

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

VYUŽITIE ČASTICOVÉHO FILTRA NA LOKALIZÁCIU V
INDOOR PROSTREDÍ

Bakalárska práca

2019

Jakub Džama

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

VYUŽITIE ČASTICOVÉHO FILTRA NA LOKALIZÁCIU V
INDOOR PROSTREDÍ

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	9.2.1. Informatika
Školiace pracovisko:	Ústav informatiky
Vedúci práce:	RNDr. Miroslav Opiela



Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

- Meno a priezvisko študenta:** Jakub Džama
Študijný program: Informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: Bakalárska práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický
- Názov:** Využitie časticového filtra na lokalizáciu v indoor prostredí.
Názov EN: Indoor Localization using Particle Filter.
- Cieľ:**
1. Preskúmať a analyzovať rôzne implementácie a voľbu parametrov časticového filtra
 2. Preskúmať a porovnať známe prístupy lokalizácie v indoor prostredí s využitím časticového filtra.
 3. Implementovať a prakticky overiť použiteľnosť časticového filtra na lokalizáciu používateľa smartfónu vybaveného senzormi v indoor prostredí.
- Literatúra:**
1. Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on signal processing, 50(2), pp.174-188.
 2. Chen, Z., 2003. Bayesian filtering: From Kalman filters to particle filters, and beyond. Statistics, 182(1), pp.1-69.
 3. Radu, V. and Marina, M.K., 2013, October. Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting. In Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on (pp. 1-10). IEEE.
- Vedúci:** RNDr. Miroslav Opiela
Ústav : ÚINF - Ústav informatiky
Riaditeľ ústavu: prof. RNDr. Viliam Geffert, DrSc.
- Dátum schválenia:** 07.05.2019

Pod'akovanie

Rád by som pod'akoval RNDr. Miroslavovi Opielovi, vedúcemu tejto bakalárskej práce, za jeho čas, ústretovosť, ochotu, vedenie a cenné rady počas tvorby tejto práce.

Abstrakt

V súčasnosti je možné určiť polohu človeka pomocou systému GPS, ktorý je zabudovaný v takmer všetkých moderných zariadeniach. Kvôli slabej intenzite GPS signálu vo vnútri budov sa na lokalizáciu v interiéri využívajú iné metódy. Jednou z nich je PDR (Pedestrian Dead Reckoning), ktorá je založená na detekcii krokov používateľa využitím senzorov (najmä akcelerometra a kompasu) zabudovaných v smartfónoch. Dáta z týchto senzorov sú však zašumené, preto sa na modelovanie tejto nepresnosti používa Bayesovské filtrovanie. Najpoužívanejšou implementáciou Bayesovského filtrovania v indoor lokalizačných systémoch je časticový filter (Particle filter), ktorý je hlavným skúmaným predmetom tejto práce. Časticový filter je definovaný množinou častíc, pomocou ktorej modeluje aktuálnu pozíciu používateľa v budove. V tejto práci skúmame správanie časticového filtra v rôznych situáciách a analyzujeme vplyv voľby parametrov na priebeh a presnosť lokalizácie. Vytvorili sme aplikáciu na vizualizáciu časticového filtra, pričom pracujeme s vygenerovanými dátami a manuálne vytvorenými mapami budov, aby sme si vedeli modelovať ľubovoľné situácie typické pre indoor prostredie.

Kľúčové slová: *Indoor lokalizácia, Bayesovské filtrovanie, Časticový filter.*

Abstract

It is now possible to determine the location of a person using GPS, which is built in almost all modern devices. Due to the weak GPS signal inside buildings, other methods are used to locate indoors. One of them is the PDR (Pedestrian Dead Reckoning), which is based on the step detection by using inertial sensors (especially accelerometer and compass) embedded into smartphones. However, sensor measurements are noisy, so Bayesian filtering is used to model this inaccuracy. The most widely used implementation of Bayesian filtering in indoor localization systems is the Particle filter, which is the main subject of this work. The particle filter is defined by a set of particles to model the current user position in the building. In this work, we investigate the behaviour of particle filter in different situations and analyze the influence of parameter selection on the accuracy of localization. We have created a particle filter visualization application, while we were using the generated data and manually created maps of buildings to model any situation typical for indoor environment.

Keywords: *Indoor localization, Bayesian filtering, Particle filter.*

Obsah

Úvod	7
1 Indoor lokalizácia	8
2 Bayesovské filtrovanie	10
2.1 Implementácie Bayesovského filtrovania	11
2.1.1 Kalmanov filter	11
2.1.2 Mriežkový filter	12
2.1.3 Časticový filter	13
2.2 Lokalizačné systémy využívajúce Bayesovské filtrovanie	13
3 Časticový filter	16
3.1 Vzorkovanie	16
3.2 Prevzorkovanie	17
3.3 Časticový filter a indoor lokalizácia	20
4 Evaluácia	22
4.1 Vlastná implementácia časticového filtra	22
4.2 Prehľad aplikácie	23
4.3 Experimenty	25
4.3.1 Počet častíc	25
4.3.2 Dĺžka kroku a strata pozície používateľa	26
4.3.3 Nastavenie váh časticiam	30
4.3.4 Natočenie ako súčasť stavu častíc	31
4.3.5 Dynamická dĺžka kroku	32
Záver	34
Zoznam použitej literatúry	36
Prílohy	37

Úvod

Predstavme si situáciu, že vojdeme do veľkého obchodného centra, s účelom ísť do jedného konkrétneho obchodu spomedzi všetkých tisíc. Je pravdou, že vo veľkých budovách sa zvyknú nachádzať rôzne značenia, mapy a informačne tabule, ktoré pomáhajú ľuďom orientovať sa v budove. Niekedy je ale štruktúra budovy natoľko zložitá, že zorientovať sa v nej môže byť náročná úloha. Pri týchto okolnostiach by bolo príjemné mať vo svojom smartfóne aplikáciu, ktorá by nás dokázala navigovať k zadanému miestu v budove. Aj takéto prípady sú motiváciou pre vyvíjanie indoor lokalizácie.

V súčasnosti sa na lokalizáciu ľudí alebo rôznych objektov využíva Global Positioning System (GPS), ktorý je zabudovaný takmer vo všetkých moderných zariadeniach. Vo vnútri budov je však intenzita GPS signálu nízka, až nulová. Preto bolo potrebné vynásť iné metódy na určenie pozície človeka v budove. Niektoré z metód využívajú špeciálne doplnkové zariadenia (napr. riešenia založené na Bluetooth Low Energy (BLE) zariadeniach) [20], iné zas existujúcu infraštruktúru budovy (napr. metóda Wi-Fi fingerprinting merajúca intenzitu Wi-Fi signálu z prístupových bodov) [13]. Ďalším možným riešením je využitie počítačového videnia [12] (napr. pomocou kamery smartfónu). Iný prístup, nazývaný Pedestrian Dead Reckoning (PDR) [10], je založený na detekcii krokov používateľa využitím senzorov zabudovaných v smartfónoch.

Metóda PDR je podrobnejšie opísaná v prvej kapitole tejto práce. Druhá kapitola popisuje Bayesovské filtrovanie, ktorým sa modeluje neurčitost' spôsobená nepresnými a zašumenými dátami. Časticový filter, ktorý je hlavnou témou našej práce, je popísaný v tretej kapitole. Vo štvrtej kapitole predstavujeme našu implementáciu časticového filtra, aplikáciu na vizualizáciu lokalizácie používateľa v budove pomocou časticového filtra a tiež jednotlivé experimenty vykonané na analýzu našej implementácie časticového filtra.

1 Indoor lokalizácia

Vývoj smartfónov v dnešnej dobe veľmi rýchlo napreduje a popri zvyšovaní ich výkonov a vylepšovaní iných ďalších parametrov, začali sa do nich pridávať rôzne senzory ako súčasť hardvéru. Softvérovými senzormi nazývame také nástroje, ktoré spracúvajú dáta z iných sensorov. Medzi senzory, ktoré produkujú údaje využiteľné pri určovaní aktuálnej pozície používateľa, patria:

- **Akcelerometer** je hardvérový senzor, zabudovaný takmer v každom smartfóne. Meria zrýchlenie zariadenia v smere troch osí, vďaka čomu dokážeme zachytiť rôzne pohyby zariadenia, ako napr. chvenie, nakláňanie, kývanie... Pri indoor lokalizácii sa všeobecne využíva na rozpoznanie aktivity používateľa, ale najmä na detekciu kroku, odhad dĺžky kroku a natočenia používateľa.
- **Kompas** využíva údaje z hardvérového senzora magnetometra (s prípadnou spoluprácou akcelerometra) na výpočet uhlu medzi momentálnym natočením zariadenia a severom. Kvôli prítomnosti kovu v budovách kompas môže produkovať nepresné údaje.
- **Gyroskop** tiež dokáže zistiť natočenie zariadenia a v kombinácii s magnetometrom alebo akcelerometrom môže zvýšiť presnosť celkového merania.
- **Barometer** meria atmosférický tlak vo svojom okolí. Potom na základe výraznejšej zmeny tlaku je v prípade indoor lokalizácie možné bližšie určiť poschodie, na ktorom sa používateľ v danom čase nachádza.

Pedestrian Dead Reckoning (PDR) je metóda, ktorá je založená na detekcii krokov používateľa. Spracovaním dát zo sensorov zariadenia je tiež možné vypočítať približnú dĺžku detegovaného kroku a smer, ktorým sa používateľ pohol. Na základe týchto poznatkov a informácií o predošlej pozícii PDR počíta relatívnu pozíciu používateľa vzhľadom k predchádzajúcim odhadovaným pozíciám, a preto pre získanie absolútnej pozície používateľa je potrebné túto metódu skombinovať s inou, ktorá absolútnu pozíciu je schopná poskytnúť. Vypočítaná dĺžka kroku a smer však nemusia zodpovedať skutočným hodnotám, pretože dáta zo sensorov sú ovplyvnené šumom, a teda

počítaná pozícia používateľa nie je presná. Dokonca nepresnosť lokalizácie sa zvyšuje po každom detegovanom kroku. Preto sa častokrát odhad pozície zlepšuje na základe informácii o podrobnej štruktúre budovy, alebo sa PDR kombinuje s inými lokalizačnými metódami (najčastejšie s Wi-Fi fingerprinting).

Podstatnou časťou PDR je detekcia kroku. Pri riešení tejto úlohy je potrebné vziať do úvahy, že používateľ môže držať smartfón v rôznej polohe, napr. drží ho vodorovne v ruke, má ho pri uchu počas telefonovania, nesie ho v taške, vo vrecku, atď. Z tohto dôvodu potom počet správne detegovaných krokov môže byť nižší.

Ešte zložitejšou úlohou je odhadnúť dĺžku kroku používateľa. Nielenže je dĺžka kroku odlišná medzi jednotlivými používateľmi, ale aj ten istý používateľ môže robiť kroky rôznej dĺžky. Vo všeobecnosti sú dva prístupy k odhadovaniu dĺžky kroku:

- *Statický model* uvažuje, že dĺžka kroku je konštanta. Na začiatku sa jej priradí nejaká hodnota (napr. na základe výšky používateľa), ale počas behu programu sa nemení. Takýto prístup častokrát vedie k neuspokojivým výsledkom.
- *Dynamický model* ráta s premenlivou dĺžkou kroku. Jeden z prístupov predpokladá, že dĺžka kroku a frekvencia krokov sú v lineárnej korelácii [19], [14]. Cho a Park vo svojom riešení [5] odhadovali dĺžku kroku s využitím neurónovej siete. Iné riešenie počíta dĺžku kroku na základe rozdielov lokálnych extrémov v dátach z akcelerometra [10].

Kang a Han v článku [10] predstavujú svoje riešenie SmartPDR, ktoré je prispôbené pre smartfóny a je založené na metóde PDR. Na detekciu pohybu používateľa použili akcelerometer, magnetometer a gyroskop, ktoré sú prítomné vo väčšine dnešných smartfónoch. Toto riešenie bolo navrhnuté tak, aby bolo nezávislé od informácií o mape, či infraštruktúre budovy. Dĺžku kroku odhadovali na základe dát z akcelerometra a na odhad smeru použili algoritmus vychádzajúci z ich predošlej práce [11], ktorý je založený na korelácii magnetometra a gyroskopu, a tiež sa dal dôraz na určenie najlepšieho momentu, kedy presne počas detegovaného kroku počítať daný odhad smeru.

2 Bayesovské filtrovanie

Tým, že senzory smartfónu produkujú zašumené dáta, nedokážeme určiť presnú polohu používateľa v budove so stopercentnou istotou. Preto sa pri riešení lokalizačného problému častokrát využíva Baysovské filtrovanie [1], ktoré modeluje stav systému (polohu používateľa) pravdepodobnostným rozdelením. Cieľom Bayesovského filtrovania je určiť, resp. aproximovať funkciu hustoty pravdepodobnosti p , ktorá popisuje aktuálny stav systému. Za predpokladu, že táto funkcia nezávisí od predchádzajúcich stavov, ale iba od posledného stavu, je vyhovujúce použiť rekurzívny Bayesovský filter, implementovaný ako iteratívny algoritmus, ktorý prijaté dáta spracúva sekvenčne.

Vo všeobecnosti na konci iterácie $k - 1$ máme k dispozícii postupnosť stavov $\mathbf{x}_{0:k-1} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{k-1}\}$ a meraní $\mathbf{z}_{1:k-1} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{k-1}\}$. Potom vektory \mathbf{x}_k a \mathbf{z}_k v k -tej iterácii sú dané rovnicami:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k), \quad (2)$$

kde $\mathbf{f}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$, $\mathbf{h}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_z}$ sú známe, nie nutne lineárne funkcie a \mathbf{v}_{k-1} , \mathbf{w}_k modelujú procesný šum a šum merania. Potom n_x, n_z, n_v, n_n sú v poradí dimenzie stavu, merania, procesného šumu a šumu merania. [1]

V prípade riešenia problému indoor lokalizácie v rámci jedného podlažia by jednoduchou reprezentáciou stavu $\mathbf{x}_t \in \mathbb{R}^2$ mohli byť súradnice používateľa $\langle x, y \rangle$. Samozrejme, stavový vektor môže obsahovať aj iné údaje, ako napr. tretiu súradnicu, natočenie, rýchlosť, atď. Prechodová funkcia \mathbf{f}_k je definovaná pomocou PDR prístupu, teda pri detekcii kroku sa aktuálna pozícia vypočíta na základe predošlej pozície s využitím odhadu dĺžky kroku a natočenia.

Rekurzívny Bayesovský filter pri každej iterácii vykonáva tieto dva kroky:

- **Predikcia** na základe modelu systému a predošlého stavu odhaduje funkciu hustoty pravdepodobnosti aktuálneho stavu. Táto operácia vo všeobecnosti defor-

muje a rozširuje hustotu pravdepodobnosti p . Predikcia v čase k je charakterizovaná touto rovnicou:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \quad (3)$$

V prípade indoor lokalizácie táto fáza algoritmu odhaduje pozíciu používateľa na základe prijatých dát zo senzorov zariadenia, kde $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ interpretuje model pohybu používateľa.

- **Aktualizácia (oprava)** upravuje predikovanú hustotu pravdepodobnosti stavu z hľadiska dodatočných informácií z nových meraní využitím Bayesovej vety. Tento krok algoritmu je daný rovnicou:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\alpha_k}, \quad (4)$$

kde $p(\mathbf{z}_k | \mathbf{x}_k)$ reprezentuje model merania. Splnenie podmienky hustoty pravdepodobnosti, tj. aby súčet pravdepodobností všetkých stavov bol rovný 1, zabezpečuje normalizačná konštanta $\alpha_k = \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k$. [1]

Počiatočná hustota pravdepodobnosti $p(\mathbf{x}_0 | \mathbf{z}_0) = p(\mathbf{x}_0)$ môže byť získaná na základe Wi-Fi signálu, rozpoznanej aktivity používateľa, alebo môže byť zadaná manuálne. V prípade, že nemáme žiadnu informáciu o počiatočnej pozícii používateľa, hustota pravdepodobnosti $p(\mathbf{x}_0)$ môže byť aj uniformná, teda rovnomerne rozložená po celom priestore definovanom mapou budovy.

2.1 Implementácie Bayesovského filtrovania

Medzi najznámejšie implementácie Bayesovských filtrov patrí Kalmanov filter, mriežkový filter, ktoré za istých podmienok poskytujú optimálne Bayesovské riešenie počítajúce presnú funkciu hustoty pravdepodobnosti, a tiež časticový filter. Keď analytické riešenie neexistuje, využívajú sa metódy, ktoré sa približujú k optimálnemu riešeniu, ako napr. rozšírený Kalmanov filter, aproximačný mriežkový filter a časticový filter.

2.1.1 Kalmanov filter

Kalmanov filter predpokladá, že posteriórna hustota pravdepodobnosti je v každom čase k normálnym rozdelením, tj. $p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k; \mu_k, \Sigma_k)$, kde μ_k je strednou hodno-

tou rozdelenia a Σ_k je $d \times d$ matica disperzií (kde d je dimenzia stavu). Je dokázateľné, že ak $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ je normálne rozdelenie, tak $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ je tiež normálne rozdelenie, ak sú dodržané tieto podmienky:

- $\mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$ je známa lineárna funkcia \mathbf{x}_{k-1} a \mathbf{v}_{k-1} ,
- $\mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k)$ je známa lineárna funkcia \mathbf{x}_k a \mathbf{w}_k ,
- \mathbf{v}_{k-1} a \mathbf{w}_k sú vzorky vytiahnuté z gaussovských distribúcií so známymi parametrami.

Potom rovnice 1 a 2 je možné zapísať v tvare:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \quad (5)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k, \quad (6)$$

kde \mathbf{F}_k a \mathbf{H}_k sú známe matice definujúce lineárne funkcie. Za týchto podmienok Kalmanov filter poskytuje optimálne riešenie Bayesovského filtrovania, teda počíta presnú hustotu pravdepodobnosti. [1]

Silnou stránkou Kalmanovho filtra je jeho implementácia predikcie a aktualizácie pomocou efektívnych maticových operácií. Nevýhodou však je, že dokáže presne modelovať len unimodálne distribúcie podobné gaussovej krivke. Avšak, indoor lokalizácia je považovaná za problém, kde je stav negaussovský a nelineárny. [6] Preto sa na modelovanie aktuálnej polohy používateľa v budove častejšie používajú mriežkový alebo časticový filter. Kalmanov filter alebo rozšírený Kalmanov filter sa ale častokrát používa na iné úlohy v rámci indoor navigácie, ako napr. na kalibráciu modelu dĺžky kroku [18], rozpoznávanie aktivity používateľa na základe dát zo sensorov, či na určenie podlažia, kde sa používateľ práve nachádza.

2.1.2 Mriežkový filter

Podstatou mriežkového filtra (Grid-based filter) je diskretizácia stavového modelu systému do n stavov, reprezentovaných uniformne rozloženými n bunkami mriežky. V prípade určovania aktuálnej pozície používateľa v budove v rámci jedného podlažia, mapa daného podlažia budovy by mohla byť rozdelená do 2D mriežky. pre stred každej bunky mriežky sa potom pri každom detegovanom kroku používateľa počíta pravdepodobnosť, že používateľ sa nachádza práve na danom mieste. Takáto mriežka poskytuje diskretnú aproximáciu funkcie hustoty pravdepodobnosti momentálnej pozície používateľa.

V prípade, že skúmaný systém má konečne veľa možných stavov, mriežkový filter poskytuje optimálne riešenie Bayesovského filtrovania. Odhad predikovanej funkcie hustoty pravdepodobnosti sa dá zlepšiť zmenšením veľkosti buniek mriežky, čím sa zvýši počet simulovaných stavov. Týmto spôsobom sa ale tiež zvýši výpočtová zložitosť algoritmu, ktorá je najväčším problémom mriežkového filtra. Tento filter nie je tiež vhodný na problémy s viacrozmernými stavmi, pretože výpočtová zložitosť exponenciálne rastie so zvyšujúcim sa počtom dimenzií stavov. [7]

Galčík a Opiela vo svojej práci [8] skúmali vplyv diskretizácie, štruktúry mriežky, presnosti odhadu dĺžky kroku a spôsobu modelovania budovy na presnosť lokalizácie pomocou mriežkového filtra. V rámci tejto analýzy skúsili napr. aj zmeniť tvar bunky mriežky zo štvorca na šesťuholník. Využitie mriežkového filtra je typické pre oblasť robotiky [2].

2.1.3 Časticový filter

Idea časticového filtra (Particle filter) spočíva v modelovaní hustoty pravdepodobnosti p pomocou množiny častíc. Každá častica je reprezentovaná svojím stavom (v prípade lokalizácie pozíciou) a váhou, ktorá udáva, s akou pravdepodobnosťou sa používateľ práve nachádza v danom stave. Potom aj pomocou menšieho počtu častíc sa môžeme s väčšou presnosťou sústrediť na oblasti, kde sa s najväčšou pravdepodobnosťou používateľ nachádza. Tým pádom je časticový filter oproti mriežkovému filtru výpočtovo menej náročný. Ďalšou výhodou časticového filtra je, že na rozdiel od Kalmanovho filtra dokáže modelovať aj hustotu pravdepodobnosti, ktorá nie je gaussovská, ani unimodálna.

Hafner et al. v práci [9] porovnávajú presnosť lokalizácie Kalmanovho a časticového filtra v systéme, ktorý spracúva Wi-Fi signál z prístupových bodov a ten kombinuje s dátami zo senzorov zariadenia. Z testovania autori vyvodili záver, že časticový filter dosahuje v lokalizácii lepšie výsledky, najmä ak je v budove málo Wi-Fi prístupových bodov, alebo ak nie sú k dispozícii kvalitné dáta.

2.2 Lokalizačné systémy využívajúce Bayesovské filtrovanie

Radu a Marina vo svojej práci [16] prezentujú projekt HiMLoc, ktorý využíva kombináciu metód PDR s rozpoznávaním aktivity používateľa a crowd-sourced Wi-Fi fingerprinting. Spojením týchto dvoch metód chcú znížiť negatíva oboch prístupov a tým zlepšiť presnosť lokalizácie. HiMLoc potrebuje pre svoju funkčnosť prístup

k dát z akcelerometra, kompasu a Wi-Fi karty, na základe ktorých dokáže sledovať meniacu sa pozíciu používateľa v budove. Zjednotenie viacerých metód realizuje pomocou časticového filtra. Každá častica má svoj PDR komponent, ktorý riadi jej pohyb. Váha častíc sa potom aktualizuje na základe prijatých Wi-Fi odtlačkov a rozpoznanej aktivity používateľa. Výhodou HiMLoc-u je tiež, že nevyžaduje detailný model budovy. Stačí mu informácia o pozícií schodov, výťahov, rohov, hlavných vstupov a výške jednotlivých podlaží. Evaluácia HiMLoc-u sa uskutočnila v troch rôznych scenároch: lokalizácia na jednom podlaží budovy s veľkou tréningovou množinou Wi-Fi odtlačkov, lokalizácia v rámci viacerých podlaží a lokalizácia v novej budove bez akýchkoľvek počiatočných Wi-Fi odtlačkov. Z testovania zistili, že medián celkovej chyby lokalizácie pri využití HiMLoc-u je približne 3 metre.

Chen et al. v článku [4] predstavili svoje riešenie lokalizačného problému. Podobne ako projekt HiMLoc, aj ich riešenie je založené na využití PDR, Wi-Fi signálu a orientačných bodov. Namiesto metódy Wi-Fi fingerprinting však použili WPL (Weighted Path Loss) algoritmus a namiesto časticového filtra použili výpočtovo menej náročný Kalmanov filter. Taktiež oproti HiMLoc-u, ktorý odhadoval dĺžku kroku používateľa na základe lineárnej korelácie s frekvenciou krokov, toto riešenie zisťovalo dĺžku kroku pomocou algoritmu priamo z dát zo sensorov zariadenia.

Riešenie, predstavené autormi Li, Zhao et al. v ich práci [14], využíva na lokalizáciu iba dáta zo sensorov smartfónu a relatívne podrobný model budovy. Dané riešenie je špeciálne odhadovaním dĺžky kroku používateľa, ktoré je založené na lineárnom modeli frekvencie krokov. Tento lineárny model však nie je stály, ale vyvíja sa počas priebehu lokalizácie. Zrealizované je to pomocou časticového filtra, ktorý popri modelovaní hustoty pravdepodobnosti, odhaduje správny lineárny model frekvencie a dĺžky kroku. Stav každej častice je reprezentovaný pozíciou a dvojicou koeficientov (a, b) lineárneho modelu. Na začiatku sa časticiam nastavujú rôzne hodnoty koeficientov. Ak počas priebehu lokalizácie niektorá častica prejde cez stenu, zanikne. Potom sa objaví v blízkosti inej žijúcej častice, od ktorej si skopíruje dvojicu koeficientov, ak zanikla kvôli zle odhadnutému kroku. Takýmto spôsobom budú žiť častice s dobrou dvojicou koeficientov a systém tak dokáže dynamicky zlepšovať lineárny model frekvencie a dĺžky kroku.

Rai et al. vytvorili projekt Zee [17], ktorý dynamicky, bez aktívneho vstupu používateľa vytvára databázu Wi-Fi odtlačkov v budove. Na základe PDR metódy (s využitím akcelerometra, kompasu a gyroskopu) Zee sústavne sleduje aktuálnu používateľovu pozíciu pomocou časticového filtra a súčasne, na pozadí aplikácie, skenuje Wi-Fi signál z prístupových bodov. Tento projekt si však vyžaduje poznať podrobnú mapu budovy s vyznačenými chodbami a stenami.

Liu et al. v článku [15] prezentujú svoje riešenie indoor lokalizácie, ktoré využíva

dáta zo sensorov smartfónu na identifikovanie pohybu používateľa a odhad aktuálnej pozície zlepšuje aj na základe meraní intenzity Wi-Fi signálu. Na odhad pozície autori použili dva algoritmy: mriežkový filter a *Viterbi* algoritmus, ktorých efektívnosť a presnosť lokalizácie vo svojej práci skúmali.

3 Časticový filter

Hlavnou myšlienkou časticového filtra je modelovať hustotu pravdepodobnosti p pomocou množiny častíc $\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n$, kde \mathbf{x}_k^i reprezentuje stav i -tej častice v čase k . V prípade indoor lokalizácie stav častice môže byť charakterizovaný napr. polohou a prípadným natočením, teda $\mathbf{x}_k^i = \langle x_k^i, y_k^i, \theta_k^i \rangle$. Váha častice w_k^i je nezáporná hodnota, ktorá udáva, s akou pravdepodobnosťou sa používateľ nachádza v stave \mathbf{x}_k^i v čase k . Pre súčet všetkých váh v čase k platí $\sum_{i=1}^n w_k^i = 1$, nakoľko všetky častice spolu reprezentujú odhadovanú hustotu pravdepodobnosti aktuálneho stavu. Tento filter nie je optimálnym riešením problému Bayesovského filtrovania, ale zvyšovaním počtu častíc sa k nemu približuje.

3.1 Vzorkovanie

Vo všeobecnosti sa množina častíc v každej iterácii vytvára vzorkovaním z návrhovej (*proposal*) distribúcie q (tiež nazývanej *importance density*), keďže väčšinou je nemožné vzorkovať zo skutočnej posteriórnej hustoty pravdepodobnosti. Výber návrhovej distribúcie má veľký vplyv na celkový výkon časticového filtra.

Časticový filter je rekurzívny Bayesovský filter založený na Monte Carlo simuláciách. Existuje viacero typov časticového filtra, pričom každý z nich vychádza zo SIS (Sequential Importance Sampling) algoritmu. Schéma jednej iterácie SIS algoritmu v čase k vyzerá takto [1]:

Algoritmus 1 SIS Časticový filter

Input: $\{\langle \mathbf{x}_{k-1}^i, w_{k-1}^i \rangle\}_{i=1}^n$ - množina častíc v čase $k - 1$

\mathbf{z}_k - merania z rovnice 2

Output: $\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n$ - výsledná množina častíc v čase k

```
1: procedure PROCESS( $\{\langle \mathbf{x}_{k-1}^i, w_{k-1}^i \rangle\}_{i=1}^n, \mathbf{z}_k$ )
2:   for  $i \leftarrow 1, n$  do
3:     Draw  $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$            ▷ Vytiahne sa vzorka z distribúcie  $q$ 
4:      $\hat{w}_k^i \leftarrow w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$            ▷ Nastavenie váhy častice
5:      $sumWeight \leftarrow \sum_{i=1}^n \hat{w}_k^i$            ▷ Súčet váh všetkých častíc
6:     for  $i \leftarrow 1, n$  do
7:        $w_k^i \leftarrow \frac{\hat{w}_k^i}{sumWeight}$            ▷ Normalizácia váh
8:   return  $\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n$ 
```

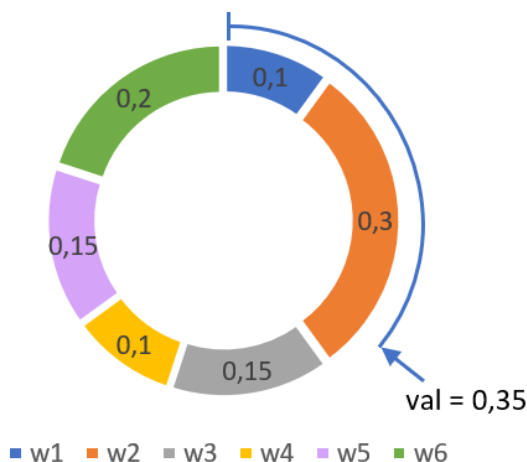
Častým problémom takéhoto algoritmu je *degeneracy phenomenon*, kedy po niekoľkých iteráciách všetky častice nadobudnú zanedbateľnú váhu a len zopár častíc ju má veľmi veľkú (pozri obr. 2). To spôsobuje, že veľa výpočtovej sily sa použije na aktualizáciu častíc, ktoré na celkovú aproximáciu $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ majú takmer nulový vplyv. Priamočiarym riešením, ako zredukovať efekt tohto problému, je zvýšenie počtu častíc, čo je ale vo väčšine prípadoch nepraktické alebo nemožné kvôli výpočtovým obmedzeniam zariadenia.

3.2 Prezorkovanie

Častokrát sa na riešenie problému *degeneracy phenomenon* využíva prezorkovanie (resampling). Prezorkovaním vygenerujeme novú množinu častíc zo starej takým spôsobom, že čím má častica väčšiu váhu, tým je väčšia pravdepodobnosť, že táto častica bude vybraná do novej množiny.

Elegantný spôsob, ako implementovať prezorkovanie, je reprezentovať starú množinu častíc kruhom (*resampling wheel*), pričom na obr. 1 je možné vidieť, že jednotlivé kruhové výseky tvoria jednotlivé váhy častíc. Keďže pred krokom prezorkovania sa váhy častíc normalizujú, obvod takéhoto kruhu je rovný 1. Hlavnou myšlienkou je teda vygenerovať náhodné číslo z intervalu $(0, 1)$ a prechádzať od začiatku jednotlivými váhami, kým súčet doteraz prezretých váh nepresiahne náhodne vygenerovanú hodnotu. Do novej množiny sa priradí tá častica, ktorej váhu sme prezreli ako poslednú. Tento algoritmus sa opakuje n -krát, pričom každej častici vybranej do novej množiny sa nastaví uniformná váha, ako je to možné vidieť v algoritme 2.

Resampling wheel



Obr. 1: Resampling wheel - jedna iterácia prevzorkovania časticového filtra so šiestimi časticami, ktorým prislúchajú váhy $w_1 = 0.1, w_2 = 0.3, \dots, w_6 = 0.2$. Tieto váhy reprezentujú pravdepodobnosť, že daná častica v aktuálnej iterácii bude vybraná do novej množiny častíc, napr. pre časticu s váhou w_3 je táto pravdepodobnosť rovná 15%. V danej iterácii náhodnému číslu z intervalu $(0, 1)$ val bola priradená vygenerovaná hodnota 0.35 a do novej množiny bola vybraná častica s váhou w_2 .

Algoritmus 2 Prevzorkovanie (Resampling)

Input: $\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n$ - množina častíc po k -tej iterácii algoritmu 1

Output: $\{\langle \mathbf{x}_k^j, w_k^j \rangle\}_{j=1}^n$ - nová množina častíc po prevzorkovaní

```

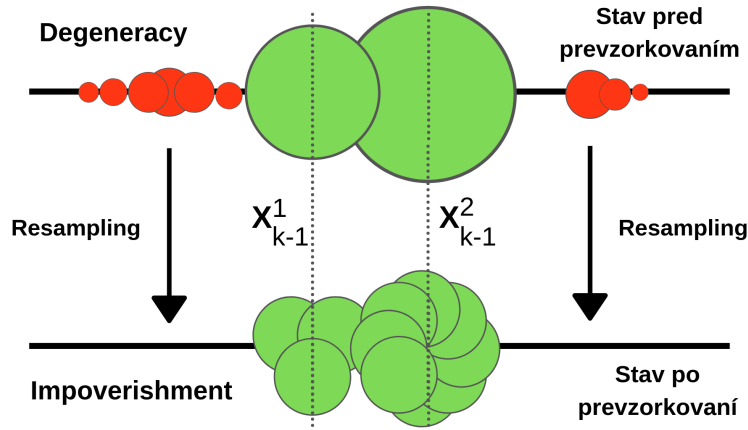
1: procedure RESAMPLE( $\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n$ )
2:   for  $j \leftarrow 1, n$  do
3:      $i \leftarrow 0$ 
4:     Draw  $val \sim \mathcal{U}(0, 1)$        $\triangleright$  Vygenerovanie náhodného čísla z intervalu  $(0, 1)$ 
5:     repeat
6:        $i \leftarrow i + 1$ 
7:        $val \leftarrow val - w_k^i$ 
8:     until  $val > 0$ 
9:      $\mathbf{x}_k^j \leftarrow \mathbf{x}_k^i$            $\triangleright$  Priradenie častice
10:     $w_k^j \leftarrow \frac{1}{n}$        $\triangleright$  Nastavenie uniformnej váhy
11:   return  $\{\langle \mathbf{x}_k^j, w_k^j \rangle\}_{j=1}^n$ 

```

Chen vo svojom článku [3] rozpracoval viacero implementácií prevzorkovania, ako

napr. *Multinomial resampling*, *Residual resampling*, *Systematic resampling*, či *Local Monte Carlo resampling*.

Prevzorkovaním sa eliminujú častice so zanedbateľnou váhou, pričom ale niektoré častice s veľkou váhou sa pravdepodobne budú nachádzať v novej množine viackrát. Časté používanie prevzorkovania však môže spôsobovať ochudobnenie vzorky - *sample impoverishment*, ktoré sa prejavuje stratou diverzity medzi časticami. Totiž po niekoľkých iteráciách časticového filtra s krokom prevzorkovania sa môže stať, že všetky častice sa zoskupia do jedného zhluku (pozri obr. 2). Tento problém je častý najmä v systémoch s malým procesným šumom. Jedným z riešení, ako predísť tomuto problému, je vykonať krok prevzorkovania len vtedy, keď hodnota vypočítaného odhadu $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^n (w_k^i)^2}$ klesne pod nami určený prah.



Obr. 2: *Degeneracy phenomenon* (horná časť) a *sample impoverishment* (dolná časť) - častice s jednorozmerným stavom sú vizualizované kruhmi, pričom veľkosť a farba kruhu reprezentuje váhu danej častice. Počas prevzorkovania bola do novej množiny 3-krát vybraná častica so stavom \mathbf{x}_{k-1}^1 a 8-krát častica so stavom \mathbf{x}_{k-1}^2 .

Jedným z najpoužívanejších druhov časticového filtra je SIR (Sampling Importance Resampling) algoritmus, ktorý je charakteristický vykonávaním kroku prevzorkovania v každej iterácii a výberom návrhovej distribúcie ako predchádzajúcej distribúcie: $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$. Častica $\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$ sa môže teda nastaviť pomocou procesnej funkcie $\mathbf{x}_k^i = \mathbf{f}_k(\mathbf{x}_{k-1}^i, \mathbf{v}_{k-1}^i)$, kde \mathbf{v}_{k-1}^i je vygenerovaná vzorka z distribúcie procesného šumu. Váhy častíc sa nastavujú na základe meraní a v prípade vybratia takejto návrhovej distribúcie pre ne potom platí:

$$w_k^i = w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i) \quad (7)$$

Avšak tým, že pri prevzorkovaní sa všetkým časticiam v novej množine priradí rovnaká konštantná váha, aplikovanie kroku prevzorkovania pri každej iterácii spôsobí, že $w_{k-1}^i = \frac{1}{n}$, a preto:

$$w_k^i = p(\mathbf{z}_k | \mathbf{x}_k^i). \quad (8)$$

Algoritmus 3 ilustruje proces jednej iterácie SIR algoritmu:

Algoritmus 3 SIR Časticový filter

Input: $\{\langle \mathbf{x}_{k-1}^i, w_{k-1}^i \rangle\}_{i=1}^n$ - množina častíc v čase $k - 1$

\mathbf{z}_k - merania z rovnice 2

Output: $\{\langle \mathbf{x}_k^j, w_k^j \rangle\}_{j=1}^n$ - výsledná množina častíc v čase k

```

1: procedure PROCESS( $\{\langle \mathbf{x}_{k-1}^i, w_{k-1}^i \rangle\}_{i=1}^n, \mathbf{z}_k$ )
2:   for  $i \leftarrow 1, n$  do
3:     Draw  $\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$   ▷ Vytiahnutie vzorky z predchádzajúcej distribúcie
4:      $\hat{w}_k^i \leftarrow p(\mathbf{z}_k | \mathbf{x}_k^i)$   ▷ Nastavenie váhy častice
5:      $sumWeight \leftarrow \sum_{i=1}^n \hat{w}_k^i$   ▷ Súčet váh všetkých častíc
6:     for  $i \leftarrow 1, n$  do
7:        $w_k^i \leftarrow \frac{\hat{w}_k^i}{sumWeight}$   ▷ Normalizácia váh
8:      $\{\langle \mathbf{x}_k^j, w_k^j \rangle\}_{j=1}^n \leftarrow \text{RESAMPLE}(\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n)$   ▷ Prevzorkovanie (alg. 2)
9:   return  $\{\langle \mathbf{x}_k^j, w_k^j \rangle\}_{j=1}^n$ 

```

3.3 Časticový filter a indoor lokalizácia

Časticový filter je najčastejšie využívaný Bayesovský filter v indoor lokalizačných systémoch. Stav častice je v takýchto systémoch najčastejšie reprezentovaný polohou používateľa, teda súradnicami x, y a v prípade lokalizácie v rámci viacerých podlaží aj súradnicou z . V niektorých implementáciách však stav častíc môže obsahovať navyše natočenie, zrýchlenie, dĺžku kroku, atď.

Vo všeobecnosti sa častice pri každej iterácii časticového filtra vzorkujú z návrhovej distribúcie. V prípade indoor lokalizácie využívajúcej PDR prístup tento postup vedie k tomu, že častice si aktualizujú svoju polohu - na základe prijatých dát z pohybových senzorov sa častice pohnú o odhadovanú dĺžku kroku odhadovaným smerom, pričom k daným odhadom sa pripočíta malá náhodná hodnota, aby sa tým modelovala nepresnosť dát.

Váhy častíc sa potom nastavujú na základe dostupných meraní - intenzita Wi-Fi signálu z jednotlivých prístupových bodov, rozpoznanie aktivity používateľa (napr.

kráča po schodoch, ide vo výťahu, atď.), alebo sa tiež často berie do úvahy model budovy, kde sa časticiam, nachádzajúcim sa na nedostupnom mieste, nastaví nulová váha.

Po normalizácii váh sa môže vykonať krok prevzorkovania a následne sa odhadne aktuálna pozícia používateľa, napr. pomocou váženého priemeru. Tento postup sa opakuje po každom detegovanom kroku.

4 Evaluácia

Hlavným cieľom tejto bakalárskej práce je preskúmať správanie časticového filtra v rôznych situáciách pri indoor lokalizácii a analyzovať vplyv voľby parametrov na priebeh a presnosť lokalizácie používateľa v budove. V práci uvažujeme, že máme prístup k dátam senzorov smartfónu, z ktorých vieme vyextrahovať dĺžku a smer kroku, a tiež predpokladáme dostupnosť detailnej mapy budovy. Aby sme vedeli modelovať ľubovoľné situácie, pracujeme so simulovanými dátami a manuálne vytvorenými 2D mapami budov, ktoré zachycujú podmienky typické pre indoor prostredie.

4.1 Vlastná implementácia časticového filtra

V našej práci sme implementovali časticový filter v programovacom jazyku Java, najmä kvôli možnému budúcemu testovaniu algoritmu na smartfóne s operačným systémom Android. Uvažujme teda množinu častíc $\{\langle \mathbf{x}_k^i, w_k^i \rangle\}_{i=1}^n$, pričom stav každej častice je reprezentovaný dvojicou $\mathbf{x}_k^i = \langle x_k^i, y_k^i \rangle$, kde $x_k^i, y_k^i \in \mathbb{R}$ predstavujú 2D pozíciu používateľa v budove. Schéma jednej iterácie algoritmu po detegovaní kroku v čase k vyzerá nasledovne:

1. **Posun častíc:** na základe prijatých dát (natočenie a odhadovaná dĺžka kroku používateľa) zo senzorov sa všetky častice pohnú o odhadovanú dĺžku daným smerom. Keďže ale prijaté dáta môžu byť zašumené, k danému natočeniu a dĺžke kroku sa pripočítajú náhodné hodnoty vygenerované z normálneho rozdelenia s nulovou strednou hodnotou. Tento krok algoritmu je reprezentovaný danými rovnicami:

$$x_k^i = x_{k-1}^i + (l_k + \delta_k^i) \cos(\alpha_k + \beta_k^i), \quad (9)$$

$$y_k^i = y_{k-1}^i + (l_k + \delta_k^i) \sin(\alpha_k + \beta_k^i), \quad (10)$$

kde α_k je vypočítaný smer, ktorým sa používateľ pohol, l_k je odhadovaná dĺžka

kroku a vzorky $\beta_k^i \sim \mathcal{N}(0, \sigma_{\alpha_k}^2)$, $\delta_k^i \sim \mathcal{N}(0, \sigma_{l_k}^2)$ modelujú šum natočenia a dĺžky kroku.

Keďže v našej aplikácii pracujeme s umelými dátami, zašumené dáta z kompasu musia byť vygenerované. Na základe niekoľkých vzoriek z reálnych dát sme zistili, že šum z kompasu sa dá modelovať normálnym rozdelením, pričom priemerne smerodajná odchýlka nadobúdala hodnotu 7.66° . Smer, ktorým sa používateľ pohol, sa potom počíta: $\alpha_k = \alpha_k^* + \gamma_k$, kde α_k^* je skutočný smer a $\gamma_k \sim \mathcal{N}(0, 7.66)$ modeluje daný šum. Dĺžka kroku l_k je v našej aplikácii nastaviteľná konštanta.

- Aktualizácia váh:** na základe dostupného modelu mapy budovy (napr. obrázok) sa časticiam, ktoré sa nachádzajú na nedostupných miestach, priradí váha $\hat{w}_k^i = 0$. Či sa častica nachádza na dostupnom mieste, alebo mimo budovy, v našej aplikácii kontrolujeme na základe farby pixelu, ktorý zodpovedá pozícii danej častice.
- Normalizácia váh:** $w_k^i = \frac{\hat{w}_k^i}{\sum_{j=1}^n \hat{w}_k^j}$
- Prevzorkovanie:** Podľa algoritmu 2
- Odhad pozície:** Mapa budovy sa rozkrája do mriežky a následne, ak dve susediace bunky obsahujú nejaké častice, spoja sa do jednej. Takýmto spôsobom sa častice rozdelia do klastrov. Potom sa vyberie klaster s najväčším súčtom váh svojich častíc a z častíc, ktoré patria do daného klastra, sa vypočíta vážený priemer pozícií. Výslednú hodnotu vyhlásime za nový odhad aktuálnej pozície používateľa.

4.2 Prehľad aplikácie

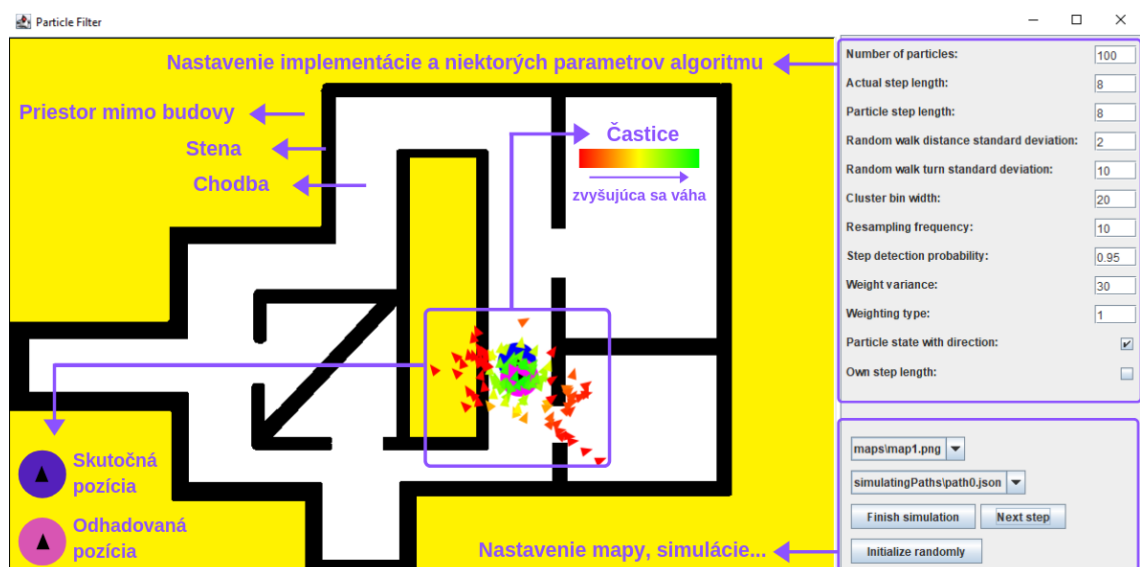
V rámci tejto práce sme vytvorili demo aplikáciu (pozri obr. 3) v programovacom jazyku Java, s využitím frameworkov Swing a JPAZ2¹. Pomocou tejto aplikácie vieme vizualizovať simuláciu časticového filtra pri určovaní pozície používateľa v budove. Pohyb používateľa sa realizuje pomocou šípok na klávesnici, alebo na základe umelo vygenerovaných dát.

Popri vizualizácii časticového filtra, aplikácia počíta chybu odhadu pozície po každom kroku a na bočnom paneli umožňuje zvoliť si mapu budovy, inicializovať algoritmus so známou alebo neznámou počiatočnou pozíciou používateľa, a tiež meniť implementáciu a hodnoty niektorých parametrov algoritmu (v zozname sú pri jednotlivých parametroch uvedené ich prednastavené hodnoty):

¹<https://paz1a.ics.upjs.sk/storage/jpaz/doc/index.html>

- počet častíc = 100,
- skutočná dĺžka kroku používateľa = 80 cm,
- odhadovaná dĺžka kroku = 80 cm,
- smerodajná odchýlka šumu dĺžky kroku = 3 cm,
- smerodajná odchýlka šumu natočenia = 10° ,
- veľkosť klastrov pri odhadovaní pozície = 2 m,
- frekvencia, ako často sa má vykonávať krok prevzorkovania = 1,
- pravdepodobnosť detekcie kroku = 95%,
- spôsob nastavovania váh,
- model častíc s vlastným natočením
- model častíc s vlastnou dĺžkou kroku

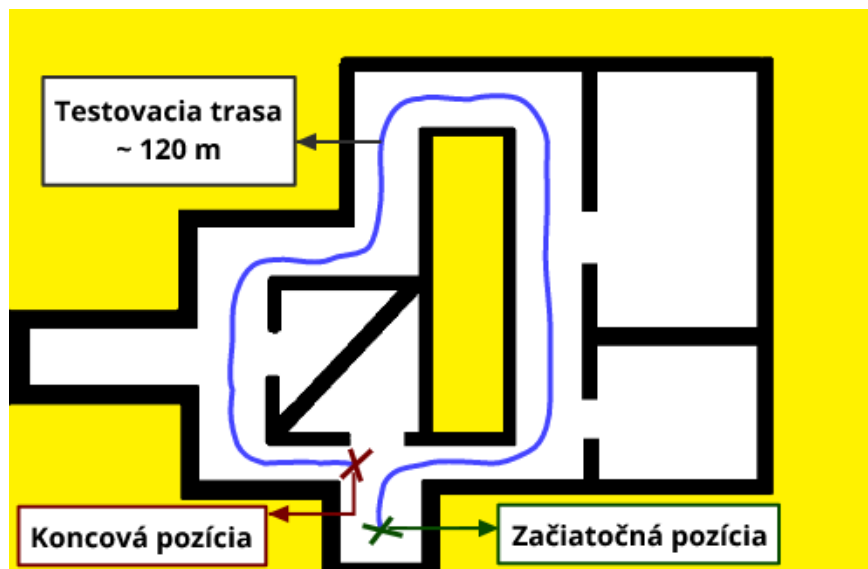
Taktiež je možné spustiť simuláciu lokalizácie s aktuálne nastavenými parametrami algoritmu na vybratej, vopred pripravenej trase. Po skončení simulácie sa do json súboru uloží nastavenie parametrov, jednotlivé odhadované pozície a k nim vypočítané chyby oproti skutočnej pozícii.



Obr. 3: Aplikácia na analýzu časticového filtra a na porovnanie rôznych konfigurácií parametrov.

4.3 Experimenty

Presnosť lokalizácie a celkové správanie konkrétnej implementácie časticového filtra pri aktuálne nastavených parametroch testujeme na pripravenej trase, ktorá je znázornená na obr. 4.



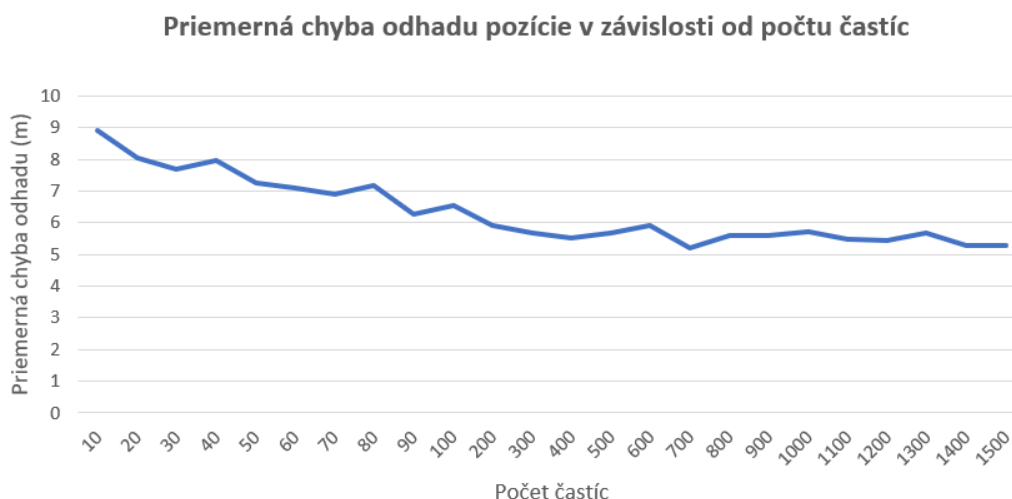
Obr. 4: Testovacia trasa dlhá približne 120 m.

4.3.1 Počet častíc

Z teoretického hľadiska platí, že čím väčší je počet častíc, tým časticový filter vierohodnejšie a presnejšie modeluje hustotu pravdepodobnosti aktuálnej pozície. Zvýšenie počtu častíc si však žiada svoju daň vo forme vyšších výpočtových nárokov zariadenia. Jeden z prístupov je teda použiť čo najväčší počet častíc s podmienkou, aby na danom zariadení algoritmus bežal dostatočne rýchlo, teda výpočet musí prebehnúť predtým, než je detegovaný ďalší krok.

S vyšším počtom častíc dokážeme súčasne modelovať viacej pozícií, kde sa používateľ práve môže nachádzať, a teda skôr by sme mali byť schopní používateľa správne lokalizovať a nestratiť jeho pozíciu. Viditeľný rozdiel v presnosti lokalizácie používateľa pri zvýšení počtu častíc je možné pozorovať najmä pri menšom počte častíc (do 100), ako to naznačuje graf na obr. 5.

Pri nepresnom odhade dĺžky kroku a smeru pohybu používateľa, väčší počet častíc prispieva k presnosti lokalizácie, avšak často sa vyskytuje úplná strata pozície používateľa.

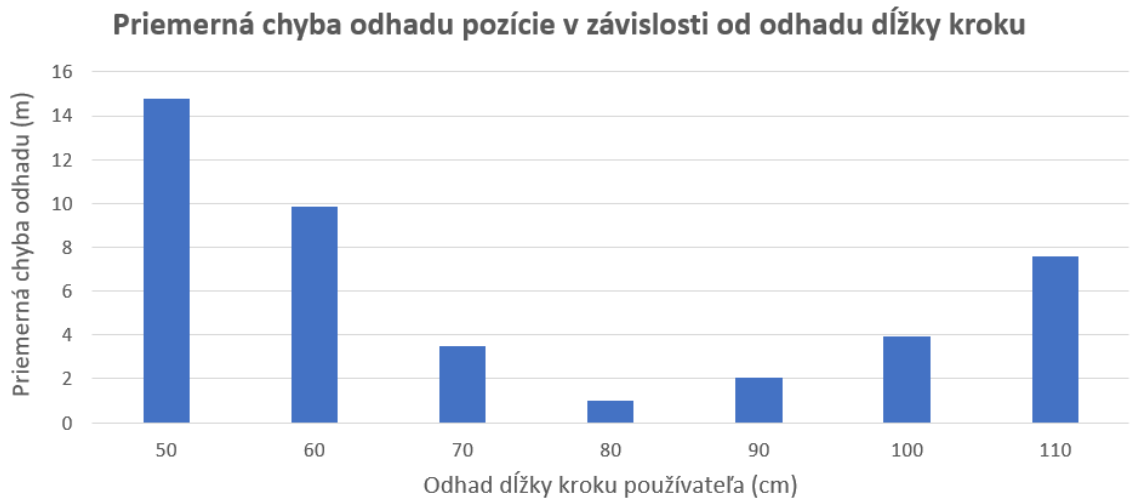


Obr. 5: Vplyv počtu častíc na presnosť lokalizácie. Chyba odhadu pozície používateľa je počítaná ako priemer všetkých chýb odhadov po každom detegovanom kroku.

Väčší počet častíc je však veľmi potrebný vtedy, ak nemáme žiadne informácie o počiatočnej polohe používateľa, a teda musíme rozložiť častice rovnomerne po celom priestore budovy. S dostatočným počtom častíc, ktorý závisí aj od veľkosti budovy a presnosti odhadu dĺžky kroku používateľa, algoritmus dokáže používateľa po niekoľkých krokoch úspešne lokalizovať, čomu napomáha štruktúra budovy s chodbami a stenami.

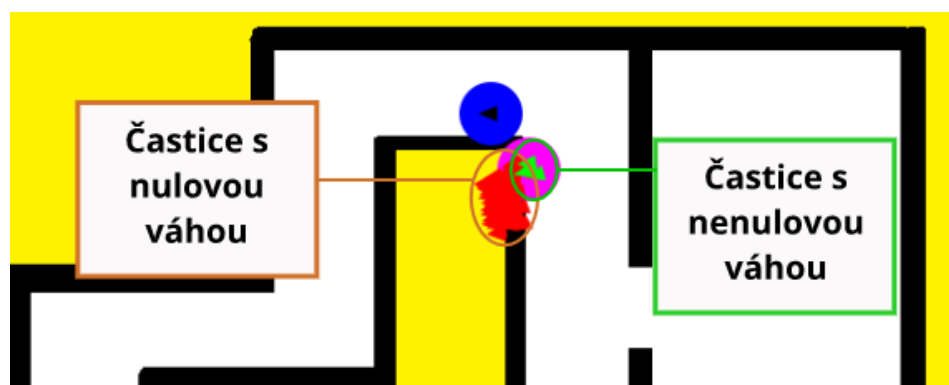
4.3.2 Dĺžka kroku a strata pozície používateľa

V prípade modelovania odhadu dĺžky kroku pomocou konštanty, jej hodnota má na presnosť lokalizácie používateľa výrazný vplyv, čo sa ukázalo aj na našich výsledkoch testovania (pozri obr. 6). Pri tejto evaluácii bola skutočná dĺžka kroku používateľa 80 cm a daný algoritmus bol spustený 360-krát pre každú dĺžku kroku z intervalu [50 cm, 110 cm], pričom v jednotlivých testoch sme menili počet častíc a hodnotu smerodajnej odchýlky šumu dĺžky kroku. Z grafu je ľahko vidieť, že čím je odhad dĺžky kroku menej presný, tým je chyba odhadu lokalizácie väčšia.



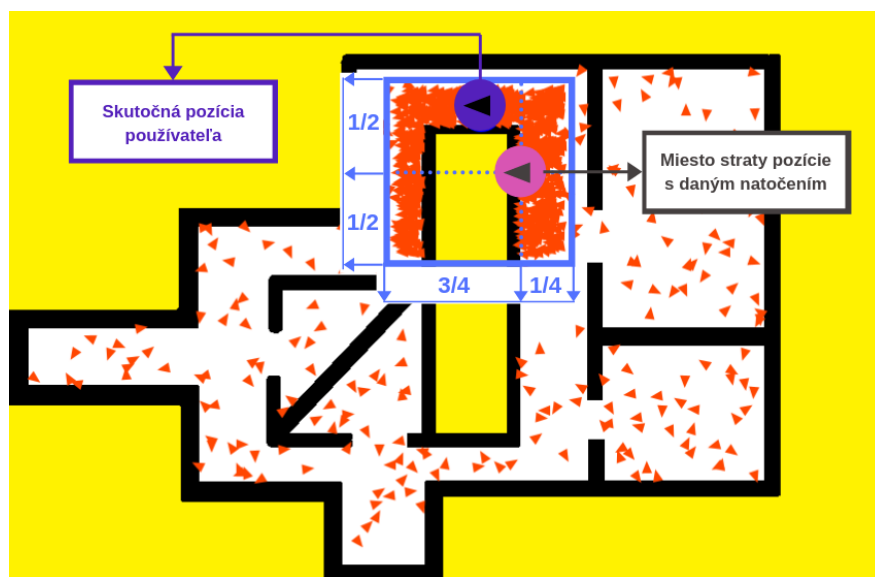
Obr. 6: Priemerná chyba odhadu lokalizácie používateľa pri rôznych hodnotách dĺžky kroku (skutočná dĺžka kroku používateľa bola 80 cm).

Jedným z problémov, ktorému časticový filter môže čeliť v prípade nepresného odhadu dĺžky kroku, je strata pozície používateľa. Napr. ak používateľ robí krok dĺžky 80 cm a náš odhad je 70 cm (teda o len 10 cm menší), a používateľ kráča chodbou dlhou 32 m, v skutočnosti sa dostane na koniec chodby po 40-tich krokoch, no s naším podhodnoteným odhadom dĺžky kroku budeme predpokladať, že používateľ zatiaľ prešiel pravdepodobne iba 28 m, teda chyba odhadu pozície v danom čase bude približne 4 m. Ak používateľ následne na konci chodby zabočí do niektorej strany a všetky častice sú rozptýlené v okolí menšom ako 4 m od zle odhadovanej pozície, po zopár krokoch v novom smere takmer všetky častice narazia do steny, čím sa im nastaví nulová hodnota váhy (pozri obr. 7). Ak neexistuje žiadna častica s nenulovou váhou, dôjde k strate pozície používateľa. Keďže potom pri prevzorkovaní algoritmus nemá z čoho vyberať, táto strata pozície je trvalá, a teda musí sa riešiť špeciálnym spôsobom.



Obr. 7: Detekcia straty pozície používateľa z dôvodu podhodnotenia dĺžky kroku.

Navrhli sme metódu na zotavenie sa systému po strate pozície, ktorá sa aktívuje, ak vyše 90% častíc po detegovanom kroku zanikne. Pri riešení tohto problému v našej implementácii detegujeme tento jav práve na základe nízkej hodnoty súčtu váh všetkých častíc po aktualizácii váh. V takomto prípade tri štvrtiny častíc rozmiestnime v okolí odhadovanej pozície takým spôsobom, že dané okolie je posunuté v odhadovanom smere, ako je znázornené na obr. 8. Zvyšným časticiam nastavíme náhodnú pozíciu v rámci celej budovy. Ukázalo sa, že dané opatrenie pomohlo algoritmu v takýchto situáciách opätovne modelovať pozíciu používateľa. Avšak, niekedy algoritmus súčasne modeloval aj ďalšie možné pozície, pričom pri ďalšej strate pozície sa už algoritmu častokrát nepodarilo opätovne zotaviť.



Obr. 8: Reakcia na detegovanú stratu pozície používateľa rozmiestnením troch štvrtín častíc v okolí odhadovanej pozície a jednej štvrtiny po celej budove.

Obmedziť následky nepresného odhadu dĺžky kroku a predísť strate pozície používateľa sa čiastočne dá zvýšením počtu častíc a tiež rozšírením normálneho rozdelenia šumu dĺžky kroku, teda zvýšením hodnoty smerodajnej odchýlky σ_{l_k} (pozri rovnice 9 a 10). S väčším počtom častíc je väčšia pravdepodobnosť, že nejaké častice sa budú pri každom detegovanom kroku posúvať o väčšiu (resp. menšiu) vzdialenosť, než je samotný odhad dĺžky kroku l_k . Teda vzorky $\delta_k^i \sim \mathcal{N}(0, \sigma_{l_k}^2)$ budú nadobúdať také hodnoty, že súčet $\delta_k^i + l_k$ sa bude približovať skutočnej dĺžke kroku používateľa. Zvýšením hodnoty smerodajnej odchýlky σ_{l_k} sa zasa zvýši pravdepodobnosť, že vzorky δ_k^i budú mať väčšie hodnoty. Potom komponent modelujúci šum bude mať v porovnaní so samotným odhadom dĺžky kroku väčšiu váhu, a teda výsledný posun častice bude vybraný z väčšieho intervalu. To v skratke znamená, že častice sa budú pohybovať viac

náhodne.

Pri testovaní vplyvu týchto parametrov na stratu pozície používateľa sme zaznamenávali počet testov, v ktorých časticový filter detegoval stratu pozície používateľa pri aktuálne nastavených parametroch algoritmu. Z údajov tabuliek 1 a 2 vyplýva, že zvýšením počtu častíc a zväčšením hodnoty smerodajnej odchýlky šumu dĺžky kroku sa početnosť strát pozícií znižuje. Takýto trend dát je aj v prípade iných odhadovaných dĺžok kroku. Pozn.: Údaj v tabuľkách, napr. 5/27, znamená, že v 5-tich prípadoch z 27-mich algoritmus detegoval stratu pozície používateľa.

Dĺžka kroku = 80 cm Odhad = 50 cm	Smerodajná odchýlka					
Počet častíc	10 cm	20 cm	30 cm	40 cm	50 cm	Celkový súčet
10 - 90	27/27	27/27	27/27	7/27	5/27	93/135
100 - 500	15/15	15/15	3/15	0/15	0/15	33/75
600 - 1000	15/15	15/15	0/15	0/15	0/15	30/75
1100 - 1500	15/15	15/15	0/15	0/15	0/15	30/75
Celkový súčet	72/72	72/72	30/72	7/72	5/72	186/360

Tabuľka 1: Počet strát pozícií používateľa pri rôznom ohodnotení parametrov počtu častíc a smerodajnej odchýlky šumu dĺžky kroku, pričom odhad dĺžky kroku bol podhodnotený o 30 cm.

Dĺžka kroku = 80 cm Odhad = 100 cm	Smerodajná odchýlka					
Počet častíc	10 cm	20 cm	30 cm	40 cm	50 cm	Celkový súčet
10 - 90	27/27	26/27	22/27	16/27	7/27	98/135
100 - 500	15/15	15/15	1/15	0/15	0/15	31/75
600 - 1000	15/15	11/15	0/15	0/15	0/15	26/75
1100 - 1500	15/15	9/15	0/15	0/15	0/15	24/75
Celkový súčet	72/72	61/72	23/72	16/72	7/72	179/360

Tabuľka 2: Počet strát pozícií používateľa pri rôznom ohodnotení parametrov počtu častíc a smerodajnej odchýlky šumu dĺžky kroku, pričom odhad dĺžky kroku bol nadhodnotený o 20 cm.

V tabuľke 3 môžeme vidieť, že v prípade presného odhadu dĺžky kroku sa strata pozície používateľa vyskytuje menej často a pri použití 100 alebo viacej častíc, k strate

pozície nedochádza vôbec. Dôkladnejšia analýza výsledkov testovania ukázala, že už pri počte častíc väčšom ako 50 nedochádza k strate pozície používateľa.

Dĺžka kroku = 80 cm Odhad = 80 cm	Smerodajná odchýlka					
Počet častíc	10 cm	20 cm	30 cm	40 cm	50 cm	Celkový súčet
10 - 90	4/27	5/27	4/27	3/27	4/27	21/135
100 - 500	0/15	0/15	0/15	0/15	0/15	0/75
600 - 1000	0/15	0/15	0/15	0/15	0/15	0/75
1100 - 1500	0/15	0/15	0/15	0/15	0/15	0/75
Celkový súčet	2/72	8/72	4/72	3/72	4/72	21/360

Tabuľka 3: Počet strát pozícií používateľa pri rôznom ohodnotení parametrov počtu častíc a smerodajnej odchýlky šumu dĺžky kroku, pričom odhad dĺžky kroku bol totožný so skutočnou dĺžkou.

4.3.3 Nastavenie váh časticiam

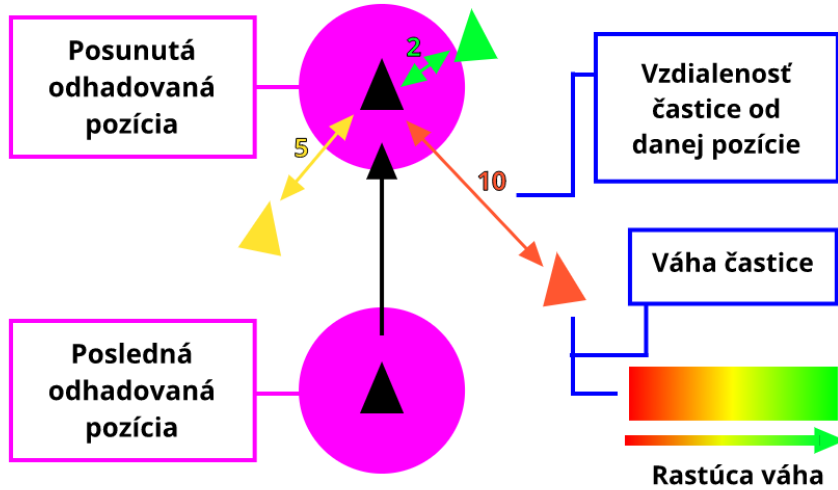
Vo všeobecnosti sa časticiam váhy nastavujú na základe dostupných meraní (intenzita Wi-Fi signálu, rozpoznávanie aktivity používateľa, atď.), pričom v našej implementácii využívame iba známu štruktúru budovy, a teda časticiam, nachádzajúcim sa na nedostupných miestach (steny, miesta mimo budovy), priradíme váhu s hodnotou nula.

V rámci tejto práce sme implementovali aj takú verziu časticového filtra, ktorá zvyšným časticiam, nachádzajúcim sa na dostupných miestach budovy, nastaví váhy na základe ich vzdialeností k poslednej predikovanej pozícii, po jej posunutí o odhadovanú dĺžku kroku používateľa vypočítaným smerom. Čím je častica od danej pozície viac vzdialená, tým menšiu váhu jej algoritmus priradí (pozri obr. 9).

Označme danú aktualizovanú odhadovanú pozíciu ako $\langle \hat{x}_k, \hat{y}_k \rangle$ a vzdialenosť i -tej častice od tejto pozície v čase k ako d_k^i . Daná vzdialenosť je potom vypočítaná podľa vzťahu:

$$d_k^i = \sqrt{(\hat{x}_k - x_k^i)^2 + (\hat{y}_k - y_k^i)^2} \quad (11)$$

Konkrétnu hodnotu váhy i -tej častice \hat{w}_k^i počítame na základe vzdialenosti d_k^i pomocou funkcie hustoty pravdepodobnosti normálneho rozdelenia $\mathcal{N}(0, \sigma_d^2)$ s nulovou strednou hodnotou, pričom v našom prípade je $\sigma_d = 3$ m.



Obr. 9: Nastavenie váh časticiam v závislosti od ich vzdialeností k poslednej predikovanej pozícii používateľa posunutej o odhadovanú dĺžku kroku daným smerom.

Z pozorovania sme zistili, že daná implementácia časticového filtra, podľa očakávania, dokáže používateľa presne lokalizovať len v prípade správneho odhadu dĺžky kroku a natočenia. Takýto spôsob nastavovania váh je teda použiteľný a vhodný v prípadoch, kedy máme záruku pomerne presných vstupných údajov o pohybe používateľa.

4.3.4 Natočenie ako súčasť stavu častíc

V tomto experimente sme našu implementáciu obmenili tak, že stav každej častice je reprezentovaný navyše aj jej natočením θ , teda $\mathbf{x}_k^i = \langle x_k^i, y_k^i, \theta_k^i \rangle$. Táto zmena nám ponúka nový spôsob posúvania častíc. Namiesto toho, aby sa všetky častice posúvali (takmer) tým istým smerom podľa merania kompasu, ako to robia v prípade dvojrozmerného stavu, častice si pamätajú svoje aktuálne natočenie. Pri detekcii kroku sa častice otáčajú o uhol, ktorý sa počíta ako rozdiel aktuálneho a predošlého odhadu smeru, a potom sa posunú vo svojom smere o odhadovanú dĺžku. Vďaka tomu, že častice si pamätajú svoje natočenie, táto implementácia môže byť odolnejšia voči anomáliam z kompasu. Tento spôsob posúvania častíc je popísaný nasledovnými rovnicami:

$$\theta_k^i = \theta_{k-1}^i + (\alpha_k - \alpha_{k-1}) + \beta_k^i, \quad (12)$$

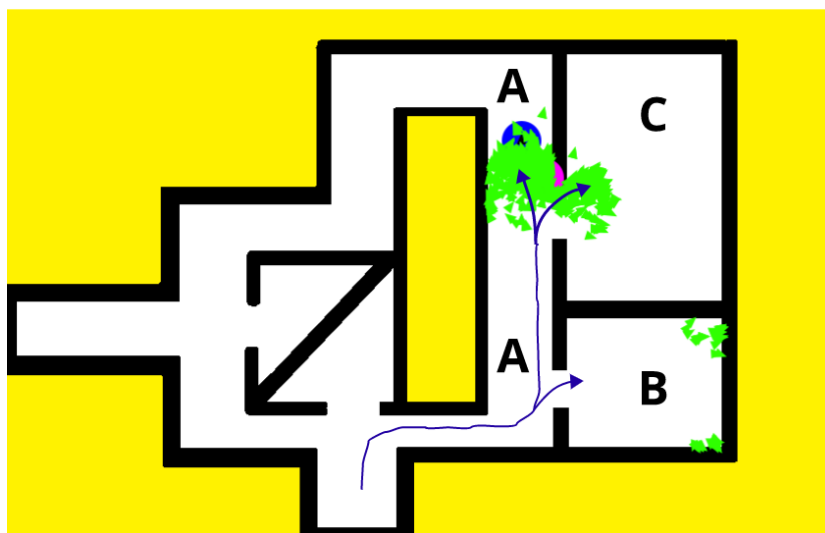
$$x_k^i = x_{k-1}^i + (l_k + \delta_k^i) \cos \theta_k^i, \quad (13)$$

$$y_k^i = y_{k-1}^i + (l_k + \delta_k^i) \sin \theta_k^i, \quad (14)$$

kde α_k, α_{k-1} sú údaje z kompasu (smer, ktorým sa používateľ pohol) v čase k a $k-1$, l_k je odhadovaná dĺžka kroku v čase k a vzorky $\beta_k^i \sim \mathcal{N}(0, \sigma_{\alpha_k}^2), \delta_k^i \sim \mathcal{N}(0, \sigma_{l_k}^2)$ modelujú šum natočenia a dĺžky kroku.

Pri pozorovaní tejto implementácie v našej aplikácii sme zistili, že častice sú viac rozptýlené, a teda pokrývajú na mape väčšiu plochu. Taktiež, ak používateľ prechádza chodbou, z ktorej sa dá odbočiť do nejakej miestnosti alebo inej chodby, istá časť častíc sa v tej chvíli odchýli z doterajšej trasy a snaží sa modelovať situáciu, že používateľ naozaj zmenil smer, čo je žiadúce správanie časticového filtra (pozri obr. 10). Opäť je ale potrebný o niečo vyšší počet častíc, aby algoritmus dokázal dostatočne dobre modelovať viacero možných trás.

Ak je smerodajná odchýlka σ_{α_k} rozdelenia, ktoré modeluje šum natočenia, príliš veľká (z pozorovania väčšia ako 10), rozdiel jednotlivých natočení častíc je veľmi veľký. Častice sa potom začnú točiť dokola a úplne sa rozchádzať na všetky strany. Algoritmus tak nedokáže používateľa správne lokalizovať.



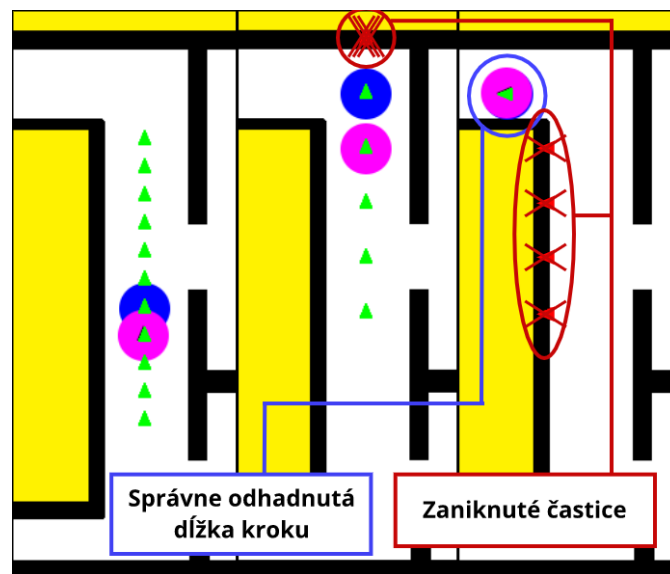
Obr. 10: Častice prechádzajúce chodbou A okolo vstupu do miestnosti C sa rozdelili, aby modelovali obe možné situácie - používateľ pokračoval v doterajšom smere po chodbe A, alebo zabočil do miestnosti C. (Častice v miestnosti B po pár krokoch zaniknú, čím sa zavrhnú možnosť, že používateľ zmenil smer a vošiel do miestnosti B.)

4.3.5 Dynamická dĺžka kroku

Nesprávny odhad dĺžky kroku pri fixovanej dĺžke kroku spôsoboval veľmi veľké problémy s presnosťou lokalizácie. V rámci tejto práce sme implementovali časticový filter, ktorého častice si okrem aktuálneho stavu pozostávajúceho z 2D pozície uchovávajú

aj svoju dĺžku kroku. Pri inicializácii sa každej častici vygeneruje náhodná dĺžka kroku v rámci nejakého určeného intervalu. Pri detegovaní kroku používateľa sa potom každá častica posunie odhadovaným smerom o hodnotu svojej dĺžky kroku.

Naša implementácia je založená na myšlienke, že ak by senzory neprodukovali zašumené dáta a používateľ by robil len kroky rovnakej, celočíselnej dĺžky, algoritmus by po niekoľkých iteráciách bol schopný zistiť skutočnú dĺžku kroku používateľa, keďže častice, ktorých dĺžka kroku nebola správna, by skôr či neskôr narazili do steny a pri kroku prevzorkovania by zanikli. Tak by nakoniec v celej množine ostala jedna multiplikovaná častica so správnou dĺžkou kroku (pozri obr. 11).



Obr. 11: Nájdenie správnej dĺžky kroku v laboratórnych podmienkach pomocou častíc, ktoré majú vlastnú dĺžku kroku.

Dĺžka kroku používateľa sa však môže meniť v čase, a preto pri prevzorkovaní sa každej vybratej častici pripočíta k dĺžke kroku malá náhodná hodnota $d_k^i \sim \mathcal{N}(0, \sigma_{d_k}^2)$ (v našej implementácii používame $\sigma_{d_k} = 3$ cm), pričom sa kontroluje, aby dĺžka kroku nejakej častice neprekročila určené hranice. Takýmto spôsobom väčšina častíc bude mať dĺžku kroku veľmi blízku skutočnej dĺžke. Niektoré častice, modelujúce situáciu zväčšenia dĺžky kroku používateľa, ju budú mať o niečo väčšiu a niektoré zas o niečo menšiu.

Ukázalo sa, že takáto implementácia časticového filtra dosahuje lepšie výsledky v lokalizácii, ako filter so statickým modelom dĺžky kroku. Pre uspokojivú presnosť a spoľahlivosť lokalizácie je však potrebný dostatok častíc. Podľa nášho pozorovania je vhodný počet okolo 500, alebo viac častíc.

Záver

Práca poskytuje stručný úvod do problematiky indoor lokalizácie s vysvetlením hlavnej myšlienky PDR a venuje sa Bayesovskému filtrovaníu a jeho implementáciám, špeciálne časticovému filtru.

V práci sme uvažovali, že máme prístup k dátam senzorov smartfónu, z ktorých vieme vyextrahovať dĺžku a smer kroku, a tiež sme predpokladali dostupnosť detailnej mapy budovy. Skúmali sme správanie časticového filtra v rôznych situáciách a analyzovali sme vplyv zmien hodnôt parametrov algoritmu na priebeh a presnosť lokalizácie pri konkrétnych podmienkach. Pre tento účel sme vytvorili aplikáciu na vizualizáciu lokalizácie používateľa pomocou časticového filtra, pričom sme využívali umelo vygenerované dáta a manuálne vytvorené mapy budov.

Zistili sme, že v prípade indoor lokalizácie založenej na PDR prístupe s dostupnou mapou budovy, ale bez využitia inej metódy, ako napr. Wi-Fi fingerprinting, je správny odhad dĺžky kroku kľúčový prvok. Ukázalo sa, že algoritmus so statickým modelom dĺžky kroku v prípade nepresného odhadu danej dĺžky nie je schopný používateľa správne lokalizovať - častokrát dochádza k strate pozícií. V rámci práce sme v našom algoritme implementovali detekciu a následnú reakciu na danú stratu pozície, ktorá sa vie vysporiadať s odchýlkami v odhade dĺžky kroku, ale v extrémnych prípadoch je neúčinná. Preto sme implementovali aj verziu časticového filtra s dynamickou dĺžkou kroku, kde si častice uchovávajú navyše vo stavovom vektore aj svoju dĺžku kroku. Daná implementácia sa ukázala byť pri dostatočnom počte častíc presná a spoľahlivá.

V budúcnosti by bolo vhodné urobiť podrobnejšiu analýzu jednotlivých parametrov algoritmu a rozličných implementácií časticového filtra na viacerých testovacích trasách v rôznych budovách. Pozrieť sa na priebeh algoritmu na reálnych dátach by mohlo byť tiež veľkým prínosom pre túto tému.

Zoznam použitej literatúry

- [1] ARULAMPALAM, M., MASKELL, S., GORDON, N., AND CLAPP, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50, 2 (2002), 174–188.
- [2] BURGARD, W., FOX, D., HENNIG, D., AND SCHMIDT, T. Position tracking with position probability grids. *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROBOT '96)* (1996), 2–9.
- [3] CHEN, Z., ET AL. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics* 182, 1 (2003), 1–69.
- [4] CHEN, Z., ZOU, H., JIANG, H., ZHU, Q., SOH, Y. C., AND XIE, L. Fusion of WiFi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors* 15, 1 (jan 2015), 715–732.
- [5] CHO, S. Y., AND PARK, C. G. Mems based pedestrian navigation system. *The Journal of Navigation* 59, 1 (2006), 135–153.
- [6] FETZER, T., EBNER, F., BULLMANN, M., DEINZER, F., AND GRZEGORZEK, M. Smartphone-based indoor localization within a 13th century historic building. *Sensors* 18, 12 (2018), 4095.
- [7] FOX, D., HIGHTOWER, J., LIAO, L., SCHULZ, D., AND BORRIELLO, G. Bayesian filtering for location estimation. *IEEE pervasive computing*, 3 (2003), 24–33.
- [8] GALČÍK, F., AND OPIELA, M. Grid-based indoor localization using smartphones. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (2016), IEEE, pp. 1–8.
- [9] HAFNER, P., MODER, T., WIESER, M., AND BERNOULLI, T. Evaluation of smartphone-based indoor positioning using different bayes filters. In *2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013* (oct 2013), IEEE, pp. 1–10.

- [10] KANG, W., AND HAN, Y. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal* 15, 5 (2015), 2906–2916.
- [11] KANG, W., NAM, S., HAN, Y., AND LEE, S. Improved heading estimation for smartphone-based indoor positioning systems. In *PIMRC (2012)*, pp. 2449–2453.
- [12] KOSECKA, J., ZHOU, L., BARBER, P., AND DURIC, Z. Qualitative image based localization in indoors environments. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* (2003), vol. 2, IEEE, pp. II–II.
- [13] LEDLIE, J., PARK, J.-G., CURTIS, D., CAVALCANTE, A., CAMARA, L., COSTA, A., AND VIEIRA, R. Molé: a scalable, user-generated wifi positioning engine. *Journal of Location Based Services* 6, 2 (2012), 55–80.
- [14] LI, F., ZHAO, C., DING, G., GONG, J., LIU, C., AND ZHAO, F. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing (2012)*, ACM, pp. 421–430.
- [15] LIU, J., CHEN, R., PEI, L., GUINNESS, R., AND KUUSNIEMI, H. A hybrid smartphone indoor positioning solution for mobile lbs. *Sensors* 12, 12 (2012), 17208–17233.
- [16] RADU, V., AND MARINA, M. K. Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting. In *International conference on indoor positioning and indoor navigation (2013)*, IEEE, pp. 1–10.
- [17] RAI, A., CHINTALAPUDI, K. K., PADMANABHAN, V. N., AND SEN, R. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking (2012)*, ACM, pp. 293–304.
- [18] RENAUDIN, V., DEMEULE, V., AND ORTIZ, M. Adaptive pedestrian displacement estimation with a smartphone. In *2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013 (oct 2013)*, IEEE, pp. 1–9.
- [19] RENAUDIN, V., SUSI, M., AND LACHAPPELLE, G. Step length estimation using handheld inertial sensors. *Sensors* 12, 7 (2012), 8507–8525.
- [20] YANG, J., WANG, Z., AND ZHANG, X. An ibeacon-based indoor positioning systems for hospitals. *International Journal of Smart Home* 9, 7 (2015), 161–168.

Prílohy

Ako prílohu k tejto práci odovzdávame CD, na ktorom sa nachádzajú nasledovné položky:

- Bakalárska práca vo formáte pdf
- ParticleFilterSimulator - aplikácia na simulovanie a vizualizáciu lokalizácie používateľa v budove pomocou časticového filtra
- Návod k spusteniu aplikácie
- Výsledky testovania vo formáte xlsx