

Hybrid indoor/outdoor location determination technologies

Tim TUTENEL

promotor :
Prof. dr. Wim LAMOTTE



Abstract

Steeds vaker is het niet langer voldoende om statische applicaties af te leveren, er is nood aan context-awareness. Niet zelden is de context van een applicatie nauw gekoppeld aan de plek waar de gebruiker zich bevindt. Daarom is er een vraag gerezen naar localisatietechnieken die deze informatie aan een applicatie kunnen afleveren. Voor outdoor omgevingen zijn er reeds talloze systemen op de markt die in deze behoefte voorzien. Wanneer men echter indoor een locatie wil gaan bepalen van gebruikers zijn er heel wat minder mogelijkheden.

Deze thesis had als doel een bruikbaar systeem te ontwikkelen dat met behulp van Wireless LAN access points een oplossing kan bieden aan het probleem van de indoor localisatie. Uitgaande van een basissysteem zijn er stap voor stap verbeteringen aangebracht om uiteindelijk een afgewerkte library te verkrijgen die eenvoudig toegevoegd kan worden in applicaties die locatieafhankelijk zijn.

Woord vooraf

Vooraf wil ik graag mijn promotor Wim Lamotte en begeleiders Peter Quax, Daniël Teunkens en Kris Luyten bedanken voor hun hulp bij de totstandkoming van deze thesis. Ook het Expertisecentrum Digitale Media wil ik bedanken voor het leveren van de nodige apparatuur die nodig was bij het testwerk.

Inhoudsopgave

Abstract.....	2
Woord vooraf.....	3
1 Inleiding.....	6
2 Positiebepalingstechnieken.....	8
2.1 Outdoor.....	8
2.1.1 Hyperbolische plaatsbepaling.....	8
2.1.2 Satelliet navigatie.....	11
GPS.....	13
Galileo.....	14
Beidou.....	14
GLONASS.....	15
Vergelijking van de systemen.....	15
2.1.3 GSM Cell/Sub Cell Networks.....	15
2.1.4 Radar.....	16
2.2 Indoor.....	16
2.2.1 Infrarood.....	16
2.2.2 Wireless LAN.....	18
2.2.3 RFID.....	20
2.2.4 Bluetooth.....	21
2.2.5 GSM Fingerprinting.....	24
3 Gerelateerde Technieken.....	26
3.1 Richtingbepaling.....	26
3.1.1 Softwarematig.....	26
3.1.2 Elektronisch kompas.....	27
3.1.3 Direction Finding Antenna.....	27
3.2 Hybride Systemen.....	27
3.2.1 Algemeen.....	27
3.2.2 Samenwerkingsproblemen.....	30
3.2.3 Privacy en beveiliging.....	30
4 Bestaande systemen.....	31
4.1 IR tags.....	31
4.1.1 Active Badge next generation.....	31
4.1.2 Versus.....	32
4.2 Ultrasonic Ranging.....	32
4.2.1 Cricket.....	32
4.2.2 The Bat.....	32
4.3 Wireless LAN.....	32
4.3.1 WhereNet.....	33
4.3.2 Airlocation	33
5 Implementatie.....	34
5.1 Algemene aanpak.....	35
5.2 Voorafgaande testen.....	37
5.3 Eerste systeem.....	38
5.4 Tweede systeem.....	42

5.5 Derde systeem.....	47
5.6 Gebruik van het systeem.....	48
6 Algemene verbeteringen.....	51
6.1 Toevoegen van de richting.....	51
6.2 Prediction en History-based technieken.....	52
6.2.1 History-based technieken.....	52
6.2.2 Prediction technieken.....	53
6.3 Interpolatie van posities.....	55
6.4 Informatie over de omgeving.....	57
7 Voorbeeldapplicatie.....	60
8 Conclusie.....	65
9 Index van tabellen.....	68
10 Index van figuren.....	69
11 Bibliografie.....	70
Appendix: Reference.....	72

1 Inleiding

In dit werk was de doelstelling een systeem te ontwikkelen dat met behulp van Wireless LAN access points aan localisatiebepaling kon doen. Belangrijk was dat dit systeem zonder veel moeite kon ingevoegd worden in nieuwe applicaties die wensten gebruik te maken van de localisatie.

In tal van situaties kunnen we ons voorstellen dat het weten van onze locatie ook indoor heel nuttig kan zijn. Wanneer een electricien in een bedrijf onderhoudswerken wil gaan uitvoeren of een nieuw systeem moet gaan aansluiten wil hij uiteraard informatie hebben over hoe de huidige bekabelingen en opstellingen eruit zien. Wanneer hij automatisch op een laptop of tablet pc zou kunnen zien hoe de situatie is op de plek waar hij zich op dat moment bevindt, is dit natuurlijk veel handiger dan wanneer hij zelf moet gaan ingeven waar hij zich bevindt.

Wanneer iemand voor het eerst in een groot bedrijf aankomt en op zoek moet gaan naar het kantoor waar een afspraak gemaakt is, is het vaak vrij moeilijk om snel de juiste weg te vinden. Wanneer die persoon dan met behulp van een pda geleid wordt doorheen het gebouw en liefst dan nog op de snelst mogelijk manier kan dit heel wat tijdswinst opleveren en heel wat frustraties vermijden. Bovendien kan er makkelijk voor gezorgd worden dat informatie over afgesloten vleugels van het gebouw of een defecte lift gebruikt kan worden. Zo kunnen ook mensen die het gebouw al gedeeltelijk kenden in onvoorziene omstandigheden toch makkelijk geholpen worden zonder dat er iemand tijd moet vrijmaken om de persoon te begeleiden.

Wanneer bezoekers aan tentoonstellingen, musea of opendeurdagen tijdens hun bezoek bijkomende informatie willen over tentoongestelde voorwerpen of opstellingen kan men dit uiteraard leveren in de vorm van borden met de nodige informatie. Maar bij drukte is het vaak heel moeilijk om iedereen die dit wil lezen de kans te geven om de hele uitleg te bestuderen. Men zal veel sneller geneigd zijn om verder te gaan zonder alles gelezen te hebben wanneer men merkt dat er een aantal mensen staan te wachten. Daarom is het vaak heel handig om deze informatie mee te geven aan de gebruiker op een audiodrager. Het nadeel bij de huidige systemen is dat de bezoeker dan wel gebonden is aan de aangeboden volgorde. Niet alleen kan de gebruiker niet zelf bepalen in welke volgorde hij zijn bezoek aflegt, ook kan deze niet bepalen waar en wanneer hij extra informatie wil aangezien men beperkt is tot de keuze die men vooraf gemaakt heeft. Aangezien deze keuze vaak is ingegeven door het feit dat vele mensen niet al te veel randinformatie willen, enkel de hoofdzaken, is het onmogelijk om meer geïnteresseerde bezoekers de gevraagde extra informatie te leveren zonder een gids. Ook hier kan localisatie soelaas brengen. Als het audiosysteem weet waar de gebruiker zich op het huidige ogenblik bevindt kan automatisch overgeschakeld worden op het afspelen van de daarbij horende informatie. Wil de gebruiker alles weten over een bepaald voorwerp of kunstwerk zal hij er lang bij blijven staan tot hij alle informatie die

erover te krijgen is beluisterd heeft. Andere gebruikers zullen misschien na het beluisteren van de hoofdzaak al doorgaan naar een volgend voorwerp waardoor het systeem meteen doorschakelt naar de informatie die daarbij hoort. Bovendien is ook meteen een manier geboden om de gebruiker zijn eigen volgorde te laten uitkiezen. Als men naast audioinformatie dan ook nog eens andere informatie zoals foto's of tekst zou kunnen weergeven op bijvoorbeeld een pda dan zou dit de ervaring van de bezoekers uiteraard nog meer verrijken.

Kortom in heel wat omstandigheden kan men zich inbeelden dat het localiseren van een toestel en de gebruiker die dat toestel bedient heel wat voordelen zal opleveren. Om aan deze behoefte ook in binnenlocaties te voldoen zijn echter nieuwe systemen nodig. In dit geval hebben we gekozen voor een systeem met behulp van Wireless LAN access points. Zonder zelfs maar verbinding te maken met deze access points kan men namelijk al door middel van de beacons die ze uitzenden gaan opmeten hoe sterk het signaal van elk access point ontvangen wordt. Deze informatie kan dan gebruikt worden om de localisatie tot stand te brengen.

Het gebruik maken van de WLAN access points heeft trouwens ook een behoorlijk groot voordeel. Als eerste en tweede voorbeeld hebben we een situatie geschetst van een bedrijf. Heel wat grote bedrijven hebben reeds een draadloos netwerk en dus ook de bestaande infrastructuur van access points die normaal gebruikt wordt om draadloze toestellen de mogelijkheid te bieden op het netwerk te komen. Vaak is zo'n infrastructuur al in orde – of kan deze met het bijplaatsen van slechts een aantal access points in orde gemaakt worden – om de localisatie te gaan uitvoeren. Het enige wat dan nog nodig is, is het toevoegen van het systeem aan de applicaties die gebruik zullen maken van de localisatie en het opstellen van een fingerprintdatabase. Om deze op te stellen moeten op een aantal punten metingen gedaan worden van de signalen om deze later dan te kunnen vergelijken met de huidige signalen om dan na te gaan op welk punt het toestel zich bevindt.

Bij het ontwikkelen van het systeem is gekozen voor een stapsgewijze aanpak. Door te beginnen met een basissysteem en dan geleidelijk aan uit te breiden met verbeteringen en andere technieken is er stap voor stap duidelijk geworden welke technieken al dan niet een gunstige invloed hadden op het systeem. Er zijn grofweg twee belangrijke technieken ontwikkeld en toegevoegd aan het systeem. De eerste gaat, op basis van de gemiddelde signaalssterkte die binnenkomt van elk access point, checken hoeveel gelijkenis er bestaat tussen een fingerprint en een huidige meting. Het tweede systeem werkt op basis van histograminformatie voor elke locatie, voor elk access point op elke positie. Op basis daarvan kan een kans afgeschat worden dat een binnengekomen signaalsterkte hoort bij elke locatie in de database. Om het systeem uit te testen en om na te gaan hoe eenvoudig het gebruik ervan is, is er een voorbeeldapplicatie ontwikkeld die een simpele tracking van een Pocket PC voorziet.

Voor dit alles zijn we echter eerst op zoek gegaan naar informatie over reeds bestaande systemen. Eerst hebben we gekeken naar hoe localisatie buiten verloopt. Hierbij is er vooral gekeken naar de satellietnavigatie, hyperbolische plaatsbepaling, GSM fingerprinting en de radar. Daarna hebben we bestaande technieken voor indoorlocalisatie besproken zoals andere WLAN systemen, RFID en infrarood technieken.

2 Positiebepalingstechnieken

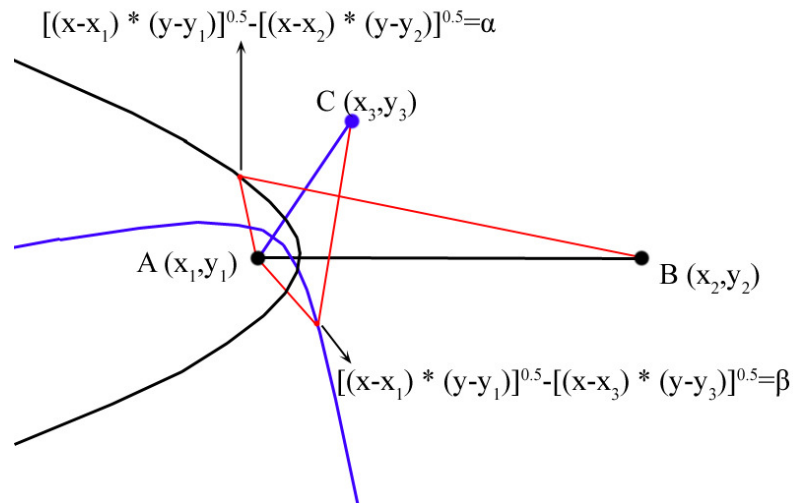
Er zijn in het verleden al tal van situaties gerezen waarin men zijn exacte locatie wilt kennen. Vooral in de scheepvaart en in de militaire wereld is men al decennia geleden begonnen met het zoeken naar geautomatiseerde positiebepalingstechnieken. Deze waren veelal gericht op buitenlocaties. Maar ook in gebouwen zijn al een aantal technieken uitgedokerd om de positie te bepalen. Omdat de outdoor technieken vooral gebruik maken van satelliet signalen of andere signalen die moeilijk doorheen gebouwen kunnen dringen – of waarbij de afwijking bij het binnendringen van de gebouwen onmogelijk af te leiden valt – zijn er voor indoor positiebepaling andere of aangepaste methodes ontwikkeld moeten worden. Vandaar dat de outdoor en indoor technieken apart besproken worden. Outdoor zijn er aantal erg bekende, en veel gebruikte, systemen bekend zoals de radar of de satellietnavigatie met onder andere GPS. De indoor technieken worden vooralsnog minder toegepast maar er zijn toch al heel wat technieken uitgetoetst. Zo zullen onder meer RFID en Wireless LAN technieken bekeken worden.

2.1 Outdoor

2.1.1 Hyperbolische plaatsbepaling

De voorlopers van positiebepaling met satellieten wat tegenwoordig vaak gebruikt wordt, bevonden zich nog op de begane grond. Een vaak voorkomende techniek was de hyperbolische plaatsbepaling. Deze maakt gebruik van de wiskundige kennis van hyperbolen om een plaats van een ontvanger te bepalen. Het is namelijk zo dat voor elk punt van een hyperbool de volgende stelling geldt: $|d_1 - d_2| = 2a$ waarin d_1 de afstand van het punt tot het eerste brandpunt van de hyperbool is en d_2 de afstand tot het tweede brandpunt. Het verschil van de afstanden tot de twee brandpunten is voor elk punt van de hyperbool dus gelijk. Hiervan kan gebruik gemaakt worden wanneer een ontvanger radiosignalen opvangt vanuit twee stations die op hetzelfde moment uitgestuurd zijn. Men kan ofwel het verschil in tijd in beschouwing nemen ofwel het verschil in afstand tot de radiostations (die dan berekend wordt aan de hand van de snelheid van de radiogolven). Men vindt dan een hyperbool waar de ontvanger zich op bevindt. Maar men wil natuurlijk een exacte locatie vinden en daarom is het nodig om nog een tweede paar van zendstations te beschouwen. Het procédé wordt ook voor deze stations herhaald en daardoor wordt een tweede hyperbool gevonden. De ontvanger bevindt zich dan op het snijpunt van de twee gevonden hyperbolen. Uiteraard moet het tweede paar zendstations niet bestaan uit twee nieuwe stations, maar er moeten dus wel signalen van minstens drie zendstations opgevangen kunnen worden om tot een resultaat te komen. Een voorbeeld hiervan wordt getoond in figuur 2.1.

Deze techniek is echter verre van optimaal. De beste resultaten krijgt men wanneer men zich in de nabijheid van



Figuur 2.1: Snijpunt van twee hyperbolen met brandpunten A-B en A-C

de lijn tussen de twee zendstations bevindt – wiskundig dus nabij de lijn tussen de twee brandpunten – omdat hier de afstanden tot de zendstations minimaal zijn. Wanneer men zich steeds verder op de divergerende hyperbolen bevindt zullen de signalen afzwakken en zullen metingen dus onnauwkeuriger worden. Daarenboven worden de signalen ook nog eens gestoord door gebouwen waardoor deze techniek minder geschikt is op het land. Op zee daarentegen kunnen betere resultaten bekomen worden. Deze techniek van positiebepaling is al langer bekend maar door de militaire interesse in het systeem kwam het onderzoek ernaar in een stroomversnelling tijdens de tweede wereldoorlog. De Britten waren de eerste die een dergelijk systeem gebruikten om bombardementen uit te voeren. Het systeem werd nog tijdens de tweede wereldoorlog in gebruik genomen. De Amerikanen zouden nog geen jaar later volgen met een eigen systeem genaamd Loran¹ A. Tegen het einde van de oorlog werd al dertig procent van het aardoppervlak bereikt. Er werden steeds verbeteringen aangebracht vooral qua bereik en dit resulteerde in Loran C. Deze opvolger was 24 uur per dag en in alle weersomstandigheden te gebruiken. Er werden in de loop der jaren steeds meer Loran C stations opgetrokken maar midden jaren negentig begonnen de Verenigde Staten hun stations te sluiten of over te geven aan lokale besturen om over te schakelen op het GPS systeem (zie verderop in dit hoofdstuk). Het Loran C systeem bestaat uit verschillende stations die een paar honderd kilometer uit mekaar staan. Meestal zijn ze ook gegroepeerd tot groepen waarin minstens één station een basisstation is en minstens twee stations nevenstations.

Alle stations binnen een bepaalde groep dienen op exact dezelfde tijdstippen signalen uit te zenden om tot correcte resultaten te komen. Het systeem biedt echter een beperkte nauwkeurigheid. Om een exacte positie te bepalen geeft Loran C slechts een nauwkeurigheid van een paar honderd meter. Maar wanneer met het systeem twee locaties gaan vergeleken worden – bijvoorbeeld wanneer men wil weten of men zich op dezelfde positie bevindt als bij een vorige meting – dan komt men tot een nauwkeurigheid van een tiental meter. Deze twee verschillende vormen van nauwkeurigheid noemt men respectievelijk absolute en herhaalbare nauwkeurigheid. Bovendien zijn er een heel aantal factoren die één of beide nauwkeurigheden negatief beïnvloeden. In tabel 1

¹ Loran: 'Long Range Navigation'

krijgen we een overzicht van deze negatieve factoren.

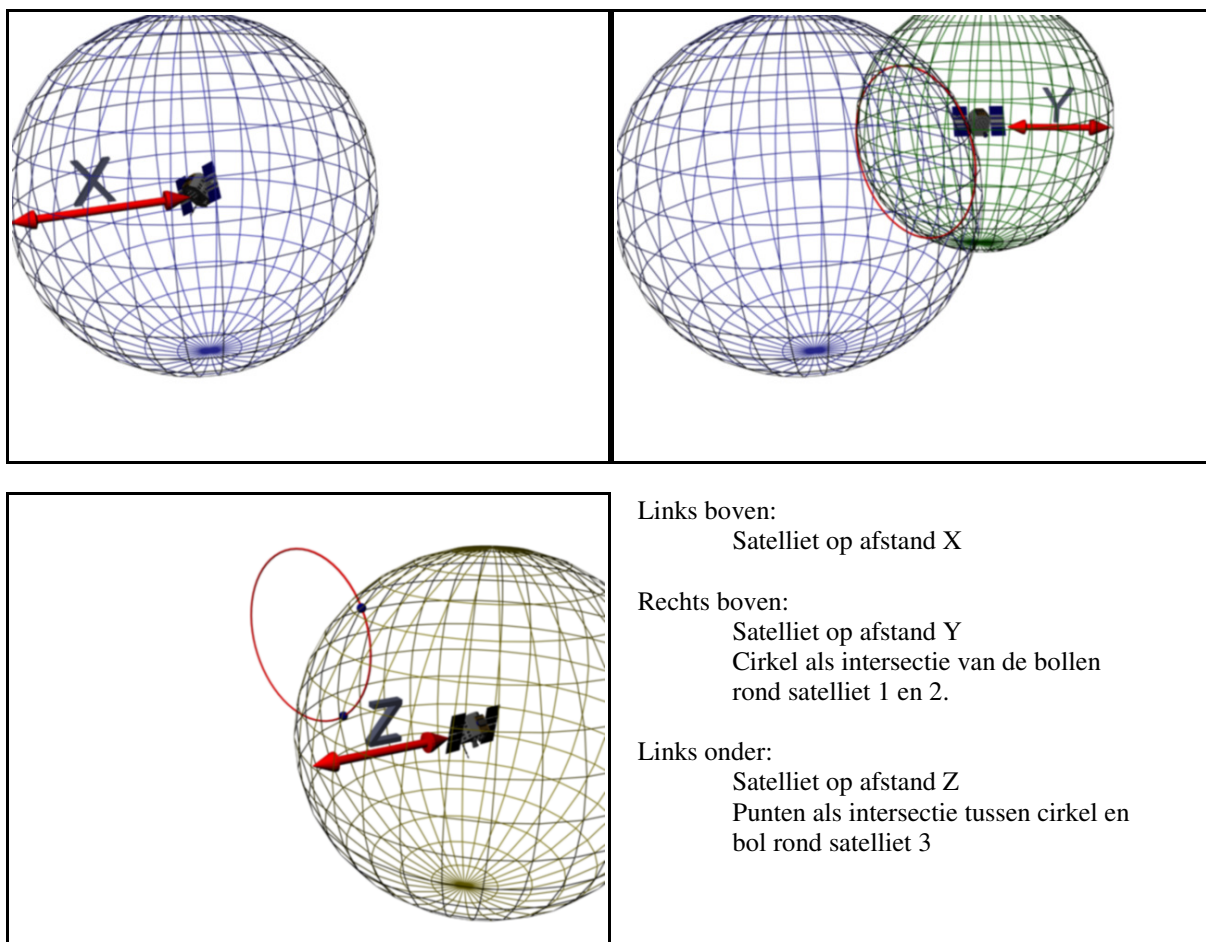
<i>Negatieve factoren</i>	Invloed op Absolute Nauwkeurigheid	Invloed op Herhaalbare Nauwkeurigheid
Hoek en gradiënt van de hyperbolen	Ja	Ja
Stabiliteit van het uitgezonden signaal	Ja	Ja
Parameters van een groep van stations (bijv.: de delay bij het uitsturen van de signalen, ...)	Ja	Ja
Elektronisch ruis	Ja	Ja
Variatie in de propagatiesnelheid van de signalen onder invloed van bijv. het weer.	Ja	Ja
Plotse storingen in de ionosfeer	Ja	Ja
Kwaliteit en gevoeligheid van het gebruikte materiaal	Ja	Ja
Elektronisch ruis op een schip dat Loran C gebruikt	Ja	Ja
Nauwkeurigheid van het mappen op reële coördinaten	Ja	Nee
Nauwkeurigheid van de omzettingsalgoritmes	Ja	Nee
Gebruikersfouten	Ja	Ja

Tabel 1: Factoren die Loran C negatief beïnvloeden (naar tabel 1207 uit pollux.nss.nga.mil/NAV_PUBS/APN/Chapt-12.pdf)

Naast de mogelijkheden voor positiebepaling wordt Loran C ook nog voor andere doeleinden aangewend. Aangezien het systeem op exacte tijdstippen signalen moet uitsturen heeft elke zender een ingebouwde tijdswaarneming met behulp van Cesium Beam Oscillators¹. Hierdoor zijn de zenders ook geschikt voor tijdsaanduiding. Ook wordt Loran C gebruikt om data door te sturen en vaak wordt het op schepen nog steeds gebruikt als backup systeem bovenop een GPS systeem.

Zoals net al aangehaald werd, wordt Loran C vooral gebruikt in de scheepvaart. In de luchtvaart gebruikte men vroeger het Omega systeem. Omega wordt gezien als het eerste wereldwijde radionavigatiesysteem voor de luchtvaart (al werd het later ook gebruikt voor duikboten). Goedgekeurd in 1968 werd het voornamelijk naar voor geschoven als een systeem om nucleaire bommenwerpers te navigeren. Het duurde echter tot 1983 tot alle acht zendstations volledig operationeel waren. Deze waren zeer duur aangezien ze antennes van om en bij de 360 m nodig hadden. Elk zendstation had zijn eigen herkenningsfrequentie waardoor elke ontvanger wist welk signaal van welk station kwam. Elke ontvanger had ook een lijst van propagatie correcties per zendstation waardoor er een nauwkeurigere correctie kon uitgevoerd worden op basis van de kwaliteit van het zendstation. Zo haalde men een nauwkeurigheid van 4 tot 7 kilometer. Het Omega systeem werkt op een gelijkaardige wijze aan Loran C. Elk station zendt met intervallen van 0,2 seconden niet-ambigue signalen uit die op alle acht stations gesynchroniseerd verlopen. Ook in dit systeem waren signalen van drie zenders nodig om tot de vooropgestelde accuraatheid te komen. Ook Omega werd in een latere periode gebruikt voor tijdswaarneming. Het systeem is in gebruik gebleven tot 30 september toen alle zendstations ophielden met zenden.

¹ Cesium Beam Oscillator: Een apparaat dat gebruik maakt van de resonantie frequentie van het cesium atoom om tijd bij te houden. De nauwkeurigheid van een dergelijke oscillator is 10^{-12} tot 10^{-14}



Links boven:

Satelliet op afstand X

Rechts boven:

Satelliet op afstand Y

Cirkel als intersectie van de bollen
rond satelliet 1 en 2.

Links onder:

Satelliet op afstand Z

Punten als intersectie tussen cirkel en
bol rond satelliet 3

Figuur 2.2: Positieberekening mbv. satellieten

2.1.2 Satelliet navigatie

Het allereerste navigatiesysteem dat gebruik maakte van satellieten was het NAVSAT¹ systeem. Dit maakte gebruik van de TRANSIT satellieten waarvan de eerste² gelanceerd is op 13 april 1960. Vier jaar later kon men met een opstelling van een zestal satellieten met rotatie-stabilisatie³ het systeem in gebruik nemen. Vooral in de scheepvaart – zowel onder als boven het wateroppervlak – werd het toegepast. Het systeem maakte gebruik van de afwijking van een radiosignaal door het Doppler effect. Omdat de baan van de satelliet bekend is, kan men uit die afwijking de locatie van de ontvanger afleiden. De eerste tuigen genaamd Oscar hadden een levensduur van twaalf jaar terwijl de latere, meer geavanceerde (maar ook een stuk zwaardere) Nova tuigen het slechts negen jaar uithielden. De laatste TRANSIT werd in augustus 1988 gelanceerd.

De meer moderne systemen gebruiken meer informatie van de satelliet. Door middel van een atoomklok kan de satelliet zijn eigen tijd bijhouden. Deze tijd kan meegestuurd worden met het signaal dat verstuurd wordt waardoor elke ontvanger weet wanneer het signaal startte. Door ook nog eens de positie van de satelliet mee te geven in de boodschap kan de ontvanger een bepaalde omgeving afbakenen waarin het zich bevindt. Uiteraard is

¹ NAVSAT: Navy Navigation Satellite System

² Eerste TRANSIT satelliet: TRANSIT 1B

³ Rotatie-stabilisatie: Objecten in de ruimte die een initiële rotatiesnelheid meekrijgen houden deze constant aan door de minimale wrijving in de ruimte. Hierdoor kan men een satelliet een rotatiesnelheid geven om steeds een optimale richting aan te houden.

dit niet voldoende om een exacte positie te gaan bepalen. Daarom moeten ook hier boodschappen van meerdere satellieten beschouwd worden. Met de signalen van vier satellieten kan een behoorlijk gedetailleerde positie bekomen worden. Dit gebeurt op de volgende manier.

Aan de hand van de signalen van de satelliet kan de ontvanger de afstand tot die satelliet gaan berekenen. Wanneer er dan van meerdere satellieten signalen opgevangen kunnen worden kan een positie bepaald worden (zie figuur 2.2). De afstand tot twee satellieten beschrijft een cirkel. Deze cirkel samen met de afstand tot een derde satelliet brengen twee mogelijke posities voort. Deze techniek noemt men trilateratie. Wanneer de afschatting gemaakt wordt dat de ontvanger zich ergens op zeeniveau bevindt kan meestal één van de twee punten geschrapt worden als kandidaat en zijn de drie satellieten dus voldoende om een positie te bepalen. Wanneer de exacte positie met een hoogte gezocht moet worden (we denken hierbij aan bijvoorbeeld vliegtuigen) dan zijn er ook signalen van een vierde satelliet nodig.

Het bepalen van de afstand tot een satelliet aan de hand van de gestuurde signalen is echter niet evident. Zoals reeds gezegd bevat een satelliet een atoomklok om zeer nauwkeurig de tijd bij te houden. Maar eenzelfde atoomklok in elke ontvanger inbouwen zou veel te duur zijn. Vandaar dat een wat minder nauwkeurige klok in de ontvanger gebruikt wordt die telkens gesynchroniseerd wordt met die in de satelliet. Aangezien geweten is dat de snelheid van radiosignalen 300.000 km/s bedraagt, kan dit gebeuren met de formule $\text{afstand} = \text{tijd} \times \text{snelheid}$. Maar aan een snelheid van 300.000 km/s leggen de signalen één meter af in 3.3333... nanosecondes. Wanneer er dus een nauwkeurigheid van een paar meter vereist is moeten de klokken dus ook nauwkeurig zijn (en gesynchroniseerd zijn) tot op een paar nanosecondes. Bovendien stelt zich nog een probleem doordat de signalen zich niet door het luchtledige voortbewegen. Er moeten dan ook een aantal berekeningen uitgevoerd worden om de invloed van de verschillende luchtlagen die de signalen doorkruisen op te heffen.

Om de klok van een ontvanger te synchroniseren zijn er vier satellieten nodig. De nauwkeurige atoomklokken in de satellieten lopen uiteraard synchroon. Dit wil zeggen dat wanneer een ontvanger zijn positie gaat bepalen aan de hand van de vier satellieten terwijl zijn klok niet nauwkeurig is afgesteld, dat de vier bollen rond de satellieten nooit gaan snijden in één punt. De ontvanger weet dan dat zijn klok niet meer synchroon is. Maar doordat de atoomklokken in de satellieten wel synchroon lopen, zullen de fouten in de vier berekende afstanden wel steeds in een vaste verhouding staan ten opzichte van mekaar. Op basis hiervan kan de ontvanger dus uitrekenen hoeveel de tijd tussen de ontvanger en de satelliet afwijkt.

Zoals eerder reeds vermeld hebben de GPS signalen te kampen met de atmosfeer (dit wordt door herberekeningen in de ontvanger opgelost) en is een onnauwkeurigheid van de ontvangersklok een bron van fouten. Maar er zijn ook nog andere factoren die verkeerde vaststellingen kunnen veroorzaken. Naast een foutlopende klok kan ook een verkeerde positie van de satelliet doorgespeeld worden. Het spreekt voor zich dat er dan een foute locatie van de ontvanger volgt. Naast het aantal satellieten dat gebruikt wordt voor een waarneming is ook de onderlinge afstand ervan belangrijk. Hoe verder de satellieten uit mekaar staan hoe nauwkeuriger de meting. Bovendien kunnen de signalen verstoord worden door hoge gebouwen. De signalen weerskaatsen erop en zijn daardoor langer onderweg dan wanneer ze gewoon in een rechte lijn zouden aankomen.

Omdat de 'reistijd' van de signalen rechtstreeks gebruikt wordt bij het berekenen van de afstand tussen satelliet en ontvanger zal ook dit een verkeerde waarneming veroorzaken.

In de loop der jaren zijn een aantal verschillende van deze technieken uitgevoerd. We bespreken nu kort vier van de belangrijkste in iets meer detail. Het GPS of Global Positioning System is het Amerikaanse systeem, Galileo dat van Europa, Beidou dat van China en GLONASS dat van Rusland.

GPS

GPS is een directe opvolger van het TRANSIT-systeem. Het is een systeem van vierentwintig ICO¹ satellieten die per dag tweemaal een baan rond de aarde maken. De satellieten zijn zo gepositioneerd dat bijna overal op aarde en op elk tijdstip signalen van minstens vier satellieten opgevangen kunnen worden waardoor dus een systeem bekomen is dat overal heel behoorlijke resultaten geeft (zie figuur 2.3). Er zijn naast de vierentwintig satellieten die constant in gebruik zijn ook drie reservesatellieten. Deze worden in normale omstandigheden niet gebruikt en zijn dus enkel nodig bij het uitvallen van een andere. GPS is oorspronkelijk ontwikkeld door het Amerikaanse leger maar is tegenwoordig door iedereen gratis te gebruiken, alhoewel het in stand houden van het netwerk jaarlijks om en bij de 340 miljoen euro kost.



Figuur 2.3: GPS satelliet banen
Copyright © 2005 lordpercy.com

Oorspronkelijk werd er encryptie gebruikt om de correcte data enkel voor het Amerikaanse leger toegankelijk te maken. Bij burgergebruik werd automatisch een bepaalde fout van een paar tientallen meters ingebouwd. Maar toen tijdens de Golfoorlog door een gebrek aan legerontvangers, soldaten burgertoestellen moesten gebruiken werd dit foutstelsel, Selective Availability genaamd, uitgeschakeld. Op dit moment kan het leger nog steeds nauwkeuriger locaties bepalen maar enkel in Noord-Amerika door het WAAS² netwerk. Er zijn ook nog andere manieren waarop GPS-data verbeterd kan worden. LAAS³ is een lokale variant van WAAS. Maar er zijn ook softwarematige technieken die gebruikt worden om de positiebepaling te verbeteren. Vooral bij bewegende

1 ICO: Intermediate circular orbit, de baan van satellieten tussen 1.400 km en 35.790 km hoogte boven de aarde

2 WAAS: Wide Area Augmentation System, een systeem van satellieten en grondstations die de nauwkeurigheid van GPS verbetert

3 LAAS: Local Area Augmentation System, een lokale variant van WAAS die op een vaste locatie ondersteunende informatie uitzendt (bijvoorbeeld op een luchthaven)

voertuigen kunnen er technieken zoals bijvoorbeeld dead reckoning toegepast worden om de bewegingsinformatie van een voertuig te gebruiken om korte periodes van onbereikbaarheid van het systeem (bijvoorbeeld tussen een groot aantal hoge gebouwen) op te vangen.

Galileo

De Europese Unie en het Europese Ruimte Agentschap hebben samen het Galileo project opgezet wat de Europese tegenhanger van GPS moet worden. Het systeem zal complementair zijn met zowel GPS als GLONASS. Het zal bestaan uit dertig satellieten (zevenentwintig actieve plus drie reserve). De nauwkeurigheid zal tot op één meter zijn, ook voor burgerdoeleinden. Men verwacht dat het gebruiksklaar is in 2008. Het systeem zal vijf verschillende services bevatten. De vier basisservices (zie tabel 2) worden aangevuld met een extra 'Search and Rescue' service voor noodsituaties. Ten eerste is er de O.S. Of 'Open Service'. Deze service wordt aangeboden aan de consumenten en general-interest gebruikers. Dan is er de meer dringende service die gebruikt moet worden wanneer er levens op het spel staan en die dan ook toepasselijk de SoL of 'Safety of Life Service' wordt genoemd. Ten derde is er een C.S. of 'Commercial Service' die gebruikt kan worden als toegevoegde waarde, maar enkel voor commerciële en professionele applicaties. De P.R.S. of 'Public Regulated Service' dient enkel voor gebruik door de overheden. Deze service biedt een hoge continuïteit. Tenslotte is er nog een aparte 'Search and Rescue' service om noodboodschappen van overal ter wereld zeer nauwkeurig te traceren.

	<i>O.S.</i>	<i>SoL</i>	<i>C.S.</i>		<i>P.R.S.</i>	
Bereik	Globaal	Globaal	Globaal	Lokaal	Globaal	Lokaal
Nauwkeurigheid -Horizontaal (H) -Verticaal (V)	Duale frequentie: H = 4 m V = 8 m Enkelvoudige frequentie: H = 15 m V = 35 m	4-6 m (duale frequentie)	<1 m (duale freq.)	<10 cm (lokaal verbeterde signalen)	H = 6,5 m V = 12 m	1 m (lokaal verbeterde signalen)
Bereikbaarheid	99,8 %	99,8 %	99,8 %		99-99,9%	
Integriteit	Nee	Ja	Service met toegevoegde waarde		Ja	

Tabel 2: Galileo services (naar tabel pagina 4 [44])

Beidou

Het grote verschil van Beidou¹ met de andere systemen is dat het gebruik maakt van geostationaire satellieten. Deze blijven dus steeds op dezelfde plaats hangen (relatief gezien ten opzichte van de aarde). Hierdoor zijn er aanzienlijk minder satellieten nodig, maar daardoor werkt het systeem wel enkel in bepaalde gebieden.

China ziet in het systeem vooral een nuttig hulpmiddel voor hun groeiende economie door de verbeterde navigatiemogelijkheden voor wagens en trein- en zeeverkeer. Ook zal het volgens de Chinese overheid een belangrijke rol spelen bij weersvoorspellingen, telecommunicatie en bij het controleren van rampen zoals bijvoorbeeld overstromingen of bosbranden.

1 Beidou: de Chinese naam voor het sterrenbeeld 'Ursa Major' of 'Grote Beer'

Hoe het systeem precies functioneert is niet helemaal geweten. Naar alle waarschijnlijkheid werkt het zoals het Amerikaanse WAAS (zie deel over GPS). Het begint met een signaal dat door een controlecentrum uitgestuurd wordt naar twee satellieten. Deze sturen de signalen terug naar de ontvangers. Deze sturen op hun beurt signalen terug naar de satellieten, die het op hun beurt terugsturen naar het controlecentrum. Het controlecentrum kan dan berekenen wat de 2D positie is van de ontvanger en op basis van geografische informatie van China een bijhorende 3D positie vinden om terug te sturen naar de ontvanger. Het is duidelijk dat aangezien er signalen moeten teruggestuurd worden door de ontvangers, dat een veel krachtigere batterij nodig is en uiteraard hardware om signalen te verzenden én ontvangen. Hierdoor zijn de ontvangers van het Beidou systeem groter en zwaarder dan de GPS varianten.

GLONASS

GLONASS is een Russische variant van GPS en is de afkorting voor 'Global'naya Navigatsionnaya Sputnikovaya Sistema' of 'Global Navigation Satellite System'. Het systeem werkt op dezelfde manier als GPS, maar het wordt beheerd door de Russische ruimtevaart eenheid in opdracht van de Russische overheid. Het is operationeel sinds september 1993 en gebruikt net als GPS ook vierentwintig satellieten. Ook de GLONASS satellieten cirkelen twee keer per dag rond de aarde en het is ook een globaal systeem zoals de naam reeds laat vermoeden.

Vergelijking van de systemen

De verschillen tussen GPS, Galileo en GLONASS (vooral tussen GPS en GLONASS) zijn technisch gezien vrij klein. Galileo zal een duidelijk signaal geven wanneer het systeem door één of ander defect slecht functioneert zodat de gebruikers weten wanneer ze niet kunnen vertrouwen op het systeem. Dit zou bij GPS en GLONASS niet zo zijn. Het meest nauwkeurige (in iedere geval voor vrij gebruik) is Galileo met een nauwkeurigheid tot op één meter. De Europese overheid oppert ook dat het betrouwbaarder zal zijn dan de Amerikaanse en Russische variant. Het verschil met Beidou is uiteraard een stuk groter aangezien dit systeem gebruikt maakt van geostationaire satellieten en ook een totaal ander systeem gebruikt om de posities te bepalen.

Het feit dat er zoveel verschillende systemen bestaan die grotendeels hetzelfde doel hebben wordt veelal verklaard door militaire motieven. Alhoewel het nooit met zo veel woorden gezegd wordt willen ze allemaal een grotere militaire onafhankelijkheid.

2.1.3 GSM Cell/Sub Cell Networks

Wanneer een mobiele telefoon een netwerk zoekt, dan gebeurt dit door eerst te zoeken naar de dichtstbijgelegen zendmast. Deze staan in een hiërarchische structuur van cellen en subcellen opgesteld. De technieken die gebruikt worden om uit te zoeken welke cel het dichtst staat, kunnen ook gebruikt worden om afstanden te bepalen tot die zendmast. Ten eerste is er de sterkte van de signalen. Hoe zwakker het signaal, hoe verder de mast verwijderd is. Het is makkelijk te begrijpen dat wanneer men over functies beschikt die de verzwakking van de signalen uitgezet ten opzichte van de tijd beschrijven, men dan de afstand tot de zendmast kan afschatten. Bovendien kan ook de voortplantingstijd van een signaal gemeten worden (vergelijkbaar met de technieken met de satellieten). Tenslotte kan ook de hoek waaronder stralen van verschillende masten toekomen gebruikt worden. Met deze technieken kan een behoorlijke bepaling van de positie van de GSM gemaakt worden.

Echter werken geen van deze technieken goed in een dichtbebouwd gebied. Om dit te verhelpen kan een techniek gebruikt worden die we later ook zullen zien terugkomen bij de indoor-bepaling van locaties. Een gebied wordt onderverdeeld in vakjes. Voor elk vakje wordt bijgehouden hoe sterk de signalen ontvangen vanaf elke zendmast zijn in het bijhorende gebied. Wanneer een GSM dan zijn positie wilt kennen, wordt een vector doorgestuurd bestaande uit de signaalsterktes opgemeten vanaf elke zendmast die bereikbaar is op die locatie. Door de vooraf opgeslagen informatie te doorzoeken kan dan het best matchende vakje gevonden worden. De grootte van de vakjes hangt natuurlijk af van de noden van de applicatie. Wanneer er een query moet uitgevoerd worden naar bijvoorbeeld het dichtsbijgelegen tankstation is er misschien maar een precisie van een paar honderd meter vereist, terwijl wanneer men op zoek is naar de exacte locatie van een gezocht gebouw is een precisie van een paar meter vereist. Een voor de hand liggend probleem bij een dergelijk systeem is het onderhouden van de database. Wanneer een groot gebied met een hoge precisie moet opgenomen worden in de database zal dit een werk van lange adem worden.

2.1.4 Radar

De radar is zeker de meest bekende manier om aan locatiebepaling te doen. Het heeft tal van toepassingen waarvan de bekendste waarschijnlijk het opsporen van schepen of van vliegtuigen is. Maar het wordt ook gebruikt door de politie om de snelheid van voertuigen te checken. De meest gebruikelijke toepassingen van radars zijn het opsporen van bewegende voorwerpen, het bepalen van de snelheid van voorwerpen of om gebieden in kaart te brengen (toepassingen in de ruimtevaart zijn het mappen van gebieden op andere planeten).

Een radar is eigenlijk een radiozender die ook geluidssignalen kan opvangen. De radiozender gaat signalen op een welbepaalde frequentie uitzenden. Wanneer deze signalen tegen objecten botsen zullen ze op een bepaalde manier weerkaatst worden. Deze signalen worden dan terug opgevangen door de radar. Als een object van een radar weg beweegt zullen de geluidsgolven het object minder snel bereiken. Hierdoor worden ook minder snel signalen teruggestuurd waardoor de frequentie van de opgevangen signalen lager zal zijn dan toen het object zich dichterbij de radar bevond. Dit zogenaamde Doppler effect is ook door de mens waarneembaar aangezien geluiden van bewegende objecten anders klinken wanneer de objecten weggaan van de luisteraar of dichterbij komen. Wanneer de radar gedurende een korte periode signalen uitzendt, kan het daarna gedurende een bepaalde periode naar inkomende signalen gaan luisteren. Aangezien het uiteraard langer duurt eer er weerkaatste signalen ontvangen worden wanneer het object zich verder van de radar bevindt kan op deze manier de afstand van een object bepaald worden. Dit principe met geluidsgolven wordt toegepast in een sonar, vaak gebruikt in duikboten. Maar door de gelimiteerde afstand die geluidsgolven kunnen afleggen worden meestal radiogolven gebruikt en dan spreekt men van een radar.

2.2 Indoor:

2.2.1 Infrarood:

Steeds vaker zijn in draagbare elektronische toestellen zoals pda's of laptops infraroodzenders en -ontvangers verkrijgbaar. Toestellen kunnen onderling informatie doorsturen via IrDA¹. De fysieke laag van IrDA heeft twee belangrijke sublagen. De Encoder/Decoder die signalen omzet in data en omgekeerd en de Ir Transducer Module

¹ IrDA: Infrared Data Association

die de signalen opvangt en uitzendt. Het bereik van IrDA devices varieert van één tot twee meter. Er zijn ook low-power varianten die slechts 20-30 centimeter ver gaan. Naast het bereik speelt ook de ontvangst- en uitzendhoek een belangrijke rol. Deze bepalen namelijk of een ander apparaat in het zicht van een ander apparaat zit. De transmitter heeft een uitzendhoek van 30° tot 60° terwijl de ontvanger doorgaans een ontvangsthoek kleiner dan 30° heeft.

De belangrijkste factor die een rol speelt in de storingen bij IrDA is uiteraard licht, maar ook magnetische velden. Deze storingen kan IrDA nog net aan:

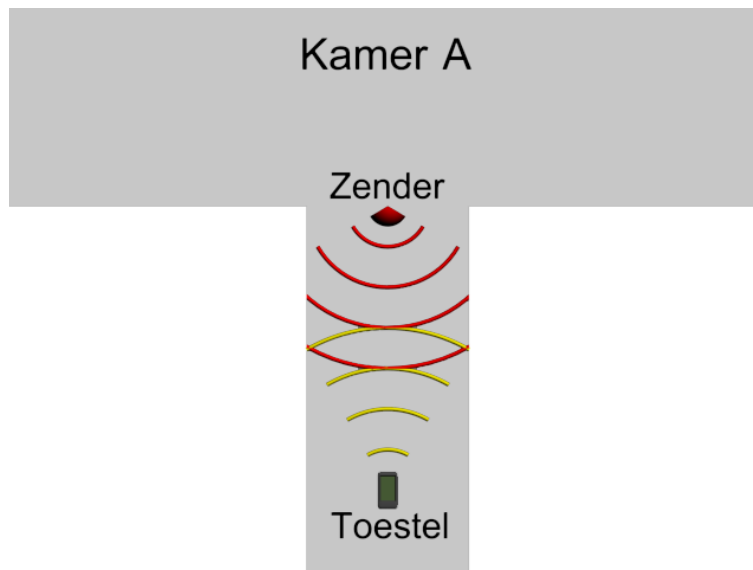
- Een elektromagnetisch veld van maximaal 3 V/m
- Zonlicht van maximaal 10 kilolux
- Gewoon en fluorescerend licht van maximaal 1000 lux

Er is natuurlijk geen garantie dat een device meerdere van deze storingselementen tegelijk aankan.

Het basisprotocol voor IrDA is IrLAP¹. In dit protocol is er een meester die beslist wie op welk moment mag sturen (zodat geen twee apparaten tegelijk info doorsturen naar hetzelfde apparaat) en slaven die uiteraard pas mogen sturen als ze van de meester toestemming hebben gekregen. Om apparaten te ontdekken gebruikt een apparaat een Request-boodschap. Als antwoord hierop zal een ander apparaat een bevestigingsboodschap doorsturen met de lijst van beschikbare apparaten. Ook zal een apparaat dat een aanvraag ontvangt via een indicatieboodschap de andere apparaten in zijn lijst op de hoogte brengen van het nieuwe apparaat.

De opsporingsprocedure om een connectie op te zetten gebeurt als volgt. Een apparaat wordt even wakker en luistert of er dataverkeer ontvangen wordt. Bij verkeer gaat het apparaat naar de slaapstand en wacht om het later opnieuw te proberen. Als er geen verkeer wordt vastgesteld zal het een XID (Exchange Identification) frame sturen met een unieke id om aan te geven dat het apparaat een connectie wil leggen. Wanneer er reactie komt op deze XID dan weet het apparaat dat het connectie kan maken. Zoniet gaat het apparaat terug in zijn slaapstand en wacht twee tot drie seconden om de procedure te herbeginnen. De id's zijn 32-bit adressen die at random worden gegenereerd. Het is dus in een uitzonderlijk geval mogelijk dat twee apparaten toevallig hetzelfde adres gebruiken. Wanneer een apparaat dit opmerkt zullen meteen de apparaten in kwestie verwittigd worden hiervan zodat ze elk een nieuw adres kunnen aannemen.

1 IrLAP: Ir Link Access Protocol



Figuur 2.4: IrDA toestel dat kamer A nadert

Een bijkomend protocol is IrLMP¹. Dit heeft meerdere functies. Ten eerste is er LM-MUX². Dit kan de signalen van verschillende apparaten over eenzelfde IrLAP connectie multiplexen zodat het communiceren niet langer gecoördineerd moet worden. In de tweede plaats is er ook het LM-IAS³ protocol. Hiermee kan een client informatie opvragen die door een server in een database in een bepaalde applicatie opgezocht en teruggestuurd kan worden. IrDA heeft geen encryptie of beveiliging. Dit is echter niet nodig. Door het beperkt bereik en invalshoek van IrDA apparaten is het quasi onmogelijk om ongemerkt mee te luisteren op een bepaalde communicatie.

Dit principe kan vrij makkelijk gebruikt worden om aan een primitieve vorm van locatiebepaling te gaan doen. Wanneer een toestel namelijk infrarode signalen opvangt van een zender dan bevindt deze zich dus met zekerheid in de buurt van die zender. Bovendien kan soms ook een richting afgeleid worden uit deze informatie omdat de ontvanger slechts onder een bepaalde hoek signalen kan opvangen. Dit is het best te begrijpen aan de hand van figuur 2.4. In de figuur nadert een IrDA toestel een kamer via een gang. Aan de ingang van de kamer hangt boven de deur een zender. Wanneer het toestel het signaal van die zender opvangt weet men dus niet alleen dat de gebruiker van dat toestel in de buurt van die zender staat maar dat deze ook in de richting staat van kamer A. Men kan dus afleiden dat de gebruiker van het toestel naar alle waarschijnlijkheid op het punt staat kamer A te betreden.

2.2.2 Wireless LAN

We bespreken om te beginnen een cel-gebaseerde aanpak met Wireless LAN (WLAN). Er zijn om te beginnen twee verschillende mogelijkheden om cel-gebaseerde positiebepaling te verkrijgen: met of zonder client-based software. Het hoofdprincipe is echter telkens gelijkaardig. Op basis van de signaalsterkte van ontvangen radiosignalen, al dan niet aangepast door vergelijking met vorige signalen, een afstand bepalen tot de zenders van

¹ IrLMP: Ir Link Managment Protocol

² LM-MUX: Link Managment Multiplexer

³ LM-IAS: Link Managment Information Access Service

de ontvangen signalen en dan met die informatie een positie berekenen. Ook hier merken we weer sterke gelijkenissen met het systeem met satellieten, zij het op kleinere schaal.

Cel-gebaseerde positiebepaling van terminals bijvoorbeeld PDA's gebeurt door contact te leggen met een draadloze cel. Deze draadloze 'access points' kunnen uitgerust zijn met SNMP en in sommige gevallen met RADIUS wat de positiebepaling makkelijker en sneller kan doen verlopen. De nauwkeurigheid van deze cel-gebaseerde systemen staat of valt uiteraard met de nauwkeurigheid van de access point cellen. In het RADIUS systeem zal elke communicatie tussen een terminal en een access point doorgegeven worden aan de beschikbare RADIUS server. Deze server houdt eerdere communicaties bij in een logfile en zal op basis van de logs en nieuwe communicaties de positie kunnen gaan bepalen. Er moet in dit systeem wel gecheckt worden op de liveness van de terminals. Wanneer namelijk een terminal uitgeschakeld wordt en terug ingeschakeld wordt op een andere locatie zullen eerder gelogde communicaties waardeloos zijn. Daarom zal de server de terminals gaan pingen als die over de IP-adressen beschikt om zo te bepalen welke terminals wel of niet geactiveerd zijn.

Wanneer het SNMP protocol gebruikt wordt gaat men uit van data die op de access points zelf wordt bijgehouden. In de zogenaamde Bridge Learn Table zitten de MAC-adressen en de gebruikte interface van elke client. Gebruikmakend van deze informatie kan opgevraagd worden met welke access points de terminal in contact is geweest. Omdat de data in deze tabellen voorbijgestreefd is wanneer een terminal buiten het wireless bereik van het access point verdwijnt wordt er ook informatie bijgehouden van communicaties over een andere interface zoals ethernet. Eens de databases deze informatie bevatten kan een aanvraag gestuurd worden naar alle access points. Op basis van de resultaten van deze ondervraging kan de positie bepaald worden. De delay in dergelijk systeem kan jammer genoeg wel erg groot worden aangezien telkens élk access point ondervraagd moet worden.

Een laatste besproken techniek is wel geholpen door client-software. De hoofdbedoeling is dat de terminal de signaalsterktes van de zich omringende access points beschouwt. Hiermee kan dan bepaald worden hoe ver die terminal zich van de access points bevindt om daarmee dan de positie te gaan bepalen. De besproken techniek maakt gebruik van een vingerafdruk-server. In een eerste fase moet de database op deze server gevuld worden met de nodige data. Er wordt een tabel opgesteld met de positiecoördinaten en de bijhorende gemeten signaalsterktes. In een tweede fase zal dan een willekeurige terminal zijn signaalsterktes doorsturen naar deze server waardoor een positie kan bepaald worden op basis van de waarden uit de database. De waarden van de signaalsterktes worden vergeleken met diezelfde waarden uit de database. De positiecoördinaten behorende bij de dichtstbijgelegen signaalsterktes worden dan als positie meegegeven. Het spreekt voor zich dat hoe meer gegevens er in de database worden gezet hoe nauwkeuriger de positie zal zijn.

Er zijn nog andere manieren om de uitkomst te verbeteren. Aangezien er soms erg onverwachtste en grote afwijkingen zitten op de signaalsterktes kan de client in plaats van de gemeten signaalsterktes het gemiddelde doorsturen van een reeks opgemeten sterktes. Op die manier worden de uitschieters weggewerkt. Een andere mogelijkheid is om op de server in plaats van het minst afwijkende item, een x aantal minst afwijkende items te beschouwen en dan een interpolatie maken tussen de positiewaarden die bij die items horen. Om van

verschillende access points te ontvangen moet de client op verschillende kanalen gaan luisteren. Meestal is een 100 ms per kanaal voldoende. Als de techniek van meerdere signaalwaardes per access point wordt gebruikt, moet er uiteraard langer gewacht worden. Als dan een zogenaamd beacon-pakket van een access point is ontvangen moet ook de netwerknaam gecheckt worden. Anders is het onmogelijk dit pakket te onderscheiden van een pakket op hetzelfde kanaal maar van een ander netwerk. Om het nauwkeurigst te werken mogen ook enkel de pakketen van het exacte kanaal beschouwd worden. De access points sturen de signalen naar een bepaalde range van vijf kanalen rond een bepaald kanaal in plaats van naar één specifiek kanaal. De goede werking van de techniek is natuurlijk afhankelijk van het aantal access points waarvan de client signalen ontvangt. Het is onvoldoende dat de terminal signalen van één access point ontvangt. Daarom is het best dat deze access points zo opgesteld worden dat ze mekaar overlappen. Het nadeel van deze techniek in vergelijking met de twee vorige is uiteraard de client-software die geïnstalleerd dient te zijn. Aangezien nogal wat toestellen op verschillende besturingssystemen werken, is dit vaak een nadeel.

Uiteraard is een systeem dat gebruik maakt van de beacon signalen wenselijk. Aangezien sommige gebouwen sowieso al beschikken over een draadloos netwerk moet er geen nieuwe hardware gekocht worden, tenzij sommige access points zo opgesteld staan dat er bijna geen overlapping is waardoor het systeem dus minder nauwkeurig zou worden.

De invloed van objecten op de signalen is een gebied dat zeker nog verder onderzocht moet worden. Het is in vele gevallen wenselijk dat men weet hoe de signalen gaan fluctueren wanneer er zich veel andere mensen in de ruimte bevinden. Maar over het algemeen is de techniek al nauwkeurig genoeg om een behoorlijk goede indoor locatiebepaling te implementeren. Bij de meeste systemen komen uit testen resultaten van gemiddeld drie meter nauwkeurigheid voort.

2.2.3 RFID:

RFID is een draadloos identificatiesysteem vergelijkbaar met de streepjescodes. Er zijn drie componenten aan een RFID systeem: een transponder (RF tag), een lezer en programmers. Een transponder kan data sturen naar een lezer wanneer er een voorafbepaald signaal ontvangen wordt. Die data kan identificatiedata zijn, maar in tegenstelling tot streepjescodes kan een RFID transponder ook andere data doorsturen. De meer complexe transponders zijn zelfs in staat encryptie toe te voegen aan de verstuurde data. Er zijn transponders met of zonder eigen energiebron. Diegene zonder eigen energiebron (passieve transponders) zijn meestal minder krachtig met een iets kleiner bereik en ze halen hun energie van het veld gegenereerd door de lezer. De actieve transponders daarentegen zijn krachtiger maar door het laag energieverbruik kunnen ze toch nog tot tien jaar meegaan. De data wordt opgeslagen in ROM, WROM of RAM. De ROM-transponders zijn goedkoper aangezien ze geen energiebron vereisen en er geen schrijfcircuits nodig zijn. De grootte van de transponders kan sterk variëren. Er zijn exemplaren van 1mm x 10mm, maar ook van 120mm x 100mm x 50m. Ze kunnen ook in bepaalde vormen gefabriceerd worden zoals in schroefvorm om in hout ingebracht te worden

De lezers moeten de transponders activeren door een bepaald activeringssignaal te sturen. Daarna moeten ze de

door de transponder gestuurde data kunnen opslaan en eventueel ook verwerken. Het is ook mogelijk dat de data wordt doorgespeeld aan een computer of een ander apparaat dat de data nodig heeft. Ze kunnen ook aan signal conditioning doen en pariteitsfouten opsporen en corrigeren. Aangezien er zich meerdere transponders in de omgeving van een lezer kunnen bevinden moet een lezer de mogelijkheid hebben om te communiceren met afzonderlijke transponders. Eén techniek hiervoor is hands down polling. Als de lezer merkt dat het bepaalde data opnieuw ontvangt, kan door het 'Command Response Protocol' de transponder opgedragen worden het sturen te staken. Een veiligere maar tragere methode is hands up polling. Hierbij zal de lezer elke transponder apart vragen om hun data te sturen. Dit zijn echter niet de enige gebruikte methodes.

Elke transponder heeft een reeks van instructies. Enkel de WROM en RAM versies kunnen na de constructie nog geherprogrammeerd worden. Hiervoor zijn dan wel RFID transponder programmers nodig. Deze werken ongeveer zoals een lezer. Ze zullen de transponder de opdracht geven om bepaalde data te ontvangen en in hun geheugen op te slaan.

Veel gebruikte frequenties voor RFID zijn 13.56 MHz of <135 kHz. Een belangrijk voordeel van het RF veld van 13.56 MHz is dat de signalen zo goed als niet geabsorbeerd worden door mensen of water. Het kan dus perfect gebruikt worden in een omgeving met heel wat mensen zonder de transmissies te verstoren. Het nadeel is echter dat het wel verstoord kan worden door metalen voorwerpen. Er worden ook frequenties gebruikt van 400-1000 MHz UHF en 2450 MHz microgolven. De systemen met deze hogere frequenties hebben veelal een groter bereik. De signalen verliezen bijna door alle stoffen kracht door absorptie.

De manier waarop deze techniek gebruikt kan worden bij positiebepaling is gelijkaardig aan deze van IrDA (zie deel 2.2.1). Men kan in bepaalde kamers of bij bepaalde voorwerpen in bijvoorbeeld een museum tags plaatsen. Als de gebruikers, in het geval van het museum dus de bezoekers, een lezer hebben kan die dus bepalen aan de hand van welke tag hij nabij weet, bepalen waar de gebruikers zich bevinden. Zo kan de lezer dus bepalen nabij welk voorwerp de museumbezoeker zich bevindt.

2.2.4 Bluetooth:

Bluetooth is een kort bereik radio technologie waarmee zogenaamde PAN's (Personal Area Networks) kunnen opgezet worden. Er zijn heel wat protocols die gehanteerd worden via bluetooth, waarvan slechts een gedeelte door alle bluetoothapparaten ondersteund worden. Anderen zijn optioneel. Bluetooth gebruikt een veel gebruikte frequentieband (ISM). Om zoveel mogelijk storingen te vermijden en de invloed van storingen ook tot een minimum te beperken past men bij bluetooth het principe van het frequency hopping toe. 1600 keer per seconde wordt er volgens een pseudo-willekeurige volgorde van frequentie veranderd. Storende signalen zullen een communicatie dus nooit voor lange tijd bemoeilijken. Het bereik van Bluetooth hangt af van de energie die erin gepompt wordt:

- 100 mW >> 100 m
- 2.5 mW >> 20 m
- 1 mW >> 10 m

Baseband: de netwerken met Bluetooth werken volgens een master/slave principe. Elke master kan een zogenaamd piconet opzetten met zeven slaves. Twee of meer piconets kunnen met mekaar verbonden zijn tot een scatternet wanneer één van de slaves uit het éne piconet de master is van een ander piconet. Master en slave rollen kunnen wisselen tijdens een sessie. Slaves kunnen enkel data verzenden als de master dit gevraagd heeft. Masters en slaves kunnen afwisselend zenden. Ofwel kan ieder één, maar soms ook drie of vijf tijdslots lang zenden. Daarna neemt de ander het weer over. Wanneer devices op mekaar zijn aangesloten vormen ze een ACL link die gebruikt kan worden om data te zenden. Enkel wanneer een slave uitdrukkelijk de toestemming heeft gekregen om een tijdslot te gebruiken mag deze zenden. Er kunnen ook rechtstreekse SCO links tussen een master en een slave opgezet worden (tot drie per master) die meestal gebruikt worden voor streaming audio. Wanneer een master SCO-links heeft opgezet worden een aantal van de master-slots speciaal gebruikt voor deze links. Ook bepaalde van de slave-slots worden gereserveerd voor SCO-links. Dit betekent dat in deze verbindingen de slave geen toestemming nodig heeft om deze tijdslots te gebruiken. Een Bluetooth pakket bestaat uit de volgende drie elementen:

- Access Code
- Header
- Payload

De access code maakt duidelijk welk soort pakket het is en waarvoor het bedoeld is. We onderscheiden hierin vier soorten:

- CAC: Channel Access Code wordt gebruikt door alle delen van een piconet en wordt bepaald door de master in dat piconet
- DAC: Device Access Code wordt gebruikt wanneer een specifiek apparaat moet aangesproken worden en wordt bepaald door het apparaat zelf
- GIAC: General Inquiry Access Code is een pakket om te informeren welke apparaten er zich in de buurt bevinden
- DIAC: Dedicated Inquiry Access Code heeft hetzelfde doel als de GIAC maar dan enkel om apparaten te zoeken die tot een bepaalde groep behoren

De header bevat naast het adres van de slave in het piconet ook nog een pakkettype en een slotlengte. Er kunnen ook bevestigingen inzitten van het ontvangen van vorige boodschappen en het bevat ook een CRC voor de header.

Een ACL pakket kan 2712 bits aan data verzenden. Er zit een extra header in dit pakket om aan te geven of het een L2CAP-bericht of een LMP-bericht is. Ook de lengte van de data moet meegestuurd worden in deze header. LMP pakketten kunnen slechts één pakket lang zijn, maar L2CAP-berichten kunnen veel langer zijn, dus moet er ook een flag meegestuurd worden om aan te geven het hoeveelste deel het pakket is van de total boodschap. Een SCO pakket heeft een vaste lengte van dertig bytes. Het bevat altijd tien, twintig of dertig bytes aan echte informatie en respectievelijk twintig, tien of nul bytes voor de FEC (Forward Error Correction). De twee bovengenoemde pakketten kunnen ook samen vermengd worden en zo doorgestuurd worden. Het SCO gedeelte

is dan 10 bytes lang zonder FEC. Het ACL gedeelte kan tot 72 bits innemen en heeft een 2/3 FEC verhouding.

NULL: Een lege payload. Zo'n pakket dient dus enkel ter bevestiging van een ander bericht wanneer er geen data is om terug te zenden. Deze pakketten hoeven niet bevestigd te worden.

POLL: Deze pakketten hebben dezelfde samenstelling als NULL-pakketten maar moeten wel bevestigd worden.

FHS: Hiermee wordt het synchroniseren van de klok en het sequentie hoppen met het zendende apparaat geregeld.

De verschillende states van de link controller zijn:

- Standby: het toestel staat uit en er is geen communicatie mogelijk
- Inquiry: dichtbijzijnde apparaten worden gezocht
- Inquiry scan: er wordt geluisterd naar toestellen in de Inquiry-state om hierop te reageren
- Page: een ander toestel proberen te bereiken door een rechtstreekse verbinding proberen op te zetten
- Page scan: zijn klaar om aangesproken te worden door andere toestellen die het willen bereiken met een Page-state
- Connection - Active
- Connection - Hold: een toestel kan even in de hold-state gaan om bijvoorbeeld naar andere toestellen te zoeken. Doordat het piconet adres behouden blijft kan na deze actie terug deel uitgemaakt worden van het piconet
- Connection - Sniff: een toestel zal gedurende een door de master en slave samen bepaalde duur luisteren
- Connection - Park: een lage energie slaapmodus: wordt enkel wakker om speciale synchronisatie beacons te ontvangen van de master. Na de sleep mode kan er gehersynchroniseerd worden met de master

De Link manager zal de beste configuratie voor een connectie bepalen. Voor een SCO-verbindingen zal de master deze instellingen bepalen omdat de slave over onvoldoende informatie beschikt. Onder andere de FEC-verhouding wordt in deze configuratie vastgelegd. Bij ACL-verbindingen wordt de lengte van de pakketten en het aantal beschikbare slots hiermee vastgelegd. Als een slave orders krijgt om naar andere instelling over te gaan moet deze instemmen. Op een sterk verstoorde 'noisy' connectie is het aantal fouten hoog en moeten langere pakketten dus vermeden worden. Dit kan dan ook afgesproken worden in de connectie-configuratie. Om energie te besparen moet de signaalsterkte tot een minimum herleid worden. Wanneer één van de toestellen een te sterk signaal opmeet wordt er gevraagd deze sterkte te verminderen (kan niet onder een bepaald minimum gaan). Wanneer de signalen te zwak zijn wordt het omgekeerde gevraagd (hier is er een maximum). Eerst moet uiteraard wel gecheckt worden of een toestel over bepaalde features beschikt vooraleer er afspraken daaromtrent gemaakt worden.

L2CAP¹ heeft de volgende kenmerken. Door hogere protocols te multiplexen kunnen ze samen over een lagere layer connectie verstuurd worden. Het kan langere pakketten aan, vaak langer als deze in de baseband. Alle applicaties moeten L2CAP ofwel rechtstreeks ofwel onrechtstreeks gebruiken als protocol. Het RFCOMM protocol emuleert seriële kabel instellingen van een seriële poort. Bijna alle applicaties gebruiken dit om data te

1 L2CAP: Logical Link Control and Adaptation Protocol

verzenden. RFCOMM maakt dan ook gebruik van L2CAP. Er wordt gebruik gemaakt van kanalen om verschillende services te voorzien (maximaal 30).

Aangezien toestellen met Bluetooth vaak gebruikt worden terwijl mensen zich voortbewegen zullen er dus telkens weer andere toestellen in de buurt van een ander toestel komen. Daarom moet er vaak gezocht worden naar dichtbijzijnde toestellen waarvan telkens de beschikbare services van moeten opgevraagd worden. Hiervoor zorgt SDP¹. De SDP database bevat voor elk apparaat welke services het aankan. Elke service heeft een aantal vaste attributen zoals de identiteit van de service, een icoontje voor in de gui, een naam, ... Ook moet elke attribuut heeft drie delen: datatype, lengte van de data en de data zelf. Om makkelijk te kunnen zoeken in de services wordt dit in een hiërarcische structuur bijgehouden. Elke service heeft ook een Universal Unique Identifier of UUID van 128 bits. De standaardservices hebben een 16 of 32 bits UUID die wel omgezet kan worden naar een 128 bits model om te vergelijken. Een fabrikant kan zelf een UUID vastleggen voor een nieuwe service. Het proces om zo een UUID aan te maken is op zo'n manier opgezet dat elke UUID uniek is.

Het is duidelijk dat qua technieken vooral Wireless LAN overeenkomt met Bluetooth. Ook hier kunnen de signaalsterktes eventueel gebruikt worden. Maar er zijn nog aantal manieren die specifiek zijn aan Bluetooth. Door de communicatieprotocollen van Bluetooth kunnen verschillende toestellen met elkaar communiceren. Dit wil zeggen dat zelfs een toestel zonder enige mogelijkheid om zijn positie te bepalen maar met bluetooth connectiemogelijkheden toch zijn positie te weten kan komen. Dit gebeurt dan door een connectie op te zetten met een ander toestel dat hopelijk wél weet op welke positie het zich bevindt. Deze informatie kan dan overgegeven worden. Dit geeft dan een nauwkeurigheid van honderd, twintig of tien meter afhankelijk van het bereik. Uiteraard moet bijgehouden worden hoe lang het geleden is dat het tweede toestel zijn positie berekend heeft. Ook moet nagegaan worden of dat tweede toestel zelf zijn positie heeft bepaald of het ook maar van een ander toestel te weten gekomen is. In het tweede geval wordt het systeem nog onnauwkeuriger. Het spreekt voor zich dat een dergelijk systeem niet aan te raden is. Het dient dan ook enkel als een backup te gebruiken in situaties waarin het echt niet anders kan of eventueel als extra check voor de zelf berekende informatie.

2.2.5 GSM Fingerprinting

De netwerken voor GSM's zijn bijzonder efficiënt. Bovendien is in de meeste gebieden ook de nodige infrastructuur aanwezig om quasi de volledige oppervlakte van dat gebied te bedekken. De signalen kunnen ook door gebouwen dringen waardoor het bijzonder handig zou zijn wanneer het bestaande netwerk met beperkte aanpassingen ook zou gebruikt kunnen worden om posities te bepalen. In deel 2.1.3 hadden we het al over het GSM-netwerk dat bestaat uit cellen en subcellen.

Het GSM-netwerk heeft als voordeel dat alhoewel op verschillende locaties totaal verschillende signaalsterktes gemeten kunnen worden dat ze weinig fluctueren in functie van de tijd, dit in sterke tegenstelling met bijvoorbeeld de signalen van Wireless LAN. Wanneer de signaalsterktes opgevangen zijn kan het reeds bekende verhaal van fingerprinting toegepast worden.

1 SDP: Service Discovery Protocol

Ook hier moet in een eerste fase op zo veel mogelijk verschillende locaties de signaalsterktes gaan opgemeten worden. Aangezien deze vrij onvoorspelbaar zijn voor elke afzonderlijke locatie moet dit gebeuren voor elk punt van een gebied dat als positie moet teruggeven kunnen worden. Dit moet dan bijgehouden worden in een database. Eens dit gebeurd is kan een gebruiker zijn positie opvragen door een lijst van de huidige signaalsterktes los te laten op de database waardoor een hopelijk exacte positie berekend kan worden.

We hadden het al eerder over hoe dit kan gebeuren. Uiteraard is er het doorzoeken van de database naar de meest passende signaalsterktevectoren. Maar dit geeft uiteraard slechts een mogelijke nauwkeurigheid die afhankelijk is van de eerste fase. Bijvoorbeeld wanneer in die eerste fase een lokaal is opgemeten door horizontaal en verticaal steeds tien meter verder te gaan kan er slechts een nauwkeurigheid van tien meter bekomen worden. Men kan echter met eenzelfde nauwkeurigheid in de eerste fase toch een grotere nauwkeurigheid van positie bekomen. Dit kan door meerdere vectoren te zoeken die gelijk zijn op de meegegeven vector. Vervolgens kunnen hieraan gewichten toegevoegd worden afhankelijk van hun gelijkenis. Vervolgens kan men die gewichten ook koppelen aan de posities die bij de vectoren horen waardoor een uiteindelijke positie geïnterpoleerd kan worden. Op die manier kan een grotere nauwkeurigheid bekomen worden. Stel dus dat drie vectoren A, B en C gevonden worden die allen gelijk zijn op meegegeven vector V en die overeenkomen met respectievelijk posities (x_1, y_1) , (x_2, y_2) en (x_3, y_3) . De bijhorende gewichten zijn respectievelijk a, b en c. Dan zou de te zoeken positie dus berekend worden als: $(a * x_1 + b * x_2 + c * x_3, a * y_1 + b * y_2 + c * y_3)$.

Met dit systeem zijn reeds een aantal experimenten uitgevoerd. In [43] heeft men naar eigen zeggen bij experimenten een nauwkeurigheid tussen 2.48 m tot 5.44 m bereikt.

3 Gerelateerde Technieken

In het volgende hoofdstuk worden een aantal technieken besproken die gerelateerd zijn aan de positiebepalingssystemen. Naast de positie is het – zeker in indoor omstandigheden – erg handig dat ook de richting van een persoon bekend is. Hiervoor zijn er een aantal mogelijkheden. Ten eerste zijn er softwarematige oplossingen, maar ook hardwarematige oplossingen zoals het elektronisch kompas en de 'Direction Finding Antenna' die we beide zullen bespreken. Bovendien is er ook onderzoek gedaan naar het combineren van verschillende in het voorgaande hoofdstuk besproken technieken. Deze hybride oplossingen komen ook aan bod.

3.1 Richtingbepaling

3.1.1 Softwarematig

Met behulp van de reeds besproken technieken om posities te bepalen in hoofdstuk 2 kan men al een eind op weg om softwarematig deze data om te zetten naar een richting. De meest voor de hand liggende methode is uiteraard deze van de padbepaling. Wanneer een persoon in beweging is komen er steeds nieuwe posities voort uit de positiebepalingstechnieken. Hiermee kan het gevolgde pad van een persoon in kaart gebracht worden. Op basis daarvan kan de richting die de persoon uitwandelt of waarin het voertuig zich beweegt makkelijk afgeleid worden. De richting tijdens het bewegen is dan ook met een redelijke zekerheid te bepalen. Wanneer een persoon zich te voet voortbeweegt is het uiteraard mogelijk dat deze zich plots omkeert, maar bij voertuigen is dit uiteraard al minder makkelijk. De problemen treden pas op wanneer de persoon of het voertuig stilstaat. Bij bijvoorbeeld wagens is het bij een positienauwkeurigheid van een paar meter nog wel te achterhalen wanneer een wagen zich keert, maar bij voetgangers is het uiteraard niet te voorspellen uit de positie alleen waarnaar een stilstaand persoon gericht staat.

Toch zijn er ook hiervoor een paar technieken te bedenken alhoewel deze al minder voor de hand liggen. Het is dikwijls nog wel te bepalen of een persoon zich voortbeweegt of stilstaat. Wanneer in een Wireless LAN techniek een stilstand gedecteerd wordt kan aan de veranderingen in signaalsterktes afgeleid worden waar een persoon naar gericht is. De positie net voor de stilstand kon nog een vrij behoorlijke schatting voortbrengen van de richting op basis van het pad. Uit de veranderde signaalsterktes kan men dan gaan afleiden hoe deze richting veranderd is. Het is inderdaad zo dat wanneer een persoon zich met zijn rug richt naar een access point de stralen zwakker zullen aankomen bij het toestel aangezien het lichaam van de gebruiker deze afschermt. Net omgekeerd als een signaal uit een bepaalde access point plots versterkt kan misschien worden afgeleid dat de persoon zich meer gericht heeft naar dat access point toe. Toch is dit verre van een garantie aangezien het grote probleem met

Wireless LAN de sterk fluctuerende signaalsterktes in functie van de tijd zijn.

In het vorige hoofdstuk tijdens het deel over IrDA is reeds een andere techniek voor richtingbepaling ter sprake gekomen. Door de beperkte invalshoek waaronder infrarode signalen opgevangen kunnen worden is het bij IrDA wel mogelijk om te bepalen of een gebruiker zijn toestel naar een bepaalde zender heeft gericht.

3.1.2 Elektronisch kompas

Een andere, hardwarematige oplossing voor de bepaling van de richting van een toestel is het elektronische of digitale kompas. Er zijn op de markt al heel wat modellen op de markt (soms geïntegreerd in navigatietoestellen). Een mogelijkheid om elektronisch de richting te bepalen is de Hall methode. Wanneer een elektrische stroom door een plat staafje gestuurd wordt zullen de elektronen in deze stroom door invloed van een magnetisch veld zich meer gaan ophopen aan de ene of de andere kant in het staafje. Door de verschillen in voltages op meerdere posities op het staafje op te meten kan afgeleid worden waar het noorden ligt. Bij deze vorm is het wel noodzakelijk dat het kompas horizontaal gehouden wordt aangezien de verticale factor van het magnetisch veld van de aarde niet op deze manier bepaald kan worden. Het spreekt voor zich dat een dergelijk systeem zijn beperkingen heeft. Ten eerste moet het goed opgesteld worden. Bovendien moet het gebied waarin het gebruikt wordt vrij zijn van magnetische storingen.

3.1.3 Direction Finding Antenna

De direction finding antenna is een antenne die bestaat uit een N aantal six-port reflectometers. Dit zijn kleine radioontvangers die vaak gebruikt worden in hoge-frequentie elektronica. Het wordt het meest gebruikt om de reflectie coëfficiënt te meten. Zowel de amplitude als de fase van een signaal worden dan opgevangen. Zo heeft de direction finding antenne er dus een N-tal. Doordat er meerdere van deze ontvangers aanwezig zijn in de antenne kan men de invalrichting van signalen met dezelfde frequentie gaan opmeten. Hierdoor kan dan een algemene richting afgeleid gaan worden.

3.2 Hybride Systemen

3.2.1 Algemeen

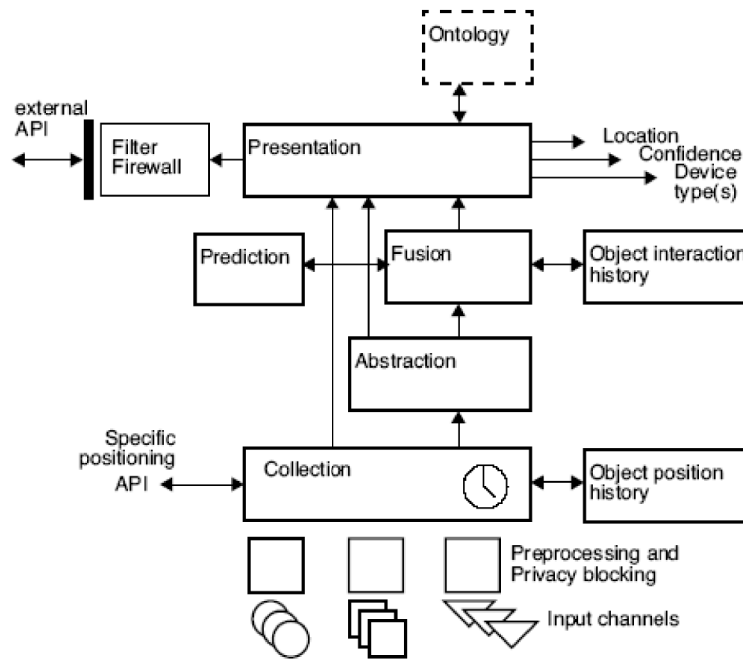
We zijn ondertussen al een hele reeks technieken tegengekomen om posities te bepalen. Maar steeds werden ze apart gebruikt. Het spreekt voor zich dat wanneer verschillende systemen samengevoegd worden dat men tot een beter resultaat zal komen. Bovendien kan in sommige gevallen ook informatie afgeleid worden uit de context of uit de geschiedenis van vorige posities: position inference. In de meeste gevallen zal het samenvoegen van data uit verschillende technieken en uit position inference een veel betere localisatie bezorgen.

Het is echter niet evident om al deze technieken laten samen te vloeien om uiteindelijk tot één enkele positie te komen, wat uiteraard de eindbedoeling zal worden. We bespreken nu een architectuur die voorgesteld is in [33]. Een ideale architectuur voor positiebepaling zou een combinatie van alle mogelijke systemen zijn. Maar om posities van verschillende systemen te vergelijken moet eerst het begrip 'degrees of imperfection' ingevoerd

worden. Sommige systemen hebben namelijk een beperkte precisie waardoor er meetfouten op de meting zitten (Dilution of Precision of DoP). Bovendien moet er rekening gehouden worden met hoe lang een meting correct blijft. Hierover moet een voorspelling gebeuren wat ook weer aanleiding geeft tot fouten. Als iemand's kaart getrackt wordt aan een bankautomaat dan mag men aannemen dat de persoon die de kaart gebruikt niet langer dan een paar tientallen seconden op die plaats zal blijven (tot zijn transactie is afgerond). Terwijl wanneer iemand met zijn bankkaart betaalt in een cafetaria zou je mogen aannemen dat die persoon een paar tientallen minuten op die locatie zal blijven om zijn maaltijd te nuttigen. Maar hier kunnen dan al fouten optreden aangezien een persoon tijdens zijn maaltijd plots kan weggeroepen worden. Bovendien kunnen relaties tussen objecten veranderen. Als je een persoon zijn positie bepaalt door een object te volgen van deze persoon is het natuurlijk altijd mogelijk dat deze dat doorgeeft aan iemand anders waardoor alle aannames op basis van de positie van dat object fout zijn. Wanneer mensen iemand herkennen gebeurt dat dan ook aan de hand van een heleboel kenmerken. Als iemand een nieuwe bril opheeft zullen we deze persoon toch nog herkennen door alle aanvullende informatie. Er zijn dus meerdere redenen waarom een meting fout kan zijn en daarom is het nodig om aan een meting ook een numerieke betrouwbaarheid te koppelen.

De invoerkanalen van een positiebepalingsarchitectuur zijn bijvoorbeeld:

- GPS: Ook voor indoor positiebepaling kan GPS informatie gebruikt worden. Wanneer het GPS signaal plots wegvalt nabij de ingang van een gebouw mag aangenomen worden dat de bezitter van het toestel net dat gebouw binnen is gegaan.
- RFID: Het inlezen van een RFID-tag nabij bijvoorbeeld een in- of doorgang kan de nabijheid van een persoon detecteren.
- Visuele trackers: Het beeld van een bewakingscamera kan uit de contouren afleiden waar op het beeld zich mensen bevinden. Wanneer deze informatie gekoppeld wordt aan andere metingen kan men bepaalde personen mappen op deze locaties.
- Floors: Vloersensoren (zoals het smart carpet) kunnen het traject van iemand detecteren.
- Wireless Sensor Networks (WSN): Dit soort systemen kunnen de nabijheid van personen checken (bijvoorbeeld door een temperatuursmeting).
- Active Badges
- Biometric Sensors: Biometrische kunnen geen absolute maar wel een hoge betrouwbaarheid garanderen voor de authenticatie.
- Ultrasonen en Ultra-Wideband positionering: Als de kosten per kamer binnen te perken kunnen blijven kunnen ze andere kanalen helpen calibreren.
- WiFi Positionering: Deze vorm van positionering is erg degelijk en nogal precies. Het is dus uitstekend geschikt als basis om aangevuld te worden met de andere systemen.
- GSM: De cel en sub-cel positionering in het GSM-netwerk is in theorie bruikbaar maar door de concurrentieslag in de telecomindustrie zal het moeilijk worden om een systeem te maken dat gebaseerd is op samenwerking tussen verschillende toestellen.
- Locatie afleiden: De locatie kan afgeleid worden van toestellen die niet onmiddellijk gemaakt zijn om dit te doen (kaartlezers, Bluetooth-nabijheid, ...).



Figuur 3.1: Architectuur voorgesteld in figuur 1 van [33]

De architectuur (zie figuur 3.1) houdt rekening met een onbepaald aantal verschillende invoerkanalen en is opgebouwd uit de volgende vijf componenten: Deel A zorgt voor het pre-processen en de privacy. Een voorbeeld hiervan is een bewakingsysteem met gezichtsherkenning. Misschien mag het positiebepalingssysteem geen gebruik maken van deze gezichtsherkenning en moet deze functie verhinderd worden tijdens de communicatie. Het is soms ook noodzakelijk om bij een scanner informatie over sommige items niet door te geven. Zo kan men bij de uitgang van een firma bijvoorbeeld enkel checken op bedrijfsspullen terwijl een persoonlijke voorwerpen genegeerd zullen worden. Op deze manier kan de privacy gerespecteerd blijven. In deel B vindt het samenrapen van de verschillende invoermethodes plaats. Wanneer de info nog geen timestamp heeft meegekregen in deel A zal dit hier toegevoegd worden om zo makkelijker data te kunnen vergelijken en om een geschiedenis bij te kunnen houden. Eens de verschillende invoeren samengebracht zijn moeten ze nog geabstraheerd worden tot een éénduidige vorm. Dit gebeurt in deel C. In het Fusion gedeelte (deel D) wordt data van verschillende momenten samengevoegd om een hogere accuraatheid te bekomen. Bovendien kan er ook een geschiedenis bijgehouden worden van de interactie tussen objecten (welke objecten al gebruikt of gedragen zijn door welke personen bijvoorbeeld). Op basis hiervan kunnen dan voorspellingen gedaan worden. In deel E wordt de data gepresenteerd aan de eindgebruiker. Let wel dat soms een persoon getracked moet worden op basis van de objecten die hij/zij vaak meedraagt, maar soms is het ook nodig om het object zelf te lokaliseren. Daarom is het belangrijk om op te merken dat deze presentatielaag E naast laag D ook laag C en B kan gebruiken om informatie te presenteren. Van de ontologie-module, die achtergrondinformatie uit de omgeving kan koppelen aan het systeem, moet nog onderzocht worden of het een verbeterd resultaat kan opleveren. Door het toevoegen van een filter en firewall module kan de hoeveelheid en de soort data die uitgewisseld wordt aangepast worden aan de graad van authenticatie die de externe gebruiker kan afleveren.

Door meerdere domeinen te laten samenwerken met mekaar kan elk systeem instaan voor de positionering op een

beperkte ruimte. Hierdoor wordt de druk op elk systeem verminderd terwijl door de samenwerking van de systemen nog altijd evenveel informatie verkrijgbaar is.

3.2.2 Samenwerkingsproblemen

Wanneer we meerdere technieken willen laten samenwerken kan dit vaak voor heel wat problemen zorgen. Zo is er bijvoorbeeld het probleem van de integratie van Bluetooth en Wireless LAN. Beide systemen gebruiken namelijk signalen in de 2.4 GHz band. Tot overmaat van ramp gebruiken ze ook nog allebei verschillende manieren om de signalen over die band te verdelen. Wireless LAN gebruikt één van een drietal vastliggende en niet-overlappende deelbanden in de 2.4 GHz band. Dit wil zeggen dat tot drie verschillende toestellen tegelijk kunnen zenden omdat ze elk een andere deelband kunnen gebruiken. Hierdoor kunnen meerdere WLAN toestellen samen functioneren. Bluetooth echter gebruikt een heel ander systeem, namelijk dat van het frequency hopping. Aan een snelheid van 1600 hops per seconde kiest Bluetooth één van de 79 kleinere deelbanden van elk 1 MHz. Dit wil dus zeggen dat de Bluetooth signalen honderden keren per seconde de signalen van één WLAN toestel zullen verstoren dat aan het uitzenden is. Uiteraard is het mogelijk dat het Bluetooth apparaat de deelbanden die overlappen met een zendend WLAN apparaat zal overslaan. Maar dat wil wel zeggen dat er heel wat delay komt te zitten op het Bluetooth signaal aangezien telkens bij het overslaan van een deelband uiteraard niks verstuurd mag worden om het WLAN toestel niet te storen. In sommige toepassingen is dit nog tolereerbaar maar toepassingen die streaming data nodig hebben zoals Bluetooth oortelefoontjes zullen storingen verkrijgen in zo'n geval. Het is dus noodzakelijk dat bij een dergelijke samenwerking protocollen opgesteld moeten worden waardoor het één het ander niet stoort. Een toestel zal dus bijvoorbeeld geen pakket gaan versturen wanneer een gesprek via een Bluetooth oortelefoontje gevoerd wordt.

3.2.3 Privacy en beveiliging

Bij het samenwerken van meerdere systemen moet een firewall voorzien worden om ook hier weer de samenwerking van de systemen te laten afhangen van de authenticatiegraad. Bovendien ontstaan er ook beveiligings- en privacyproblemen wanneer informatie wordt uitgewisseld tussen verschillende onafhankelijke systemen. Er zijn limieten binnen bepaalde technieken. Misschien kan bij een systeem heel wat informatie opgevraagd worden die de privacy kan schenden terwijl die niet gebruikt kan worden voor positiebepaling. Toch kan het systeem andere informatie doorspelen die wel interessant kan zijn. Wanneer een bepaald beveiligingsprotocol gebruikt wordt voor informatie in een dergelijke architectuur kan dit opgevangen worden doordat het systeem via dit protocol enkel niet-private informatie door zal sturen die nuttig kan zijn bij het localiseren van personen of objecten. Zo moet het kind niet met het badwater weggegooid worden.

4 Bestaande systemen

Er zijn reeds een groot aantal commerciële systemen op de markt die in min of meerdere mate aan positiebepaling doen. Ze hier allemaal gaan opnoemen zou uiteraard onbegonnen werk zijn, maar er volgt nu wel een greep uit de verkrijgbare systemen uit twee categorieën. De eerste categorie is die van de IR tags. Met infrarood signalen worden tags behorend bij een bepaalde persoon gescand op verschillende locaties. Op die manier kan het systeem bijhouden waar de persoon zich bevindt. Het tweede systeem is die van het Ultrasonic Ranging. Dit is een techniek die doet denken aan dat van de radar. Het werkt met de fluctuatie in frequentie van een radiogolf na weerkaatsing op objecten.

4.1 IR tags

4.1.1 Active Badge next generation

Deze techniek is een uitbreiding van het Active Badge systeem. Dit systeem werd ontwikkeld en getest tussen 1989 en 1992. Elke persoon heeft een badge die om de vijftien seconden een persoonlijke code uitzendt. Elke ruimte van een gebouw kan dan één of meerdere ontvangers bevatten die deze signalen kunnen opvangen. Op die manier kan dan bepaald worden in welk ruimte een persoon zich bevindt. Het systeem is oorspronkelijk ontwikkeld om de semafoons te vervangen. Met de active badges weet men meteen waar iemand zich bevindt en kan men meestal ook afleiden waar die persoon mee bezig is. Wanneer iemand in een bedrijf zich in het rooklokaal of een ontspanningsruimte bevindt weet men onmiddellijk dat de persoon vrij is.

De eerste werkende versie werd geïmplementeerd als hulpmiddel voor receptionistes. Het systeem kon een aantal queries uitvoeren op het systeem. Men kon de locatie opvragen van een persoon, dit was dan de locatie van de laatste zender die de code van diens badge ontvangen heeft. Ook kon opgevraagd worden welke personen zich in de buurt bevinden van een persoon. Dit gebeurt uiteraard op een gelijkaardige wijze. De personen wiens badgescodes laatst opgevangen zijn door dezelfde ontvangers of door ontvangers in dezelfde ruimte zullen als samen beschouwd worden. Men kan ook op dezelfde manier opvragen welke personen zich in een bepaalde ruimte bevinden. Er werd ook een functie voorzien die de receptionistes verwittigde op het moment dat iemands code opnieuw werd opgevangen door het systeem. Tenslotte kon van elke persoon ook een geschiedenis opgevraagd worden waar deze zoal had rondgehangen in het gebouw en voor hoe lang.

In de Active Badge next generation is de belangrijkste innovatie de zogenaamde Wonder Room. Er wordt dankzij het systeem een soort gepersonaliseerde ruimte gecreëerd wanneer deze die ruimte binnenkomt. Dit kan

gaan van het automatisch doorschakelen van telefoontjes naar het vast toestel in die ruimte tot het automatisch instellen van de belichting over andere toestellen in de ruimte naar de wensen van de gebruiker.

4.1.2 Versus

Het Versus-systeem is gelijkaardig aan het Active Badge systeem maar is speciaal ontwikkeld voor hospitalen. De niet al te zware infrarood signalen zijn uiteraard erg geschikt omdat ze veel minder dan andere signalen de toestellen in hospitalen kunnen verstoren. VIS¹ zorgt met behulp van badges die net zoals bij het Active Badge systeem een eigen unieke code verspreiden dat dokters, patiënten en zelfs bezoekers met de badges meteen opgespoord kunnen worden. Hierdoor wordt het zoeken naar een dokter die dringend bij een patiënt verwacht wordt een stuk makkelijker. Ook in dit systeem zit blijkbaar een techniek verwerkt om telefonisch meteen in contact te komen met die personen door middel van vaste toestellen. Uiteraard is dit erg belangrijk in ziekenhuizen aangezien gsm's uit den boze zijn vanwege de storingen in de apparaten van het ziekenhuis die ze veroorzaken.

4.2 Ultrasonic Ranging

4.2.1 Cricket

Cricket gebruikt naast ultrasone technologieën ook nog radiosignalen om localisaties af te leveren. Beacons die ofwel aan de plafonds ofwel aan de muren van een ruimte bevestigd zijn sturen radiosignalen uit op een bepaald kanaal. Tegelijkertijd wordt er ook een ultrasone trilling uitgestuurd. De ontvangers die op apparaten bevestigd kunnen worden, zullen wanneer ze de radiosignalen opvangen meteen beginnen te luisteren naar de ultrasone trilling. Op basis van de kennis over het verschil in voortplantingstijd van beide signalen kan dan een afstand tot de zenders afgeschat worden. Met behulp van een aantal algoritmes kan daar dan een behoorlijk nauwkeurige localisatie gaan bereikt worden. Naar eigen zeggen kan het systeem een nauwkeurigheid bereiken van één centimeter.

4.2.2 The Bat

The Bat werkt ongeveer gelijkaardig. Dit systeem werkt enkel met zenders aan het plafond. Deze sturen ook weer ultrasone trillingen uit. Wanneer de ontvanger deze opvangt zal een afstand tot de zenders afgeschat gaan worden op basis van de tijd die de trilling onderweg was. Drie punten zijn reeds voldoende om met triangulatie een locatie te verkrijgen. Bovendien kan het systeem ook op basis van de invalshoek van de trillingen een richting van de gebruiker gaan bepalen.

4.3 Wireless LAN

Ook al zijn er al een aantal van deze Wireless LAN positiebepalingssystemen verkrijgbaar, meestal worden er vrij weinig specifieke gegevens over de gebruikte technieken vrijgegeven. Enkel summiere verkoopsgegevens worden doorgespeeld. We bespreken kort een tweetal van deze systemen.

1 VIS: Versus Information System

4.3.1 WhereNet

Het WhereNet Real Time Locating System systeem bestaat uit twee apparaten. De WhereLAN Location Sensor en de Locating Access Point. De sensor en de access point ontvangen tag transmissies. Deze worden doorgestuurd samen met signaalinformatie naar een server die ook geleverd wordt door WhereNet. Alle berekeningen gaan op die server gebeuren. De database moet daarom enkel op de server aanwezig zijn. Het systeem kan een groot aantal devices tegelijk tracken.

4.3.2 Airlocation

Hitachi biedt hotspots aan die aan positiebepaling kunnen doen. Het systeem wordt "Airlocation" genoemd. Het systeem werkt zowel binnen als buiten en met een nauwkeurigheid van één tot drie meter door triangulatie. Het systeem heeft ook een ingebouwde beveiliging.

5 Implementatie

Het doel van de implementatie is het maken van een localisatielibrary met behulp van Wireless LAN die in verschillende toepassingen gebruikt kan worden. Het uiteindelijke resultaat moet met een minimum aan code gebruikt kunnen worden opdat het systeem makkelijk toegevoegd kan worden aan locatieafhankelijke applicaties. De werkwijze zal stapsgewijs gebeuren waardoor elk deelsysteem en elke optimalisatie afzonderlijk beoordeeld kan worden op efficiëntie. Bovendien zullen ook drie onafhankelijke systemen ontwikkeld en getest worden om de nauwkeurigheid van elke aanpak na te gaan.

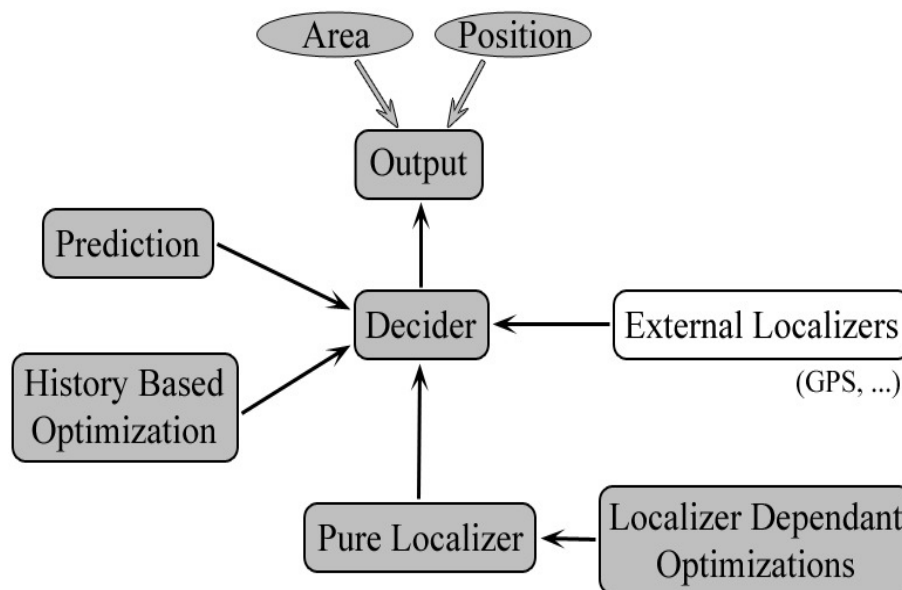
De eerste twee systemen zijn fingerprinting systemen. Het eerste systeem zal werken met statistische informatie over locaties. Deze wordt in een testfase opgedaan en bijgehouden in een database. Bij het gebruik van het systeem zal telkens nagegaan worden hoe groot de kans is dat de gebruiker zich op elk van de opgeslagen posities bevindt. Het tweede systeem is min of meer gelijkaardig. Ook nu wordt er met een database gewerkt met vooraf opgeslagen resultaten maar dit keer wordt de probabiliteit berekend op de signaalsterktes van elk access point: hoe groot is de kans dat signaalsterkte s van access point A opgemeten wordt op positie P. Tenslotte zal er nog een ander systeem uitgetest worden dat niet van fingerprints uitgaat maar wel van de posities van de access points. Er gaat telkens een schatting gemaakt worden van hoe ver men zich van elk access point bevindt. Deze afstanden tot de locaties van de access points kunnen dan gebruikt worden om een localisatie te bekomen.

Er zullen in alle systemen ook optimalisaties uitgetest worden. Deze zijn opgevat als algemene optimalisaties waardoor er zoveel mogelijk gestreefd wordt naar een implementatie waarbij ze in elk van de drie systemen kunnen gebruikt worden. Er zijn uiteraard een aantal optimalisaties die eigen zijn aan het gebruikte systeem. Maar technieken die werken op basis van voorgaande uitkomsten, de history-based systemen, kunnen overal toegevoegd worden. Ook verbeteringen op het vlak van de gemeten signaalsterktes kunnen in elk systeem ingebouwd worden. Het is ook de bedoeling dat deze systemen vertrekken van abstracte klassen waardoor bij het gebruik van het systeem eventueel zelf een optimalisatie geschreven kan worden die door het systeem gebruikt kan worden.

Wanneer in het nu volgend hoofdstuk gesproken wordt over een meting dan wordt er geen gemeten signaalsterkte bedoeld tenzij anders staat aangegeven. Er wordt hiermee dan een reeks van signaalsterktes bedoeld telkens gekoppeld aan een mac adres van het access point dat de beacon verstuurt. Een meting zal dus een lijst van tuples zijn: (mac 1, sterkte 25); (mac 2, sterkte 57); (mac 3, sterkte 39); ...

5.1 Algemene aanpak

In het concept diagram uit figuur 5.1 wordt de structuur van het voorgestelde systeem duidelijk. Het belangrijkste gedeelte is de 'Pure Localizer'. Elk van de drie besproken systemen in de inleiding van dit hoofdstuk zal zo'n 'Pure Localizer' gaan vormen. Het 'Localizer Dependant Optimization' concept beslaat alle optimalisaties die specifiek zijn voor elk systeem. Sommige van deze optimalisaties zullen echter ook door meerdere 'Pure Localizer' gebruikt kunnen worden.



Figuur 5.1: Concept diagram localisatiesysteem

Vervolgens is er het 'Decider' concept. Dit zal uiteindelijk gaan bepalen wat de applicaties die het systeem gebruiken zullen terugkrijgen via de 'Output'. Zoals zichtbaar in het concept diagram kan die output zowel een gebied – 'Area' – als een positie – 'Position' – zijn. Meestal zal de 'Decider' van een 'Localizer' een gebied terugkrijgen of een lijst van posities met een probabiliteit eraan gekoppeld. Als de gebruiker echter een exacte positie opvraagt zal de decider uit zijn informatie een positie gaan bepalen. Dit kan op verschillende manieren gebeuren dus er kunnen verschillende soorten deciders gemaakt worden. Als extra hulpmiddel voor de decider zijn er nog een drietal andere concepten voorzien. Ten eerste zijn er de 'External Localizers'. Het is niet ongebruikelijk dat een toestel dat gebruik zal maken van het systeem nog andere localisatiemethodes heeft zoals bijvoorbeeld een GPS ontvanger. Als de GPS informatie beschikbaar is, zou het onverstandig zijn deze niet te gebruiken. Een decider zal deze informatie kunnen gebruiken als aanvulling, of als er geen andere informatie beschikbaar is (een plek waar dus geen WLAN beacons opgevangen worden) de positie van de external localizer onmiddellijk doorspelen.

Tenslotte zijn er nog de 'Predicion' en de 'History Based Optimization' concepten. Beide systemen werken in tegenstelling tot de andere systemen niet enkel in het heden. Alle voorgaande concepten baseren zich voor de localisatie op informatie die men op het huidige moment heeft opgedaan. History based optimizations daarentegen gaan een aantal van de laatst verkregen posities bijhouden en op basis daarvan een verbetering

teruggeven. Dit kan eenvoudigweg door wanneer bij de laatste vier posities driemaal een ongeveer gelijkaardige positie werd opgemerkt maar ook één totaal afwijkende positie deze afwijking te gaan negeren. Het prediction concept gaat nog een stap verder. Ook hiervoor zijn een aantal voorgaande posities nodig maar het gaat ook een voorspelling maken van de toekomstige posities. Hierbij denken we dan spontaan aan een soort dead reckoning systeem waarbij op basis van de voorgaande posities en de daaruit af te leiden snelheid en richting een volgende positie voorspeld kan worden. Ook deze systemen kunnen uitschieters gaan detecteren. Wanneer een positie doorgekregen wordt die tien meter verder ligt dan de positie van één seconde geleden is het toegestaan aan te nemen dat de nieuwe positie verkeerd is (ten minste wanneer men met een redelijke zekerheid kan aannemen dat de vorige positie níet verkeerd is). Het zou namelijk vrij bizar zijn dat iemand ineens aan een snelheid van zesendertig kilometer per uur de laatste seconde heeft afgelegd.

Zoals reeds gezegd in de inleiding is het de bedoeling om elke toevoeging aan het systeem afzonderlijk te gaan beoordelen. Dit zal gebeuren door in de code steeds stap voor stap een nieuwigheid in te voegen en vervolgens meteen weer te gaan testen. Op die manier is het ook mogelijk om systemen die geen verbetering opleveren op te merken en te verwijderen uit het uiteindelijke systeem.

Om het systeem veel eenvoudiger te kunnen testen is er ook een voorbeeldapplicatie rond geschreven die het systeem zal gebruiken. Deze applicatie is een positietracker voor Pocket PC waarbij een plattegrond van een ruimte getoond wordt. De gebruiker kan simpelweg een fingerprint aan het systeem toevoegen en de database hiervan opslaan om later terug in te laden. Eens de database samengesteld is kan deze gebruikt gaan worden. Hierbij wordt dan ongeveer elke seconde een update van de geschatte positie weergegeven op de plattegrond. Om later makkelijk de nauwkeurigheid van het systeem te gaan bepalen wordt er gewerkt met logfiles die zowat alle informatie die opgedaan wordt bij gaat houden. Uiteraard is dit enkel in de debugfase nodig. Er zal ook een configuratiebestand gekoppeld worden waarin voor het uitvoeren van de applicatie een aantal instellingen van de applicatie gewijzigd kunnen worden. Dit uiteraard ook met het oog op de vergelijking van de verschillende uitbreidingen. Deze kunnen namelijk allemaal afzonderlijk aan en uitgeschakeld worden in het configuratiebestand.

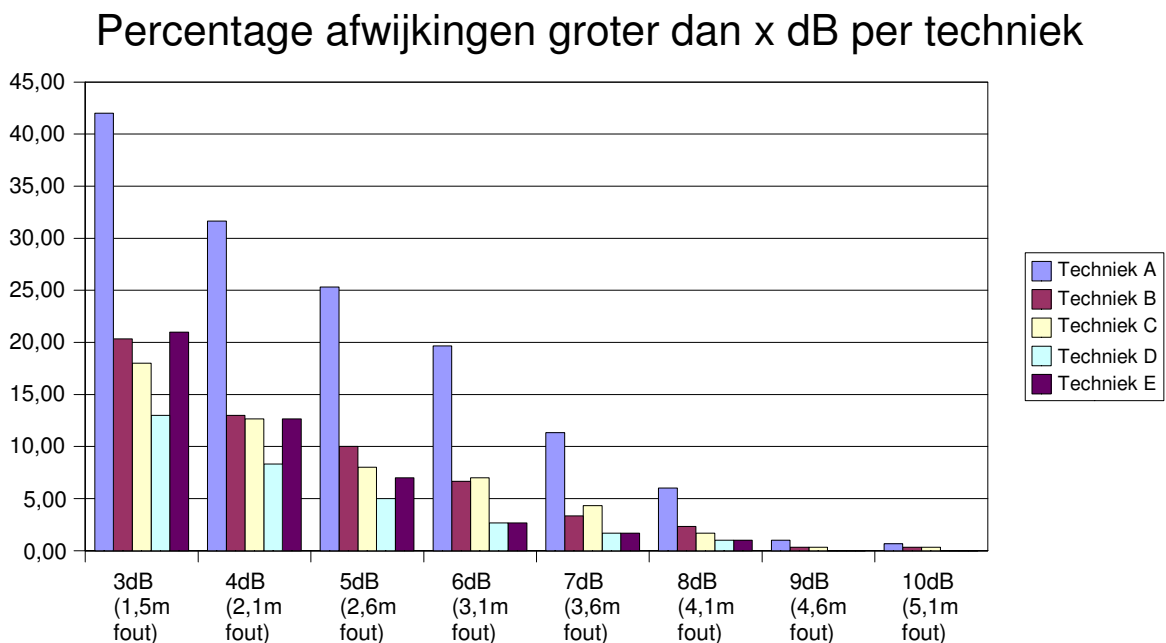
Ter voorbereiding op het systeem dat een positie zal bepalen op basis van een interpolatie tussen een aantal posities zullen de systemen eerst afgesteld worden door op zoek te gaan naar de dichtsbijzijnde fingerprint. Dit uiteraard enkel bij de twee systemen die gebruik maken van fingerprintingtechnieken. Wanneer met een redelijke zekerheid gesteld kan worden gezegd welk punt van een fingerprintmap het dichtsbijgelegen is, kan in een dichtere fingerprintmap een nauwkeurigere positie gaan bepaald worden door middel van interpolatie maar dan enkel van de punten die dichtbij het eerder gevonden punt liggen. Op die manier wordt voorkomen dat verafgelegen punten een negatieve invloed hebben op het resultaat. Zelfs al worden aan deze punten lage gewichten toegekend dan zullen ze toch nog een invloed hebben op de uitkomst van de interpolatie.

Bij de implementatie is ook gebruik gemaakt van het OpenNETCF pakket. Er moest namelijk op een manier te weten gekomen worden met welke signaalsterke de beacons van de op te vangen access points zijn aangekomen. Via een aantal functies van het OpenNETCF pakket kon die zonder problemen gebeuren.

5.2 Voorafgaande testen

Vooraleer te beginnen met het implementeren van het systeem moesten eerst een aantal tests uitgevoerd worden in verband met de verwachte fluctuaties in de metingen van de signaalsterkte. Deze testen zijn uitgevoerd met een linksys access point en een HP pda. Op drie verschillende afstanden tussen access point en pda werden telkens gedurende twee minuten zoveel mogelijk samples van de signaalssterkte genomen, dit waren op die twee minuten telkens honderd samples. Er werd gemeten op één meter, vijf meter en tien meter. Van al deze waardes werd dan een gemiddelde signaalssterkte berekend. Het bleek dat een getrimd gemiddelde – in dit geval een gemiddelde waarbij de laagste 25 % en de hoogste 25 % weggelaten wordt – weinig afweek van het gewone gemiddelde en dus is in de berekeningen steeds het gewone gemiddelde gebruikt.

Om deze afwijkingen wat te compenseren zijn een aantal eenvoudige technieken losgelaten op de gegevens. De eerste techniek was het weglaten van gegevens wanneer er een verschil van vijf decibel opgemeten werd ten opzichte van de vorige sterkte, in de grafiek is dit techniek B. Dit had natuurlijk als kwalijk gevolg dat niet in alle gevallen een resultaat kon berekend worden. Bij de afstanden van één en tien meter ongeveer in een kwart van de gevallen en bij de afstand vijf meter zelfs in de helft van de gevallen, werd het resultaat op deze manier verwaarloosd. Maar het verschil in de afwijkingen was duidelijk merkbaar. Vooral in de waarnemingen op vijf meter was het verschil groot. In plaats van 72 %, gaven nu slechts 33 % van de samples een verschil van drie decibel en 20 % in plaats van 52 % gaf een afwijking groter dan zes decibel.



Figuur 5.2: Percentage van de afwijkingen groter dan x aantal dB in functie van de gebruikte techniek

Een tweede techniek die toegepast is, is in plaats van steeds enkel de laatste waarde te beschouwen, het gemiddelde van de laatste twee of laatste drie waarden te beschouwen. Het gemiddelde van de laatste drie werd afzonderlijk berekend (techniek E) en ook in combinatie met de eerste techniek (techniek D) en het gemiddelde

van laatste twee is enkel beschouwd gecombineerd met de eerste techniek (techniek C).

De grafiek is als volgt opgesteld. Voor de waarden drie tot en met tien is er nagegaan voor hoeveel procent van de samples een afwijking groter dan deze waarden – ten opzichte van de gemiddelde waarde voor de bijhorende afstand – gemeten werd. Deze percentages zijn voor de drie afstanden gemeten en het gemiddelde percentage van de drie meetafstanden is in de grafiek opgenomen.

Volgens eerste gegevens wordt een fout van vijf decibel te veel of te weinig aangenomen als een fout van ongeveer 2,5 meter. Deze fout zou dus zeker niet meer dan vijf procent van de gevallen overschreden mogen worden om voldoende nauwkeurigheid te verkrijgen. Het is duidelijk dat met de eerste resultaten enkel techniek D dit resultaat haalt. Vanaf zes decibel is er echter weinig verschil te merken tussen techniek D en E en vanaf acht decibel gaven bijna alle technieken ongeveer gelijkaardige resultaten.

Op basis van deze resultaten is gekozen voor een eerste optimalisatie op de signaalsterktes zelf. Ten eerste is er een drempelwaarde aangemaakt. Wanneer afwijkingen in signaalsterktes vanaf een bepaald access point deze drempelwaarde overschrijden wordt gewoon verdergewerkt met de voorgaande sterkte. In de praktijk betekent dit dat een volgende rij van waardes:

15 14 13 15 17 25 15 16 13 15 20 25 27 24 26 28 29 30 27 28 35 27 26

met een drempelwaarde van vijf zal weergegeven worden aan het systeem als:

15 14 13 15 17 17 17 16 13 15 15 15 27 24 26 28 29 30 27 28 28 27 26

Tenslotte wordt er ook gewerkt met een gewogen gemiddelde van de laatste x resultaten. Ook deze x is net als de drempelwaarde door de gebruiker van het systeem – een andere applicatie dus – zelf in te geven. In de voorbeeldapplicatie zijn beide gegevens in te brengen in het configuratiebestand. Met het gewogen gemiddelde wordt de bovenstaande lijst verbeterd tot:

15 14,4 13,7 14,17 15,7 16,7 17 16,5 14,7 14,5 14,7 15 21 23,5 25,5 26,7 28,2 29,3 28,3 28 27,8 27,5
26,7

Het gewogen gemiddelde dat gebruikt is in deze verbetering is: $\frac{1}{6} \cdot x_{i-2} + \frac{1}{3} \cdot x_{i-1} + \frac{1}{2} \cdot x_i$ met i als index voor de laatst gemeten signaalssterkte.

5.3 Eerste systeem

Zoals reeds vermeld werkt het eerste systeem door aan een meting (lijst van tuples – zie pagina 34) een

probabiliteit toe te kennen ten opzichte van elk punt uit de database van fingerprints. De eerste zorg gaat om de reeds bekende redenen uit naar een techniek om het dichtsbijgelegen punt in de fingerprintmap te zoeken. De fingerprints in het eerste systeem worden bepaald door het nemen van een aantal samples. Ten allen tijden kunnen er samples aan een fingerprint toegevoegd worden om de nauwkeurigheid te verhogen. Wat per fingerprint bijgehouden wordt is uiteraard ten eerste een positie in units (het systeem werkt in elke gewenste maat zolang er maar consequent dezelfde maat aangehouden wordt). Verder gaan van de meegegeven samples gegevens bijgehouden worden per access point. De access points worden bepaald door hun mac-adres. Per access point wordt bijgehouden in hoeveel samples van de fingerprint een signaal van het access point binnengekeken is. Bovendien worden ook de minimale en maximale opgemeten signaalsterkte en de gemiddelde signaalsterkte van het access point bijgehouden. Omdat een signaalsterkte enkel maar een bedoeling heeft in combinatie met een mac-adres wordt een Signal-object gebruikt waarin zowel een mac-adres als de signaalsterkte wordt bijgehouden.

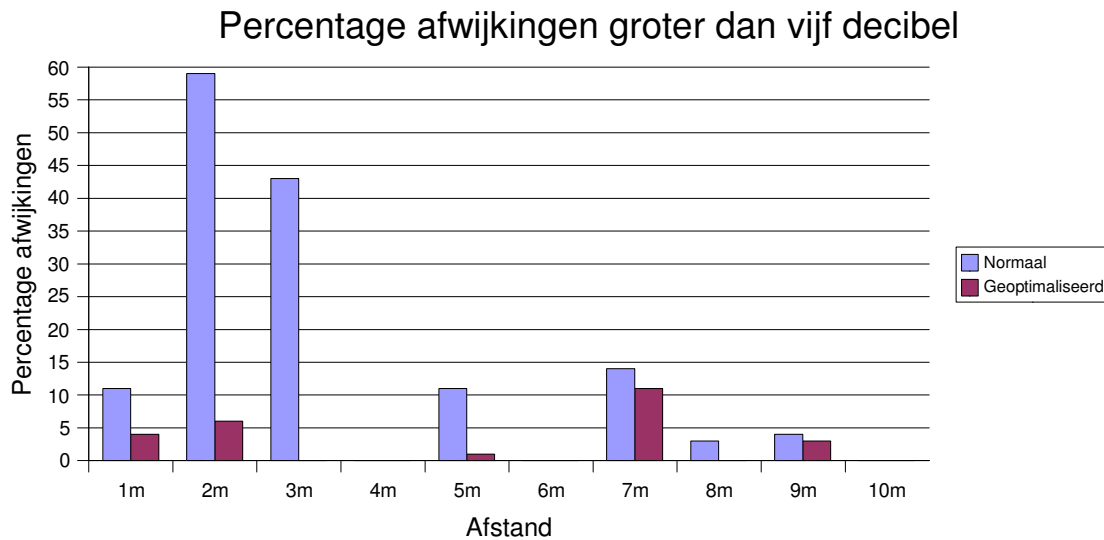
Wanneer een meting nu vergeleken moet worden met een fingerprint gaat het systeem eerst elk gemeten Signal-object vergelijken met de bijgehouden informatie van het mac-adres van het Signal-object. Dit gebeurt als volgt. Wanneer een sterkte slechts een paar decibel afwijkt van het gemiddelde zal de kleinste waarde (het grootst mogelijke gewicht) toegekend worden aan het signaal. Dit is de waarde nul. Wanneer de sterkte meer afwijkt maar zich toch nog binnen het gemeten interval van het mac-adres bevindt zal een waarde tussen nul en één teruggegeven worden, hoe groter de afwijking van het gemiddelde hoe groter het getal, maar steeds kleiner dan één. Wanneer het signaal zich niet meer binnen het interval bevindt zal een getal groter dan één teruggegeven worden.

Wanneer dit procédé herhaald is voor elk Signal van de meting wordt hierop een machtsverheffing uitgevoerd om de Signal's die het minst afwijken van de fingerprint belangrijker te maken. Door het getal te verkleinen, stijgt het gewicht en aangezien de minst afwijkende signalen tussen nul en één liggen zullen ze gaan verkleinen na een machtsverheffing met een exponent groter dan één, de getallen groter dan één zullen nog gaan vergroten en dus nog minder belangrijk worden in de vergelijking. De exponent van deze machtsverheffing kan vooraf ingegeven worden. Uiteindelijk zullen al deze waardes opgeteld gaan worden en gedeeld door het aantal signalen.

Aan dit algoritme zijn twee optimalisaties toegevoegd. De eerste kwam er na de observatie dat signalen dichtbij een access point vaak nauwkeuriger zijn dan signalen veraf. Daarom is een lineaire gewichtsfunctie toegevoegd die de vergelijking van een signaal met een hoge signaalsterkte een groter gewicht geeft dan diegene met een kleine sterkte. De andere optimalisatie is het minder belangrijk maken van signalen die in de fingerprint minder gemeten samples hebben (zoals eerder gezegd wordt dit per access point bijgehouden). Het is logisch dat op een plek waar niet altijd een signaal gemeten kan worden van een bepaalde access point, de signalen hiervan minder nauwkeurig zullen zijn dan die waarvan steeds een signaal binnengekeken wordt.

Nu volgt een korte bespreking van een aantal uitgevoerde tests met dit systeem. Allereerst zijn de voorafgaande tests die besproken zijn in het vorige punt herhaald met optimalisaties. De eerste tests waren het over een bepaalde tijdspanne opmeten van de sterktes steeds op een vaste afstand van het access point. Dit keer zijn deze

tests opnieuw uitgevoerd met de op basis van toenmalige resultaten voorgestelde optimalisaties om te checken of ze hun doel bereikt hebben. In de onderstaande grafiek krijgen we een overzicht van het aantal gevallen dat een opgemeten waarde afweek van het gemiddelde. Deze test is uitgevoerd op afstanden van één tot tien meter van het access point. Voor elke afstand zijn honderd metingen beschouwd. Steeds worden de niet geoptimaliseerde waarden vergeleken met de geoptimaliseerde waarden.



Figuur 5.3: Vergelijking van afwijkingen in signaalsterkte voor en na de optimalisaties

We zien in de grafiek van figuur 5.3 dat bij de geoptimaliseerde waardes slechts één keer de grens van de tien percent afwijkingen overschreden wordt, terwijl dit bij de normale resultaten vijf keer gebeurt. Bij de geoptimaliseerde waardes wordt ook slechts op twee afstanden de gewenste grens van vijf percent overschreden. Gemiddeld werden bij deze metingen in 14,5 % van de gevallen afwijkingen groter dan vijf decibel gemeten terwijl na het optimaliseren er slechts in 2,5 % een grotere afwijking was. Dit is dus bijna zes keer beter dan de opgemeten gegevens.

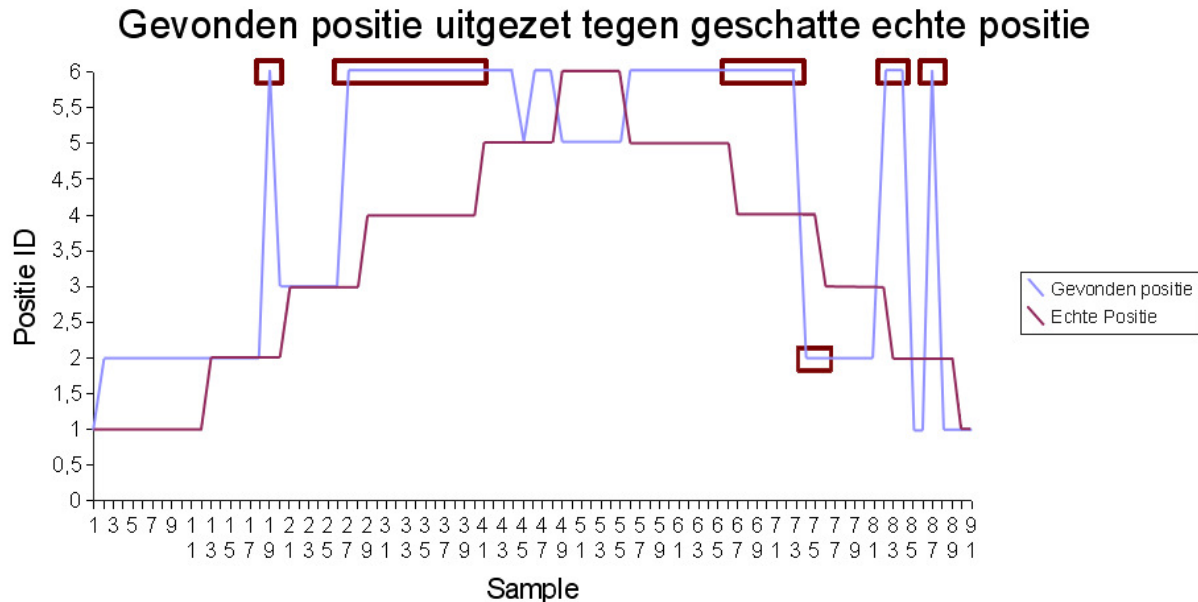
Er zijn ook een aantal tests uitgevoerd op het hele algoritme. In de eerste test is een beginsituatie gecreëerd van zes fingerprints op ongeveer vier à vijf meter afstand van mekaar. De fingerprints zijn weergegeven op onderstaande figuur 5.4:



Figuur 5.4: Zes fingerprints van de test uitgezet op een grondplan

De test is uitgevoerd door van de locatie van de eerste fingerprint op wandeltempo naar de locatie van de zesde fingerprint te gaan en weer terug. Door een mark in te geven in de logfile van het programma op het moment dat een nieuwe fingerprintlocatie bereikt werd kon bepaald worden wat de echte locatie was. Een locatie is in deze context één van de zes fingerprintlocaties. Deze kon dan vergeleken worden met de geschatte fingerprintlocatie.

Aangezien de échte locatie steeds ergens tussen twee fingerprints in ligt is een uitkomst die één fingerprintlocatie afwijkt van de 'juiste' locatie nog als goed te beschouwen. De schattingen die meer afwijken worden als slecht beschouwd. Op de onderstaande grafiek weergegeven in figuur 5.5 worden in functie van de tijd de geschatte fingerprintlocaties vergeleken met de 'juiste' fingerprintlocaties. De slechte resultaten zijn aangegeven met rode markeringsen.



Figuur 5.5: Gevonden posities vergeleken met de echte posities in functie van de tijd

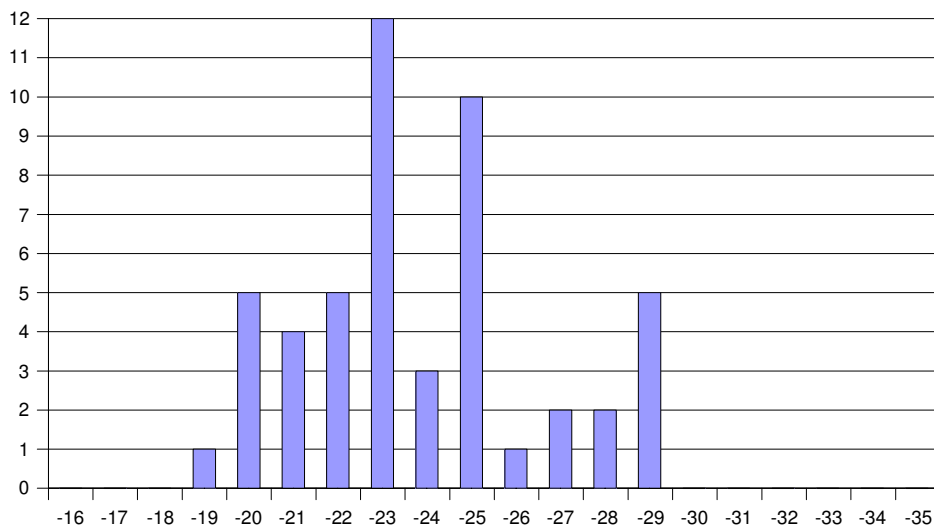
De tests leverden een redelijk resultaat op. Alhoewel slechts in 17,58 % van de gevallen een exacte positie verkregen werd, zijn er toch in 69,23 % van de gevallen goede resultaten gekomen.

Een tweede uitgevoerde test bestond in het lopen over hetzelfde parcours maar met een halt bij elk van de fingerprintposities. Aangezien steeds de gemiddeldes genomen worden van de laatste drie metingen duurt het meestal een tweetal seconden (één meting per seconde) eer het systeem echt bij is. Dit was duidelijk te merken in deze test. Wanneer bij het halthouden gecheckt werd welke fingerprint het systeem aangaf, duurde het soms een seconde of twee, maar in bijna 85 % procent van de gevallen werd dan wel degelijk de correcte fingerprint gevonden. Het is duidelijk dat in deze omstandigheden het systeem een nog beter resultaat geeft.

Een laatste test gaf dan weer teleurstellendere resultaten. Het ging over een gelijkaardige test als de vorige maar met fingerprints die slechts twee tot drie meter van mekaar lagen. In deze test konden niet eens in de helft van de gevallen de juiste fingerprint aangeduid worden. En in een aantal gevallen werd zelfs een totaal verkeerd resultaat gegeven. Om dit fenomeen te counteren zou best een functie ingebouwd worden in het systeem die kan afschatten of een fingerprint voldoende afwijkt van een andere fingerprint. Dit om dergelijke resultaten te vermijden. In de dichtere fingerprintmap kan dan eventueel wel minder afwijkende fingerprints gebruikt worden aangezien het resultaat op dat moment toch al met een behoorlijke nauwkeurigheid bekend is.

5.4 Tweede systeem

In het eerste systeem werd gebruik gemaakt van statistische informatie van elk trainingspunt zoals de gemiddelde signaalsterkte. In het tweede systeem wordt echter een histogram bijgehouden van elke sterkte die op het trainingspunt werd opgemeten en dit voor elk access point dat gevonden werd op dat punt. Een voorbeeld van een dergelijk histogram van in totaal vijftig samples gekoppeld aan één locatie en één accesspoint is te zien in figuur 5.6. Op de X-as staan de signaalssterkte in decibel gegeven en op de Y-as het aantal keer dat een sample deze signaalssterkte had.



Figuur 5.6: Histogram van gemeten signaalssterktes van één access point

Het eerste wat meteen in het oog springt is dat het histogram geen mooie verdeling vertoont. Het gaat dus niet op om het histogram gaan af te schatten door bijvoorbeeld een gaussiaanse functie. Het is dus nodig dat de volledige histograminformatie wordt bijgehouden in een database om er gebruik van te kunnen maken.

Deze informatie wordt als volgt bijgehouden. Ten eerste hebben we een object dat een locatie en een aantal bijhoudt. Dit wordt PointStat genoemd. In een StrengthInfo object wordt een dergelijk PointStat object bijgehouden voor elke mogelijke locatie. Dit StrengthInfo object wordt op zijn beurt dan weer bijgehouden in een APIInfo object. Per APIInfo object wordt er voor elke mogelijke signaalssterkte een StrengthInfo object bijgehouden. Deze signaalssterktes liggen typisch tussen -10 en -100 decibel. Er moeten dus per APIInfo object in het slechtste geval 90 StrengthInfo objecten bijgehouden worden. Deze hoeveelheid zal in de praktijk echter nooit voorkomen aangezien nooit elke signaalssterkte op een plaats zal worden opgemeten van één access point. Vervolgens wordt er in een APList object voor elk gevonden access point een APIInfo object bijgehouden op basis van het macadres. Een overzicht van deze structuur is dus als volgt:

APList: per mac adres:

 APIInfo: per signaalssterkte:

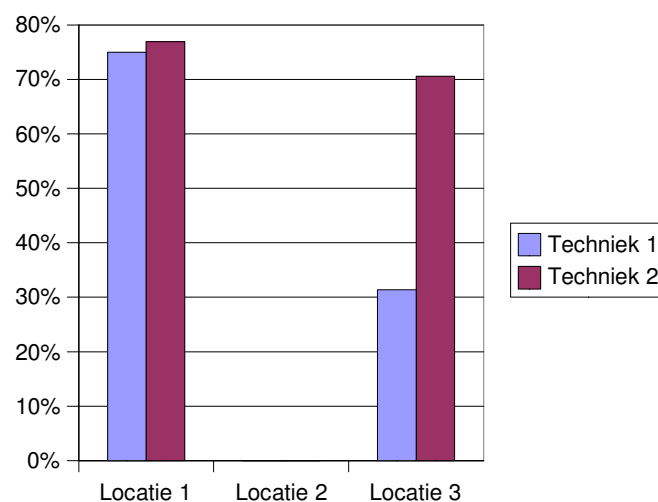
 StrengthInfo: per locatie:

 PointStat: locatie + aantal

Wanneer een meting op een locatie L wordt toegevoegd aan de database zal het systeem voor elk tuple van een macadres M en signaalssterkte S als volgt te werk gaan. In de APList wordt met behulp van adres M het bijhorende APIInfo object opgevraagd. Op basis van de signaalssterkte S zal in APIInfo een StrengthInfo gevonden worden. Ten eerste zal aan dit StrengthInfo object een PointStat object toegevoegd worden of wanneer deze al bestaat het aantal met één verhoogd worden. Bovendien zal ook het totale aantal van het StrengthInfo object verhoogd worden. Zo kan men makkelijk een probabilliteit verkrijgen door het aantal van een locatie te delen door het totale aantal van het StrengthInfo object.

Wanneer nu een meting moet vergeleken worden met de database gaat men gelijkaardig te werk. Opnieuw zal op basis van M en S een StrengthInfo object gevonden worden voor elk tuple van de meting. Voor elk tuple zal dus een lijst van locaties teruggegeven worden met hun probabilliteit op basis van de signaalssterkte. Als we dit in een voorbeeld bekijken waarin we een database hebben waarin van macadres M_1 vijftien keer een signaalssterkte van -25 werd gevonden op locatie L_1 en vijf keer dezelfde signaalssterkte op locatie L_2 , merken we dus dat locatie L_1 75 % probabilliteit krijgt toegewezen en locatie L_2 25 %. Door op die manier elk tuple af te gaan en de gevonden lijsten samen te voegen krijg je een hele lijst van locaties gekoppeld aan een probabilliteit. Door in deze lijst dan de maximale probabilliteit te zoeken en de bijhorende locatie terug te geven hebben we dan de meest waarschijnlijke locatie.

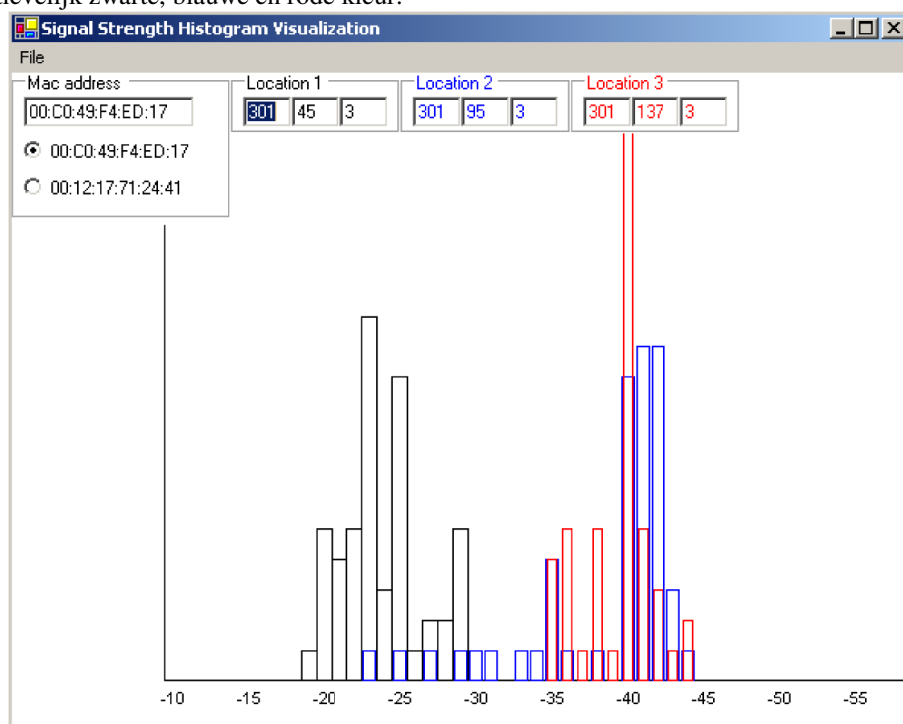
Om de techniek op zich te testen hebben we een eenvoudige opstelling gebruikt. Hierin werden twee access points op ongeveer acht meter van mekaar geplaatst. Vervolgens werd met het systeem telkens vijftig samples genomen op drie locaties: de eerste was nabij het eerste access point, de derde nabij het tweede access point en de tweede locatie was net in het midden van de andere twee. Bij het meten werd ook enkel rekening gehouden



Figuur 5.7: Resultaten van de basisversie

met de signalen van deze twee access points ook al konden er nog signalen ontvangen worden van andere. Op deze manier kon de techniek op zich beter getest en geëvalueerd worden. Tijdens het testen werden steeds twee technieken beschouwd. In de eerste werden de door het systeem gevonden probabilliteiten gewoon overgenomen om het maximum te zoeken. In de tweede techniek echter, werden deze probabilliteiten ook nog eens een gewicht toegekend op basis van het aantal samplepunten in een bepaalde locatie. Dit om na te gaan of het uitgangspunt

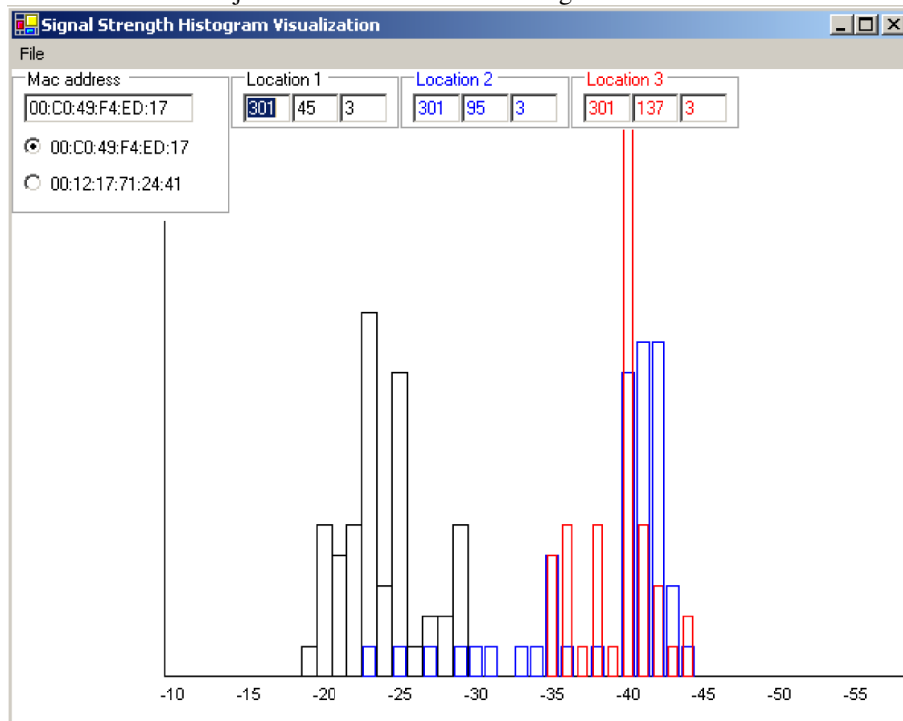
klopt dat de signaalssterktes die het meest voorkomen een belangrijkere waarde dient gegeven te worden om de nauwkeurigheid te verhogen. Dat een specifieke test op de overkoepelende techniek van het tweede systeem nodig was bleek al meteen uit de eerste test waarvan de resultaten te zien zijn in de grafiek in figuur 5.7. De eerste en derde locatie werden al vrij behoorlijk teruggevonden met de tweede techniek: respectievelijk in 77 en 70 procent van de gevallen werd de juiste locatie weergegeven. Met de eerste techniek was het derde punt meteen al veel minder goed terug te vinden, namelijk slechts in 31 procent van de gevallen. Met de tweede locatie was het echter dramatisch gesteld aangezien het nooit teruggevonden werd. Om het probleem bij dit punt in het midden van de twee access points beter te kunnen beoordelen is er een klein hulpprogramma gecreëerd om histogrammen bij elke locatie over mekaar weer te geven. In figuren 5.8 en 5.9 worden screenshots getoond van het programma met de histograminformatie voor twee access points. Links boven wordt het macadres van het access point ingegeven en bovenaan in het midden drie locaties waarvan de histogrammen weergegeven worden in een respectievelijk zwarte, blauwe en rode kleur.



Figuur 5.8: Screenshot hulpprogramma met de histogrammen van het eerste access point

Het eerste access point is dus hetgene dat nabij de eerste locatie gelegen is. De histogrammen hiervan worden getoond in figuur 5.8. We zien aan het histogram in het zwart dat op de eerste locatie duidelijk sterkere signalen opgevangen kunnen worden dan op de andere twee locaties. Wat ook meteen opvalt is dat de blauwe en rode histogrammen van de tweede en derde locatie sterkt door mekaar lopen. In de histogrammen van het tweede access point, getoond in figuur 5.9, zien we een gelijkaardig fenomeen. In dit geval staat het access point nabij de derde locatie. Dit valt op doordat hier de rode histogram duidelijk sterkere signaalssterktes weergeeft. En nog meer dan bij het vorige access point zien we dat hier de eerste en tweede locatie in mekaars vaarwater zitten. Dit is uiteraard de verklaring waarom de middenste locatie nooit teruggevonden werd. De signaalssterktes die bij locatie twee horen worden op basis van het eerste access point aan locatie twee en drie gekoppeld met een ongeveer gelijke probabiliteit. Maar op basis van het tweede access point werden ze gekoppeld aan de locatie één

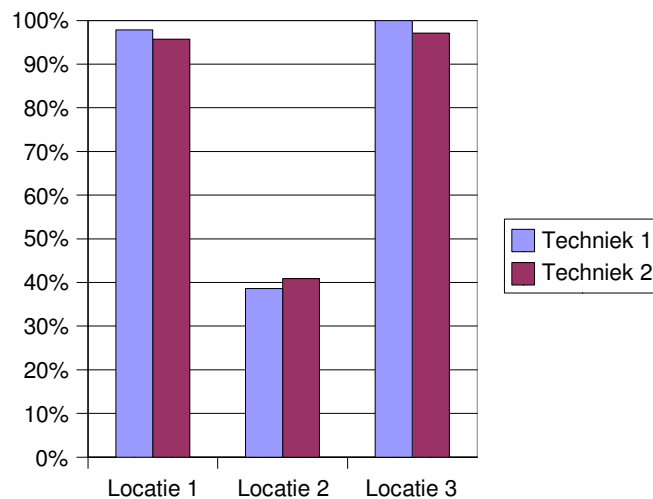
en twee. De probabilliteit die aan de eerste en derde locatie werd toegekend was dus steeds groter dan die die aan de tweede locatie werd toegekend omdat geen rekening gehouden werd met het aantal keer dat een locatie teruggevonden werd in de histogrammen. Aangezien locatie twee meestal twee keer teruggevonden zal worden – dit in de histograminformatie van zowel het eerste als het tweede access point – zou deze een hogere kans moeten maken om gekozen te worden terwijl dit in de basisversie niet het geval is.



Figuur 5.9: Screenshot hulpprogramma met de histogrammen van het tweede access point

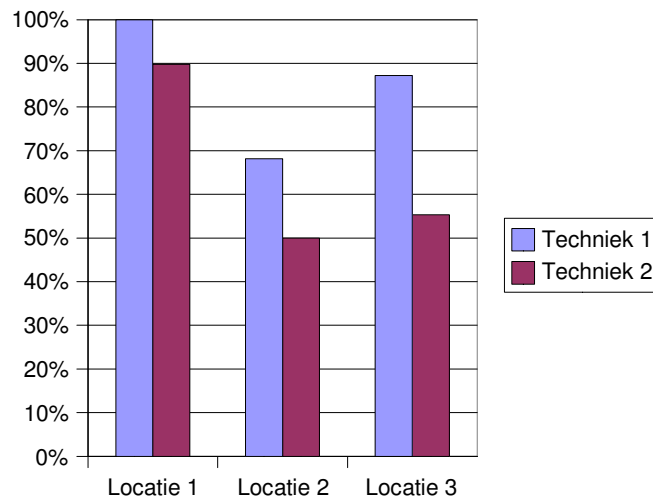
Hiermee rekening houden was dus de eerste aanpassing aan de basisversie. Door hoe meer access points een signaalssterkte werd teruggevonden op een bepaalde locatie, hoe belangrijker deze locatie moest worden in de probabiliteitsverdeling. In de eerste optimalisatie van het systeem werd dan ook een toenemend gewicht toegekend op basis van het aantal keer een locatie teruggevonden werd in de histograminformatie. De tests mét deze optimalisatie leverden de verwachte resultaten op. In figuur 5.10 worden de resultaten van deze test weergegeven.

We zien een duidelijke verbetering voor locatie twee. Deze werd nu al in ongeveer 40 procent van de gevallen teruggevonden. Bovendien zijn er met respectievelijk 98 en 100 procent van de gevallen met de eerste techniek ook duidelijke verbeteringen te zien voor locatie één en drie. Wat wel opvalt bij deze twee locaties is dat de tweede techniek slechtere resultaten geeft dan de eerste.



Figuur 5.10: Resultaten met de eerste optimalisatie

Een tweede optimalisatie voor dit systeem bestond erin elke probabiliteit ook te laten afhangen van het totaal aantal keer een sample teruggevonden werd bij access point. Inderdaad is het zo dat de access points waarvan weinig signalen opgevangen worden waarschijnlijk veraf zijn en dus een mindere rol mogen spelen in de beslissingsfase. Bovendien is ook hier net als bij het eerste systeem een optimalisatie toegevoegd door de probabiliteit te verhogen wanneer de gemeten signaalssterkte hoger is. Dit omwille van dezelfde reden als bij de tweede optimalisatie. Aangezien deze optimalisaties weinig zin hadden in de basisopstelling hebben we voor de laatste test van dit systeem alle teruggevonden access points laten meebepalen. De resultaten van deze test zijn te zien op figuur 5.11.



Figuur 5.11: Resultaten met alle optimalisaties en alle access points

Wat meteen opvalt is dat de tweede techniek hier nog slechter scoort dan in de vorige test. Het is dus duidelijk dat deze techniek achterwege gelaten moet worden. Ook hier scoort – tenminste wanneer we ons nu baseren op de eerste techniek – de eerste locatie weer honderd procent. De derde locatie wordt echter minder vaak herkend met een 87 procent. Maar dit is nog steeds meer dan behoorlijk. Bovendien wordt nu ook de tweede locatie dat in de vorige tests steeds het zorgenkind bleek, al in 68 procent van de gevallen correct teruggevonden.

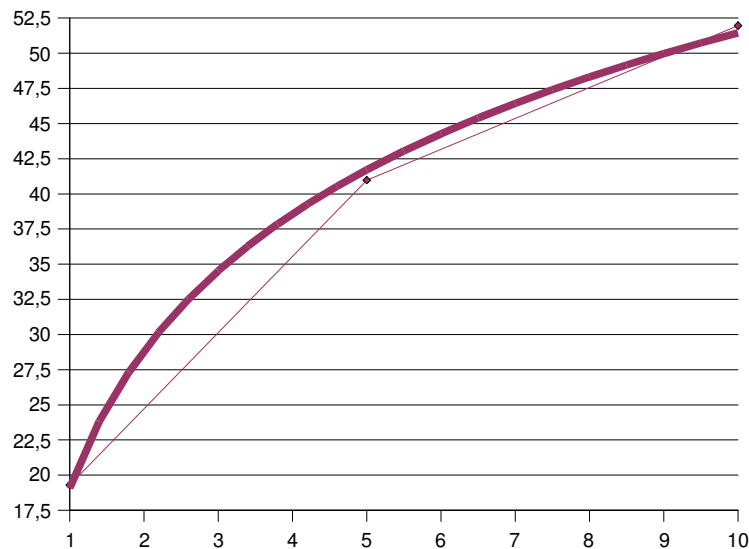
Er zijn ook nog bijkomende testen uitgevoerd met dit systeem. Deze verliepen steeds als volgt. Vooreerst werd een database opgesteld met een bepaald aantal posities. Vervolgens werden steeds op vaste posities een aantal metingen uitgevoerd. Later werd dan gecheckt welke posities het systeem had voorgesteld. Deze konden dan vergeleken worden met de exacte positie. Per geteste positie werd dan een gemiddelde afwijking gezocht van alle metingen. Door dan ook weer een gemiddelde te nemen van alle geteste posities kon dan nagegaan worden wat de totale gemiddelde afwijking of nauwkeurigheid is van het systeem. We hebben telkens de interpolatietechniek beschouwd naast de techniek van het dichtstbijzijnde punt uit de database. In totaliteit is bij de techniek van het dichtstbijzijnde punt een gemiddelde fout van 2,8 meter gevonden en bij de interpolatietechniek 4,84 meter. De reden dat de interpolatie het in deze tests nog zo slecht deed is aangezien er nog steeds rekening gehouden werd met alle punten. In tests die later besproken zullen worden zijn ook interpolatietechnieken getest die wél een keuze maken tussen de punten. Alle testen die later nog besproken worden zijn steeds op dezelfde manier gebeurd om de continuïteit van de resultaten te waarborgen.

5.5 Derde systeem

Het laatste systeem is er een gebaseerd op de locaties van de uitzendende access points. Het is inderdaad mogelijk dat wanneer we de afstanden kennen tot minstens drie access points dat we dan op basis van deze afstanden en de posities van de access points kunnen afschatten waar we ons bevinden. Deze techniek wordt ook wel triangulatie genoemd. Deze techniek is met Wireless LAN echter vrij problematisch gebleken. We zitten namelijk met het probleem dat we heel moeilijk de afstand tot een access point kunnen afschatten op basis van de signaalssterkte van dat access point. Om dit aan te tonen hebben we met behulp van de gemiddelde waardes op vaste afstanden uit de allereerste test een grafiek geplot met een voorgestelde afstandsfunctie. Deze functie is te zien op figuur 5.12. Op basis van deze gegevens zouden we wel een eenvoudige logaritmische functie of een vierkantsvergelijking kunnen opstellen die de afstand zou kunnen afschatten. Maar wanneer we gaan kijken naar resultaten van een bijkomende en nauwkeurigere test zien we al meteen meer problemen opduiken. Voor deze test werden de gemiddelde signaalssterktes opgemeten op één meter tot tien meter van eenzelfde access point telkens met één meter tussenafstand. Voor elke afstand werden honderd samples beschouwd. De resultaten van deze test zijn weergegeven in figuur 5.13. Wat meteen opvalt uit de vergelijking van de twee grafieken is dat er grote verschillen zijn. Laten we de afstand van een, vier en zeven meter beschouwen:

- Op 1 meter: grafiek 1: 19, grafiek 2: 29, verschil: 10
- Op 4 meter: grafiek 1: 38, grafiek 2: 45, verschil: 7
- Op 7 meter: grafiek 1: 46, grafiek 2: 52, verschil: 6

We zien dus erg grote verschillen. Bovendien zijn op de andere afstanden ook vaak verschillen van twee of drie decibel te vinden.



Figuur 5.12: Een eerste afstandsfunctie

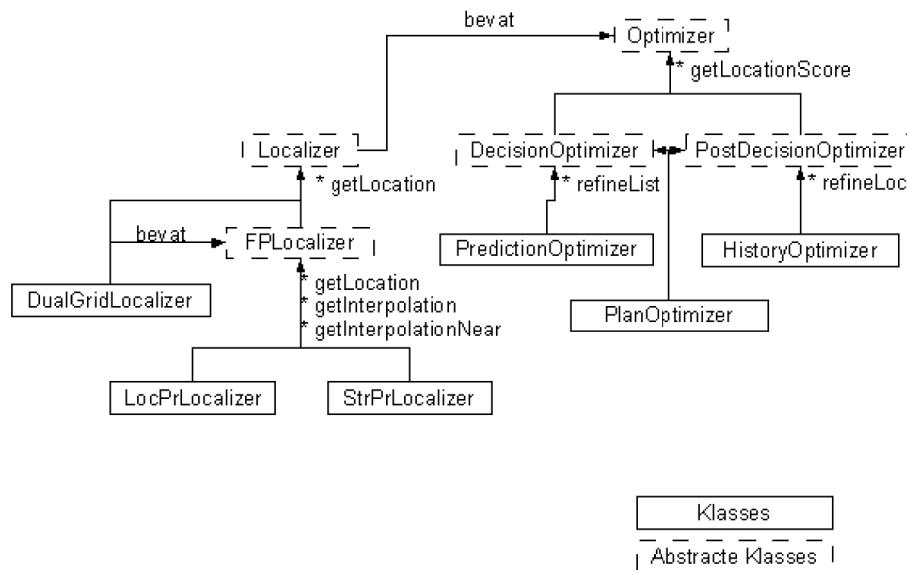
Maar dit is nog niet het grootste probleem. Wanneer we kijken naar figuur 5.13 zien we dat de functie niet constant stijgt. Dit is een wel erg vreemd gedrag, zowel na ongeveer vier meter als na ongeveer zeven meter – merk op dat dit zoals daarnet aangetoond werd de afstanden waren waar de grootste verschillen gemeten werden – een daling in de grafiek merkbaar is. Aangezien er in geen van de testsituaties objecten tussen de pda en de access point gebracht werden, is het dus totaal niet in te schatten hoe de functie precies zou moeten verlopen. Bovendien is het ook onmogelijk om te checken of er mensen tussen de access point en de ontvanger staan waardoor deze obstakels niet in rekening kunnen gebracht worden terwijl ze toch een sterke invloed hebben op het signaal en dus op de geschatte afstand. Aangezien het vrijwel onmogelijk is om een fatsoenlijke afstandsfunctie op te stellen is het duidelijk dat deze techniek veel minder geschikt is dan de twee voorgesteld in de vorige deelhoofdstukken. Daar de testen hieromtrent zo tegenvallende resultaten hadden aangetoond werd het niet opportuun geacht dit systeem verder uit te werken. De triangulatietechniek op basis van Wireless LAN zal duidelijk minder nauwkeurig zijn dan de fingerprinting systemen. Om deze reden werd het dan ook niet opportuun geacht om deze techniek verder uit te werken.

5.6 Gebruik van het systeem

In figuur 5.14 zien we de klassestructuur van de belangrijkste klassen van het systeem. Ten eerste zijn er de twee abstracte klassen Localizer en Optimizer. De localizers zijn de eigenlijke systemen die de algemene localisatietechnieken gaan verzorgen. De optimizers zijn klassen die voor algemene verbeteringen kunnen zorgen. Dit wil zeggen dat ze dus niet afhankelijk zijn van de gebruikte techniek en dus aan alle localizers kunnen toegevoegd worden.

Ten eerste gaan we de structuur van de localizers bespreken. De systemen die aan fingerprinting doen worden ondergebracht in de abstracte klasse FPLocalizer. Deze hebben naast de standaard getLocation functie ook een getInterpolation en een getInterpolationNear functie om respectievelijk een interpolatie van een aantal goed matchende fingerprints te krijgen en een interpolatie van fingerprints die nabij een bepaalde locatie liggen. De

geïmplementeerde voorbeelden van deze FPLocalizers zijn de LocPrLocalizer dewelke locaties gaat teruggeven op basis van vergelijkingen van de huidige meting met de gemiddeldes in de database en StrPrLocalizer die een localisatie uitvoert op basis van de histogrammen. Beide systemen zijn hierboven in dit hoofdstuk uitgelegd. Tenslotte is er ook nog de DualGridLocalizer. Deze gaat als parameters een localizer en een FPLocalizer meekrijgen. De localizer zal de eerste grid van het systeem gaan uitmaken en de FPLocalizer zal de tweede grid gaan uitmaken. Een verdere beschrijving van een systeem met twee grids wordt gegeven in het volgende hoofdstuk. Het komt erop neer dat er eerst een vaste locatie gevonden wordt met de gewone localizer. Vervolgens zal deze gebruikt worden in een specifieke FPLocalizer om een getInterpolationNear functie te voeden.



Figuur 5.14: Beperkt overzicht van de klassen van het systeem

De optimizers worden ook in twee soorten opgedeeld. De DecisionOptimizer gaat een lijst van locatie probabilities gaan verfijnen tijdens de beslissingsfase. Een voorbeeld hiervan is de PredictionOptimizer. De andere optimizersoort is de PostDecisionOptimizer. Deze gaat een positie verfijnen na de beslissingsfase. Tenslotte is er ook nog een PlanOptimizer die zowel een DecisionOptimizer als een PostDecisionOptimizer is en die informatie over de omgeving gaat gebruiken. Voor een uitgebreide uiteenzetting van hoe deze optimalisaties werken kan u terecht in het volgende hoofdstuk over de algemene verbeteringen.

Hoe moet een applicatie nu exact tewerkgaan om het systeem te gebruiken. Ten eerste moet wanneer een FPLocalizer gebruikt gaat worden de database gevoed worden. Dit doet men door met een device dat uiteindelijk gebruikt zal worden voor de applicatie op een aantal punten gaan staan in een grid en de locatie toevoegen aan de FPLocalizer. Bij elk punt moet men wachten tot er voldoende samples genomen zijn om de database te voeden. Eens deze informatie verzameld is kan met behulp van een functie de database gesaved worden. Later kan deze informatie eveneens met een functie terug ingeladen worden.

Eens dit gebeurd is kan de localizer gebruikt gaan worden. Hiervoor zijn twee methodes. Ofwel gaat de applicatie zelf een meting opvragen van de SignalAnalyzer. Deze meting kan dan meegegeven worden aan de

localizer om er een locatie uit af te leiden. Zowel het opvragen van een meting als het opvragen van de locatie gebeurt met het aanroepen van een functie. Men kan ook de eerste functie weglaten. Dan zal deze automatisch aangeroepen worden in de localizer. De reden waarom dit ook zelf door de applicatie kan gebeuren, buiten de localizer is dat op die manier een meting aan meerdere localizers kan gevoed worden. Wanneer we dus twee localizers gebruiken (zoals het geval is in de DualGridLocalizer) moet er slechts één meting genomen worden om beide localizers te voorzien. Als we weten dat het nemen van de meting de meeste tijd in beslag neemt dan is het duidelijk dat dit heel wat winst oplevert. Uiteraard is het ook altijd mogelijk om nieuwe technieken toe te gaan voegen door zelf localizers te gaan schrijven afgeleid van de abstracte klassen.

Ook bij de optimizers kan dit gebeuren. Het gebruiken van de optimizers gebeurt als volgt. Het enige wat moet gebeuren is een optimizer object aanmaken. Vervolgens moet deze dan toegevoegd worden aan één of meerdere localizer(s). Eens dit is gebeurd gaan de localizers zelf zorgen voor het gebruik van de optimalisaties.

6 Algemene verbeteringen

Zoals aangekondigd in het vorige hoofdstuk is het in het belang van de structuur van het systeem dat zoveel mogelijk verbeteringen algemeen gehouden worden zodat ze in alle gemaakte localisatietechnieken gebruikt kunnen worden. In het vorige hoofdstuk zijn bij elke techniek uiteraard een aantal specifieke verbetering besproken en nu wordt verdergegaan met het creëren van de algemene verbeteringen. Aangezien we gemerkt hebben dat het derde systeem in het vorige hoofdstuk er als de minst haalbare uitkwam zullen de verbeteringen toch vooral gericht zijn op de eerste twee technieken. Vooral ook omdat ze beiden werken met fingerprintingstechnieken waardoor de gelijkenissen in het systeem optimaal uitgebuit kunnen worden in het maken van de verbeteringen.

6.1 Toevoegen van de richting

Er zijn een aantal redenen waarom het handig is om bij posities in de database ook een richting bij te houden. Ten eerste is er het feit dat er daardoor zonder het systeem te moeten uitbreiden de richting kan gevonden worden van de gebruiker van het systeem. Het systeem gaat in de database van fingerprints opnieuw dezelfde technieken toepassen om de meest matchende fingerprint eruit te pikken, of tussen de meest matchende een positie – en nu daarbijhorend ook een richting – gaan interpoleren. Men krijgt dus in plaats van een locatie als returnwaarde een locatie én een richting. Het is uiteraard een heel handige uitbreiding die quasi geen aanpassingen verlangt van het systeem. Er moeten aan locaties gewoon een richting vastgeplakt worden en bij het ingeven van trainingsdata in de database moet dus ook een richting meegegeven worden. Verder verloopt alles volledig gelijkaardig.

De richting kan interessant zijn voor de toepassing en zijn gebruiker. Zo kan de applicatie namelijk weten in welke richting de gebruiker op dat moment kijkt. Dit is voornamelijk handig voor applicaties die als nut hebben het geven van extra informatie bij gebouwen of voorwerpen waar de gebruiker op dat moment naar kijkt. De richting kan echter ook nog eens nuttige informatie verlenen aan het systeem zelf. Dit uit zich op twee manieren. Ten eerste is er de storing in de signalen veroorzaakt door de gebruiker zelf. Naast het feit dat het intuïtief duidelijk is ook nog eens in tests opgedoken dat fingerprints danig kunnen verschillen naar gelang de positie van de gebruiker. Wanneer de gebruiker met zijn rug naar één van de access points gericht staat gaat deze hele andere geluidssterktes opmeten dan wanneer de gebruiker naar het access point kijkt. Wanneer je dus geen richtingen bijhoudt in de database is het dus nodig om bij het opstellen van de trainingsdata samples te nemen in zoveel mogelijk richtingen. Maar dit is natuurlijk niet ideaal. Daarom is het systeem met richtingen een stuk beter. De trainingsdata zal voor elk punt fingerprints bevatten in meerdere richtingen. Men kan bijvoorbeeld per punt in de database samples opmeten in vier of acht richtingen. Op die manier zal er voor elke richting van de gebruiker een betere afschatting kunnen gemaakt worden. Maar er is ook een tweede manier waarop het systeem de richting kan gaan gebruiken en dit in de Prediction systemen. Wanneer men een voorspelling gaat maken van de nieuwe richting op basis van de oude positie is het heel handig om te weten welke richting de gebruiker uitkijkt. Het is namelijk zo dat er een grotere kans bestaat dat de gebruiker zich in de die richting gaat voortbewegen. Dit is

uiteraard geen garantie, dus moet deze informatie in de Prediction systemen steeds gebruikt worden als waarschijnlijke informatie en niet als een zekerheid. Maar op deze problematiek wordt uitgebreid ingegaan in het volgende deel dat volledig gewijd is aan Prediction/History aanvullingen.

Uit testen is gebleken dat na het toevoegen van de richtingsinformatie een beter resultaat bekomen werd. Er zijn een aantal testen uitgevoerd waarbij in plaats van voor elke database positie één datalijst bij te houden twee of vier lijsten bij te houden voor elke hoofdrichting. Dit waren dan vier richtingen in open ruimtes en twee richtingen in bijvoorbeeld een gang waar mensen meestal slechts in twee richtingen bewegen. Net als alle andere testen voor de optimalisaties zijn deze uitgevoerd op het tweede systeem (dat van de signaalsterkte histogrammen). In het systeem van de dichtstbijgelegen punten werd een verbetering van een halve tot één meter opgemerkt en bij de interpolatie zelfs een verbetering tot anderhalve meter. Dit valt te verklaren door het groter aantal punten. Wanneer een databasepositie vier punten heeft voor elke richting, zullen deze meestal alle vier zwaar gaan doorwegen op het resultaat. De andere locaties zullen zelden met alle vier de punten gaan doorwegen. Hierdoor gaan deze posities dus ook een kleiner negatief effect hebben op het resultaat.

6.2 Prediction en History-based technieken

Aan het begin van het vorige hoofdstuk zijn deze technieken reeds aangehaald bij het beschrijven van het concept diagram van het systeem. Nu zullen we wat dieper ingaan op de werking van deze technieken en hoe ze ingevuld worden in het geïmplementeerde systeem. Hetgeen Prediction en History-based technieken gemeen hebben is het feit dat ze gebruik maken van vorige metingen om zo een verbetering gaan aan te brengen aan de huidige meting in het geval van de History-based technieken ofwel een voorspelling gaan maken van de volgende metingen in het geval van de Prediction technieken. We beschrijven kort beide systemen en hoe ze concreet worden ingevuld in ons systeem.

6.2.1 History-based technieken

Om de geschiedenis van posities te gaan gebruiken moet deze uiteraard eerst bijgehouden worden. Er zijn hier een aantal mogelijkheden denkbaar. Men kan bij elke meting bijvoorbeeld gaan bijhouden wat de afgeschatte positie was (eventueel aangevuld met de richting, zie vorig deel). Maar het is waarschijnlijk ook nuttig om gaan bij te houden wat de aangepaste positie is, zoals ze voorgesteld wordt nadat de history-technieken erop losgelaten zijn. In onze voorbeeldapplicatie wordt de positie elke seconde getrackt. Men krijgt dus nieuwe posities binnen op vaste tijdsintervallen. Maar om het systeem zo flexibel mogelijk te maken moet er ook rekening gehouden worden met applicaties die geen vaste tijdsintervallen hanteren. Het is niet ondenkbaar dat applicaties slechts op bepaalde momenten informatie over de locatie nodig hebben. Om deze gevallen op te vangen is het dus nodig dat er timestamps bijgehouden worden in de geschiedenis database.

Eens deze database is opgesteld kunnen we een aantal manieren bedenken hoe we er gebruik kunnen van maken. Deze database – of eventueel een andere maar gelijkaardige database – zal trouwens ook gebruikt worden bij de Prediction systemen. Het is uiteraard aan te raden om in dit geval dezelfde database te gebruiken aangezien beide

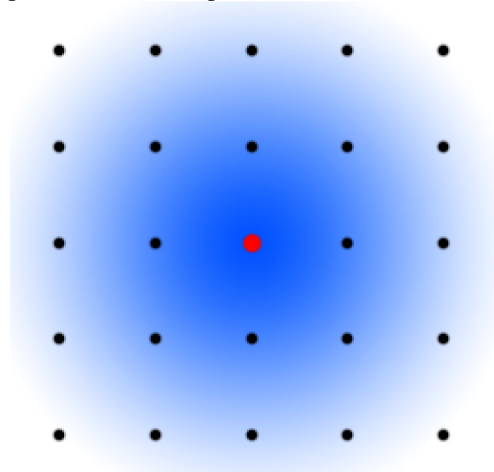
dezelfde informatie nodig hebben en we dus overbodig gebruik van het geheugen willen vermijden. In de history-based systemen zal deze database onder andere kunnen gebruikt worden om de huidige positie te gaan wijzigen in een gewogen gemiddelde van de vorige x aantal posities. Aangezien deze posities – zelfs diegene met een richting – steeds in een vectornotatie kunnen weergegeven worden is zo'n gewogen gemiddelde nemen ervan vrij eenvoudig. Zo'n gewogen gemiddelde kan er uiteraard voor zorgen dat uitschieters in de metingen verder worden weggewerkt. Inderdaad zullen deze uitschieters minder gaan doorwegen op de positie: in plaats van ineens een totaal andere positie weer te geven als positie zal er een weinig veranderde positie gegeven worden die meer in de richting gaat van de uitschieter. Toch is dit niet ideaal. Het is trouwens ook zo dat een uitschieter op deze manier ook langer een rol spelen terwijl dat deze in het normale systeem, dus zonder de history-based aanvulling, slechts bij één meting een rol gaat spelen. Daarom is het nodig om op een betere manier uitschieters gaan op te sporen. Dit kan bijvoorbeeld gebeuren door bij locaties waarvan de afstand tot de vorige locatie een drempelwaarde overschrijdt nog niet meteen de nieuwe positie terug te geven. Men kan ofwel gewoon de vorige positie terug gaan geven of de vorige positie met slechts een kleine beïnvloeding van de nieuwe positie. Wanneer dan bij de daaropvolgende meting een positie wordt weergegeven die weer meer aanleunt bij de eerste positie dan kan men er rustig van uitgaan dat de tweede positie een uitschieter was. En deze heeft dan geen invloed gehad op de localisatie. Daarentegen als in de nieuwe meting de tweede positie wordt bevestigd, wanneer de nieuwe positie dus dicht bij de tweede dan bij de eerste positie ligt, dan kan men er vanuit gaan dat het geen uitschieter betrof. In dit geval gaat men dan verderwerken met de derde positie, eventueel in combinatie met de tweede. In dit geval wordt de beslissing wel één of een paar metingen uitgesteld wanneer de gebruiker plots een serieuze afstand overbrugt, maar in de praktijk zal dit niet zo veel problemen opleveren. In ieder geval gaan ze minder storend overkomen dan af en toe een localisatie die twintig meter uit de richting is.

Er is een dergelijk systeem geïmplementeerd in het systeem. Deze houdt een aantal van de laatst voorgestelde locaties bij. Daarop zal er gaan gecheckt worden hoe ver deze punten verwijderd liggen van de nieuwe positie. Op die manier kan men achterhalen of die nieuwe positie realistisch is. Bovendien zal het systeem ook even afwachten vooraleer een positie gevalideerd wordt. Er moeten eerst een drietal posities na mekaar binnenkomen die dicht bij mekaar liggen vooraleer die positie wordt gevalideerd. In tegenstelling van wat verwacht werd geeft dit systeem zoals uit de testen bleek geen al te groot voordeel. Door de foutmarge op de testen waren de verschillen zelfs te klein om met zekerheid van een voordeel te spreken. Het is namelijk zo dat uitschieters wel degelijk verwijderd worden met deze techniek, maar het is ook mogelijk dat wanneer een uitschieter een paar keer na mekaar voorkomt dat deze dan gevalideerd wordt, waardoor het weer een tijdje duurt (meestal een drietal samples) tot terug de juiste positie wordt gevalideerd.

6.2.2 Prediction technieken

In het voorgestelde systeem zullen de prediction technieken dezelfde database gaan gebruiken als de history-based technieken. Men zal namelijk ook bij deze technieken gebruik moeten maken van de voorgaande posities. Maar deze keer om voorspellingen te gaan doen van de volgende posities. Deze kunnen dan gebruikt worden om een aanpassing te maken van de probabiliteit van bepaalde locaties. In tegenstelling tot de history-based technieken zullen de Prediction technieken gebruikt gaan worden nog voor het afsluiten van de localisatie in de

hoofdtechnieken. De history-based technieken gaan namelijk enkel de huidige geschatte positie aanpassen. Dit gebeurt dus nadat het localisatiesysteem een positie bepaald heeft. De Prediction systemen gaan daarentegen ingrijpen tijdens de beslissingsfase van het localiseren. Zoals reeds vermeld gaat het localisatiesysteem een lijst van posities teruggeven met daaraan gekoppeld een probabiliteit. Op basis van die probabiliteit zal dan ofwel de positie gekozen worden die het meeste kans maakt of een interpolatie van de meest waarschijnlijke posities. De Prediction-systemen gaan net die probabiliteiten aanpassen.

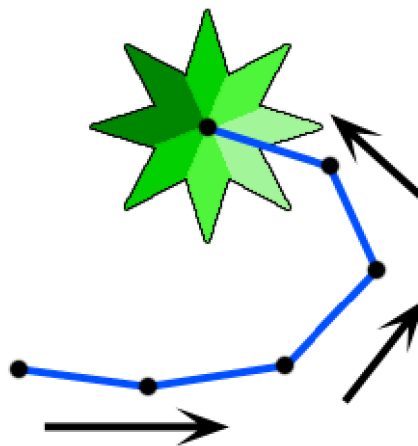


Figuur 6.1: Punten veraf van de huidige positie moeten een lagere probabiliteit krijgen

In figuur 6.1 zien we een afbeelding waarop in het rood de huidige positie is aangegeven. De zwarte stippen tonen een grid van eventuele fingerprintlocaties. In het blauw wordt nu aangegeven hoeveel kans een bepaald punt maakt om de volgende positie gaan te worden op basis van de afstand tot de huidige positie. Diegene in het blauw hebben een hoge probabiliteit, die in het lichtblauw een lagere probabiliteit en de punten in het wit hebben een probabiliteit van quasi nul. De grootte van deze cirkel hangt uiteraard af van de snelheid van de gebruiker. Dit zal zeker afhankelijk zijn van de applicatie. Het is goed in te beelden dat iemand in een museum als bezoeker zich veel minder snel zal gaan voortbewegen dan iemand die voor zijn werk bepaalde locaties moet afdraaien in een bedrijf. Daarom is het nodig dat de applicatie zelf aan het systeem kan meegeven wat de verwachte 'topsnelheid' van de gebruiker zal zijn. Op basis daarvan kan men dan gaan afschatten of de gebruiker een punt bereikt kan hebben op de tijd sinds de vorige meting.

Een Prediction systeem kan ook nog anders te werk gaan. Het kan op basis van een aantal van de vorige posities bijvoorbeeld gaan bepalen of de gebruiker stilstaat of zich voortbeweegt. Dit kan in vele applicaties handige informatie zijn. Wanneer de informatie die de applicatie moet geven op basis van de locatie erg summier is kan dit perfect realtime tijdens het wandelen geraadpleegd worden. Wanneer het echter gaat om meer gedetailleerde informatie is dit onnodig. Dan is het beter dat men tijdens het voortbewegen even geen nieuwe informatie gaat weergeven. Er kan dan beter gewacht worden tot de gebruiker opnieuw stilstaat om dan weer informatie gaan weer te geven op basis van de nieuwe positie. Tijdens het voortbewegen kan er ook een pad opgesteld worden dat de gebruiker bewandelt. Op basis van dit pad kan het systeem uiteraard ook een voorspelling gaan maken van de huidige richting die de gebruiker uitgaat. Wanneer we dit dan in verband brengen met het systeem met richting informatie kunnen we aan de posities in combinatie met de richting ook opnieuw de probabiliteit gaan aanpassen.

op basis van de afgeschatte richting die gehaald is uit de padinformatie. In figuur 6.2 zien we dit systeem uitgebeeld. In het blauw zien we het pad dat de gebruiker in de laatste zes metingen heeft afgelegd. In de achthoofdige ster zien we dan de acht mogelijke richtingen (dit kan natuurlijk ook een ander aantal richtingen zijn wanneer er meer of minder mogelijke richtingen gebruikt worden in de database). Aan elk van de acht mogelijke richtingen wordt een probabileriteit toegekend op basis van het pad. De richtingen die het donkerste groen hebben als kleur zullen plausibeler zijn dan diegene in het lichtere groen. Inderdaad het is beter te begrijpen dat een gebruiker gewoon zijn pad verderzet dan dat hij ineens rechtsomkeer maakt. Maar omdat dit natuurlijk niet volledig uit te sluiten is moet er toch nog altijd een voldoende probabileriteit gegeven worden aan de lichter gekleurde richtingen. Ook hier weer moeten we er vanuit gaan dat het slechts een schatting is en geen zekerheid. Op basis van deze nieuwe probabileriteiten van de richtingen kunnen dan de gevonden koppels van locaties met richting aangepast worden wat betreft hun probabileriteit op basis van hun eigen richting.



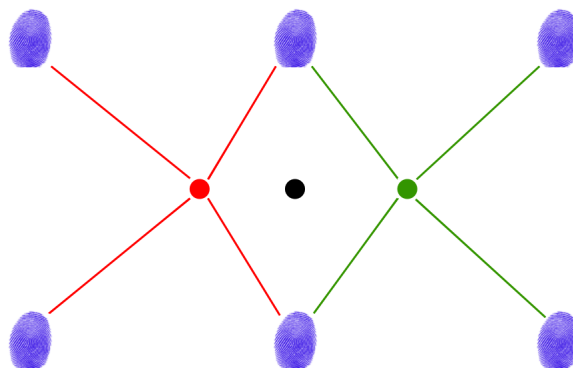
Figuur 6.2: Het pad dat een gebruiker aflegt en de voorspelling van de huidige richting

6.3 Interpolatie van posities

Er is al een paar keer eerder ter sprake gekomen dat we posities en ook de bijbehorende richtingen kunnen gaan interpoleren op basis van hun probabileriteit. In een dergelijk geval gaan we dus niet meer louter op zoek naar het meest plausibele positie-richting koppel, maar naar een aantal van de meest plausibele van dergelijke koppels. Wanneer we deze dan gevonden hebben kunnen we deze gaan uitmiddellen met als gewichten hun bijhorende probabileriteiten. Op die manier kunnen we gaan naar een meer algemene positiebepaling. We zijn namelijk niet meer beperkt tot het teruggeven van punten in de database. Tot hier toe gaven we telkens de positie terug van de locatie van de fingerprint die het meeste kans maakte om overeen te komen met de huidige meting. Nu kunnen we dus alle mogelijke posities gaan innemen gesteld dat ze zich binnen de grenzen van de databaselocaties bevinden uiteraard. Met dit feit moet dan wel rekening gaan gehouden worden bij het opstellen van de database. Het is namelijk niet meer voldoende om een aantal punten af te gaan waar de gebruiker zich vaak zal gaan bevinden, maar men moet ook de uitersten van de ruimten gaan in kaart brengen aangezien anders de grenzen te nauw zouden zijn. Wanneer er in een kamer bijvoorbeeld enkel in het midden punten worden toegevoegd aan de database zal er namelijk nooit een positie gevonden kunnen worden in de hoeken van de kamer.

Deze techniek brengt helaas wel enkel problemen met zich mee. Het grootste probleem is hier dat elk punt in principe kan meetellen. Het basissysteem bevat bij het vergelijken van de fingerprints met de huidige meting namelijk nog geen voorspellende informatie uit de Prediction systemen. Deze worden pas bij de beslissingsfase in rekening gebracht. Daardoor is het zo dat elk van de punten uit de database in theorie een mogelijkheid is. Dus moet er rekening gehouden gaan worden met de probabiliteiten van elk punt. Dit wil zeggen dat een punt dat veraf ligt van de huidige positie maar die toch nog een kleine probabilliteit wordt toegekend door het systeem, toch nog een invloed gaat hebben op de schatting. Het is namelijk zo dat hoewel de probabilliteit en dus bijgevolg ook het gewicht in de uitmiddelling klein is, het door de grote afstand tot de echte locatie van de gebruiker toch nog een rol gaat spelen in de schatting. En dit is uiteraard onaanvaardbaar.

Bovendien is het niet altijd verstandig om dit probleem op te lossen door een maximum aantal fingerprints te gaan beschouwen. Op het eerste zicht lijkt dit namelijk een meer dan behoorlijke tegemoetkoming voor het probleem. Als we bijvoorbeeld zouden kiezen enkel de vier meest plausibele locaties te gaan beschouwen, kunnen we er namelijk zeker van zijn dat veraf gelegen punten geen invloed gaan hebben op het resultaat aangezien deze nooit tot de vier grootste kanshebbers gaan behoren. Maar toch kan dit voor problemen zorgen. Dit wordt uitgebeeld in figuur 6.3.



Figuur 6.3: De blauwe fingerprints kunnen op de zwarte positie evenzeer naar de rode als naar de groene afwijken wanneer vier fingerprints beschouwd worden

We zien in de figuur dat bij het nemen van de vier belangrijkste fingerprints zowel de twee linkse als de twee rechtse fingerprints kunnen genomen worden in combinatie met de twee middenste. In de praktijk kan dit gaan betekenen dat het systeem constant gaat zitten switchen en er posities gaan teruggegeven worden die steeds gaan verspringen tussen de rode en groene positie. We zien dus dat nog een bijkomende verbetering nodig is.

Het is dan ook daarom dat we het reeds eerder voorgestelde principe van de dubbele fingerprint grids zullen toepassen. De bedoeling is als volgt. In plaats van met één grid te gaan werken waarin we met de vorige techniek gaan interpoleren is het beter om met twee grids te gaan werken. De eerste een grid waarin de punten vrij ver uit mekaar liggen en de tweede een veel dichtere grid. We kunnen dan eerst in de eerste grid het dichtsbijgelegen punt gaan opzoeken. Vervolgens kunnen we die informatie gaan gebruiken wanneer we het algoritme loslaten op

de tweede grid. In plaats van dus de vier beste punten uit die grid te gaan beschouwen gaan we enkel die punten beschouwen die het dichtst bij de in de eerste grid gevonden locatie liggen. Op die manier hebben we geen last meer van het vorige probleem.

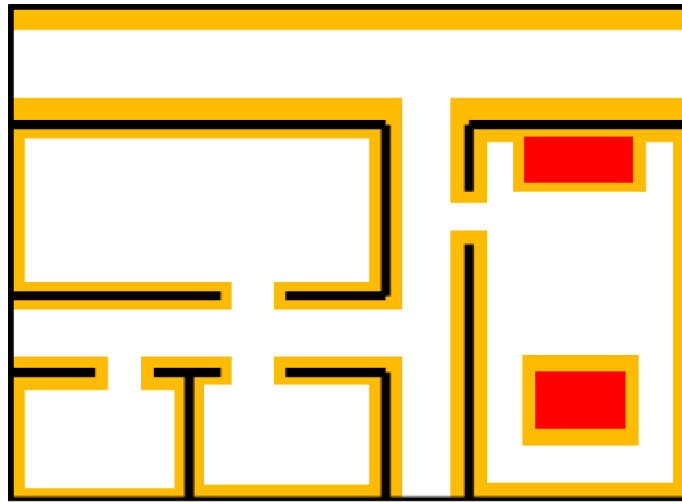
Hier moet echter een kanttekening bijgeplaatst worden. Het is namelijk vrij belangrijk om de juiste afstelling tussen de twee grids te vinden in combinatie met de afstand tot het punt uit de eerste grid waarop punten in de tweede grid nog beschouwd mogen worden. Het is namelijk zo dat wanneer die afstand en de grids zo gekozen worden dat bij elk punt uit grid één enkel exclusieve punten uit grid twee horen ook hier weer problemen gaan optreden. De gevonden punten gaan ook nu weer vaak verspringen en dit wanneer een nieuw punt gevonden wordt in grid één. Daarom is het belangrijk dat de afstand waarop punten nog beschouwd moeten worden ongeveer dubbel zo groot is – in ieder geval zeker groter dan één keer zo groot – als de afstand tussen de punten in grid één. Dit om geen al te gekke bokkensprongen te verkrijgen.

6.4 Informatie over de omgeving

Vaak, zoniet altijd, is het zo dat localisatiesystemen gebaseerd op Wireless LAN zich vooral toespitsen op indoor localisatie. Nu is het zo dat we vaak heel wat kunnen afleiden over de omgeving binnenin de gebouwen waarin we gaan localiseren. Zo hebben we bijvoorbeeld muren en andere obstakels waar we best wel rekening mee kunnen houden. We zijn bijvoorbeeld zeker dat een gebruiker zich niet op dezelfde plaats kan bevinden als een muur, dat hij geen pad kan volgen doorheen een muur, dat hij zich niet op of in bepaalde obstakels kan bevinden, ...

Om van deze informatie gebruik te kunnen maken stellen we nu een omgevingsoptimalisatie voor. De bedoeling is dat we gegevens over een gebouw die vast liggen zoals bepaalde zware grote obstakels of muren gaan vastleggen in een soort plan. Het spreekt uiteraard voor zich dat wanneer deze gegevens niet meteen voor handen liggen of niet makkelijk ingegeven kunnen worden (door bijvoorbeeld tijdsgebrek), dat dit systeem niet kan gebruikt worden. Maar dit is uiteraard geen ramp aangezien het enkel een aanvulling biedt voor het systeem en zeker niet noodzakelijk is. Maar wanneer we toch over deze informatie beschikken kunnen we ze gaan gebruiken in onze beslissingsfase of ook in de post-beslissingsfase. Maar hierover later meer. Laten we eerst eens een voorbeeld gaan bekijken van hoe dergelijke informatie er zou kunnen uitzien. In figuur 6.4 hebben we een plan van een aantal kamers in een gebouw gegeven.

Op het plan is in grijstinten een aantal obstakels aangegeven: in het donkergrijs de echte obstakels en in het zwart de muren. Bovendien zijn er in het lichtgrijs ook de gebieden aangegeven die vlak langs muren of obstakels lopen. Het is namelijk vrij ongebruikelijk (maar wel niet uitgesloten) dat iemand zich plat tegen de muur gaat aandrukken. De gebieden in het donkergrijs en in het zwart zijn dus zeker niet toegankelijk, de gebieden in het lichtgrijs zijn wel toegankelijk maar niet echt waarschijnlijk en de witte gebieden zijn de normale gebieden die wel toegankelijk zijn.



Figuur 6.4: Een gebouw waarvan we de informatie kunnen gebruiken bij de beslissingsfase

Zoals daarnet reeds aangegeven werd kunnen we deze informatie op twee momenten in het algoritme gaan gebruiken. Ten eerste kunnen we dit gaan doen tijdens de beslissingsfase. In deze fase hebben we dus een lijst van posities met elke een probabilliteit. Op basis van de informatie van de omgeving kunnen we een lagere probabilliteit gaan geven aan de punten in de lichtgrijze gebieden. Aangezien deze punten steeds locaties van de fingerprints zijn kunnen deze zich uiteraard niet gaan bevinden in de zwarte en donkergrijze gebieden aangezien deze niet toegankelijk zijn. Een op het eerste zicht logische opmerking zou zijn dat we bij het opstellen van de fingerprint database geen fingerprints gaan nemen in het lichtgrijze gebied. Dan kan deze fase gewoon overgeslagen worden. Maar dan vergeten we de opmerking uit het vorige hoofdstuk in verband met de interpolatie. Het is namelijk zo dat ook van alle uithoeken van de gebouwen zo dicht mogelijk tegen de grens punten moeten toegevoegd worden. Anders kunnen namelijk niet alle gebieden gevonden gaan worden door interpolatie.

Het tweede moment waarop we deze informatie gaan kunnen gebruiken is bij de post-beslissingsfase. Dit is dus wanneer het systeem al een locatie heeft bepaald. Aangezien deze locatie wellicht door interpolatie verkregen is kan deze uiteraard wel in de zwarte en donkergrijze gebieden gaan liggen. Toch weten we dat deze locatie dan onmogelijk is. Dan hebben we een aantal mogelijkheden om hieraan een oplossing te bieden. Ofwel kunnen we een techniek gaan gebruiken om het dichtsbijzijnde punt te gaan zoeken dat wel mogelijk is. Maar dit is uiteraard niet optimaal. Beter is het dat we het beslissingsproces opnieuw kunnen laten verlopen met een bepaalde constraint rondom die locatie. Een volledig andere mogelijkheid bestaat erin deze informatie te gaan gebruiken op het moment dat de locatie gaat bepaald worden bij de interpolatie. Wanneer dan een onmogelijke positie verkregen wordt kunnen we met behulp van een aantal extra gewichten op bepaalde posities een andere interpolatie gaan opvragen. Dit proces kan dan herhaald worden tot een mogelijke positie gevonden is.

Bovendien is de omgevingsinformatie ook nuttig in de Prediction fase. Wanneer we namelijk een pad gaan opstellen met behulp van alle vorige punten kunnen we gaan nagaan of dit pad wel mogelijk is op basis van de informatie over de muren.

De omgevingsoptimalisatie is geïmplementeerd en getest in het systeem. De resultaten zijn zeker even goed als deze zonder het systeem maar een grote winst in nauwkeurigheid levert het niet op. Het is dan ook voornamelijk bedoeld om kleine afwijkingen die een locatie aangeven in een muur of dichtbij een obstakel licht te gaan bijstellen zodat de locatie realistischer wordt. Dit levert uiteraard ook een kleine verbetering op in de nauwkeurigheid maar deze is gezien de foutenmarges verwaarloosbaar.

7 Voorbeeldapplicatie

Zoals reeds vermeld is de voorbeeldapplicatie die gemaakt is om het localisatiesysteem uit te testen een simpele locatie tracker bedoeld voor Pocket PC. Er wordt dus niet echt gebruik gemaakt van de locatie-informatie, maar de locatie wordt gewoon op een plan afgebeeld. Dit is uiteraard ideaal om het localisatiesysteem zelf te testen, maar het wil niet zeggen dat het systeem niet even makkelijk kan gebruikt worden in bijvoorbeeld context-aware applicaties waarbij de werking van de applicatie zal afhangen van welke kamer iemand zich in bevindt. De localisatie kan ingevoegd worden in elk programma dat draait op toestellen die Wireless LAN ondersteunen. Alle screenshots die in dit hoofdstuk getoond worden zijn gemaakt met de Pocket PC 2002 Windows CE emulator versie 4.1.0 (Build 14).

De interface die gebruikt wordt in de voorbeeldapplicatie is simpel gehouden. Op de screenshot in figuur 7.1 zijn de hoofdonderdelen zichtbaar.

De titelbalk bovenaan en het icoontje om het schermtoetsenbord te activeren rechtsonderaan zijn standardelementen van de Pocket PC applicaties. Deze behoren dus niet specifiek tot de interface van de locatie tracker. Verspreid over het hele scherm zien we een plattegrond van het gebouw waarin alle tests zijn uitgevoerd. Dit is een gewone figuur en kan heel makkelijk aangepast worden met welk plan van welk gebouw dan ook. Aangezien het systeem met niet gedefinieerde units werkt kan je dus met elke plattegrond op welke schaal dan ook werken zolang er maar dezelfde figuur gebruikt wordt bij het gebruiken van de applicatie als bij het opstellen van de trainingsdata. Er kan op eenvoudige manier doorheen het kaartje gepand worden. Door de vier knoppen in het midden van de zijkanten van het scherm met de letters U, D, L en R erin (komende van respectievelijk 'Up', 'Down', 'Left' en 'Right') kan men de focus van de kaart verplaatsen.



Figuur 7.1: Screenshot van de Location tracker interface

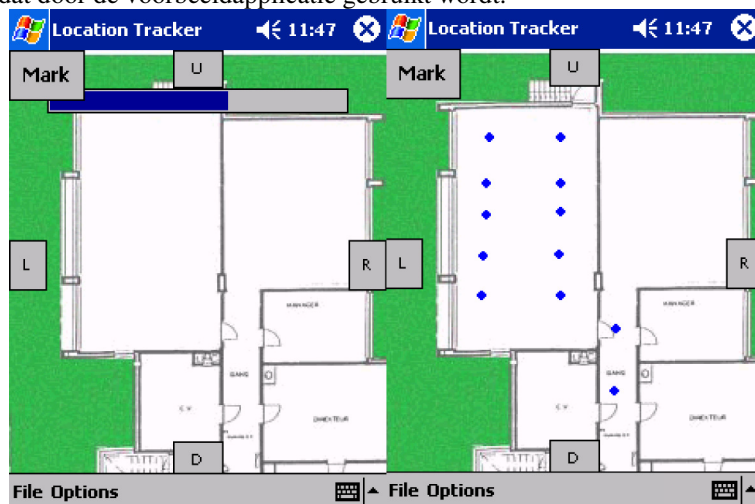
De knop met als label 'Mark' is toegevoegd om het testen te vergemakkelijken. Wanneer deze wordt ingedrukt wordt er in alle logbestanden die bijgehouden worden een markering aangebracht. Dit is een horizontale lijn in de tekstuele logbestanden. Op die manier kunnen belangrijke events makkelijk teruggevonden worden wanneer later de logbestanden geanalyseerd worden. Om een voorbeeld te geven waarin dit van pas kan komen kijken we naar één van de uitgevoerde testen. Deze bestond erin om al wandelend een parcours af te leggen dat langsheen alle fingerprintlocaties ging om dan later te checken hoe vaak het systeem een correcte afschatting gemaakt had. Uiteraard is het onmogelijk voor het systeem om de exacte locatie te kennen om deze te vergelijken met de schatting. Daarom werd er tijdens de test telkens wanneer een nieuwe fingerprintlocatie bereikt werd een markering aangebracht in de logfiles. Op die manier kon snel nagegaan worden hoe vaak in de periode tussen twee markeringen de correcte locatie teruggevonden kon worden.

We zullen nu een overzicht geven van de commandos in de menubalk. Ten eerste is er een functie toegevoegd om databaseinformatie op te slaan en terug in te laden. Het spreekt voor zich dat dit voor heel wat tijds winst zal zorgen want er moet niet steeds opnieuw fingerprints gaan toegevoegd worden. Dit is een vrij tijdverslindende bezigheid. Er moet namelijk op elk punt in de database meerdere samples genomen worden. Vijftig samples is vooral bij de tweede techniek – deze die werkt met de probabiliteit op basis van de signaalssterkte histogrammen – een absoluut minimum. Aangezien elke sample net iets langer dan één seconde duurt, neemt dit dus al gauw één

minuut per fingerprint in. Wanneer er dan nog meerdere fingerprints genomen moeten worden spreekt dit voor zich dat zo'n save en laadfunctie zelfs in een testapplicatie onontbeerlijk zijn. Er is ook een functie toegevoegd die een tekstveld op het scherm zichtbaar maakt om vervolgens tekstuele commandos in te geven. Dit is uiteraard ook een handigheid gebleken tijdens het testen omdat op die manier niet voor elke testfunctie dialoogvensters opgesteld moeten worden voor operaties waarvoor meerdere variabelen ingegeven moeten worden.

Er is ook een functie in het leven geroepen om te veranderen van verdieping. Aangezien er in de database van fingerprints locaties worden bijgehouden in drie dimensies was zo'n functionaliteit noodzakelijk. Om dit toe te voegen in de applicatie moeten er uiteraard van alle verdiepingen plattegronden aanwezig zijn die ook allemaal dezelfde oorsprong en schaal hebben. Er kan met andere woorden niet gewerkt worden met andere assenstelsels per verdieping. Tenslotte is er uiteraard ook een functie om het tracken van de positie in te schakelen. Bij het opstellen van de database of wanneer er nog geen database is ingeladen is het uiteraard nutteloos om deze functie te gaan gebruiken dus kan deze in- en uitgeschakeld worden.

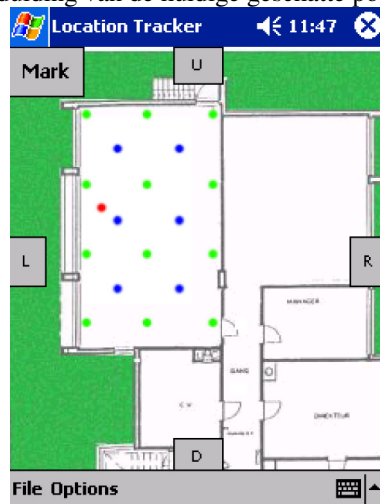
De applicatie is naast het tracken ook bedoeld voor het toevoegen van fingerprints in de database en dus bij uitbreiding ook voor het opstellen van een volledige fingerprint database om ze dan te gaan opslaan voor later gebruik. Het kan dus perfect gebruikt worden om databases op te stellen van andere applicaties. Wanneer dit gebeurt moet uiteraard wel opgelet worden dat dezelfde waarde wordt gegeven aan de units. Het opstellen van een database gebeurt als volgt. De gebruiker kan gewoon ergens op het kaartje drukken met de stylus van de pda om een fingerprint te gaan toevoegen. Uiteraard is het de bedoeling dat er gedrukt wordt op de plaats op het kaartje die overeenkomt met de plaats waar de gebruiker zich op dat moment bevindt. Wanneer dit gebeurt zal er een progressiebalkje verschijnen die aangeeft hoe lang het nog zal duren eer alle samples genomen zijn voor de fingerprint zoals te zien is op de screenshot in figuur 7.2 links. Hoeveel samples dit zijn wordt ingegeven via het configuratiebestand dat door de voorbeeldapplicatie gebruikt wordt.



Figuur 7.2: Dubbel screenshots – l:Progressie bij het ingeven van een fingerprint, r:De weergave van een heleboel reeds ingegeven fingerprints

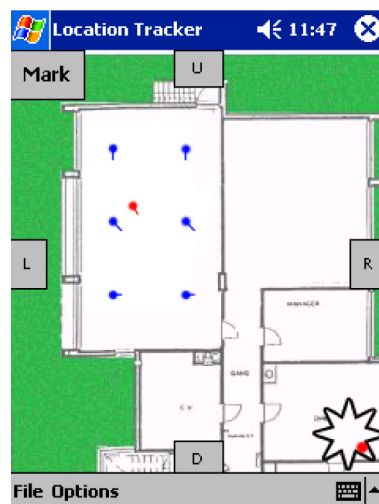
Rechts in figuur 7.2 zien we een screenshot na het ingeven van een aantal fingerprints. In het blauw worden de punten van deze fingerprints aangegeven. Wanneer er met twee grids gewerkt wordt zullen er ook groene

aanduidingen gebruikt worden om de fingerprintlocaties van de tweede grid aan te geven. Tijdens het tracken zullen er ook een aantal aanduidingen zijn om de huidige locatie weer te geven die geschat wordt door het systeem. De uiteindelijke locatie wordt in het rood weergegeven. In figuur 7.3 wordt er het systeem in werking getoond met twee grids en met de aanduiding van de huidige geschatte positie.



Figuur 7.3: Screenshot van systeem met twee grids

Tenslotte moesten er ook nog een aantal functionaliteiten toegevoegd worden om de richting te kunnen aangeven. Om dit makkelijk te kunnen aanduiden is er gekozen om in de rechteronderhoek van het beeld een windroos te plaatsen. Hierop kan de gebruiker dan zijn huidige richting aangeven uitgedrukt in één van de acht gebruikte richtingen. Wanneer er dan een punt zal toegevoegd worden zal deze richtingsinformatie meegegeven worden bij het opstellen van de database. Door middel van een lijntje in de richting die er is meegegeven zal dit ook aangeduid worden op kaart met de aanduidingen van de fingerprintlocaties. Op een zelfde manier zal dit ook aangegeven worden op de marker die aangeeft welke positie op dit moment afgeschat is door het systeem. Een screenshot van dit systeem is te zien in figuur 7.4.



Figuur 7.4: Screenshot van systeem met richtingsinformatie

Deze applicatie is ontwikkeld, net als de rest van het systeem trouwens, in C#. De basiselementen zoals de

knoppen en het menu zijn dan ook standaard C# widgets. Om de markeringen aan te brengen is er anders te werk gegaan. Aangezien er at runtime geen C# widgets aangemaakt kunnen worden, konden er geen labels bijvoorbeeld gebruikt worden voor deze markeringen. Daarom is er een PictureBox afgeleid tot een aangepaste klasse met een geherimplementeerde OnPaint functie. Met behulp van een hulpklasse OnMapElement was het mogelijk om dit soort markeringen op de gewenste positie te gaan tekenen. Ook de windroos waarop de richting ingegeven moet worden wordt getekend in deze OnPaint functie.

Zoals vermeld werd er in de voorbeeldapplicatie ook gebruik gemaakt van een configuratiebestand. Dit is een normaal tekstbestand dat begint met een getal dat aangeeft hoeveel lijnen er beschouwd moeten worden als configuratie informatie. Er is dus een mogelijkheid om achteraan het bestand commentaar toe te voegen aangezien enkel het aantal regels gecheckt wordt dat door het getal aangegeven wordt. Om een variabele die een getal voorstelt in te stellen moet er een commando dat eruit ziet als 'variabelenaam=15' ingegeven worden. Om booleaanse waarden in te geven gebruikt men dezelfde methode maar het getal wordt dan vervangen door de woorden 'true' of 'false'. Elk ander woord dan deze twee zal trouwens als 'false' beschouwd worden.

8 Conclusie

Alles in beschouwing genomen kunnen we stellen dat voor heel wat applicaties het localiseren met Wireless LAN perfect bruikbaar is. Heel wat systemen behalen nauwkeurigheden tot drie meter. Men kan dus niet alleen gaan kijken in welke kamer iemand zich bevindt maar ook nog grof geschat waar in die kamer.

Er zijn met het systeem een heel aantal testen uitgevoerd. Wat eerst opvalt is dat de fingerprintingtechnieken een heel verschil in nauwkeurigheid vertonen al naargelang men zich op een databasepositie bevindt of er tussenin. Wanneer men zich op een databaselocatie bevindt zal de techniek die enkel kijkt naar de meest overeenkomende match de beste resultaten opleveren. We zijn zelfs tot een nauwkeurigheid van twee meter gekomen. De interpolatietechnieken kwamen niet verder dan 3,5 meter in deze gevallen. Maar bij de tussenliggende posities was er nog een terugval in nauwkeurigheid te merken. De beste match techniek is bij deze niet echt interessant, maar bij de belangrijkere interpolatietechniek werd slechts een nauwkeurigheid van 6,5 meter vastgesteld.

Verder is het uiteraard ook van belang hoe ver de punten uit mekaar liggen en hoeveel samples er in de database worden opgenomen. De vorige tests werden steeds uitgevoerd met punten met tussenafstanden van om en bij de tien meter. Steeds werden er zeventig samples gebruikt per locatie. In een andere test hebben we in dezelfde omstandigheden getest maar dan wel met databasepunten die slechts vijf meter uit mekaar liggen en telkens slechts dertig samples. Merk op dat hierdoor de tijd om de database samen te stellen ongeveer gelijk blijft waardoor deze gevallen mooi afgewogen kunnen worden op de nauwkeurigheid. Bovendien is het ook zo dat de tussenliggende posities steeds ongeveer binnen de 2,5 meter van databasepunten zullen liggen wat uiteraard voordeliger is. Bij dit systeem kwamen we tot een nauwkeurigheid van 3,5 meter met de interpolatietechniek en 3 meter bij de beste match. De beste match is dus slechter maar aangezien deze enkel goede resultaten geeft wanneer men zich op databasepunten bevindt is het resultaat van de interpolatietechniek interessanter. De behaalde 3,5 meter geldt voor alle posities, dus ook de tussenliggende posities. Dit is uiteraard een stuk beter dan de 6,5 meter van de vorige database. We kunnen dus stellen dat een dichter net van databasepunten voordeliger is zelfs wanneer er minder samples per punt genomen worden.

Om de robuustheid van het systeem uit te testen hebben we de test met deze database nog eens herhaald maar dan met één belangrijk access point uitgeschakeld. Het spreekt voor zich dat de nauwkeurigheid te lijden zal hebben onder deze omstandigheden maar het is wel zo dat het systeem nog altijd blijft functioneren zoals we zometeen verder bespreken. We bekwamen in deze omstandigheden een gemiddelde fout van 6 à 6,5 meter dus slechts de helft van de nauwkeurigheid van het vorige systeem.

Tenslotte hebben we de vorige twee databases ook gecombineerd in een dubbel grid. Met de eerste database werd de beste match gezocht om dan vervolgens de interpolatie te gaan zoeken tussen de punten uit de tweede database die het dichtst lagen bij de beste match uit de eerste database. We kwamen opnieuw tot een nauwkeurigheid van ongeveer 3,5 meter. Op het eerste zicht biedt deze techniek dus niet erg veel winst.

We kunnen dus constateren dat een nauwkeurigheid van ongeveer drie meter zeker haalbaar moet zijn in de juiste omstandigheden. Hierbij spelen drie factoren een belangrijke rol. Ten eerste is er de opstelling van de access points. Deze moeten elkaar zeker overlappen zodat er op elk punt meerdere access points gevonden kunnen worden door het systeem. Vervolgens speelt ook de database een grote rol. Het nauwkeurig uitkiezen van de punten die zullen opgenomen worden in de database en de dichtheid van de grid die ze gaan vormen is een must. Tenslotte speelt ook het aantal samples waarmee de database rekening zal houden een niet te onderschatten rol. Vooral bij de signaalsterkte histogrammen moet men rekenen op minstens een dertigtal samples. Anders wordt de histogram onmogelijk representatief. Liefst zou men zelfs het dubbel of nog meer samples nemen. Dit hangt uiteraard ook af van het gebied dat zal overkoepeld moeten worden met het systeem aangezien het opstellen van de database een werk van lange adem is.

Alhoewel heel wat applicaties wel handig gebruik zouden kunnen maken van deze localisatiemethodes wordt het vaak nog niet toegepast omdat er geen algemene pakketten bestaan die dit toelaten. Er zijn wel commerciële pakketten verkrijgbaar maar dan moeten vaak zowel tracking toestellen als access points bij dezelfde fabrikant aangekocht worden om het systeem te kunnen laten functioneren. In het systeem dat ontwikkeld is naar aanleiding van deze thesis is een pakket samengesteld dat onafhankelijk werkt en met relatief weinig code kan toegevoegd worden. Eens de trainingsfase afgelopen is kan het zonder veel moeite draaiende gehouden worden tenzij er verschuivingen optreden in al te grote obstakels of in gebruikte access points zelf.

Dit brengt ons bij de robuustheid van het systeem. Op zich zal het systeem tijdens de trainingsfase op zoek gaan naar alle access points waarvan er beacons opgevangen kunnen worden. Maar het is best denkbaar dat van één of meerdere access points niet gegarandeerd kan worden dat ze steeds op dezelfde plaats blijven staan. In dat geval kan er een lijst van mac-adressen meegegeven worden aan het systeem die net wel of net niet in beschouwing genomen moeten worden bij het localiseren. Wanneer een gebruikt access point zou uitvallen is dit geen probleem. Het systeem zal gewoon verder blijven functioneren, alleen zal de nauwkeurigheid misschien verlagen als er daardoor een zone ontstaat waar er te weinig access points aanwezig zijn.

Kortom we kunnen stellen dat voor systemen die geen nauwkeurigheid kleiner dan *** meter (* ik wil eerst de laatste tests afwachten vooraleer ik er een definitief getal op plak *) vereisen het systeem een geschikte tool is om localisatie toe te voegen aan een applicatie. Men moet zich er natuurlijk wel van bewust zijn dat het systeem enige tijd nodig heeft om een meting uit te voeren en te analyseren. Om enige ademruimte te verlenen aan de applicaties die constant willen tracken is het misschien noodzakelijk – tenminste wanneer uitgebreide en tijdrovende analyse technieken gebruikt worden – om de analyse te laten gebeuren op een server. Men zal dan de signaalgegevens gaan doorsturen via WLAN naar een server die de analyse gaat uitvoeren. In dit soort systemen hoeven de fingerprintdatabases ook niet op alle toestellen aanwezig te zijn. Eens de server de analyse heeft

uitgevoerd kan het resultaat teruggeven worden. Het spreekt voor zich dat in dergelijke systemen op de veel krachtigere servers ook veel uitgebreidere analysetechnieken toegepast kunnen worden. Maar in het geïmplementeerde systeem hebben we de voorkeur gegeven aan de technieken die op de devices zelf uitgevoerd kunnen worden.

9 Index van tabellen

Tabel 1: Factoren die Loran C negatief beïnvloeden (naar tabel 1207 uit pollux.nss.nga.mil/NAV_PUBS/APN/Chapt-12.pdf).....	10
Tabel 2: Galileo services (naar tabel pagina 4 uit http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galilei_brochure.pdf).....	14

10 Index van figuren

Figuur 2.1: Snijpunt van twee hyperbolen met brandpunten A-B en A-C.....	9
Figuur 2.2: Positieberekening mbv. satellieten.....	11
Figuur 2.3: GPS satelliet banen.....	13
Figuur 2.4: IrDA toestel dat kamer A nadert.....	18
Figuur 3.1: Architectuur voorgesteld in figuur 1 van [33].....	29
Figuur 5.1: Concept diagram localisatiesysteem.....	35
Figuur 5.2: Percentage van de afwijkingen groter dan x aantal dB in functie van de gebruikte techniek.....	37
Figuur 5.3: Vergelijking van afwijkingen in signaalsterkte voor en na de optimalisaties.....	40
Figuur 5.4: Zes fingerprints van de test uitgezet op een grondplan.....	40
Figuur 5.5: Gevonden posities vergeleken met de echte posities in functie van de tijd.....	41
Figuur 5.6: Histogram van gemeten signaalsterktes van één access point.....	42
Figuur 5.7: Resultaten van de basisversie.....	43
Figuur 5.8: Screenshot hulpprogramma met de histogrammen van het eerste acces point.....	44
Figuur 5.9: Screenshot hulpprogramma met de histogrammen van het tweede acces point.....	45
Figuur 5.10: Resultaten met de eerste optimalisatie.....	46
Figuur 5.11: Resultaten met alle optimalisaties en alle access points.....	46
Figuur 5.12: Een eerste afstandsfunctie.....	48
Figuur 5.13: Een tweede afstandsfunctie.....	48
Figuur 5.14: Beperkt overzicht van de klassen van het systeem.....	49
Figuur 6.1: Punten veraf van de huidige positie moeten een lagere probabiliteit krijgen.....	54
Figuur 6.2: Het pad dat een gebruiker aflegt en de voorspelling van de huidige richting.....	55
Figuur 6.3: De blauwe fingerprints kunnen op de zwarte positie evenzeer naar de rode als naar de groene afwijken wanneer vier fingerprints beschouwd worden.....	56
Figuur 6.4: Een gebouw waarvan we de informatie kunnen gebruiken bij de beslissingsfase.	58
Figuur 7.1: Screenshot van de Location tracker interface.....	61
Figuur 7.2: Dubbel screenshots – l:Progressie bij het ingeven van een fingerprint, r:De weergave van een heleboel reeds ingegeven fingerprints.....	62
Figuur 7.3: Screenshot van systeem met twee grids.....	63
Figuur 7.4: Screenshot van systeem met richtingsinformatie.....	63

11 Bibliografie

- [1] <http://www.boats.com/content/boat-articles.jsp?contentid=2984>
- [2] http://pollux.nss.nga.mil/NAV_PUBS/APN/Chapt-12.pdf
- [3] <http://tf.nist.gov/general/enc-c.htm>
- [4] <http://zone.ni.com/devzone/nidzgloss.nsf/10ce50b61f969f5a86256879000d3a01/d7defb18c3fb84878625689a0056f5b7?OpenDocument>
- [5] <http://www.jproc.ca/hyperbolic/omega.html>
- [6] http://www.daviddarling.info/encyclopedia/T/Transit_satellite.html
- [7] <http://www.nasm.si.edu/gps/before.html>
- [8] <http://www.fas.org/spp/military/program/nav/transit.htm>
- [9] <http://www.go-gps.nl>
- [10] <http://electronics.howstuffworks.com>
- [11] http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html
- [12] http://www.lordpercy.com/gps_explained.htm
- [13] http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galilei_brochure.pdf
- [14] <http://www.globalsecurity.org/space/world/china/beidou.htm>
- [15] http://www.gisdevelopment.net/news/viewn.asp?id=GIS:N_pbgismxz
- [16] <http://www.spacedaily.com/news/china-01b.html>
- [17] http://www.spaceandtech.com/spacedata/constellations/glonass_consum.shtml
- [18] <http://www.thetrumpet.com/index.php?page=article&id=1962>
- [19] http://www.agcc.gov.au/gnss/GNSS_Platforms.aspx
- [20] <http://www.tssg.org/people/tpfeifer/publish/kivs2003final-intern.pdf>
- [21] <http://www.sonear.com/docs/SIMLBSSONEAR.pdf>
- [22] <http://www.utlib.ee/ekollekt/diss/mag/2005/b17418136/otsason.pdf>
- [23] http://www.cs.ut.ee/~veljoo/works/GSMindoor_ubicomp05.pdf
- [24] <http://opennetcf.org>
- [25] <http://www.wherenet.com/>
- [26] http://www.hitachi.co.th/cat_index_217.shtml
- [27] <http://www.bvsystems.com/Products/WLAN/Beetle/beetle.htm>
- [28] http://www.geocities.com/frank_wiedmann/sixport.html
- [29] <http://cricket.csail.mit.edu/>
- [30] <http://www.cl.cam.ac.uk/Research/DTG/attarchive/bat/>
- [31] Nguyen, D. "Where Am I? An Evaluation of Localization for Wireless Ethernet". 3 mei, 2004
- [32] Carmine Ciavarella, Fabio Paternò: Design Criteria for Location-Aware, Indoor, PDA Applications. Mobile HCI 2003: 131-144
- [33] Pfeiffer, T. Redundancy vs. Imperfect Positioning for Context-dependent Services. 2004. Workshop on Advanced Context Modelling, Reasoning and Management, in conjunction with Ubicomp, 6th International ACM Conf. on Ubiquitous Computing, Nottingham, England
- [34] Golmie, Nada, and Frederic Mouveaux. Interference in the 2.4 GHz ISM Band: Impact on the Bluetooth Access Control Performance. IEEE International Conference on Communications, 8:2540-2545, 2001
- [35] Jeffrey Wojtiuk. Bluetooth and WiFi integration: Solving co-existence challenges.

- PRIMEDIA, Oktober 2004
- [36] Andreas Butz. Between location awareness and aware locations: where to put the intelligence. *Applied Artificial Intelligence*, Volume 18, Nummer 6, Juli 2004, pp. 501-512(12)
 - [37] Josef Hallber, Marcus Nilsson. Positioning with Bluetooth, IrDA and RFID. Master Thesis, Luleå, University of Technology, 2002
 - [38] Antti Seppänen, Jouni Ikonen, Jari Porras. Extracting and Using Position Information in WLAN Networks. Lappeenranta University of Technology, Finland, 2003
 - [39] Zhe Xiang, Song Song, Jin Chen, Hao Wang, Jian Huang, Xingxin Gao. A Wireless LAN-based Indoor Positioning Technology. *IBM Journal of Research and Development*, Volume 48, No. 5/6, 2004
 - [40] Paul Hii, Arkady Zaslavsky. Improving Location Accuracy by Combining WLAN Positioning and Sensor Technology. Monash University, AU
 - [41] Kay Eckert. Overview of Wireless LAN based Indoor Positioning Systems. Mobile Business Seminar, 14 december, 2005
 - [42] David Kernévès, Bernard Huyart, Xavier Begaud, Eric Bergeault, Louis Jallet. Direction Finding with Six-Port Reflectometer Array. 8th COST 260 Management Committee Meeting in Rennes, France, 1-3 Oktober 2000
 - [43] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, Eyal de Lara. Accurate GSM Indoor Localization. Ubicomp2005: Tokyo, Japan, 11-14 September, 2005.
 - [44] http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galilei_brochure.pdf

Appendix: Reference

Deze appendix geeft een reference van de klassen van het systeem die gebruikt zullen worden wanneer men het gaat toevoegen aan een applicatie. De klassen die enkel gebruikt worden door het systeem zijn hier niet weergegeven.

- public abstract class Localizer
 - Virtuele localisatiebepaling waarop uitbreidingen kunnen geschreven worden
 - public Localizer()
Constructor
 - public abstract void Initialise()
 - public static double fingerPrintEqualityTreshold
drempelwaarde die instelt hoeveel fingerprints best zouden verschillen van mekaar. Default: 0.031
 - public static double fingerPrintDistanceTreshold
drempelwaarde die instelt hoeveel units fingerprints van mekaar verwijderd moeten liggen. Default: 35
 - public abstract Location getLocation()
De localizer laten uitvoeren om tot een locatie te komen
 - public abstract Location getLocation(FingerPrint fp)
De localizer laten uitvoeren om tot een locatie te komen, de parameter is een uitgevoerde meting.
Door de meting niet in deze functie te laten uitvoeren kan dezelfde meting in meerdere localizers gebruikt worden.
 - public abstract void loadDB(string dbName)
Databaseinformatie inladen met bestandsnaam als parameter
 - public abstract void saveDB(string dbName)
Databaseinformatie opslaan met bestandsnaam als parameter
 - public void addNormalOptimizer(Optimizer opt)
Een standaard Optimizer toevoegen
 - public void addDecisionOptimizer(DecisionOptimizer opt)
Een DecisionOptimizer, die gebruikt moet worden door de localizer tijdens de beslissingsfase, toevoegen.
 - public void addPostDecisionOptimizer(PostDecisionOptimizer opt)
Een PostDecisionOptimizer, die gebruikt moet worden door de localizer na de beslissingsfase, toevoegen.
- public abstract class FPLocalizer: public Localizer
 - Virtuele localisatiebepaling dat werkt met een fingerprintingdatabase
 - public abstract Location getInterpolationLocation()

- De localizer laten uitvoeren om tot een locatie te komen via interpolatie
- `public abstract Location getInterpolationLocation(FingerPrint fp)`
De localizer laten uitvoeren om tot een locatie te komen via interpolatie, de parameter is een uitgevoerde meting. Door de meting niet in deze functie te laten uitvoeren kan dezelfde meting in meerdere localizers gebruikt worden.
 - `public abstract Location getInterpolationNotToFarFrom(FingerPrint fp, Location nearLoc)`
Gelijkaardig aan de vorige functie maar enkel de access points nabij de parameter nearLoc beschouwend.
 - `public abstract bool addLocation(Location loc)`
- `public class LocPrLocalizer: public FPLocalizer`
Localisatiebepaling op basis van de probabiliteit van een meting
- `public FPLocalizer ()`
Constructor
 - `public static double exponentChecks`
Om slechte matches minder belangrijk te maken wordt een machtsverheffing uitgevoerd met deze exponent. Hoe hoger de exponent, hoe minder belangrijk de slechte matches. Default: 8
 - `public static double compTreshold`
De drempelwaarde waarop locaties niet meer in beschouwing worden genomen. Default: 2
 - `public static uint checkBestX`
Een locatie zal wanneer er een interpolatie gevraagd wordt uitgaan van dit aantal best matchende waardes. Default: 5
 - `public override void Initialise()`
 - `public abstract Location getLocation()`
 - `public abstract Location getLocation(FingerPrint fp)`
 - `public override Location getInterpolationLocation()`
 - `public override Location getInterpolationLocation(FingerPrint fp)`
 - `public override Location GetBestMatchNotToFarFrom(FingerPrint fp, Location nearLoc)`
 - `public override void loadDB(string dbName)`
 - `public override void saveDB(string dbName)`
 - `public override bool addLocation(Location loc)`
- `public class StrPrLocalizer: public FPLocalizer`
Localisatiebepaling op basis van de probabiliteit van een Signaalssterkte
- `public SPLocalizer()`
Constructor
 - `public override void Initialise()`
 - `public override Location getLocation()`
 - `public override Location getLocation(FingerPrint fp)`

- public override void loadDB(string dbName)
- public override void saveDB(string dbName)
- public override Location getInterpolationLocation()
- public override Location getInterpolationLocation(FingerPrint fp)
- public override Location getInterpolationNotToFarFrom(FingerPrint fp, Location nearLoc)
- public override bool addLocation(Location loc)

- public class DoubleGridLocalizer: public Localizer
 - Localisatiesysteem dat werkt met een normaal grid en een dichter grid om in de tweede te interpoleren nabij de positie verkregen uit de eerste grid.
 - public DoubleGridLocalizer(FPLocalizer firstGrid, FPLocalizer secondGrid)
 - Constructor
 - public override void Initialise()
 - public override Location getLocation()
 - public override Location getLocation(FingerPrint fp)
 - public override void loadDB(string dbName)
 - public override void saveDB(string dbName)

- public class Optimizer
 - Klasse die dienst doet als standaard optimalisatie. Deze kunnen in elk soort localizer gebruikt worden.
 - public double getLocationScore(Location loc)
 - Deze functie geeft een score terug die uitdrukt hoe groot de mogelijkheid is dat de meegegeven locatie correct is.
 - Public void addLatestLocation(Location loc)
 - De laatst geschatte locatie toevoegen om ze later te gebruiken in de optimalisatie.

- public class DecisionOptimizer: public Optimizer
 - Abstracte klasse die gebruikt kan worden om een lijst van locaties met probabiliteit te verbeteren tijdens de beslissingsfase
 - public abstract void refineProbabilityList(LocationProbability [] list)

- public class PredictionOptimizer: public DecisionOptimizer
 - Prediction optimalisatie
 - public override void refineProbabilityList(LocationProbability [] list)

- public class PostDecisionOptimizer: public Optimizer
 - Abstracte klasse die gebruikt kan worden om een geschatte locatie te verbeteren na de beslissingsfase

- public abstract void refineLocation(ref Location loc)
- public class HistoryOptimizer: public PostDecisionOptimizer
 - History-based optimizer
 - public override void refineLocation(ref Location loc)
- public class PlanOptimizer: public DecisionOptimizer, PostDecisionOptimizer
 - Optimalisatie op basis van omgevingsgegevens
 - public override void refineProbabilityList(LocationProbability [] list)
 - public override void refineLocation(ref Location loc)
 - public void loadPlanInfo(System.IO.FileStream file)
 - De gegevens van het plan inladen uit een file.
- public class Location
 - public Location()
 - Constructor
 - public Location(double x, double y, double z, angle a)
 - public Location(double x, double y, double z)
 - public Location(double x, double y)
 - public Location(System.IO.FileStream file)
 - public void save(System.IO.FileStream file)
 - public override string ToString()
 - public static bool operator== (Location loc1, Location loc2)
 - public static bool operator!= (Location loc1, Location loc2)
 - public override bool Equals(object obj)
 - public override int GetHashCode()
 - public double dist(Location loc)
 - public string getLogStr()
- public struct LocationProbability
 - public Location mLocation
 - public double mProbability
 - public uint mSize
 - Uit hoeveel afzonderlijke probalities dit object is opgebouwd (enkel van toepassing bij het mergen).
 - public LocationProbability(Location location, double probability, uint size)
 - Constructor
 - public void multProb(double factor)
 - De probabiliteit vermenivuldigen met een meegegeven factor.
 - public static LocationProbability[] merge(LocationProbability[] list1, LocationProbability[] list2)

Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen en uw akkoord te verlenen.

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

Hybrid indoor/outdoor location determination technologies

Richting: **Master in de informatica**

Jaar: **2006**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Deze toekenning van het auteursrecht aan de Universiteit Hasselt houdt in dat ik/wij als auteur de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij kan reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

U bevestigt dat de eindverhandeling uw origineel werk is, en dat u het recht heeft om de rechten te verlenen die in deze overeenkomst worden beschreven. U verklaart tevens dat de eindverhandeling, naar uw weten, het auteursrecht van anderen niet overtreedt.

U verklaart tevens dat u voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen hebt verkregen zodat u deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal u als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze licentie

Ik ga akkoord,

Tim TUTENEL

Datum: