Feasibility of Collaborative Indoor Positioning Using Smartphones

Jaakko Vallinoja

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 30.9.2016

Thesis supervisor:

Prof. Jari Saramäki

Thesis advisors:

Prof. Simo Särkkä

D.Sc. (Tech.) Arno Solin

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Jaakko Vallinoja

Title: Feasibility of Collaborative Indoor Positioning Using Smartphones

Date: 30.9.2016 Language: English Number of pages: 6+55

Department of Computer Science

Professorship: Computational Science

Supervisor: Prof. Jari Saramäki

Advisors: Prof. Simo Särkkä, D.Sc. (Tech.) Arno Solin

A reliable indoor positioning service for smartphones is a service that is often requested. There are several competing technologies already available but a lot of basic research is still done on the subject. This thesis studies the applicability and technological possibilities of improving the performance of a positioning service using peer to peer collaboration.

The Bluetooth low energy technology (BLE) offers a possibility to use peer to peer radio signal measurements with smartphones. This could be used to improve the performance of existing positioning algorithms if enough service users are in close proximity to each other.

In this thesis a pedestrian simulation system was implemented to study the probability that two positioning service users are in close enough proximity to each other for BLE usage. The suitability of BLE as the collaboration technology was studied by implementing a particle filter based positioning system that uses BLE measurements to track a smartphone. Finally the collaborative BLE system was integrated on top of an existing geomagnetic tracking algorithm and the effect on the positioning performance was studied.

It was concluded that the BLE as a technology is suitable for positioning use despite the large measurement uncertainty. BLE based collaboration is feasible in improving the positioning results provided that the basic positioning technology is reliable enough. The pedestrian simulations concluded that with realistic expected number of users in one building most sessions would not benefit from collaboration but it would still likely happen frequently.

Keywords: Indoor positioning, IPS, Particle filter, Bluetooth, BLE, Beacon, Collaboration, Cooperation

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Jaakko Vallinoja

Työn nimi: Laitteiden välisen yhteistyön soveltuvuus älypuhelimilla toteutettavaan sisätilapaikannukseen

Päivämäärä: 30.9.2016 Kieli: Englanti Sivumäärä: 6+55

Tietotekniikan laitos

Professuuri: Laskennallinen tiede

Työn valvoja: Prof. Jari Saramäki

Työn ohjaajat: Prof. Simo Särkkä, TkT. Arno Solin

Luotettava sisätilapaikannuspalvelu on haluttu ominaisuus mobiilipalveluiden kehityksessä. Useita kilpailevia ratkaisuja on jo markkinoilla, mutta ongelman parissa tehdään vielä huomattavan paljon perustutkimusta. Tässä diplomityössä tutkitaan mahdollisuutta parantaa paikannusjärjestelmän toimintaa käyttäen vertaisyhteistyötä.
Bluetooth low energy -teknologia (BLE) tarjoaa mahdollisuuden käyttää laitteiden välisiä radiosignaalimittauksia älypuhelimilla. Tätä voidaan mahdollisesti hyödyntää parantamaan olemassa olevien paikannusalgoritmien toimintaa, jos riittävästi käyttäjiä on riittävän lähellä toisiaan.
Tässä diplomityössä toteutettiin ihmisjoukkojen liikettä sisätiloissa mallintava järjestelmä, jolla tutkittiin todennäköisyyttä, että kaksi paikannusjärjestelmän käyttäjää olisi riittävän lähellä toisiaan käyttääkseen BLE-radiomittauksia. BLE:n soveltuvuutta paikannusteknologiana tutkittiin toteuttamalla partikkelisuotimeen perustuva paikannusjärjestelmä, joka käyttää BLE-mittauksia älypuhelimen seuraamiseen. Lopuksi BLE mittausjärjestelmä integroitiin olemassa olevaan magneettikenttään perustuvaan paikannusalgoritmiin ja BLE-yhteistyön vaikutusta algoritmin toimintaan tutkittiin.
Työ osoitti, että BLE on paikannuskäyttöön soveltuva teknologia suuresta mittausepävarmuudesta huolimatta. BLE-perusteinen yhteistyö paikannustuloksen parantamisessa on toimiva ratkaisu, mikäli varsinainen paikannusteknologia on riittävän luotettava. Realistisesti odotettavissa olevilla paikannuspalvelun käyttäjämäärillä BLE-yhteistyötä todennäköisesti tapahtuisi suhteellisen usein, vaikka suurin osa paikannussessioista ei pääsisikään hyötymään siitä.

Avainsanat: Sisätilapaikannus, IPS, Partikkelisuodin, Bluetooth, BLE, Beacon, Yhteistyö

Foreword

This thesis has been done with my employer IndoorAtlas. I want to thank all my colleagues at IndoorAtlas for this opportunity and all the support they gave me during this process. I also want to thank for the extensive creative freedom I was given in defining and realizing this project.

Especially I want to thank my instructors Arno Solin and Simo Särkkä for all the help in creating this work. The time spent rewriting large sections of the text after each new round of comments was invaluable. I also thank my colleague Toni Makkonen for implementing some of the software tools used for the experiments.

Finally I want to thank my friends at Teekkarispeksi and GrexMusicus for doing their best to provide distractions and prevent working.

Espoo, 26.9.2016

Jaakko Vallinoja

Contents

Operators and Abbreviations

Operators

$\sum_{i=a}^{b}$	Sum over index i from a to b
$\prod_{i=a}^{b}$	Product over index i from a to b
$\int_{a}^{b}$	Integral from a to b
$\|\mathbf{a}\|$	Euclidean norm of vector $\mathbf{a}$
$\nabla\phi$	Gradient of ϕ
$E(x)$	Expected value of x

Abbreviations

API	Application programming interface
BLE	Bluetooth low energy
EKF	Extended Kalman Filter
GPS	Global positioning system
IPS	Indoor positioning system
MAC	Media access control
OS	Operating system
PDR	Pedestrian dead reckoning
PF	Particle filter
RMSE	Root mean squared error
RSSI	Received signal strength indicator
SIR	Sequential importance resampling
SIS	Sequential importance sampling
URL	Uniform resource locator
UUID	Universally unique identifier
UWB	Ultra-wideband
WGS84	World geodesic system 1984
ZUPT	Zero Velocity Update

1 Introduction

In the modern world we are more and more interacting with technology and automatic systems. Life has also become more fast paced and it is more and more relevant that we know where we are and where we are heading. In the most literal meaning of the expression.

Satellite positioning systems such as GPS have long since provided a working solution to positioning and navigation in outdoor environments by offering an accurate real time location service. However satellite based systems are only available when multiple satellites are on sight. This is not true for indoor environments and sometimes not even in bigger cities between large buildings (urban canyons). While some attempts have been made to improve the performance of current satellite navigation systems in urban environment it can be concluded that there is a need for an alternative system. (Agarwal et al., 2002; Watson et al., 2005)

Studies show that urban people spend significantly more time indoors than outdoors (Schweizer et al., 2007). In fact we spend almost all our time indoors. While a positioning system is not necessarily needed at home or at work office, there are several indoor environments where one would bring much needed help in daily life. These include big malls, public transportation hubs, airports, museums, and exhibitions where one might want an automated audio tour. In general any big buildings where new people not familiar with the place regularly visit or where automated tracking of people would be desired might benefit from indoor positioning. The lack of reliable positioning system also restricts innovations in new digital services and products such as virtual or augmented reality.

Collaborative or *cooperative* positioning refers to a set of methods, technologies and algorithms in which the devices communicate and share information with each other while positioning and use this data to make the process faster, more accurate or more robust. Automatic device to device communication for performance improvement is an increasing trend on many industries and services. While the possibilities are somewhat limited on mobile phones due to security issues, these kind of technologies will probably become more and more commonplace in the future. Any data about the environment that can be shared could in theory be used in comprehensive positioning system. In practice however the possible useful information is usually very limited due to the platform and hardware limitations.

The most direct way to collaborate is to provide information about own position and some sort of measurement of spatial relationship between the devices. This kind of system has been used in many of experimental systems (e.g. Chan et al., 2006; Strömbäck et al., 2010). The spatial relationship can be estimated numerous ways ranging from radio detection to lighting, sound or image classification. However as the system is meant to be easy to use with smartphones the methods based on radio detection are the most suitable. The measurement can be a distance estimate, a direction estimate or a simple proximity indication. Another possible view on collaboration is sharing calibration information when the system already knows the devices are in close proximity to each other. For example the barometer or thermometer bias could in theory be estimated by comparing the values to other

nearby devices.

This thesis explores the possibilities and feasibility of positioning collaboration using modern smart phones with no additional hardware installed. The main research questions are how the amount of users in the same indoor venue affects the collaboration possibilities and what kind of technologies are available to improve other positioning systems using collaboration. Also the platform limitations currently set by the different operating systems are examined.

The first part of the thesis studies the number of required users by simulating pedestrians in a real environment and studying the probabilities that the pedestrians are in small enough range from each other to collaborate in a meaningful way. A pedestrian simulator is implemented in MATLAB[1] and simulations are run on a plausible real user environment.

The technology for collaboration proposed in this work is phone to phone Bluetooth low energy (BLE) advertisement transmissions. The quality of BLE radio signal strength based distance estimates is examined. A particle filter based BLE positioning system is implemented in MATLAB and tested using commercially available smartphones. Finally the lessons learned from BLE positioning tests are used to implement the BLE distance estimate system on top of an existing geomagnetic positioning algorithm and it's effect on positioning accuracy and robustness is tested.

[1]MATLAB Release 2015b, MathWorks Inc.

2 Background

All modern smart phone operation systems offer some kind of *platform location* system that estimates the location of the phone using the best available approach. If satellite navigation is not available it might use cell tower information and possibly information about known Wi-Fi access points. At least Google and Apple have their own automatically updating global Wi-Fi access point database. However this kind of approaches can usually only reliably achieve building level accuracy. Sometimes not even that (Zandbergen, 2009). Many applications would need a more accurate solution, one that would allow accurate positioning and navigation inside buildings. What makes the problem even more difficult is that the position error tolerance in indoor positioning can be much lower than outdoors. An error radius that is completely manageable in outdoor environment might cover several rooms in an indoor floor plan and if you want to for example show a customer the way to a product you really need accuracy in one meter range. A further complication arises from the fact that indoor environments are three-dimensional. Outdoors it is enough to get the position in the two-dimensional plane but most large indoor venues have multiple floors. Differentiating between floors can be complicated and makes convergence to correct location much harder.

There are several approaches to the indoor positioning problem. One of the most extensively researched approach is to make a map of the Wi-Fi access points inside a building and use that map to position devices. This approach works in principle but the accuracy is very dependent on how many access points have been installed and how their positions are distributed on the map (Zandbergen, 2009). It might be that Wi-Fi based solution is accurate on some parts of a building but very inaccurate on others. Even bigger a problem is that at least for now Apple's iOS operating system does not offer a public API for Wi-Fi data and thus does not allow third party applications an access to information about detected Wi-Fi networks. In the current market situation any commercial indoor positioning system or IPS must necessarily support the iOS platform, so a Wi-Fi only positioning system is not a good option. Apple's own solution is based on Bluetooth low energy beacons (see Chapter 3.2). These are radio beacons that can be used to estimate the position of the receiver. The problem is that their range is very short and to build a comprehensive and reliable positioning solution to a big building might require installing hundreds or even thousands of beacons.

Pedestrian dead reckoning (PDR) refers to positioning by measuring the steps and inertial track from a starting point. The steps can essentially be counted from accelometer data and gyro measurements, accelometer and compass can be used to estimate direction changes. This is used in many positioning systems but since it is entirely dependent on good measurements and accurate starting point a step based PDR always starts to drift on longer distances. This is why it is usually used to support other technologies. The drift can be reduced for example by frequent zero velocity updates (ZUPT) (Foxlin, 2005) to correct the drift but these are only feasible in foot mounted sensors that actually stop between steps. With smartphones an accurate ZUPT would basically require holding the phone against a stationary

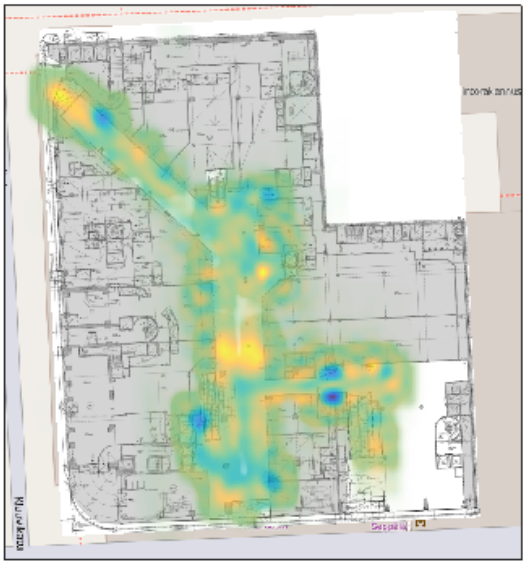


Figure 2.1: Mapped anomalies in the norm of the magnetic field vectors on an indoor map.

surface and this is not feasible in actual positioning systems. Dead reckoning is used extensively on wheeled devices such as automatic cleaning robots as wheel movement is much easier to measure accurately than steps but even these applications suffer from cumulative drifting (Chung et al., 2001; Hofner and Schmidt, 1994). There are fairly accurate PDR solutions (e.g. Fang et al., 2005) but none of the currently available systems work using only smart phone sensors.

A relatively novel approach is using the geomagnetic anomalies for positioning. It has been known for a long time that compasses are unreliable indoors and in urban areas in general. The Earth magnetic field interacts with steel and concrete structures in buildings and produce constant variations in the field around them creating an unique geomagnetic anomaly pattern or *fingerprint* in the building (Haverinen and Kemppainen, 2009). Most modern smartphones include an electronic compass chip. This chip is usually a general purpose magnetometer and can be used to measure the geomagnetic fields in three dimensions. If this anomaly field inside buildings is mapped a position of a device can be found by using the compass and fitting the measurements to the map to find the most likely path. An example of a magnetic map is shown in Figure 2.1.

The commercial motivation to offer an indoor positioning system might include improving the customer experience in some environment for example by directing him or her to the desired product. Globally the indoor market is worth hundreds of billions and if offering a location service can influence even a fraction of that it could be worth it. Another use case would be proximity marketing. A location service application could send directed location specific advertisements to the user

while directing him or her to the destination. Companies like Google already offer extensive map service outdoors and benefit from the data they can gather from users. Similar systems might be implemented indoors. Lastly there is the social networking aspect. When the applications know where the users are they can for example offer directed services to people in close proximity.

Slightly less commercial use cases include safety systems like tracking patients or children and coordination of first responders such as firefighters or ambulance crew in a building. Public transportation providers would also want to provide their customers with a convenient way to find the right platform or the correct exit in a large metro station or navigate them to the right gate in a big international airport. As an example case Virgin Atlantic Air ran a trial[2] where their application automatically loaded the boarding pass when the customer was approaching the gate.

All in all the goal that the people never get lost in any environment be it indoors or outdoors. A truly ubiquitous positioning system would seamlessly transfer from satellite based outdoor system to the best available IPS when the user walks trough an entrance. However we are not quite there yet. Technology is supposed to improve the life of the users and indoor location services have been one area where technology is clearly lacking.

Collaborative Positioning

A geomagnetic positioning system works well in tracking the movement of a device provided that the measurement quality is good. However it can have difficulties in finding the initial position quickly as it only gets information when the measuring device moves in the building. Convergence times averaging over 10 seconds and over 10 meters walked distance requirements have been reported (Solin et al., 2016). In a fast paced environment a user cannot wait 10 or 20 seconds for the system to converge to the right position. This initialization problem can be solved using Wi-Fi and beacon measurements if they are available. However with the Apple iOS operating system the Wi-Fi data is not available and beacons are only installed to few places. Magnetometer calibration issues might make the convergence slower and sometimes the magnetometer measurements might have anomalies not related to the building structures. These can result for example from power lines or large moving structures such as trains. Furthermore sometimes the magnetic map might not have strong enough features for reliable positioning.

One at least partial solution to these problems might be a collaborative positioning system. If there is a smartphone using the Android platform that can get the Wi-Fi location and fix the position easily, it could then help the phones using the iOS platform that are in close proximity. Also if one phone gets bad magnetic measurements, loses the location and cannot find itself back on the right track, its position could be fixed by help from other devices around it. Even in the situations where the magnetic field is unusable due to for example moving trains a PDR

[2] `https://blog.virgin-atlantic.com/t5/Our-Future/Virgin-Atlantic-lights-the-way-with-Apple-s-iBeacon-technology/ba-p/26359#.VabN6WB4teM`, accessed 7.6.2016

enhanced by collaboration with other devices might keep the system on track until the magnetic field is again usable.

In the context of indoor location services the duration of connections between devices would usually be short as a typical use case session is not very long. Thus the information shared between the devices must be something that can be helpful even if one particular device is only connected for a short while. For easy and streamlined user experience the system has to be automatic. In other words, it must not require user input or even knowledge. After all the GPS would not be very convenient if the user had to separately accept the transmission from each satellite. The devices should simply share some information on the background to improve the user experience. Still the shared data has to be secure in a manner that an outsider cannot extract information about the user.

Most of the research done on positioning cooperation is in the context of robot swarms or specialized equipment. Purely inertial dead reckoning drifts over longer distances and the accuracy can be significantly improved if some kind of distance measure, even inaccurate one, is obtained to some reference point. The work by Baccou et al. (2001) considered autonomous underwater vehicles that used a distance measurement to a leader unit to correct their positioning drift. Kurazume et al. (1994) studied similar system with robots where the robots moved in turns and used each other as a reference position. There is some work done with smart phones. Mostly however it is basic research and not easily implemented in practice on current phone technologies. For example they often make large assumptions on phones' ability to communicate freely with each other. Communication is in practice always severely limited by user privacy regulations in any mobile OS.

Iwase and Shibasaki (2013) proposed a system where accuracy of pedestrian dead reckoning (PDR) would be improved using inter device collaboration. This can significantly reduce the cumulative error due to the biased noise in the sensors. In their system Wi-Fi communication signal strength was used to measure distance between the smart phones. They concluded that the distance estimate is too unreliable in larger distances so they limited its use to proximity detection. A step counter based PDR with fairly high heading errors was corrected using this proximity detection to fix device positions close each other when a close range was detected. This solution assumed that the pedestrians usually walk in relatively straight lines and created a node link system from these paths where the optimized variables were the angles between the links. It is not clear how well their results would apply to a more free system where any curvature is allowed in paths.

Strömbäck et al. (2010) also studied improving the PDR accuracy with collaboration. Their work however was intended mainly for emergency first responder staff and military who can use specialized sensor and communication equipment instead of a mobile phone. They describe a system where a PDR is formed using foot mounted sensors, which in itself allows a much better PDR than just a hand held device, and UWB ranging radio information is used with a Kalman filter for sensor fusion to correct the cumulative drifting of the position. The results are good at least for relatively short sessions of first responder use case. It is clear that the results cannot be assumed to work similarly with the mobile device case but the basic principle of

using a Bayesian filter implementation for optimal sensor fusion to get a good track estimate is promising.

Chan et al. (2006) used distance estimates to improve Wi-Fi positioning results. A confidence value was assigned to positioners according to the underlying positioning algorithm and for each detected pairs in close proximity corrected the lower confidence node according to a distance measurement. They used ZigBee radios to estimate the distances between users and achieve significant improvements in location accuracy. ZigBee[3] is a local area network radio protocol mainly intended for embedded applications. As ZigBee is not a standard feature in current smart phones this solution is not available for a system that tries to achieve positioning without any extra equipment or infrastructure. ZigBee has many advantages over standard Bluetooth such as lower power consumption and simpler network management (Larranaga et al., 2010) but no comparisons were available between ZigBee and more recent Bluetooth low energy systems.

Some systems have been implemented where collaboration is used for mapping or *fingerprint* creation and maintenance (e.g. Bolliger, 2008). In these systems the users provide information about the environment to be used by other users. Wi-Fi fingerprint *auto-healing* is essential part of any Wi-Fi based positioning system as the access points are not completely static but can be moved, added or removed form the venue. There are already some patented systems for auto-healing (Morgan et al., 2009). In more general sense the simultaneous localization and mapping (SLAM) possibilities have attracted a lot of interest because of the possibility to crowd source the laborious map creation phase in the IPS setup.

While there are multiple studies about using the BLE beacons in positioning (e.g. Faragher and Harle, 2014), no previous studies about phone to phone BLE transmissions in positioning could be found.

[3]ZigBee specification, 2012, Zigbee Alliance, `http://www.zigbee.org`

3 Theory and Methods

This chapter goes trough in detail the mathematical basis and theory and practical system implementations used this thesis. The first part presents the pedestrian simulation model. Following that the BLE technology and its use is positioning is presented.

3.1 Simulating pedestrian movement

To help with the complicated first fix problem a collaborating positioning session should be available at the initialization of a new positioning session. What is the actual probability of that happening as a function of the amount of users is a relevant research question. If we have a large indoor map, even if we had multiple users on it, they might be so far from each other that the collaboration would not work.

Figure 3.1 shows an example amount of simultaneous users on one of the actual IndoorAtlas IPS maps. This kind of data gives some direction on the usefulness of collaboration but lacks a large amount of relevant information. For example we don't know where the positioning sessions are relative to each other or how many people were there in total. Also it would be useful to know how the situation changes when in the future the number of users will significantly increase.

Since real data about indoor positioning users movement in indoor environment is not readily available it becomes necessary to find another way to answer the question. Fortunately it is possible to simulate plausible pedestrian paths in indoor environments.

Intuitively it would be simple to distribute simulated people evenly to an indoor environment and check the fraction of the overall area covered by positioning sessions. However no real indoor environment actually works like that. Some paths, like ones leading to the main entrances, are more common than paths in peripheral areas. This leads to bigger crowding in some areas. Also the positioning sessions are more likely to start on some areas over other areas. By simulating actual plausible pedestrian activity in a real indoor map the results should be much more realistic.

There are several mechanisms for pedestrian simulation presented in literature. These have been mainly developed for two distinct use cases. Firstly there are simulators for architects used for example for testing emergency escape situations for big crowds. Secondly there are simulators for computer animation and game purposes used to automatically create realistic models of crowd movement.

For this use case the movement dynamics model can be relatively simple as the only needed functionality really is collision avoidance. Hence the chosen model for that is a cellular automata model where the crowd dynamics is relatively simple to model. The system is designed following guidelines laid by Dijkstra et al. (2001a,b). Their model is based on a simple cell lattice of walkable environments and state updates for each pedestrian using a simple rule structures. While the core structure is based on the work by Dijkstra et al. the model is significantly modified for this use case. The step dynamics is adapted from social force ideas presented in article *Social force model for pedestrian dynamics* (Helbing and Molnár, 1995). The main idea is

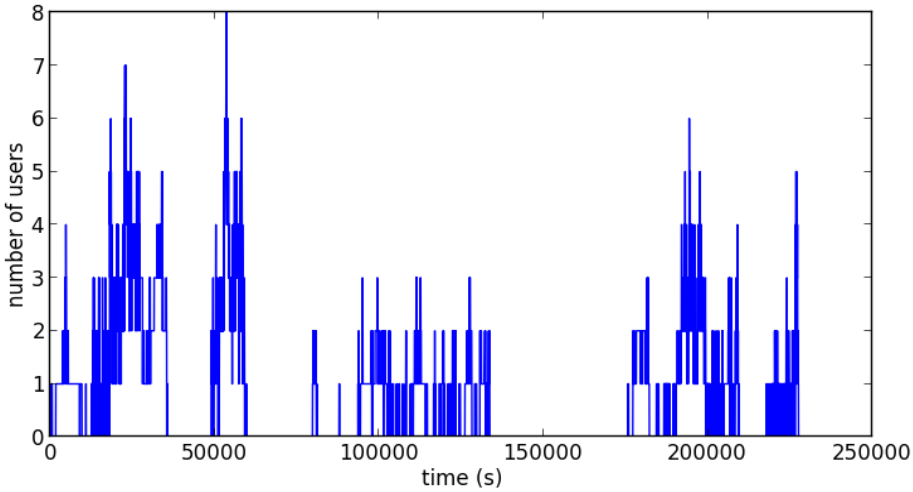


Figure 3.1: Amount of IndoorAtlas service users on one of the maps second by second during a weekend in January 2016.

that obstacles and other pedestrians cause repulsion forces to different directions and these are summed with the force caused by the desire to reach the objective. Following is the description of the cellular automata crowd simulator implemented for the purposes of this thesis.

Cellular Automata Pedestrian Simulator

The simulator consists of two distinct parts. First there is the underlying waypoint network of nodes forming all possible macro paths in the map. Secondly there is the cell lattice defining walkable space and other pedestrians to implement the actual crowd dynamics and micro paths. Here I define the term macro path to mean the general map path from start node area to end node area visiting all node areas in the path and term micro path to mean the actual path in the lattice avoiding obstacles and other pedestrians.

The cell lattice is a two dimensional square lattice with every cell defined to have width of 0.4 meters. The cell width is chosen to be the plausible space needed by a human in a very crowded situation. Also it is what was used by Dijkstra et al.. The lattice actually consists of two layers. The first layer is a binary layer having value 0 for a free cell and 1 for a cell occupied by a pedestrian. The second layer defines the constant obstacles in the map such as walls. The second layer also defines the different decision areas of the map such as entrance areas and end node areas. These are described in detail later. The second layer has a negative value for a non walkable area such as walls, 0 for walkable normal area and a positive integer for different decision areas. The integer corresponds to waypoint number. Formally the cells are defined as c_{ij}, where $1 \leq i \leq H$ and $1 \leq j \leq W$ and W and H are width and height of the lattice. The state and type of cell $c \in \{\{occupied, free\}; \{empty, wall, entrance, endpoint, waypoint\}\}$.

Pedestrians are created randomly from a Poisson process. Tuning the parameters of the random process affects how often a new pedestrian is created to the simulation. To make the simulation more realistic with respect to groups a random draw from a Poisson distribution decides how many similar pedestrians are created at the same time. At every update step there is a probability of

$$p = \lambda \Delta t \tag{3.1}$$

that a new pedestrian is created. Parameter λ is the probability per second and Δt is the length of the time step. If there is more than one entrance node in the map, the entrance is chosen randomly with possible weights affecting the choice. The amount of similar pedestrians created is drawn from the distribution

$$N_{\text{pedestrians}} \sim \text{Poi}(1) \tag{3.2}$$

creating a situation where 1 pedestrian created at a time is the most common situation but groups occur frequently.

Every time a pedestrian is created a new goal path is created too. Every pedestrian has a number of goals in the map. The amount of goals is drawn from a Poisson distribution

$$N_{\text{goals}} \sim \text{Poi}(0.5) + 1 \tag{3.3}$$

and randomly chosen from the possible endpoint nodes. Also for each goal node a random wait time is assigned. This wait time can be adjusted for different situations but for these simulations a draw from interval $[240, 1200]$ update cycles was used. One update cycle roughly corresponds to 0.5 seconds in simulation time so this means wait times ranging from two minutes to ten minutes. In a mall example case this represents the time spent at a shop. Each goal sequence ends back in the entrance and when every goal has been visited the pedestrian is deleted. The random elements are decided for each group so pedestrians created together have similar goal sequence. To summarize, each pedestrian starts at a random entrance node, goes trough a sequence of random endpoint nodes waiting a random time on each node and returns to an entrance to be deleted. Each pedestrian group also has a slightly randomized basic walking speed. The actual speed each step has additional small random element. This system of pedestrian behavior is extremely simplified but should be adequate for this use case since the goal is not to model individual behavior but rather the large scale distribution of the pedestrian paths in an indoor floor plan.

The social forces affecting a pedestrian are calculated away from non walkable objects such as walls and other pedestrians in the neighborhood. The forces can be described by

$$F_{\alpha B} = -\nabla U(\|r_{\alpha B}\|) \tag{3.4}$$

where $r_{\alpha B} = (r_\alpha - r_B)$ is the vector between the object B and the pedestrian α. U is a function that monotonically decreases as $\|r_{\alpha B}\|$ grows. The desire to reach objective can be modeled similarly using attractive force instead of repulsion. The forces caused by different objects are summed to determine the next desired step direction.

The update and the movement dynamics of a pedestrian can be described in several steps. The steps are shown in Algorithm 1. The speed of the pedestrians is tuned so that the maximum number of cells crossed during an update is 1. The check for the shortest path in step 4 prevents crossing of narrow walls while taking a step over a cell.

Algorithm 1 Pedestrian update

```
Step1: check if active
  if active then
    Continue to Step2
  else
    Increment wait counter
  end if
  if wait_counter > wait_time then
    active ← True
    End update
  end if
Step2: Check if goal sequence is finished
  if goal_state ≥ goal_sequence_length then
    Delete pedestrian
    End update
  else
    Continue to Step3
  end if
Step3: Determine next path
  if waypoint_ path_state ≥ waypoint_path_length then
    Calculate network path to next goal
    Continue to Step4
  end if
Step4: Check if in waypoint
  if Current_cell_type = next_waypoint_in_path then
    Increment waypoint path state
    if waypoint path state ≥ waypoint path length then
      active ← False
      Start wait timer
      End update
    else
      Continue to Step5
    end if
  end if
```

The waypoints are defined in the lattice to be an area around the node centerpoint. The area size can be individually tuned for each map but here it is a square with width of five cells. The direction to the waypoint is calculated by picking a random cell

---

Step5: Calculate direction to next waypoint

if no walls close in chosen direction **then**
  Continue to Step6
else
  Iteratively turn to both directions until a free direction has been found
  Continue to Step6
end if

Step6: Calculate step

Determine actual walking direction by checking the neighborhood and calculating repulsive forces caused by walls and other pedestrians. Calculate the next cell.
if next cell and path to it is empty **then**
  Continue to Step7
else
  Check the Moore-2-neighborhood of the target cell. Choose randomly from empty cells within the same distance.
  if no empty cells **then**
    Check the opposite direction (to give way in narrow path collisions)
    if no empty cells **then**
      End update
    end if
  end if
  Continue to Step7
end if

Step7: Walk

current_cell_state ← empty
current_cell ← next_cell
current_cell_state ← full
End update

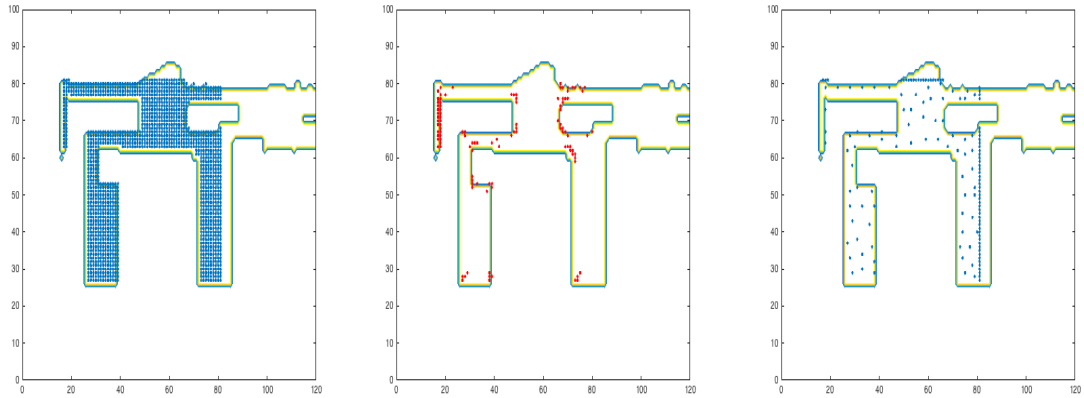


Figure 3.2: Figures representing the map graph creation. In the left side is the initial nodes for a partial map, in the middle is the detected corners and on the right side is the clustered graph for the same partial map. In the right side image the borders of the partial map have been left unclustered for connecting to the next map part. The slight alignment issue with the walls is due to the plotting. The full final map can be seen in Figure 4.1.

from this area. The waypoint path between goals is calculated using the Dijkstra's algorithm(Dijkstra, 1959) on the distance weighted graph.

Automatically Generating the Waypoint Graph

For the first tests of the simulator the waypoint graph was manually defined. This proved to be an inconveniently laborious task as the position and type of each node had to be manually defined. Thus an automatic system was designed. Waypoint graphs are commonly used in computer games to calculate paths for AI character movement and there are existing methods for at least partly automatic waypoint generation. However, many of those systems include significant portion of manual work.

The method in this study is based on ideas presented by Warhana et al. (Wardhana et al., 2013) and adapted for this use case. It consists of manual map mask building and automatic simulation map generation. The manual part of creating the mask can essentially be done in any simple drawing software and takes less than ten minutes even for bigger maps. In this part the walkable area is painted in one color. Then the entrance and endpoint areas are added in distinct colors. These colors are given to the simulator and the automatic map creation will use them to determine the cell lattice and the waypoint node types. If the endpoint areas are large (as is the case with the example mall model) and there are many endpoint nodes the map generator will allow a random selection of them to be endpoints.

The automatic waypoint generation first creates a connected square graph. Basically this means creating a node for each cell of the map lattice. The amount of

nodes is then reduced to reasonable level by clustering. Since for bigger maps the first part can include tens of thousands of nodes resulting in huge adjacency matrices, the waypoints are generated in parts. The procedure is described in Algorithm 2. It is also visualized in Figure 3.2.

Algorithm 2 Automatic Waypoint Generation

Step1: Preparation

1: Create the lattice based on manually created map mask
2: Divide the map to smaller sub maps (e.g. 80×80 cells)

Step2: Do for each sub map

1: **for all** sub maps **do**
2: Create a square graph with a node for each lattice cell and mark the border nodes
3: Remove nodes that are not in the walkable area
4: Detect corners in the mask and mark the corner nodes
5: Reduce the amount of nodes around the corner nodes and fix the remaining corner nodes in place
6: Reduce the amount of nodes around every node excluding the border nodes and the corner nodes
7: Unite the sub maps by connecting the corresponding border nodes
8: **end for**

Step3: Do for the entire map

1: Reduce the amount of nodes around the border nodes in the unified graph
2: Create the distance weighted adjacency matrix for the graph and define the entrance and endpoint nodes according to the mask

The detection of corners in step 2.4 of Algorithm 2 is done based on two dimensional Harris' algorithm (Harris and Stephens, 1988). The algorithm defines the matrix

$$\mathbf{S} = \begin{bmatrix} \sum v_{xx} & \sum v_{xy} \\ \sum v_{xy} & \sum v_{yy} \end{bmatrix}$$

where $\sum v_{xy}$ represents the sum of the element product of the x and y derivatives of the map mask at the neighborhood of the node. The corner are detected by analyzing the eigenvalues of the matrix. If the node neighborhood is in the corner, both eigenvalues are relatively large. If only one eigenvalue is large the node is in the edge of walkable space but not in the corner.

The reduction of nodes is done by iterating trough the nodes in random order. For each of these pivot nodes the three walk neighborhood is selected. That is all the nodes that are connected to it by at most three steps. This neighborhood is unified into one node maintaining the connections outside the neighborhood. The position of the new node is the mean of the unified node positions. On average, assuming original node distances of 0.4 meters, this results in node distance of around two to three meters. If the new position is inside a wall, the original position of the pivot node is chosen instead. The iteration process makes sure the new unified nodes

are not counted to be in the neighborhood of any other node so every node is only reduced once. This reduction is first done using the corner nodes as pivots, then all the other nodes excluding the corner and border nodes and finally the border nodes in the unified graph.

3.2 Estimating the Relative Position of Another Device

The most direct way of cooperating in indoor positioning would be sharing information about your own estimated location and receiving the same information from others. If the positioning sessions are running on same service this can be done easily on server. However this is only useful if the devices can also estimate their actual relative positions and compare it to the estimation if their own position.

Position Estimation Using Radio Signals

There are several ways to estimate the relative positions between two radio devices. These include methods based on signal strength, signal round trip latency and the angle of arrival.

Localizing based on round trip latency is a simple enough system but it is very dependent on accurate clocks and predictable message processing times as the signals move with the speed of light and latencies due to distance are extremely small. Many devices do not have accurate enough clocks for the task but there are ways to improve performance on bad time resolution for example by sampling fast and averaging (Günther and Hoene, 2005). Problem is that many of the methods have fairly slow communication frequency so it would take too long to average several measurements for this to be useful. For instance the BLE method explained later in this work can have anything from 0 Hz to 10 Hz received sample rate. In indoor environments the signal reflections from walls can also affect the actual traversed distance. However in the context of this work the biggest problem is that the round trip latency requires two way communication and reliable signal reception that can be hard to achieve indoors using only existing smartphone systems and hardware.

Angle of arrival estimation is an old method used in many systems such as RADARs. This however usually requires a specialized antenna array that is not available in standard mobile phones. The angle of arrival approach also has the problem of not being very useful when only one reference point is available and if the phone orientation can change spuriously in users hands.

By far the most promising method in the context of smart phones is using the signal strength. Using some kind of radio signal propagation model a distance estimate can be calculated when the transmission power and received signal strength is known. Ideally this model would have to take into account the properties of indoor environments as they can be very different to a simple free space model. However as this kind of information is not necessarily available some kind of approximation usually has to be used.

Bluetooth

Bluetooth is a communication and data transfer technology primarily designed to connect portable devices to a master device. It was originally developed by Ericsson Inc. for connecting wireless headsets to mobile phones. It uses UHF radio around 2.4 GHz band. If the device is Bluetooth discoverable (generally means that the Bluetooth option is turned on but some new mobile devices offer a possibility to scan

without being discoverable) other devices can find it by querying active devices in range. Bluetooth divides the data into packets and sends it over 79 channels using a frequency hopping system.

Classic bluetooth communication has been tried in positioning systems in several published works (e.g Anastasi et al., 2003; Feldmann et al., 2003; Hallberg et al., 2003). However there are several problems with the implementation of these systems. First of all to get a signal strength indicator, the devices have to be connected to each other in a same *Bluetooth piconet*. This would require a permission from the user to connect the devices, which is too intrusive for a practical use case. Also the signal strength indicator in these kind of connection is not in any way reliable distance estimate as the transmission strength is actively adjusted so that the received signal stays in preferable range.

Another possibility is to use inquiry responses. If a device makes a Bluetooth inquiry other devices around it with Bluetooth enabled will answer and these answers could in principle be used to estimate the position of a device. There is an extended inquiry response protocol that includes signal strength indicators but it is not standardly used. It would also be possible to send inquiries with different transmission power levels and check which devices answer. This method however has the problem of slow communication. It can take over 5 seconds to get the responses and still miss out a lot of devices (Anastasi et al., 2003). Especially if you want to have an average of many measurements this method is simply too slow.

BLE, Beacons and the iBeacon

Since Bluetooth Core Specification version 4.0 published in 2010 the standard has included the Bluetooth Low Energy or BLE technology (also marketed as Bluetooth Smart). Main features of BLE are ultra low power consumption in both peak and average situations, interoperability between different vendors and manufacturers and very low cost. The power consumption of a BLE device is so low that a single unit can run years with a standard coin-cell battery. The BLE is not compatible with the classic Bluetooth protocol but the specification allows devices to use both technologies with single radio and antenna.[4]

BLE can be used for several different tasks but the essential feature for positioning applications is the BLE advertisement technology. The advertising allows a device to transmit a standardized signal with regular intervals and standard power. This signal can hold 31 bytes of data in addition to the Bluetooth MAC address of the advertising device. The data payload of 31 bytes has to hold all the technical information needed by the protocol so the actual usable data payload is even smaller. However it is entirely possible to include enough data for a positioning service. Very important features of these advertisement signals are that the receiving device gets the received signal strength indicator or RSSI value and the communication is entirely one directional. The receiving device can be passive listener and does not need to connect to anything. This feature is important for practical applications since a

[4]Bluetooth Specification version 4.2 , 2014, Bluetooth SIG

Table 3.1: The iBeacon data structure. The values are meant to be used having a single proximity UUID for a beacon owner and using the Major to identify the installation location (for example a building) and the Minor to identify the individual beacon in that location. However since the data is open, the usage is largely up to the implementation.

Byte(s)	**Name**	**Value**	**Notes**
0	Flags	0x02	Specified by the Bluetooth Spec
1	Flags	0x01	Specified by the Bluetooth Spec
2	Flags	0x06	Specified by the Bluetooth Spec
3	Data Length	0x1A	Specified by the Bluetooth Spec
4	Data Type	0xFF	Specified by the Bluetooth Spec
5	Company ID	0x4C	4C00 is the Apple Bluetooth identifier
6	Company ID	0x00	4C00 is the Apple Bluetooth identifier
7	Beacon Type	0x02	0215 is the iBeacon identifier
8	Beacon Type	0x15	0215 is the iBeacon identifier
9–24	Proximity UUID	0xnn..nn	Identifies the beacon installation
25–26	Major	0xnnnn	Freely definable value
27–28	Minor	0xnnnn	Freely definable value
29	Reference Power	0xnn	The calibrated signal power value

smartphone OS usually heavily limits the communication possibilities to protect the user privacy.

In 2013 Apple introduced their solution for using BLE for positioning. The iBeacon is essentially a protocol defined on top of the BLE advertisement system [5]. An iBeacon message includes an UUID value for identifying beacon owner, 4 bytes of freely definable data to identify individual beacons and most importantly a calibrated reference power level of the transmission. According to the specification this power level is the RSSI received by iPhone 5s at 1 meter distance. As the receiving device gets a RSSI value they can use it with the reference to estimate a distance to the beacon. A possible approach to the estimation is presented in Equation 3.6. The iBeacon data structure is presented in detail in Table 3.1.

Other vendors have created other similar implementations. Notable are Radius Networks AltBeacon[6], which is essentially an open source solution similar to iBeacon, and Google's Eddystone[7] which allows various different messages, for example direct sharing of URLs which is extremely convenient for sharing information from a remote source. However at the moment iBeacon has established itself as the industry standard.

Choosing iBeacons over other technologies is also often advisable on iOS devices since it requires less permissions from the OS. iBeacon messages are filtered in iOS to a separate service from other BLE and can be used with CoreLocation framework

[5] Proximity Beacon Specification Release R1 2015, Apple Inc.
[6] AltBeacon, RadiusNetworks Inc. `https://github.com/AltBeacon/spec`
[7] Eddystone, Google Inc. `https://github.com/google/eddystone`

and only the general location permissions used by all map applications while other BLE recording requires CoreBluetooth framework and Bluetooth permissions. The limitation of the technology is that the CoreLocation framework on iOS only allows an application to listen to beacons with predefined proximity UUIDs. Android devices however can listen to all of the beacons or BLE signals with no restrictions.

The amount of permissions an application has to ask from the user is a significant issue since the permission pop-ups worsen the user experience and sometimes when it is not immediately clear why the application has to access the resource it is asking a permission for the user might end up refusing it. When many of the actual use cases might require permissions for significant amount of OS services and from the user privacy view point it is good to try to limit the requirements of the IPS to the minimum.

Even with iBeacon there are notable differences on how the data can be accessed in Android and iOS systems. With iBeacons the Android OS allows access to the raw measurement data while iOS beacon API puts the data trough a preprocessing pipeline and outputs a result once every second. While the exact preprocessing pipeline is not known it can be assumed that it does some kind of filtering or averaging to the measurements and the result is likely to be of better quality than the unprocessed data in Android phones. However in this thesis the Android system and unprocessed data is used as it makes the implementations more transparent and clear.

Since iOS 9.0 and Android 5.0 (API level 21) it has also been possible to use a mobile phone as a BLE transmitter. This means that phones can essentially act as beacons for each other. In addition to providing an intuitive way of collaboration this feature makes it possible without the actual users even knowing they are helping others. The transmission on iOS requires CoreBluetooth framework and user permissions for Bluetooth usage anyways so it is possible that when using BLE transmission it would be better to define a completely new proximity beacon standard like Google did with Eddystone instead of limiting to iBeacon. Basically any data could be openly shared between the phones this way. However this thesis studies the basics of the BLE advertisement technology in positioning and iBeacon standard is readily available so it is used.

BLE Signal Strength and Distance

As noted before using some kind of a propagation model and having knowledge of some known reference power it is possible to estimate the distance between the transmitter and the receiver. in iBeacon standard the reference power is included in the BLE data so the distance calculation can in theory be done easily. There are however a large number of variables especially in indoor environments that affect the result. Furthermore the main purpose of this thesis is to study smartphone to smartphone system and in such solution the reference power would at least in principle have to be defined separately for each device as their bluetooth radios and antennae configurations differ. The problems with BLE signal strength and device differences are illustrated in Figures 3.3 and 3.4 respectively. It would be possible to

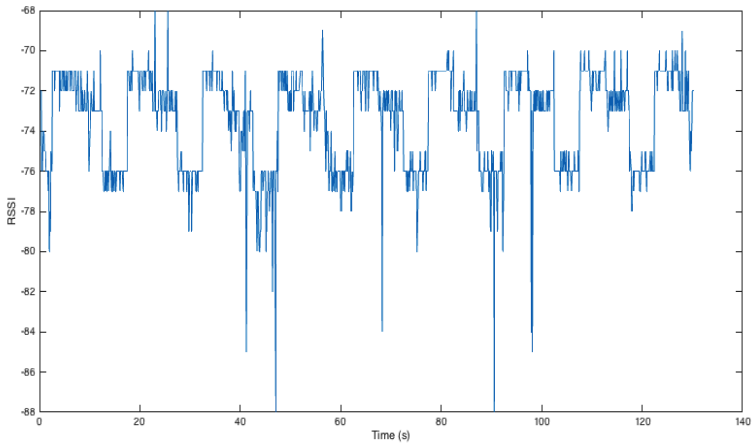


Figure 3.3: The received signal strength of BLE transmission between two smartphones. The phones are stationary on a table in a quiet room. The distance is 3 meters. Phone models used here are Samsung Galaxy S5 transmitting and Samsung Galaxy S6 receiving. Figure 3.4 implies that the periodic component results from some process in the receiving phone.

define different calibration values but it would also be highly impractical. In this thesis the calibrated reference power value for LG Nexus 5 smartphone is used for all phones with the assumption that the differences are small enough to be handled with larger measurement uncertainty.

Figure 3.3 shows interesting periodic behavior that probably results from some internal system of the phones used but the source cannot be conclusively discerned from the data. Figure 3.4 shows the same situation with three different transmitter phones. The periodic change in the signal level is highly correlated between the different transmitters suggesting that the source is in the receiving phone. Faragher and Harle (2014) suggested that Wi-Fi scanning might interfere with received Bluetooth signal strength. This might be a possible explanation for the periodic behavior of the RSSI if the device is set to scan the available Wi-Fi networks at regular intervals.

Since the signal power is not stable and the distance estimates are correspondingly unreliable Apple recommends using the iBeacon as a proximity detection system instead of actual distance estimator. However that might be overly pessimistic about the possibilities of beacon based location systems.

Experiments with the Distance Estimation

A generalized radio signal propagation models for indoor environment have a large number of parameters and would really require taking into account the shape of the rooms and locations of the transmitters to be accurate. However a less accurate approximate solution can be made more practical. An approximation for the signal loss as a function of distance would be (Fink et al., 2009)

$$L = L_0 + 10n \times \log(d) \tag{3.5}$$

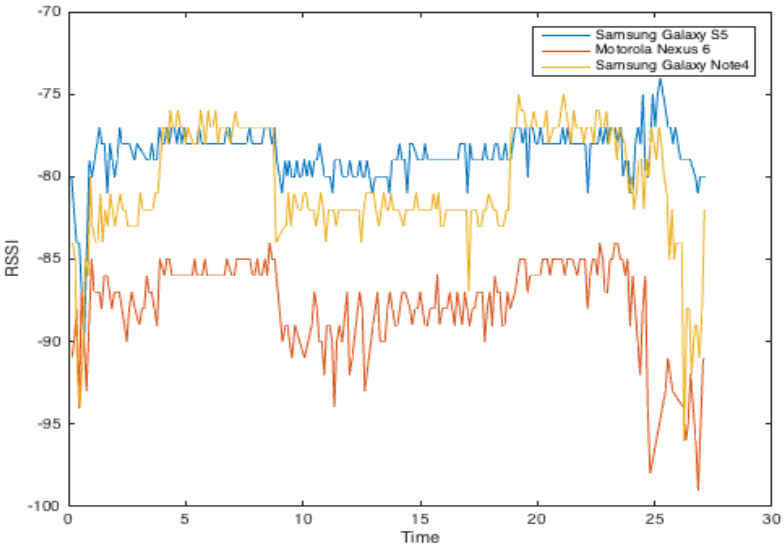


Figure 3.4: The received signal strength of BLE transmission between two smartphones using 3 different phones as a transmitter. The phones are stationary on a table in a quiet room. The distance is 3 meters. The receiving phone is Samsung Galaxy S6. The disturbance in the end is due to a human walking in front of the phones.

and when we know the RSSI the reference power and a calibration value n we can solve for the distance d. The value for n would have to be empirically determined for a phone model and environment.

However in this thesis a different method was chosen. Radius Networks has openly published their AltBeacon open beacon alternative code. They have measured the signal strength in different distances with LG Nexus 5 smartphone and fitted a distance curve to the measurements. The resulting signal strength ratio to distances equation is

$$d = a \times \left(\frac{\text{RSSI}}{\text{Reference}}\right)^b + c \tag{3.6}$$

where a, b and c are the fitted curve parameters. The Radius Networks values for Nexus 5 are 0.42093, 6.9476 and 0.54992 respectively. The distance estimate works fairly well when the phones are stationary or move in a stable fashion. It was not possible to make any real calibration to the calibrated transmission reference power level reported in iBeacon data so the tests were conducted using the "default" value of -59. This produced reasonable results with the tested devices.

The estimate was tested by placing smartphone on a table in a corridor and configuring it to send a beacon transmission. Another smartphone was held in hand and the corridor was walked in constant velocity back and forth. The true path was interpolated between spots marked in the data. Figure 3.5 shows the distance estimates calculated from the RSSI measurements using LG Nexus 5 parameters. The RMSE over the test shown is 4.9898 meters. This is quite large but most of

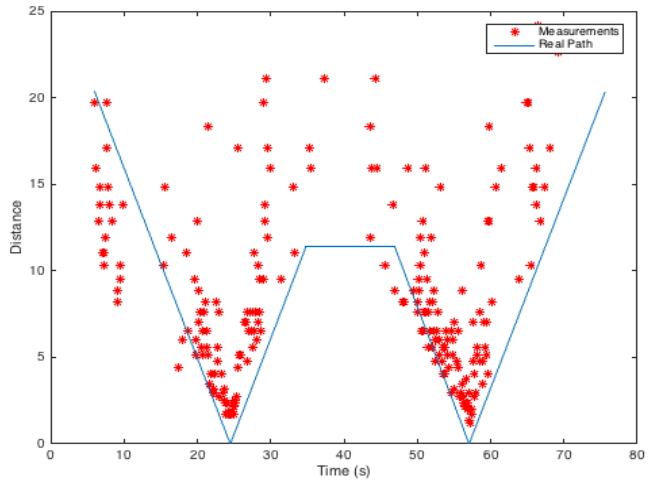


Figure 3.5: Calculated distance estimates when walking past the transmitting phone twice (in opposite directions) while holding the receiving phone in positioning orientation in front of the walker.

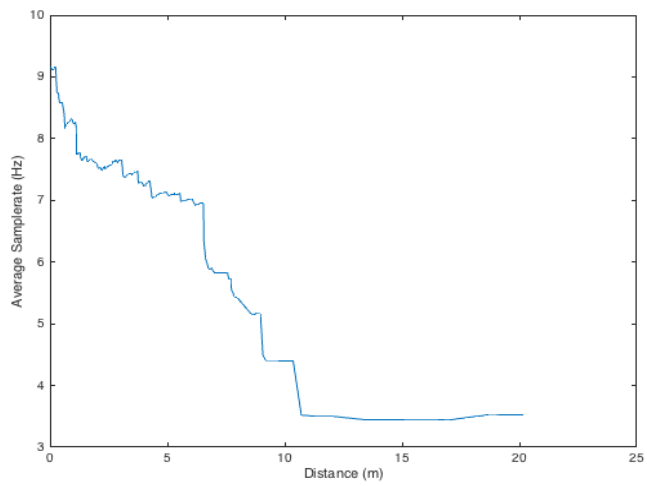


Figure 3.6: The average sample rate of the received advertisements (moving average over 1 meter) as a function of the distance. The real transmitting sample rate is 10 Hz.

the error comes from the error resulting from antennae orientation and human body absorption. The estimates are systematically too small when approaching the target and too large when distancing. Also error is larger when further away from the target as shown in Figures 3.7 and 3.8. Figure 3.6 shows how the frequency of successful received advertisements also depends on the distance.

In conclusion the data shows that the distance estimation from BLE RSSI is more reliable the closer the transmitter and the receiver are. The wrong estimates due to obstacles or signal reflections are less likely the closer the devices are and the

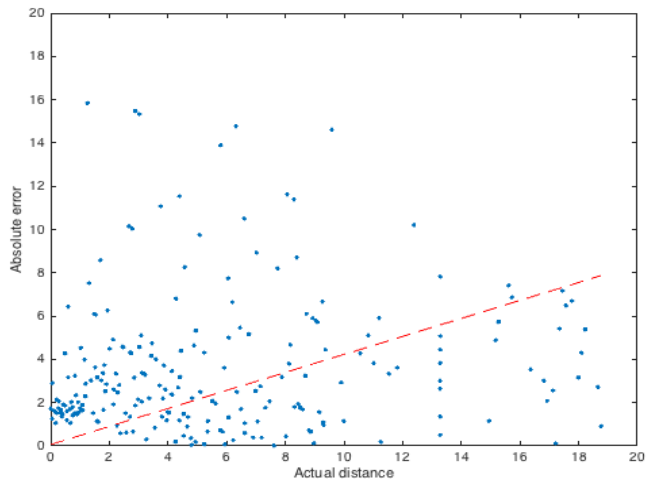


Figure 3.7: Absolute measurement error as a function of the actual distance. The red dashed line shows the linear regression.

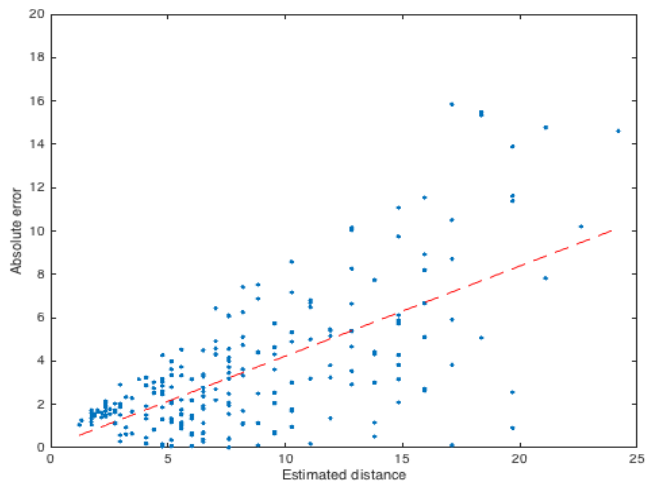


Figure 3.8: Absolute measurement error as a function of the measured distance. The red dashed line shows the linear regression.

frequency of successfully received signals is higher when the devices are close to each other. Also if the chosen calibration values are wrong the error would get higher as the distance grows. Furthermore Figure 3.8 shows that the error is clearly dependent on the received signal power. This effect is even clearer than the effect of actual distance. The stronger signals are in general more reliable than the weaker ones and this finding can be implemented into beacon based IPS. The Bluetooth signal propagation and reliability as a function of distance was also studied by Madhavapeddy and Tse (2005). Their conclusion also was that the bit error rate quickly increases with the distance. One directional communication cannot really include any correction protocols so lost bits would usually result in failed communications.

3.3 BLE in Positioning

As noted in the previous section the quality of the BLE RSSI distance estimates is not really good. In fact many seem to think that the beacon technology cannot provide the "blue dot" or a real time accurate location. However with smart ways to handle the inaccuracies in the measurements the final location estimation quality might be improved beyond the actual measurement quality.

In smartphone solutions it would be sensible to include some data from the inertial sensors a modern phone has, in short to include the PDR information, but the feasibility of pure BLE navigation in spite of the low quality of the distance estimates is an interesting question as it is also indicative of the usability of BLE in collaborative positioning.

If a distance to more than one reference point is known, a trilateration approach can be used to determine the position. As in this case the distance estimates fluctuate a lot, some kind of a filtering setup need to be implemented. With this kind of nonlinear estimation problems simple filtering such as moving average is not sufficient. Instead, Bayesian approaches are often used. These include namely Kalman filters (Kalman, 1960) and particle filters. The result from these filters is a posterior probability distribution for the position rather than an explicit position. A suitable statistic of the distribution can then be used as an explicit location. An additional advantage of Bayesian filtering methods is that they offer a natural way for sensor fusion to integrate information from multiple different measurement modalities (Fox et al., 2003).

Particle Filters in Radio Navigation

A particle filter (PF) is an application of sequential Monte Carlo methods to a Bayesian filtering problem. Monte Carlo provides a numerical approximations by drawing samples and calculating sample averages instead of direct computations. It surpasses other Bayesian filtering methods for non linear problems such as the extended Kalman filter (EKF) or unscented Kalman filter (UKF) especially in problems where nonlinearities are not sufficiently smooth for local linearization, filtering distributions are multi-modal or when the measurement noise is largely non-Gaussian. The UKF and particle filter solutions to radio signal based tracking were compared by Lee et al. (2010) and they concluded that PF outperforms UKF in real life tracking situations. A particle filter is very flexible in large and complicated problems but unlike Kalman filters it's robustness and accuracy are proportional to the computational cost. Thus the PF has in the past been considered too computationally expensive for many practical applications. However, the increase in available computing power has made them applicable even in mobile device use.

A sequential importance sampling (SIS) particle filter draws samples from a prior distribution, gives these particles weights according to their likelihood given the measurements and uses the weighted sample distribution as the posterior. In a practical case the sampling distribution should have a Markovian property

$$\pi(x_k|x_{0:k}, y_{1:k}) = \pi(x_k|x_{k-1}, y_{1:k}) \tag{3.7}$$

meaning that the next state depends only on the previous state and possible measurements instead of the entire state history. A weight $\omega_k^{(i)}$ is calculated for each particle sampled and the weighted average of the particles is used for the optimal estimate of the system state. In navigation and tracking applications the importance sampling distribution usually implements some kind of dynamic model for target movement. Thus the sampling distribution for the next step is formed from the previous particle distribution and the the movement dynamics (Särkkä, 2013, Ch. 7).

In a usual situation most of the particle likelihoods will quickly have close to zero value. This *degeneracy problem* can be countered by resampling when the effective number of particles gets too low. The effective number N_{eff} can be approximated as

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (\omega_k^{(i)})^2} \quad . \tag{3.8}$$

The term sequential importance resampling (SIR) is used when referring to SIS filters with the resampling step. The resampling can be done by interpreting the particle weight as a probability of sampling that particle and drawing new particles from the resulting discrete distribution. In other words every new particle is chosen with replacement from the set of old particles $x_k^{(i)} : i = 1, ..., N$ using the normalized weight $\omega_k^{(i)}$ as the probability of drawing from index i. This effectively removes the zero weight particles whenever the resampling is performed. The information reduction or *sample impoverishment* resulting from effectively copying particles when resampling is reduced by increasing the noise to be larger in the filter than it would be in the real dynamic model. This is sometimes called *jittering* or *roughening*. Also the measurement noise model can be increased to larger values than expected in the data to slow the decay of the particle likelihood (Gustafsson, 2010; Särkkä, 2013).

Resampling is done if $N_{\text{eff}} < N_{\text{threshold}}$. The threshold value for resampling is a major parameter for the filter performance as it effectively decides how often the resampling happens and at the same time how much impoverishment happens. A filter that resamples too often might end up with a jumping estimate that follows the noisy measurement too much while resampling too infrequently will result in a smooth but probably inaccurate estimate. The particle filter used in this work for BLE positioning is described in Algorithm 3.

In positioning application a natural state formulation contains variables for the position and movement dynamics of the system. A model where the position and velocity is included in a simple two-dimensional map would have 4 variables. The velocity vector can be expressed either in cartesian or polar coordinate values resulting in slightly different dynamic systems.

If no other data about the dynamics such as the inertial measurements is included, some kind of a random walk model has to be implemented. One natural way to implement the dynamic model would be a Wiener velocity model where the acceleration is a noise process. This was done for example by Evennou et al. for Wi-Fi based positioning solution (Evennou et al., 2005). A two-dimensional system

Algorithm 3 Particle Filter for Bluetooth Positioning

Initialization

When first measurement received create N_{init} particles randomly to 20 meter radius around the measured beacons.
Resample to reduce the number of particles to N.

The Filter

while Algorithm running **do**
  Predict by propagating the particles according to the dynamic model
  Evaluate the particle likelihood given the BLE measurements
  Calculate the effective number of particles N_{eff}
  if $N_{\text{eff}} < N_{\text{reinit\_threshold}}$ **then**
    Reinitialize by creating N_{init} particles randomly to 20 meter radius around the latest measured beacons.
  end if
  if $N_{\text{eff}} < N_{\text{resample\_threshold}}$ **then**
    Resample by picking with replacement N particles from the old particles weighted by the likelihood
    Add jitter to reduce impoverishment
  end if
  Calculate the position estimate as the weighted particle mean
end while

can be defined by the dynamic model

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ V_{x_{k+1}} \\ V_{y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ V_{x_k} \\ V_{y_k} \end{bmatrix} + \mathbf{v_k} \tag{3.9}$$

where Δt is the elapsed time from the previous update and $\mathbf{v_k}$ is a gaussian random process with zero mean and covariance of

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & 0 & \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} q_p \\ q_p \\ q_v \\ q_v \end{bmatrix} \tag{3.10}$$

Rodas et al. in their article *Bayesian filtering for a Bluetooth Positioning System* (Rodas et al., 2008) implemented a slightly different particle filter solution for Bluetooth signal strength positioning system. They defined the velocity as speed and direction instead of x and y velocities and included information about normal human walking. In practice they added feasible direction change rate for a pedestrian compared to the walking speed. Their particle update is defined as

$$x_{k+1} = x_k + \Delta t V_k \cos(\alpha_k) \tag{3.11}$$

$$x_{k+1} = x_k + \Delta t V_k \sin(\alpha_k) \tag{3.12}$$

$$V_{k+1} = V_k + v_{k+1} \tag{3.13}$$

$$\alpha_{k+1} = \alpha_k + u_{k+1} \tag{3.14}$$

where V_k is the velocity and α_k is the direction in radians. v_k is a gaussian random process with zero mean and variance of q_v and u_k is a uniform random process defined as

$$u_{k+1} \sim \pi \times U\left(\frac{V_{k+1}}{V_{\max}} - 1, 1 - \frac{V_{k+1}}{V_{\max}}\right) \tag{3.15}$$

where $V_{\max}$ is the maximum plausible walking speed set to be $1.5\frac{m}{s}$.

In this work the previous model will be slightly modified to include some inertial information. Most modern smartphones have at least a built in acceleration and gyro measurement chip. To improve from a random walk model especially turning information would be important for steering the particles to the correct directions for example when turning from a corridor to a room. A Wiener velocity model does not really handle sharp turns well. The turned angle can be calculated by integrating the gyro measurements over the filter update period. As we are only interested in two-dimensional systems we only need the rotation around the vertical axis which, when the phone is oriented the screen upwards, is the z-axis on standard smartphone sensors. However, the phone might not be level when the positioner walks so we need a system of correcting the possible orientation differences.

With pedestrians the accelerations measured from normal movement are so small that the gravitational acceleration can be assumed to dominate the measurements.

When the phone is level the screen upwards the gravitational acceleration points to the positive z-axis. The three dimensional rotation matrix can be calculated between the measured acceleration vector and the reference orientation vector and this rotation matrix can then be used to rotate the gyro measurements to correct coordinates making the measurements effectively orientation invariant. This is not necessarily very accurate as the acceleration vector can be assumed to fluctuate a lot but large measurement noise will be used in the system which should cover the errors. A more accurate and robust gravitation tracking would be possible for example by using a Kalman filter based solution (Särkkä et al., 2015) but for the purposes of this work the simpler direct solution is good enough.

For a full PDR model an accelometer based step counter could be created to estimate the speed and traversed distance but that is not really necessary in the scope of this thesis since there needs to be large random noise on particle velocities anyways to account for possible differences in step length. The random speed model should suffice here. Thus only the Equation 3.14 of the dynamic model will be modified

$$\alpha_{k+1} = \alpha_k + \int_{t_k}^{t_{k+1}} g_z[t]dt + u_k \tag{3.16}$$

where $g_z[t]$ is the orientation transformed gyroscope measurements around the vertical z axis and t_k refers to the measurement time at update step k. u_k is zero mean Gaussian random noise with covariance of R_k^g which is the uncertainty assigned to the gyro measurement at a time step k. Both the Wiener velocity random walk model and the model with gyro measurements are tested in this thesis.

The previous paragraphs described two options for the prediction step of the filter derived from a dynamic model. Next step in the Algorithm 3 is to update the particle weights. The particle likelihood for one measurement is calculated as

$$p(y_k^{(j)}|x_k^{(i)}) = \frac{1}{2\pi\sigma_k^j} \exp\left(-\frac{1}{2(\sigma_k^j)^2}(y_k^{(j)} - d_k^{(ij)})^2\right) \tag{3.17}$$

where $y_k^{(j)}$ is a distance estimate calculated from a measurement from beacon j at the kth update step, $x_k^{(i)}$ is the particle i at the kth update step, $d_k^{(ij)}$ is the particle i distance from the beacon and σ_k^j represents the distance measurement uncertainty. In the previous sections it was noted that the further away measurements are and the weaker the received power is the less reliable the estimates are and the received sample rate lower further away. This has been taken into account by making the uncertainty a function of the measured distance

$$\sigma_k^j = a\left(y_k^{(j)}\right)^2 + c \tag{3.18}$$

where a is a scaling factor. This results in very large uncertainties for further away beacons and weights heavily the stronger signals received from smaller distances. Constant c makes sure the uncertainty never gets too low. Another adaptation was that distance measurements less than two meters were interpreted as being at the beacon. That is **if** $y_k^{(j)} < 2\,\mathrm{m}$ **then** $y_k^{(j)} \leftarrow 0$. This was done because given the two

dimensional system the beacon rarely was in a position where the very close values would be given and passing a beacon is difficult for the filter in a situation where no other signals is heard as it cannot always know if the user passed by the beacon or turned back at the beacon.

Considering the likelihood over M measurements we obtain

$$p(y_k|x_k^{(i)}) = \prod_{j=1}^{M} p(y_k^{(j)}|x_k^{(i)}) \tag{3.19}$$

where M is the number of measurements received during the update period. The complete weight update calculation for particle i is

$$p(x_k^{(i)}|y_k) \propto \left(\prod_{j=1}^{M} p(y_k^{(j)}|x_k^{(i)}) \right) p(x_k^{(j)}|x_{k-1}^{(i)}) \tag{3.20}$$

where $p(x_k^{(j)}|x_{k-1}^{(i)})$ is the proposal distribution calculated from the previous particle state and the dynamic model. After the update the the normalized weights $\omega_k^{(i)}$ are given by

$$\omega_k^{(i)} = \frac{p(x_k^{(i)}|y_k)}{\sum_{j=1}^{N} p(x_k^{(j)}|y_k)} \tag{3.21}$$

Because of the measurement inaccuracies and the beacon positions the location estimate likely traverses outside the walkable space in the map. This can be addressed by using a predefined binary mask of walkable space. The mask prevents the particles from crossing walls by setting the particle weight to zero if a wall is crossed. This should improve the positioning accuracy significantly. However when using the mask there is a risk that the particles get trapped behind a wall and cannot converge to the right position forcing the algorithm to reinitialize. The system is tested with and without the mask in Chapter 4.2.

Integrating BLE Measurements to Another System

The BLE signals might be usable in real time accurate positioning. However when using only BLE that would require a beacon installation that is dense enough so that at least one, preferably multiple, beacons are in range at all times. On many large venues this might require hundreds or thousands of beacons. If we however use the beacons to improve another positioning system such a comprehensive beacon installation might not be needed. Instead just a handful of beacons in strategic positions would suffice. If the beacons are installed on the most probable positioning starting points such as major doorways or to known difficult areas such as near escalators, the beacon measurements could be used to reduce uncertainty of the positioning estimate and in the best case prevent the algorithm from converging to a wrong solution and getting lost.

The geomagnetic positioning algorithm used in this work provides an estimated position on an indoor map combined with an estimate of uncertainty for the position. The uncertainty value essentially tells how confident the algorithm is that it's result

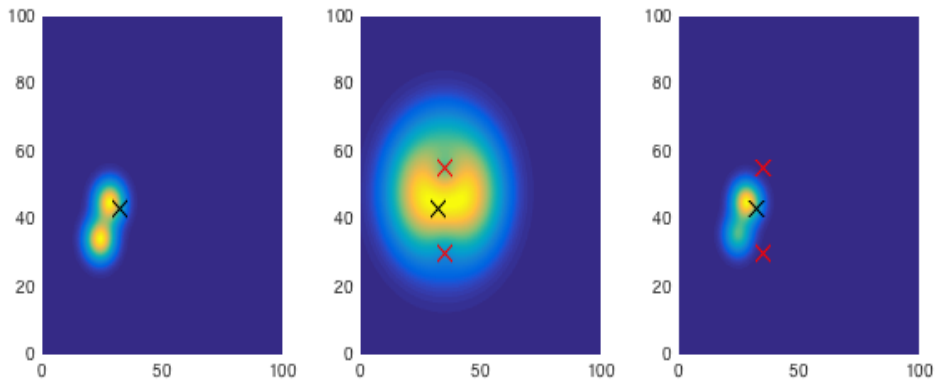


Figure 3.9: A fictional example illustrating a beacon update in positioning. The first picture represents a situation where the algorithm has two equally probable position estimates. The black cross shows the real position. The second picture represents the combined likelihood of measurements from two beacons marked with red crosses. The final picture has the posterior distribution of the position estimate showing how the true location now has higher probability than the alternative.

is the real position. The system is based on matching the measured magnetic field track to a previously recorded map. A larger uncertainty can result for example from the map having too few clear features over a large area or from bad quality of magnetic measurements during positioning. Also map matching system requires movement before it can find the real position so a reliable convergence to a single position might take some time and the uncertainty can be used to signify the state of the convergence.

The fusion of the positioning technologies can again be done in the context of the Bayesian framework. The initial positioning result can be thought of as a location prior and the posterior is the scaled result of the product of the prior and the BLE measurement likelihoods as presented in Equation 3.20. The principle of using the large measurement variance BLE to fix problems in more accurate system is visualized in Figure 3.9.

4 Testing and Results

The experiments have been divided in four separate sections. The first part considers the user simulation, the second part using a particle filter in BLE positioning system and the third part using the BLE measurements with other positioning technologies. In fourth part the actual collaborative system is tested.

4.1 Pedestrian Simulation

The experiment goal for pedestrian simulation is to get some information about the probability that BLE collaboration situations actually happens in real world. For this task a simulator described in Chapter 3.1 was implemented in MATLAB.

The crowd simulator has a probability value defining the fraction of pedestrians being indoor positioning service users. Other pedestrians are in the simulation just to create realistic crowd and track the total number of simulated people on the map. The service users have a chance to start positioning at some point during their walk. This value was chosen so that the mean time to happen was 30 seconds in simulation time. The positioning can only be active during the transitions from goal to goal in the map and goes inactive while the pedestrian is in passive wait mode.

In this simulation collaboration is defined as being in 15 meter radius from another active positioning session. The value is chosen to represent the plausible reliable signal range of mobile phone Bluetooth radio. Both the positioning sessions and the collaborative positioning sessions are counted at each update step. Also the number of positioning session starts and positioning session starts with a collaborative user are counted. This produces the desired information about the probability of collaboration in different use cases.

The simulation environment is the ground floor of Iso Omena mall is Espoo Finland. The map is shown in Figure 4.1.

Simulation Results

The probability values for the simulation were tuned so that on average a pedestrian creation occurred once per second. The equilibrium amount of simulated pedestrians was around 800 out of which around 200–300 were in active mode and 500–600 in waiting mode in target nodes. As an average simulated pedestrian combined wait time can be calculated in a simple fashion, theoretically with the given parameters the equilibrium for the waiting pedestrians should be

$$\begin{aligned} N_{\text{waiting}} &= 1 \times \Delta t \times (E(\text{Poi}(0.5)) + 1) \times \frac{240 + 1200}{2} \\ &= 540 \end{aligned} \tag{4.1}$$

so the simulated results seem reasonable.

Simulation was run using two settings for the fraction of people using the positioning service. Five runs was performed for each setting and the results averaged over the runs. One run consisted of 3200 updates corresponding to roughly 20 minutes in

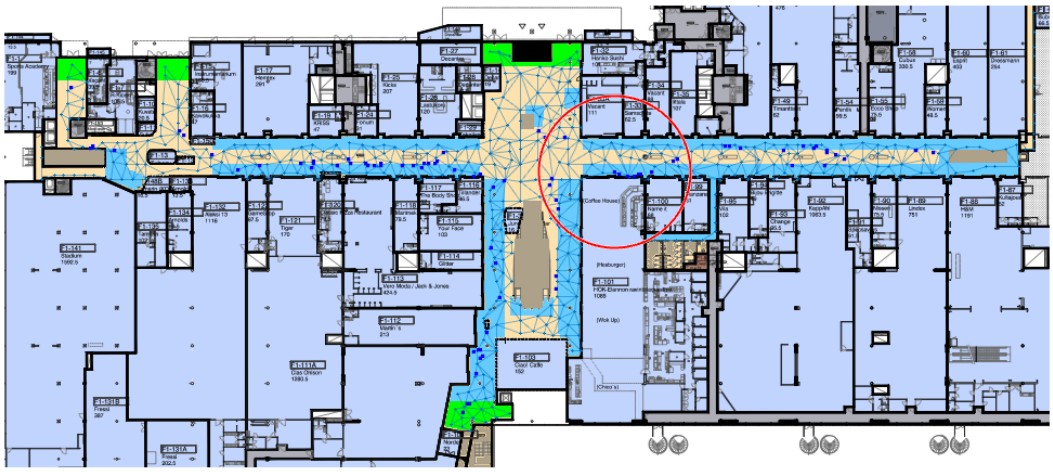


Figure 4.1: The crowd simulation. The automatically created path network is visible on the background. The dots represent pedestrians and the large circle represents the 15m radius around a positioning session. Map colors define the entrance and endpoint nodes. The real world length of the main corridor in the map is roughly 200 meters.

simulation time. This was enough to reach the equilibrium point of population on the map and see the simulation results with different amounts of pedestrians. Figure 4.1 illustrates the map, the underlying network of paths and the simulation process.

The fraction of positioning users was tuned by adjusting the probability value p_{user}. Figures 4.2 and 4.3 illustrate the simulation results for 1/100 and 1/50 fraction of positioning users respectively. The total number of simultaneous positioners seems to be similar to the numbers shown in the actual real world data in Figure 3.1. Since one of the big problems collaboration aims to solve is the initialization of the positioning session the really interesting result value is the fraction of positioning sessions having a collaborating session at start. In other words, how many positioning sessions start at a close proximity to an already running session. These values are illustrated in Table 4.1.

Table 4.1: Average number of positioning sessions per simulation and average number of sessions starting in the vicinity of already running session for different user probabilities. Averaged over three simulation runs on each user probability.

p_{user}	Total pos sessions	Collaborating at start
0.100	135	88
0.050	51	23
0.020	45	11
0.010	41	9
0.005	36	8
0.001	26	4

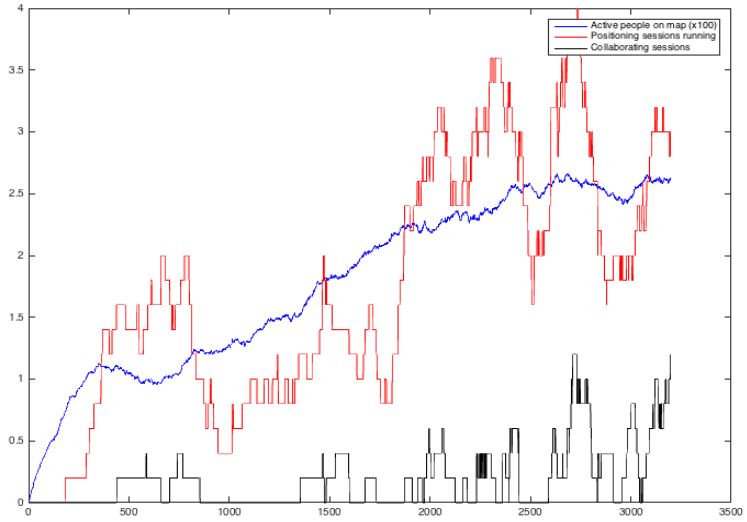


Figure 4.2: Average results of the simulation (5 runs of 3200 updates) for $p_{\text{user}} = 0.01$.

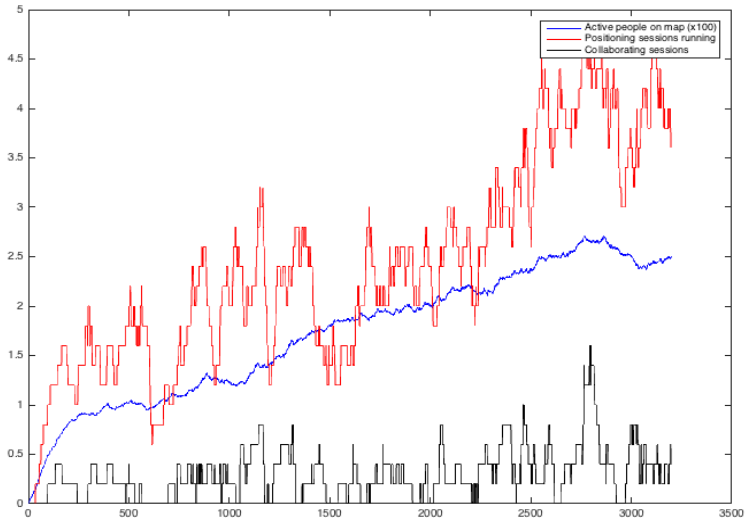


Figure 4.3: Average results of the simulation (5 runs of 3200 updates) for $p_{\text{user}} = 0.02$.

Table 4.2: The x and y direction RMSE (meters) for both of the filter implementations with and without using the walkable area mask. The x value is generally larger. This is most likely due to the fact that in this map the beacons are mostly approached in x direction causing the antennae orientation and body absorption effects to be more visible on that direction. Also due to the map topology (the narrow corridors are in x direction) the mask helps noticeably more in y direction.

	Wiener with mask	Wiener without mask	Gyro with mask	Gyro without mask
RMSE x	2.2269	2.2199	1.8182	1.7686
RMSE y	0.6687	1.5966	0.7451	1.6031

4.2 BLE Positioning

The particle filter for BLE only positioning was implemented following the theory presented in Chapter 3.3. Two different particle filter dynamic model implementations are considered here: The Wiener velocity model and the gyro based direction change model. In addition, using a binary mask of walkable area is examined. In practice this means that the weight of a particle crossing a non walkable space such as a wall is set to zero so the posterior distribution should stay inside the walkable space. As possible multi modal situations are not addressed here, the sample average can still be inside a wall if there are possible particles on both sides of the wall. This problem could be addressed by for example clustering the particles and only using the most likely cluster or using some kind of a median value instead of mean in calculating the average but since a weighted geometric median of the particle positions is not a trivial calculation it is not considered in this work.

The tests were performed in the IndoorAtlas Helsinki office. There were 5 phones in total involved. A Samsung Galaxy S6 was used as the positioning phone while Samsung Galaxy S5, Samsung Galaxy Note4, Motorola Nexus6 and Apple iPhone6 were used as beacon transmitters. The positioning phone was held in hand while walking trough a predetermined path. The transmitting phones were set on tabletops in predefined positions. A custom software had been implemented to handle the beacon transmission control.

The particle filters in these experiments use 2000 particles except for initialization or reinitialization step that uses 10000 particles to find the plausible initial positions. The positions for beacons, the real path and the filter path for different configurations are shown in Figures 4.4 and 4.6.

Results

The errors for the different configurations are shown in Figures 4.5 and 4.7. The RMSE values are shown in Table 4.2. The error plots are very similar in all configurations meaning that in general they are the result of the map and beacon configuration rather than the filter itself. The errors are also generally smaller than the actual measurement errors as shown in Figure 3.7.

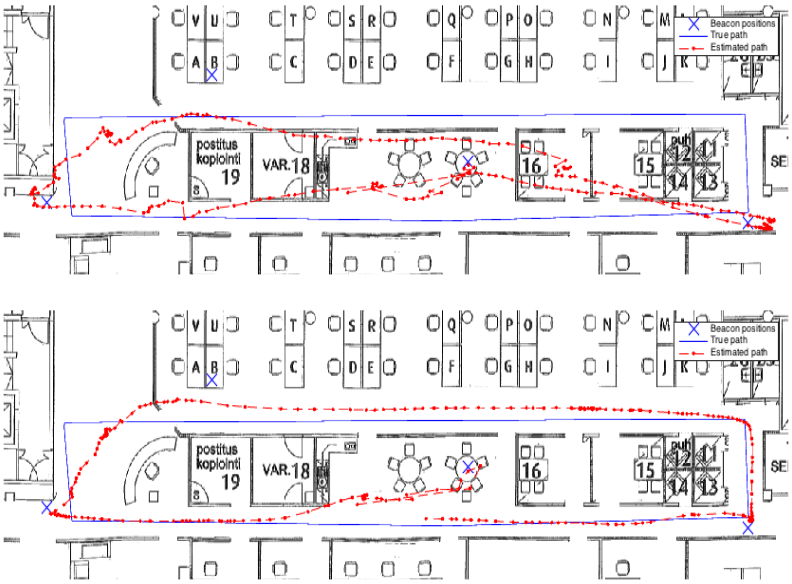


Figure 4.4: Positioning tracks using Wiener velocity random walk dynamic model without and with the map mask information.

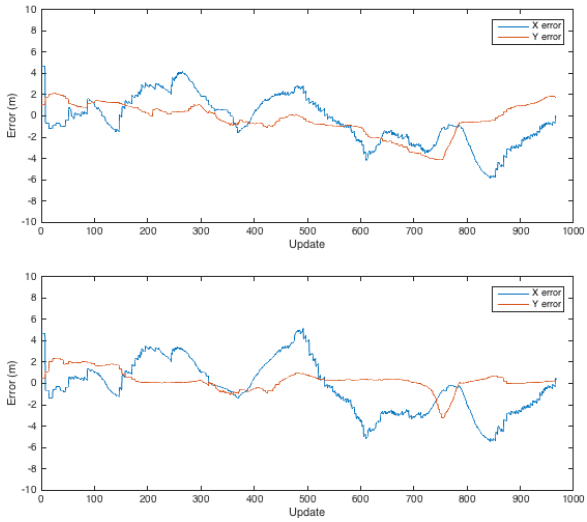


Figure 4.5: Filter error in x and y directions using Wiener velocity random walk dynamic model without and with the map mask information.

For comparison between the adaptive and static measurement uncertainty the BLE+gyro algorithm was run using a static measurement uncertainty value of 15 meters. This was value was empirically chosen. Lower values made the estimate jump around and larger values severely over smoothed the path. The results are shown in Figures 4.8 and 4.9. The RMSE in x and y directions values for this run was [2.6178, 2.2238]. Using walkable area mask did not work at all since the estimate got stuck behind corners and had to reinitialize multiple times.

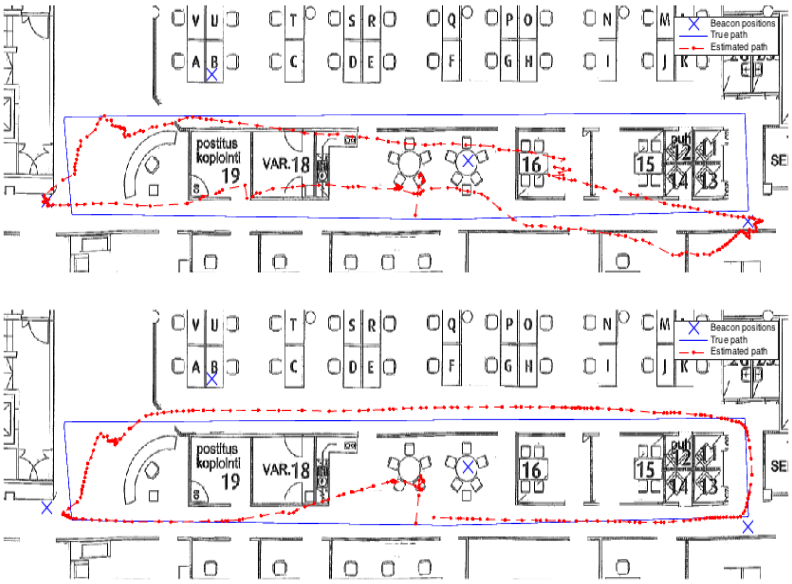


Figure 4.6: Positioning tracks using speed and direction with gyro measurements dynamic model without and with the map mask information.

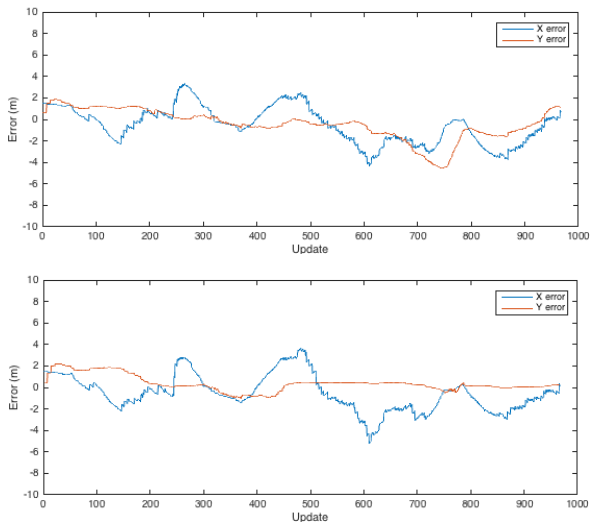


Figure 4.7: Filter error in x and y directions using speed and direction dynamic model and gyro measurements without and with the map mask information.

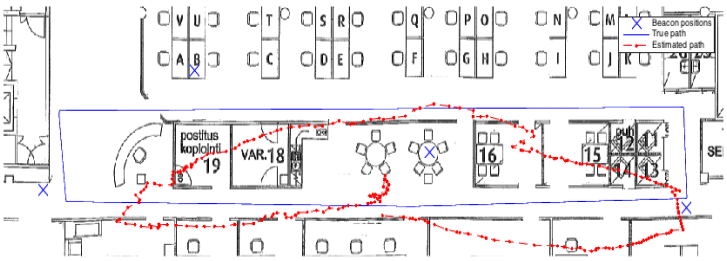


Figure 4.8: Positioning tracks using a static measurement uncertainty of 15 meters.

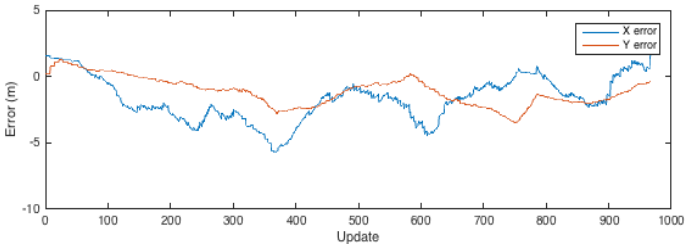


Figure 4.9: Positioning error using a static measurement uncertainty of 15 meters.

4.3 Beacon Enhanced Positioning

The fusion of BLE measurements to existing positioning system was tested on the same pre mapped office building than the tests in Chapter 4.2. The first test involved using the magnetic positioning and checking how a single beacon device would affect it. The same route was walked as shown in Figure 4.4 but instead of 4 beacons only a single beacon was positioned in the middle of the map. The beacon phone was Samsung Galaxy S5 and the positioning phone was Samsung Galaxy S6. The positioning system was run twice on the same data. First without the beacon support and the with the beacon support. Unfortunately this test setup does not include a "ground truth" path to measure actual accuracy of positioning. However the performance can still be evaluated by visually checking the result tracks and looking at the confidence values of the algorithm. This particular algorithm reports a confidence radius value in meters. It is good to remember that this uncertainty value is not an error value. The uncertainty can be high even if the positioning estimate error is low and vice versa.

The geomagnetic position used in these tests is a particle filter based positioning system that uses geomagnetic map matching together with Wi-Fi signal strength measurements and a PDR estimate to produce the estimated location of the phone (Solin et al., 2016). The BLE is integrated to the system by adding the BLE distance measurement likelihood calculation to the particle weighting process.

The BLE signal measurement model used in the experiments is similar to the one presented in Chapter 3.3. The geomagnetic positioning algorithm implementation uses the WGS84 coordinate system so the distance calculations between map coordinates had to be revised. The most efficient way would probably be using a local linearisation of the spherical coordinates as the distances are very small in global scale but here it was chosen to use the more general haversine formula to calculate the distances. The mean radius of Earth was used in the formula. The earths radius is only around 1 percent larger in the poles than on the equator so the error due to the planet not being entirely spherical should be within one percent range. The haversine formula implementation in this work for distance between latitude - longitude coordinate points p^1 and p^2 is

$$\begin{aligned} a &= \frac{1-\cos((p^2_{\text{lat}} - p^1_{\text{lat}})\frac{\pi}{180})}{2} \\ &+ \frac{(\cos((p^1_{\text{lat}})\frac{\pi}{180})\cos((p^2_{\text{lat}})\frac{\pi}{180})(1-\cos((p^2_{\text{lon}} - p^1_{\text{lon}})\frac{\pi}{180}))}{2} \\ \text{distance} &= 2 \times R_{\text{Earth}} \times \arcsin(\sqrt{a}) \end{aligned} \tag{4.2}$$

Figure 4.10 shows the average confidence radius during positioning over 6 different runs on the same path. The algorithm was run multiple times to account for it's stochastic nature. It is clear that there are some difficulties with the magnetic positioning in the end of the path and that the beacon was able to improve the situation. More importantly at no point did the BLE measurement significantly decrease the confidence even though the measurement quality has already concluded to be weak. This indicates that the large uncertainty approach to bad measurements

is working. Apart from decreased uncertainty there was no noticeable difference in the systems ability to track the user.

In the positioning data used in Figure 4.10 the path starts far away from the beacon's position and it is not heard in the beginning of the session. Thus it does not help with the initialization problem. However if the dataset is cut before starting to a point where the beacon is heard it can be used to test the effect on convergence speed too. To simulate the behavior of an iOS system, in the next test the Wi-Fi usage has been prevented and the algorithm converges using only the magnetometer and beacon information. The results are shown in Figure 4.11. It can be seen that the beacon measurements make the convergence clearly faster, almost instant when the beacon is close at the beginning of the positioning. The number of beacon messages received during an update sequence in the test varies according to the distance from the beacon. This is visualized in Figure 4.12.

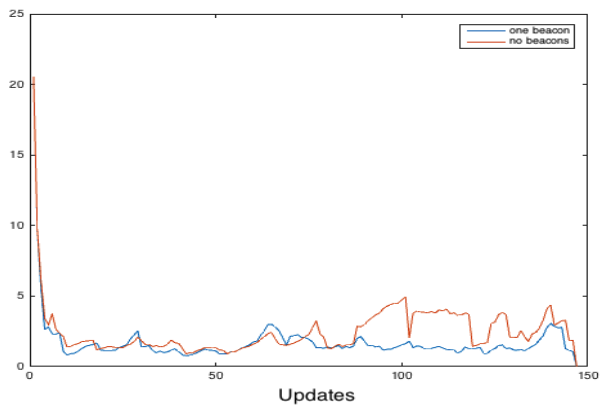


Figure 4.10: Average uncertainty radius (meters) during a positioning session with and without a beacon.

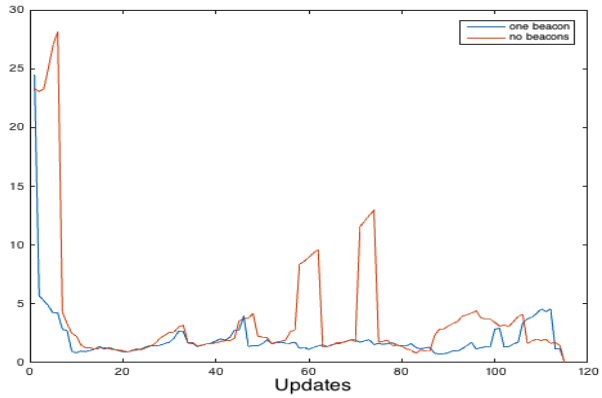


Figure 4.11: The uncertainty radius (meters) during a magnetic positioning session starting next to a beacon. The red plot line shows the convergence for the same data without the beacon.

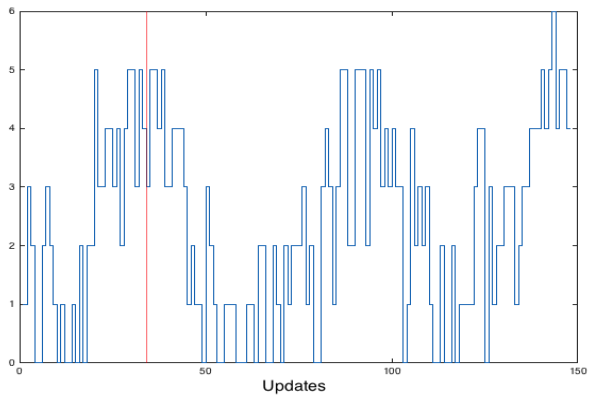


Figure 4.12: The number of beacon signals received for each positioning update from the single beacon. The red line shows where the convergence test of Figure 4.11 was started.

4.4 Collaborative Positioning

When using the BLE measurements in a collaborative sense, the reference "beacon" is another phone and is also moving and positioning. This presents a problem in that we are not entirely sure where the beacon is. In other words we have some additional uncertainty in the distance estimate. Intuitively this can be easily addressed by handling the reference position uncertainty as an additive to the BLE measurement uncertainty. However in the empirical tests it was found that the positioning uncertainty the particle filter algorithm gives does not necessarily correspond well to the actual positioning error. Instead usually if the uncertainty was high that only indicated that the position was not the correct one. The Equation 3.18 was changed so that the measurement uncertainty depends on the sum of the squares of the distance estimate and the positioning uncertainty. The new equation for measurement uncertainty is

$$\sigma_k^{(j)} = a(y_k^{(j)})^2 + b(\sigma_{\text{pos}}^{(j)})^2 + c \tag{4.3}$$

where σ_{pos}^{j} is the uncertainty of the last position of the reference device j and the constants a and b are scaling constants. The values for a and b in the tests were 1/6 and 1/5 respectively. These changes decrease the likelihood of multiple devices getting lost because some bad positioning session was trusted too much by others. The solution is not perfect as a device can have incorrect position even if it's confidence is high but this problem has more to do with the algorithm used for positioning rather than the collaboration technique itself. If the the positioning algorithm can reliably detect when the estimate is not in the correct location it can also share that uncertainty information with others. To decrease the likelihood of possible problems and slightly reduce the weight of the BLE measurements in the estimates the minimum possible value for $b(\sigma_{pos}^{(j)})^2$ was set to five meters.

The collaborative experiments were concluded by gathering sensor data in Kluuvi mall in Helsinki and running two positioning systems offline simultaneously and time synchronized to simulate having multiple users in the same building. The phones sent iBeacon signal and listened to signals sent by others. During the simulations the individual positioning systems updated their position estimates to a central location manager object that simulated a server where others could fetch the position. To be precise, the devices defined an ID formed by major and minor values, saved their position to the location manager with this ID on every update and used the same major and minor values in the iBeacon transmission. Other devices could then fetch the location with the ID they received in the transmissions.

The collaborative positioning tests have been divided into 3 separate experiments. The experiments are designed to test different scenarios for collaboration. Every experiment is performed using two phones moving the same path in the same indoor venue. These tests are done without defining the walkable area for the algorithm. This allows the estimate to escape the boundaries of the indoor space which, while inconvenient for positioning, allows better visual evaluation of the algorithm performance. The venue is rather narrow so with the walkable area defined all the tracks would be quite nicely in the right area.

(a) The tracks without collaboration.

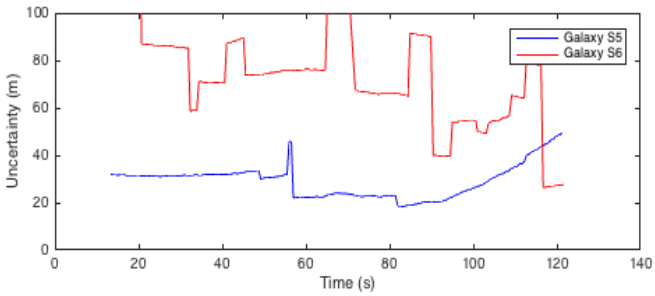
(b) Same tracks with collaboration.

Figure 4.13: The estimated tracks using PDR and Wi-Fi for positioning with and without collaboration.

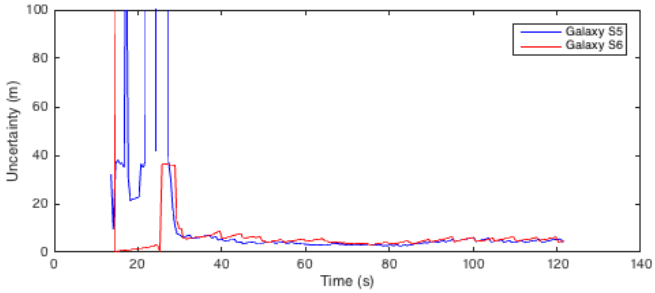
Collaborative Experiment 1: PDR and Wi-Fi

In the first experiment only PDR and pre trained Wi-Fi map is used in the positioning algorithms. The BLE collaboration is then added to the system. This attempts to simulate a situation when magnetic positioning is for some reason impossible. Two people walked next to each other holding a test phone. The resulting tracks are shown in Figures 4.13a and 4.13b respectively. The real track goes from the northern entrance to the southern entrance and back. It can clearly be seen that without collaboration the positioning system has only a very general building level idea about where the device is and the estimate jumps a lot. With collaboration, while the track is not quite correct, it is still much closer to the real location and seems to correspond roughly to the raw PDR. It is likely that with more accurate initial position and slightly better Wi-Fi map the collaborative track could be better. The uncertainty radius values for the tests are shown in Figures 4.14a and 4.14b.

The BLE collaboration improves the positioning by effectively preventing the jumping of the Wi-Fi location estimate if the separate positioning sessions have converged to the same place. It can be thought of as combining information from the Wi-Fi measurements of both devices. The situation similar to this experiment would arise if the magnetic field is unusable for some reason. The random jumping is fixed by collaboration but if the Wi-Fi map is bad as in this case it still does not guarantee an accurate result. The uncertainty values with collaboration are clearly too small compared to the actual error. This is because as the Wi-Fi map quality is not good the walked track fits the most of the measurements well.



(a) Uncertainty without collaboration



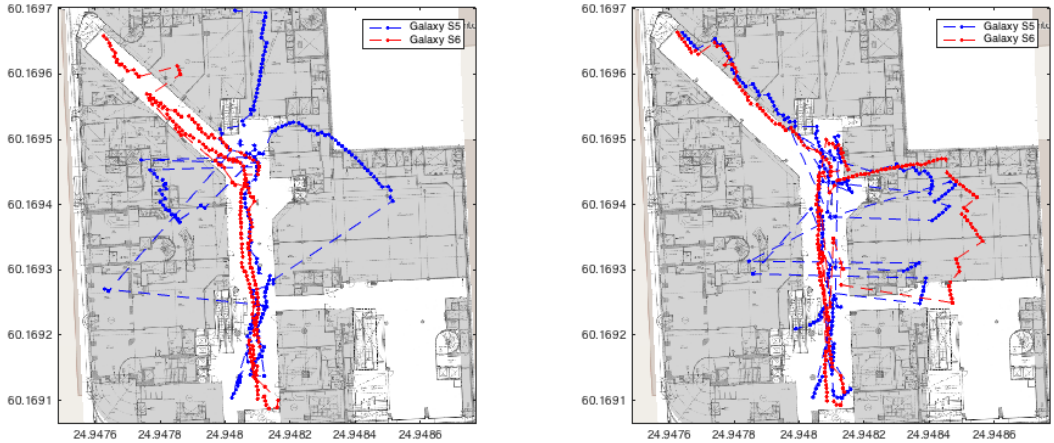
(b) Uncertainty with collaboration.

Figure 4.14: The algorithm uncertainty radius using PDR and Wi-Fi for positioning.

Collaborative Experiment 2: Collaboration and Weak Magnetic Performance

Tests have shown that the quality of the magnetic field measurements in some phones is too low for reliable positioning. There might be several reasons for this ranging from low quality sensors to problematic preprocessing done to the sensor data. If some other device is close by and has better quality measurements collaboration between these devices might help the worse behaving one to stay on right track. The collaboration tests here are conducted using two phones positioning simultaneously on the same indoor area. In this test both devices were walked next to each other from the northern entrance to the southern entrance and back on the same Kluuvi test map. Positioning session was run first using only magnetic and Wi-Fi positioning for reference and then using collaboration improve the performance.

The phones used are Samsung Galaxy S5 and Galaxy S6. The S5 is configured to use wrong settings in magnetometer calibration to make the geomagnetic IPS performance fall. The tracks without collaboration are shown in Figure 4.15a. It can be seen that Galaxy S5 fails on magnetic positioning while Galaxy S6 has fairly accurate position. Figure 4.15b shows the same path while using BLE collaboration. The S5 track has clearly improved but the S6 track has deteriorated slightly. Figures 4.16a and 4.16b show the uncertainty radius for the tests. The place where S6 lost the track is clearly visible as increased uncertainty. The spikes in S5 uncertainty are algorithm reinitializations that the system does when it thinks it is no longer on track. Large amount of reinitializations indicate that the magnetic map matching



(a) The tracks without collaboration.

(b) Same tracks with collaboration.

Figure 4.15: Estimated tracks walking next to each other with one well behaving and one more unreliable phone.

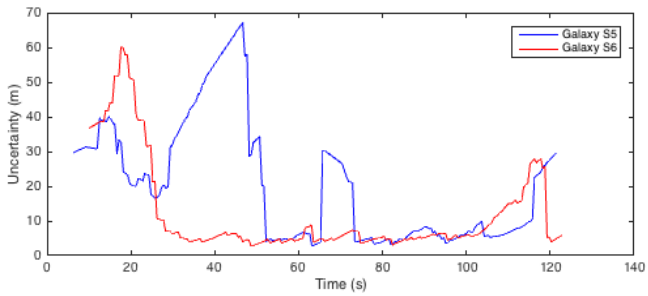
with that phone did not work at all at that part of the map.

In the uncertainty figures we can see a problem mentioned earlier. The uncertainty with a problematic phone is generally clearly smaller than it's true error. If other devices used it as a collaboration reference their positioning accuracy would likely suffer. This too small uncertainty with bad magnetometer measurements is however characteristic to this development version of the particle filter algorithm. It results from constant particle resampling when the measurements do not fit the map well. It should be possible to find solution to make the uncertainty estimate act more realistically.

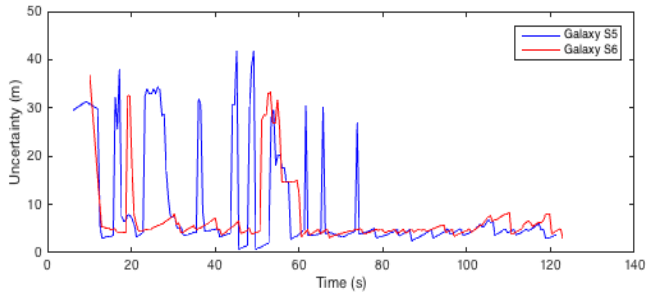
Collaborative Experiment 3: Collaboration in Ordinary Use Case

In the previous experiment the phones were walked next to each other so the BLE signal was constantly strong. Also the test cases involved a problem setting of some sort. In this test two generally well behaving phones (Samsung Galaxy S6 and Motorola Nexus 6) are tested using the combination of PDR, geomagnetic, Wi-Fi and BLE collaboration. The walked path is the same than in the previous tests except that the phones start from opposite ends of the path and meet in the middle. This should better represent a real user situation where the collaboration happens only occasionally in the middle of the sessions. The Galaxy S6 walks from north to south and back and Nexus 6 from south to north and back. Since both phones have reasonably accurate magnetic positioning result the goal of collaboration this test is not to improve the accuracy but instead show that using unreliable BLE estimate does not make it worse.

The positioning algorithm was run with and without the BLE collaboration option. The resulting tracks are shown in Figures 4.17a and 4.17b. It can be seen that the results are approximately similar with and without the collaboration. The



(a) Uncertainty without collaboration

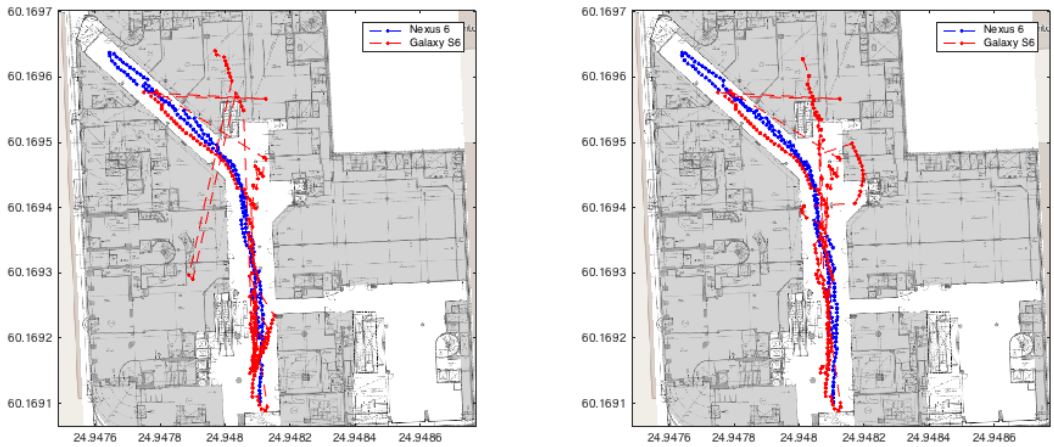


(b) Uncertainty with collaboration.

Figure 4.16: The algorithm uncertainty radius walking next to each other with one well behaving and one more unreliable phone.

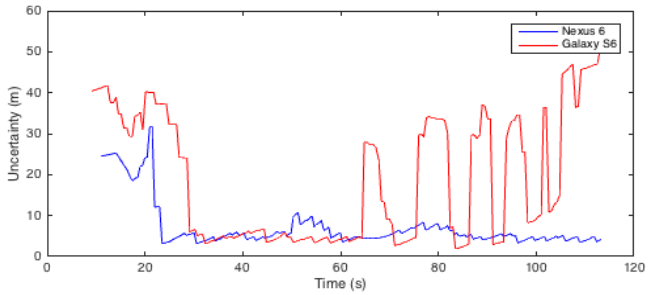
uncertainty estimates are shown in Figures 4.18a and 4.18b. The uncertainty values seems to indicate that using collaboration might have periodically improved the Galaxy S6 track but given the stochastic nature of the algorithms these can be random effects. The jumpiness is reduced but the accuracy looks roughly the same.

The collaboration does not seem to have any effect on Nexus 6. This is good because the Nexus series phones generally have very high quality sensors and behave very well in geomagnetic positioning. It seems that when the geomagnetic map matching works with high confidence the collaborative BLE does not affect the resulting track significantly.

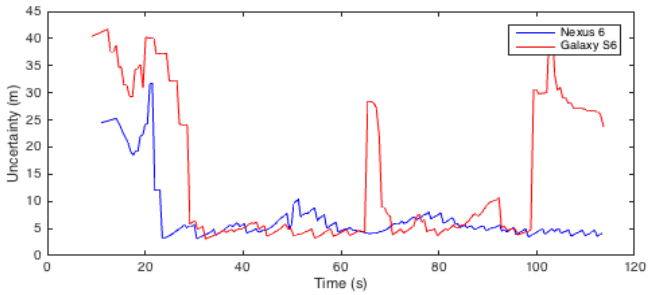


(a) The tracks without collaboration. (b) Same tracks with collaboration.

Figure 4.17: The estimated tracks when walking to opposite directions with two well behaving phones.



(a) Uncertainty without collaboration



(b) Uncertainty with collaboration.

Figure 4.18: The algorithm uncertainty radius when walking to opposite directions with two well behaving phones.

5 Discussion

The conclusions from the simulation results about the required amount of users are slightly mixed. While the numbers in Table 4.1 look promising it must be remembered that it is very optimistic to think that even one percent of people walking in a mall would use a positioning or navigation service. One percent user probability on the simulations with the example map it seems that around one quarter of the positioning sessions start with another session in range. Even with $p_{\text{user}} = 0.001$ about one sixth of the positioning sessions started with collaboration. That number is very promising considering that this simulation considered a mall which is probably not the best venue for collaboration. In a public transportation hub or a museum the number might be higher. The public transportation case has large number of people moving in the same path at the same time as a transport arrives or leaves. The museum case might involve automatic audio tours that are used by dozens of people in the same room. These cases would be much more promising for collaboration and especially the museum case might greatly benefit from increased accuracy.

In conclusion the simulations show that in general BLE collaboration situations would be relatively rare but still occur regularly. Even the collaboration at the session initialization would happen regularly so it would help at least some sessions to converge quicker. A simple BLE collaboration system can essentially be implemented similarly to any beacon based positioning system except for the transmission part so there is no particular reason not to implement the system to get the real world data even though it would not yet be used for actual positioning. However collaboration would not be the final conclusive solution to the initial convergence problems of iOS system. It would greatly help some sessions but most of the time there would not be help available.

The experiments with BLE show that it is possible to estimate distance between devices using Bluetooth low energy advertisement messages. However it is also clear that these estimates are very unreliable and that the reliability deteriorates quickly when distance between the devices grows. This deterioration is due both the environment variability increasing with distance and inaccuracies in calibration values having more effect in the lower signal strength range. As the Bluetooth is a very narrow band radio system also the BLE signal reception reliability is not good on larger distances. The errors are very non Gaussian so simple smoothing does not fix them.

It was proposed that using an adaptive measurement uncertainty that grows quickly as the signal strength decreases would help with the systematic errors in BLE measurements. It is now concluded that this approach produces reasonably good results compared to static uncertainty values. This is however assumes that when all the beacons are far away there is some other method to guide the positioning. A simple PDR is enough as long as some beacon is heard regularly.

The measurement uncertainty that grows quickly with the distance can be thought to be like a compromise between direct distance estimation and simple proximity detection. It makes the algorithm detect proximity with high confidence but still retain a smaller effect when the devices are further apart. It also means that in case

of multiple beacons the data from the closest ones are always trusted more than the data from the further away ones.

The test cases in this work were done on very easy environments without any extra distractions (e.g. large crowds blocking the signals). Still the experiments with particle filters shows that the pure BLE positioning system is feasible and can produce decent results provided the beacon installation is comprehensive enough. A possible problem yet to be solved is the significant differences between phone models. The estimates are reasonably good with the chosen test phones but some other phones on the market might have much worse results, especially since the chosen test phones are very high-end devices which is not the case for all the user devices. The error caused by differences between models is again likely to be higher on larger distances so the choice of giving larger uncertainties to low RSSI measurements might help.

If the BLE positioning was implemented on a large scale in a system like IndoorAtlas IPS where the algorithms run on server side it would actually be possible to "crowd source" the calibration of signal strength measurements. If different models move on the same map a lot the average difference between the detected BLE signal strength should in theory give a fairly good estimate of the differences in radio signal strength calibration. This would however require a large userbase to include large number of measurements from same areas with as many phone models as possible.

Using the walkable area binary mask clearly reduces the error in the direction where the walkable area is narrow. Even if it does not have much effect in the direction along the corridors it still makes the positioning output seem more trustworthy and reasonable. For end user experience this can be more important than the absolute accuracy. Even if the actual estimate error is equal the experience is likely to be better if the estimate is on a corridor rather than inside the surrounding walls. Inability to cross walls might cause problems for example in office buildings if the particles ends up in a wrong room it could be difficult to get to the correct one without fully reinitializing the algorithm.

The tests with BLE and geomagnetic positioning show that using the beacon distance measurements does not necessarily negatively affect the positioning despite their low accuracy given the high uncertainties used in the calculations. However the tests do not conclusively tell how the system works in a situation where the beacons are heard only very occasionally as would be the case in a large indoor map with only a few beacons installed. Unfortunately there was no feasible testing environment ready for studying this in the context of this work. In general more beacon sources have been found to improve the positioning at until about 6 beacons (Faragher and Harle, 2014).

The beacons can be used to make the geomagnetic algorithms converge faster and to keep them in the right track when there is spurious low quality data or when the map does not have strong magnetic features. If the algorithm gets lost and has to reinitialize itself, having a beacon in range can make the reinitialization significantly faster especially in situations where Wi-Fi measurements are not available. This should work regardless of whether the beacon is a real iBeacon or another phone as long as the beacons position is known relatively accurately.

The experiments with BLE beacon positioning were conducted to study the

general usability and methods for BLE data in positioning systems. This was done to explore the possibilities of using it in collaborative sense. The experiments with collaboration show that on a good case using peer to peer BLE can be very useful and improve the performance of an IPS especially when the other positioning systems (such as the geomagnetic map) have problems. Badly functioning devices could have significantly improved performance if they could be helped by better devices around them.

The collaborative BLE data can only be as good as the positioning algorithm output used as the reference positions. For the collaborative system to be robust there has to be a reliable way to detect when a device does not really know the correct position. Otherwise one bad positioning session can seriously hinder other better sessions. It would be possible to use pairwise collaboration and only correct the session estimated to be worse but this would not completely remove the problem when the algorithm doesn't realize it has the wrong position. Also if there are device models known to produce unreliable results with the IPS it would be advisable not to use those as the beacons in a collaborative system.

How exactly the reference position uncertainty is handled is another optimizing task. In this thesis the robustness was weighted over accuracy and the reference position uncertainty was handled as an additive component proportional to the square of the actual algorithm uncertainty. This approach automatically makes sure that when the algorithm has not converged to a single clear position the device's transmitted beacon signal has so large an uncertainty that it will not affect anything.

The possibility to test the system was severely limited in the context of this thesis so the results are not fully conclusive. The effect of the collaboration also depends on how the other parts of the positioning algorithm work so better tests should be conducted as the IPS is developed further. Also it would be good to test the system with an actual ground truth path defined so accuracy metrics could be calculated.

Even with the unreliability of positioning based on other sessions the BLE collaboration would be easy to use to help the iOS convergence problem in environments with no static beacons placed. At least it would find the right building and floor with reasonable accuracy. As noted before it would be unlikely to help most of the sessions but a significant number would find improvement especially in the most popular venues.

Suggestions for Future Work

The results presented in this work are first experiments with this kind of collaborative positioning system. The smartphone BLE transmission is so novel a technology that there are no extensive studies about it. This is why much larger scale tests are required before anything really conclusive can be said about how this kind of system would affect the indoor positioning experience. Furthermore the collaborative experiments in this thesis used only two phones. It would be interesting to see what kind of results would multiple phones produce.

All the phones used in this work were expensive high end smartphones. The Bluetooth radio is such a basic component that it should be found on most lower

end phones too. However it would be good to test if there are systematic differences in BLE transmissions between phones. The system would not be very useful if it only worked with the most high end equipment.

The next step in studying the BLE collaboration would be to design and implement the transmission and recording functionality for a large scale system to gather data about how much this kind of collaboration actually happens and run offline simulations about how using these measurements would actually affect the positioning sessions. This way the algorithm parameters could be tuned for actual data before affecting the user experience.

The first steps in actually using the BLE collaboration in a geomagnetic IPS could be to only use it in finding the right initial area in global scale and then use it when the magnetometer calibration quality is low and the measurements thus unusable or to help convergence when the algorithm uncertainty is very large.

There might also be other use cases for device to device BLE. It is effectively a user proximity detection system so the IPS might offer a service to very reliably find which users of an application are close by regardless of their actual positions. That could be useful for for example in social media or augmented reality services.

References

Agarwal, N., Basch, J., Beckmann, P., Bharti, P., Bloebaum, S., Casadei, S., Chou, A., Enge, P., Fong, W., Hathi, N., et al. (2002). Algorithms for gps operation indoors and downtown. *GPS solutions*, 6(3):149–160.

Anastasi, G., Bandelloni, R., Conti, M., Delmastro, F., Gregori, E., and Mainetto, G. (2003). Experimenting an indoor bluetooth-based positioning service. In *Proceedings of 23rd International Conference on Distributed Computing Systems Workshops.*, pages 480–483.

Baccou, P., Jouvencel, B., Creuze, V., and Rabaud, C. (2001). Cooperative positioning and navigation for multiple auv operations. In *Oceans. MTS/IEEE Conference and Exhibition*, volume 3, pages 1816–1821. IEEE.

Bolliger, P. (2008). Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 55–60, New York, NY, USA. ACM.

Chan, L.-w., Chiang, J.-r., Chen, Y.-c., Ke, C.-n., Hsu, J., and Chu, H.-h. (2006). *Collaborative localization: Enhancing wifi-based position estimation with neighborhood links in clusters*, pages 50–66. Springer.

Chung, H., Ojeda, L., and Borenstein, J. (2001). Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope. *IEEE Transactions on Robotics and Automation*, 17(1):80–84.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dijkstra, J., Jessurun, J., and Timmermans, H. J. (2001a). A multi-agent cellular automata model of pedestrian movement. *Pedestrian and Evacuation Dynamics*, pages 173–181.

Dijkstra, J., Timmermans, H. J., and Jessurun, A. (2001b). A multi-agent cellular automata system for visualising simulated pedestrian activity. In *Theory and Practical Issues on Cellular Automata*, pages 29–36. Springer.

Evennou, F., Marx, F., and Novakov, E. (2005). Map-aided indoor mobile positioning system using particle filter. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 4, pages 2490–2494 Vol. 4.

Fang, L., Antsaklis, P. J., Montestruque, L. A., McMickell, M. B., Lemmon, M., Sun, Y., Fang, H., Koutroulis, I., Haenggi, M., Xie, M., and Xie, X. (2005). Design of a wireless assisted pedestrian dead reckoning system - the navmote experience. *IEEE Transactions on Instrumentation and Measurement*, 54(6):2342–2358.

Faragher, R. and Harle, R. (2014). An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+'14)*.

Feldmann, S., Kyamakya, K., Zapater, A., and Lue, Z. (2003). An indoor bluetooth-based positioning system: Concept, implementation and experimental evaluation. In *International Conference on Wireless Networks*, pages 109–113.

Fink, J., Michael, N., Kushleyev, A., and Kumar, V. (2009). Experimental characterization of radio signal propagation in indoor environments with application to estimation and control. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2834–2839.

Fox, D., Hightower, J., Liao, L., Schulz, D., and Borriello, G. (2003). Bayesian filtering for location estimation. *IEEE Pervasive Computing*, (3):24–33.

Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications*, 25(6):38–46.

Günther, A. and Hoene, C. (2005). Measuring round trip times to determine the distance between wlan nodes. In *Networking 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, pages 768–779. Springer.

Gustafsson, F. (2010). Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82.

Hallberg, J., Nilsson, M., and Synnes, K. (2003). Positioning with bluetooth. In *10th International Conference on Telecommunications.*, volume 2, pages 954–958 vol.2.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer.

Haverinen, J. and Kemppainen, A. (2009). Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035. 5th International Conference on Computational Intelligence, Robotics and Autonomous Systems (5th CIRAS).

Helbing, D. and Molnár, P. (1995). Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286.

Hofner, C. and Schmidt, G. (1994). Path planning and guidance techniques for an autonomous mobile cleaning robot. In *International Conference on Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI*, volume 1, pages 610–617. IEEE.

Iwase, T. and Shibasaki, R. (2013). Infra-free indoor positioning using only smartphone sensors. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

Kurazume, R. Y., Nagata, S., and Hirose, S. (1994). Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation.*, pages 1250–1257. IEEE.

Larranaga, J., Muguira, L., Lopez-Garde, J. M., and Vazquez, J. I. (2010). An environment adaptive zigbee-based indoor positioning algorithm. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8.

Lee, K. C., Oka, A., Pollakis, E., and Lampe, L. (2010). A comparison between unscented kalman filtering and particle filtering for rssi-based tracking. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 157–163.

Madhavapeddy, A. and Tse, A. (2005). *Proceedings in UbiComp 2005: Ubiquitous Computing: 7th International Conference.*, chapter A Study of Bluetooth Propagation Using Accurate Indoor Location Mapping, pages 105–122. Springer Berlin Heidelberg.

Morgan, E., Shean, M., Alizadeh-Shabdiz, F., and Jones, R. (2009). Continuous data optimization by filtering and positioning systems. US Patent 7,474,897.

Rodas, J., Escudero, C. J., and Iglesia, D. I. (2008). Bayesian filtering for a bluetooth positioning system. In *2008 IEEE International Symposium on Wireless Communication Systems*, pages 618–622.

Särkkä, S. (2013). *Bayesian filtering and smoothing.* Cambridge University Press.

Schweizer, C., Edwards, R. D., Bayer-Oglesby, L., Gauderman, W. J., Ilacqua, V., Jantunen, M. J., Lai, H. K., Nieuwenhuijsen, M., and Künzli, N. (2007). Indoor time–microenvironment–activity patterns in seven regions of europe. *Journal of Exposure Science and Environmental Epidemiology*, 17(2):170–181.

Solin, A., Särkkä, S., Kannala, J., and Rahtu, E. (2016). Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning. In *Proceedings of the European Navigation Conference.* Accepted for publication.

Strömbäck, P., Rantakokko, J., Wirkander, S.-L., Alexandersson, M., Fors, K., Skog, I., and Händel, P. (2010). Foot-mounted inertial navigation and cooperative sensor fusion for indoor positioning. In *ION International Technical Meeting (ITM).*

Särkkä, S., Tolvanen, V., Kannala, J., and Rahtu, E. (2015). Adaptive kalman filtering and smoothing for gravitation tracking in mobile systems. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7.

Wardhana, N. M., Johan, H., and Seah, H. S. (2013). Enhanced waypoint graph for surface and volumetric path planning in virtual worlds. *The Visual Computer*, 29(10):1051–1062.

Watson, R., Lachapelle, G., Klukas, R., Turunen, S., Pietilä, S., and Halivaara, I. (2005). Investigating gps signals indoors with extreme high-sensitivity detection techniques. *Navigation*, 52(4):199–213.

Zandbergen, P. A. (2009). Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning. *Transactions in GIS*, 13(s1):5–25.