoľvek iný pohyb alebo pohyb medzi týmito pozíciami nevieme zaznamenať keďže zariadenie nie je pripojené na internet. V týchto prípadoch sa strácajú jednotlivé pohyby používateľa, ktoré by mohli ako celok poskytnúť cenné informácie o vzoroch pohybu používateľa. Stopy s väčšími vzdialenosťami medzi jednotlivými polohami nazývame **čiastočné stopy** (obr. 4).

Ďalšou podstatnou chybou pri zaznamenávaní pohybu používateľa sú vynechané polohy v stope. Aplikačné rozhranie pre získavanie polohy oznamuje zmenu polohy najskôr po uplynutí minimálnej vzdialenosťi a času. V niektorých prípadoch



Obr. 4: Čiastočné stopy (siet')

sa stáva, že aj po uplynutí vzdialenosť a času neobdržíme zmenu polohy dlhšiu dobu. V týchto prípadoch sú určité stopy prerušené. Úseky s chýbajúcimi (nezaznamenanými) polohami nazývame **dيري** (obr. 5).



Obr. 5: Diera - chýbajúce polohy

Uvedené identifikované nedostatky a chyby pri zaznamenávaní minimalizujeme prípadne úplne odstraňujeme v návrhoch pre zaznamenávanie pohybu Locator a FunFi.

Metóda Locator predstavuje kombinovanú metódu s pauzami, v ktorej využívame obidva spôsoby (GPS a siet') pre zaznamenávanie polohy a pridávame aj relaxáciu v prípade nedostupnosti získania polohy (vnútri budovy resp. bez internetového pripojenia). Za najväčší problém pri zaznamenávaní pohybu považujeme vnútornú chybu statickej polohy (hniezdo). Výskyt vnútornej chyby statickej polohy je najčastejší pri zdržiavaní sa v oblasti so zlým alebo žiadnym príjomom GPS signálu. V týchto oblastiach získanie polohy GPS čipom trvá dlho a keď sa podarí zachytiť určitý signál, tak kvôli odrazom je vypočítaná poloha nepresná. Pre

úspešné získanie polohy pomocou GPS je potrebné aby zariadenie malo GPS fix. V prípade, že GPS dlho nenastal tak predpokladáme, že zariadenie je vo oblasti so zlým prípadne žiadnym GPS signálom (napr. v budove, tuneli, garáži).

Ak výsledkom zaznamenávania je stopa $\mathcal{S} = (\mathcal{L}_1, \dots, \mathcal{L}_n)$ pre $i = 1 \dots n$, potom čas pre získanie fixu **FDT** (Fix Duration Time) počítame nasledovne:

Ak $\mathcal{S} = \emptyset$:

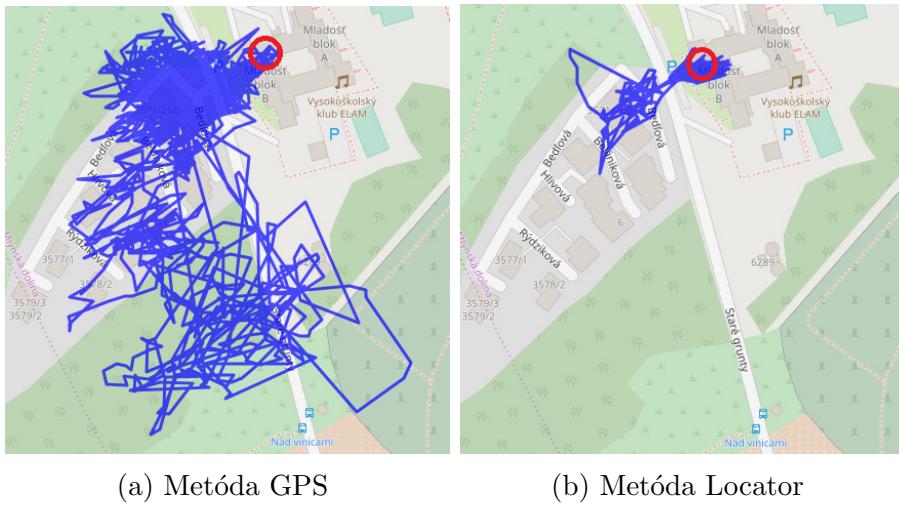
$$FDT = \text{time}() - \mathcal{T}_0,$$

inak

$$FDT = \text{time}() - \mathcal{T}_n,$$

kde $\text{time}()$ je aktuálny čas v sekundách, \mathcal{T}_0 je čas kedy bolo spustené zaznamenávanie a \mathcal{T}_n je čas poslednej zaznamenannej polohy.

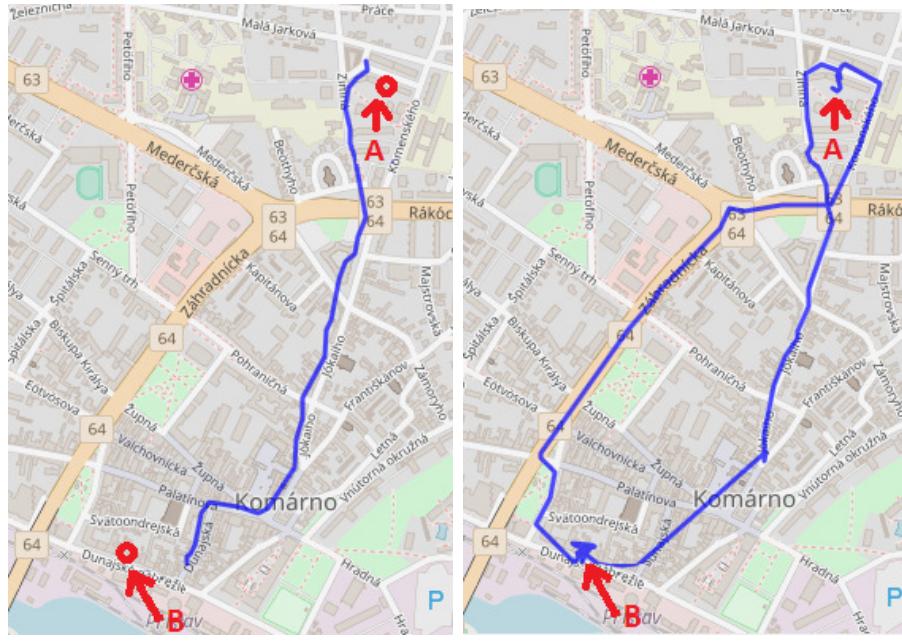
Ak fix trvá dlhšie ako φ ($FDT \geq \varphi$), tak ho voláme **neúspešný fix**. Po neúspešnom fixe pozastavujeme zaznamenávanie pomocou GPS na interval ϑ a prechádzame na zaznamenávanie pomocou siete. V prípade, že je zariadenie pripojené k internetu, tak vie presnejšie určiť svoju polohu ako pomocou GPS. Po uplynutí intervalu ϑ zastavíme zaznamenávanie sieťou a znova umožníme metóde GPS získať fix. Uvedenými intervalmi na relaxáciu s kombináciou získania polohy pomocou siete odstraňujeme výskyty vnútorných chýb statickej polohy (obr. 6).



Obr. 6: Ukážka nepresnosti pri vnútornej chybe statickej polohy

Pomocou intervalov na relaxáciu a použitím kombinácie oboch spôsobov okrem odstránenia hniezd odstraňujeme zároveň aj problém so zaznamenávaním čiastočných stôp. Čiastočné stopy pri metóde GPS vznikali najmä pri rýchlych prechodom z oblasti bez GPS signálu do oblasti už s dostatočným príjomom GPS signálu. Opäťovné získanie GPS fixu a prvej polohy počas tohto prechodu trvá odlišne v závislosti od aktuálnych podmienok prostredia (obr. 7).

Za najväčšiu nevýhodu v metóde Locator považujeme opakovanie získavanie GPS vo vnútri budov. Preto prichádzame následne s návrhom metódy FunFi, v ktorej sa snažíme úplne vypnúť získavanie polohy vo vnútri budov alebo všeobecne v prípadoch, kedy sa používateľ nepohybuje. Týmto spôsobom očakávame ešte výraznejšie minimalizovanie nepresností pri zaznamenávaní pohybu.



(a) Metóda GPS

(b) Metóda Locator

Obr. 7: Ukážka zaznamenanej stopy pri pohybe zariadenia

Hlavnou ideou návrhu a testov metódy FunFi pre zaznamenávanie pohybu je zistenie kedy sa zariadenie nepohybuje a v týchto prípadoch vypnúť zaznamenávanie pohybu a obnoviť ho opäť v prípade, že sa začne pohybovať. Na identifikáciu či sa zariadenie nachádza dlhšiu dobu na jednom mieste využívame v našom prípade WiFi siete. Nás predpoklad je: **Ak je zariadenie na rovnakom mieste, tak aj WiFi siete v jeho okolí sú v priebehu tohto času identické.**

WiFi siet \mathcal{N} je pätnica $(BSSID, SSID, frekvencia, level, tsf)$,

kde $BSSID$ (Basic Service Set Identifier) je MAC adresa (media access control address - identifikačné číslo sietového adaptéra) hardvéru, ktorý vysiela WiFi signál [2]. $SSID$ (Service Set Identifier) je identifikátor WiFi siete (tiež známy ako názov WiFi siete) [6]. $Frekvencia$ v MHz určuje kanál na ktorom zariadenie komunikuje s prístupovým bodom (AP). TSF (Timing synchronization function) je čas, ktorý označuje vek WiFi siete [5].

Pre získanie WiFi sietí v dosahu zariadenia používame triedu $WiFiManager$ [4] z Android API. Použitím tejto triedy na získanie WiFi sietí v okolí získame WiFi sken. Obsahuje WiFi siete v okolí zariadenia v čase vykonania skenu WiFi sietí.

WiFi sken \mathcal{W} je množina dvojíc $\{(t_i, \mathcal{N}_i), \dots, (t_n, \mathcal{N}_n)\}$,

kde pre každé $i = 1 \dots n$ platí, že t_i je neklesajúca postupnosť časov získania WiFi siete a \mathcal{N}_i je WiFi siet \mathcal{N} .

Dôležitým ukazovateľom v algoritme sú zmeny WiFi sietí v dosahu mobilného zariadenia, ktoré sú súčasťou každej polohy pri metóde FunFi.

Preto poloha (\mathcal{L}_v) je už pätnica $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathcal{V}, \mathcal{W})$,

kde \mathcal{W} je WiFi sken spustený v čase \mathcal{T} . Ak sa menia WiFi siete v okolí zariadenia, predpokladáme, že sa zariadenie pohybuje. Tento predpoklad sme otestovali v kontrolovanom teste s 5 rôznymi zariadeniami. Výsledkom je, že nás predpoklad

funguje spoľahlivo s určitými obmedzeniami. Po dôkladnejšej analýze sme zistili, že dôvodom pre obmedzenia sú hlavne 2 príčiny.

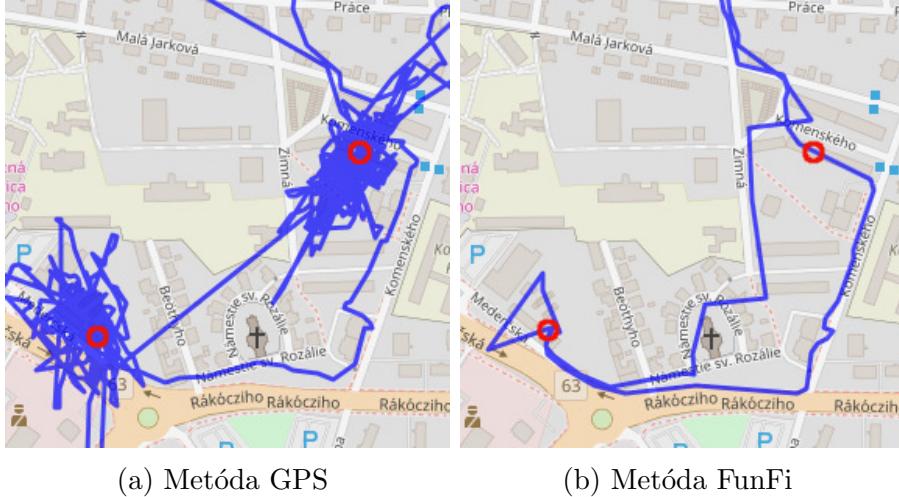
Dočasná pamäť Android API pre WiFi siete. Pri používaní Android API WiFi Manager [4] sa nám potvrdilo, že získaný zoznam WiFi sietí v okolí, obsahuje aj WiFi siete, ktoré nie sú v okolí. Android dočasne uchováva WiFi siete z predchádzajúcich WiFi skenov. Čím sa zariadenie častejšie pripája ku WiFi sieti, tým je dočasné uchovávanie tejto WiFi siete dlhšie (je potrebný dlhší časový interval aby sa WiFi vymazala s dočasnej pamäte). Tento spôsob je pre Android implementovaný kvôli náhodným kolísaniam signálu WiFi siete (RSSI). Spôsob ako ihneď vymazať túto sieť z dočasnej pamäte je pokus o prihlásenie do nej, kedy systém zistí, že naozaj nie je v dosahu. Tento problém komplikuje a veľmi ovplyvňuje nás implementovaný algoritmus.

WiFi siete v pohybe. V testoch sme objavili aj ďalší fakt. Nie všetky WiFi siete sú statické. WiFi siete napríklad v dopravných prostriedkoch ako autobusy či vlaky môžu poskytovať cestujúcim WiFi pomocou WiFi routeru vo vnútri dopravného prostriedku. V tomto prípade je možné diskutovať o pohybe používateľa so zariadením z pohľadu dvoch rôznych referenčných bodov. Vzhľadom na dopravný prostriedok sa používateľ nepohybuje (sedí vo vlaku, autobuse) ale zároveň vzhľadom na budovy a cestu sa pohybuje. Účel aplikácie je však zaznamenávať celkový pohyb používateľa (napr. voči budovám, ceste). Pohybujúce sa WiFi siete preto spôsobujú 2 nasledovné chyby pri určovaní pohybu používateľa:

- **Falošná statická poloha.** Ak je v dosahu zariadenia len jedna WiFi sieť, ktorá je navyše v pohybe (prípad vlaku, autobusu) aplikácia môže tvrdiť, že zariadenie sa nepohybuje. V skutočnosti sa pohybuje.
- **Falošný pohyb.** Ak používateľ stojí pri ceste alebo pri nástupišti, môžu v jeho dosahu pribúdať a miznúť WiFi siete (z okolitých dopravných prostriedkov). Aplikácia na základe toho môže tvrdiť, že zariadenie je v pohybe aj keď používateľ so zariadením sedí na lavičke a nepohybuje sa.

Pomenovaný problém riešime pomocou stacku posledných γ WiFi skenov, kde hľadáme početnosť výskytov rovnakých WiFi sietí. Pohyb zistujeme na základe početnosti jednotlivých WiFi sietí z aktuálneho skenu v posledných γ skenoch stacku. Ak neexistuje žiadna sieť z aktuálneho skenu, ktorá sa vyskytuje vo všetkých posledných γ skenoch, tak predpokladáme, že zariadenie je v pohybe. Pre vylúčenie pohybu zariadenia spočítame počet sietí z aktuálneho skenu, ktoré sa vyskytujú vo viac ako α_0 percentách posledných γ skenoch. Ak je pomer počtu týchto WiFi sietí oproti celkovému počtu posledných skenov väčší ako stanovená hodnota β , tak predpokladáme, že zariadenie sa nepohybuje. Použitím viacerých skenov dokážeme presnejšie určovať, či je zariadenie v pohybe. Identifikáciu pohybu je hlavný prínos metódy zaznamenávania FunFi. Túto metódu sme publikovali aj v článku *Energy Efficient Trajectory Recording of Mobile Devices Using WiFi Scanning* [1].

V našej navrhutej metódy FunFi sme nakoniec minimalizovali hniezda a osstatné nedostatky pri zaznamenávaní pohybu používateľa. Na obrázku 8 môžeme vidieť, že hniezda sú minimalizované až odstránené v porovnaní s metódou GPS. Viditeľné zlepšenie vidieť aj pri porovnaní s predchádzajúcou navrhnutou metódou Locator, kde na obrázku 6 nie sú tak dokonale minimalizované hniezda.



Obr. 8: Porovnanie nepresnosti pri vnútornej chybe statickej polohy

3.2 Metodológia vyhodnotenia metód - COhaveCompare

Pre vyhodnotenie a určenie prínosov jednotlivých navrhnutých metód sme sa zaoberali metodológiou porovnávania zaznamenaných pohybov. Cieľom návrhu metódy COhaveCompare je možnosť porovnania a vyhodnotenia stôp aj napriek ich obsahu rôznych chýb a nedostatkov. Nakol'ko každá stopa obsahuje časovú (\mathcal{T}) a priestorovú zložku (\mathcal{P}), uvažujeme o troch druhoch podobností stôp: **priestorová**, **časová** a **priestorovo-časová** [23]. Priestorová stopa obsahuje len postupnosť pozícii (\mathcal{P}_i)_{n=1}ⁱ⁼¹ bez ohľadu na časovú zložku stopy. Časová stopa naopak obsahuje len postupnosť časov (\mathcal{T}_n)_{n=1}ⁱ⁼¹ kedy bola zaznamenaná zmena polohy. Priestorovo-časová stopa sa uvažuje o polohách v stanovenom čase v plnom rozsahu definície (\mathcal{S}_n)_{n=1}ⁱ⁼¹. V navrhutej metóde COhaveCompare používame len priestorovo-časovú reprezentáciu stopu. Stopu počas porovnávania nazývame aj ako **trajektória**. Pre stopu definujeme dva kvantitatívne ukazovatele **dĺžku** a **veľkosť**. Dĺžku stopy $\mathcal{S} = (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n)$ počítame ako súčet vzdialenosí medzi jednotlivými nasledujúcimi dvojicami polôh. Veľkosť stopy $\mathcal{S} = (\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n)$ považujeme za počet polôh v stope.

$$size(\mathcal{S}) = n; length(\mathcal{S}) = \sum_{i=1}^{n-1} distance(\mathcal{L}_i, \mathcal{L}_{i+1})$$

V navrhutej metóde porovnávania COhaveCompare využívame viacero spôsobov, ktoré spolu zvyšujú úspešnosť a znižujú chybovosť pri porovnávaní stôp zaznamenaných používateľmi. Pre urýchlenie času behu metódy najprv dátu predspracujeme a následne porovnávame. V porovnávaní využívame hlavne poznatky z oblasti bioinformatiky na zarovnávanie stôp. Na základe získaných výsledkov z porovnávania vyhodnocujeme podobnosť medzi jednotlivými stopami.

V pred-spracovaní potrebujeme identifikovať chybné, prípadne nekompletné stopy, ktoré vylúčime z porovnávania. Z tohto dôvodu v prvom kroku vylučujeme stopy s menším počtom polôh ako $size_{min}$ ($size(\mathcal{S}) < size_{min}$). Ďalším podstatným krokom v algoritme je vylúčenie stôp obsahujúcich diery ako aj čiastočných stôp. Vylúčenie týchto chybných stôp zvyšuje presnosť porovnania algoritmu, kedže pracujeme s kvalitnejšími stopami. Z dôsledku nepravidelného získavania polôh nie je možné efektívne porovnať jednotlivé stopy. Riešením je doplniť a rekonštruovať

pohyb používateľa pomocou interpolácie na základe zaznamenaných polôh. Interpolovanú stopu nazývame **cesta**. Cesta sa skladá z nových interpolovaných polôh ďalej nazývaných ako **klastre**.

Definícia 5. Klaster $\mathcal{K} = (\mathcal{P}, \delta)$ je množina polôh $\{\mathcal{L}_1, \dots, \mathcal{L}_n\}$, kde

$$\forall \mathcal{L}_i, \mathcal{L}_j : \text{distance}(\mathcal{L}_i, \mathcal{L}_j) < \delta \text{ metrov},$$

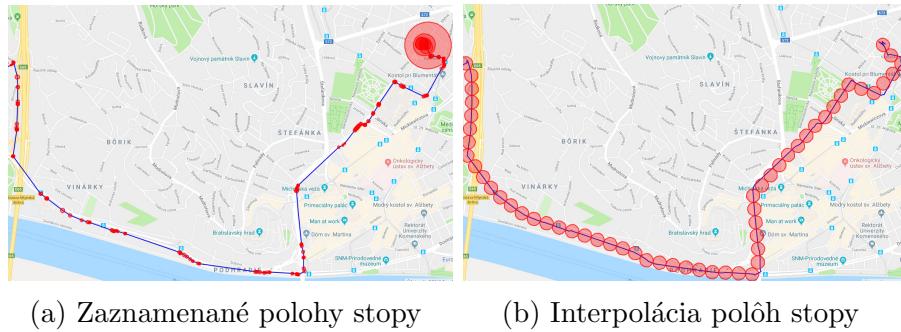
pričom $i = 1 \dots n$ a $j = 1 \dots n$, δ je priemer klastra a \mathcal{P} je stred klastra.

Definícia 6. Cesta \mathcal{C} je usporiadaná postupnosť klastrov $(\mathcal{K}_1, \dots, \mathcal{K}_n)$, kde $\forall \mathcal{K}_i = (\mathcal{P}_a, \delta_a), \mathcal{K}_{i+1} = (\mathcal{P}_b, \delta_b)$:

$$\delta_a = \delta_b \text{ a súčasne } \text{distance}(\mathcal{P}_a, \mathcal{P}_b) = \delta \text{ metrov},$$

pričom $i = 1 \dots (n - 1)$.

Cestu, v ktorej je aspoň jedna vzdialenosť klastrov $(\mathcal{P}_i, \mathcal{P}_{i+1})$ väčšia ako priemer klastra nazývame **neúplnou cestou**. Inak povedané cesta je v niektorom úseku prerušená.



Obr. 9: Zaznamenaná a rovnomerná stopa

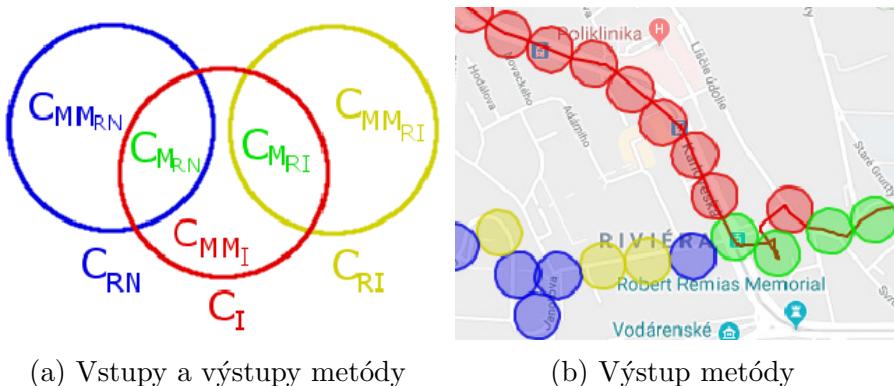
Pre dosiahnutie zarovnania ciest pre porovnanie využívame poznatky z bioinformatiky z oblasti zarovnávania sekvencií. Sekvenčné zarovnávanie [15] je základná bioinformatická metóda, ktorá dokáže zoradiť dve a viac sekvencií DNA, RNA alebo proteínu pod seba tak, aby zodpovedajúce si nukleotidové bázy či aminokyselinové zvyšky ležali pod sebou. Jedna z prvých metód pre zarovnávanie sekvencií je Needleman-Wunsh algoritmus [14, 19], publikovaná v roku 1970. Ide o algoritmus dynamického programovania, ktorý vytvára globálne zarovnanie sekvencií (z ohľadom na celú sekvenciu) dovoľujúce medzery. Najčastejšie ide o sekvenciu nukleotidov v nukleových kyselinách či sekvencie amínokyselín v proteínoch. Vstupom algoritmu sú dve sekvencie, bodovacia matica a hodnota pokuty za vloženie medzery. Tradične ide o celočíselné hodnoty. V bodovacej matici sú zapísané hodnoty bodov za zhodu/nezhodu príslušných znakov z abecedy používanej v sekvencii. Algoritmus najprv zostaví maticu $(m + 1) * (n + 1)$, kde m a n sú dĺžky sekvencií. Riadky matice prislúchajú jednotlivým znakom jednej sekvencie a stĺpce znakom druhej sekvencie. Prvý riadok a prvý stĺpec nepatrí žiadnemu znaku. Pre každý znak v sekvencii môže nastať teda jeden z prípadov MATCH (zhoda), MISMATCH (nezhoda) alebo GAP (medzera). Jediný prípad, v ktorom sú obidve sekvencie zhodné na určitej pozícii je MATCH. Prípady MISMATCH a GAP nastávajú súčasne, kedy v jednej sekvencii je MISMATCH a v druhej je vložená medzera GAP.

V problematike GPS stôp v zmysle ciest chápeme jednotlivé klastre ako znaky v bioinformatike. Zhodu medzi znakmi je ľahké interpretovať. Pri klastroch sme zhodu medzi dvoma klastrami považovali vtedy, ak je vzdialenosť medzi nimi menšia alebo rovná ako δ metrov. Ak sú dva klastre od seba vzdialené viac ako δ metrov tak sú odlišné.

Cesty $\mathcal{C}_a, \mathcal{C}_b$ sú podobné, ak väčšia obsahuje aspoň 75% zhôd z celkovej svojej veľkosti a súčasne sa nevyskytli viac ako 3 nezhody po sebe.

Uvedená metóda na porovnávanie ciest spĺňa všetky nami špecifikované náležitosti (porovnávanie smeru, problém s hniezdam, skutočná vzdialenosť) pre validné porovnávanie. Preto ju používame pre vyhodnotenie a porovnávanie výsledkov zaznamenaných pomocou rôznych metód na zaznamenávanie pohybu používateľa mobilným zariadením. Porovnanie ciest \mathcal{C}_R a \mathcal{C}_I metódou COhaveCompare vytvorí 5 kategórií neúplných cest. Vzťahy medzi jednotlivými neúplnými cestami znázorňujeme pomocou Vennových diagramov na obrázku 10a.

Jednotlivé indexy prvej úrovne predstavujú zhodu M alebo nezhodu MM s cestou v indexe druhej úrovne. Neúplné cesty $\mathcal{C}_{M_{RN}}$ a $\mathcal{C}_{M_{RI}}$ obsahujú zhodu medzi zaznamenanou cestou a ideálnou cestou. Neúplné cesty $\mathcal{C}_{MM_{RN}}$, $\mathcal{C}_{MM_{RI}}$ a \mathcal{C}_{MM_I} neobsahujú zhodu medzi zaznamenanou a ideálnou cestou. Úseky ciest zhodné s ideálnou cestou predstavujú úspešné meranie a úseky kde zhoda z ideálnou cestou nie je predstavujú chybu merania. Cieľom úspešného zaznamenávania je minimálizácia klastrov v neúplných cestách $\mathcal{C}_{MM_{RN}}$, $\mathcal{C}_{MM_{RI}}$ a \mathcal{C}_{MM_I} .



Obr. 10: Vstupy a výstupy metódy porovnania COhaveCompare

Na výsledku z metódy COhaveCompare určujeme vzťahy pre porovnávanie a vyhodnocovanie, ktoré rozdeľujeme na dve kategórie: porovnateľné a neporovnateľné medzi meraniami (označujeme aj ako online a offline kategórie). Vzájomne neporovnateľné sú: **pravidelnosť zariadenia, efektívnosť zariadenia, presnosť zariadenia a presnosť cesty**. Vzájomne porovnateľné sú: **pravidelnosť cesty, úplnosť cesty, efektívnosť validnej cesty a efektívnosť**. Dôvodom prečo niektoré výsledky nie sú vzájomne porovnateľné, je že neobsahujú rovnakú základný dataset (ideálnu cestu). Na obrázku 10b uvádzame výsledné zarovnanie dvoch ciest. Zelenou farbou označujeme zhody (MATCH) medzi cestami a MISS-MATCH pre jednotlivé cesty označujeme modrou, žltou alebo červenou farbou (farby zhodné s diagramom). Popri viacerých výsledkov pre porovnávanie metód potrebujeme aj konkrétny vzťah, ktorý reprezentuje celkové vyhodnotenie. Ukažovateľ, ktorým určujeme celkové vyhodnotenie metódy nazývame efektivita.

Definícia 7. Efektivitu metódy pre zaznamenávanie pohybu zariadením počítame nasledovne:

ak $|C_{M_{RN}}| + |C_{M_{RI}}| > 0$ tak

$$\chi \cdot \left(\frac{\sum_{i=1}^{|C_{M_{RN}}|} \frac{1}{|K_i| - \beta}}{|C_{M_{RN}}|} \right) + (1 - \chi) \cdot \left(\frac{|C_{M_{RN}}| + |C_{M_{RI}}|}{|C_I|} \right)$$

inak

$$\chi \cdot \left(\frac{\sum_{i=1}^{|C_{M_{RN}}|} \frac{1}{|K_i| - \beta}}{|C_{M_{RN}}|} \right),$$

kde klaster $K_i \in C_{M_{RN}}$ a ked' $|K_i| \leq \beta$, tak $|K_i| = \beta + 1$. χ je váha. $|C_{M_{RN}}|$ je veľkosť neúplnej cesty, ktorá predstavuje počet obsahujúcich klastrov. Výsledok je z intervalu $[0, 1]$, hodnota blížiaca sa k 1 znamená vyššiu efektivitu metódy pre zaznamenávanie pohybu.

Efektivita má dve strany mince - používateľ a výskumník. Používateľ aplikácie vyžaduje čo najvyššiu energetickú efektívnosť (najmenej polôh v klastri). Výskumník naopak požaduje čo najpresnejšie zaznamenanie pohybu používateľa (najvyššiu možnú podobnosť s ideálnou cestou) aj na úkor vysokej spotreby energie batérie zariadenia. Po rozsiahlych testoch medzi používateľmi a analýze dát sme s ohľadom na to, že daná aplikácia je primárne výskumná rozdelili význam jednotlivých častí v pomere

$$\chi = \frac{1}{3} \text{ používateľ a } (1 - \chi) \text{ výskumník.}$$

Na základe uvedených výsledkov v dizertačnej práce považujeme FunFi metódu za presnejšiu ako metódu Locator. Taktiež môžeme potvrdiť, že pri testovaní metódy FunFi sme už od používateľov nedostávali negatívne odozvy ako počas testov metódy Locator. Mobilné zariadenie vydržalo z ich pohľadu primerane bez potreby nabíjania. Týmto hodnotíme našu metódu FunFi za úspešnú. Podrobnejšie čiastkové aj celkové výsledky testu uvedených metód aj s vizualizáciou na mape sú uvedené na stránke [12].

4 Splnenie cieľov dizertačnej práce

V nasledujúcich odsekoch sa vyjadrujeme k jednotlivým cieľom práce. Zhrnutím môžeme povedať, že cieľom práce je prispieť k oblasti kontextovo uvedomelých mobilných aplikácií uceleným zhrnutím analýzy aktuálnych možností pri zaznamenávaní pohybu používateľa s následným návrhom vlastných metód pre zaznamenávanie pohybu používateľa, ktoré následne vyhodnotíme s ohľadom na presnosť zaznamenaného pohybu.

V práci popisujeme rôzne modifikácie lokalizačných systémov ako GPS, A-GPS prípadne Glonass či Galileo. Uvádzame princíp fungovania lokalizačného systému GPS a tiež jeho obmedzenia a chyby pri meraní. Ako alternatívne určenie polohy používateľa uvádzame spôsoby na základe základňových rádiových staníc GSM (BTS) prípadne pomocou WiFi sietí. Pre jednotlivé spôsoby uvádzame ich nedostatky, obmedzenia a chyby pri určovaní polohy používateľa. Následne v práci analyzujeme konkrétnie spôsoby pre získanie polohy pomocou mobilného zariadenia. Na základe uvedených nedostatkov, obmedzení a chýb pre získavanie polohy uvádzame ich vhodné použitie na základe aktuálnej situácie pre zaznamenávanie pohybu. Popísaním uceleného zhrnutia analýzy aktuálnych možností pri zaznamenávaní pohybu používateľa spolu s ich výhodami/nevýhodami s prehľadom obmedzení a chýb obsahujeme dosiahnutie prvých dvoch cieľov dizertačnej práce.

Ďalej v práci uvádzame návrh a implementáciu jednotlivých metód, ktoré poskytuje Android API: metóda pomocou GPS, metóda pomocou Siete (Network). Na ich základe predstavujeme návrh vlastných metód na zaznamenávanie pohybu používateľov: metóda Locator (kombinácia GPS a sieti) a FunFi (Locator s pomocou WiFi sietí). Pre jednotlivé metódy uvádzame obmedzenia, ktoré v nich odstraňujeme. Po popise a vysvetlení implementácie metód a ich testovaní vyhodnocujeme ich funkčnosť ako aj používateľský zážitok používateľov (User Experience). Do úvahy berieme kvalitu zaznamenaných údajov, ako aj priaté sťažnosti na funkčnosť aplikácií od používateľov z Google Play obchodu. V práci uvádzame aj zaužívané pojmy pre jednotlivé chyby a nedostatky vznikajúce počas zaznamenávania pohybu používateľov. Jednotlivé výhody a nevýhody pre každú metódu prehľadne uvádzame v bodoch za každou metódou. Návrh finálnej metódy FunFi, publikovaný v článku [1] vychádza z analýzy WiFi sietí, ktorý sme publikovali v článku [9]. Návrhom vlastných metód pre zaznamenávanie pohybu používateľov napĺňame tretí cieľ dizertačnej práce.

Pri návrhu vyhodnotenia jednotlivých navrhnutých metód sme sa inšpirovali algoritmom **Needleman-Wunsch** [14] používaného v bioinformatike pre zarovnávanie sekvencií DNA. Na jeho základe definujeme vzťahy pre porovnávanie a vyhodnocovanie, ktoré rozdeľujeme na dve kategórie: porovnatelné a neporovnatelné medzi meraniami (označujeme aj ako online a offline kategórie). Vzájomne neporovnatelné sú: **pravidelnosť zariadenia, efektívnosť zariadenia, presnosť zariadenia a presnosť cesty**. Vzájomne porovnatelné sú: **pravidelnosť cesty, úplnosť cesty, efektívnosť validnej cesty a efektívnosť**. Uvedenú navrhnutú metodológiu porovnávania nazývame COhaveCompare. Navrhli sme ju na základe výsledkov a skúseností z vedených prác a pri testoch na datasetoch od firmy Sygic [20] a údajov z výskumu Microsoft Geolife data [24]. Z dôsledku nedostatku testovacích zariadení sme rozdelili vyhodnotenie na dve časti: pre metódu Locator a metódu FunFi. Výsledky metód na získané pomocou metódy COhaveCompare dopĺňame aj grafmi a obrázkami pre jednoduchšie porozumenie. Štvrtý cieľ di-

zertačnej práce pokrývame práve návrhom metódy COhaveCompare publikovanej v článku [8] a jej použitím na vyhodnotenie jednotlivých navrhnutých metód.

Okrem spomenutých publikácií som počas PhD štúdia aktívne spolupracoval so študentmi, čoho výsledkom je odovzdaných a úspešne obhájených 19 bakalárskych projektov, a 3 tímové projekty. Niektoré zo získaných výsledkov boli súčasťou publikovaných prác. Celkovo môžeme povedať, že sa nám úspešne podarilo pokryť všetky stanové ciele dizertačnej práce.

5 Riešiteľ projektov

1. Založenie výskumného centra pre analýzu a ochranu dát. Operačný program Výskum a vývoj, spolufinancovaný z Európskeho fondu regionálneho rozvoja a štátneho rozpočtu. 26240120037, 04/2014-09/2015
2. PinSpace: priestor naokolo je tvoj poznámkový blok, Grantový program E-Talent, Nadácia Tatra Banka, 2014
3. SECOSYS – Security and privacy in mobile computing ecosystems, DAAD, zodpovedný riešiteľ Dr. rer. nat. Martin Drozda, 2015-2016

6 Zoznam prác dizertanta

1. Maroš Čavojský a Martin Drozda. Energy Efficient Trajectory Recording of Mobile Devices Using WiFi Scanning. V: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences. IEEE. 2016, s. 1079–1085
2. Maroš Čavojský, Marek Uhlár, Marian Ivanis, Martin Molnar and Martin Drozda. User Trajectory Extraction Based on WiFi Scanning. 2018 6th International Conference on Future Internet of Things and Cloud Workshops, pp. 115-120, IEEE, 2018
3. Maroš Čavojský a Martin Drozda. Comparison of User Trajectories with the Needleman-Wunsch Algorithm“. podané a prijaté na MobiCASE 2019, 10th EAI International Conference on Mobile Computing, Applications and Services. IEEE. 2019.

Okrem uvedených prác som počas PhD štúdia aktívne spolupracoval so študentmi, čoho výsledkom je odovzdaných a úspešne obhájených 19 bakalárskych projektov, 3 tímové projekty a 4 diplomové projekty kde som bol konzultant.

Interaktívnu súčasťou práce sú aj online internetové stránky a mobilné aplikácie s výsledkami dosiahnutými počas PhD štúdia a tiež odkazmi na stiahnutie aplikácií.

- Location Research [11] - výsledky výskumu zameraného na pohyb pomocou GPS, stiahnutie APK pre navrhnuté aplikácie
- WiFi Research [13] - výsledky výskumu zameraného na pohyb pomocou WiFi sietí, stiahnutie APK pre navrhnuté aplikácie
- Research Data [7] - vizualizácia dát zozbieraných počas výskumu
- GPS Tracks comparison and grouping [10] - vizualizácia porovnávania GPS stôp a ich zoskupovanie

Počas výskumu pre potreby napísania predkladanej dizertačnej práce som vytvoril samostatne aj viac ako 15 webových stránok (front-end / back-end) a viac ako 10 rôznych mobilných aplikácií pre platformu Android.

Literatúra

- [1] Maroš Čavojský a Martin Drozda. „Energy Efficient Trajectory Recording of Mobile Devices Using WiFi Scanning“. In: *Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences*. IEEE. 2016, s. 1079–1085.
- [2] Todor Cooklev. *Wireless Communication Standards: A Study of IEEE 802.11, 802.15, 802.16*. IEEE Standards Association, 2004.
- [3] Alexandros Doulamis, Nikos Pelekis a Yannis Theodoridis. „EasyTracker: An Android application for capturing mobility behavior“. In: *2012 16th Panhellenic Conference on Informatics*. IEEE. 2012, s. 357–362.
- [4] Google. *WiFi — Android Developers*. URL: <https://developer.android.com/reference/android/net/wifi/package-summary.html> (cit. 01/2018).
- [5] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard for Information technology*. 2007. DOI: <http://dx.doi.org/10.1109/IEEESTD.2007.373646>.
- [6] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Standard for Information technology*. 2003. DOI: <http://dx.doi.org/10.1109/IEEESTD.2003.94389>.
- [7] Maroš Čavojský. *COhave v2-v4*. URL: <http://data.mcomputing.eu/> (cit. 01/2018).
- [8] Maroš Čavojský a Martin Drozda. „Comparison of User Trajectories with the Needleman-Wunsch Algorithm“. In: *MobiCASE 2019, 10th EAI International Conference on Mobile Computing, Applications and Services*. IEEE. 2019.
- [9] Maroš Čavojský et al. „User Trajectory Extraction Based on WiFi Scanning“. In: *FiCloud 2018, The IEEE 6th International Conference on Future Internet of Things and Cloud*. IEEE. 2018.
- [10] Maros Cavojsky. *GPS Tracks comparison and grouping*. URL: <http://ctu.mcomputing.eu> (cit. 04/2018).
- [11] Maros Cavojsky. *Location Research — COhave - Mobile Computing Group*. URL: <http://location.mcomputing.eu> (cit. 04/2018).
- [12] Maros Cavojsky. *Vyhodnotenie metódy FunFi*. URL: <http://location.mcomputing.eu/funfi/stats.html> (cit. 04/2018).
- [13] Maros Cavojsky. *WiFi Research — COhave - Mobile Computing Group*. URL: <http://wifi.mcomputing.eu> (cit. 04/2018).
- [14] Saul B Needleman a Christian D Wunsch. „A general method applicable to the search for similarities in the amino acid sequence of two proteins“. In: *Journal of molecular biology* 48.3 (1970), s. 443–453.

- [15] Valery O Polyanovsky, Mikhail A Roytberg a Vladimir G Tumanyan. „Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences“. In: *Algorithms for molecular biology* 6.1 (2011), s. 25.
- [16] Rajib Kumar Rana et al. „Ear-phone: an end-to-end participatory urban noise mapping system“. In: *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM. 2010, s. 105–116.
- [17] Manav Singhal a Anupam Shukla. „Implementation of location based services in android using GPS and web services“. In: *International Journal of Computer Science Issues (IJCSI)* 9.1 (2012), s. 237.
- [18] Skyhook Wireless, Inc. *Skyhook Wireless — Location Data Provider*. URL: <http://www.skyhookwireless.com/> (cit. 01/2018).
- [19] Temple F Smith a Michael S Waterman. „Comparison of biosequences“. In: *Advances in applied mathematics* 2.4 (1981), s. 482–489.
- [20] Sygic. *GPS Navigation - Sygic — Bringing life to maps*. URL: <http://www.sygic.com> (cit. 18. 05. 2018).
- [21] Arvind Thiagarajan et al. „VTrack: accurate, energy-aware road traffic delay estimation using mobile phones“. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2009, s. 85–98.
- [22] Waze Mobile. *Bezplatná, komunitne založená mapová, dopravná a navigačná aplikácia*. URL: <http://www.waze.com> (cit. 17. 01. 2016).
- [23] Yu Zheng a Xiaofang Zhou. *Computing with spatial trajectories*. Springer Science & Business Media, 2011.
- [24] Yu Zheng et al. *Geolife GPS trajectory dataset - User Guide*. Júl 2011. URL: <https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/>.

7 Summary

The aim of the thesis is to contribute to the area of context aware mobile applications. We summarize the analysis of current possibilities of user moving tracking. Then we design our own methods for user movement tracking, which we evaluate with respect to the accuracy of the recorded data.

We describe various modifications of localization systems like GPS, A-GPS or Glonass or Galileo. We present the GPS localization system with its limitations and errors in measurement. As an alternative method for user movement tracking, we provide methods based on GSM base transceiver station (BTS) or via WiFi networks. For each method, we mention their deficiencies, limitations, and errors in determining user location.

We present the design and implementation of the individual methods provided by the Android API: GPS method, Network method. Based on these, we introduce our own methods for user movement tracking: Locator method (combination of GPS and Network method) and FunFi (Locator with support of WiFi networks). For each method, we list the limitations that we solve in them. Based on introduced tests, we evaluate their functionality as well as user experience. We take into account the quality of recorded data. Our designed FunFi method allows sustained continuous user movement tracking with sufficient recording quality.

To evaluate each of the proposed methods, we proposed a method inspired by the Needleman-Wunsch algorithm applied in bio-informatics for aligning DNA sequences. We refer to proposed comparison methodology as COhaveCompare. We tested it on Sygic datasets and Microsoft Geolife data research data. Our results show that methods for aligning DNA sequenced could be successfully applied for comparing tracks recorded by users.