

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# WiFi lokalizace pro Android

BAKALÁŘSKÁ PRÁCE

**Michael Rais**

Brno, jaro 2014

## Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Michael Rais

**Vedoucí práce:** RNDr. Bc. Jonáš Ševčík

## Poděkování

Rád bych poděkoval vedoucímu své práce Jonáši Ševčíkovi za jeho pomoc při řešení této práce. Dále bych chtěl poděkovat rodině a přátelům za jejich podporu při studiu a v životě.

## Shrnutí

Cílem této bakalářské práce je prozkoumat metody lokalizace mobilního zařízení. Dále se zaměřit na systémy využívající WiFi, vybrat nejvhodnější a implementovat je na zařízení s operačním systémem Android. Práce pokrývá hlubší pohled na existující WiFi lokalizační systémy, porovnává je a na závěr vybírá nejvhodnější pro mobilní zařízení. Implementační část vyúsťuje v testování, kde jsou shrnuta výsledná data a stanoven závěr v porovnání s implementovaným systémem.

## **Klíčová slova**

WiFi, mobilní zařízení, Android, interiér, lokalizace, navigace, kompas

# Obsah

1	<b>Úvod</b>	1
2	<b>Metody lokalizace</b>	2
2.1	<i>GNSS</i>	2
2.2	<i>INS</i>	3
3	<b>WiFi lokalizace</b>	4
3.1	<i>Radar</i>	7
3.2	<i>Horus</i>	10
3.3	<i>Compass</i>	12
3.4	<i>Srovnání systémů</i>	14
4	<b>Implementace</b>	16
4.1	<i>ScanningEngine (Offline fáze)</i>	17
4.2	<i>SearchingEngine (Online fáze)</i>	21
5	<b>Testování</b>	23
5.1	<i>Průběh testování</i>	23
5.2	<i>Výsledky testování</i>	28
6	<b>Závěr</b>	29

## Kapitola 1

### Úvod

Již od pradávna pocituje lidstvo potřebu nalezení cesty z bodu A do bodu B. Prvními milníky lidstva bylo plné objevení celé planety a zavedení obchodních cest do map. Ještě před pár desítkami let jsme se řídili pomocí map, kompasu, buzoly či slunce. K tomu bylo třeba mít jistou dávku smyslu pro orientaci. S příchodem informačních technologií tuto úlohu částečně přebraly počítače, kdy se spuštěním internetu v 70. letech minulého století byly mapy dostupné prakticky kdykoliv a kdekoliv. Dalším důležitým milníkem byl rok 1973, kdy byl započat vývoj systému GPS (Global Positioning System). Od jeho uvedení do provozu (80. léta) [13] navigaci plně převzaly počítače, a to od distribuce (aktualizace) map po samotné vypočítávání trasy.

Roku 2007 byly představeny první tzv. chytré telefony (smartphones). Jejich příchod znamenal velký posun ve světě mobilních zařízení. Do této doby bylo nemyslitelné, aby lidé mohli být 24 hodin denně připojeni na internet, využívat jeho možnosti a protokoly (např. SMTP, FTP a další) a přitom být mobilní. Díky výbavě, kterou dostaly chytré telefony (GPS modul, senzory pro inerciální navigaci), je můžeme používat i k navigaci.

S dnešním rozrůstajícím se světem roste potřeba navigace pro interiéry (např. pro nalezení cesty na úřadech, kampusech, nemocnicích atd.). Nicméně systém GPS lze efektivně využívat pouze v exteriérech, jak bude popsáno v kapitole 2.1. Proto byly navrženy systémy jiné, využívající signálů (WiFi, Bluetooth atd.), nebo využívající samotného telefonu a jeho senzorů tj. inerciální navigace. Zde se budeme zabývat využitím WiFi pro navigaci.

Cílem práce je využít potenciálu mobilních telefonů a jejich senzorů pro lokalizační systém, vybrat nejvhodnější a implementovat jej. Tento vybraný systém následně testovat a porovnat výsledky měření. Tato práce je rozdělena do 4 kapitol. V první kapitole se krátce podíváme na systémy mimo WiFi. V druhé kapitole pak rozebereme existující WiFi systémy, porovnáme a vybereme nejvhodnější pro implementaci. Třetí kapitola je věnována vlastní implementaci vybraného systému. V poslední kapitole se podíváme na výsledky implementace, resp. přesnost měření.

## Kapitola 2

### Metody lokalizace

V této kapitole si krátce představíme některé metody lokalizace mimo WiFi a srovnáme jejich použitelnost se systémy využívajícími právě WiFi. V současné době je nejpoužívanějším systémem pro navigaci systém GPS, řadící se do GNSS.

#### 2.1 GNSS

GNSS (Global Navigation Satellite Systems) jsou systémy, mezi které se řadí už zmiňovaný GPS spravovaný vládou USA, ruský systém GLONASS nebo evropský projekt GALILEO. GNSS využívají ke své činnosti specifický hardware - satelity, které krouží po oběžné dráze, a díky nim jsou tyto systémy schopny lokalizace.

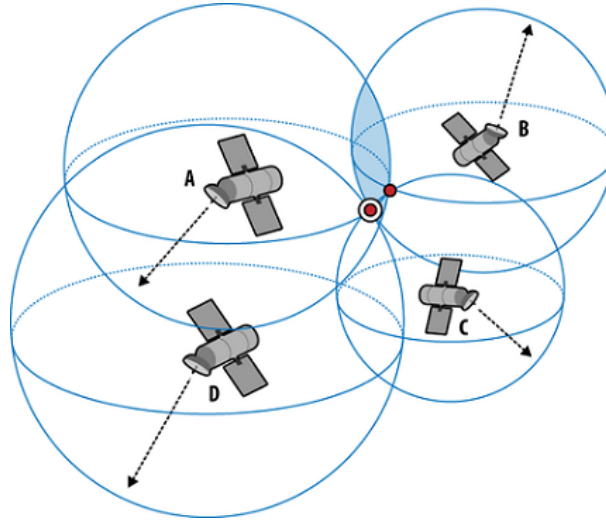
V současné době obíhá planetu Zemi minimálně dvacet čtyři funkčních satelitů systému GPS [13]. Zprávy, které satelity vysílají, se dělí na kategorie L1,...,L5, podle frekvence a určení[15].

Kategorie	Frekvence(MHz)	Použití
L1	1575.42	Civilní a vojenské
L2	1227.60	Civilní a vojenské
L3	1381.05	Sledování balistických střel
L4	1379.05	Ionosferická korekce
L5	1176.45	Signály SoS (Safety-of-Life signal)

Tabulka 2.1: Kategorie GPS signálů

Každý satelit periodicky vysílá zprávu, tzv. NAV Message, která se skládá ze tří částí. První část obsahuje datum, čas a status satelitu, druhá část orbitální data, ze kterých je schopen příjemce zprávy získat polohu satelitu, a poslední část obsahuje identifikátory všech satelitů, tzv. PRN čísla[13]. Pro výpočet fyzických souřadnic entity je potřeba alespoň čtyř jedinečných signálů pro určení délky, šířky a výšky, kdy se porovnávají časy odeslání a doručení. Trilaterací pak zjistíme naši pozici, která je průsečíkem funkcí signálu vysílaných družicemi [15], viz obrázek 2.1.





Obrázek 2.1: Určení polohy trilaterací 4 satelitů [10]

K dosažení co nejrelevantnějších výsledků je žádoucí, aby  $T = t_{receive} - t_{send}$ , kdy  $T$  je rozdíl času doručení NAV Message a času odeslání NAV Message, byl co nejmenší. Ideální případ je zachytávat přímý signál ze všech (minimálně tří) družic. Na volném prostranství, kde není prostor pro odraz signálu, se tohoto minimálního času dá spolehlivě dosáhnout. V tomto případě je chybová vzdálenost rovna řádově na metry [15]. S přibývajícím množstvím odrazů se tento čas zvětšuje, což zvětšuje i chybovou vzdálenost. V interiérech je použití GPS zcela nemyslitelné. Velký počet odrazů i blokování signálu střechami či zdmi budov buď zcela znemožňuje použití GPS, nebo je chybová vzdálenost tak velká, že nejsme schopni určit přesnou polohu.

## 2.2 INS

INS (Inertial Navigation Systems) jsou systémy, kde je veškerý potřebný hardware obsažen v tzv. IMU (Inertial Measurement Unit). Jeho součástí jsou senzory, 3-ortogonální gyroskop a 3-ortogonální akcelerometr, poskytující úhlovou rychlost, respektive lineární zrychlení [14]. Na základě znalosti těchto dat a počáteční pozice entity jsou tyto systémy schopny určovat naši polohu. Dříve byla inerciální navigace vzhledem k rozměrům gyroskopu a akcelerometru využívána převážně v letadlech, ponorkách, lodích apod. [14]. S příchodem MEMS (Micro-Electrical Mechanical Systems) se tyto senzory dostaly i do menších zařízení, jako jsou právě mobilní telefony, tablety atd., a díky nim jsou MZ schopny inerciální navigace.

Inerciální navigaci lze používat spolu s dalším lokalizačním systémem. Tímto spojením zvýšíme celkovou přesnost lokalizace.

## Kapitola 3

### WiFi lokalizace

V této kapitole se budeme již věnovat WiFi-based lokalizačním systémům. Systémy, kterými se budeme zabývat detailněji, potřebují ke své činnosti již vybudovanou infrastrukturu sestavenou z jednotlivých přístupových bodů (access point, AP). Tato nutná podmínka je v dnešní době splněna v téměř všech objektech, ve kterých má smysl využívat navigaci (zmiňované úřady, nemocnice atd.). Důvod, proč má smysl tyto systémy implementovat, je popsán v kapitole o GPS, kdy tuto navigaci nejsme schopni efektivně využívat v interiérech. Do budoucna je možné upravit GNSS systémy tak, aby jejich signály překonaly současné překážky, ale v dnešní době nelze tyto systémy v interiérech použít.

Nyní se budeme věnovat samotnému WLAN a WiFi principu, poté si představíme Ad-hoc topologii a nakonec se zaměříme na lokalizace nad infrastrukturovanými sítěmi.

WiFi je označení standardů postavených nad IEEE 802.11, popisující bezdrátovou komunikaci v počítačové síti. Standard 802.11 má čtyři rozšíření, která se liší parametry sítě, jako je frekvence, přenosová rychlost a vlnová délka, jež hraje roli při odchylkách WLAN kanálů, které budou popsány dále [19][1].

802.11 Protokol	Frekvence (GHz)	Vlnová délka (mm)
a	5	59
b	2.4	125
g	2.4	125
n	2.4/5	125

Tabulka 3.1: Rodina standardů 802.11

Variace 802.11b se časem stala primárním standardem pro WLAN (Wireless Local Area Network) namísto původního 802.11. Rozšíření 802.11b používá k přenosu dat rádiové vlny (RF). RF je část spektra elektromagnetického záření. K modulaci a šíření RF používá mechanismus DSSS (Direct Sequence Spread Spectrum), kdy nosný signál pokrývá plnou šířku pásma (spektra) přenosové frekvence vysílače[19].

Název Wi-Fi je ochrannou značkou organizace **Wireless Ethernet Compability Alliance, WECA**, která mimo jiné certifikuje zařízení splňující rodinu standardů 802.11. Těm zařízením splňujícím 802.11x, kde  $x \in \{a, b, g, n\}$ , uděluje WECA příznak Wi-Fi.

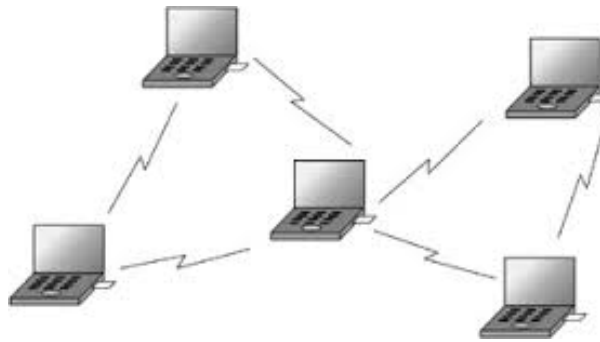
Podle toho, jak tyto zařízení zapojíme do WiFi sítě a jak celou síť nastavíme, rozlišujeme dvě hlavní topologie. Liší se implementovanými protokoly, způsobem komunikace atd. Vzhledem ke značným rozdílům jsou systémy navigací pro jednotlivé topologie principiálně velmi rozdílné.

- Ad-hoc síť
- Client/Server (infrastrukturované) síť

### Ad-hoc síť

#### Topologie

Ad-hoc sítě jsou sítě, jejichž základním kamenem je radio client, uzel (node). Uzly spolu komunikují přímo, není potřeba žádný AP, žádný centralizovaný server[8].



Obrázek 3.1: Topologie Ad-hoc sítě [11]

#### Navigace

Navigační systémy pro ad-hoc sítě pracují např. na principu využití empirické propagace rádiových modelů k triangulaci lokalizace MZ, jak je popsáno v [5].

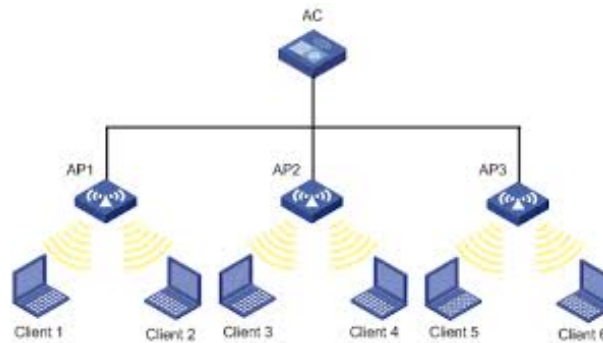
### Infrastrukturované sítě

#### Topologie

Základem této sítě je soubor APs, přes které komunikujeme se sítí. Všechny APs musí mít stejné SSID (identifikátor) sítě. V síti je hlavní (jeden, či více) směrovač (router), který zajišťuje směrování a tok dat po své LAN (Local Area Network). Jednotlivé APs pak slouží jako přístupové body k síti.

#### Navigace

Systémy navigace v infrastrukturovaných sítích se dělí do tří kategorií podle způsobu využití signálu vysílaného jednotlivými APs:



Obrázek 3.2: Topologie infrastrukturované sítě [12]

- Cell of Origin - Deandalus
- Signal Strength - Radar, Horus, Compass
- Time of Arrival - LI00

V této práci se budeme zabývat systémy využívající sílu signálu (SS). Než se dostaneme k samotným systémům, zaměříme se nejprve na specifické vlastnosti WLAN kanálu, způsobující odchylky RF signálu..

## Vlastnosti WLAN kanálu

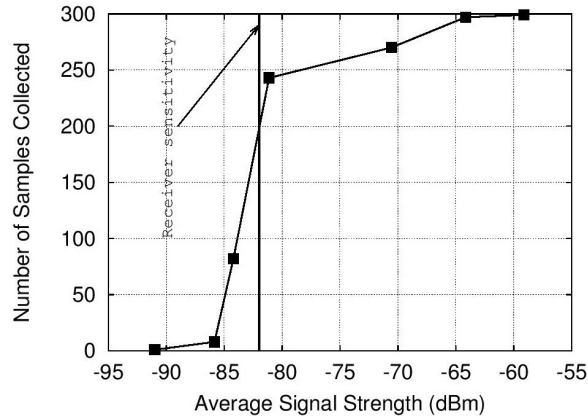
Propagace RF signálu prostorem způsobuje dva typy odchylek:

- dočasné odchylky měření signálu (odchylky při statickém měření SS)
- prostorové odchylky měření signálu (odchylky při měření SS v pohybu)

### Dočasné odchylky

Tyto odchylky se projevují během času, pokud je MZ zafixované na jednom místě. Při snímání síly signálu z jednoho AP se histogramy se zaznamenanými daty během času lišily o hodnotu 10 dBm a víc. Je žádoucí, aby systémy využívaly k určení pozice více APs než jen jeden. Tuto hodnotu SS je však možno korelovat, nicméně vysoké korelace mohou celý systém s přibývajícím množstvím APs degradovat[1].

Odchylka při detekování síly signálu z více APs je znázorněna v grafu 3.3, kde je vidět monotónně rostoucí funkci vyjadřující počet zachycených signálů vzhledem k jejich průměrné SS. Vzhledem ke konstantnímu šumu kanálu platí, že čím větší SS, tím větší poměr signál/šum. Z toho vyplývá, že pravděpodobnost zachycení signálu síťovou kartou je větší [1].



Obrázek 3.3: Vztah mezi průměrnou hodnotou SS a procentem zachycených signálů [1]

### Prostorové odchylky

Tyto odchylky se projevují, pokud je MZ v pohybu. Při pohybu na delší vzdálenost vzniká odchylka díky útlumu RF signálu, na druhou stranu při pohybu na malé vzdálenosti vznikají odchylky díky fyzikálním vlastnostem RF signálu. Pro 802.11b síť pracující na frekvenci 2,4 GHz je vlnová délka 12,5 cm, kdy při měření průměrných SS můžeme dostávat odchylku až do 10 dBm [1], ve vzdálenosti je to pak 7,6 cm.

## 3.1 Radar

### Úvod

Systém Radar, je prvním ze systémů využívajících bezdrátovou WiFi síť, který bude v této práci představen. Radar je postaven na RF bezdrátových LAN sítích nad protokolem IEEE 802.11 [2].

### Princip lokalizace

Základní myšlenkou systému Radar je detekování síly signálu (SS) jednotlivých APs. Tato hodnota pak vyjadřuje vzdálenost příjemce signálu vzhledem k AP. Jinými slovy, tato hodnota je funkcí relativní polohy příjemce signálu. Tato hodnota přirozeně roste s bližší vzdáleností příjemce vůči AP a naopak. Hodnoty SS jednotlivých s jsou na sobě nezávislé. Tuto znalost využívá Radar pro lokalizaci MZ.

V první fázi je potřeba vytvořit mapu otisků (radio map, RM) jednotlivých APs. RM je databáze otisků snímaných na různých místech budovy, resp. místnosti. Záznam v RM je k-tice  $(x, y, z, ss_{i(i=1\dots n)})$ , kde  $(x, y, z)$  jsou fyzické souřadnice místa snímání otisku a  $ss_{i(i=1\dots n)}$  je síla signálu vysílaná  $i^{th}$  AP. Samotné vytváření RM spočívá ve snímání otisků na vybraných bodech (reference point, RP) rozprostřených po budově, reps. místnosti, které



po místnosti o např. 5 metrů za sekundu, ani nepřeskakuje jednotlivé RPs. Systém Radar proto implementuje algoritmus založený na Viterbiho algoritmu.

### Viterbi algoritmus (VA)

VA je algoritmus dynamického programování, který hledá nejpravděpodobnější posloupnosti skrytých stavů, tzv. Viterbiho cestu. Tento algoritmus představil Andrew Viterbi roku 1967 pro dekódování konvolučních kódů na digitálních komunikačních linkách se šumem[9]. Algoritmus pracuje se skrytými Markovovými procesy[7], kde je stavový prostor  $S$ , počáteční pravděpodobnosti  $\pi_i$  začátku ve stavu  $i$  a přechodové pravděpodobnosti  $a_{i,j}$  pro přechod ze stavu  $i$  do stavu  $j$ . Pro danou výstupní posloupnost  $y_1, \dots, y_T$  je nejpravděpodobnější stavová sekvence  $y_1, \dots, y_T$  splňující rekuretní vztahy[20]:

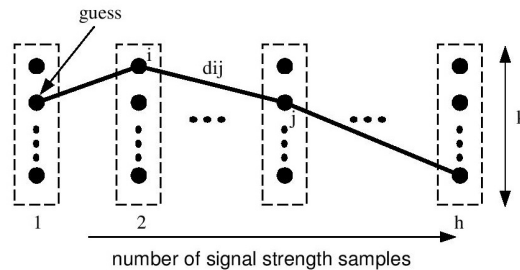
$$V_{1,k} = P(y_1|k) \cdot \pi_k$$

$$V(t, k) = P(y_t|k) \cdot \max_{x \in S} (a_{x,k} \cdot V_{t-1,x})$$

Kde  $V_{t,k}$  je pravděpodobnost nejpravděpodobnější sekvence stavů  $t \rightarrow k$ ,  $k$  je koncový stav.

### Viterbi-like algoritmus

Systém Radar implementuje obměnu Viterbiho algoritmu. Při každém získání SS n-tice se provede NNSS vyhledávání  $k$  nejbližších sousedů ( $k$ -NNSS), tj.  $k$  nejlepších shod pro výběr RP jako místa, kde se nachází MZ. V současné době se udržuje historie o hloubce  $h$  takových  $k$ -NNSS vyhledávání. Kolekce takových  $h$   $k$ -NNSS množin je vidět na grafu, kde vrcholy reprezentují jednotlivé RPs a hrana, která je ohodnocena metrikou, tj. Eukleidovská vzdálenost mezi fyzickými souřadnicemi daných RPs, vyjadřuje trajektorii MZ. Čím větší je váha hrany, tím klesá pravděpodobnost. Každým aktualizováním této historie  $k$ -NNSS (kde se uloží nejnovější záznam a nejstarší se vymaže) se spočítá nejkratší cesta mezi nejstarším a nejnovějším záznamem. Tato nejkratší cesta je reprezentována jako nejpravděpodobnější trajektorie MZ[2].



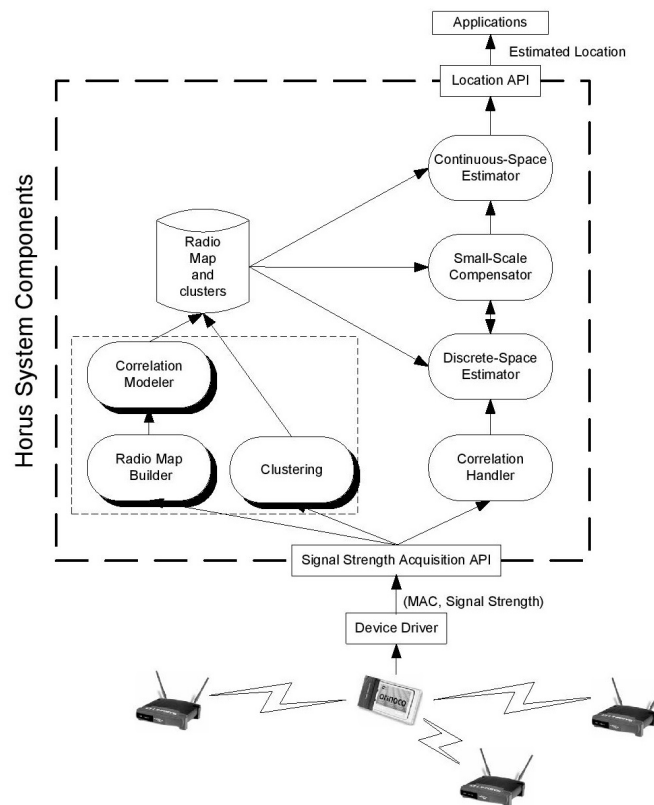
Obrázek 3.5: Ukázka stavu Viterbi-like algoritmu [2]

### 3.2 Horus

Horus je systém založený na RF, implementovaný na 802.11 bezdrátových LAN (WiFi). Využívá síly signálů vysílaných APs pro určení polohy MZ. Tak jako Radar, tak i systém Horus pracuje ve dvou fázích:

- Offline fáze, sestává se ze snímání SS APs v daných RPs a budování RM
- Online fáze, spočívá v samotném vypočítání polohy MZ na základě získaných SS

Zatímco Radar používá deterministické techniky pro určení polohy MZ (algoritmem NNSS), Horus pracuje s pravděpodobnostmi, kdy ukládá distribuční funkce signálu do RM a pak využívá pravděpodobnostních technik pro určení polohy MZ. Nyní si přiblížíme jednotlivé prvky systému Horus viz 3.6.



Obrázek 3.6: Části systému Horus [1]

#### Matematický model

Horus stanovuje dvou-rozměrný fyzický prostor  $\mathbb{X}$ , kde v každé pozici  $x \in \mathbb{X}$  můžeme snímat síly signálu  $k$  APs. Takový  $k$ -rozměrný prostor sil signálů značíme jako  $\mathcal{S}$ . Každý



prvek tohoto prostoru  $s \in \mathbb{S}$  je pak  $k$ -rozměrný vektor, jehož záznamy reprezentují síly signálu jednotlivých APs. Po získání vektorů  $s = (s_1, \dots, s_k)$  je cílem systému Horus najít takovou pozici  $x \in \mathbb{X}$ , která má největší pravděpodobnost  $P(x/s)$ , tzn.

$$\operatorname{argmax}_x [P(x/s)].$$

Díky Bayesově teorému můžeme vztah vyjádřit jako:

$$\operatorname{argmax}_x [P(x/s)] = \operatorname{argmax}_x [P(s/x)]$$

$P(x/s)$  je pak vyčíslitelná s využitím RM jako:

$$P(x/s) = \prod_{i=1}^k P(s_i/x).$$

Tímto postupem získáme jedinečný záznam z RM, pro zvýšení přesnosti využívá Horus dvě techniky pro určení pozice v kontinuálním prostoru.

První technika se nazývá **Center of Mass of the Top Candidate Locations**. Každá pozice v RM je reprezentována jako objekt ve fyzickém prostoru, jehož váha je rovna přiřazené pravděpodobnosti získané rozhodovacím algoritmem. Poté jsme schopni získat těžiště takovýchto  $N$  objektů s největší vahou, kde  $N$  je systémový parametr  $1 \leq N \leq \|X\|$ . Formálně, necht  $p(x)$  je pravděpodobnost pozice  $x \in \mathbb{X}$  a  $\bar{X}$  je seznam pozic z RM v seřazeném pořadí podle pravděpodobnosti. Výše uvedená technika pak určí aktuální pozici  $x$  jako:

$$x = \frac{\sum_{i=1}^N p(i) \bar{X}}{\sum_{i=1}^N p(i)}$$

Druhá technika **Time-Averaging in the Physical Space (TAPS)** je založena na použití průměrného času k doladění výsledných odhadů pozice. Tato technika získává výslednou pozici na základě zprůměrování posledních  $W$  odhadů z minulosti. Formálně, z minulosti získaných odhadů pozice  $x_1, x_2, \dots, x_t$ , TAPS určuje aktuální pozici  $\bar{x}_t$  v čase  $t$  jako:

$$\bar{x}_t = \frac{1}{\min(W, t)} \sum_{i=t-\min(W, t)+1}^t x_i$$

## Clustering

Systém Horus si klade dva cíle: lokalizovat MZ a snížit výpočetní nároky pro určení polohy. Dosažení prvního cíle je popsáno výše. Pro snížení náročnosti rozhodovacího algoritmu používá systém Horus *Incremental Triangulation clustering technique* (IT).

Horus definuje tzv. *clusters* (shluky, klastry), kdy *cluster* je množina pozic, ve kterých se vyskytují známé APs. Množina známých APs je pak *cluster key*. Pro dané  $x$  chceme určit, do kterého shluku patří. Myšlenka IT spočívá v tom, že každý AP definuje podmnožinu z RM, která je pokryta daným AP. Tyto pozice z RM jsou shlukem pozic, jehož klíč (key) je AP pokrývající pozice v tomto shluku. Při určování polohy MZ pak postupujeme od jednoho AP k dalšímu. Tzn. první AP vybere podmnožinu záznamů RM obsahující ty záznamy, jejichž pozice jsou pokryty právě prvním AP. Při vybírání shody druhého AP pak bereme jako vstupní RM vybranou podmnožinu získanou v předchozím kroku. Tento druhý AP opět vybere podmnožinu krytou druhým AP atd. Pozice  $x$  pak patří do shluku s klíčem AP  $a$  pouze tehdy, jestliže v pozici  $x$  detekujeme signál z AP  $a$ .

IT dále pracuje následovně. Na vstupu je sekvence APs s jejich SS uspořádána sestupně. IT následně vezme jednotlivé seřazené APs a postupně vybírá podmnožinu pozic v RM. Pokud je pravděpodobnost daného AP, např.  $a_1$ , prokazatelně větší (na základě odchylky) než pravděpodobnost následujícího AP  $a_2$ , vrací IT jako výsledek  $a_1$ . V opačném případě pokračuje IT ve vybírání podmnožiny na základě  $a_2, a_3$  atd. Tímto postupným vybíráním podmnožiny Horus pracuje rychleji, než kdyby se při každém dotazu na shodu AP musela procházet celá RM[1].

### Prostorové odchylky

Jelikož RM obsahuje RPs, které jsou typicky ve vzdálenosti  $\geq$  metr od sebe, nepokrývá RM odchylky na malém prostoru. Proto systém Horus využívá *Perturbation technique*, která je založena na rozpoznání těchto odchylek a jejich kompenzování.

Pro detekci využívá heuristik, člověk s MZ se pohybuje jen určitou rychlostí mezi RPs. Systém určí pozici s klasickou RM a rozhodovacím algoritmem. Poté se vypočítá vzdálenost mezi nalezenou pozicí a předchozí pozicí MZ. Pokud je tato vzdálenost větší než odchylka (daná v závislosti na rychlosti pohybu MZ mezi RPs), systém detekuje odchylku.

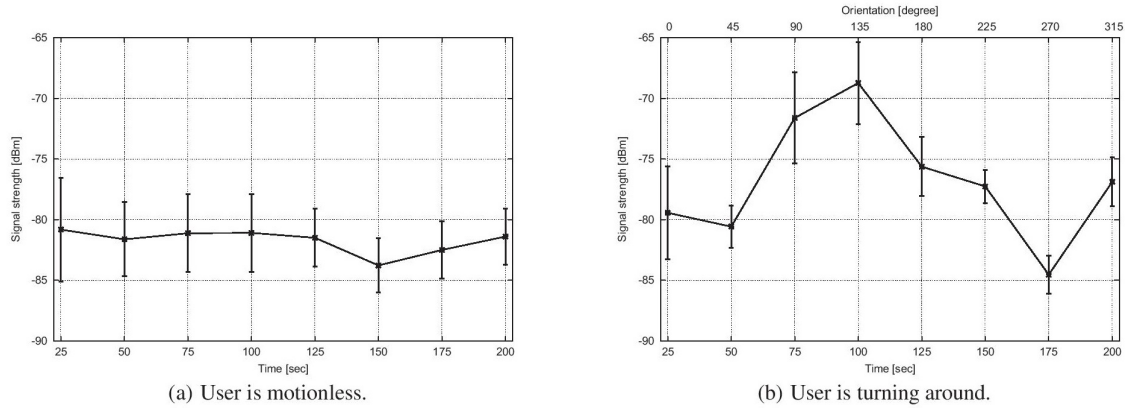
Jako kompenzaci takových odchylek systém rozebere přijaté SS vektory, znovu odhadne pozici a vybere tu, která je nejbližší k předchozí pozici MZ[1].

### 3.3 Compass

Systém Compass pracuje stejně jako jeho předchůdci ve dvou fázích. Offline fáze slouží pro sběr SS na daných RPs a budování RM a online fáze, ve které se hledá aktuální pozice MZ. Stejně jako Horus používá k určení pozice *probabilistic technique*. Liší se však v selektování podmnožiny z RM pro výběr nejvhodnějšího kandidáta[4].

Uživatel v typickém případě nese svoje MZ před sebou, aby viděl na displej, případně mohl s MZ interagovat. Lidské tělo je tvořeno z cca 70% vodou, čili s velké části blokuje 802.11 RF. Tzn. detekovaná SS z daného AP závisí na orientaci našeho těla vůči AP. Tato skutečnost pak degraduje systémy založené na snímání SS.

Na základě [4] byl proveden experiment o vlivu lidského těla na blokování 802.11 RF signálu. Výsledky jsou vidět na grafech 3.7.



Obrázek 3.7: Rozdíl v SS na základě orientace uživatele[4]

V tomto experimentu autoři [4] měřili SS, kdy záměrně blokovali RF signál z AP, čili postavení bylo AP  $\rightarrow$  Uživatel  $\rightarrow$  MZ. Průměrné výsledky jsou v levém grafu, kde se měřená průměrná síla signálu pohybuje stabilně mezi -80 a -83 dBm. V druhém kroku experimentu se již uživatel otáčel vždy o  $45^\circ$ , výsledky jsou v druhém grafu. Rozdíl je oproti statickému uživateli až o 15 dBm (orientace  $135^\circ$  při rozestavení AP  $\rightarrow$  MZ  $\rightarrow$  Uživatel). I v případě částečného blokování signálu jsou rozdíly stále větší než 5 dBm.

Na základě tohoto experimentu přidává systém Compass k základní myšlence další komponentu, a to kompas[4]. Dnešní MZ jsou již vybaveny potřebnými senzory, jako je gyroskop spolu s magnetometrem, čili jsou schopny určit orientaci uživatele.

## Matematický model

Mějme fyzický prostor pokryt mřížkou  $q$  referenčních bodů (RPs) vzdálených metr od sebe. Na každém RP provedeme sběr SS APs v 8 směrech. Formálně, necht  $S = \{s_1, \dots, s_n\}$  je stavový prostor  $q$  referenčních bodů  $R = \{r_1, \dots, r_q\}$  a orientací  $O = \{0^\circ, 45^\circ, \dots, 315^\circ\}$ . Každý stav  $s \in S$  je pak definován jako  $S = R \times O$ , tj. souřadnicemi kartézského prostoru a orientací.

Distribuce SS AP získaná měřením je agregací několika měření. Každé měření  $M = \{m_1, \dots, m_k\}$  obsahuje sílu  $k$  přítomných APs. Dále  $m_i = (p_i, id_i), i \in \{1, \dots, k\}$  je n-tice obsahující sílu AP  $p_i (p_i \in \{0, 1, \dots, 255\})$  a unikátní MAC adresu  $id_i$ .

Dále definujeme *podobnost stavů*. Dva stavy  $a$  a  $b$  reprezentované souřadnicemi a orientacemi  $o_a$  a  $o_b$  jsou *podobné* vzhledem k  $\alpha$  odchylce právě tehdy, když  $|o_a - o_b| \leq \alpha$ . Tato podobnost stavů se využívá v online fázi pro výběr podmnožiny z RM. Dle [4] je  $\alpha$  pro dosažení minimální chybové vzdálenosti rovna  $\alpha = 69^\circ$ .

### Online Fáze

V online fázi uživatel získá měření  $M$  SS okolních APs a orientaci  $O$ . Na základě dané odchylky  $\alpha$  získáme podmnožinu z RM. Tyto podobné stavy následně spojíme dohromady zkombinováním jejich histogramů, kdy pro každý referenční bod  $i$  a AP  $j$  těchto podobných stavů vypočítáme průměr  $\mu_{i,j}$  a standardní odchylku  $\sigma_{i,j}$ .

Pravděpodobnost získání množiny měření  $M$  v referenčním bodě  $i$  je vyjádřena podmíněnou pravděpodobností:

$$P(M|r_i) = \prod_{a=1}^k P(m_a|r_i)$$

kde

$$\begin{aligned} P(m_a|r_i) &= P((p_a, id_a)|r_i) \\ &= \frac{G_{i,id_a}(p_a) + \beta}{\sum_{m=0}^2 55G_{i,id_a}(m) + \beta} \end{aligned}$$

a

$$G_{i,id_a}(p_a) = \int_{p_a-1/2}^{p_a+1/2} \frac{e^{-(x-\mu_{i,id_a})^2/(2\sigma_{i,id_a}^2)}}{\sigma_{i,id_a}\sqrt{2\pi}} dx$$

$G_{i,id_a}(p_a)$  reprezentuje pravděpodobnost, že v referenčním bodě  $i$  naměříme sílu signálu  $p_a$  AP  $id_a$ .  $\beta$  pak reprezentuje hodnotu šumu, kterou přidáváme do výpočtu pravděpodobnosti, a která nám vyvažuje odchylky WLAN kanálu.

V posledním kroku se tyto hodnoty použijí k výpočtu  $\pi'_i$  vektoru:

$$\pi'_i = \frac{\pi_i * P(M|r_i)}{\sum_j = 1^n \pi_j * P(M|r_j)}$$

pro každý RP  $i$ . Na začátku algoritmu je  $\pi_i = 1/q$ , kde  $q$  je počet RPs[4].

### 3.4 Srovnání systémů

Všechny systémy pracují ve dvou fázích. V první fázi, offline, pokrývají daný prostor sítí referenčních bodů, ve kterých následně provedou měření okolních APs. Tato data následně zpracují a uloží do databáze, tj. RM. V druhé fázi, online, na základě aktuálních měření okolních APs dokážou tyto systémy určit naši aktuální pozici vůči síti RPs uložených v RM.

- Radar: k určování aktuální pozice používá deterministické techniky, jako je výpočet Eukleidovské vzdálenosti algoritmem NNSS. Pro sledování MZ v kontinuálním prostoru používá obměnu Viterbiho algoritmu. Tyto techniky však nepočítají s odchylkami WLAN kanálů, které jsou popsány v 3.

- Horus: tento systém určuje aktuální pozici pravděpodobnostními technikami. K ošetření WLAN odchylek používá *Petrubačnických technik*. Dále pro snížení výpočetní doby je implementován *Incremental Triangulation clustering technique*
- Compass: lokalizační technika stejná jako v případě Horus, doplňuje ošetření blokování signálu lidským tělem. Tato skutečnost má největší dopad na výsledné srovnání. Orientace MZ vůči AP má zásadní význam pro výsledný výstup rozhodovacího algoritmu.

Z výše uvedených informací spolu s [6] se budeme v této práci dále zabývat systémem Compass. Rozhodujícím aspektem je zohlednění orientace při určení pozice.

## Kapitola 4

### Implementace

Pracovní název aplikace je WifiIndoorNavigation (WIT), která je implementována na platformu Android. Vzhledem k tomu, že systém Compass vyžaduje informaci o orientaci MZ, je třeba mít v mobilním zařízení senzory (magnetometr nebo magnetometr s gyroskopem pro větší přesnost) a Android API ve verzi minimálně 11. Dřívější Android verze neposkytují potřebné API pro práci s těmito senzory.

V této práci budeme pracovat s metodami a třídami ze dvou balíků:

- `android.hardware`
- `android.net.wifi`

#### **`android.hardware`**

Tento balík slouží k práci se senzory. Jedná se konkrétně o tři třídy [16]:

- `SensorManager`, poskytuje rozhraní a přístup k jednotlivým sensorům ze třídy `Sensor`
- `SensorEvent`
- `Sensor`, obsahuje všechny dostupné hardwarové senzory a poskytuje z nich získaná data

#### **`android.net.wifi`**

V tomto balíku budeme využívat dvě třídy [17]:

- `WifiManager`, rozhraní obsahující veškeré konstanty ke konfiguraci okolních WiFi a metody pro práci s WiFi senzorem
- `ScanResult`, třída, která popisuje aktuální konfiguraci detekované WiFi sítě, jako je BSSID (MAC adresa AP), SSID (identifikátor sítě), frekvenci, sílu signálu, informace o autentizaci, klíče a čas skenování AP

Stejně, jako je systém Compass rozdělen do dvou fází, tak i WIT obsahuje dvě hlavní části, tj. třídy:

- ScanningEngine, pokrývá veškeré dotazy na snímání otisků a detekci orientace MZ
- SearchingEngine, rozhodovací algoritmus

## 4.1 ScanningEngine (Offline fáze)

### Orientace

Orientaci uživatele s MZ získáme ze senzorů MZ. Sensory potřebné ke zjištění orientace jsou popsány v sekci `android.hardware`. Konstanty přidělené senzorům magnetometru a gyroskopu jsou `TYPE_MAGNETIC_FIELD` resp. `TYPE_GYROSCOPE`. Data ze senzoru obsahující informaci o orientaci nám poskytují třídy z Android API. Tuto informaci převedeme na stupně pomocí vzorce:  $heading = \frac{\pi * data}{180}$ .

Jelikož podle [4] potřebujeme orientaci  $O = \{0^\circ, 45^\circ, \dots, 315^\circ\}$  a *heading* nabývá hodnot  $< 0, \dots, 360$ ), přepočítáme *heading* funkcí  $heading \rightarrow O$ , kdy mezní hranice je  $22,5^\circ$ .

Tzn.:

```
function getOrientation() {
    if (22,5 >= heading && heading < 67,5) {
        return 45
    }
    fi
    if (67,5 >= heading && heading < 112,5) {
        return 90
    }
    fi
    .
    .
    .
}
```

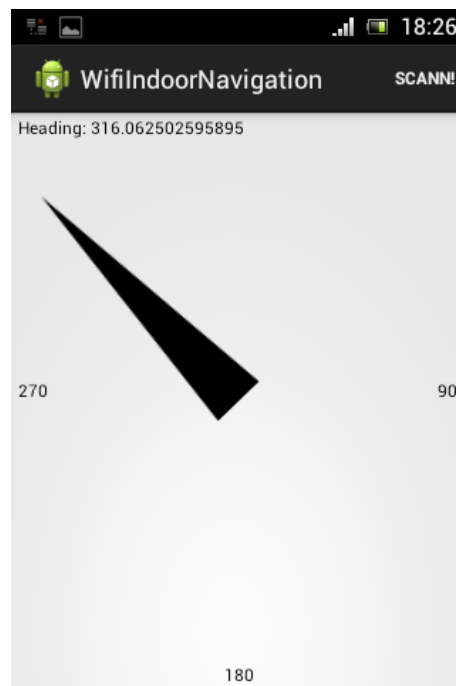
Výstupem je pak orientace MZ  $o \in O$ .

Data z těchto senzorů jsou vizualizována ve `ScanningActivity`, viz obr.4.1, kde černá šipka ukazuje naši orientaci ve stupních (tj. *heading*), a také je uvedena vlevo nahoře jako textové pole.

### Sběr otisků

Ke zpracování dat z AP používáme třídy z `android.net.wifi`:

- ScanResult
- WifiManager



Obrázek 4.1: ScanningActivity

A to konkrétně dvě metody z třídy `WifiManager`. Metoda `startScan` zahájí skenování okolních APs a metoda `getScanResults` vrací výsledky skenování jako seznam `ScanResult`.

Tento seznam dále převádíme na základní model WIN `RssFingerprint`.

`RssFingerprint` obsahuje proměnné pro uložení konfiguračních informací daného AP:

- BSSID
- orientation,  $orientation \in O$
- RP, referenční bod, ze kterého bylo prováděno měření
- average, průměrná síla signálu AP
- deviation, odchylka při měření SS AP

Po převedení `ScanResult`  $\rightarrow$  `RssFingerprint` se seznam ukládá do databáze.

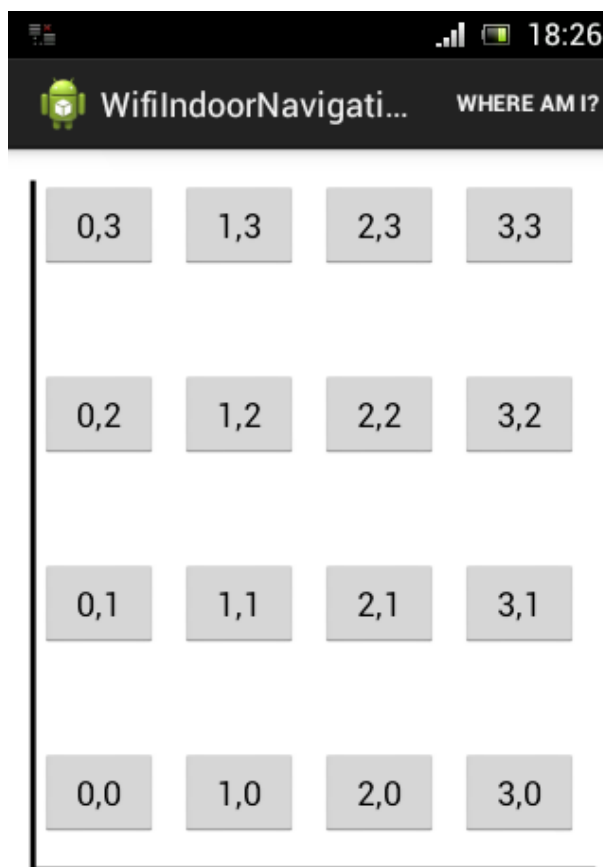
### Databázový model

WIT používá nativní SQLite databázi z balíku `android.database.sqlite`. Vlastní implementovaná databáze obsahuje dvě tabulky:



- TABLE\_RSSFINGERPRINT, obsahuje snímané otisky
- TABLE\_MERGE, tabulka pro uchování otisků podobných stavů, jak bylo popsáno v 3.3

### Offline fáze



Obrázek 4.2: Úvodní obrazovka WIT

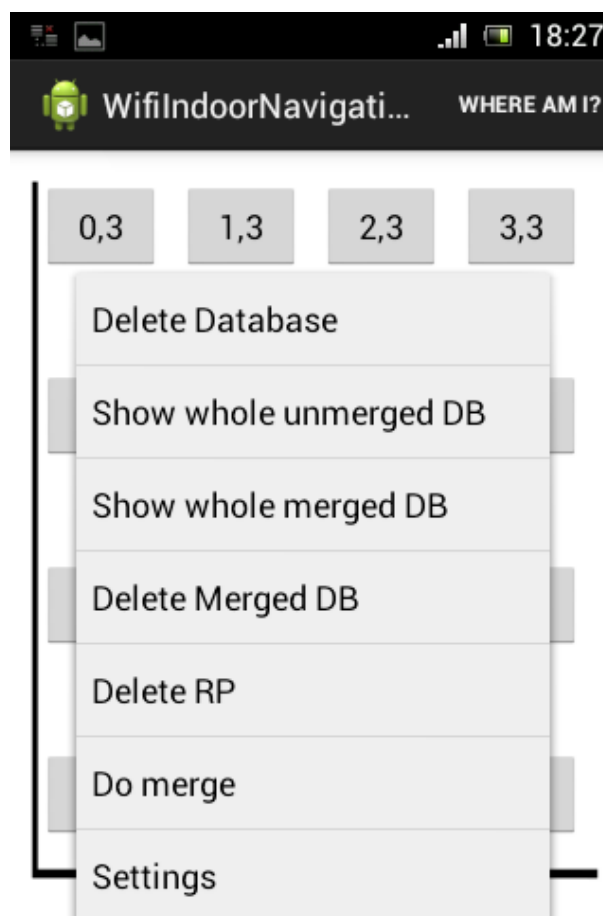
Pro testovací účely uvažujeme testovací prostor rozdělený do mřížky, která se skládá z 4×4 tlačítek, jak je vidět na obrázku 4.2. Označení tlačítek odpovídá označení bodů v kartézské soustavě souřadnic.

Offline fáze spočívá v budování RM. Jakmile máme testovací prostor pokrytý sítí z šestnácti referenčních bodů, na každém z nich provedeme měření. Po stisknutí daného tlačítka, reprezentující RP, kde právě stojíme, se zobrazí skenovací obrazovka, viz obr. 4.1. V této části aplikace provádíme měření. Dle [4] měříme v osmi směrech, vždy po 45°. Textové pole vlevo

nahoře i střelka označuje náš aktuální úhel. Měření spouštíme tlačítkem *SCANN!* umístěným v horní liště.

Pro pokrytí odchylek WLAN při statickém měření snímáme okolní APs po dobu 5 sekund. Všechna data se uloží do mezipaměti MZ a dále se zpracovávají. Spočítá se aritmetický průměr SS APs a jejich odchylka. Vypočítaná odchylka není nutností pro další činnost algoritmu, ale slouží pro pozorovací a testovací účely. Po zprůměrování hodnot se data uloží do tabulky TABLE\_RSSFINGEPRINT. Po naměření hodnot v osmi směrech se vrátíme zpět na hlavní obrazovku tlačítkem *Zpět* a pokračujeme na další RP. Tímto způsobem naskenujeme hodnoty u všech šestnácti bodů v osmi směrech.

Jakmile máme naměřeny hodnoty ve všech RPs, vrátíme se na základní obrazovku a použijeme tlačítko *Možnosti* (4.3) a stiskneme *Do merge*.



Obrázek 4.3: Možnosti

*Do merge* spouští spojování podobných stavů, jak je popsáno v 3.3. Dle [4] je odchylka  $\alpha$  rovna  $69^\circ$ . Za podobné stavy jsou považovány ty, jejichž rozdíl orientací je  $\pm 45^\circ$ , tj. např. pro stav  $s_o = 0^\circ$  jsou podobné stavy  $s_o = 45^\circ$  a  $s_o = 315^\circ$ . Jednotlivé hodnoty se opět průměrují a počítá se nová odchylka, která je již součástí rozhodovacího algoritmu. Spojené stavy se uloží do tabulky TABLE\_MERGE.

## 4.2 SearchingEngine (Online fáze)

Dotazy na aktuální pozici MZ zpracovává třída `SearchingEngine`. Prvním krokem je získání aktuálních otisků okolních APs v dané orientaci  $o$ . Poté se vybere podmnožina RM na základě  $o$ . Vstupem algoritmu je seznam aktuálních otisků okolních APs a mapa databázových dat. Klíčem jsou RPs a hodnotou pak jejich naměřená data.

Nyní se již počítají pravděpodobnosti jednotlivých RPs, kdy se prochází jejich data a porovnávají se s aktuálními měřeními. Při nalezené shodě (tj. shodují se BSSID aktuálního otisku s databázovým otiskem) se počítá dílčí pravděpodobnost pomocí vzorce:

$$P(m_a|r_i) = G_{i,id_a}(p_a)$$

$G_{i,id_a}(p_a)$  se dále počítá intergrálem, viz 3.3. Výpočet integrálu ovšem není Androidem nativně podporován, proto je potřeba použít knihovnu nebo dále vzorec vhodně upravit. Použití knihovny jsme zavrhlí vzhledem k výpočetní náročnosti.

Rozholdi jsme se tedy pro ekvivalentní úpravu integrálu. Každý určitý integrál lze vyjádřit jako:

$$\int_a^b f(x) = g(b) - g(a)$$

kde

$$g'(x) = f(x)$$

Integrál  $G_{i,id_a}(p_a)$  můžeme vyjádřit pomocí Gaussovy  $\text{erf}(x)$ . K vyjádření integrálu jsme použili program Maple.

$$G_{i,id_a}(p_a) = \frac{1}{2} \text{erf} \left( \frac{1}{4} \frac{\sqrt{2}(-2p_a + 1 + 2\mu_{i,id_a})}{\sigma_{i,id_a}} \right) - \frac{1}{2} \text{erf} \left( \frac{1}{4} \frac{\sqrt{2}(-2p_a - 1 + 2\mu_{i,id_a})}{\sigma_{i,id_a}} \right)$$

$\text{Erf}(x)$  lze dále vyjádřit např. Taylorovým polynomem, nebo inverzní funkce  $\text{erf}^{-1}(x)$  je vyjádřitelná Maclaurinovou řadou. Všechny tyto způsoby nicméně neřeší původní problém schnopnosti výpočtu mobilním zařízením.

Pro účely určení pozice však není potřeba přesné vyčíslení, proto jsme se rozhodli  $\text{erf}(x)$  aproximovat. Využili jsme aproximace S.Winitzkiho [18], kdy

$$\text{erf}(x) \approx [1 - e^{-x^2 \frac{4+ax^2}{1+ax^2}}]^{1/2}$$

kde  $a$  je konstanta

$$a \equiv 0,14$$

Tuto aproximaci již MZ dokáže jednoduše vypočítat. Výstupem algoritmu je pak mapa, kde klíčem je RP a hodnota je pak vypočítaná pravděpodobnost. Další krok je nalezení RP

s největší hodnotou, která se zobrazí na hlavní obrazovce. Relevantnější jsou však data, která program ukládá do souboru na externím úložišti. Zde vidíme jednotlivé mezikroky a hlavně kompletní výstup algoritmu.

## Kapitola 5

# Testování

### 5.1 Průběh testování

Testování WIN probíhalo na FI MU, Botanická 68a, Brno, v budově D (před vstupem do posluchárny D2) na zařízení LG Galaxy Nexus s verzí Android API 18. Jako podklad pro RM byla stanovena mřížka  $4 \times 4$  RPs vzdálených metr od sebe, viz obr. 5.1



Obrázek 5.1: Testování, síť RP

Dále byly rozmístněny okolo RP-sítě další 4 APs (např. jeden z nich je vidět na obrázku 5.1 vlevo nahoře, nalevo od oranžového kuželu, v podstavci nástěnky).

V každém RP bylo provedeno snímání otisků okolních APs v osmi směrech, tj. budování RM. Jednotlivé názvy RPs odpovídají názvům RPs aplikace, tzn. (0,0), (1,0) atd.

Při online fázi jsme pak na každém RP provedli čtyři dotazy na aktuální polohu ve čtyřech směrech po  $45^\circ$ . Data, která se uložila do logů, jsme následně převedli do tabulky

pro lepší vizualizaci výsledků. Dalším krokem testování bylo postupné vypínání jednotlivých APs a sledování dopadů na výsledná data. Tabulky reprezentují naši síť, kdy buňka vlevo dole odpovídá RP 0,0 a analogicky vpravo nahoře 3,3. Zeleně jsou vybarvené ty buňky, kdy pravděpodobnost daného RP byla maximální. Naopak červeně jsou vybarveny ty buňky s minimální pravděpodobností daného RP.

### 1. Zapnuty všechny APs

V tomto testu byly všechny APs zapnuty. Na obrázcích jsou výsledky měření v bodě 0,0 ve čtyřech směrech.

	0	1	2	3
3	9,73E-07	3,16E-12	-8,17E-09	0,009901
2	-6,97E-12	1,89E-09	2,44E-14	3,78E-11
1	9,61E-09	5,16E-09	-3,76E-13	4,16E-11
0	3,18E-11	8,18E-14	1,35E-10	-8,42E-09

Obrázek 5.2: Všechny APs, orientace  $o_a$

	0	1	2	3
3	9,71E-07	9,90E-03	6,31E-09	0,012655
2	9,57E-11	-9,26E-11	6,94E-09	5,55E-09
1	4,65E-10	4,30E-09	1,40E-14	1,34E-09
0	1,23E-09	-1,61E-08	-1,45E-12	9,71E-07

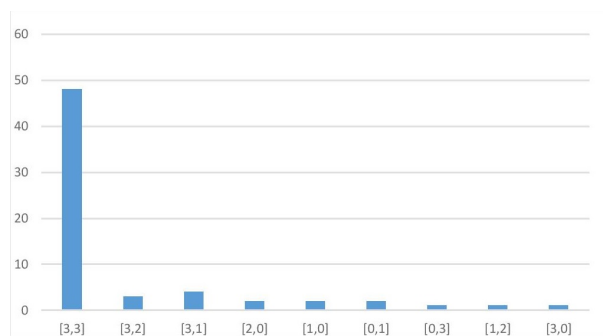
Obrázek 5.3: Všechny APs, orientace  $o_b$

	0	1	2	3
3	1,15E-06	5,89E-07	3,45E-06	1,61E-06
2	1,54E-06	2,91E-06	1,50E-06	6,10E-07
1	7,34E-09	3,93E-09	-1,38E-06	1,91E-06
0	1,29E-07	1,41E-08	3,33E-07	6,60E-07

Obrázek 5.4: Všechny APs, orientace  $o_c$

Z naměřených dat byl RP 3,3 vybrán jako aktuální bod s maximální pravděpodobností. Z kompletních dat, která jsou v příloze, je vidět, že bod 3,3 se nachází v naprosté většině případů jako výsledek vyhledávacího algoritmu. Při výběru pouze jednoho RP s maximální pravděpodobností se tato vzdálenost rovná  $errorDistance = 2,17m$ . Při agregování RPs s podobnou největší hodnotou je tato vzdálenost rovna  $errorDistance = 2,09m$ .

	0	1	2	3
3	2,99E-07	6,27E-07	1,50E-06	4,94E-07
2	8,93E-07	-9,81E-08	1,62E-07	9,12E-07
1	1,03E-07	5,38E-07	-7,25E-07	2,94E-07
0	-1,57E-06	-3,41E-07	-5,43E-07	3,60E-07

Obrázek 5.5: Všechny APs, orientace  $o_d$ 

Obrázek 5.6: Výskyt RP

## 2. Zapnuty dílčí 4 APs

Při zjišťování aktuální polohy byly brány v potaz pouze MAC adresy našich 4 APs, které byly rozmístěny v rozích sítě RPs. Hodnoty byly opět v bodě 0,0 pro čtyři orientace:

	0	1	2	3
3	0,010483	1,70E-04	5,99E-05	0,00990099
2	0,009894	3,13E-05	6,51E-05	9,75679E-05
1	9,80E-05	8,77E-05	9,80E-05	9,80E-05
0	6,97E-05	5,06E-05	0,009893676	0,00899272

Obrázek 5.7: Dílčí APs, orientace  $o_a$ 

	0	1	2	3
3	0,003903	-0,01405	1,62E-04	0,00990099
2	0,009675	6,67E-05	9,80E-05	-2,06321E-05
1	3,86E-05	3,60E-05	9,80E-05	-1,39E-04
0	9,78E-05	9,80E-05	1,04E-04	0,00990099

Obrázek 5.8: Dílčí APs, orientace  $o_b$

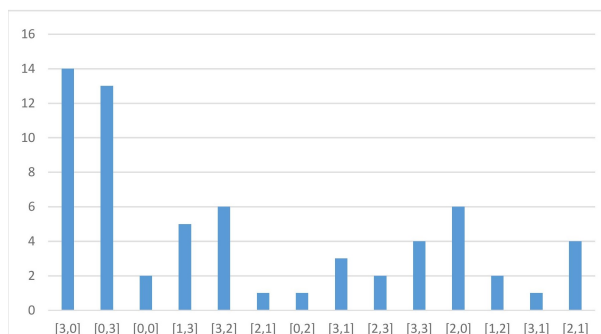
	0	1	2	3
3	-0,01405	0,016386	9,90E-03	0,010133993
2	0,013896	9,91E-03	1,08E-02	-2,50304E-03
1	9,80E-05	7,22E-05	1,59E-02	1,59E-02
0	1,94E-02	1,25E-04	9,90E-03	0,00990099

Obrázek 5.9: Dílčí APs, orientace  $o_c$ 

	0	1	2	3
3	0,000211	0,012655	1,20E-04	-0,002503037
2	-0,00739	1,79E-02	5,38E-04	3,90318E-03
1	7,34E-05	7,30E-05	5,38E-04	3,38E-02
0	1,08E-02	7,41E-04	-1,40E-02	0,017891373

Obrázek 5.10: Dílčí APs, orientace  $o_d$ 

Zde kromě RP 3,3 vystupují i další RPs. Z celkových dat je vidět mnohem širší rozmístění RPs při dotazování na aktuální polohu, viz graf 5.11:



Obrázek 5.11: Výskyt RP

Chybová vzdálenost je pak pro neagregované hodnoty rovna  $errorDistance = 2,33m$  a pro agregované činí  $errorDistance = 2,45m$ . I když je tato hodnota vyšší, než u předchozího případu, je relevantnější z důvodu širšího rozptylu výsledných RPs. V předchozím testu byl v 75% případů vrácen RP 3,3 jako výsledek. V případě použití dílčích 4 APs byl nejčastější výsledek vrácen ve  $\sim 22\%$  případů.

Pro lepší představu zde uvádíme příklad měření v orientaci  $o_a$  uprostřed RM, v bodě 2,2 v případě všech APs a v případě dílčích 4 APs, viz obr. 5.12, 5.13.

### 3. Vypínání dílčích APs

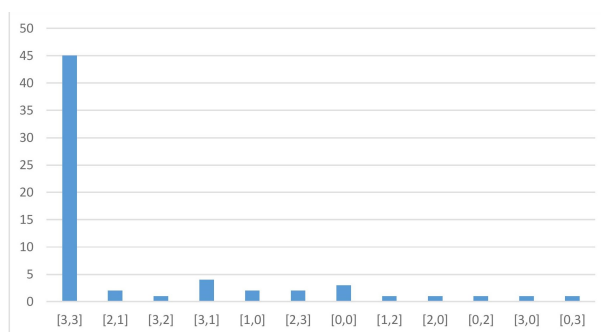
Posledním krokem testování bylo vypínání dílčích APs. V příloze jsou data z dalších měření při vypínání dílčích APs. Výsledky jsou téměř totožné. Uvádíme zde dva grafy rozložení výsledných RPs při vypínání dvou APs.:



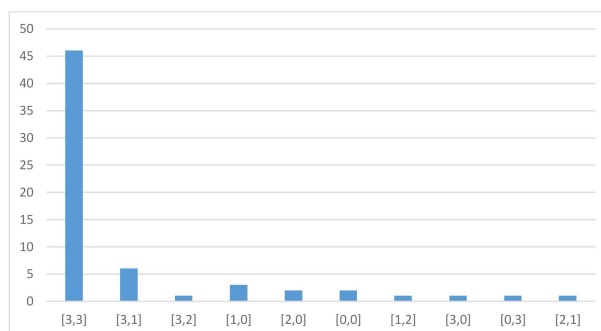
	0	1	2	3
3	9,71E-07	1,41E-10	8,19E-10	0,0099
2	-6,31E-10	1,79E-08	8,39E-12	3,53E-08
1	2,95E-08	4,74E-09	4,07E-10	3,55E-08
0	3,81E-11	2,30E-11	2,60E-08	-4,60E-07

Obrázek 5.12: Výsledky měření v  $o_b$  v bodě 2,2 při všech APs

	0	1	2	3
3	0,009901	0,009319	9,29E-05	0,00990099
2	0,009851	1,18E-04	2,69E-04	-7,32073E-05
1	-1,39E-04	-1,32E-04	2,69E-04	9,80E-05
0	2,69E-04	3,35E-04	9,82E-05	0,00990099

Obrázek 5.13: Výsledky měření v  $o_b$  v bodě 2,2 při dílčích 4 APs

Obrázek 5.14: Vypnutý 1.AP



Obrázek 5.15: Vypnutý 2.AP

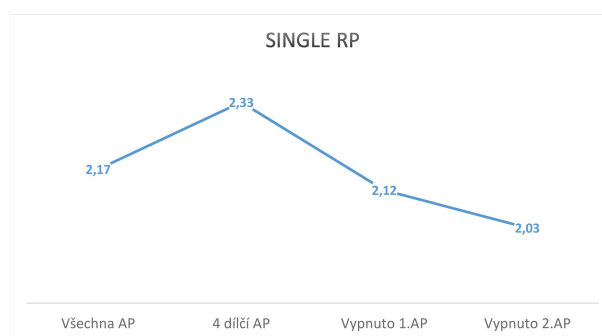
## 5.2 Výsledky testování

### Výběr APs pro měření

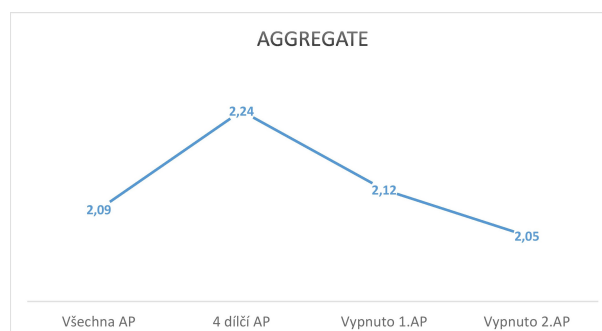
Intuitivně se nabízí, že čím více APs pro měření, tím větší přesnost. Tuto informaci potvrzují i autoři [3]. Z našich výsledků vyplývá, že pro měření je důležitý i vhodný výběr APs. Při měření všech APs byl výstupem algoritmu ve většině případů jeden určitý RP, viz 5.6. V měření dílčích 4 APs vhodně rozmístněných okolo sítě RPs nebyl výstup algoritmu tak jednoznačný ve prospěch jednoho RP, jako v předchozím případě, viz 5.11. Po vypnutí jednotlivých APs byly výsledky totožné s případem zapnutých všech APs. Testováním jsme zjistili, že i vhodný výběr APs pro měření je podstatný vedle celkového počtu APs.

### Agregované vs. neagregované výsledky

Níže na grafech jsou celkové odchylky při dílčích měřeních:



Obrázek 5.16: Neagregované výsledky



Obrázek 5.17: Agregované výsledky

Odchylky jsou téměř totožné. Při srovnání výsledků na obr. 5.12 a 5.13 jsme zjistili, že při agregování výsledků a zohlednění postavení APs (tj. při měření 4 APs vs. při měření všech APs) je výsledná chybová vzdálenost menší. Z toho vyplývá, že výsledné hodnoty je třeba agregovat.

## Kapitola 6

### Závěr

Cílem této práce bylo využít potenciál MZ a jejich senzorů pro lokalizační systém. Naším úkolem bylo prostudovat existující WiFi lokalizační systémy, vybrat nejvhodnější a implementovat jej na platformu Android. Postupně jsme si představili v dnešní době nepoužívanější systémy mimo WiFi. Ukázali jsme si jejich přednosti, ale hlavně nedostatky, díky kterým má smysl vyvíjet jiné systémy, založené např. na WiFi. Dále jsme se již zaměřili na tři WiFi systémy: Radar, Horus a Compass. Porovnali jsme je a vybrali nejvhodnější pro implementaci. Z těchto systémů jsme vybrali Compass, protože jako jediný bere při určení polohy v potaz orientaci MZ. Po implementaci jsme tento systém testovali a porovnávali výsledky měření.

Při porovnávání našich výsledků se systémem Compass jsme zjistili zásadní rozdíl ve výběru APs pro měření. Z našeho testování vyplývá, že vedle počtu APs je důležitý i jejich výběr pro měření. Na rozdíl od výsledků systému Compass, který zohledňuje pouze počet APs [4]. Chybové vzdálenosti pro systém Compass jsou 1,62m při použití přenosného počítače a externí WiFi antény [4]. Naše výsledky jsou  $\sim 2\text{m}$  při použití MZ a výběru vhodného rozestavení APs.

Domníváme se, že tyto rozdíly v odchylkách jsou způsobeny nestabilními daty získanými WiFi senzorem MZ ve srovnání s WiFi senzorem, který používali v [4]. Experimentálně bylo ověřeno, že signál kolísá o jednotky dBm při měření na statické poloze v průběhu času. Toto kolísání může mít rozhodující dopad na chybovou vzdálenost. Testování na citlivějším WiFi senzoru (např. v přenosném počítači) nebylo možné, protože dnešní Android emulátory nepodporují emulaci hardware senzorů.

Pro budoucí rozšíření této implementace přichází v úvahu částečné propojení systémů Compass a Horus. Horus lépe pokrývá WLAN odchylky, jak při statickém měření, tak i pro měření v kontinuálním prostoru.

## Literatura

- [1] YOUSSEF M. a AGRAWALA A. The horus wlan location determination system. Dostupné z: [http://www.cs.umd.edu/~moustafa/papers/horus\\_usenix.pdf](http://www.cs.umd.edu/~moustafa/papers/horus_usenix.pdf), 2005.
- [2] BAHL P. a BALACHANDRAN A. a PADMANABHAN V. Enhancements to the RADAR User Location and Tracking System. Dostupné z: <http://citeseer.ist.psu.edu/bahl00enhancements.html>, 2000.
- [3] KING T. a HAENSELMANN T. a EFFELSBURG W. Deployment, calibration, and measurement factors for position errors in 802.11-based indoor positioning systems, 2007.
- [4] KING T. a KOPF S. a HAENSELMANN T. a LUBBERGER Ch. a EFFELSBURG W. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. Dostupné z: [http://www.researchgate.net/publication/220926437\\_COMPASS\\_A\\_probabilistic\\_indoor\\_positioning\\_system\\_based\\_on\\_802.11\\_and\\_digital\\_compasses/file/9c9605282124fe2a07.pdf](http://www.researchgate.net/publication/220926437_COMPASS_A_probabilistic_indoor_positioning_system_based_on_802.11_and_digital_compasses/file/9c9605282124fe2a07.pdf), September 2006.
- [5] LUNDGREN R.H. a LUNDBERG D. a NORDSTROM E. a TSCHUDIN C. a NIELSEN J. A large-scale testbed for reproducible ad hoc protocol evaluation, March 2002.
- [6] ELADIO M. a ORIOL V. a GERALD F. a RUZENA B. Precise indoor localization using smart phones. Dostupné z: [https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDQQFjAA&url=http%3A%2F%2Fwww.icsi.berkeley.edu%2Fpubs%2Fspeech%2Fpreciseindoor10.pdf&ei=qNN1U9T\\_LafD7Abdm4DIDw&usq=AFQjCNGKf9XT-KghV2Ceh25hJGcSBFg22g&sig2=DxmKLiCmepNqyhJUxNLPgA&bvm=bv.66699033,d.ZGU](https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDQQFjAA&url=http%3A%2F%2Fwww.icsi.berkeley.edu%2Fpubs%2Fspeech%2Fpreciseindoor10.pdf&ei=qNN1U9T_LafD7Abdm4DIDw&usq=AFQjCNGKf9XT-KghV2Ceh25hJGcSBFg22g&sig2=DxmKLiCmepNqyhJUxNLPgA&bvm=bv.66699033,d.ZGU).
- [7] BAUM E.L. a PETRIE T. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 1966.
- [8] TOUMPIS S. a TOUMPAKARIS D. Wireless ad hoc networks and related topologies: applications and research challenges. *Elektrotechnik und Informationstechnik*, 2006.

- 
- [9] VITERBI A.J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, April 1967.
- [10] Figure. Dostupné z: <http://answers.oreilly.com/topic/\discretionary{-}{ }{ }2815-how-devices-gather-location-information/>.
- [11] Figure. Dostupné z: <http://www.globalspec.com/reference/81338/203279/\discretionary{-}{ }{ }chapter-12-mobile-ad-hoc-networks>.
- [12] Figure. Dostupné z: [http://www.h3c.com/portal/res/200812/26/\discretionary{-}{ }{ }20081226\\_709524\\_image005\\_624021\\_57\\_0.png](http://www.h3c.com/portal/res/200812/26/\discretionary{-}{ }{ }20081226_709524_image005_624021_57_0.png).
- [13] Global positioning system, standard positioning service, performance standard. Dostupné z: <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>, September 2008.
- [14] WOODMAN J.O. An introduction to inertial navigation. Technical report, University of Cambridge, August 2007.
- [15] ENGE Petr MISHRA Pratap. *Global Positioning System: Signals, Measurement and Performance Second Edition*. Ganga-Jamuna Press, Massachusetts, 2006.
- [16] Android API Reference. android.hardware. Dostupné z: <http://developer.android.com/reference/android/hardware/package-summary.html>.
- [17] Android API Reference. android.net.wifi. Dostupné z: <http://developer.android.com/reference/android/net/wifi/package-summary.html>.
- [18] WINITZKI S. Approximation to error function. Dostupné z: <http://www.scribd.com/doc/82414963/Winitzki-Approximation-to-Error-Function>, February 2008.
- [19] Understanding wi-fi. Dostupné z: <https://cs.uwaterloo.ca/~brecht/courses/856/readings/background-intro/understandingWiFi.pdf>, January 2002.
- [20] Wikipedia. Viterbi algorithm. Dostupné z: [http://en.wikipedia.org/w/index.php?title=Viterbi\\_algorithm&oldid=604736226](http://en.wikipedia.org/w/index.php?title=Viterbi_algorithm&oldid=604736226), 2014. [Online, cit. 18.3.2014].

## Přílohy

Součástí příloh jsou:

- Instalační balíček WifiIndoorNavigation.apk
- Data z testování. Jednotlivé logy obsahují kompletní data v původní podobě, data v čitelné podobě a MS Excel soubor, kde jsou data vizualizována