



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Magistro baigiamasis darbas

Padėties nustatymas patalpose naudojant WiFi

Atliko:

Kęstutis Opulskis

parašas

Linas Skardžius

parašas

Vadovas:

dr. Valdas Rapševičius

Vilnius
2017

Turinys

Santrauka	4
Summary	5
Įvadas	6
1. Padėties nustatymo schemas	7
1.1. Atstumo nustatymo metodai	7
1.2. Atstumo nustatymo metodai naudojantys signalo atvykimo laiką	7
1.2.1. Signalo atvykimo laiko skirtumo matavimas	8
1.2.2. Abipusis signalo lėkio trukmės matavimas	9
1.2.3. Simetrinis dvikryptis nuotolio nustatymas	10
1.2.4. Signalo atvykimo kampo matavimas	11
1.2.5. Priimto signalo stiprumo matavimas atstumo įvertinimui	11
1.3. Padėties nustatymo algoritmai naudojantys atstumo matavimus	12
1.3.1. Trilateracijos algoritmas	12
1.3.2. Min-max algoritmas	13
1.3.3. Geo-n algoritmas	14
2. Padėties nustatymo schemas paremtos ne atstumų matavimu	16
2.1. Lokacijos algoritmas paremtas centroidų skaičiavimu	16
2.2. Lokacijos algoritmas naudojantis atstumo vektorius ir šuolius tarp objektų	17
3. Algoritmai tinkami naudoti padėties nustatymui WiFi antspaudų metodu	18
3.1. Euklido atstumas	18
3.2. Tanimoto koeficientas	18
3.3. K artimiausių kaimynų	19
3.4. Gauso-Bajeso metodas	19
4. Aukšto nustatymo algoritmai	21
4.1. BCFI algoritmas	21
4.2. Intervalų algoritmas	22
4.3. Aukšto nustatymas naudojant Gauso tikimybių tankio funkciją	22
5. Matavimų kalibravimas	24
5.1. Mažiausių kvadratų metodas	24
6. Prototipinės padėties nustatymo sistemos struktūra	25
6.1. Žemėlapis	26
6.2. QGIS	28
7. Prototipinės sistemos serverio dalis	29
7.1. Duomenų bazė	29
7.2. Web servisai	30
7.2.1. Web servisų kūrimas	31
7.2.2. GET GetWifiData	31

7.2.3.	POST SaveWifiData	31
7.2.4.	DELETE DeleteWifiData	32
7.2.5.	POST GetPosition	32
7.2.6.	POST Calibration	33
7.3.	Lokalizacijos aplikacija	34
8.	Mobiliosios programėlės	35
8.1.	Pasiruošimas mobiliųjų programėlių kūrimui	35
8.1.1.	Mobiliųjų įrenginių platformos kriterijai	35
8.1.2.	Bibliotekos darbai su žemėlapiais	36
8.2.	Duomenų rinkimo programa	36
8.2.1.	V1 versija	37
8.2.2.	V2 versija	40
8.2.3.	V3 versija	40
8.2.4.	Duomenų rinkimas	42
8.3.	Padėties nustatymo programa	44
8.3.1.	Padėties nustatymo programėlės V1 versija	44
8.3.2.	Padėties nustatymo programėlės V2 versija	46
9.	Tyrimai	50
9.1.	Poligonų dydžio ir duomenų matavimo trukmės tyrimas	50
9.2.	Skirtingais telefonais surinktų duomenų tyrimas.	54
9.3.	Aukšto nustatymo tyrimas	57
9.4.	Lokalizacijos algoritmo paremto Gauso-Bajeso metodu tikslumo tyrimas	60
9.4.1.	Tyrimo etapų aprašas	60
9.4.2.	Lokalizacijos algoritmo variantai ir jų rezultatai	61
9.4.3.	Tyrimo rezultatų aptarimas ir pastebėjimai	63
9.5.	Atsitiktinio poligono duomenų tyrimas	63
	Išvados ir rekomendacijos	65
	Gairės	66
	Literatūros šaltiniai	67

Santrauka

Šio darbo tikslas, sukurti padėties nustatymo patalpose sistemos prototipą, naudojančią WiFi atspaudų metodą. Šiame darbe atlikta išsami padėties nustatymo patalpose schemų, metodų, bei algoritmų analizė. Išnagrinėtos lokalizacijos patalpose schemas, paremtos atstumų tarp įrenginių matavimu. Taip pat išanalizuoti padėties nustatymo metodai nenaudojantys atstumų tarp belaidžių įrenginių matavimų. Atlikti aukšto nustatymo pastate ir lokacijos nustatymo aukšte algoritmų tyrimai bei jų analizės. Išanalizuoti algoritmai buvo pritaikyti sukurtai prototipinei padėties nustatymo pastate sistemai. Šiame darbe pateikti detalūs sukurtos prototipinės sistemos dalių aprašymai, aprašyti sąryšiai tarp jų. Įgyvendinta prototipinė padėties nustatymo pastate sistema nenaudoja papildomų resursų, apart virtualizuoto serverio MIF infrastruktūroje, išmaniojo telefono ir pastate veikiančio WiFi belaidžio tinklo. Šio darbo metu sukurta prototipinė sistema veikia MIF pastate Didlaukio g. 47 2-ame, 3-ame ir 4-tame aukštuose. Prototipinėje sistemoje naudojami algoritmai geba dideliu tikslumu įvertinti naudotojo padėtį pastate, tačiau realiai veikiančios sistemos rezultatai labai smarkiai priklauso, nuo sistemos veikimui būtinų, pirminių duomenų naujumo.

Summary

Indoor Localization using WiFi

The main goal of this work is to create a prototype of indoor localization system, which would be able to determine the floor of a building, and location on the floor. In this work methods which are used for indoor localization were analyzed. Also localization schemes using range-based and range-free methods were analyzed and in detail explained in this work. The selected base method for our prototype of indoor localization system is one of range-free methods - WiFi fingerprint method. Other goals of this work are analysis of floor determination algorithms and analysis of localization on the floor algorithms. Best results giving algorithms were adapted to work with our system prototype. Furthermore, the implemented system uses, only MIF infrastructure (existing wireless network in the building, and virtualized server in the MIF cloud system) and smartphone. The created prototype of indoor localization system is implemented to work in the building of MIF faculty (Didlaukio g. 47), on 2nd, 3rd and 4th floors. According to our analysis of algorithms, which are being used in our implemented localization system prototype, they can estimate correct location with high accuracy, but as we have seen in our prototype system, the accuracy highly suffers from the recency of the reference data. It means that we get best results when reference data is newly collected. And after that as time passes by, and our reference isn't renewed the accuracy of our system's localization decreases.

Ivydas

Šiais laikais, kai visame pasaulyje yra statomi didžiuliai prekybos centrai, parodų rūmai, muziejai, gamyklos, mokslinių tyrimų centrai tampa aktualu greitai ir efektyviai susiorientuoti, šių statinių erdvėje. Kadangi dažniausiai pastatų viduje, palydovinės pozicionavimo sistemos neveikia, atsiranda padėties nustatymo sistemos veikiančios pastatų viduje poreikis. Tokią sistemą galima pritaikyti nuo automatizuotų muziejų giduose, reklamų vaizdavimo, priklausomai nuo parduotuvių esančių aplink vartotojo lokaciją, prekybos centre ar darbuotojų efektyviam judėjimui gamykloje užtikrinti iki navigacijos programų pastatuose. Tačiau, kad ir koks būtų galimas tokios sistemos pritaikymas, ši sistema pirmiausia turi sugebėti nustatyti padėtį pastate.

Kadangi daugumos pastatų viduje yra įrengiami bevielės prieigos WiFi tinklai, aktualu išsiaiškinti ar panaudojus šiuos tinklus galima sukurti padėties nustatymo sistemą veikiančią patalpose. Šio darbo tikslas sukurti prototipinę padėties nustatymo sistemą, kuri sugebėtų nustatyti aukštą pastate ir padėtį aukšte bei veiktų išmaniajame telefone. O darbo uždaviniai: aukšto nustatymo pastate tyrimas bei jo analizė bei padėties nustatymo aukšte tyrimas ir jo analizė. Darbo metu buvo atlikti uždaviniuose iškelti tyrimai. Parinkti bei pritaikyti prototipinei sistemai, padėties bei aukštų nustatymo algoritmai. Pasiiekti rezultatai dviprasmiški: naudojant prieš tai minėtus algoritmus galima gauti didelio tikslumo lokacijos įvertinimus, tačiau realiai veikiančioje sistemoje šie įvertinimai labai smarkiai priklauso nuo pirminių duomenų naujumo.

Iš "Mokslo tiriamojo darbo projekto" paimta 1-ojo ir 2-ojo skyrių medžiaga. Taip pat 3.1, 3.2, 3.3, 8.1 poskyrių medžiaga bei dalis medžiagos esančios 6.1, 6.2, 8.2.1, 8.2.2 poskyriuose.

Kęstutis Opulskis atsakingas už prototipinėje sistemoje naudojamų žemėlapių apdorojimą. Taip pat už prototipinės sistemos serverio dalies įgyvendinimą. Linas Skardžius atsakingas už prototipinėje sistemoje naudojamų mobiliųjų programėlių sukūrimą. Abu darbo autoriai, visus šiame darbe aprašytus tyrimus atliko kartu.

1-ajame ir 2-ajame skyriuose aprašyta padėties nustatymo metodų pastatuose apžvalga. 3-ajame skyriuje aprašyti algoritmai tinkami padėties nustatymui patalpose, naudojant WiFi atspaudų metodą. 4-ajame skyriuje aprašyti aukšto nustatymo pastate algoritmai. 5-ajame skyriuje aprašytas matavimų kalibravimas. 6-ajame, 7-ajame ir 8-ajame skyriuose detalai aprašyta sukurta prototipinė padėties nustatymo sistema ir jos veikimo principas. 9-ajame skyriuje pateikiami atlikti tyrimai ir jų analizės.

1. Padėties nustatymo schemas

Objekto lokacijos (vietos) nustatymui naudojamas schemas galima suskirstyti į paremtas atstumo, tarp lokalizuojamų objektų ir žinomų padėčių objektų skaičiavimu [angl. *range-based*], ir schemas, kurios remiasi objektų tarpusavio ryšiais arba topologine informacija [angl. *range-free*]. Atitinkamai šias lokacijos nustatymo schemų grupes galime išskirti į naudojančias orientyrus (objektus siųstuvus-imtuvus, kurių padėtis yra žinoma)[angl. *anchor based*] ir jų nenaudojančias [angl. *anchor free*]. Tiek atstumų matavimu paremtos schemas, tiek kitos rūšies schemas, gali naudoti ne vieną metodą, o kelis - tokios schemas vadinamos mišriomis [5]. Toliau šiame skyriuje aprašyti atstumo nustatymą naudojančios metodai, o 2 skyriuje aprašytos schemas paremtos, ne atstumų matavimu.

1.1. Atstumo nustatymo metodai

Atstumų matavimu paremtų schemų bendras bruožas - pirmiausia įvertinamas atstumas arba kampas tarp dviejų objektų, o tada vykdomas nežinomos lokacijos objekto padėties apskaičiavimas. Atstumas gali būti įvertinamas keliais skirtingais būdais:

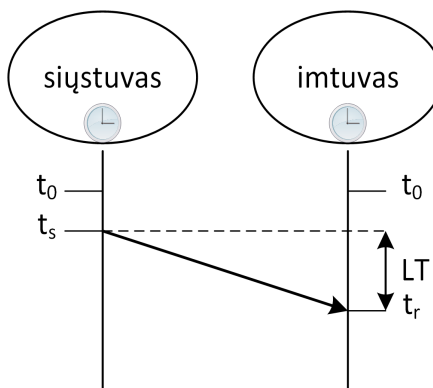
- naudojant signalo stiprumo matavimus [angl. *Received Signal Strength Indication (RSSI)*];
- matuojant signalo atvykimo kampą [angl. *Angle of Arrival*];
- naudojant laiko, signalo lėkio trukmės [angl. *Time of Flight*] matavimus:
 - signalų atvykimo laiko skirtumo matavimus [angl. *Time Difference of Arrival*];
 - abipusius signalo lėkio trukmės matavimus [angl. *Round Trip Time of Flight*].

1.2. Atstumo nustatymo metodai naudojantys signalo atvykimo laiką

Atstumą tarp dviejų objektų naudojant signalo atvykimo laiką, kai žinomas išsiuntimo laikas (turint signalo lėkio trukmę [angl. *Time of Flight*]) galima apskaičiuoti, kadangi elektromagnetinės bangos sklinda šviesos greičiu c (vakuume), o medžiagose $v = c/n$, kur n - medžiagos lūžio rodiklis. Pagal kelio formulę:

$$d = v \cdot t \tag{1.1}$$

kur d - atstumas, c - elektromagnetinės bangos sklidimo greitis, t - signalo lėkio trukmė.

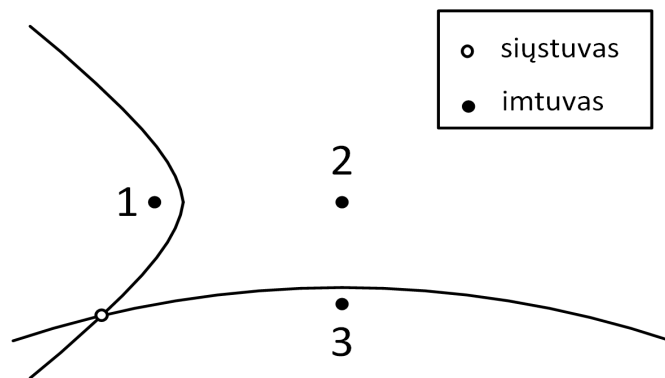


1 pav. Lėkio trukmės matavimas, kai imtuvas ir siųstuvus sinchronizuoti

1 paveiksle pateiktas pavyzdys, kaip gaunama lėkio trukmė idealiu atveju, kai siųstuvas ir imtuvas yra sinchronizuoti laike. Iš siųstuvo išsiunčiamas signalas, kuriame nurodomas tikslus išsiuntimo laikas. Tada šis signalas yra priimamas imtuve, kadangi laikas tarp siųstuvo ir imtuvo yra sinchronizuotas, tai imtuve iš laiko momento kuriuo buvo priimtas signalas atimamas signale nurodytas išsiuntimo laikas. Apskaičiuotas laikas t - signalo lėkio trukmė, o naudojantis 1.1 apskaičiuojamas atstumas tarp siųstuvo ir imtuvo. Reikia atkreipti dėmesį į tai, kad net menkiausias laiko nesutapimas siųstuve ir imtuve, matuojant tokiu metodu, lemia didelę paklaidą atstumo matavime (1ns paklaida laiko sinchronizavime lemia 30cm paklaidą atstumo matavime). [6, 20]

1.2.1. Signalo atvykimo laiko skirtumo matavimas

Signalo atvykimo laiko skirtumo matavimas, dar kitaip vadinamas vienos krypties nuotolio nustatymu [*angl. one way ranging*]. Norint šiuo būdu nustatyti siųstuvo atstumą iki imtuvo reikia laike sinchronizuotų imtuvų, o siųstuvas gali būti nesinchronizuotas. Signalo atvykimo laiko skirtumo matavimas gali būti atliekamas keliais būdais.



2 pav. Signalo atvykimo laiko skirtumo vertės radimas, remiantis signalo lėkio trukmės matavimais

Vienas iš jų pateiktas 2 paveiksle. Turime siųtuvą, kurio padėtį norime surasti ir du (pvz. 1-ąjį ir 2-ąjį) imtuvus, kurie yra sinchronizuoti laike. Siųstuvu išsiunčiamas signalas, kuris yra priimamas 1-ajame imtuve laiko momentu t_1 ir 2-ajame laiko momentu t_2 . Kadangi imtuvai tarpusavyje yra sinchronizuoti laike, tai skirtumas tarp siųstuvo laiko ir imtuvų laiko yra vienodas abiejuose imtuvuose. Iš to išeina, kad signalo atvykimo laiko skirtumas yra lygus:

$$t_{TDOA} = t_2 - t_1 \quad (1.2)$$

Tada, pasinaudojus 1.1 formule, gauname galimą siųstuvo poziciją - hiperbolės, kurios židiniai yra žinomos imtuvų padėties, šaką. Jeigu į sistemą pridėdame dar vieną imtuvą ir pasinaudoję 2 formule gauname kitos hiperbolės šaką, kuri nusako galimą siųstuvo poziciją 2-ojo ir 3-ojo imtuvų atžvilgiu. Šių hiperbolių sankirtos taškas yra siųstuvo padėtis plokštumoje.

Kitas būdas, gauti signalo atvykimo laiko skirtumo vertes - skaičiuoti tarpusavio koreliaciją, tarp signalų priimtų imtuvuose. Tarpusavio koreliacijos formulė:

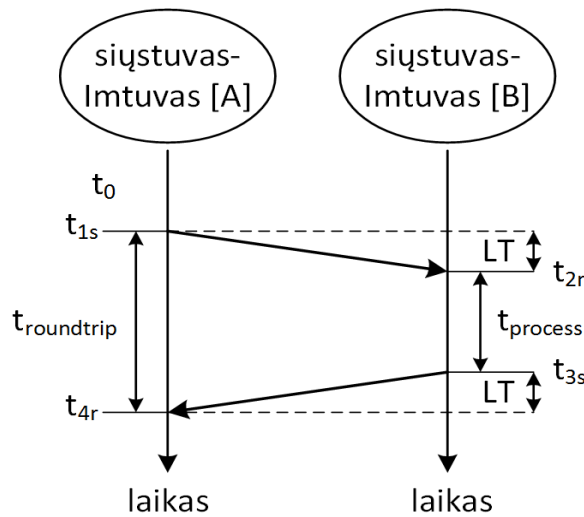
$$K_{1,2}(\tau) = \frac{1}{T} \int_0^T f_1(t)f_2(t + \tau)dt \quad (1.3)$$

kur $f_1(t)f_2(t)$ - signalai atitinkamai 1-ajame ir 2-ajame imtuvuose, T - stebėjimo trukmė. Kadangi žinome, jog signalas imtuvus pasiekia skirtingais laiko momentais arba, kitaip tariant, vienas vėluoja kito atžvilgiu. Tai didžiausią tarpusavio koreliaciją gausime, tuo momentu, kai vienas iš signalų bus pastumtas per vėlinimo trukmę τ . τ reikšmė, ties kuria tarpusavio koreliacija didžiausia ir yra signalo atvykimo laiko skirtumo vertė t_{TDOA} . [25, 13]

Pagrindinis privalumas naudojant vienos krypties atstumo nustatymą yra tas, jog siųstuvas naudoja labai mažai energijos, kadangi jam reikia tik išsiųsti signalą - nereikia būti budėjimo režime, laukiant signalų iš kitų siųstuvų-imtuvų. Tačiau norint lokalizuoti siųstuvą su pakankamai dideliu tikslumu, imtuvai turi būti labai tiksliai sinchronizuoti laike.

1.2.2. Abipusis signalo lėkio trukmės matavimas

Abipusis signalo lėkio trukmės matavimas, dar vadinamas dvikrypčiu nuotolio nustatymu [*angl. Two-way ranging*]. Naudojantis šiuo metodu galima rasti atstumą, tarp imtuvo ir siųstuvo, kurie nėra sinchronizuoti laike.



3 pav. Dvikryptis nuotolio nustatymas

Iš siųstuvo-imtuvo A išsiunčiamas signalas laiko momentu t_{1s} , laiko momentu t_{2r} šis signalas priimamas siųstuve-imtuve B, apskaičiuojamas siųstuve-imtuve B jo apdorojimo laikas $t_{process}$ ir su šia informacija atgal išsiunčiamas signalas siųstuvui-imtuvui A laiko momentu t_{3s} . Siųstuve-imtuve A priimamas iš B išsiųstas signalas, kuriame pateikta signalo apdorojimo trukmės $t_{process}$ (arba signalo priėmimo siųstuve-imtuve B ir išsiuntimo iš siųstuvo-imtuvo B laiko momentai t_{2r} ir t_{3s}) informacija siųstuve-imtuve B. Siųstuve-imtuve A yra žinomas pirmojo signalo išsiuntimo laiko momentas t_{1s} ir laiko momentas t_{4r} kuriuo buvo priimtas iš siųstuvo-imtuvo B siųstas signalas. Todėl atėmus šias laiko momentų vertes ir padalinus gautą rezultatą iš dviejų (kadangi signalas buvo siunčiamas 2 kartus), - gaunama signalo lėkio trukmė. Kaip matome iš 3 paveikslo, lėkio trukmė bus lygi:

$$T_{LT} = \frac{t_{4r} - t_{1s} - t_{process}}{2} = \frac{t_{roundtrip} - t_{process}}{2}, \quad (1.4)$$

o atstumas tarp siųstuvo ir imtuvo apskaičiuojamas pagal 1.1.

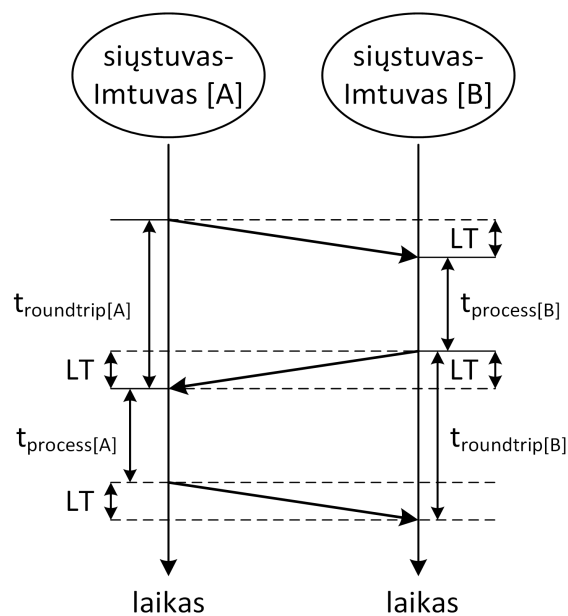
Būtent galimybė, siųstuvams-imtuvams tarpusavyje nebūti sinchronizuotiems laike, lemia šio metodo dažną naudojimą praktiniuose taikymuose, nes bevielių įrenginių tiksliai sinchronizuoti tarpusavyje laike, dažnai neleidžia fizinės įrangos galimybės. Tačiau siųstuvai-imtuvai turi turėti aukštos raiškos vidinius laikrodžius, kad gebėtų pakankamu tikslumu (nano sekundžių eilės) pateikti laiko skirtumus $t_{roundtrip}$, $t_{process}$. Dar vienas šio metodo trūkumas yra tas, kad vienas siųstuvai-imtuvai turi pamatuoti signalo lėkio trukmę su kiekvienu kitu pasiekiamu siųstuvu-imtuvu atskirai ir tai turi daryti skirtingais laiko momentais, kai tuo tarpu lokacija naudojant vienos krypties nuotolio matavimo metodus vyksta tuo pačiu metu visuose imtuvuose, esančiuose signalo pasiekiamumo (ryšio) zonoje. Taip pat reikia paminėti, kad norint nustatyti lokalizuojamo siųstuvo-imtovo padėtį, reikalingas papildomas įrenginys, serveris, į kurį siunčiami matavimų duomenys (arba laiko momentų vertės, arba jau apskaičiuotos atstumų vertės), o jame yra naudojami lokalizacijos algoritmai, apskaičiuojantys siųstuvo padėtį. [8, 7, 20]

Visuose įrenginiuose, kuriais galima atlikti lėkio trukmės, matavimus yra osciliatoriai, kurie generuoja pagrindinį įrenginio veikimo dažnį, nuo kurio priklauso ir laiko trukmės matavimai. Kadangi osciliatoriai generuoja dažnį su tam tikra paklaida, todėl laiko vertės svyruoja kiekviename įrenginyje atskirai.

1.2.3. Simetrinis dvikryptis nuotolio nustatymas

Tam, kad paklaida dėl laiko deviacijos įrenginiuose turėtų mažesnę įtaką matavimams, naudojamas simetrinis dvikryptis nuotolio nustatymas [*angl. Symmetric double-sided two-way ranging*].

Kaip matome 4 paveiksle, nuo paprasto dvikrypčio nuotolio nustatymo šis metodas skiriasi tik tuo, kad signalas iš siųstuvo-imtovo A yra išsiunčiamas papildomai dar vieną kartą, kai apdoroja atsakymą iš siųstuvo-imtovo B.

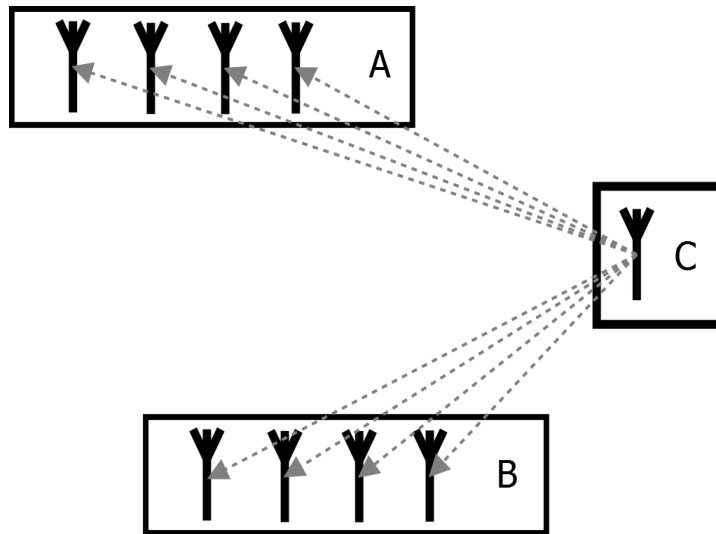


4 pav. Simetrinis dvikryptis nuotolio nustatymas

Tačiau nors papildomas signalo siuntimas, sumažina paklaidą, kurią sukelia laiko deviacija įrenginiuose, šis veiksmas pailgina lokalizacijos trukmę.

1.2.4. Signalo atvykimo kampo matavimas

Signalo atvykimo kampo iki imtuvo antenos nustatymas pasiekiamas keliais būdais. Vienas iš jų - imtuvuose naudoti kryptines antenas, kurios gali mechaniškai judėti. Tokiu atveju siųstuvas išsiunčia signalą, o imtuvas keičia kryptinės antenos padėtį, tol kol priimto signalo stiprumas pasiekia didžiausią vertę. Tada tarp žinomos orientacinės krypties (pavyzdžiui šiaurės krypties) ir maksimalaus priimto signalo stiprumo krypties turime kampą. Jei turime 2 imtuvus, tai galime nustatyti siųstuvo padėtį plokštumoje, jei turime 3 imtuvus (kurie nėra išdėstyti kolineariai) galime lokalizuoti siųstuvą erdvėje. Tačiau šis signalo atvykimo kampo nustatymo būdas netinka naudojimui patalpose, kadangi dėl atspindžių nuo aplinkui esančių objektų signalo stiprumo vertės yra iškreipomos.



5 pav. Signalo atvykimo kampo matavimas, naudojant antenų masyvus.

Kitas metodas gauti signalo atvykimo kampą iki imtuvo antenos pateiktas 5 paveiksle. Čia A ir B imtuvai turintys antenų masyvus, o C - siųstuvas. Iš C pasiųstas signalas pasiekia kiekvieno iš imtuvų antenas skirtingais laiko momentais, o imtuvas tiksliai išmatuoja laiko momentus, kuriais signalas pasiekia kiekvieną iš antenų. Tada, žinant laiko skirtumus ir atstumus tarp antenų, galima nustatyti signalo atvykimo kryptį, o tuo pačiu ir kampą. Šis metodas tinka matavimams atvirose erdvėse, kur yra užtikrinamas tiesioginis matomumas tarp siųstuvo ir imtuvo bei nėra atspindžių nuo pašalinių objektų. [6, 19]

1.2.5. Priimto signalo stiprumo matavimas atstumo įvertinimui

Šis metodas remiasi fizikine elektromagnetinių bangų sklidimo savybe neapribotoje erdvėje [*angl. free-space*] - signalo stiprumas slopsta tostant nuo signalo šaltinio. Tai galima pamatyti iš *Frii* perdavimo lygties, kuri parodo signalo galią priimtą antenoje P_R , kai žinoma kitos atstumu d nutolusios antenos išspinduliuota galia P_T .

$$P_R = P_T \frac{G_R G_T \lambda^2}{(4\pi)^2 d^n} \quad (1.5)$$

kur P_R - priimto signalo galia (W), P_T - išspinduliuoto signalo galia (W), G_R - priimančios antenos stiprinimo koeficientas, G_T - išspinduliuojančios antenos stiprinimo koeficientas, λ - signalo bangos ilgis, d - atstumas tarp antenų (m). n - signalo sklidimo konstanta, kuri priklauso nuo

aplinkos, idealiu atveju (neapribotoje erdvėje), esant antenų tiesioginiam matomumui bei nesusidarant signalo atspindžiams nuo aplinkoje esančių objektų, lygi 2. Dažniausiai svyruoja $2 < n < 4$, bet gali apkrautose erdvėse, patalpose išaugti iki 6. Taip pat įmanomas variantas, kai $n < 2$ - tokiu atveju aplinka veikia kaip bangolaidis. Kadangi priimto signalo stiprumas dažniausiai matuojamas dBm (decibelais milivato atžvilgiu), tai galia bus lygi:

$$P[dBm] = 10 \log_{10}(P[W] \cdot 10^3) \quad (1.6)$$

o formulė 1.5 bus lygi:

$$P_R[dBm] = P_T + G_T + G_R + 20 \log_{10} \left(\frac{\lambda}{4\pi} \right) - n \cdot 10 \log_{10} \left(\frac{d}{d_0} \right) \quad (1.7)$$

kur d_0 - atskaitinis atstumas (toliau $d_0 = 1$ m) tada priimto signalo stiprumas decibelais:

$$P_R[dBm] = A - n \cdot 10 \log_{10}(d) \quad (1.8)$$

čia A iš anksto gauta vertė (priimto signalo stiprumas [dBm] atstumu d_0), priklausantis tik nuo fizinių antenų savybių. Iš 1.8 formulės išsireiškiame atstumą [12]:

$$d = 10^{\left(\frac{A - P_R}{10 \cdot n}\right)} \quad (1.9)$$

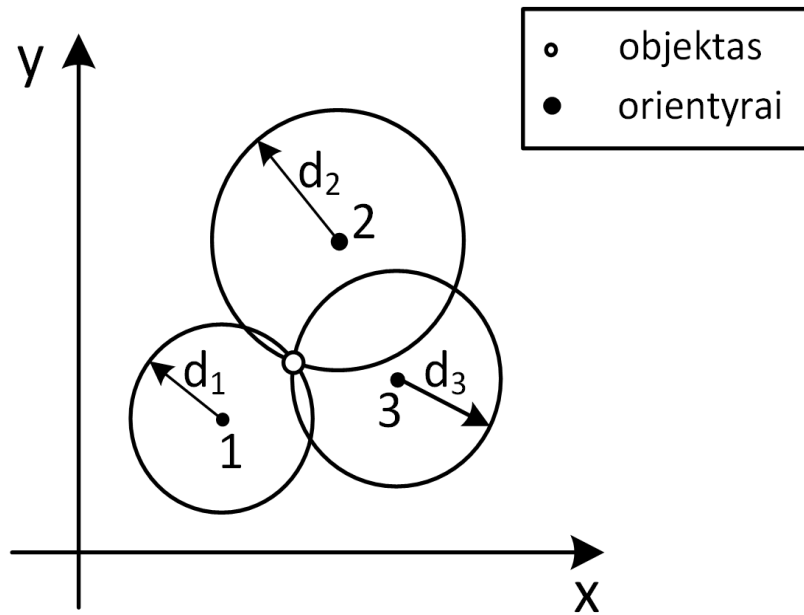
Šio atstumo matavimo metodo vienas pagrindinių privalumų yra tas, jog siųstuvas ir imtuvas neprivalo būti sinchronizuoti laike. Taip pat signalo stiprumą pamatuoti nėra sudėtinga užduotis. Tačiau šiuo būdu matuotuoju atstumą tarp objektų uždaroje patalpose bei aplinkui esant kitiems objektams gaunami labai netikslūs matavimai. Viena iš pagrindinių priežasčių, tai nuo sienų ar kitų objektų susidarys signalo atspindžiai, kurie interferuoja su pagrindiniu signalu ir sukelia konstruktyvias bei destruktivias interferencijas, kurių metu matuojamos priimto signalo stiprumo vertės gaunamos didesnės arba mažesnės nei vertės, kurias gautume matuodami vien pagrindinio signalo stiprumą. [6, 20]

1.3. Padėties nustatymo algoritmai naudojantys atstumo matavimus

1.3.1. Trilateracijos algoritmas

Tai geometrija paremtas algoritmas. Turėdami, orientyrų koordinates (x_i, y_i) ir atstumus d_i (išmatuotus atstumo nustatymo metodais) nuo jų iki objekto, kurio lokaciją norime rasti, brėžiame d_i ilgio spindulio apskritimus (jei pozicionuojame plokštumoje). Tokia situacija pateikta 6 paveiksle.

Kaip matome idealiu atveju, kai atstumai tarp objekto ir orientyrų išmatuoti tiksliai, tai objekto padėtis yra visų apskritimų bendras susikirtimo taškas.



6 pav. Trilateracijos veikimo principas

Jo koordinatės gaunamos išsprendus netiesinių lygčių sistemą, trilateracijos atveju:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = d_3^2. \end{cases} \quad (1.10)$$

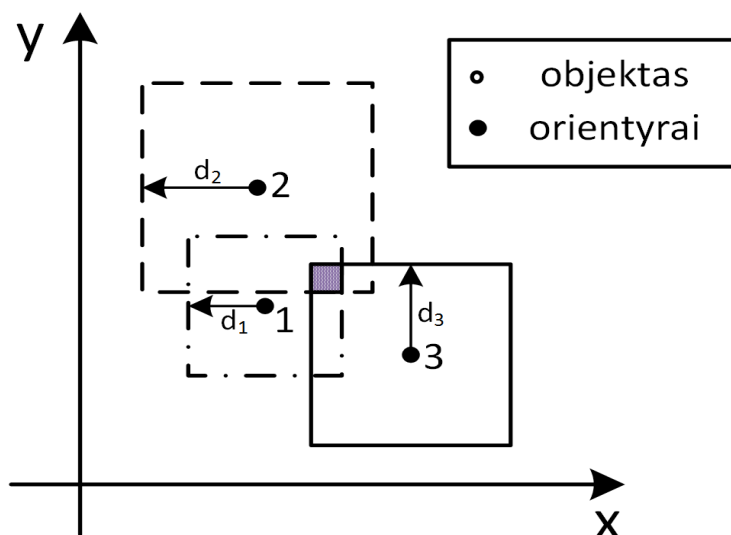
O multilateracijos atveju:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = d_3^2 \\ : \\ (x - x_N)^2 + (y - y_N)^2 = d_N^2 \end{cases} \quad (1.11)$$

kur N - orientyrų skaičius. Tokios lygčių sistemos gali būti sprendžiamos mažiausių kvadratų metodu, arba netiesinę lygčių sistemą paverčiant tiesine lygčių sistema. Multilateracijos atveju tai galima padaryti paėmus 1 iš 1.11 lygčių ir ją atėmus iš likusių sistemos lygčių.

1.3.2. Min-max algoritmas

Tai taip pat kaip ir trilateracija, min-max yra geometrija paremtas algoritmas. Turėdami orientyrų koordinates (x_i, y_i) ir atstumo nustatymo metodais gautas atstumų vertes d_i . Brėžiame kvadratus (apribojančias dėžes), kurių centre yra orientyrai, o kraštinių ilgių lygūs $2d_i$.



7 pav. Min-max algoritmo veikimo principas

Tada kaip matome 7 paveiksle, yra ieškoma, kur gauname nubrėžtų kvadratų sankirtą - kitą stačiakampį. Objektui, kurio padėtis yra ieškoma priskiriamos, sankirtų suformuoto stačiakampio centro koordinatės. Matematiškai ieškant objekto lokacijos min-max algoritmu, reikia apskaičiuoti minimalias ir maksimalias vertes - apribojančios dėžės viršūnių x ir y reikšmes :

$$\begin{cases} x_{left} = \max(x_i - d_i) \\ x_{right} = \min(x_i + d_i) \\ y_{bottom} = \max(y_i - d_i) \\ y_{top} = \min(y_i + d_i) \end{cases} \quad (1.12)$$

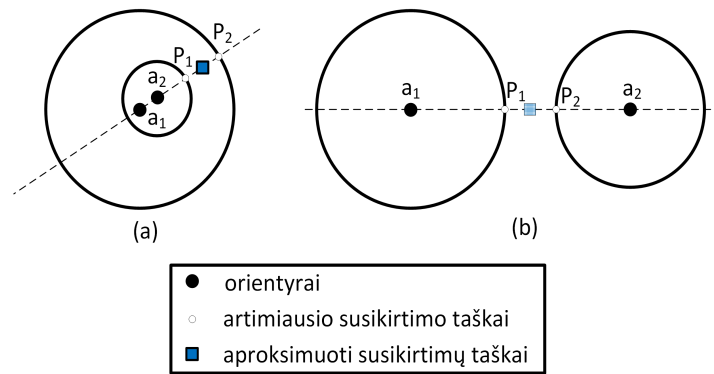
Iš jų matome, kad sankirtų suformuoto stačiakampio koordinatės yra $(x_{left}, y_{bottom}) \times (x_{right}, y_{top})$ arba kitaip perrašius $(\max(x_i - d_i), \max(y_i - d_i)) \times (\min(x_i + d_i), \min(y_i + d_i))$ Tada min-max algoritmu nustatytos objekto koordinatės yra lygios šio stačiakampio centro koordinatėms [2, 12, 14].

1.3.3. Geo-n algoritmas

Geo-n algoritmas specialiai sukurtas nustatyti objektų lokaciją pastatuose. Kur pasireiškia atspindžiai nuo pašalinių objektų, aplinka triukšminga signalų atžvilgiu ir dalis atstumo matavimų gali būti gauti esant netiesioginiam matomumui tarp objekto ir orientyro.

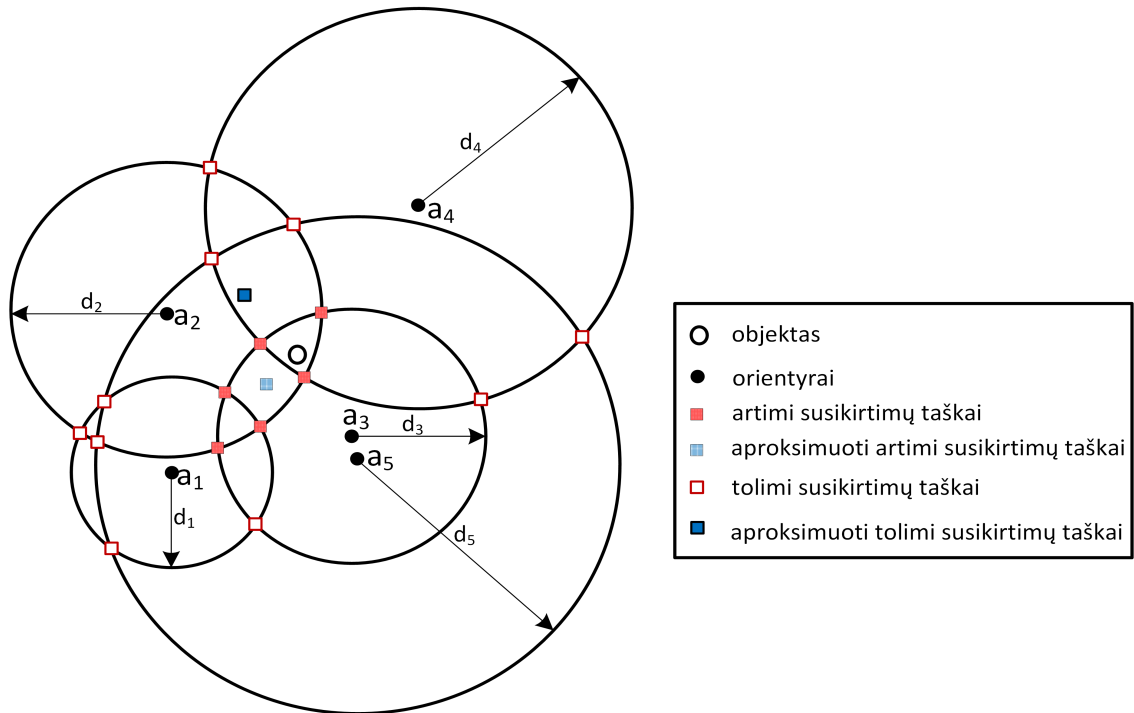
Turint objektą o , kurio padėtį reikia surasti ir rinkinį orientyrų a (nuo žodžio [*angl. anchor*]) - $A = a_1, a_2, a_3, \dots, a_N$, kurių padėtį žinome. Taip pat turint išmatuotus atstumus d_i , tarp objekto ir kiekvieno iš orientyrų. Galime sudaryti rinkinį apskritimų $C = C_1, C_2, C_3, \dots, C_N$, kurių centrai yra orientyrų padėtyse, o spinduliai yra atstumai d_i .

Pirmiausia ieškoma visų sankirtų taškų tarp kiekvienos iš apskritimų esančių rinkinyje C porų. Daugiausia sankirtos taškų tarp dviejų skirtingo dydžio apskritimų gali būti 2. Iš kurių vienas būna arčiau objekto pozicijos, o kitas toliau, tačiau pasitaiko, kad objektas yra nutolęs vienodu atstumu iki abiejų sankirtos taškų. Taip pat gali būti vienas sankirtos taškas, jei atstumai būtų išmatuoti tiksliai, tai visi apskritimai turėtų susikirsti viename taške. Kadangi realūs matavimai yra su paklaidomis, tai sankirtos taškų tarp dviejų apskritimų gali nebūti visai.



8 pav. Geo-n aproksimuotų susikirtimo taškų radimas

Kaip matome 8 paveiksle tokiu atveju įmanomi 2 variantai, kai vienas apskritimas patenka į kito vidų (a) - variantas, arba kai išmatuoti atstumai gauti per trumpi ir apskritimai yra vienas šalia kito, bet nesikerta (b) - variantas. Abiem atvejais geo-n algoritme yra skaičiuojama sankirtos aproksimacija, kurios padėtis yra linijos, jungiančios arčiausiai dviejų apskritimų sankirtos esančius taškus P_1 ir P_2 vidurys.



9 pav. Geo-n algoritmo sankirtų ieškojimo vizualizacija

Kaip matome 9 paveiksle, dažniausiai apskritimai susikerta dviejuose taškuose ir tiek vienas iš jų būna arčiau objekto padėties, o didžiausias kiekis sankirtų yra arti ieškomo objekto lokacijos. Šios grupės taškų radimas ir apdorojimas yra pagrindinė geo-n algoritmo užduotis.

2. Padėties nustatymo schemos paremtos ne atstumų matavimu

Skirtingai, nei prieš tai aprašytos lokacijos schemos paremtos atstumų matavimu, kurios pirma įvertina atstumą, tarp orientyro ir ieškomo objekto lokacijos, o paskui naudoja algoritmus panaudojančius atstumų matavimus; šios schemos neturi tokios bendros struktūros.

2.1. Lokacijos algoritmas paremtas centroidų skaičiavimu

Lokacijos algoritmas paremtas centroidų skaičiavimu [*angl. Centroid localization Algorithm (CA)*] viena iš padėties nustatymo schemų nenaudojančių atstumų tarp objektų matavimų, bet naudojanti orientyrus (atskaitos objektus).

Sistema sudaryta iš siųstuvų - orientyrų, naudojančių įvairiakryptes [*angl. omnidirectional*] antenas (kurios spinduliuoja signalą į visas puses vienodai), pastatytų taip, kad jų dengiamos zonos kuo labiau persiklotų tarpusavyje, ir lokalizuojamų objektų - imtuvų. Orientyrus žymėkime R_1, R_2, \dots, R_n , o jų koordinatas - $((X_1, Y_1), \dots, (X_n, Y_n))$. d - atstumas tarp gretimų orientyrų, R - orientyro signalo perdavimo spindulys, T - laiko periodas, tarp orientyrų skleidžiamų informacinių [*angl. beacon*] signalų, t - laiko intervalas kurio metu imtuve renkami, orientyrų išspinduliuoti duomenys, $N_{sent}(i, t)$ - per laiko intervalą t iš orientyro išsiųstų informacinių signalų skaičius, $N_{recv}(i, t)$ - per laiko intervalą t imtuve gautų informacinių signalų skaičius, CM_i - ryšio metrika tarp objekto ir atitinkamo orientyro R_i , S - informacinių pranešimų kiekis imtuve ryšio metrikai [*angl. sample size for connectivity metric*], $CM_{tresh} - CM$ (ryšio metrikos) slenkstis, (X_{est}, Y_{est}) - algoritmo apskaičiuota imtuvo padėtis. Iš kiekvieno pasiekiamo orientyro surinkta informacija, charakterizuojama ryšio metrika [*angl. connectivity metric*]:

$$CM_i = \frac{N_{recv}(i, t)}{N_{sent}(i, t)} \cdot 100 \quad (2.1)$$

Ryšio metrikos patikimumui pagerinti, nustatomas minimalus reikalingas priimtų signalų skaičius S (parametras kurio vertę galima derinti), t.y $N_{sent}(i, t) = S$. Kadangi mums yra žinomas laiko periodas T , tai imtuvas duomenis renka laiko intervale t :

$$t = (S + 1 - \epsilon) \cdot T, \quad kur \quad (0 < \epsilon \ll 1) \quad (2.2)$$

Laikome, kad orientyrai esantys greta vienas kito yra sinchronizuoti laike, tam kad nebūtų ryškaus vėlavimo tarp išsiųstų signalų laike. Visi orientyrai per laiko periodą T sėkmingai išsiunčia vieną pranešimą. Imtuvas laiko intervalu t renka informacinius pranešimus. Tada iš per laiko intervalą t surinktų pranešimų atrenkami tie orientyrai, kurių ryšio metrika viršija slenkstinę CM_{thres} , pvz. 90%. Tada skaičiuojamas centroidas tarp orientyrų k , kurių ryšio metrika viršijo slenkstinę:

$$(X_{est}, Y_{est}) = \left(\frac{X_{i1} + \dots + X_{ik}}{k}, \frac{Y_{i1} + \dots + Y_{ik}}{k} \right) \quad (2.3)$$

Tačiau kaip paminėta [18, 5] naudojant šią schemą gaunama didelė lokalizacijos paklaida, net atvirose erdvėse, todėl jį naudoti patalpose nerekomenduojama.

2.2. Lokacijos algoritmas naudojantis atstumo vektorius ir šuolius tarp objektų

Naudojant šį padėties nustatymo algoritmą, reikalinga turėti orientyrus, kurių (X, Y) koordinatės yra žinomos. Tada orientyrai vykdo atstumo vektorių apsikeitimą [*angl. distance vector exchange*] - tarp kaimyninių orientyrų apsikeičia atstumais (tarp jų euklidinių koordinatė) ir taip vykdo šį atstumo vektorių apsikeitimą iki tol kol, kiekvienas orientyras žino euklidinius atstumus ir minimalų šuolių skaičių [*angl. hop count*] iki kitų orientyrų. Jei objektų (orientyrų ir objektų su nežinomomis koordinatėmis) bendras skaičius yra didelis, tai galima parametrais riboti, iki kiek artimiausių kaimyninių orientyrų reikia išsiminti. Verta pažymėti, kad kiekvienas orientyras išsaugo euklidinį atstumą tarp jo paties ir kito orientyro ir minimalų šuolių skaičių tarp jo paties ir kito orientyro, bet šuoliai gali vykti ne tik per orientyrus, bet ir per objektus kurių koordinatės iš pradžių nėra žinomos. Tokiu principu kompiuterių tinkluose veikia, atstumo vektorius naudojantys maršrutizavimo protokolai.

Antro šio lokalizacijos algoritmo etapo metu, kai orientyras sužino atstumą iki kito orientyro ir žino minimalų šuolių skaičių iki jo, orientyras apskaičiuoja vidutinį šuolio nuotolį a_i :

$$a_i = \frac{\sum_{i \neq j} \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum_{i \neq j} h_{ij}} \quad (2.4)$$

kur (X_i, Y_i) ir (X_j, Y_j) yra orientyrų koordinatės, o h_{ij} šuolių skaičius tarp jų.

Sekančiu algoritmo etapu kiekvienas orientyras, tik artimiems nelokalizuotiems objektams siunčia, vidutinį apskaičiuotą šuolio nuotolį a_i . Tada objektai, kurių koordinatės nėra žinomos, gavę vidutinį atstumą iki orientyro iš kurio gavo informaciją ir žinodami šuolių skaičių iki jo, apskaičiuoja atstumą d_i iki to orientyro:

$$d_i = a_i \cdot L_i \quad (2.5)$$

kur L_i - šuolių skaičius iki informaciją apie vidutinį šuolio nuotolį suteikusio orientyro.

Objektai gavę tokią informaciją bent iš trijų orientyrų gali naudoti trilateracijos metodus, kuriais apskaičiuojamos jų pačių koordinatės.

Remiantis [10, 5] šis algoritmas tinkamas naudoti patalpose, kur tiek orientyrai, tiek objektai, kurių lokaciją reikia nustatyti yra išsidėstę izotropiškai. Kuo išsidėstymas anizotropiškesnis, tuo šio algoritmo rezultatų paklaidos yra didesnės.

3. Algoritmai tinkami naudoti padėties nustatymui WiFi antspaudų metodu

Šiais laikais WiFi naudojimas patalpose labai išplito. WiFi naudojamas ne tik viešose vietose, bet ir daugelio žmonių namuose. WiFi išplitimą lėmė, tai, kad WiFi yra patikimas, greitas, o perdavimo signalas gali siekti 100 metrų ir daugiau. Taip pat WiFi signalas gali prasiskverbti per sienas, lubas, duris ir t.t. Šios savybės WiFi signalą leidžia išnaudoti vietos nustatymui patalpose.

Padėties nustatymo naudojant WiFi atspaudus veikimo principas: kiekviename patalpos plotelyje (suskirsčius plotą į pavyzdžiui kvadratus) išmatuojamas WiFi signalų stiprumas sklindantis iš visų aplink esančių WiFi įrenginių ir gauti duomenys suvedami į duomenų bazę. Tokiu būdu surenkamas WiFi signalų atspaudas kiekviename patalpos plotelyje. O navigacijos naudotojo įrenginys, taip pat matuoja pastate sklindančius WiFi signalus ir lygina juos su esančiais duomenų bazėje - taip nustatydami navigacijos naudotojo buvimo vietą [23]. Šiame skyriuje aprašyta keletas algoritmų tinkamų naudojami padėties nustatymui pasitelkus WiFi atspaudus.

3.1. Euklido atstumas

Nustatyti padėtį patalpose galima apskaičiuavus Euklido atstumus. Norint tai padaryti reikia sukurti analizuojamos patalpos žemėlapi ir surinkti pradinis WiFi prieigos taškų duomenis. Pirmiausia žemėlapis yra suskirstomas dalimis ir kiekvienoje dalyje fiksuojami WiFi prieigos taškų RSSI ir BSSID duomenys - S_i .

Tada norint nustatyti padėtį analizuojamoje patalpoje, bet kuriame nežinomame taške X , fiksuojami RSSI vektoriai S iki m prieigos taškų. Taip gaunamas rinkinys RSSI vektorių iki m prieigos taškų, taške X : $S = s_j | j \in [1 : m]$ [21]. Tada lyginant S su pradiniais duomenimis $S_i = s_{i,j} | j \in [1 : m]$ skaičiuojamas minimalus Euklido atstumas tarp S ir S_i , pagal 3.1 formulę ir nustatomos taško $X = (x_i, y_i)$ koordinatės.

$$d_{min} = \min \sqrt{\sum_{j=1}^m (S_j - S_{ij})^2} \quad (3.1)$$

Šis algoritmas apskaičiuoja tikslesnes koordinates, kai žemėlapis padalinamas į mažesnes dalis, tačiau to pasekoje padidėja pradinių duomenų kiekis ir Euklido atstumų apskaičiavimo greitis [21].

3.2. Tanimoto koeficientas

Padėties nustatymas gali būti atliktas apskaičiuojant Tanimoto koeficientą tarp dviejų WiFi atspaudų X ir Y . Naudojant Tanimoto koeficientą vietos nustatymui galioja tos pačios sąlygos kaip ir nustatant vietą pagal Euklido atstumus: reikia RSSI vektorių iki WiFi prieigos taškų ir žemėlapiu. Tanimoto koeficientas nustato panašumą tarp X ir Y WiFi signalo atspaudų [9] pagal formulę:

$$C_T(X, Y) = \frac{|RSS_X \cdot RSS_Y|}{|RSS_X| + |RSS_Y| - |RSS_X \cdot RSS_Y|} \quad (3.2)$$

RSS_X - absoliuti X atspaudu signalo reikšmė, atitinkamai RSS_Y - absoliuti Y atspaudu signalo reikšmė [24]. Pasinaudojus (3.2) formule nustatomas C_T koeficientas, kuris parodo atstumą tarp X ir Y atspaudų. Kuo $C_T(X, Y)$ reikšmė didesnė, tuo atstumas tarp X ir Y mažesnis [24].

3.3. K artimiausių kaimynų

Norint nustatyti padėtį šiuo metodu reikalinga turėti N patalpoje išdėstyto WiFi prieigos taškų. Taip pat reikalingas siųstuvas-imituvas, kuris patalpos taške P_n gali matuoti ir siųsti informaciją apie iš N WiFi prieigos taškų priimtų signalų stiprumus (RSSI). Siųstuvas-imituvas gali būti išmanusis telefonas, nes jis turi galimybę fiksuoti WiFi signalus. Patalpos žemėlapis yra suskaidomas į poligonus. Poligonų dydis gali skirtis. Suskirsčius patalpas į poligonus, duomenų bazėje išsaugomos poligonų viršūnių koordinatės ir iš N WiFi prieigos taškų išmatuoti signalų stiprumai kiekvienai viršūnei. Kai siųstuvas-imituvas išsiunčia surinktus apie N WiFi prieigos taškų signalų duomenis, atliekamas prieigos taškų signalų palyginimas su prieš tai surinktais duomenimis. O koordinatės parenkamos, tos viršūnės, kurios WiFi signalų duomenys artimiausi iš siųstuvo-imituvo atsiųstiems WiFi duomenims.

Šis algoritmas skirstomas į dvi būsenas: pasyvią ir aktyvią. Pasyvios būsenos pagrindinė užduotis išgauti ir išsaugoti WiFi prieigos taškų duomenis kiekvienai poligono viršūnei P_n . Kiekvienam poligono viršūnės taškui išsaugomos taško koordinatės ir WiFi prieigos taškų stiprumai $(X_i, Y_i, WIFIS_{i1}, WIFIS_{i2}, WIFIS_{i3}, WIFIS_{i4}, \dots, WIFIS_{iN}, i = 1, 2, 3, \dots, N)$ [9].

Aktyvios būsenos užduotis nustatyti siųstuvo-imituvo buvimo vietą realiu laiku pagal gautus WiFi duomenis. Siųstuvas-imituvas patekęs į analizuojamą patalpą siunčia N grupių WiFi prieigos taškų signalų stiprumus $(WIFIS_{i1}, WIFIS_{i2}, WIFIS_{i3}, WIFIS_{i4}, \dots, WIFIS_{iN}, i = 1, 2, 3, \dots, N)$ [22]. Kiekvienam WiFi prieigos taškui apskaičiuojama vidutinė sklaidžiamo signalo S_j reikšmė, išvedant vidurkį iš siųstuvo-imituvo atsiųstų duomenų. Siųstuvo-imituvo pozicijos nustatymui naudojamas K artimiausių kaimynų algoritmas (KNN).

Padėties nustatymo algoritme naudojama KNN modifikacija, kur K – artimiausi taškai nuo siųstuvo-imituvo atsiųsto signalo, $K = 1..N$. Atstumai d_i nuo siųstuvo siunčiamų WiFi signalų padėties iki P_i taškų apskaičiuojami pagal formulę:

$$d_i = \sqrt[K]{\sum_{j=1}^N |S_j - WIFIS_{ij}|^K} \quad (i = 1, 2, \dots, n, j = 1, 2, \dots, N). \quad (3.3)$$

Apskaičiavus d_i reikšmes parenkami artimiausi P_K taškai. Siųstuvo-imituvo koordinatės apskaičiuojamos išvedus vidurkį iš P_K taškų koordinatėms [22].

3.4. Gauso-Bajeso metodas

Padėties nustatymas naudojant Gauso-Bajeso metodą yra galimas, jeigu matavimai yra pasiskirstę pagal Gausinį (normalųjį) skirstinį. Laikydami, kad WiFi signalo stiprumo matavimų (RSSI) vertės iki vieno WiFi prieigos taško yra pasiskirsčiusios pagal normalųjį skirstinį, tada padėties (mūsų sistemoje poligono) nustatymui galima naudoti Gauso-Bajeso metodą.

Kadangi Gauso skirstinys \mathcal{N} yra aprašomas, dviem parametrais vidurkiu - μ ir dispersija - σ^2 , pirmiausia reikia, nustatyti šiuos parametrus kiekvienam poligone L matomam WiFi prieigos taškui (SSID) $SSID_n$. Parametras μ yra apskaičiuojamas taip, kaip nurodyta 4.6 formulėje. WiFi signalo stiprumų matavimo atveju 4.6 formulėje esantys nariai x reiškia RSSI matavimų vertes, kur j nurodo RSSI matavimų skaičių iki vieno SSID, o n - matomų SSID skaičių viename poligone. Pagal 4.7 kiekvienam, poligone matomam WiFi prieigos taškui, apskaičiuojama RSSI matavimų dispersija σ^2 . Tokiu būdu yra surenkami visų, poligonuose P_k (kur k - poligonų skaičius) matomų,

WiFi prieigos taškų Gauso skirstinių parametrai. Šie parametrai padėties nustatymo sistemoje yra atskaitiniai duomenys.

Turint visų viename poligone P matomų SSID Gauso skirstinių parametrus, pasinaudojus Gauso tikimybės tankio funkcija, pagal 4.8 formulę, galime apskaičiuoti tikimybes g_n [*angl. likelihood*], kad RSSI matavimai yra iš atitinkamų, poligono P Gausianų $\mathcal{N}_n(\mu_n, \sigma_n^2)$. Tokiu būdu, Gauso-Bajeso algoritmo vykdymo pradžioje yra apskaičiuojamos tikimybės g_n . Šios tikimybės parodo kiek tikėtina, kad lokalizavimo metu išmatuoti duomenys $X = x_1, x_2, \dots, x_n$ (x_n - RSSI iki matomų SSID) yra iš atitinkamo poligono. Laikome, kad to paties poligono P lokalizavimo metu atlikti matavimai x_n yra nepriklausomi vieni nuo kitų. Tada bendrą tikimybę G apskaičiuojame sudauginę visas gautas g_n vertes.

$$G_i = \prod_n g_n \quad (3.4)$$

G_i parodo bendrą tikimybę $T(X|P_i)$, kad išmatuoti duomenys X yra iš poligono P_i , kur P_i vienas iš galimų poligonų P_k . Tikimybių G_k apskaičiavimas, visiems galimiems poligonams P_k yra kitas Gauso-Bayeso algoritmo etapas. Galimi poligonai - tai poligonai kuriuose atskaitinių duomenų rinkimo metu buvo matomas nors vienas SSID, iki kurio išmatuota RSSI vertė yra tarp lokalizavimo metu išmatuotų duomenų.

Pagal Bajeso formulę:

$$T(P_i|X) = \frac{T(X|P_i)T(P_i)}{\sum_k T(X|P_k)T(P)} \quad (3.5)$$

Kur $T(P_i)$ - tai pirminė tikimybė [*angl. prior*](tikimybė prieš atliekant stebėjimą X), kad poligonas P_i yra tas, kuriame atlikti lokalizavimo matavimai X . $T(X|P_i)$ - tai tikimybė, kad matavimai X yra atlikti poligone P_i , mūsų atveju, ji yra lygi G_i . $\sum_k T(X|P_k)T(P)$ - tai visų galimų poligonų P_k , tikimybių $T(X|P_k)$ ir $T(P)$ sandaugų suma. $T(P_i|X)$ - tai vėlesnė tikimybė [*angl. posterior*](tikimybė atlikus stebėjimą X), kad poligonas P_i yra tas, kuriame atlikti lokalizavimo matavimai X . Mūsų sistemos atveju, pirminės tikimybės $T(P_i)$ yra lygios visiems galimiems poligonams P_k . Vėlesnių tikimybių $T(P_i|X)$ skaičiavimas kiekvieno galimo poligono atžvilgiu yra kitas Gauso-Bayeso algoritmo etapas.

Paskutinis Gauso-Bajeso algoritmo etapas - poligono P_i , kurio vėlesnioji tikimybė $T(P_i|X)$ yra didžiausia, suradimas. Tokiu principu veikia ir padėties nustatymo sistemos algoritmas naudojamas [3] darbe.

4. Aukšto nustatymo algoritmai

Padėties nustatymo patalpose problemą galima skaidyti į dvi: aukšto nustatymo pastate ir tikslios padėties nustatymo. Padėties nustatymo algoritmai skirti tikslios padėties nustatymo problemos sprendimui gali veikti žymiai prasčiau, kai padėties nustatymas atliekamas patalpoje sudarytoje iš vieno aukšto nei iš kelių. Taigi, jei pirmiausia pavyktų nustatyti tikslų aukštą ir tik tada būtų pradėtas vykdyti tikslios padėties nustatymo algoritmas, padėties nustatymas veiktų greičiau ir tiksliau, kadangi būtų sumažintas plotas kuriame reikia atrasti tikslią poziciją. Yra atliktų bandymų nustatyti aukštą remiantis GSM signalo stiprumu naudojant KNN algoritmą (K artimiausių kaimynų su mažiausiais Euklido atstumais) [11]. Šiame darbe žemiau pateikti aukšto nustatymo algoritmai, kuriuos galima įgyvendinti naudojant tik išmaniuosius telefonus.

4.1. BCFI algoritmas

BCFI [*angl. Bayesian classification based floor identification algorithm*] metodas remiasi WiFi atspaudų rodmenimis. Šis algoritmas skiriasi nuo kitų algoritmų, kurie naudoja WiFi atspaudus tuo, kad WiFi atspaudus naudoja nustatyti dabartinį aukštą, o ne tikslią vietą pastate.

BCFI metodas nenaudoja WiFi signalo stiprumo rodmenų. WiFi signalo rodmenys iškraipomi kai duomenys renkami tarp skirtingų aukštų. Taip nutinka, todėl, kad, WiFi signalas silpnėja, kai reikia prasiskverbti pro kieta paviršių kaip lubos ar grindys. Šis metodas pranašesnis už metodus, kurie remiasi tik WiFi signalų stiprumais, tuo, kad gali nustatyti padėtį kelių aukštų pastate žinant kurie WiFi prieigos taškai matomi tam tikrame aukšte.

BCFI metodas remiasi Bajeso įverčiu apskaičiuojant tikimybę, kuriame aukšte yra vartotojas. Aukštas su didžiausiu Bajeso įverčiu laikomas aukštu, kuriame yra vartotojas. BCFI metodas susideda iš dviejų etapų. Pirmame etape apmokoma sistema, surenkami pradiniai WiFi prieigos taškų duomenys (MAC adresai, atitinkamo aukšto numeris) ir apskaičiuojama kiekvieno prieigos taško tikimybė kiekviename aukšte. Kai gaunami nauji WiFi rodmenys apskaičiuojamas Bajeso įvertis.

Tarkime $L = l_1, l_2, \dots, l_n$ analizuojamų patalpų aukštas. $Z = Z_1, Z_2, \dots, Z_p$ matomi WiFi prieigos taškai pastate. $O = o_1, o_2, \dots, o_j$ rinkinys sudarytas iš k stebėjimų, kur $o_j = z_1, z_2, \dots, z_m$ sudaryta iš m WiFi prieigos taškų. Apskaičiuojama tikimybė $P(l_p | o_j)$, kad stebėjimo duomenis o_j priklausau aukštui l_p pagal formulę:

$$P(l_p | o_j) = \frac{P(o_j | l_p) \cdot P(l_p)}{\sum_{p=1}^n P(o_j | l_p) \cdot P(l_p)} \quad (4.1)$$

Šioje formulėje $P(l_p)$ tikimybė yra lygi stebėjimo kartu atliktų aukšte l_p padalinus iš visų stebėjimų atliktų pastate. Darant prielaidą, jog WiFi prieigos taškai vienas nuo kito nepriklauso, tuomet tikimybė $P(o_j | l_p)$ yra visų WiFi prieigos taškų užfiksuotų stebėjimuose $o_j = z_1, z_2, \dots, z_m$ ($z_k \in Z$) tikimybių $P(z_k | l_p)$ sandauga. $P(z_k | l_p)$, tai tikimybė, kad WiFi prieigos taškas z_k gali būti užfiksuotas aukšte l_p , kitaip tariant tikimybė $P(z_k | l_p)$ lygi apmokymo duomenų skaičiui, kuriuose užfiksuotas prieigos taškas z_k padalintas iš visų apmokymo duomenų užfiksuotų aukšte l_p [11]. $P(z_k | l_p)$ tikimybės apskaičiuojamos pagal formulę:

$$P(o_j | l_p) = \prod_{K=1}^m P(Z_k | l_p) \quad (4.2)$$

Kai atliekamas naujas WiFi prieigos taškų skenavimas, tuomet pasinaudojus (4.1) formule apskaičiuojamos $P(l_p|o_j)$ tikimybės kiekvienam aukštui. Aukštas su didžiausia $P(l_p|o_j)$ tikimybe laikomas nustatytu aukštu. Šis aukšto nustatymo metodas geriausius rezultatus pasiektų geriausius rezultatus pastatuose su daug aukštų, kadangi šis algoritmas remiasi WiFi prieigos taškų matavimu. Jei pastatas turės mažai auštų, tikėtina, kad rezultatai bus prastesni, nes Tie patys prieigos taškai bus matomi visame pastate.

4.2. Intervalų algoritmas

Intervalų metodas remiasi barometro vidutinėmis aukščio virš jūros lygio (toliau tekste bus vartojamas terminas - altitudė) reikšmėmis ir altitudės reikšmių svyravimais tarp sekančių aukštų. Algoritmas apmokomas surenkant pradinės altitudės reikšmes kiekviename aukšte $L = l_1, l_2, \dots, l_n$, tuomet apskaičiuojama vidutinė altitudės reikšmė $VID = vid_1, vid_2, \dots, vid_n$ kiekvienam aukštui l_n ir altitudės reikšmių svyravimai tarp sekančių aukštų $S = s_1, s_2, \dots, s_{n-1}$.

Pradinės apmokymo altitudės reikšmės $A = a_1, a_2, \dots, a_n$ kur $a_n = o_{n1}, o_{n2}, \dots, o_{nj}$ renkamos kiekvienam aukštui po vienodą kiekį j pavyzdžių. Tuomet kiekvienam aukštui l_n apskaičiuojama vidutinė altitudės reikšmė vid_n pagal formulę:

$$vid_n = \frac{\sum_{i=1}^j o_{ni}}{j} \quad (4.3)$$

Apskaičiuavus vidutinės altitudės reikšmes vid_n kiekvienam aukštui l_n nustatomi altitudės reikšmių svyravimai s_{n-1} tarp sekančių aukštų. Altitudės reikšmių svyravimai nustatomi pagal formulę:

$$s_{n-1} = vid_{n+1} - vid_n \quad (4.4)$$

Kai turime altitudės reikšmių svyravimus s_{n-1} galime nustatyti intervalus kiekvienam aukštui l_n į kuriuos patenka altitudės reikšmės. Intervalus $V = v_1, v_2, \dots, v_n$ galime išreikšti:

$$v_1 \leq vid_1 + \frac{s_1}{2}, vid_1 + \frac{s_1}{2} < v_2 \leq vid_2 + \frac{s_2}{2}, \dots, v_n > vid_{n-1} + \frac{s_1}{2} \quad (4.5)$$

Užfiksavus naują barometro altitudės reikšmę b tikrinama, kuriam intervavui v_n priklauso reikšmė b . Nustačius, kad reikšmė b priklauso intervalui v_n laikoma kad nustatytas aukštas yra l_n . Šis algoritmas remiasi absoliučiomis altitudės reikšmėmis, todėl aukšto nustatymo rezultatai gali priklausyti nuo įrenginių, kuriais surenkami barometro duomenys, taip pat nuo oro sąlygų.

4.3. Aukšto nustatymas naudojant Gauso tikimybių tankio funkciją

Aukšto nustatymo metodas naudojantis Gauso tikimybių tankio funkciją [angl. *Gaussian probability density function*] taip pat naudoja barometru gautas altitudės reikšmes. Taip pat šiam metodui reikia surinkti apmokymo duomenų kiekvienam analizuojamo pastato aukštui. Šis metodas apskaičiuoja tikimybes, kad matavimas priklauso Gauso skirstiniui \mathcal{N} , kurio tikimybių tankio funkcija lygi g_n . Pažymėkime tikimybes: $G = g_1, g_2, \dots, g_n$ kiekvienam aukštui $L = l_1, l_2, \dots, l_n$. Išrinkus didžiausią tikimybių reikšmę: $\max(g_1, g_2, \dots, g_n)$ laikoma, kad nustatytas aukštas l_n .

Apskaičiuojant Gauso tikimybių tankio funkcijos vertę, pirmiausiai reikia apskaičiuoti dispersiją σ_n^2 ir vidutinę altitudės reikšmę μ_n kiekvienam aukštui l_n . Tarkime, kad pradinės apmokymo

altitudės reikšmės $A = a_1, a_2, \dots, a_n$ kur $a_n = x_{n1}, x_{n2}, \dots, x_{nj}$ renkamos kiekvienam aukštui po vienodą kiekį j , $j > 0$. Tuomet kiekvienam aukštui l_n apskaičiuojama vidutinė altitudės reikšmė μ_n pagal formulę:

$$\mu_n = \frac{\sum_{i=1}^j x_{ni}}{j} \quad (4.6)$$

Dispersija σ_n^2 kiekvienam aukštui l_n apskaičiuojama pagal formulę:

$$\sigma_n^2 = \frac{1}{n-1} \cdot \sum_{i=1}^j (x_{ni} - \mu_n)^2 \quad (4.7)$$

Kai atliekamas stebėjimas ir gaunama altitudės reikšmė b apskaičiuojama tikimybė pagal Gauso tikimybės tankio funkciją g_n kiekvienam aukštui l_n pagal formulę:

$$g_n = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(b - \mu_n)^2}{2\sigma^2}\right) \quad (4.8)$$

Apskaičiavus g_n reikšmes, kiekvienam aukštui l_n parenkama didžiausia g_n reikšmė ir laikoma, kad nustatytas aukštas l_n . Šis algoritmas taip pat remiasi absoliučiomis altitudės reikšmėmis, todėl aukšto nustatymo rezultatai gali labai skirtis, jei tarkim apmokymo altitudžių reikšmės buvo labai panašios, tada σ^2 reikšmė bus maža ir jei atliekant naują stebėjimą bus gauta altitudės reikšmė nukrypusi nuo μ , tai Gauso tikimybės tankio funkcijos vertė g mažės, tokiu atveju galima daryti prielaidą, kad aukšto nustatymo rezultatai prastės, kadangi šis algoritmas parenka aukštą pagal, tai kurio aukšto l_n tikimybė [*angl. likelihood*] g_n yra didžiausia.

5. Matavimų kalibravimas

Šio darbo metu sukurta prototipinė padėties nustatymo sistema gali nustatyti padėtį patalpose, kuriose buvo surinkti pradiniai apmokymo duomenys. Tačiau rodmenys surinkti patalpose gali labai skirtis praėjus dienai ar renkant duomenis su kitu įrenginiu (žr. 33 pav.). Be to, signalo intensyvumo reikšmės gautos priklauso nuo įrenginyje naudojamo įrenginio lustų bei antenų, ir gali labai skirtis tarp skirtingų 802.11 realizacijų [11]. Todėl, reikalingas metodas galintis gautus matavimus suvienodinti su pradiniais matavimais.

Šiame darbe atliekant eksperimentus (žr. 33 pav.), taip pat nagrinėjant literatūros šaltinius pastebėta, kad pokyčiai aplinkoje, įskaitant laiko ir skirtingas įrenginių realizacijas, yra tiesiškai priklausomi. Kadangi, žemėlapis sugeneruotas iš pradinių apmokymo duomenų gali būti pritaikomas suradus dvi komponentes c_1 ir c_2 , tai matavimų kalibravimo problemą galime išspręsti suradę kalibracijos funkciją c , kuri sujungia pastebėto signalo stiprumo reikšmę i su c_i reikšme, gausime signalo stiprumo reikšmę panašią į tą, kuri buvo gauta naudojant įrenginį renkant pradinius apmokymo duomenis lokalizacijos žemėlapiui. Jeigu c žinoma, tai c_i reikšmė gali būti duota lokacijos nustatymo algoritmui ir sugeneruotas lokalizacijos žemėlapis gali būti naudojamas nepriklausomai nuo aplinkos pokyčių [11]. Taigi c_i reikšmė sugeneruojama pasinaudojus tiesės funkcija:

$$c_i = c_1 \cdot i + c_2 \quad (5.1)$$

Nežinomas komponentes c_1 ir c_2 galima surasti naudojantis mažiausių kvadratų metodu [*angl. Least-squares method*]. [11]

5.1. Mažiausių kvadratų metodas

Mažiausių kvadratų metodo esmė nubrėžti tiesią tarp taškų taip, kad iš taškų atstumų iki linijos nubrėžtų kvadratų suma būtų pati mažiausia. Tarkime, kad turime $Y = y_1, y_2, \dots, y_n$ signalo stiprumo taškai surinkti kuriant lokalizacijos žemėlapi. $X = x_1, x_2, \dots, x_n$ signalo stiprumo taškai surinkti naudojant kitą įrenginį. Tarp signalo stiprumo taškų X ir Y yra tiesinė priklausomybė:

$$Y = c_1 \cdot X + c_2 \quad (5.2)$$

Tuomet nežinomas komponentė c_1 apskaičiuojamas pagal formulę:

$$c_1 = \frac{n \cdot \sum (X \cdot Y) - \sum(X) \cdot \sum(Y)}{n \cdot (\sum X^2) - (\sum X)^2} \quad (5.3)$$

Nežinomas komponentė c_2 apskaičiuojamas pagal formulę:

$$c_2 = \frac{(\sum Y) - c_1 \cdot (\sum X)}{n} \quad (5.4)$$

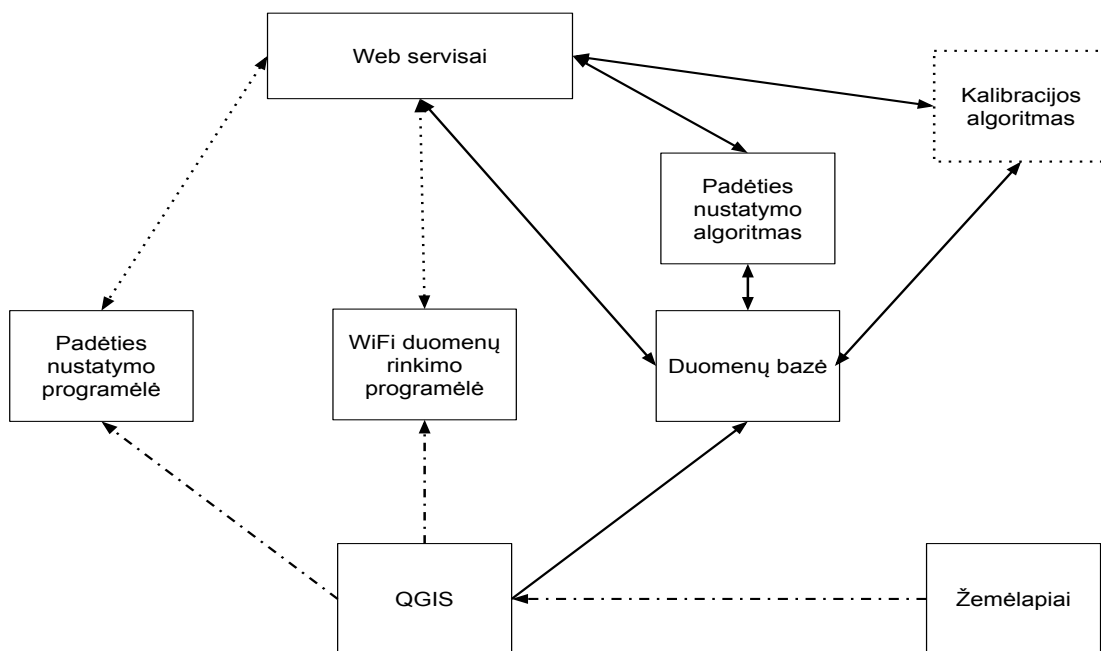
Suradus c_1 ir c_2 komponentes galime sukalibruoti naujas z_1, z_2, \dots, z_m reikšmes, kurios būtų panašios į x_1, x_2, \dots, x_n reikšmes. Šio darbo metu, mažiausių kvadratų metodas pritaikytas WiFi ir barometro rodmenims kalibruoti.

6. Prototipinės padėties nustatymo sistemos struktūra

Naudojant WiFi signalo stiprumo ir kitų matavimų atspaudus padėties nustatymui patalpose, išvengiama vienos didžiausių problemų, kuri kyla norint lokalizuoti objektus remiantis WiFi ar kitų signalų stiprumo matavimu, iš kurio skaičiuojamas atstumas tarp signalą skleidusio ir signalą priėmusio įrenginių. Iš RSSI matavimo apskaičiuoto atstumo naudojimas padėties nustatymo algoritmuose neduoda tinkamų rezultatų todėl, kad RSSI vertė labai priklauso nuo to, ar tarp siųstuvo ir imtuvo yra tiesioginis matomumas, bei nuo to ar dėl interferencijos tarp iš šaltinio skleidžiamo signalo ir jo atspindžių nuo aplinkos, vyksta signalo stiprinimas ar slopinimas. [17]. Todėl naudojant iškraipytus RSSI matavimų duomenis apskaičiuoti atstumai yra su didelėmis paklaidomis, o to pasekoje ir objekto padėtis yra nustatoma su didele paklaida. Tačiau yra naudojamos hibridinės lokalizacijos schemos, naudojant RSSI matavimus tiek atspaudų tiek atstumų nustatymui vienu metu. [21]

Dar vienas WiFi atspaudų matavimo metodo privalumas yra tas, kad daugumoje vietų jau būna įdiegti WiFi prieigos taškai, todėl lyginant su 1.2 skyriuje aprašytais matavimo metodais, nereikia tiek daug papildomos infrastruktūros. Jei naudojamas serveris, jį galima virtualizuoti, taip apsieinant be papildomos infrastruktūros. [4].

Palyginus vietos nustatymo metodus ir atsižvelgus į WiFi atspaudų teikiamus privalumus, tyrimą buvo nuspręsta tęsti naudojant WiFi atspaudų algoritmus. Žemiau pateikiama tyrimo metu kuriamos prototipinės padėties nustatymo sistemos blokinė schema:



10 pav. Padėties nustatymo sistemos blokinė schema

Tyrimo metu kuriamą sistemą galima skaidyti į tris dalis: mobiliųjų programėlių, virtualizuoto serverio ir žemėlapių apdorojimo. Mobilųjų programėlių dalis:

- Padėties nustatymo programėlė;
- WiFi duomenų rinkimo programėlė;

Virtualizuoto serverio dalis:

- Duomenų bazė;
- Web servais;
- Padėties nustatymo algoritmas;
- Kalibravimo algoritmas;

Žemėlapių apdorojimo dalis:

- Žemėlapiai;
- QGIS;

Mobiliųjų programėlių dalis atsakinga už WiFi atspaudų duomenų surinkimą, išsaugojimą lokaliaje duomenų bazėje ir duomenų perdavimą virtualizuotam serveriui.

Virtualizuoto serverio dalis atsakinga už WiFi atspaudų duomenų išsaugojimą ir padėties nustatymo algoritmo panaudojimą. Jeigu padėties nustatymo programėlė naudojama kitame įrenginyje, o ne įrenginyje su kuriuo surinkti pradiniai WiFi duomenys, tai ši dalis yra atsakinga ir už kalibravimo algoritmo panaudojimą.

Žemėlapių dalis atsakinga už pastato aukštų žemėlapių apdorojimą ir pateikimą į mobiliąsias programėles bei serverio duomenų bazę. Šiame darbe dalis žemėlapių apdorojimo vykdoma rankiniu būdu.

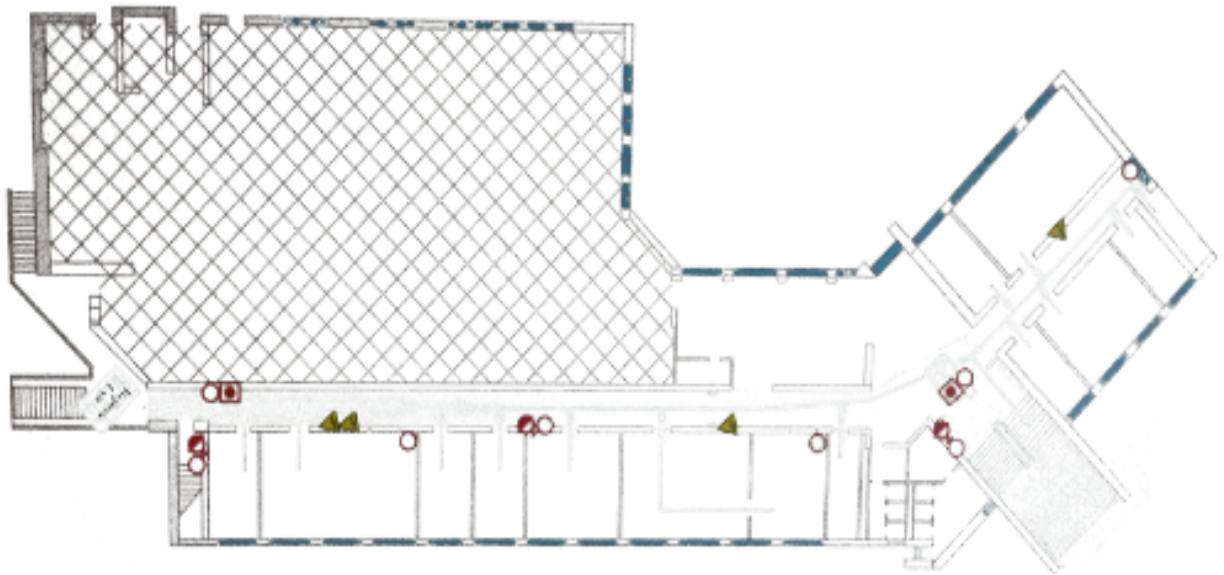
Brūkšniuotos-taškuotos linijos 10 paveiksle nurodo sąryšius, tarp sistemos dalių, kurie vykdomi rankiniu būdu. Taškuotos linijos nurodo sąryšius, kurie vykdomi Web servais. Vientisos linijos nurodo sąryšius, kurie vykdomi serverio viduje.

Padėties nustatymo sistemos veikimo principas. Iš pradžių WiFi duomenų surinkimo programėlė surenka pradinius WiFi rodmenis žinomuose žemėlapio poligonuose. Surinkti rodmenys išsaugomi lokaliaje programėlės duomenų bazėje bei siunčiami virtualizuotam serveriui. Serveris rodmenis išsaugo duomenų bazėje. Tuomet vietos nustatymo programėlė iš nežinomo žemėlapio taško fiksuoja WiFi rodmenis. Siunčia juos į serverį, o serveryje yra vykdomas padėties nustatymo algoritmas, kurio rezultatas yra WiFi rodmenų siuntėjo lokacija.

Detalesnis padėties nustatymo sistemos dalių aprašymas pateiktas kituose skyriuose.

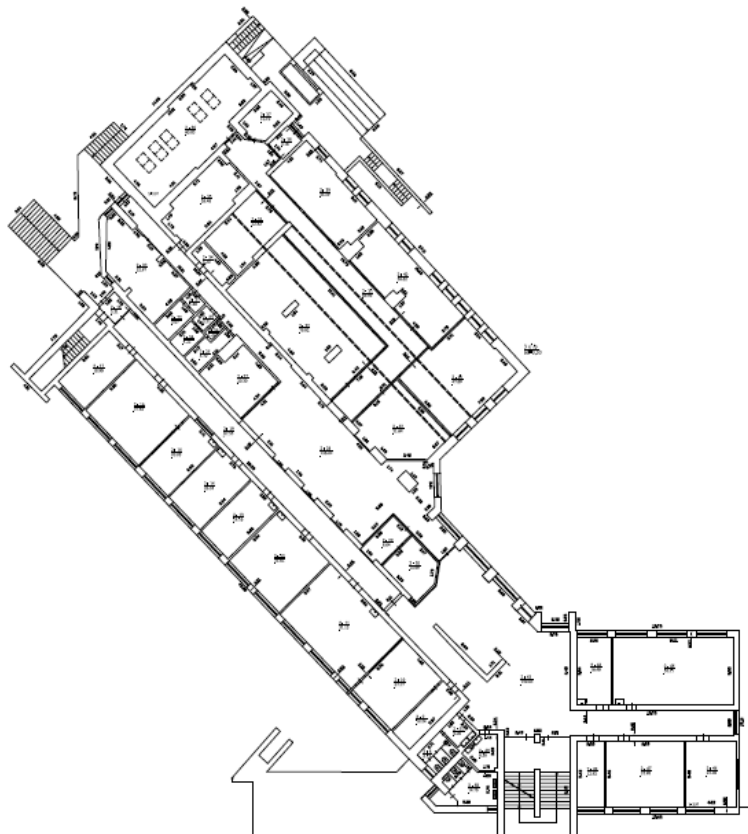
6.1. Žemėlapis

Kadangi kuriama sistema padėties nustatymui patalpose, tai naudojama ne geografinė koordinatinių sistema, o projekcinė, metrinė koordinatinių sistema EPSG:3346 (LKS94/Lithuania TM).



11 pav. MIF Didlaukio g.47 3a evakuacijos planas

Šiame tiriamajame darbe naudojama Vilniaus Universiteto infrastruktūra (WiFi prieigos taškai) esanti Matematikos ir Informatikos fakultete Didlaukyje, todėl naudojami žemėlapiai - Matematikos ir Informatikos fakulteto planai. Pirmuosiuose mobiliųjų aplikacijų variantuose buvo naudojamas 3 aukšto evakuacijos planas, pateiktas 11 paveiksle. Galutiniame padėties nustatymo sistemos variante naudojami kadastrinių matavimų aukštų planai, 12 paveiksle pateiktas naudojamas 3 aukšto planas.

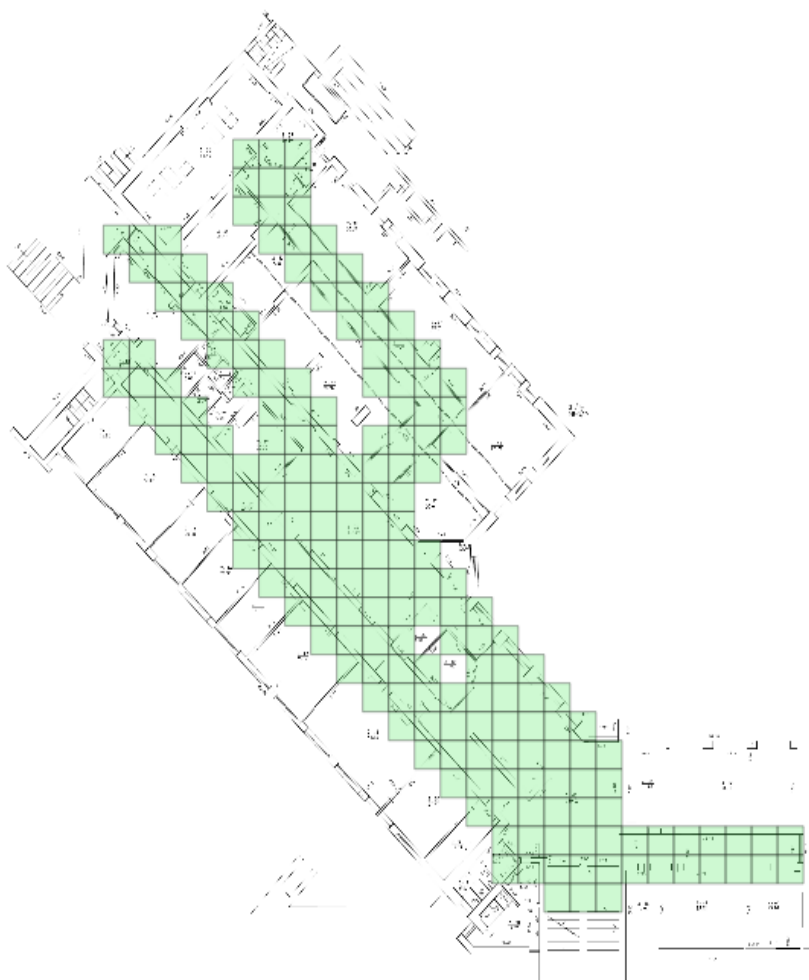


12 pav. MIF Didlaukio g.47 3a kadastrinių matavimų aukšto planas

6.2. QGIS

QGIS - viena iš labiausiai išvystytų programinių įrangų paketų darbu su geografinėmis informacinėmis sistemomis. Tai atviro kodo programinė įranga, todėl visos jos funkcijos yra visada visiems prieinamos, skirtingai nei Esri ArcGIS programinėje įrangoje, kuri turi kelis licenzijavimo lygius.

Šiame darbe sukurtas qgis projektas, kuriame buvo naudojamas QGIS geografinių nuorodų sudarymo įrankis [*angl. georeferencing tool*], juo prie žemėlapių pririšama EPSG:3346 (LKS94 /Lithuania TM) koordinatų sistema ir sukuriama rastriniai geografinės informacinės sistemos sluoksniai. Tada žemėlapių koridoriai buvo skaitmenizuoti (kadangi mes šiame darbe apsiribojame padėties nustatymu koridoriuose) ir padalinti poligonais - taip sukuriama aukštų vektoriniai geografinės informacinės sistemos sluoksniai. Naudojantis QGIS "DBmanager" įrankiu, vektoriniai aukštų sluoksniai buvo įkelti į padėties nustatymo sistemos serveryje esančią PostgreSQL duomenų bazę. Rastriniai sluoksniai eksportuoti *.tiff* formatu, o vektoriniai sluoksniai *.shp* formatu. Eksportuoti rastriniai ir vektoriniai sluoksniai yra naudojami padėties nustatymo sistemos mobiliosiose programėlėse.

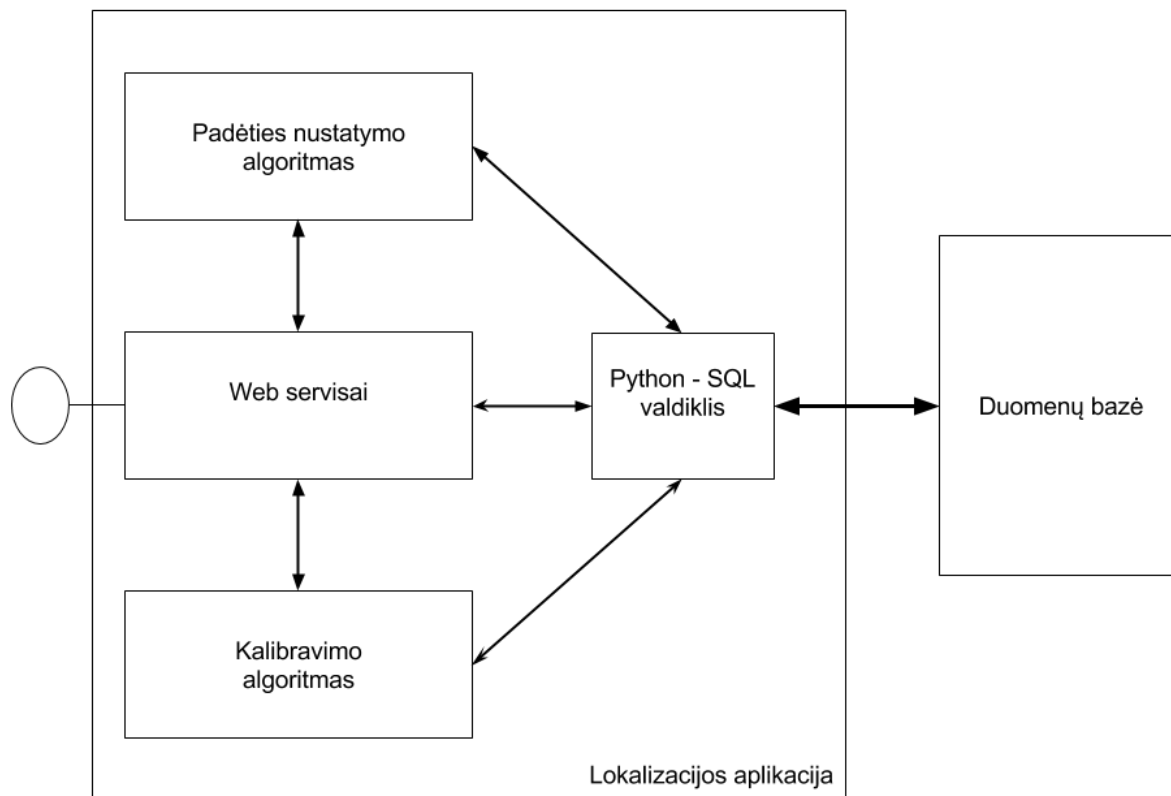


13 pav. MIF Didlaukio g.47 3a rastrinis ir vektorinis sluoksniai

Rastrinis ir jį atitinkantis vektorinis sluoksniai pateikti 13 paveiksle. Sluoksnis sudarytas iš žalių poligonų - vektorinis, fono [*angl. background*] sluoksnis - rastrinis.

7. Prototipinės sistemos serverio dalis

Padėties nustatymo sistemos serverio dalis yra realizuota virtualizuotame serveryje. Jame yra įdiegta Linux Ubuntu 14.04.3 LTS operacinė sistema.



14 pav. Padėties nustatymo sistemos serverio dalies blokinė schema

Kaip matome iš 14 paveikslo, padėties nustatymo sistemos serverio dalis yra sudaryta iš dviejų pagrindinių dalių: lokalizacijos aplikacijos ir duomenų bazės. Toliau darbe pateikiamos atskirai aprašytos lokalizacijos aplikacijos dalys ir duomenų bazė.

7.1. Duomenų bazė

Virtualizuotame serveryje įdiegta PostgreSQL duomenų bazė, ji parinkta todėl, kad turi PostGIS plėtinį, kuris leidžia duomenų bazėje disponuoti geografine informacija. Tai viena svarbiausių padėties nustatymo sistemos komponentų, kurioje saugomi atskaitiniai WiFi matavimų, poligonų duomenys. Pagal kuriuos yra vykdoma įrenginio naudojančio padėties nustatymo programėlė lokalizacija.



15 pav. Padėties nustatymo sistemos serverio duomenų bazės struktūra

Kaip matome 15 paveiksle, serveryje esanti duomenų bazė yra sudaryta iš 6 lentelių. *a2_shp*, *a3_shp* ir *a4_shp* lentelės, tai atitinkamai iš antro, trečio ir ketvirto aukštų vektorinių sluoksnių sugeneruotos lentelės. Šioje prototipinėje padėties nustatymo sistemoje naudojame 3 pastato aukštus, pakankamai, kad galėtumėme įvertinti ar padėties nustatymo algoritmai, nustato teisingą poligoną, o tuo pačiu ir pastato aukštą. 15 paveiksle matome, kad tarp *a2_shp*, *a3_shp* ir *a4_shp* nėra sąryšių, bet iš šių lentelių duomenų yra sugeneruota *polygons* lentelė. Į šią lentelę yra įtrauktos visos galimos unikalios, atributų *polygon_id* ir *floor* poros iš *a2_shp*, *a3_shp* ir *a4_shp* lentelių. 15 paveiksle matomos *polygondata* ir *polygonwifidata* lentelės yra skirtos saugoti atskaitiniams (surinktiems su WiFi duomenų rinkimo programėle) duomenims. Atsiųsti duomenys yra įrašomi tik tada, jei matavime nurodyta *polygon_id* ir *floor* verčių pora egzistuoja lentelėje *polygons*. Duomenų bazės struktūros schemeje pavaizduoti sąryšis tarp *polygons* ir *polygondata* parodo, kad *polygons* lentelėje gali egzistuoti tik vienas unikalus *polygon_id* ir *floor* įrašas, o į *polygondata* lentelę galima įrašyti tik tokius matavimus, kurių *pol_id* egzistuoja *polygons* lentelėje. Sąryšis tarp *polygondata* ir *polygonwifidata* lentelių parodo, kad vienam poligonui (*id*) gali egzistuoti daug WiFi matavimų (*ssid*, *miu*, *sigma_2*).

7.2. Web servaisi

Darbe naudojamas pagal reprezentacinės būsenos perdavimo interneto paslaugų modelį [angl. *Representational State Transfer (REST) web service*] sukurtas servisas, jame implementuoti GET,

POST, DELETE metodai. Duomenų perdavimui tarp programėlės ir serverio naudojamas JSON formatas.

7.2.1. Web servisų kūrimas

Serverio dalis. Serveryje Web servisiui implementuoti python programavimo kalba. Web servisų pagrindui naudojamas Flask karkasas bei flask_restful biblioteka. Sąryšiui tarp postgresql duomenų bazės ir web servisų naudojamas psycopg2 adapteris, sqlalchemy bei flask_sqlalchemy bibliotekos. JSON formato bei schemas tikrinimui naudojamos marshmallow bei marshmallow_jsonapi bibliotekos.

7.2.2. GET GetWifiData

Tai web servisas skirtas gauti visiems į duomenų bazę įrašytiems WiFi matavimams. Įvykdžius šią užklausą, JSON formatu gražinamas masyvas su visais WiFi matavimų duomenimis esančiais duomenų bazėje. Žemiau pateikiama lentelė, kurioje aprašyta šio masyvo elementų struktūra.

1 lentelė. GetWifiData web serviso atsakymo masyvo elementų struktūra

Parametro pavadinimas	Tipas	Aprašymas
id	Sveikas skaičius	Įrašo id duomenų bazės lentelėje (polygonwifidata).
polygons_id	Sveikas skaičius	Poligoną identifikuojantis id duomenų bazės lentelėje (polygondata).
ssid	Eilutė	WiFi prieigos taško MAC adreso ir WiFi prieigos taško pavadinimo junginys.
miu	Slankiojo kablelio skaičius	Matavimo metu surinktų RSSI verčių vidurkis.
sigma_2	Slankiojo kablelio skaičius	Matavimo metu surinktų RSSI verčių dispersija.

7.2.3. POST SaveWifiData

Tai web servisas skirtas pradiniam WiFi duomenims išsaugoti serveryje patalpintoje duomenų bazėje. Šis web servisas naudoja WiFi duomenų surinkimo programėlę. Žemiau pateikiama *SaveWifiData* web serviso užklausos parametrų lentelė.

2 lentelė. *SaveWifiData* web serviso užklausos parametrai

Parametro pavadinimas	Tipas	Aprašymas
polygon_id	Sveikas skaičius	Žemėlapiu poligoną identifikuojantis parametras.
floor	Sveikas skaičius	Pastato aukštą identifikuojantis parametras.
altitude	Slankiojo kablelio skaičius	Aukštis nuo jūros lygio (Barometro matavimas).
polygonData	Masyvas	Masyvas sudarytas iš poligone surinktų WiFi duomenų.

PolygonData masyvas sudarytas iš matavimų metu, poligone surinktų WiFi duomenų. Žemiau pateikiama lentelė, kurioje aprašyta *PolygonData* masyvo elementų struktūra.

3 lentelė. *PolygonData* masyvo elementų struktūra

Parametro pavadinimas	Tipas	Aprašymas
ssid	Eilutė	WiFi prieigos taško MAC adreso ir WiFi prieigos taško pavadinimo junginys.
miu	Slankiojo kablelio skaičius	Matavimo metu surinktų RSSI verčių vidurkis.
sigma_2	Slankiojo kablelio skaičius	Matavimo metu surinktų RSSI verčių dispersija.

Naudojant *SaveWifiData* web servisą svarbu žinoti ar buvo sėkmingai išsaugoti perduodami duomenys. Todėl reikalingas užklausos į serverį atsakymo apdorojimas mobiliojoje programėlėje. Jeigu duomenys serveryje išsaugomi, tai gražinamas HTTP kodas 201, jei duomenys neišsaugomi ir gražinamas HTTP kodas 403, tai reiškia *SaveWifiData* web serviso vykdymas buvo nutrauktas dėl netinkamo POST metu siunčiamo duomenų formato arba dėl klaidos įrašant duomenis į duomenų bazę. Jeigu web serviso vykdymas buvo nutrauktas dėl kitokių priežasčių, gražinamas atitinkamas HTTP klaidos kodas.

7.2.4. DELETE DeleteWifiData

Tai web servisas skirtas trinti WiFi duomenims esantiems serverio duomenų bazėje. Ši web servisą naudoja WiFi duomenų surinkimo programėlė, kai norima ištrinti kurio nors poligono matavimus. Sėkmingai įvykdžius šią užklausą, gražinamas HTTP kodas 204, jeigu resursas kurį norima ištrinti neegzistuoja gražinamas HTTP kodas 404, jeigu įvyko klaida duomenų bazėje gražinamas HTTP kodas 401.

7.2.5. POST GetPosition

Šis web servisas skirtas nustatyti buvimo vietą patalpose lyginant realiu laiku užfiksuotus WiFi duomenis su pradiniais WiFi duomenimis, kurie buvo surinkti naudojant WiFi duomenų rinkimo programą. Ši web servisą naudoja padėties nustatymo programėlė. Žemiau pateikiama *GetPosition* web serviso užklausos parametru lentelė.

4 lentelė. *GetPosition* web serviso užklausos parametrai

Parametro pavadinimas	Tipas	Aprašymas
altitude	Slankiojo kablelio skaičius	Aukštis nuo jūros lygio (Barometro matavimas).
WifiData	Masyvas	Masyvas sudarytas iš vieno matavimo metu surinktų WiFi duomenų

WifiData masyvas sudarytas iš matavimo metu, surinktų WiFi duomenų. Žemiau pateikiama lentelė, kurioje aprašyta *WifiData* masyvo elementų struktūra.

5 lentelė. *WifiData* masyvo elementų struktūra

Parametro pavadinimas	Tipas	Aprašymas
ssid	Eilutė	WiFi prieigos taško MAC adreso ir WiFi prieigos taško pavadinimo junginys.
rssi	Sveikas skaičius	Matavimo metu išmatuota RSSI vertė.

Atsakymas į POST *GetPosition* užklausą yra lokalizacijos algoritmo metu nustatyto poligono identifikacijos parametrai pateikti JSON formatu. Žemiau pateikiama lentelė, kurioje aprašyta *GetPosition* web serviso užklausos atsakymo elementų struktūra.

6 lentelė. *GetPosition* web serviso užklausos atsakymo struktūra

Parametro pavadinimas	Tipas	Aprašymas
polygon_id	Sveikas skaičius	Žemėlapio poligoną identifikuojantis parametras.
floor	Sveikas skaičius	Pastato aukštą identifikuojantis parametras.

Jeigu *GetPosition* web servisas grąžina HTTP kodą 403, reiškia *GetPosition* web serviso vykdymas buvo nutrauktas dėl netinkamo POST metu siunčiamo duomenų formato. Jeigu web serviso vykdymas buvo nutrauktas dėl kitokių priežasčių, grąžinamas atitinkamas HTTP klaidos kodas.

7.2.6. POST Calibration

Šis web servisas skirtas gauti WiFi matavimų kalibravimo koeficientus. Kalibravimo koeficientai skaičiuojami tarp WiFi duomenų esančių serveryje ir WiFi duomenų naudojančių padėties nustatymo programėlę. Kalibravimas reikalingas todėl, kad skiriasi skirtingų įrenginių antenų stiprinimas, o taip pat skiriasi wifi moduliai ir jų jautrumas. Žemiau pateikiama *Calibration* web serviso užklausos parametrai lentelė.

7 lentelė. *Calibration* web serviso užklausos parametrai

Parametro pavadinimas	Tipas	Aprašymas
polygon_id	Sveikas skaičius	Žemėlapio poligoną identifikuojantis parametras.
floor	Sveikas skaičius	Pastato aukštą identifikuojantis parametras.
polygonData	Masyvas	Masyvas sudarytas iš vieno matavimo metu surinktų WiFi duomenų

Šio *PolygonData* masyvo elementų struktūra tokia pati, kaip *WifiData* masyvo elementų struktūra (žr. 5 lentelę). Žemiau pateikiama lentelė, kurioje aprašyta *Calibration* web serviso užklausos atsakymo elementų struktūra.

8 lentelė. *Calibration* web serviso užklauso atsakymo struktūra

Parametro pavadinimas	Tipas	Aprašymas
m	Slankiojo kablelio skaičius	Krypties koeficientas m ($y = mx + c$).
c	Slankiojo kablelio skaičius	Pastovusis narys c ($y = mx + c$).

Jeigu *Calibration* web servisas grąžina HTTP kodą 403, reiškia *Calibration* web serviso vykdymas buvo nutrauktas dėl netinkamo POST metu siunčiamo duomenų formato. Jeigu web serviso vykdymas buvo nutrauktas dėl kitokių priežasčių, grąžinamas atitinkamas HTTP klaidos kodas.

7.3. Lokalizacijos aplikacija

Šiame poskyryje aprašytos likusios nepaminėtos lokalizacijos aplikacijos dalys (žr. 14 pav.): Padėties nustatymo algoritmo dalis, Python-SQL valdiklio dalis ir kalibravimo algoritmo dalis.

Python-SQL valdiklis. Python-SQL valdiklis parašytas python programavimo kalba. Lokalizacijos aplikacijos veikimo metu, šiame valdiklyje yra sukuriama sesija tarp duomenų bazės ir lokalizacijos aplikacijos. Taip pat Python-SQL valdiklyje yra sukurti naudojamų duomenų bazės lentelių modeliai, kuriuos naudojant galima atlikti veiksmus su serveryje esančia duomenų baze. Python-SQL valdiklyje panaudotos sqlalchemy, flask_sqlalchemy, geoalchemy2 bei flask bibliotekos. Visos likusios lokalizacijos aplikacijos (Web servisi, padėties nustatymo algoritmas bei kalibravimo algoritmas) dalys veiksmus su duomenų baze atlieka naudodamos šį valdiklį.

Padėties nustatymo algoritmas. Padėties nustatymo algoritmas, šioje prototipinėje padėties nustatymo sistemoje, parašytas python programavimo kalba. Šio algoritmo vykdymas yra iškviečiamas tada, kai gaunama web serviso *POSTGetPosition* užklausa 7.2.5. WiFi matavimų informacija priimta užklauso metu yra perduodama padėties nustatymo algoritmui ir pradedamas padėties nustatymo algoritmo vykdymas, jo rezultatas nurodo web serviso užklauso siuntėjo padėtį. Gautas rezultatas yra išsiunčiamas užklauso siuntėjui atgal. Padėties nustatymo algoritme panaudotos scipy ir numpy bibliotekos. Detalesnis padėties nustatymo algoritmo veikimas aprašytas 3.4 poskyryje.

Kalibravimo algoritmas. Kalibravimo algoritmas parašytas python programavimo kalba. Šio algoritmo vykdymas yra reikalingas tada, kai padėties nustatymo programėlę naudojantis įrenginys yra ne tas pats įrenginys su kuriuo buvo surinkti atskaitiniai WiFi atspaudų duomenys. Šio algoritmo vykdymas yra iškviečiamas tada, kai gaunama web serviso *POSTCalibration* užklausa 7.2.6. WiFi matavimų informacija gauta užklauso metu yra perduodama kalibravimo algoritmui ir pradedamas kalibravimo algoritmo vykdymas. Šio algoritmo rezultatas nurodo koeficientus, kuriais kalibruojami užklauso siuntusio įrenginio WiFi matavimai, surinktų atskaitinių WiFi duomenų atžvilgiu. Gautas rezultatas yra išsiunčiamas užklauso siuntėjui atgal. Padėties nustatymo algoritme panaudota numpy biblioteka. Detalesnis kalibravimo algoritmo veikimas aprašytas 5.1 poskyryje.

8. Mobiliosios programėlės

Kuriant prototipinę padėties nustatymo sistemą buvo nuspręsta sukurti dvi mobiliąsias programėles skirtas duomenų rinkimui ir perdavimui serveriui, bei padėčiai pastate nustatyti bei atvaizduoti. Sekančiuose skyriuose aprašyti naudoti įrankiai, programavimo kalba, bei naudotos bibliotekos. Taip pat detaliam aprašytos sukurtos mobiliosios programėlės.

8.1. Pasiruošimas mobiliųjų programėlių kūrimui

Prieš pradėdant kurti mobiliąsias programėles reikėjo išsiaiškinti tinkamiausią mobiliąją platformą ir įrankius su kuriais būtų įmanoma įgyvendinti padėties nustatymą uždaroje patalpose naudojant WiFi atspaudus. Sekančiuose skyreliuose aprašyti kriterijai pagal kuriuos buvo išrinkta tinkamiausia mobiliųjų įrenginių platforma ir reikalingi įrankiai.

8.1.1. Mobilųjų įrenginių platformos kriterijai

Renkantis mobiliųjų įrenginių platformą buvo atsižvelgta į šiuos kriterijus:

- WiFi rodmenų išgavimo galimuma;
- mobiliosios platformos populiarumas.

Mobiliosios platformos populiarumui nustatyti buvo panaudoti IDC (Tarptautinė duomenų rinkimo korporacija) tinklapio duomenys. Ši korporacija specializuojasi rinkdama duomenis apie telekomunikacijas, informacines technologijas ir kita. Žemiau pateikiama mobiliųjų platformų naudojimo statistika [1].

9 lentelė. Mobilųjų platformų naudojimas

Laikotarpis	Android	iOS	Windows Phone	BlackBerry OS	Others
2015 II ketvirtis	82.8%	13.9%	2.6%	0.3%	0.4%
2014 II ketvirtis	84.8%	11.6%	2.5%	0.5%	0.7%
2013 II ketvirtis	79.8%	12.9%	3.4%	2.8%	1.2%
2012 II ketvirtis	69.3%	16.6%	3.1%	4.9%	6.1%

Pagal pateiktą statistiką (žr. 9 lentelę) aiškiai matyti, kad Android ir iOS išsiskiria tarp visų mobiliųjų platformų pagal populiarumo kriterijų. Tačiau iOS mobili platforma netenkina antrojo kriterijaus, tai yra, su šiais įrenginiais negalima išgauti WiFi rodmenų. Įvertinus mobilias platformas pagal apibrėžtus kriterijus nuspręsta mobiliąsias programėles kurti Android įrenginiams.

Mobiliosios programėlės kuriamos Java programavimo kalba naudojant Android Studio karkasą. Kuriamos programos veiks įrenginiuose, kuriuose įdiegta 4.0.3+ Android operacinė versija. Lokaliai išsaugotų duomenų keitimui bei peržiūrai buvo naudotas Firefox naršyklės įskiepis SQLite Manager. Duomenų gavimui ir siuntimui iš/į serverį buvo naudota biblioteka Volley. Kuriamų programėlių testavimui buvo naudojami penki Android įrenginiai:

- LG G2, Android versija 4.0.3;
- HTC Desire HD, Android versija 4.4.2;
- LG Optimus L9 II, Android versija 4.4.2;
- Samsung Galaxy S5, Android versija 5.1.1;
- Samsung Galaxy S6, Android versija 6.0.1;

8.1.2. Bibliotekos darbu su žemėlapiais

Darbu su žemėlapiais mobiliose programėlėse buvo reikalinga biblioteka galinti dirbti su WMS (Web Map Service) sluoksniais. Buvo renkamosi tarp trijų bibliotekų:

- OpenLayers;
- QGIS;
- ArcGIS;

OpenLayers, tai atviro šaltinio biblioteka skirta darbu su žemėlapiais. Ši biblioteka parašyta JavaScript kalba. OpenLayers biblioteka suteikia galimybę įtraukti žemėlapius į tinklalapius ir mobilias programėles. Biblioteka dirba su WMS sluoksniais, tačiau versija skirta mobiliosioms programėlėms tinka jei mobiliosios programėlės kuriamos HTML5 kalbos pagrindu. OpenLayers biblioteka buvo atmesta kaip netinkama, nes šiame darbe mobiliosios programėlės, kuriamos Java programavimo kalba.

QGIS, tai atviro šaltinio biblioteka pasižyminti galimybę dirbti su WMS sluoksniais. Ši biblioteka turi mobiliąją versiją ir yra sukurta mobili programėlė, kurioje galima dirbti su QGIS projektais, tačiau mobilios bibliotekos versijos panaudojimas kitose mobiliose programėlėse prieinamas tik uždarai kūrėjų grupei, todėl šios bibliotekos nepavyko panaudoti.

ArcGIS, tai komercinis produktas turintis versijas skirtas tinklalapiams ir mobiliosioms programėlėms. ArcGIS turi galimybę dirbti su WMS sluoksniais. ArcGIS mobili versija pranašesnė už Openlayers mobiliąją versiją tuo, kad mobili versija skirta Java programavimo kalba parašytiems įrenginiams. ArcGIS suteikia kūrėjo licenciją, su kuria galima kurti ne komerciniams tikslams skirtas programėles.

Palyginus visas darbu su žemėlapiais skirtas bibliotekas, matyti, kad visos bibliotekos geba dirbti su WMS sluoksniais. Taip pat visos bibliotekos turi mobiliąsias versijas. Tačiau OpenLayers mobili versija skirta HTML5 kalbos pagrindu parašytoms programoms, o QGIS bibliotekos mobili versija prieinama tik uždarai grupei kūrėjų. Atsižvelgus į tai, kad ArcGIS biblioteka pasižymi panašiomis savybėmis kaip ir kitos bibliotekos buvo nuspręsta darbu su žemėlapiais naudoti ArcGIS biblioteką.

8.2. Duomenų rinkimo programa

Šio darbo metu buvo sukurtos trys duomenų rinkimo programėlės versijos. Šiame skyriuje aprašytos duomenų rinkimo programėlės versijos bei pagrindinės užduotys, kurias atlieka programėlė.

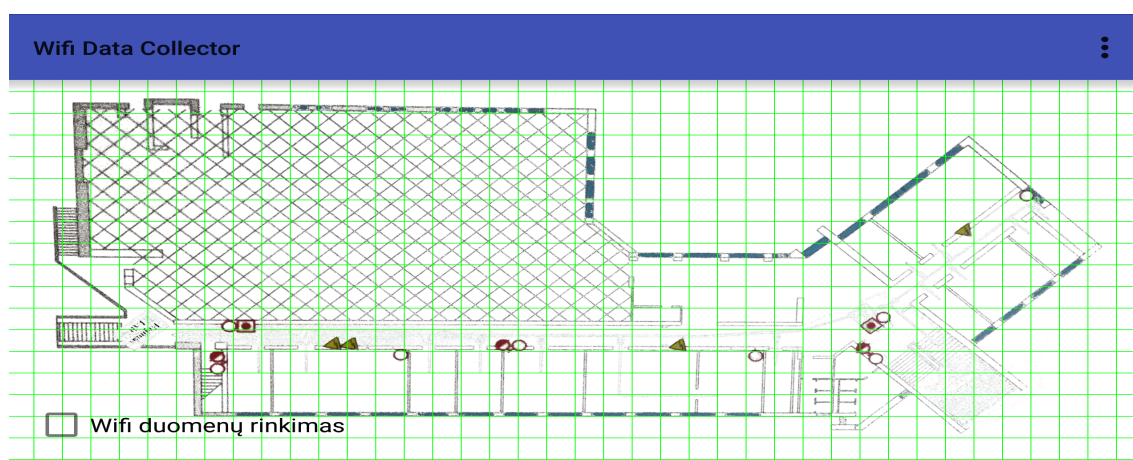
Duomenų rinkimo programėlės galutinės versijos pagrindinės užduotys:

- Analizuojamos patalpos žemėlapio atvaizdavimas;
- Žemėlapio dalijimas poligonais;
- Pradinių WiFi rodmenų išgavimas žinomose žemėlapio vietose;
- Pradinių barometro rodmenų išgavimas žinomose žemėlapio vietose;
- Pradinių magnetinio lauko rodmenų išgavimas žinomose žemėlapio vietose;
- Duomenų saugojimas lokaliajoje duomenų bazėje;
- Lokalios duomenų bazės eksportavimas į mobilaus įrenginio vidinę atmintį;
- WiFi ir barometro rodmenų siuntimas į serverį.
- Surinktų duomenų trynimas lokaliajoje duomenų bazėje.
- Surinktų duomenų trynimas serveryje.

Darbo metu buvo sukurta mobili programa gebanti atvaizduoti žemėlapi, žemėlapi skirstyti poligonais, rinkti WiFi, barometro ir magnetino lauko rodmenis ir juos išsaugoti lokaliai. Taip pat ši programėlė siunčia WiFi ir barometro rodmenis į serverį bei atlieka kitas aukščiau išvardintas užduotis. Darbo metu buvo sukurtos trys programėlės versijos V1, V2, V3. Žemiau pateikti programėlių versijų aprašymai.

8.2.1. V1 versija

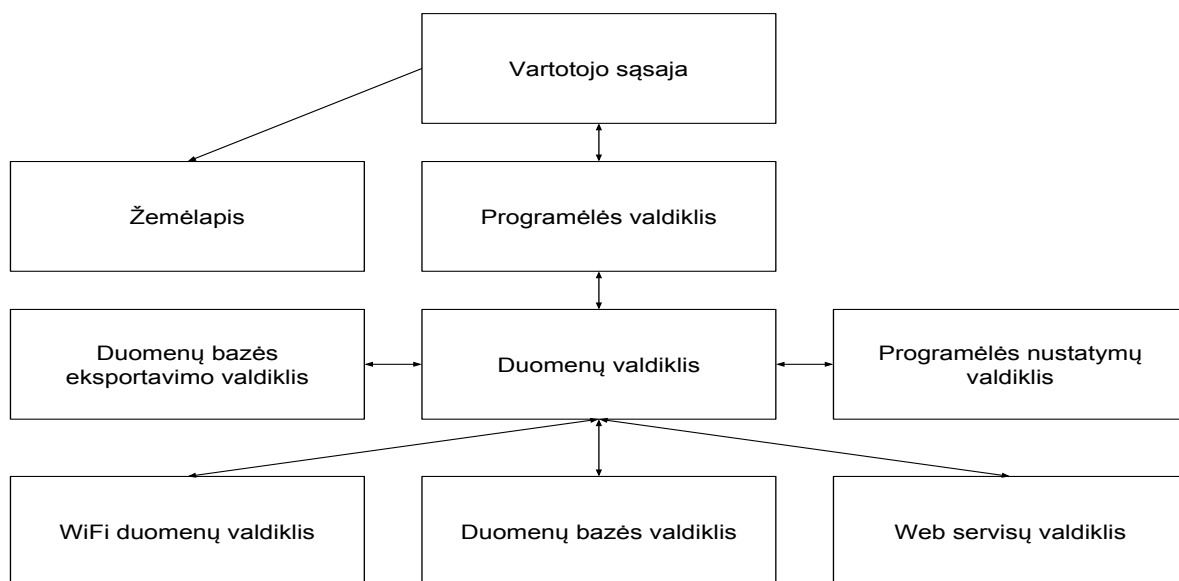
V1 versijoje analizuojamų patalpų žemėlapiu buvo laikomas evakuacijos plano paveikslėlis. Šis paveikslėlis buvo suskirstytas į poligonus. Paspaudus ant poligono surenkami reikalingi WiFi signalų stiprumai ir poligono duomenys, ir išsaugomi lokaliajoje duomenų bazėje. Surinktus duomenis galima eksportuoti į vidinę mobilaus įrenginio atmintį, kad būtų galima atlikti detalesnę surinktų WiFi rodmenų analizę. Žemiau pateikiamas programėlės V1 versijos ekranvaizdis.



16 pav. WiFi duomenų rinkimo programėlės versija V1

Pateiktame V1 programėlės versijos ekranvaizdyje (žr. 16 pav.) matyti, kad žemėlapis suskirstytas į lygias dalis (poligonus). Poligonų dydį galima keisti programėlėje keičiant poligonų ilgį ir plotį. Poligonai, kuriuose jau yra surinkti duomenys pažymimi nuspalvinant poligoną geltona spalva.

V1 programėlės versijos blokinė schema. Anksčiau buvo aptartas programėlės veikimas, o dabar aptarsime programėlės struktūrą bei jos dalis. Žemiau pateikiama programėlės blokinė schema.



17 pav. V1 programėlės versijos blokinė schema

V1 programėlės versijos struktūrą sudaro devyni pagrindiniai komponentai (žr. 17 pav.). Programėlė suprojektuota taip, kad atskiri programėlės komponentai būtų nepriklausomi vienas nuo kito. Kuriant programėles Java programavimo kalba labai brangiai kainuoja (laiko atžvilgiu) objektų kūrimas. Kad optimizuoti programėlės darbą buvo panaudotas vieno kintamojo dizaino modelis [angl. *Singleton design pattern*]. Vieno kintamojo modelis pritaikytas pagrindinėms programėlės dalims. Programėlės paleidimo metu sukuriamos nuorodos į pagrindinius programėlės komponentus. Programėlės darbo metu yra tik po vieną nuorodą į pagrindinius programėlės komponentus.

Žemėlapis. Android operacinėje sistemoje kiekvienai programėlei sukuriama privati direktorija ir išskiriama atminties (atminties kiekis priklauso nuo mobilaus įrenginio turimų resursų). Privacioje direktorijoje laikomi programėlei priklausantys failai. Šioje direktorijoje patalpintas žemėlapis *.png* formatu taip pat šioje direktorijoje patalpintas žemėlapis *.tiff* formatu. V1 programėlės versijoje buvo naudojamas žemėlapis *.png* formatu.

Vartotojo sąsaja. Vartotojo sąsaja atsakinga už analizuojamų patalpų žemėlapio atvaizdavimą, WiFi signalų rinkimo inicijavimą ir poligonų, kuriuose surinkti duomenys pažymėjimą. Atvaizduojamas žemėlapis imamas iš privačios programėlės direktorijos. Atvaizduojamas žemėlapis yra aukštos kokybės ir užima daug atminties. Todėl prieš atvaizduojant žemėlapij jis proporcingai sumažinamas, kad išlaikytų kokybę ir užimtų mažiau atminties. Žemėlapio mažinimo reikia, nes mobiliuose įrenginiuose operatyvios atminties [angl. *Random-access memory*] kiekis ribotas ir į atmintį įkeliant didelius failus gali sustoti programos veikimas dėl nepakankamo kiekio operatyvios atminties. Atvaizdavus žemėlapij yra atliekamas poligonų ribų žymėjimas, atsižvelgiant į poligonų pločio ir ilgio parametrus. Taip pat atliekamas poligonų, su jau nustatytais WiFi signalų stiprumais, pažymėjimas.

Programėlės valdiklis. Programėlės paleidimo metu, naudojant vieno kintamojo modelį, sukuriamas programėlės valdymo komponentas. Programėlės valdiklis turi nuorodą į duomenų valdymo komponentą. Atlikus veiksmą vartotojo sąsajoje, programėlės valdymo komponentas kreipiasi į duomenų valdymo komponentą ir šis inicijuoja atitinkamą funkciją, pavyzdžiui WiFi duomenų rinkimą. Programėlės valdymo komponentas taipogi yra atsakingas už duomenų perdavimą vartotojo sąsajai.

Duomenų valdiklis. Programėlėje visos operacijos atliekamos per duomenų valdymo komponentą. Duomenų valdymo komponentas atsakingas už duomenų perdavimą ir gavimą iš kitų programos komponentų. Duomenų valdymo komponentas bendrauja su WiFi, duomenų bazės, web servisų, programos nustatymų ir duomenų bazės eksportavimo komponentais. Atlikus veiksmą, bet kuriame iš aukščiau minėtų komponentų, duomenų valdymo komponentas perduoda duomenis programėlės valdymo komponentui, o šis vartotojo sąsajai.

Duomenų bazės valdiklis. Duomenų bazės valdymo komponentas atsakingas už SQLite duomenų bazės sukūrimą, duomenų įrašymą, atnaujinimą, surinkimą bei trynimą. Visas užklausas duomenų bazei pateikia duomenų valdymo komponentas, o duomenų bazės valdymo komponentas jas įvykdo ir perduoda surinktus duomenis duomenų valdymo komponentui. Duomenų bazės komponentas bendrauja tik su duomenų valdymo komponentu.

Programėlėje naudojamoje duomenų bazėje saugomi tik primityvus duomenų tipai kaip sveikieji skaičiai, skaičiai su kableliu ir tekstai. Todėl buvo pasirinkta naudoti objektų surišimo technologiją [*angl. Object-relational mapping*]. Ši technologija suriša Java programavimo kalba sukurtus objektus su duomenų bazės lentelėmis.

V1 programėlės versijoje SQLite duomenų bazė sudaryta iš dviejų lentelių. Pirmoje lentelėje saugomi WiFi surinkti duomenys. Antroje lentelėje saugomi poligonų pozicijos duomenys. Šios lentelės tarpusavyje surištos pagal poligono poziciją. Tarp šių lentelių yra 1 su N sąryšis, tai reiškia, kad visų poligonų pozicijų duomenys unikalūs, tačiau kiekvienas poligonas gali turėti keletą WiFi prieigos taškų signalų duomenis.

Duomenų bazės eksportavimo valdiklis. Duomenų bazės eksportavimo komponentas turi tik vieną užduotį - SQLite duomenų bazės eksportavimas į vidinę mobilaus įrenginio atmintį. Duomenų bazės eksportavimo prireikė detalesnei duomenų bazėje surinktų duomenų analizei. Analizuojant duomenų bazėje sukauptus duomenis buvo naudotas SQLite Manager įskiepis skirtas Mozilla Firefox naršyklei. Šis įskiepis leidžia atidaryti SQLite duomenų bazę, matyti duomenų bazės struktūrą ir duomenis. Taip pat šis įskiepis leidžia keisti duomenų bazėje išsaugotus duomenis.

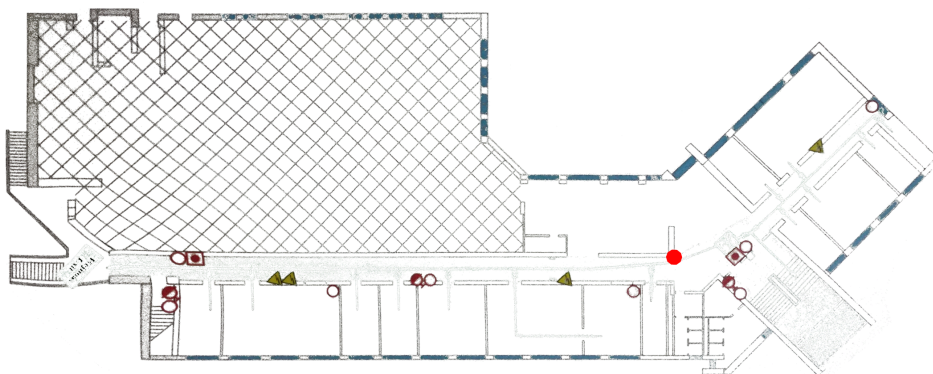
Programėlės nustatymų valdiklis. Šis komponentas atsakingas už programėlės nustatymų saugojimą ir keitimą. Programėlės nustatymai saugomi atskirame faile, privačioje programėlės direktorijoje. Pagrindinė programėlės nustatymų komponento užduotis - leisti arba uždrausti rinkti WiFi duomenis. Šio nustatymo reikia, kad nebūtų pradėtas atsitiktinis WiFi duomenų rinkimas prilietus mobilaus įrenginio ekraną.

WiFi duomenų valdiklis. WiFi duomenų valdymo komponentas atsakingas už WiFi prieigos taškų signalų stiprumų surinkimą. WiFi duomenys renkami kiekvienam poligonui. Duomenų rinkimas atliekamas kiekvieno poligono centre. Renkant WiFi duomenis atliekamas skenavimas

pasinaudojus mobiliu įrenginiu. Užfiksuoti WiFi signalų stiprumai priskiriami poligonui ir perduodami duomenų valdymo komponentui. Duomenų valdymo komponentas duomenis perduoda duomenų bazės valdymo komponentui, o šis duomenis išsaugo SQLite duomenų bazėje.

8.2.2. V2 versija

V2 versijoje naudojamas žemėlapis surištas su LKS koordinacių sistema. Šis žemėlapis laikomas įrenginio vidinėje atmintyje. Ateities planuose ketinama šį žemėlapią gauti panaudojus WMS (Web Map Service). Žemiau pateikiamas V2 programėlės versijos ekranvaizdis.



18 pav. WiFi duomenų rinkimo programėlės versija V2

Pateiktame V2 programėlės versijos ekranvaizdyje (žr. 18 pav.) pavaizduotas analizuojamų patalpų žemėlapis surištas su LKS koordinacių sistema. V2 programėlės versijoje buvo atliktas patikrinimas ar žemėlapio koordinatės tinkamai surištos. Patikrinimas buvo atliktas parinkus žinomą koordinatę ir ją nupiešus ant programėlėje atvaizduojamo žemėlapio. Atlikus šį bandymą įsitikino, kad žemėlapis teisingai surištas su LKS koordinacių sistema. Tačiau darbo metu paaiškėjo, kad naudojantis WMS nepavyksta gauti poligonų rodmenų, tai buvo nuspręsta žemėlapius bei informaciją apie kiekvieno žemėlapio poligonus saugoti lokaliai duomenų rinkimo programėlėje. Todėl buvo sukurta duomenų rinkimo programėlės versija V3.

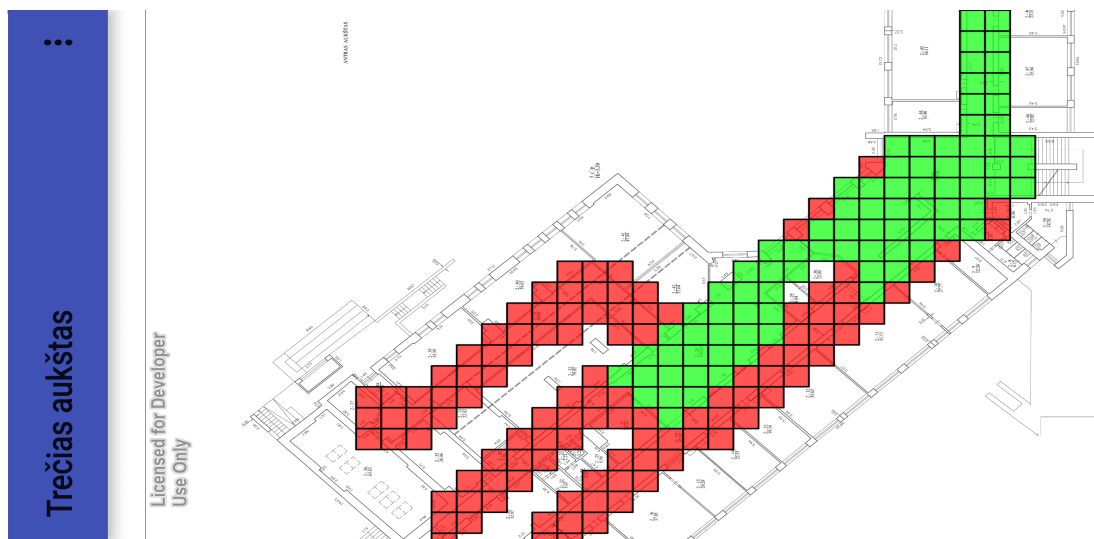
8.2.3. V3 versija

Šiame skyriuje aprašyta galutinė duomenų rinkimo programėlė ir atlikti pakeitimai lyginant su V1 versija (žr. 8.2.1 skyrių).

V3 versijoje buvo pakeistas analizuoto evakuacijos plano žemėlapis į tris kadastrinių matavimų žemėlapius. Taip buvo padaryta, nes V1 versija buvo apribota rinkti duomenis tik viename pastato aukšte. Taip pat V1 versijoje žemėlapiu buvo laikomas paveikslėlis .png formatu, todėl ant šio paveikslėlio nupiešti poligonai buvo netikslūs.

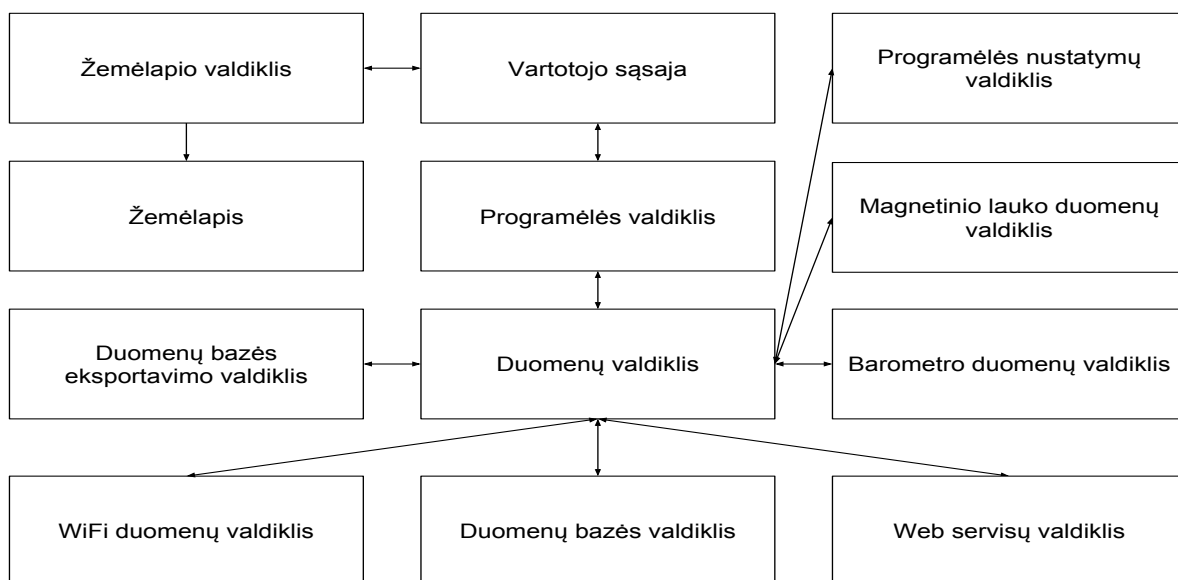
V3 versijoje duomenų rinkimo programėlės veikimo principas liko panašus į V1 versiją. Ant kiekvieno aukšto žemėlapio nupiešiami dviejų metrų ilgio ir dviejų metrų pločio poligonai. Paspaudus ant poligono surenkami WiFi, barometro, magnetinio lauko ir poligono duomenys. Tiesa gali būti, kad ne visus rodmenis pavyks išgauti, nes nevisi Android įrenginiai turi tokius pačius sensorius, todėl prototipinę sistemą teko pritaikyti atvejams, kai ne visi rodmenys gaunami. Surinkti duomenys siunčiami į serverį ir jei iš serverio gaunamas atsakymas, kad duomenys išsaugoti apie

duomenų išsaugojimą nusiųsti duomenis į serverį, tuomet surinkti duomenys taip pat išsaugomi programėlės vidinėje duomenų bazėje. V3 versija buvo papildyta galimybe ištrinti duomenis tiek lokaliai programėlėje, tiek serveryje. Žemiau pateikiamas programėlės V3 versijos ekranvaizdis.



19 pav. Duomenų rinkimo programėlės V3 versija

Pateiktame V3 programėlės versijos ekranvaizdyje (žr. 19 pav.) matomas vienas iš trijų darbe analizuotų pastato aukštų žemėlapis. Žemėlapis suskirstytas į raudonus ir žalius poligonus. Raudoni poligonai reiškia, kad šiuose poligonuose dar nėra surinkti pradiniai apmokymo duomenys, o žali nurodo, kad pradiniai duomenys tame poligone surinkti ir nusiųsti į serverį. Žemiau pateikiama atnaujinta duomenų rinkimo programėlės versijos blokinė schema:



20 pav. Duomenų rinkimo programėlės V3 versijos blokinė schema

V3 programėlės versijos struktūrą sudaro dvylika pagrindinių komponentų (žr. 20 pav.). Programėlės V3 versija suprojektuota taip pat kaip ir V1 versija (žr. 8.2.1 skyrių). V3 versija papildyta barometro, magnetinio lauko ir žemėlapių valdikliais. Žemiau pateikiami atlikti pakeitimai atskiruose komponentuose.

Žemėlapis. Programėlėje išsaugoti trys žemėlapiai *.tiff* formatu. Taip pat išsaugoti poligonų duomenis laikantys failai *.shape* formatu. Programėlėje nebenaudojamas žemėlapis *.png* formatu, todėl jis buvo pašalintas.

Žemėlapių valdiklis. Atsakingas už programėlėje naudojamų žemėlapių ir poligonų atvaizdavimą. Taip pat šis valdiklis valdo analizuojamų patalpų aukštų žemėlapių sukeitimus ir poligonų būsenas (raudonas - nesurinkti duomenys, žalias - surinkti duomenys, - mėlynas - pasirinktas poligonas). Žemėlapių valdiklis tiesiogiai bendrauja su vartotojo sąsaja.

Barometro duomenų valdiklis. Šis valdiklis atsakingas už barometro duomenų išgavimą naudojant mobilų įrenginio sensorius. Pagrindinė šio valdiklio užduotis išgauti ir saugoti laikinoje atmintyje naujausią altitudės reikšmę. Barometro duomenų valdiklis pasiekiamas tik per Duomenų valdiklį.

Magnetinio lauko duomenų valdiklis. Šis valdiklis atsakingas už magnetinio lauko duomenų išgavimą naudojant mobilų įrenginio sensorius. Pagrindinė šio valdiklio užduotis išgauti ir saugoti laikinoje atmintyje naujausias magnetinio lauko komponentų *X*, *Y*, *Z* reikšmes. Barometro duomenų valdiklis taip pat pasiekiamas tik per Duomenų valdiklį.

Web servisų valdiklis. Atsakingas už duomenų siuntimą ir gavimą iš serverio. Web servisų valdiklis sudaro užklausas naudodamas Volley biblioteką. Duomenų surišimui su Java programavimo kalbos objektais naudojama biblioteka GSON. Šis valdiklis atlieka duomenų siuntimo ir duomenų trynimo užklausas. Šių užklausų parametrai ir struktūra aprašyta ankstesniuose skyriuose (žr. 7.2 skyrių).

Kiti komponentai nurodyti blokinėje schemoje (žr. 20 pav.) pasikeitę minimaliai arba atlieka tas pačias funkcijas kaip ir duomenų programėlės V1 versijoje (žr. 8.2.1 skyrių).

8.2.4. Duomenų rinkimas

Naudojantis Android išmaniųjų telefonų sensoriais buvo surinkti WiFi, barometro ir magnetinio lauko rodmenys. Žemiau išvardinti duomenų surinkimo programėle renkami rodmenys.

Renkami WiFi rodmenys:

- BSSI - WiFi prieigos taško MAC adresas;
- SSID - WiFi prieigos taško pavadinimas;
- RSSI - signalo stiprumas iki prieigos taško.

Renkami magnetinio lauko rodmenys:

- X - magnetinio lauko x ašies komponentė;
- Y - magnetinio lauko y ašies komponentė;
- Z - magnetinio lauko z ašies komponentė.

Renkami barometro rodmenys:

- altitudė - aukštis virš jūros lygio.

Pradiniams WiFi signalų stiprumams fiksuoti buvo sukurtos kelios versijos ir lyginami gauti signalų stiprumai.

WiFi duomenų rinkimas be pauzių. WiFi rodmenys fiksuojami mobiliuoju įrenginiu iš anksto žinomoje patalpos vietoje ir išsaugomi duomenų bazėje. Duomenų bazėje išsaugomi paskutiniai atnaujinti WiFi prieigos taškų rodmenys. WiFi duomenų fiksavimas atliekamas be pauzių, tai yra duomenų bazėje išsaugomi pirmi gauti WiFi duomenų rodmenys. Rodmenis renkant šiuo būdu pastebėta, kad WiFi rodmenys mažai keičiasi nors fiksuojama įvairiose patalpos vietose.

WiFi duomenų rinkimas intervalais. WiFi duomenų rinkimo algoritmas buvo patobulintas fiksuojant WiFi rodmenis tam tikroje patalpos vietoje tam tikrą laiko tarpą T sekundžių. WiFi duomenų rinkimui buvo taikomi šie laiko intervalai:

- 1 sekundė;
- 3 sekundės;
- 5 sekundės;
- 10 sekundžių;
- 20 sekundžių.

WiFi duomenų rinkimo programėlėje nustačius norimą laiko intervalą ir pasirinkus poligoną pradedami fiksuoti WiFi prieigos taškų signalų stiprumai kas 0.5 sekundės. Tarkime jei buvo pasirinktas 20 sekundžių intervalas, tuomet buvo atliekama 40 WiFi prieigos taškų signalų fiksavimai. Signalų fiksavimai saugomi mobilaus įrenginio darbinėje atmintyje kol praeina pasirinktas laiko tarpas. Iš surinktų duomenų kiekvienam prieigos taškui nustatomas vidutinis WiFi prieigos taško signalo stiprumas.

WiFi duomenų rinkimas šalinat nuokrypius . Renkant WiFi duomenis per tam tikrą laiko tarpą pastebėjome, kad pasitaiko signalo stiprumų šuoliai ir tik po tam tikro laiko tarpo WiFi signalų stiprumai nusistovi. WiFi duomenų rinkimą patobulinome šalinant kiekvieno prieigos taško per tam tikrą intervalą surinktų WiFi signalų minimalią ir maksimalią reikšmes. Šiuo patobulinimu siekiama užfiksuoti WiFi prieigos taškų signalų reikšmes kuo artimesnes vidutinėms.

Barometro rodmenys . Nevisuose Android išmaniuose telefonuose yra barometro sensorius, todėl ne su visais įrenginiais pavyko išgauti barometro altitudės reikšmes. Įrenginiuose, kuriuose nepavyko išgauti šių reikšmių nebuvo atliekami aukšto nustatymo eksperimentai. Surinkti barometro rodmenys skirtingais įrenginiais tuose pačiuose aukštuose rodydavo skirtingas altitudės reikšmes, tačiau šios reikšmės tolygiai pasiskirstę (žr. 33 pav.).

Magnetinio lauko rodmenys . Magnetinio lauko rodmenis pavyko išgauti iš visų testuotų Android išmaniųjų telefonų. Tačiau prieš renkant magnetinio lauko rodmenis reikia sukalibruoti Android išmanųjį telefoną. Kalibracijos su duomenų rinkimo programėle padaryti nepavyko, nes Android operacinė sistema neleidžia prieiti prie telefono magnetinio lauko kalibracijos. Kalibravimą galima padaryti rankiniu būdu, tačiau telefonas per kelias valandas vėl išsikalibruoja. Taigi buvo nuspręsta magnetinio lauko rodmenų nenaudoti padėties nustatymo algoritmuose.

8.3. Padėties nustatymo programa

Darbo metu buvo sukurtos dvi padėties nustatymo programėlės versijos. Šiame skyriuje aprašytos padėties nustatymo programėlės versijos bei pagrindinės užduotys, kurias atlieka programėlė.

Padėties nustatymo programėlės galutinės versijos pagrindinės užduotys:

- WiFi rodmenų nustatymas realiu laiku nežinomoje patalpos vietoje;
- Barometro rodmenų nustatymas realiu laiku nežinomoje patalpos vietoje;
- Magnetinio lauko rodmenų nustatymas realiu laiku nežinomoje patalpos vietoje;
- Padėties nustatymas lokaliai apdorojant WiFi duomenis;
- Padėties nustatymas perduodant apdorotus WiFi rodmenis į serverį;
- WiFi ir barometro rodmenų siuntimas į serverį;
- WiFi ir barometro duomenų kalibravimas;
- Nustatytos padėties atvaizdavimas žemėlapyje;
- Automatinis žemėlapio sukeitimas;
- Aukšto nustatymo algoritmų testavimas.

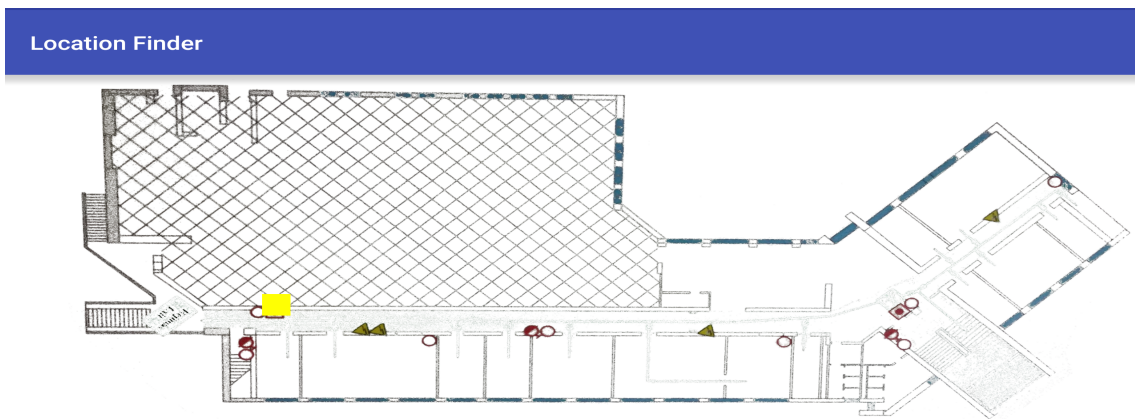
Darbo metu sukurta mobili programa gebanti atvaizduoti žemėlapi, žemėlapius automatiškai sukeisti vietomis kai atnaujinama nustatyta padėtis, rinkti WiFi, barometro ir magnetinio lauko rodmenis, kalibruoti WiFi ir barometro rodmenis, nustatyti padėti siunčiant duomenis į serverį arba lokaliai. Taip pat programėlė atlieka aukšto nustatymo algoritmų testavimą bei kitas aukščiau išvardintas užduotis. Darbo metu buvo sukurtos dvi programėlės versijos *V1*, *V2*. Žemiau pateikti programėlių versijų aprašymai.

8.3.1. Padėties nustatymo programėlės *V1* versija

Padėties nustatymo programėlės *V1* versija geba atvaizduoti žemėlapi, rinkti WiFi duomenis realiu laiku, apskaičiuoti Euklido atstumus, atvaizduoti surastą artimiausią poligoną. Šioje versijoje nebuvo rodmenų siuntimo į serverį. Šios programėlės versijos pagrindinis tikslas sujungti esmines programėlės dalis, kad būtų galima atlikti padėties nustatymo testus.

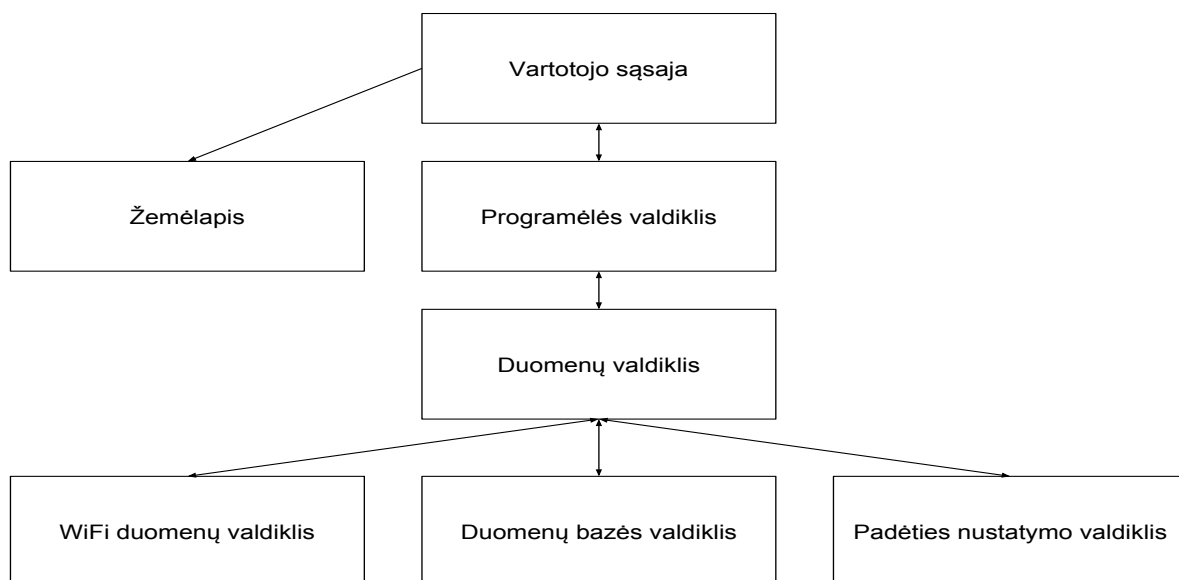
Programėlės veikimo principas. Duomenų rinkimo programa surenkami pradiniai WiFi duomenys ir eksportuojama SQLite duomenų bazė. Padėties nustatymo programos paleidimo metu importuojama SQLite duomenų bazė, kurioje surinkti pradiniai poligonų ir WiFi duomenys. Poligonų ir pradiniai WiFi duomenys surenkami iš duomenų bazės ir užkraunami į atmintį. Atliekamas WiFi duomenų skenavimas realiu laiku kas penkias sekundes. Lyginami realiu laiku gauti WiFi duomenys su pradiniais WiFi duomenimis ir apskaičiuojami Euklido atstumai. Poligonas su mažiausiu Euklido atstumu pažymimas geltona spalva vartotojo sąsajoje.

Žemiau pateikiamas padėties nustatymo programėlės ekranvaizdis.



21 pav. Padėties nustatymo programėlės V1 versijos ekranvaizdis

Pateiktame padėties nustatymo programėlės ekranvaizdyje (žr. 21 pav.) geltonu kvadratu žymimas rastas artimiausias poligonas palyginus WiFi rodmenis gautus realiu laiku su pradiniais WiFi rodmenimis. Palyginus padėties nustatymo programos ekranvaizdį (žr. 21 pav.) su WiFi duomenų rinkimo programos ekranvaizdžiu (žr. 16 pav.) matyti, kad šių programėlių vartotojo sąsajos labai panašios. Šių programėlių vartotojo sąsajos skiriasi tuo, kad padėties nustatymo programėlėje nėra nubrėžtų poligonų ribų ir geltonas kvadratis žymi artimiausią surastą poligoną, o ne poligonus, kuriuose surinkti pradiniai WiFi rodmenys. Žemiau pateikiama padėties nustatymo programos blokinė schema.



22 pav. Padėties nustatymo programėlės V2 versijos blokinė schema

Padėties nustatymo programos struktūrą sudaro septyni pagrindiniai komponentai (žr. 22 pav.) Kuriant padėties nustatymo programą buvo laikomasi tos pačios struktūros kaip ir WiFi duomenų rinkimo programoje (žr. 17 pav.). Kuriant programą buvo naudotas vieno kintamojo dizaino mo-

delis ir programėlė suprojektuota taip, kad atskiri programėlės komponentai būtų nepriklausomi vienas nuo kito. Taip pat kuriant padėties nustatymo programėlę buvo panaudoti WiFi duomenų, programėlės valdiklio, žemėlapių, duomenų valdiklio ir vartotojo sąsajos komponentai iš WiFi duomenų rinkimo programos. Šių komponentų detalus aprašymas pateiktas prie WiFi duomenų rinkimo programėlės V1 versijos blokinės schemos (žr. 17 pav.). Padėties nustatymo programėlėje sukurti du papildomi komponentai: vietos nustatymo ir duomenų bazės. Detalesnis šių komponentų aprašymas pateiktas žemiau.

Duomenų bazės valdiklis. Padėties nustatymo programėlėje SQLite duomenų bazė paimta iš WiFi duomenų rinkimo programėlės. Duomenų bazės valdymo komponentas atsakingas už SQLite duomenų bazės importavimą ir duomenų surinkimą. Visas užklausas duomenų bazei pateikia duomenų valdymo komponentas, o duomenų bazės valdymo komponentas jas įvykdo ir perduoda surinktus duomenis duomenų valdymo komponentui. Duomenų bazės komponentas bendrauja tik su duomenų valdymo komponentu. Ateityje pradiniai WiFi duomenys bus gaunami iš serverio, todėl duomenų bazės importavimo bus atsisakyta.

Padėties nustatymo programėlės paleidimo metu importuojama SQLite duomenų bazė, todėl nebuvo galima naudoti objektų surišimo technologijos. Vietoje šios technologijos buvo rašomos SQL užklausos poligonų ir WiFi rodmenų gavimui.

Padėties nustatymo valdiklis. Padėties nustatymo valdiklis atsakingas už realiu laiku užfiksuotų WiFi signalų palyginimą su pradiniais WiFi signalais. Padėties nustatymo komponentas bendrauja tik su duomenų valdymo komponentu. Padėties nustatymo programėlės paleidimo metu duomenų valdymo komponentas surenka poligonų ir pradinių WiFi signalų duomenis, ir juos perduoda padėties nustatymo valdikliui. Poligonų duomenys surenkami ir perduodami padėties nustatymo komponentui, nes rinkti duomenis iš duomenų bazės kiekvieną kartą brangiai kainuojanti operacija, o WiFi duomenų rinkimas realiu laiku atliekamas kas penkias sekundes.

Realiu laiku užfiksuoti WiFi rodmenys perduodami duomenų valdikliui, o šis juos perduoda padėties nustatymo komponentui. Padėties nustatymo komponentas lygina gautus WiFi duomenis su kiekvieno poligono pradiniais WiFi duomenimis. Lyginant duomenis apskaičiuojami Euklido atstumai. RealIU laiku fiksuotų WiFi duomenų vieta yra poligonas, kurio Euklido atstumas mažiausias. Poligonas su mažiausiu euklido atstumas perduodamas duomenų valdymo komponentui, o šis perduoda vartotojo sąsajai. Vartotojo sąsajoje nustatymas poligonas žymimas geltona spalva.

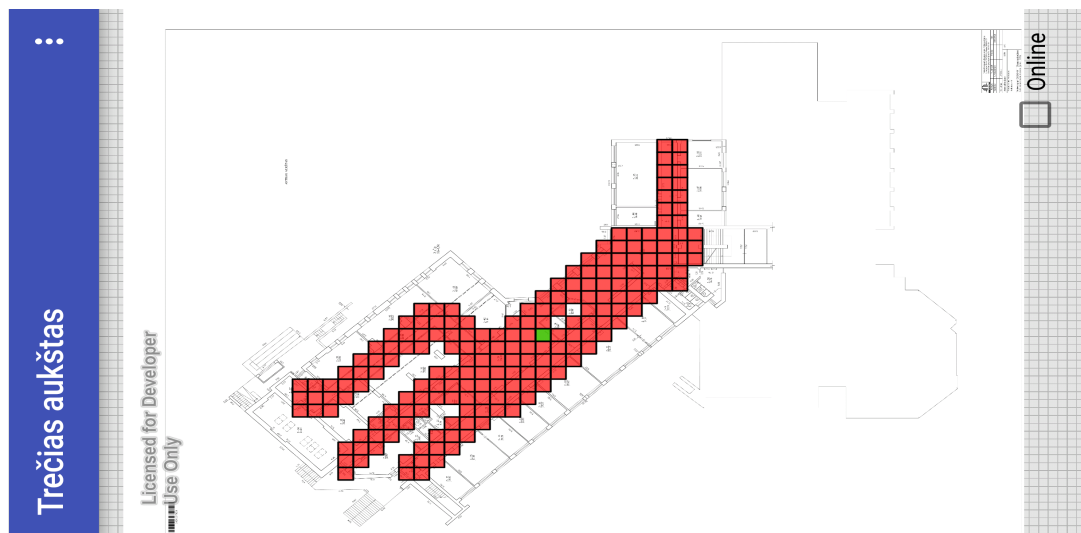
8.3.2. Padėties nustatymo programėlės V2 versija

Padėties nustatymo programėlės V2 versiją papildėme naujais komponentais. Ši versija sugeba fiksuoti ne tik WiFi, bet ir barometro bei magnetinio lauko rodmenis. Surinktus rodmenis siunčia į serverį. V2 versijoje pakeistas žemėlapis, bei padėties nustatymo algoritmas. Padaryta galimybė naudoti padėties nustatymo algoritmą tiek serverio pagalba, tiek lokaliai programėlėje. Taip pat padaryta galimybė surinktus WiFi bei barometro duomenis kalibruoti, kad šie labiau atitiktų apmokymo duomenis ir duotų geresnį padėties nustatymo procentą. V2 versija buvo naudota kaip įrankis testuojant aukšto nustatymą patalpose.

Programėlės veikimo principas. Programėlės veikimo principas versijoje V2 išliko panašus kaip ir V1 versijoje (žr. 8.3.1 skyrių). Programėlės paleidimo metu importuojama duomenų bazė su WiFi ir barometro pradiniais apmokymo duomenimis, parenkamas paskutinis nustatytas aukštas

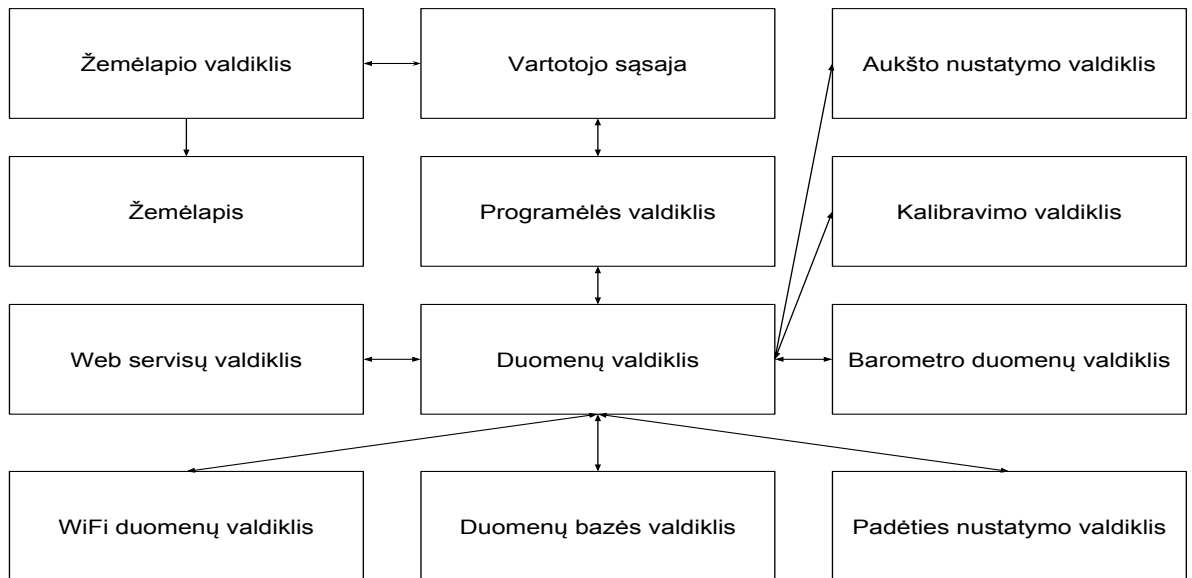
ir pradedamas WiFi bei barometro rodmenų skenavimas. Nuskenuoti duomenys siunčiami į serverį arba apdorojami lokaliai programėlėje. Jei duomenys siunčiami į serverį, tuomet serverį atliekamas padėties nustatymas, kitu atveju padėties nustatymas atliekamas programėlėje. Padėties nustatymui naudojamas Gauso-Bayeso metodas. Nustatyta padėtis pažymima žaliu kvadratu.

Žemiau pateikiamas padėties nustatymo programėlės ekranvaizdis.



23 pav. Padėties nustatymo programėlės V2 versijos ekranvaizdis

Pateiktame padėties nustatymo programėlės ekranvaizdyje (žr. 23 pav.) žaliu kvadratu žymimas rastas artimiausias poligonas atlikus padėties nustatymą Gauso-Bayeso metodu. Palyginus padėties nustatymo programos ekranvaizdį (žr. 23 pav.) su duomenų rinkimo programėlės V3 ekranvaizdžiu (žr. 19 pav.) matyti, kad šių programėlių vartotojo sąsajos labai panašios. Šių programėlių vartotojo sąsajos skiriasi tuo, kad padėties nustatymo programėlėje žalias kvadratas žymi artimiausią surastą poligoną, o ne poligonus, kuriuose surinkti pradiniai WiFi ir barometro rodmenys. Žemiau pateikiama padėties nustatymo programos V2 versijos blokinė schema.



24 pav. Padėties nustatymo programėlės V2 versijos blokinė schema

Palyginus padėties nustatymo programėlės V2 (žr. 24 pav.) ir V1 žr. 22 pav.) blokines schemas matyti, kad V2 programėlėje buvo prijungti penki nauji valdikliai: žemėlapių, web servisų, barometro duomenų, kalibravimo, aukšto nustatymo. Kiti programėlės komponentai panaudoti iš V1 versijos. Iš viso padėties nustatymo programėlės V2 versijos struktūrą sudaro dvylika pagrindinių komponentų (žr. 22 pav.) Detalesnis naujų komponentų aprašymas pateiktas žemiau.

Žemėlapių valdiklis. Atsakingas už žemėlapių ir poligono duomenų nuskaitymą iš privačios programėlės direktorijos ir šių failų saugojimą laikinoje atmintyje. Žemėlapių valdiklis nupiešia poligonus atitinkamuose žemėlapiuose. Taip pat šis valdiklis atlieka automatinius žemėlapių sukeitimus, kai nustatoma padėtis kitame aukšte. Šis valdiklis pasiekiamas tik iš vartotojo sąsajos.

Barometro duomenų valdiklis. Naudoja išmanaus telefono sensorius ir jų pagalba išgauna barometro altitudės reikšmes. Išgautas altitudės reikšmes saugo laikinoje atmintyje. Šis valdiklis bendrauja tik su duomenų valdikliu ir šiam perduoda naujausias išgautas altitudės reikšmes.

Aukšto nustatymo valdiklis. Šis valdiklis atsakingas už aukšto nustatymo algoritmų testavimą naudojantis barometro rodmenimis. Šio valdiklio pagalba atlikti Intervalų bei Gauso tikimybių tankio funkcijos metodų testavimai. Šis valdiklis bendrauja tik su duomenų valdikliu ir iš šio valdiklio gauna barometro rodmenis. Duomenų valdiklis pradinis apmokymo barometro duomenis surenka iš duomenų bazės valdiklio, o naujausias barometro rodmenis gauna iš barometro valdiklio.

Kalibravimo valdiklis. Atsakingas už komponentų C ir M reikšmių išgavimą naudojant mažiausių kvadratų metodą. Šis valdiklis išgauna C ir M komponentų reikšmes barometro altitudės ir WiFi signalo stiprumo rodmenimis. Kalibravimo valdikliui perduodami pradiniai apmokymo

duomenys bei realiu laiku išgauti barometro arba WiFi signalo stiprumo rodmenys. Kalibravimo valdiklis reikalingus duomenis gauna iš duomenų valdiklio. Sugeneruotos C ir M komponentės perduodamos duomenų valdikliui, o duomenų valdiklis komponentes išsaugo programėlės nustatymuose.

Web servisų valdiklis. Atsakingas už duomenų perdavimą į serverį, bei atsakymo iš serverio apdorojimą. Web servisų valdiklis atlieka dvi užklausas: padėties nustatymo ir kalibravimo. Šioms užklausoms reikalingus parametrus perduoda duomenų valdiklis. Web servisų valdiklis gautus duomenis iš serverio apdorojo naudojantis GSON biblioteka.

Šio darbo metu sukurta padėties nustatymo programėlės $V2$ versijoje įgyvendinti visi anksčiau išvardinti funkcionalumai. Šios programėlės pagrindinė užduotis nustatyti padėtį patalpose naudojantis apmokymo duomenimis išsaugotais lokaliai programėlėje arba serveryje. Šioje programėlės versijoje taip pat fiksuojami magnetinio lauko rodmenys, tačiau šie duomenys nenaudojami padėties nustatymui.

9. Tyrimai

9.1. Poligonų dydžio ir duomenų matavimo trukmės tyrimas

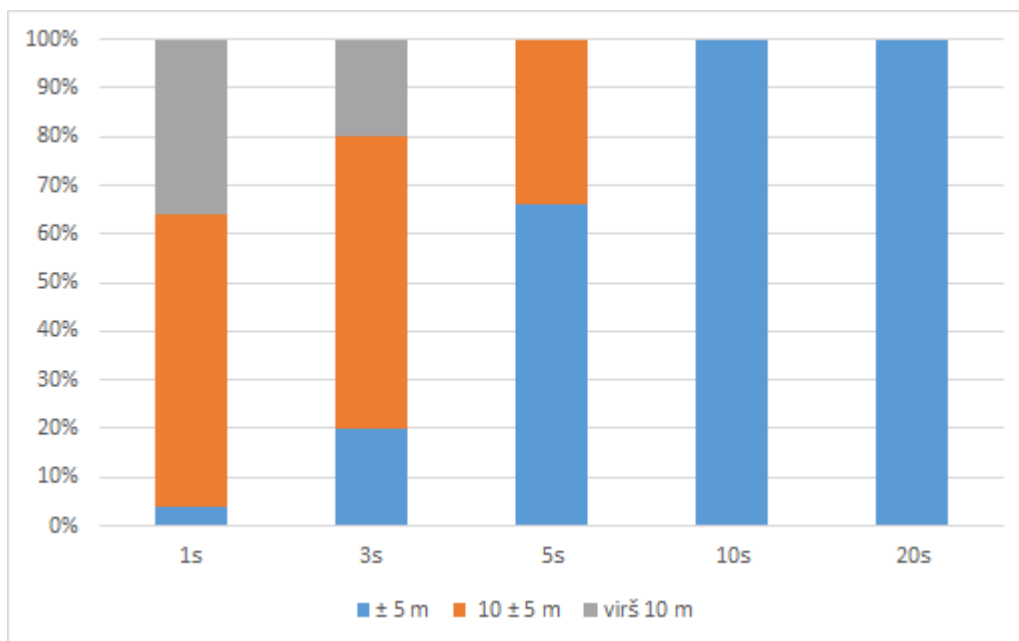
Kaip prieš tai minėjome, lokalizacijos uždavinius išskyrėme į padėties nustatymą viename aukšte ir aukšto nustatymą. Pirmiausia atliktas tyrimas, lokalizacijos nustatymo viename aukšte parametrus nustatyti. Tai poligonų dydžio bei duomenų matavimo trukmės - parametrų, lemiančių padėties nustatymo tikslumą, tyrimas. Tyrimas atliktas, uždaroje 40m ilgio 2m pločio patalpoje, su 90 laipsnių tokio paties pločio 10m ilgio atsišakojimu. Matavimai atlikti trijų aukštų pastato, kuriame išdėstyta 13 WiFi prieigos taškų, antrame aukšte. Šio tyrimo metu padėties nustatymo programėlėje naudotas Euklido atstumų 3.1 lokalizacijos algoritmas.

Poligonų dydžio tyrimo metodika. Uždaros patalpos plotas suskirstomas į tam tikro dydžio poligonus. Duomenų rinkimo programėle surenkami ir į duomenų bazę įrašomi kiekvieno poligono centro WiFi atspaudai. Tada naudojant padėties nustatymo programėlę su prieš tai surinktais WiFi atspaudų duomenimis, kiekviename poligone atliekama po 10 matavimų, registruojant lokalizacijos algoritmo rezultatus. Tyrimas atliktas naudojant 2, 4, 6, 8 bei 10 metrų atstumus tarp poligonų centrų.

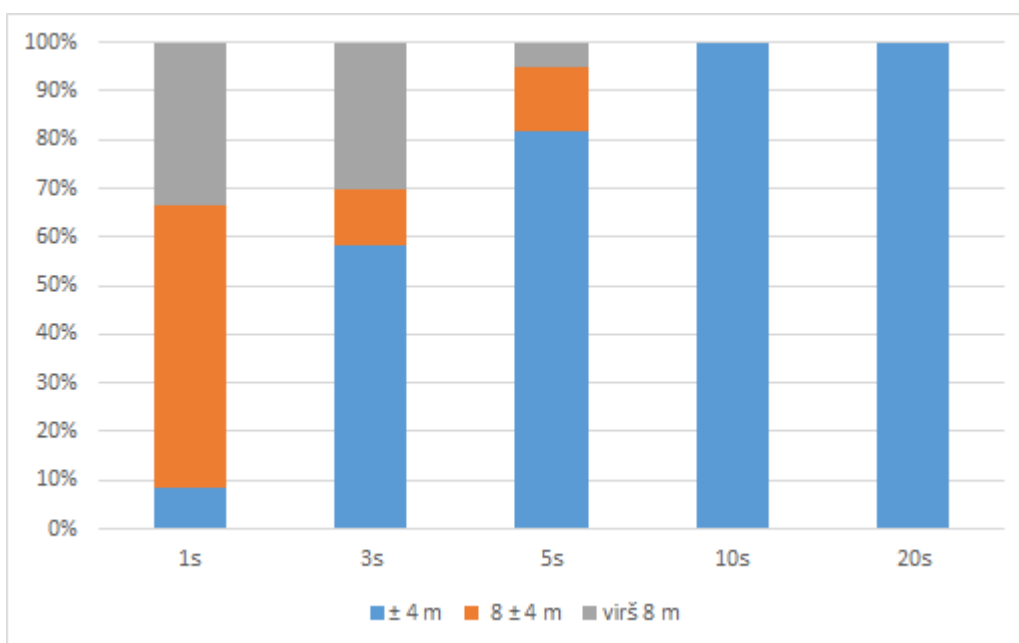
Duomenų matavimo trukmės tyrimo metodika. Duomenų matavimo/rinkimo trukmės tyrimas atliktas duomenų rinkimo programėle, kiekviename poligone, WiFi atspaudų duomenis renkant ir jų vertes vidurkinant skirtingais laiko intervalais. WiFi duomenys buvo renkami 1, 3, 5, 10 bei 20 sekundžių intervalais, kiekvieno poligono centre. Tada naudojant šiuos duomenis su padėties nustatymo programėle, kiekviename poligone atliekama po 10 matavimų, registruojant lokalizacijos algoritmo rezultatus. Tyrimas atliktas visų dydžių poligonuose, naudojant WiFi atspaudų duomenis su visomis matavimų rinkimo/vidurkinimo trukmėmis.

Rezultatai. Atliktų prieš tai aprašytų poligonų dydžio ir WiFi duomenų vidurkinimo/rinkimo trukmės tyrimų rezultatai pateikti kaip padėties nustatymo paklaidos, esant skirtingiems poligonų bei matavimų vidurkinimo intervalams.

Kaip matome iš 25 paveikslo, kai atstumas tarp poligonų centrų yra 10m (poligonų išmatavimai 10m x 2m) lokalizavimo algoritmas naudodamas 10 ir 20 sekundžių vidurkintus WiFi atspaudų matavimus 100% matavimų nurodė teisingą poligoną. Naudojant 5 sekundžių matavimus 66% matavimų nurodė teisingą poligoną, o 34% matavimų nurodė gretimą poligoną. Naudojant 3 sekundžių matavimus 20% matavimų nurodė teisingą poligoną, 60% matavimų nurodė gretimą poligoną, o 20% matavimų nurodė tolimesnį nei gretimą poligoną. Naudojant 1 sekundės matavimus 4% matavimų nurodė teisingą poligoną, 60% matavimų nurodė gretimą poligoną, o 36% matavimų nurodė tolimesnį nei gretimą poligoną.



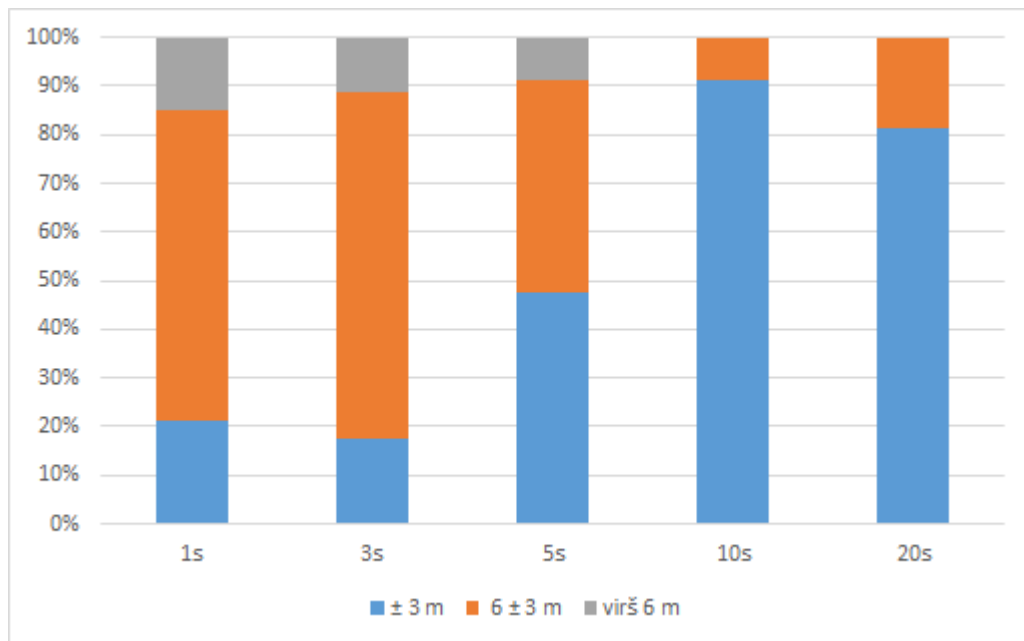
25 pav. Padėties nustatymo paklaida, kai atstumas tarp poligonų centrų - 10m



26 pav. Padėties nustatymo paklaida, kai atstumas tarp poligonų centrų - 8m

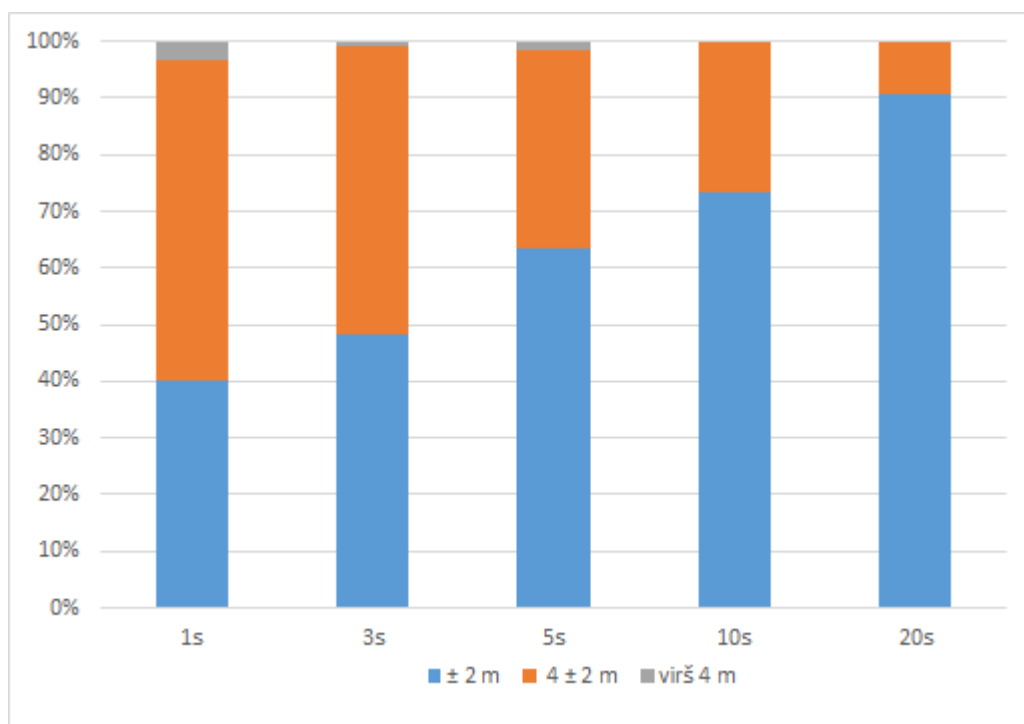
Kaip matome iš 26 paveikslo, kai atstumas tarp poligonų centrų yra 8m (poligonų išmatavimai 8m x 2m) lokalizavimo algoritmas naudodamas 10 ir 20 sekundžių vidurkintus WiFi atspaudų matavimus 100% matavimų nurodė teisingą poligoną. Naudojant 5 sekundžių matavimus 81,67% matavimų nurodė teisingą poligoną, o 13,33% matavimų nurodė gretimą poligoną. Naudojant 3 sekundžių matavimus 58,33% matavimų nurodė teisingą poligoną, 11,67% matavimų nurodė gretimą poligoną, o 30% matavimų nurodė tolimesnį nei gretimą poligoną. Naudojant 1 sekundės matavimus 8,33% matavimų nurodė teisingą poligoną, 58,34% matavimų nurodė gretimą poligoną, o 33,33% matavimų nurodė tolimesnį nei gretimą poligoną.

Kaip matome iš 27 paveikslo, kai atstumas tarp poligonų centrų yra 6m (poligonų išmatavimai 6m x 2m) lokalizavimo algoritmas naudodamas 20 sekundžių vidurkintus WiFi atspaudų mata-



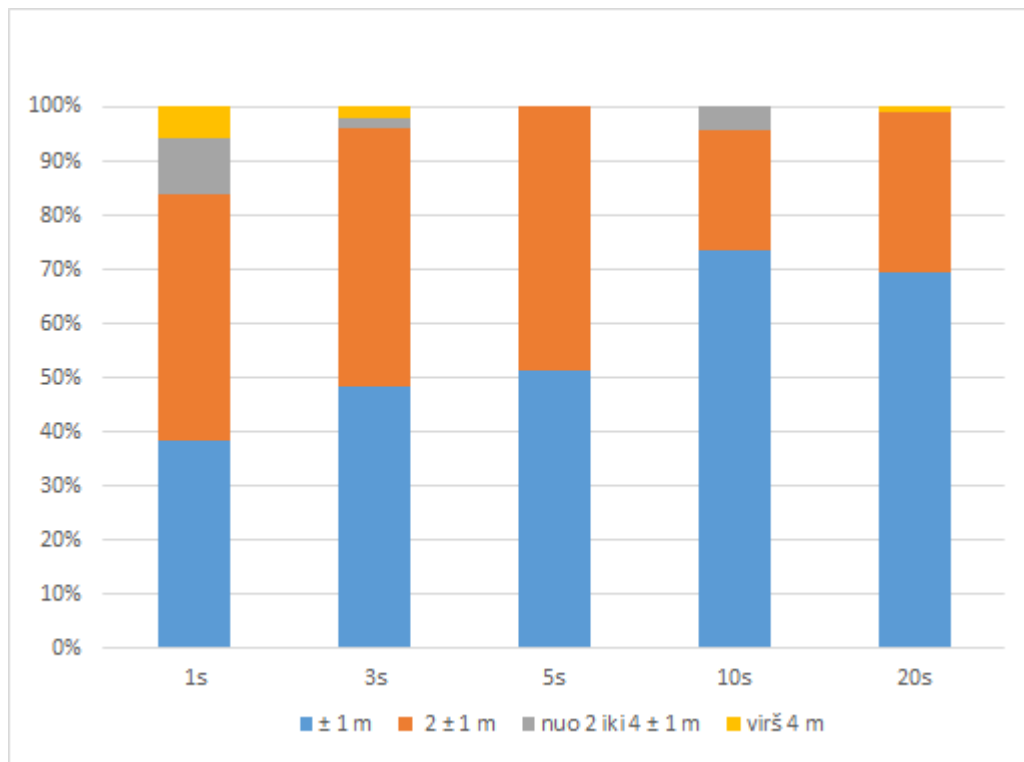
27 pav. Padėties nustatymo paklaida, kai atstumas tarp poligonų centrų - 6m

vimus 81,25% matavimų nurodė teisingą, o 18,75% gretimą poligoną. Naudojant 10 sekundžių atspaudų matavimus 91,25% matavimų nurodė teisingą, o 8,75% gretimą poligoną. Naudojant 5 sekundžių matavimus 47,5% matavimų nurodė teisingą poligoną, 43,75% matavimų nurodė gretimą, o 8,75% tolimesnį nei gretimą poligoną. Naudojant 3 sekundžių matavimus 17,5% matavimų nurodė teisingą poligoną, 71,25% matavimų nurodė gretimą poligoną, o 11,25% matavimų nurodė tolimesnį nei gretimą poligoną. Naudojant 1 sekundės matavimus 21,25% matavimų nurodė teisingą poligoną, 63,75% matavimų nurodė gretimą poligoną, o 15% matavimų nurodė tolimesnį nei gretimą poligoną.



28 pav. Padėties nustatymo paklaida, kai atstumas tarp poligonų centrų - 4m

Kaip matome iš 28 paveikslo, kai atstumas tarp poligonų centrų yra 4m (poligonų išmatavimai 4m x 2m) lokalizavimo algoritmas naudodamas 20 sekundžių vidurkintus WiFi atspaudų matavimus 90,83% matavimų nurodė teisingą, o 9,17% gretimą poligoną. Naudojant 10 sekundžių atspaudų matavimus 73,33% matavimų nurodė teisingą, o 26,67% gretimą poligoną. Naudojant 5 sekundžių matavimus 63,33% matavimų nurodė teisingą poligoną, 35% matavimų nurodė gretimą, o 1,67% tolimesnį nei gretimą poligoną. Naudojant 3 sekundžių matavimus 48,33% matavimų nurodė teisingą poligoną, 50,83% matavimų nurodė gretimą poligoną, o 0,84% matavimų nurodė tolimesnį nei gretimą poligoną. Naudojant 1 sekundės matavimus 40% matavimų nurodė teisingą poligoną, 56,67% matavimų nurodė gretimą poligoną, o 3,33% matavimų nurodė tolimesnį nei gretimą poligoną.



29 pav. Padėties nustatymo paklaida, kai atstumas tarp poligonų centrų - 2m

Kaip matome iš 29 paveikslo, kai atstumas tarp poligonų centrų yra 2m (poligonų išmatavimai 2m x 2m) lokalizavimo algoritmas naudodamas 20 sekundžių vidurkintus WiFi atspaudų matavimus 69,57% matavimų nurodė teisingą, o 29,56% gretimą poligoną bei 0,87% matavimų nurodė tolimesnį nei gretimą poligoną. Naudojant 10 sekundžių atspaudų matavimus 73,48% matavimų nurodė teisingą, o 22,17% gretimą poligoną, 4,35% matavimų nurodė sekantį gretimą poligoną. Naudojant 5 sekundžių matavimus 51,3% matavimų nurodė teisingą poligoną, 48,7% matavimų nurodė gretimą poligoną. Naudojant 3 sekundžių matavimus 48,26% matavimų nurodė teisingą poligoną, 47,83% matavimų nurodė gretimą poligoną, 1,74% matavimų nurodė sekantį gretimą poligoną o 2,17% matavimų nurodė dar tolimesnį poligoną. Naudojant 1 sekundės matavimus 38,26% matavimų nurodė teisingą poligoną, 45,66% matavimų nurodė gretimą poligoną, 10,43% nurodė sekantį gretimą poligoną, o 5,65% matavimų nurodė tolimesnį nei gretimą poligoną.

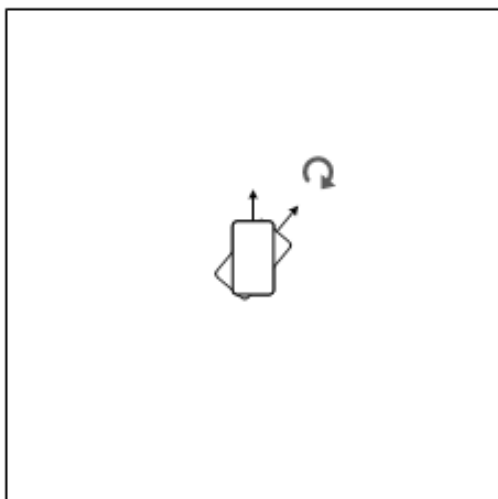
Rezultatų aptarimas ir pastebėjimai. Iš gautų rezultatų matome, kad lokalizacijos paklaida mažėja didinant WiFi atspaudų matavimo vidurkinimo trukmę iki 10 sekundžių, matavimus vidur-

kinant ilgiau (20s) lokalizacijos paklaida lyginant su 10 sekundžių vidurkinimu sumažėja nežymiai. Taip pat matome, lokalizacijos tikslumas didėja mažinant poligonų dydį. Didžiausias tikslumas pasiektas naudojant 2m dydžio poligonus.

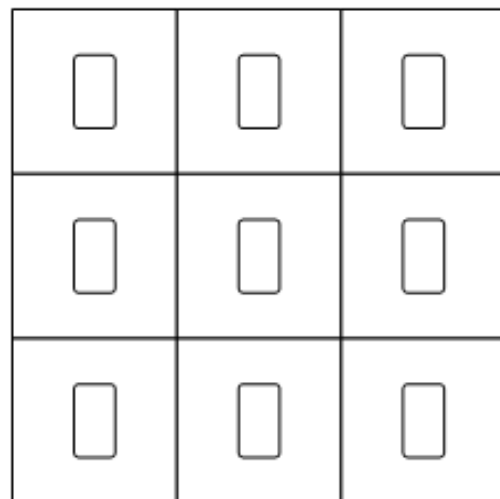
Atlikus tyrimą pastebėtas Euklido algoritmo trūkumas (kai lyginami išmatuoti duomenys su esančiais duombazėje, jeigu duombazėje WiFi atspaudas yra, o lokalizavimo metu to WiFi signalo neranda, tai lokalizacijos rezultatas iškreipiamas, nes nėra sumuojamas Euklido atstumas iki to prieigos taško atspaudu). Taip pat pastebėta, kad išmaniajame telefone WiFi modulis matavimus atlieka kas 0,5 - 4 sekundes, dėl to matavimai su trumpais vidurkinimo intervalais nebuvo tokie tikslūs kaip vidurkinant ilgiau. Dėl šio duomenų atnaujinimo užlaikymo kuris siekia iki 4 sekundžių, padėties nustatymas judant lemia dar didesnes paklaidas.

9.2. Skirtingais telefonais surinktų duomenų tyrimas.

Skirtingais telefonais surinktų WiFi RSSI duomenų tyrimo metodika. Skirtingais telefonais surinktų WiFi RSSI duomenų tyrimas atliktas 2m skersmens poligone, naudojant 10 sekundžių duomenų vidurkinimą. Duomenys buvo renkami dviem išmaniaisiais telefonais: Samsung S5 ir Samsung S6, dviem būdais.



(a) Pirmasis būdas

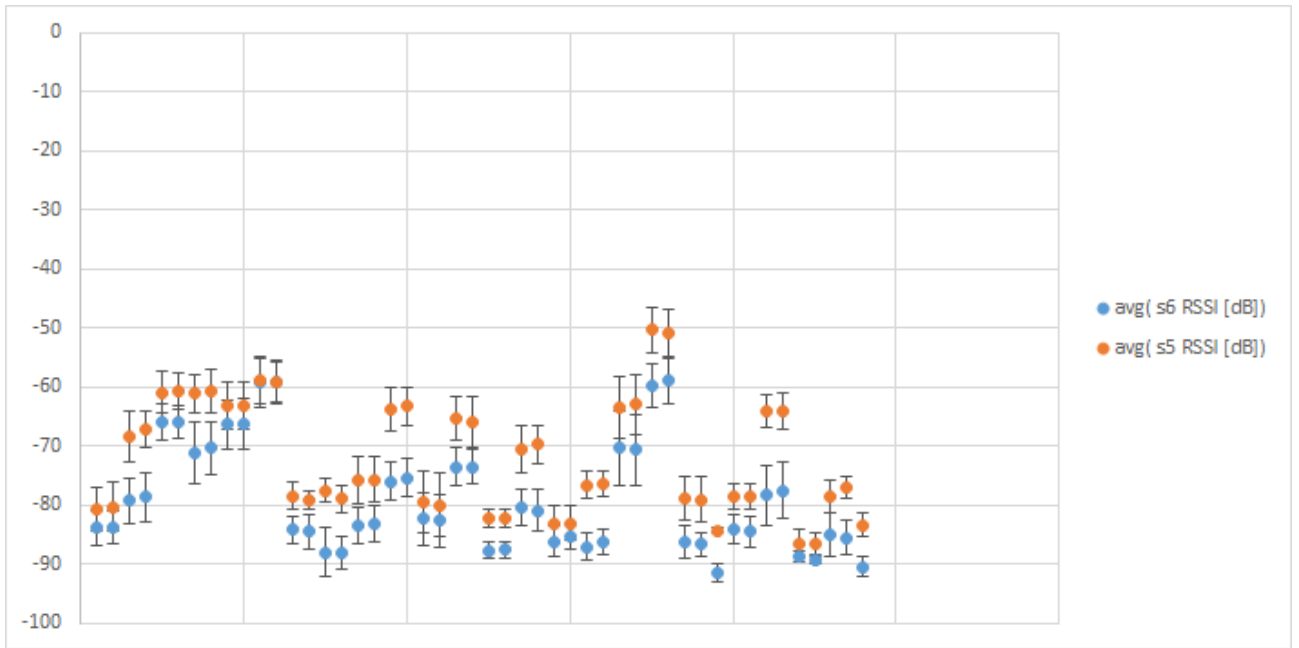


(b) Antrasis būdas

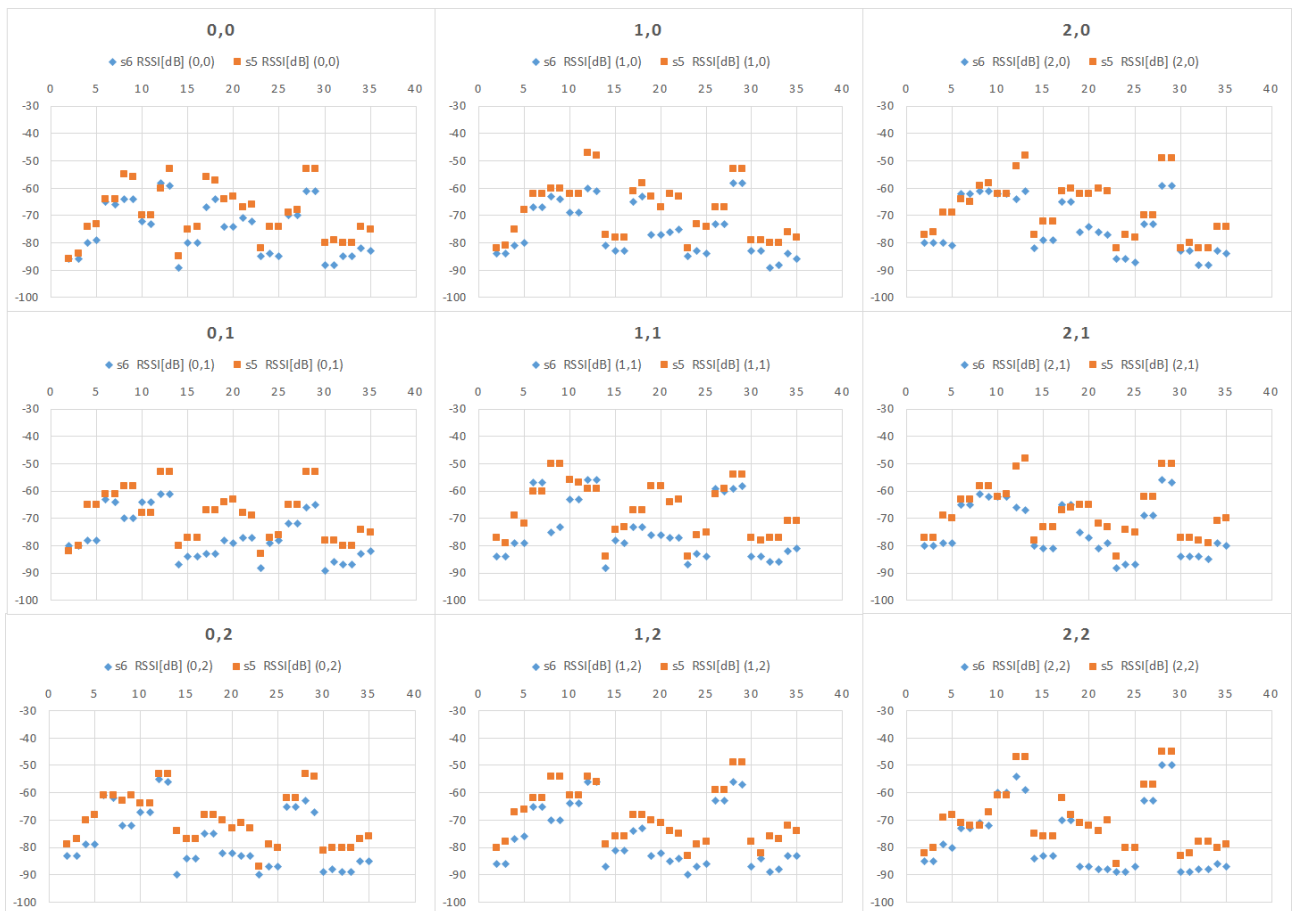
30 pav. Naudojami duomenų rinkimo metodai

Pirmasis - duomenys renkami stovint poligono centre, sukantis kas 45 laipsnius. Antrasis - 9 poligono vietose, laikant telefoną tokioje pačioje padėtyje. Telefonai matavimo metu buvo laikomi atsukti ekranu į viršų, vieno metro aukštyje. Šio tyrimo tikslas, ištirti, kaip skiriasi WiFi RSSI matavimai tarp skirtingų telefonų bei ištirti kokiose ribose kinta tuo pačiu telefonu matuojami duomenys priklausomai nuo įrenginio padėties poligone.

Rezultatai. 31 paveiksle atvaizduoti priimtų WiFi signalų stiprumo lygiai iki tų pačių prieigos taškų, surinkti dviem įrenginiais, telefoną laikant vienoje vietoje, pastoviam aukštyje, keičiant jo padėtį kaip pavaizduota 30a paveiksle. Atvaizduoti matavimų standartiniai nuokrypiai parodo, priimto WiFi signalo stiprumo lygio kitimą priklausomai nuo telefono padėties tame pačiame taške. Kaip matome Samsung S5 išmanusis telefonas fiksuoja aukštesnio lygio WiFi signalus nei Samsung S6 išmanusis telefonas. Tačiau verta paminėti, kad S6 išmanusis telefonas užfiksavo daugiau žemesnio lygio signalų, kurių dėl duomenų atvaizdavimo aiškumo neįtraukėme į 31 paveikslą.



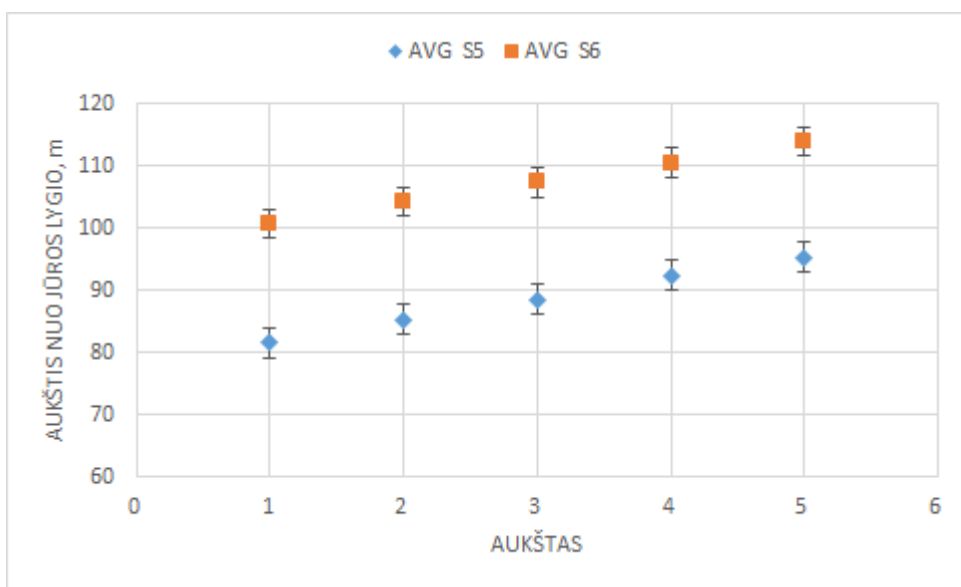
31 pav. Samsung S5 ir S6 telefonais išmatuotų WiFi RSSI verčių palyginimas, matuojant pirmuoju būdu



32 pav. Samsung S5 ir S6 telefonais išmatuotų WiFi RSSI verčių palyginimas, matuojant antruoju būdu

32 paveiksle atvaizduoti priimtų WiFi signalų stiprumo lygiai iki tų pačių prieigos taškų, surinkti dviem įrenginiais, telefoną laikant pastoviam aukštyje, pastovios padėties, bet keičiant vietą poligone, taip kaip pavaizduota 30b paveiksle. Kaip matome, matavimai iki tų pačių WiFi prieigos taškų skiriasi priklausomai nuo padėties poligone bei nuo įrenginio, kuriuo surinkti duomenys.

Barometrų matavimų palyginimas. Barometrų matavimai gauti vidurkinant (10 ar 15 sec?) matavimus tam tikro aukšto laiptinės aikštelėse. Fiksuojamos ne slėgio vertės, bet telefonų nurodomas aukštis virš jūros lygio.



33 pav. Samsung S5 ir S6 telefonų barometrais išmatuotų, aukščių verčių palyginimas

10 lentelė. Barometro matavimų palyginimas

Tarp aukštų	Aukštis, m (S5)	Aukštis, m (S6)	Skirtumas, m
1 ir 2	3.7577	3.5336	0.2240
2 ir 3	3.1825	3.1066	0.0759
3 ir 4	3.7979	3.2195	0.5783
4 ir 5	2.9558	3.4116	0.4558

Kaip matome iš 33 paveikslo, absoliučios, aukščio vertės, išmatuotos skirtingais telefonais ženkliai skiriasi, tačiau pokytis yra beveik pastovus. Iš 10 lentelės matome, kad skirtingais telefonais išmatuoti skirtumai tarp aukštų yra apylygiai, todėl šiuos pokyčius atlikus matavimų kalibravimą, būtų galima panaudoti aukšto nustatymui pastate.

Rezultatų aptarimas ir pastebėjimai. Iš gautų rezultatų matome, kad Samsung S5 išmanusis telefonas fiksuoja aukštesnio lygio WiFi signalus nei Samsung S6 išmanusis telefonas, tai matosi 31 ir 32 paveiksluose. Taip pat Samsung S6 išmanusis telefonas fiksuoja daugiau žemesnio lygio signalų nei Samsung S5 - taigi skiriasi įrenginių WiFi modulių "jautrumas". Remiantis šio tyrimo rezultatais, vykdant padėties nustatymą WiFi atspaudų metodu, reikia atsižvelgti į skirtingais įrenginiais surinktų matavimų skirtumus - tai galima pasiekti atliekant kalibravimą. Taip pat iš 31 ir 32

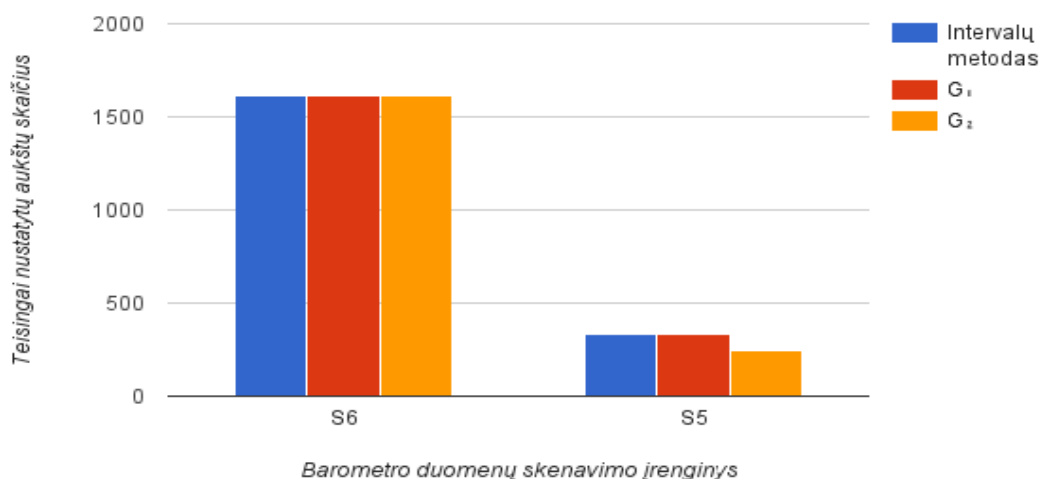
paveikslų matome, kad matavimai iki tų pačių WiFi prieigos taškų skiriasi tiek keičiant telefono vietą poligone, tiek keičiant jo padėtį vienoje vietoje. Šiuos matavimų svyravimus padėties nustatymo algoritmas, taip pat turi įvertinti. Remiantis barometrų matavimų palyginimo rezultatais nustatėme, kad barometrų matavimus galima naudoti aukšto nustatymui pastate, jei naudojamos ne absoliučios, o apskaičiuotos pokyčių vertės - tai galima pasiekti atliekant kalibravimą.

9.3. Aukšto nustatymo tyrimas

Aukšto nustatymo tyrimo metodika. Aukšto nustatymui taikyta tokia pati metodika, kaip ir padėties nustatymui aukšte. Tiesiog WiFi duomenys priskirti ne poligonui, o atitinkamo aukšto laiptinės aikštelėi. Tačiau toks bandymas nepavyko, net atlikus keletą Euklido atstumų skaičiavimu paremto algoritmo modifikacijų. Nuspręsta aukštų nustatymui naudoti papildomus, ne tik WiFi duomenis, bet ir barometro.

Tyrimo metu buvo surinkti barometro altitudės rodmenys penkių aukštų pastate naudojant S6 įrenginį. Kiekviename aukšte surinkta po 325 pradinius apmokymo rodmenis. Iš viso sistemos apmokymui surinkti 1625 rodmenys. Taip pat surinktas toks pats kiekis altitudės rodmenų su S5 įrenginiu. Šio tyrimo metu aukšto nustatymui panaudoti Intervalų (žr. 4.2 skyrių) ir Gauso tikimybių tankio funkcijos (toliau tekste G_1) (žr. 4.3 skyrių) metodai ir kalibravimo algoritmas (žr. 5.1 skyrių). Taip pat aukšto nustatymui panaudota modifikacija išreiškiant Gauso tankio tikimybės funkcija logaritmų suma (toliau tekste G_2) (priežastys aprašytos 9.4.2 poskyryje, prie 4-to paragrafo). Žemiau pateikti atlikti aukšto nustatymo tyrimai ir rezultatai.

Aukšto nustatymas pastate naudojant skirtingus įrenginius. Šiame tyrime panaudojome S6 ir S5 įrenginiais surinktus barometro duomenis. Žemiau pateikiamas grafikas kaip pavyko nustatyti aukštą pastate, naudojantis aukšto nustatymo metodais su tuo pačiu įrenginiu (S6), su kuriuo apmokyta sistema ir su dar vienu įrenginiu (S5).



34 pav. Aukšto nustatymas pastate naudojant skirtingus įrenginius.

11 lentelė. Aukšto nustatymo metodų rezultatų palyginimas

Metodas	Teisingi nustatymai, % (S6)	Teisingi nustatymai, % (S5)
Intervalų metodas	99,4461	20,5538
G_1	99,4461	20,5538
G_2	99,5097	15,3846

Kaip matome (žr. 34 pav.) aukšto nustatymas veikia kur kas geriau naudojant tą patį įrenginį, kuriuo buvo apmokyta prototipinė padėties nustatymo sistema, nei su kitu įrenginiu. Taip nutiko, nes skirtingais įrenginiais surinkti duomenys ženkliai skiriasi (žr. 33 pav.). Tikėtina, kad duomenys gauti naudojant S6 įrenginį po tam tikro laiko rodys blogesnius rezultatus dėl aplinkos pokyčių. [3]

Kaip matome rezultatų 11 lentelėje visi trys padėties nustatymo algoritmai nustatė teisingą aukštą daugiau nei 99% tikslumu. Tačiau šie rezultatai gauti atliekant testą su tuo pačiu įrenginiu kaip ir apmokyta padėties nustatymo sistema. Taigi gautas 99% tikslumas parodo maksimalias aukšto nustatymo metodų galimybes, nes šis tikslumas pasiektas naudojant idealias sąlygas, tai yra testas atliktas su tuo pačiu įrenginiu su kuriuo surinkti pradiniai sistemos apmokymo duomenys. Šiame grafike nesimato, tačiau, verta paminėti, kad testo metu nustatyti klaidingi aukštai yra vienu aukštu žemiau arba vienu aukščiau nei tikrasis aukštas.

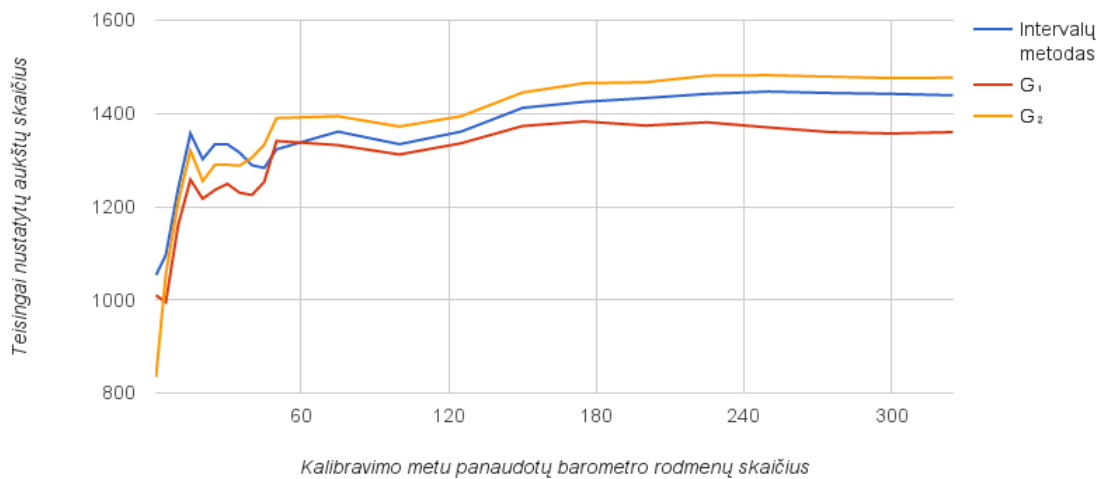
Atlikus aukšto nustatymo testą su S5 įrenginiu surinktais duomenimis matyti, kad aukšto nustatymo rezultatai ženkliai prastesni. Tai galima paaiškinti tuo, kad barometro reikšmės surinktos tuose pačiuose aukštuose kaip ir su įrenginiu S6 skiriasi 20 metrų (žr. 33 pav.) Taip pat verta paminėti, kad testo metu klaidingai nustatyti aukštai buvo nutolę daugiau nei per vieną aukštą nuo tikrojo aukšto.

Aukšto nustatymų algoritmų panašų tikslumą galima paaiškinti tuo, kad visi algoritmai remiasi vidutinėmis altitudės reikšmėmis. Atlikus aukšto nustatymo testą, iš gautų rezultatų, matyti, kad reikia išspręsti kalibracijos problemą, nes be kalibracijos, aukšto nustatymas veikia tiksliai tik su tuo pačiu įrenginiu su kuriuo surinkti sistemos apmokymo rodmenys.

Aukšto nustatymas pastate atlikus kalibravimą. Šiame tyrime panaudojome tuos pačius altitudės duomenis surinktus su S6 ir S5 įrenginiais. Be to, panaudoti tie patys aukšto nustatymo metodai kaip ir aukščiau aprašytame tyrime. Tyrime papildomai panaudotas mažiausių kvadratų metodas (žr. 5.1 skyrių) C_1 ir C_2 komponenčių radimui. Apskaičiavus komponentes kiekviena altitudės A_n reikšmė prieš perduodant į aukšto nustatymo metodą sukalibruojama pagal šią formulę:

$$B_n = C_1 \cdot A_n + C_2 \quad (9.1)$$

B_n , tai sukalibruota altitudės reikšmė, kurią perduosime aukšto nustatymo metodui. Žemiau pateikiamas aukšto nustatymo algoritmų testas atlikus kalibravimą:



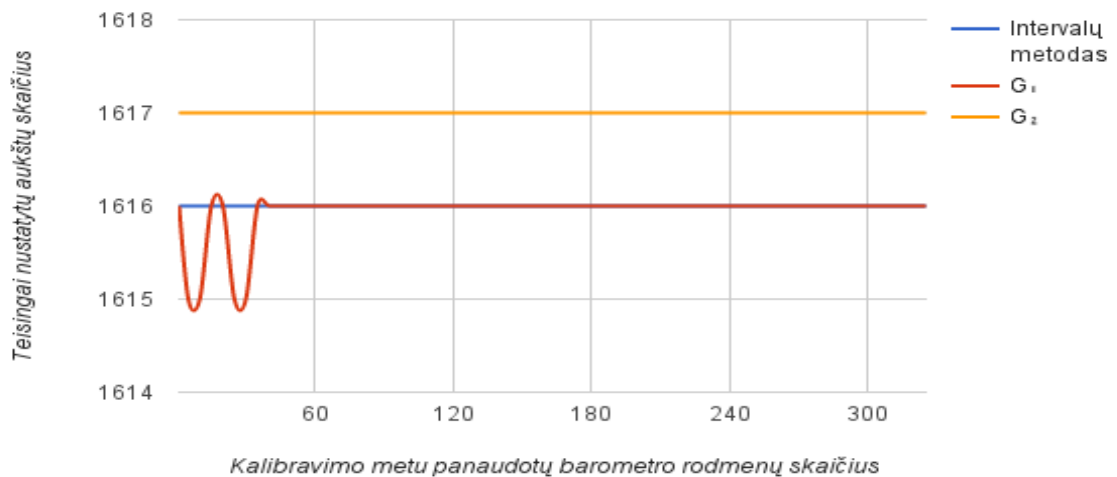
35 pav. Aukšto nustatymas atlikus kalibravimą S5 įrenginiu surinktiems barometro rodmenims.

12 lentelė. Aukšto nustatymo metodų palyginimas atlikus kalibravimą S5 įrenginiu surinktiems rodmenims

Metodas	Teisingi nustatymai prieš kalibravimą, %	Teisingi nustatymai po kalibravimo, %
Intervalų metodas	20,5538	89,0461
G_1	20,5538	85,1065
G_2	15,3846	91,2000

Kaip matome (žr. 35 pav.) kalibravimo komponentčių radimui turi įtakos kiek barometro rodmenų buvo skirta apskaičiuojant C_1 ir C_2 komponentes. Taip pat galime pastebėti, kad padėties nustatymo algoritmai blogiausiai aukšto nustatymo rezultatus demonstravo, kai komponentčių C_1 ir C_2 radimui buvo skiriama 1 - 5 barometro reikšmės. Be to, galima pastebėti, kad didžiausias aukšto nustatymo tikslumo šuolis įvyko, kai komponentčių C_1 ir C_2 reikšmių apskaičiavimui buvo skiriama 15 - 20 barometro rodmenų. Toliau didinant barometrų rodmenų kiekį apskaičiuojant C_1 ir C_2 komponentes aukšto nustatymo tikslumas ženkliai nebesikeitė, o pasiekus 150 rodmenų praktiškai sustojo didėti.

Palyginus 12 lentelėje pateiktus duomenis, matyti, kad kalibravimas aukšto nustatymo metodų tikslumą pagerino 65 - 76 procentais. Taip pat galima pastebėti, kad G1 algoritmas demonstruoja prastesnius rezultatus nei šio metodo modifikuota versija G2. Taip yra todėl, kad jei gaunama altitudės reikšmė nutolusi nuo μ , tai G1 metodo gaunama reikšmė artėja prie nulio greičiau nei G2. Taip pat verta paminėti, kad atlikus kalibravimą klaidingai nustatyti aukštai buvo vienu aukštu žemiau arba aukščiau, nebeprisitaikydavo atvejų, kad neteisingas aukštas nutolęs per du aukštus. Atlikus kalibravimą matyti, kad S5 įrenginiu surinktų barometro rodmenų kalibravimas ženkliai padidino aukšto nustatymo tikslumą, tačiau neaišku ar kalibravimas neturi įtakos barometro rodmenims surinktais tuo pačiu įrenginiu su kuriuo buvo apmokyta sistema. Žemiau pateikiamas grafikas atvaizduojantis kalibravimo įtaką aukšto nustatymo tikslumui naudojant S6 įrenginiu surinktus barometro rodmenimis.



36 pav. Aukšto nustatymas atlikus kalibraciją S6 įrenginiu surinktiems barometro rodmenims.

Palyginus aukšto nustatymo grafiką be kalibracijos (žr. 34 pav.) ir su kalibracija (žr. 36 pav.), matyti, kad kalibracija aukšto nustatymo tikslumai beveik neturėjo. Tai reiškia, kad kalibravimui reikalingos C_1 ir C_2 komponentės įgyjo reikšmes $C_1 \sim 1$, $C_2 \sim 0$.

Tyrimo rezultatų aptarimas ir pastebėjimai. Remiantis šių tyrimų rezultatais galime teigti, kad barometro rodmenys tinkami aukšto nustatymui. Tačiau barometro rodmenų priklausomybė nuo aplinkos ir paties įrenginio, kuriuo renkami barometro rodmenys, todėl prieš aukšto nustatymo metodų vykdymą reikia vykdyti barometro rodmenų kalibravimą. Taip pat tyrimų rezultatai parodė, kad barometro rodmenų kalibravimas neturi neigiamos įtakos, tai yra nesumažina aukšto nustatymo tikslumo. Be to, sukalibravus barometro rodmenimis nebepasitaikydavo klaidos, kai nustatytas aukštas yra per du aukštus nuo tikrojo aukšto.

9.4. Lokalizacijos algoritmo paremto Gauso-Bajeso metodu tikslumo tyrimas

9.4.1. Tyrimo etapų aprašas

Gauso-Bajeso metodo naudojimo lokalizacijos algoritme pagrindumas. Pagrindinė sąlyga, kad lokalizacijos algoritmo pagrindu būtų naudojamas Gauso-Bajeso metodas - matavimai turi atitikti/būti artimi normaliajam (Gauso) skirstiniui. Remiantis [3][15][16] literatūros šaltiniais, mūsų sistemoje naudojami matavimai šią sąlygą patenkina.

Lokalizacijos algoritmo padėties nustatymo tikslumo tyrimo metodika. Pirmiausiai lokalizacijos algoritmo veikimas buvo tikrinamas, su dirbtiniais [*angl. artificial*] matavimų duomenimis. Dirbtiniai matavimų duomenys, buvo sugeneruoti naudojant realius surinktų atskaitinių duomenų parametrus 3.4. Tada buvo implementuoti skirtingi lokalizacijos algoritmo, paremto Gauso-Bajeso metodu, variantai. Tikslumo įvertinimas atliekamas lyginant ar lokalizavimo algoritmo prognozuojamas poligonas sutampa su poligonu, kurio dirbtinis matavimas yra paduodamas lokalizavimo algoritmui.

Po tyrimo su dirbtiniais duomenimis, geriausią tikslumą parodęs lokalizavimo algoritmo variantas buvo palygintas su keliais klasifikatoriais iš scikit-learn bibliotekos. Klasifikatoriai apmokomi su 80% sugeneruotų, dirbtinių duomenų, o testuojami su likusiais 20% duomenų. Testavimo duomenys vienodi, tiek geriausiam lokalizavimo algoritmo, paremtu Gauso-Bajeso metodu, variantui tiek ir klasifikatoriams. Taip pat geriausią tikslumą parodęs lokalizavimo algoritmo variantas buvo testuojamas su atskaitinių duomenų rinkimo metu surinktais neapdorotais [angl. raw] duomenimis.

9.4.2. Lokalizacijos algoritmo variantai ir jų rezultatai

Pirmasis lokalizacijos algoritmo variantas. Pirmasis lokalizacijos algoritmo variantas įgyvendintas, tiksliai pagal Gauso-Bajeso metodą aprašytą 3.4 skyriuje. Šio lokalizacijos algoritmo varianto tikslumas siekė vos 6%. Atlikus detalią gautų vėlesnių tikimybių $T(P_i|X)$ (žr.3.5 formulę) (toliau šiame skyriuje vadinamų bendrų tikimybių) analizę buvo pastebėta, kad dažniausiai bendrų tikimybių vertės yra gaunamos 0. Taip atsitinka todėl, kad atliekant 3.4 formulėje aprašytą sandaugą dažniausiai bent vienas šios sandaugos narys yra lygus nuliui, o to pasekoje bendroji tikimybė taip pat tampa lygi 0.

Antrasis lokalizacijos algoritmo variantas. Antrasis lokalizacijos algoritmo variantas įgyvendintas, pakeitus Gauso-Bajeso metode 3.4 formulėje aprašytą sandaugą, visų tos formulės sandaugos narių sudėtimi, t.y. 3.4 formulę pakeičiame į:

$$S_i = \sum_n g_n \quad (9.2)$$

kur S_i yra Gauso tikimybės tankio funkcijų verčių g_n sudėtis. Kadangi pirminės tikimybės $T(P_i)$ (žr.3.5 formulę) mūsų sistemos atveju visiems poligonams yra lygios, tai jas galime supaprastinti 3.5 formulėje. Taip pat 3.5 formulėje likęs vardiklis visiems poligonams P_i bus vienodas, o mus domina ne pati reikšmė, o didžiausia reikšmė iš visų poligonų reikšmių, todėl vardiklį galime taipogi panaikinti. Tokiu atveju gauname, kad lokalizacijos algoritmo rezultatas tai poligonas P_i , kurio S_i vertė yra didžiausia. Įgyvendinus tokį lokalizacijos algoritmo variantą pasiektas 50% tikslumas.

Trečiasis lokalizacijos algoritmo variantas. Trečiasis lokalizacijos algoritmo variantas įgyvendintas, Gauso-Bajeso metode 3.4 formulėje aprašytą sandaugą pakeitus, į sandaugos narių logaritmų sumą, t.y.: 3.4 formulę pakeičiame į:

$$L_i = \sum_n \log(g_n) \quad (9.3)$$

kur L_i yra bendros tikimybės logaritmas [angl. log likelihood] poligonui P_i . Taip pat dėl tų pačių priežasčių kaip kad, nurodytų antrojo lokalizacijos algoritmo varianto aprašyme, iš 3.5 formulės išspratiname pirminę tikimybę ir panaikiname vardiklį. Tokiu atveju gauname, kad lokalizacijos algoritmo rezultatas yra poligonas P_i , kurio vertė L_i [angl. maximum log likelihood] yra didžiausia. Šio lokalizacijos algoritmo varianto tikslumas pasiekė 69%. Atlikus likusių rezultatų analizę buvo pastebėta, kad daugumai poligonų L_i vertė yra minus begalybė. Taip atsitinka tokiu

atveju, kai nors vieno lokalizavimo metu gauto matavimo g_n vertė yra gaunama lygi 0. Tada prienos vienos minus begalybės rezultato pridėjus bet kokią vertę, bendros tikimybės logaritmas L_i bus lygus minus begalybei.

Ketvirtasis lokalizacijos algoritmo variantas. Ketvirtasis lokalizacijos algoritmo variantas įgyvendintas pakeitus Gauso tikimybės tankio funkcijos g_n skaičiavimą pagal (žr. 4.8 formulę). Vietoje 4.8 formulėje esančių Gauso tikimybės tankio funkcijos narių sandaugos skaičiuojame jos narių logaritmų sumą:

$$lg_n = -0.5 \cdot \log(2\pi\sigma^2) + \left(-0.5 \cdot \left(\frac{(b - \mu_n)^2}{\sigma^2} \right) \right) \quad (9.4)$$

kur lg_n tikimybės g_n logaritmas. Kaip matome 9.4 pirmasis sumos narys priklauso tik nuo σ^2 , o antrasis sumos narys priklauso nuo σ^2 ir μ . Atlikus tokius pakitimus, net esant situacijai, kai su matavimo metu išmatuota verte b (b šioje formulėje yra RSSI matavimas iki n -tojo SSID) gaunamas antrasis sumos narys lygus 0, lg_n netampa lygus 0, o lieka pirmojo sumos nario vertė. Šiame algoritmo variante 9.5 formulė, tampa:

$$L_i = \sum_n (lg_n) \quad (9.5)$$

kur L_i kaip ir trečiojo varianto atveju yra bendros tikimybės logaritmas [*angl. log likelihood*] poligonui P_i . Tolimesnė algoritmo dalis tokia pati kaip ir trečiojo lokalizacijos algoritmo atveju. Įgyvendinus tokį lokalizacijos algoritmo variantą, jo tikslumas testuojant su dirbtiniais duomenimis siekė 98%.

13 lentelė. Lokalizacijos algoritmo ir klasifikatorių tikslumo testų rezultatai

Pavadinimas	Tikslumas
IV Gauso-Bajeso LA variantas	98,1%
Atsitiktinio Miško klasifikatorius	97,3%
Sprendimų Medžio klasifikatorius	93,0%
Gauso-Naive Bajeso klasifikatorius	99,5%

Lokalizacijos algoritmo ir klasifikatorių tikslumo palyginimas 13 lentelėje pateikti tikslumo testo rezultatai. Su dirbtiniais duomenimis ištestuoti klasifikatoriai: Atsitiktinio Miško [*angl. Random Forest*] klasifikatorius, Sprendimų Medžio [*angl. Decision Tree classifier*] klasifikatorius, Gauso-Naive Bajeso klasifikatorius bei IV lokalizacijos algoritmo paremta Gauso-Bajeso metodu variantas. Kaip matome iš rezultatų, tarp klasifikatorių didžiausias tikslumas pasiektas naudojant Gauso-Naive Bajeso klasifikatorių. Kadangi šio klasifikatoriaus veikimo principas teoriškai yra toks kaip ir IV lokalizacijos algoritmo varianto, manome, kad pasirinktas teisingas sprendimas naudoti lokalizacijos algoritmą paremtą Gauso-Bajeso metodu.

Lokalizacijos algoritmo pritaikymas realiems matavimams. Atlikus IV lokalizacijos varianto testą su neapdorotais, realiais duomenimis. Pasiektas 88% tikslumas. Atlikus neteisingų algoritmo sprendimų analizę nustatyta, kad tada, kai algoritmo rezultatas buvo neteisingas, teisingo poligono P_i bendrosios tikimybės logaritmas L_i buvo tarp 5 didžiausių. Bendrųjų tikimybių logaritmų reikšmės L_i skaičiuojant Gauso-Bajeso metodu visada gaunamos neigiamos, didžiausia reikšmė, o tuo pačiu ir algoritmo sprendimas yra mažiausia skaitinė vertė su minuso ženklu (t.y. pvz.: $-15 > -100 > -150$). Nustatėme, kad poligonas, kurio RSSI verčių iš matavimo X (žr. 3.4 skyrių) atitinka daugiausiai atskaitinių duomenų, tas poligonas yra labiau tikėtinas nei poligonas, kurio verčių iš matavimo X sutapo mažiau. Todėl, atlikta IV lokalizacijos algoritmo modifikacija, kur kiekvieno poligono bendrųjų tikimybių logaritmų vertės L_i yra padalinamos iš skaičiaus n (kur n matomų SSID skaičiaus poligone atskaitinių duomenų rinkimo metu (žr. 3.4 skyrių)), tokiu būdu normalizuojant vieno matavimo įtaką poligonui. Tada gauta vertė dar kartą padalinama iš skaičiaus c , kuris parodo kiek verčių iš matavimo X atitiko to poligono atskaitinių duomenų. Trumpiau tariant paskutiniame penktajame (V) lokalizacijos variante, ieškoma kurio poligono P_i vertė $L_i/(n + c)$ yra didžiausia. Įgyvendinus šį (V) lokalizacijos algoritmą variantą, pasiektas 96 - 97% tikslumas, testuojant su realiais duomenimis. Šis algoritmo variantas yra naudojamas prototipinėje padėties nustatymo sistemoje, kaip padėties nustatymo algoritmas (žr. 6 skyrių).

9.4.3. Tyrimo rezultatų aptarimas ir pastebėjimai

Remiantis šių tyrimų rezultatais, galime teigti, (V) penktasis lokalizavimo algoritmas, paremtas Gauso-Bajeso metodu, gali pakankamai tiksliai nustatyti teisingą poligoną, įvertindamas galimus RSSI matavimų svyravimus to pačio poligono viduje. Taip pat iš lokalizavimo algoritmo palyginimo su klasifikatoriais rezultatais, galime teigti, kad kitokioje padėties nustatymo sistemoje, kur atskaitiniai duomenys, būtų ne Gauso skirstinio parametrai, o neapdoroti matavimai, lokalizavimo algoritmais galėtų būti klasifikatoriai. Remiantis testų rezultatais, tokiai sistemai tinkamiausias būtų Gauso-Naive Bajeso klasifikatorius.

9.5. Atsitiktinio poligono duomenų tyrimas

Atsitiktinio poligono duomenų tyrimo metodika. Praėjus trims savaitėms po to, kai buvo surinkti prototipinės padėties nustatymo sistemos pradiniai apmokymo duomenys, buvo atliktas atsitiktinio poligono duomenų tyrimas. Atsitiktiniame poligone buvo renkami duomenys tuo pačiu įrenginiu su kuriuo buvo apmokyta prototipinė padėties nustatymo sistema. Duomenų rinkimas poligone buvo identiškas duomenų rinkimui apmokant sistemą. Surinkus WiFi rodmenis buvo apskaičiuotos μ ir σ reikšmės kiekvienam SSID. Žemiau pateikiama poligono duomenų lentelė:

14 lentelė. Atsitiktinio poligono duomenų pokytis po trijų savaičių

BSSI + SSID	Mu sena	Mu nauja	Skirtumas	σ^2 sena	σ^2 nauja	Skirtumas, kartais
:8a:61(AirVision)	-86	-92	6	5.5	0.56	9.9
:6c:00(eduroam)	-85	-82	3	20.7	0.3	68.3
:6c:01(MIF-open)	-85	-87	1	20.5	0.7	29.28
:6c:10(eduroam)	-85	-84	1	18.53	4.91	3.77
:6c:11(MIF-open)	-85	-83	2	15.63	4.15	3.76
:45:79:30(eduroam)	-86	-86	0	0.33	14.8	44.40
:79:31(MIF-open)	-87	-86	1	7.33	15.95	2.17
:79:40(eduroam)	-57	-76	19	10	21.16	2.11
:79:41(MIF-open)	-57	-76	19	9.95	18.32	1.84
:79:50(eduroam)	-63	-70	7	22.75	15.95	1.42
:79:51(MIF-open)	-64	-70	6	26.75	19.2	1.39
:79:70(eduroam)	-92	-80	12	3	23.55	7.85

Kaip matome 14 lentelėje surinktos μ reikšmės smarkiai nesiskiria. Tačiau σ reikšmės ženkliai pasikeitusios lyginant su prieš tris savaites surinktais duomenimis. Ankstesniuose tyrimuose buvo parodyta, kad kalibravimas gali išspręsti pasikeitusios aplinkos problemą. Tačiau kalibruojant būtų kalibruojama μ reikšmė, o ši ir taip nesmarkiai skiriasi. Dėl to kalibravimas nebūtų efektingas. Taigi, prototipinė padėties nustatymo sistema sugeba tiksliai nustatyti padėčių patalpose, kai nėra ženkliai pasikeitusi aplinka.

Padėties nustatymo patalpose tikslumą būtų galima pagerinti naudojant papildomus įrenginius išdėliotus tam tikrose patalpos vietose, kurie apmokymo duomenis siųstų į serverį, tačiau šiame darbe, kuriama sistema be papildomų įrenginių. Taip pat remiantis šiais rezultatais, galima būtų rinkti pradinis apmokymo duomenis ilgiau ir bandyti sudaryti daugiau triukšmo, kad būtų pasiekti didesni σ svyravimai, tuomet galimai pradiniai apmokymo duomenys taip smarkiai nepriklausytų nuo aplinkos veiksnių.

Išvados ir rekomendacijos

Išmaniuoju telefonu surinkti barometro duomenys yra tinkami naudoti aukšto pastate nustatymui. Tarp skirtingais įrenginiais surinktų Barometrų matavimų reikalinga atlikti kalibravimą.

WiFi atspaudų vertės yra labai įtakojamos aplinkos veiksnių, todėl norint užtikrinti tikslų padėties nustatymo sistemos veikimą reikalingas pastovus duomenų atnaujinimo sprendimas. Renkant pradinis WiFi atspaudų duomenis rekomenduojama matuojamojo poligono ribose, kuo labiau judinti, keisti, duomenis renkančio išmaniojo telefono, padėtį erdvėje, kad būtų užfiksuojamos kuo didesni matavimų svyravimai.

Tikimybiniai padėties nustatymo algoritmai duoda ženkliai tikslesnius rezultatus nei deterministiniai algoritmai.

Magnetinio lauko matavimai surinkti išmaniaisiais telefonais netinka padėties nustatymo algoritmų naudojimui, nes greitai išsikalibruoja.

Gairės

Pirmiausia toliau vystant šią padėties nustatymo patalpose sistemą reikia pritaikyti pradinių (atskaitinių) duomenų atnaujinimo sprendimą.

Vienas iš galimų variantų naudoti grįžtamąjį ryšį iš vartotojų, duomenis atnaujinant su jų matavimų vertėmis. Kitas galimas variantas būtų, naudoti papildomą infrastruktūrą (vartojančią mažai energijos), kuri galėtų būti išdėliota tam tikrose pastato vietose ir pastoviai atliktų tos vietos WiFi atspaudų skanavimus, o to ir aplink esančių poligonų duomenys būtų atnaujinami pagal tam tikras funkcijas).

Sekantis žingsnis, būtų vienas po kito sekančių padėties nustatymo algoritmų rezultatų filtravimas, kad jie nebūtų vienas nuo kito labai nutolę.

Tolimesni žingsniai priklausytų nuo sistemos pritaikymo, jei tai būtų navigacija patalpose, tai galima būtų naudoti pvz.: greičiausio kelio nustatymo algoritmą.

Literatūros šaltiniai

- [1] Idc statistikos puslapis. <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>. Accessed: 2016-01-10.
- [2] Stephan Adler, Simon Schmitt, Yuan Yang, Yubin Zhao, and Marcel Kyas. Experimental evaluation of indoor localization algorithms, 2014. <http://www.ipin2014.org/wp/pdf/4B-2.pdf>.
- [3] Andrew M. Ladd-Algis Rudys Dan S. Wallach Lydia E. Kavraki Andreas Haeberlen, Eliot Flannery. Practical robust localization over large-scale 802.11 wireless networks, 2004.
- [4] Alberto García-Sola Andrés Sánchez, Francisco Campuzano and Juan A. Botía. Building your own infrastructure based 802.11 fingerprinting service, 2014.
- [5] Nabila Labraoui Asma Mesmoudi, Mohammed Feham. Wireless sensor networks localization algorithms: A comprehensive survey., 2013.
- [6] decaWave. Application note: Aps003 - real time location systems, 2014.
- [7] decaWave. Application note: Aps011 - sources of error in dw1000 based two-way ranging (twr) schemes, 2014.
- [8] decaWave. Application note: Aps013 - the implementation of two-way ranging with the dw1000, 2014.
- [9] Deborah Estrin-Mani B. Srivastava Donnie H. Kim, Younghun Kim. Sensloc: Sensing everyday places and paths using less energy, 2010.
- [10] Badri Nath Dragos Niculescu. Ad hoc positioning system (aps), 2001.
- [11] Member IEEE Xuqiang Zhao Zhibo Pang Senior Member Hyuncheol Park Fang Zhao, Haiyong Luo. Hyfi: Hybrid floor identification based on wireless fingerprinting and barometric pressure, 2015.
- [12] Emanuele Goldoni, Alberto Savioli, Marco Risi, and Paolo Gamba. Experimental analysis of rssi-based indoor localization with ieee 802.15.4, 2010. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5483396>.
- [13] Fredrik Gustafsson and Fredrik Gunnarsson. Positioning using time-difference of arrival measurements.
- [14] Gulnur Selda Kuruoglu, Melike Erol, and Sema Oktug. Localization in wireless sensor networks with range measurement errors, 2009. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5072442>.
- [15] Reinhold Haeb-Umbach Mark Kha Hoang. Parameter estimation and classification of censored gaussian data with application to wifi indoor positioning, 2013.
- [16] Reinhold Haeb-Umbach Mark Kha Hoang, Joerg Schmalenstroeer. Aligning training models with smartphone properties in wifi fingerprinting based indoor localization, 2015.

- [17] Yubin Xu Mu Zhou, Prashant Krishnamurthy and Lin Ma. Physical distance vs. signal distance: An analysis towards better location fingerprinting, 2011.
- [18] Deborah Estrin Nirupama Bulusu, John Heidemann. Gps-less low-cost outdoor localization for very small devices, 2000.
- [19] Rong Peng and Mihail L. Sichitiu. Angle of arrival localization for wireless sensor networks.
- [20] Alejandro Ramírez. Time-of-flight in wireless networks as information source for positioning, 2011.
- [21] Anya Apavatjrut Sujittra Boonsriwai. Indoor wifi localization on mobile devices, 2013.
- [22] Ke Hu Min Yu Ting-Ting Xiao, Xing-Yu Liao. Study of fingerprint location algorithm based on wifi technology for indoor localization, 2014.
- [23] Dong Seog Han Vinjohn V Chirakkal, Myungchul Park. Navigating through dynamic indoor environments using wifi for smartphones, 2014.
- [24] Yao Guo Xiangqun Chen Xin Chen, Junjun Kong. An empirical study of indoor localization algorithms with densely deployed aps, 2014.
- [25] Mohammadreza Yavari and Bradford G. Nickerson. Ultra wideband wireless positioning systems, 2014.