The University of York Department of Computer Science

Submitted in part fulfilment for the degree of MEng.

Towards a Building Modelling Framework: Heuristic-EZ

Martin Higgs

18th March 2016

Supervisor: Iain Bate

Number of words = TODO, as counted by wc -w.
This includes the body of the report only.

Abstract

TODO

Contents

1 Introduction

The aim of this project is to use data from sources such as phones to track where individual people are in building and then this information can be used to learn a model of how the building itself is being used. For instance, models of typical movement patterns, congregation areas etc can be formed. These models can be improved, or cross referenced for verification, with other sources of information, e.g. timetable information, that would suggest who should be in which rooms at what times. The models can then be used for many reasons including optimising the energy usage within a building, e.g. only heat particular rooms shortly before the room(s) is to be used and switch off heating when there is no one in the room(s).

The phases of this project are as follows:

1. Determine and choose from the available sources of information. Where needed augment this with bespoke mobile applications
2. Investigate how classical machine learning approaches can help derive the necessary models
3. Derive models from the information obtained
4. Evaluate the quality of the models using other sources of information

The outputs of this project are useful data sets, software for deriving models from the information and a means of evaluating the quality of the models.

The main challenges of this project are:

- Determining which sources of information can easily be used
- Deriving means of determining whether a model is good
- Producing useful models from unreliable information from a diverse set of sources
- Determining credible independent sources of information to compare the information against

Interesting ethical point: I believe most smartphones (Android, iOS and Windows) request to send visible access points back to a central server for mapping availability. By developing this technique could I be inadvertantly mapping every building in the world (private and public alike) with the sheer volume of smartphones available? This is good for things like emergency services, but where do I draw the line?!

Conclude with a breakdown of what the following sections (chapters at the moment) aim to achieve.

Estimated length Breakdown (number of words for BSc):

- Abstract, ethics statement, etc. - 500 (0)
- Introduction - 1,000 (0)

- Literature review - 3,000 (4,104)
- Problem description/analysis - 1,500 (0)
- Design and implementation - 2,500 (800)
- Results and evaluation - 2,500 (0)
- Conclusion - 1,000 (0)

2 Related Work

This chapter conducts a review of various works surrounding the project's goals. First it will address the motivations and support the relevance of this study (Section 2.1). Concluding that it will provide coverage of the constituent parts relevant to the work on crowd-sourcing the data to be used, indoor positioning systems for creating a location map of the data and finally the automated generation of floor plans for analysis of the building in sections 2.2, 2.3 and 2.4 respectively.

2.1 Motivations

Modelling a building by various methods for automation and analysis can provide a multitude of benefits through better understanding of its use from emergency situation handling to day-to-day improvements. For example Gao and Whitehouse [1] provide an extremely simple method for intelligently setting the on/off times for central heating, increasing energy efficiency while minimising user discomfort. The method is built around using smart-home sensors (such as reed switches on doors or motion detectors) to determine the latest leaving (to work) and earliest entrance (from work) times and incorporating these with a user-specified amount of "miss time" when the house is occupied but the house is not already warm. As the technique is solid, with the prevalence of WiFi enabled personal devices, it could easily be adapted to monitor user's devices leaving / moving around the WLAN and scaled to larger buildings with finer grain control over their heating.

By giving the system control over a user's environment however, it enters into a category of "physical computing systems" as identified by Stankovic et al. [2]. While this is a very simple definition of any system that can take in data through some collection of sensors (e.g. a Wireless Sensor Network) and act upon that data through some actuators, one must be aware of the responsibility placed upon the system. Most of the concerns raised are context sensitive, but when designing such a technique it can help to consider these situations in order to identify its limitations. Examples of such systems might include the use of WiFi signal scattering to detect and alert of falls in vulnerable homes [3], provision of city-wide traffic management through a wireless network of cameras and signals [4] or asset and personnel tracking across a wirelessly networked environment [5].

The idea of leveraging wireless networks and the multitude of sensor-enabled devices that occupy them however has been around for quite some time. Recently Torres-Sospedra et al. [6] identified the specific trend for using such devices to provide Indoor Positioning / Localisation Systems (IPS), but no definitive way of comparing techniques applied and so created the UJIIndoorLoc database. We will expand upon this data set later, but for now it serves to show the collaborative effort going into solving the problem of indoor localisation.

2.2 Crowd-sourcing Data

With the sheer variety of sensors available in modern personal devices carried by people of all backgrounds almost everywhere, sourcing as much data as possible would seem to be relatively easy. However, when using this technique the data collection must always attempt to be as un-invasive as possible.

An early endeavour into this field was CenceMe, a classification program designed to relay information about the user to their friends through Facebook [7]. The program ran on the Nokia N95 passively collecting data whenever the phone was interacted with (i.e. through button presses) which were then translated through a simple set of classifications into facts to pass on to the server for processing to Facebook. Collected data included GPS location, microphone noise, accelerometer data, Bluetooth device visibility and random pictures for determining where the user was, whether they were having a conversation, their activity level, who they were with and verification purposes respectively. Initially, the user needed to define people and places significant to them for the classifier to recognise but after this, CenceMe only needed prompt the user for input if it recognised a place the user visited frequently, in order for them to add it as a location to notify their friends about.

Concerns raised during this study included those of privacy, portability, ease of development and power usage, most of which have been addressed by advances in mobile operating systems such as Android and iOS. When developing for the N95 the CenceMe team reported very limiting factors designing their framework including low memory, computational power and general programmability, all of which are not of concern today with multi-core, multi-gigabyte RAM devices running almost complete editions of Java with expansive APIs.

However, with increased capability comes increased potential load on the device's power supply (high-speed cellular internet access and HD video processing are particular culprits), limiting its availability; as such manufacturers, OS providers and researchers alike are continually looking into power saving methods. One such team of researchers looked into a technique they called "Piggyback Crowd-Sensing" [8]. This technique centres around the multi-functional abilities of the devices carried today, for example the GPS sensor may be used for geo-tagging pictures, navigation or social media posts. This provides a crowd-sensing application with a variety of opportunities to utilise the GPS without powering up and winding down the sensor specifically to take measurements; 'piggybacking' off of another applications sensor usage.

The problem with this technique arises with the need to narrow down the window of when / where sensor readings are taken as user application usage can be quite sporadic. The research team then complemented this method with a "Sensing Decision Engine" (prediction algorithm) that gathers data on which sensor-using apps were launched in which time windows (6 hour slots of each day), whether it was the weekend or a weekday when it was called up and which square kilometre it was used in. Based on these data, the algorithm is then able to predict whether a sensor is likely to be used in the current situation and so whether it should wait for the sensor to come into use or to power it up on its own. This technique as a whole was shown to save up to 90% of the energy required to perform crowd-sensing for a small cost to accuracy of readings. Successes in approaches like these have lead to adoptions at an operating system level, such as the Android "Sensor Batching" approach [9].

The location required for this sensing decision engine is intentionally broad as very low accuracy location estimates can be made through triangulation from nearby cellular signal towers. Gathered data however can require a much finer grain location estimate to associate with in order to make it meaningful.

2.3 Indoor Positioning Systems (IPS)

Outdoor positioning sensors are commonplace in most devices today with GPS being the dominant technique as a device can calculate its absolute position using only the visible satellites for trilateration. Indoor positioning however often requires a much finer level of precision and with GPS being largely invisible from indoors the problem is widely regarded as being unsolved such to the point that competitions are being regularly held to encourage new techniques [10]. These can be split into the infrastructured and infrastructureless categories where the former requires deployment of additional equipment around the environment (such as motion sensors or microphones) and the latter should not with the sole exception being the assumed existence of Wireless Access Points (APs).

As a fundamental component of analysing what rooms are in use when is knowing where users are at a given time, this section will cover a variety of IPS techniques in detail. As such each technique will be judged by its pre-deployment effort (cost of extra infrastructure, measurement taking, computational cost etc.), level of prior knowledge required of the building and on-line (active) location accuracy.

2.3.1 Early Techniques

One of the earliest infrastructured approaches was the Active Badge Location System wherein personnel badges were equipped with infrared emitters sending uniquely identifying signals to be picked up by detectors set up in every room [11]. While the set-up proved extensible (up to 128 badges monitored at once with low conflict rate) and flexible (more sensors can be added / moved with the building) with cheap sensors and badges it still required detectors being wired into every room and as such could only provide room-level accuracy at best and only a vague idea of where each person had been in areas not covered by detectors (such as corridors).

The team also appears to cite "radio signals that can penetrate the partitions found in office buildings" as a negative effect as their technique utilises proximity as a binary indicator of where a person is. However, while the paper may have served as an inspiration to create proximity-based positioning techniques, many more modern methods have decided to use the radios in Wireless APs as a single AP can cover a much larger space than an IR detector. This raises additional challenges though in dealing with how radio signals propagate, as the strength of a radio signal can be mangled by the distance it's travelled, the materials it's passed through (be it air, wood, jeans or people) and by waves themselves (electromagnetic waves can cause destructive and constructive interference on their own frequency, be they reflected or just transmitted from another source)! Given all of these problems Whitehouse et al. [12] evaluated the use of received Radio Signal Strength Indicators (RSSI) as a sole measure to judge distance. Their results confirmed that even in ideal outdoor situations with few obstructions "small differences in vegetation such as the height of grass can have large effects on RSSI" and "Experiments in an indoor environment revealed no discernible pattern in RSSI, even in a large room with no walls and at the very lowest transmission power.".

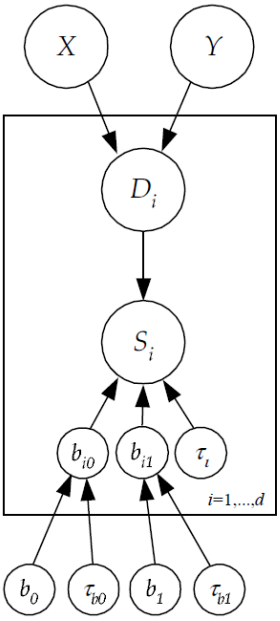
Despite this potential for wildly fluctuating RSSI based on the environment Bahl and Padmanabhan [13] were able to create a mathematical model for the RSSI based on the floor plan of the building in question; RADAR, one of the first infrastructureless IPS techniques.

RADAR is built around their Wall Attenuation Factor (WAF) model:

$$P(d) = P(d_0) - 10nlog\left(\frac{d}{d_0}\right) - \begin{cases} nW * WAF & nW < C \\ C * WAF & nW \geq C \end{cases} \tag{2.1}$$

This model is a simple combination of the standard signal path loss model and a linear drop in power for each wall that the signal travels through. To break it down, $P(d)$ is the expected RSSI (in dBm) at distance d, where d_0 is a distance of 0m or the access point's 'Transmit Power'. This signal travelling through the air then decays logarithmically according to an unknown but fixed rate n. Finally a fixed amount of power (WAF) is lost for each wall (nW) up to a pre-determined cap (C) beyond which any extra walls make no difference as the signal is pretty much unidentifiable anyway. This model is then applied to approximate the RSSI of every access point at every position in the building (using the floor plan as a guide) the create an RSSI map of the building in question. Location is then a relatively simple task of matching up the user's visible RSSI from each access point and matching them to the closest approximate location. This technique was created to as an alternative to taking 'empirical' measurements throughout the building to create the map by hand and while it is not as strong as taking empirical measurements (roughly 30% increase in median error) it saves a significant amount of effort for such a simple model.

Figure 2.1: Model M_2 from Madigan et al. [14].



Madigan et al. [14] instead utilised a graphical model to embody the relationship between position and RSSI (Figure 2.1). In their Bayesian Network, for each access point i the distance D_i, transmit power b_{i0}, path loss rate b_{i1} and noise τ_i are all conditionally dependant given the RSSI observed S_i. This signifies that given a set of these variables, the model can infer the most likely values for any connected values in the graph (for example, training data would give the X and Y co-ordinates and the signal strength, which can be used to infer all other variables given the graph's complete connected nature). Then during the on-line positioning phase, the user's most likely position is inferred from the AP variables established during the training phase

and the observed RSSI values. Using this model they are able to estimate position assuming only the degradation of RSSI is roughly logarithmic and that all access points have similar behaviours (b_0 and b_1 are the global variables around which each specific access point's values are estimated). This method does require training data in order to calculate the maximum likelihood access point parameters, but with a very small number of training samples (around 20) and no other knowledge of the building they show that the model is able to estimate the user location with a ~50% increase in median error compared to RADAR.

While providing insights into the area under observation with little prior knowledge, these techniques are 'one-shot': a model is built from known values and then followed permanently thereafter. However, should the space be modified at any point after the model is created, the solution would effectively need re-deploying to continue providing accurate estimates.

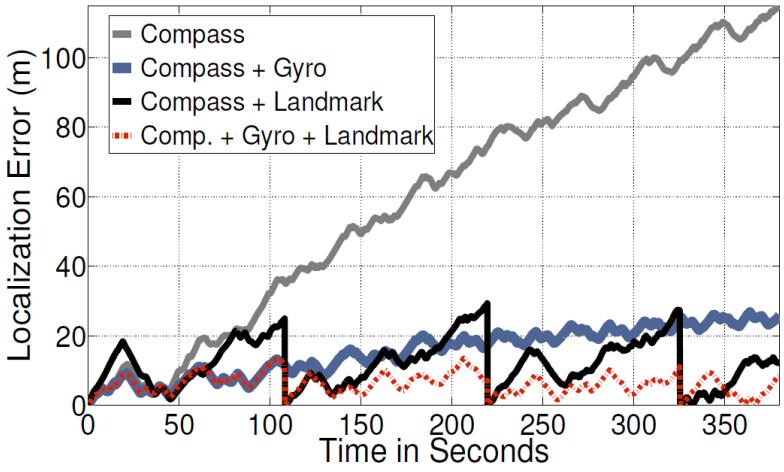
2.3.2 Adaptive Indoor Positioning Systems

All of the previous techniques discussed have had two very distinct phases: off-line and on-line. During the off-line training phase, data is gathered with ground truths (guaranteed values such as user verified location) or models are calculated and then in the on-line positioning phase the observed RSSI values are then used in the models to determine the user's most likely location. This section however will explore techniques that can improve gradually over time as more users traverse the space.

Techniques covered to this point rely on electromagnetic waves and the detection thereof to paint a picture of the user's current location, however Wang et al. [15] combined dead-reckoning (estimating relative location by tracking movement from a known point) and a broader range of sensors than most other techniques to estimate location. Traditional dead-reckoning is known to be relatively accurate at first, but random errors gradually accumulate and the estimation can become wildly inaccurate the further travelled from the starting point. This technique however relies on organically generated 'landmarks' to help reset the accuracy of a user's current position as frequently as possible (Figure 2.2). These landmarks can range from elevators detected by accelerometer spikes to metallic areas detected by the magnetometer, each uniquely identified by the RSSI values surrounding the area. As the mean of the error generated by dead-reckoning is theoretically 0, a landmark can be placed at the average location of all positions it has been identified at making exact positioning when the landmark is detected possible. While accurate to a reported 1.2m with no prior knowledge of the space and no deployment necessary, this positioning method must be permanently active from each landmark or the user will be unable to locate themselves. Also, even with techniques identified to only use the most effective landmarks this would still put a relatively heavy weight on the user's power supply through its always-on nature and the variety of sensors required compared to conventional RSSI measurement.

Such conventional methods had also been gradually improved upon to provide sub-metre levels of accuracy using by combining crowd-sourced data with probabilistic reasoning through a technique called 'Horus' [16]. The first step (and largest weakness) of the Horus system is the necessary collection of sample data from known locations to create an initial RSSI map. Data collected by users while positioning however now continues to feed back into the initial RSSI map as the team discovered that RSSI values from each AP are not independent with regards to time, so utilise an auto-correlation co-efficient (α) to help isolate useful measurements. This co-efficient defines how similar readings in an area have been and augments the expected distribution of RSSI values for that area to allow for a much larger range if all observations have been very similar.

Figure 2.2: Dead-reckoning accuracy found through experimentation by Wang et al. [15].



Using this modified RSSI map, positioning is then broken down into multiple steps; first the user's 'discrete' area must be determined as the most likely position, given all of the access points that are currently visible. From here, the system uses a function of both the N most likely locations weighted according to their normalised probabilities and the average signal strengths over a small time window to smooth out the estimate. Finally, the system enables tracking of small-scale variations by noting that a user's location cannot change more than their speed would allow. By this notion, if the user appears to have moved more than a threshold amount more than their previous movement, a small-scale error is assumed so the system tries an estimation for all received AP RSSIs multiplied by 1, $1 + d$ and $1 - d$ to find the highest likelihood and associated location.

The final technique this paper will cover requiring no prior knowledge of the space, extra deployments or specific training set is Microsoft's 'EZ' localisation system [17]. The technique is centred around solving simultaneous equations to provide accurate estimates, similar to Madigan et al. [14] in looking for its parameters. However, the technique uses occasional measurements paired with GPS readings and clustering methods to reduce the search space to the most useful subset.

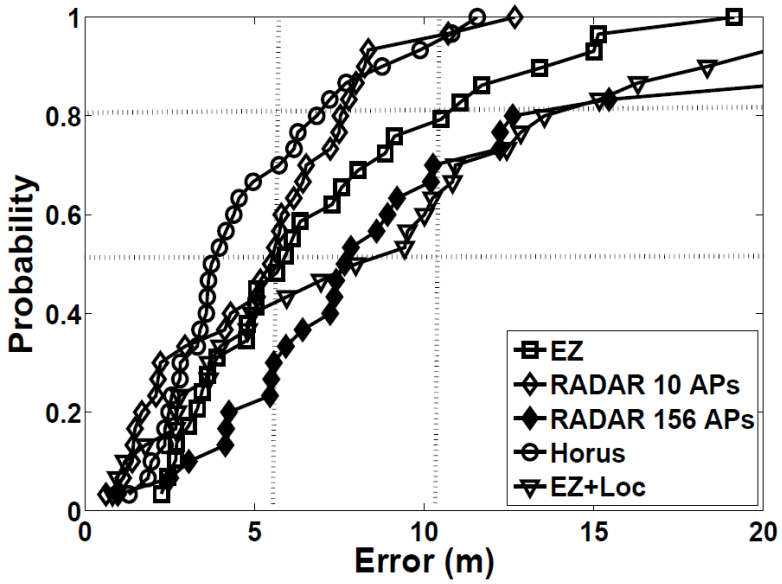
$$d_{ij} = 10^{\left(\frac{P_i - p_{ij}}{10\gamma_i}\right)} \tag{2.2}$$

For each access point (i), the system attempts to solve for its X and Y location, transmit power (P_i) and path loss rate (γ_i). If the access point is visible from at least 5 locations where GPS lock was established (j), then these parameters can be uniquely identified through solving simultaneous equations (Equation 2.2) for RSSI values (p_{ij}) from which the access point was seen and then using the gathered distances (d_{ij}) to trilaterate the 2D location of the AP. Solving even just one AP's parameters can cause a domino effect leading to more uniquely identifiable locations, solving more AP parameters and so on. Eventually though the dominoes stop and the system has to guess some parameters in order to continue and the search space can be incredibly large. To counter this, the EZ team created APSelect and LocSelect, two clustering algorithms that group access points and unknown locations into a single point if the RSSI values observed overlap ~90%, thereby reducing the number of unknown variables the system has to solve for, taking only a minimal accuracy loss in the process.

The space is then searched through to find a set of parameters with the minimum error using a combination of genetic searching and gradient descent. To start with, a set of solutions with completely random parameters are selected and their fitness (mean absolute error) evaluated. From here, the top 10% are kept unmodified, 10% are randomly re-generated, 60% are made as a combination of 2 random parents from the last generation (and enhancing them through gradient descent) and the last 20% are made by taking a single random solution from the last generation and tweaking each parameter by a random amount (from an exponential distribution to allow for occasional large changes). This combination of solution generation techniques avoids the large number of potential global minima that come with so many parameters while exploring the state space relatively quickly.

One final optimisation applied to their method is accommodation for relative gain. By identifying proximate locations in which measurements were taken by two different devices, their average RSSI values can be compared to find their approximate relative gain difference and normalise their RSSI values for use in simultaneous equations. With all of these techniques combined, EZ was able to perform comparably to RADAR on larger scale deployments without all of the pre-deployment effort and the search for optimal parameters could be re-run as the space being localised changes and more data comes in. The main disadvantages with this technique come simply from its marginally lower accuracy than techniques like Horus (Figure 2.3) and the relatively high computational effort required to implement the technique and compute the results.

Figure 2.3: Cumulative distribution function of distance error comparing Horus, RADAR and EZ IPS techniques [17].



Following the rapid development of IPS techniques Torres-Sospedra et al. [6] identified the need for a definitive data set to compare them through and created the UJIIndoorLoc database. The data set comprises of a large number of user-verified co-ordinates and their respective RSSI values to every WLAN access point being monitored. As these measurements were carried out by a number of users (integrating height as a differentiating factor) with a

number of devices this provides great ecological validity to the data set while still delivering reliable measurements. The validation set was also generated without guide markers, ensuring that data different from the training set was provided. With such a comprehensive data set it is then easy to emulate limiting factors of any data gathering technique, such as noise or poor coverage, and due to high demand the team also provided a map of the locations [18]. However, data points extraneous to the buildings in question would have been useful to provide a complete picture of a user's day both inside and outside of the buildings.

2.4 Automated Floor Plans

Floor plans of a building are often key to enabling detailed location-based analysis by providing boundary information and special characteristics of the space, but may not be available prior to a system's deployment. It is easy to see how a special purpose robot might use a combination of dead-reckoning and obstacle sensors to map a space using an exhaustive search, but the cost of the equipment and the size of the area being surveyed could easily render the technique infeasible.

Other techniques for building floor plans rely on gathering a small amount of pre-requisite data and building upon the constraints that it gives. A technique for determining layout on a small (single floor house) scale is described by Lu and Whitehouse [19] for use on "Smart Homes". Smart homes are buildings with a notion towards autonomous control, such as turning on the lights in the hallway before the user physically enters the space, detected through a collection of sensors placed about the home. Here, it is required to have motion detectors placed on both sides of every door (to determine room connectedness), a magnetometer (to determine door orientation) attached to these pairs of motion detectors and light level sensors on every window (facing inside and out, to determine orientation through sunrise / sunset). While this is a very high deployment cost through the level of user involvement, the deployed sensors are then anticipated to be in continuous use as part of the smart home itself.

By monitoring the sensors to see which sets fire simultaneously, they can then be clustered into rooms to determine how many doors and windows each room has, and which rooms they are connected to. With the room's interconnectedness established, the technique then minimises the space of potential arrangements using a set of simple heuristics (i.e. windows tend to be on external walls and buildings have as few corners as possible). Despite all of these restrictions, the technique was only able to minimise the set to 2 or 3 potential layouts from which a user must select. As this technique was designed to prevent having a user draw their own plan it satisfies its purpose, but does not provide a detailed enough picture for fine-grain analysis or larger areas of observation.

A more general approach to generating floor plans based on a set of constraints however comes from Charman [20]. This method mathematically defines a set of rooms based on their rotation, reference point and dimensions and attempts to fit them into a given space by connecting their corners. As the rooms are placed, a set of constraints are then evaluated and back-tracked upon where necessary (e.g. Bedroom #1 must be to the right of Bathroom #2). By exhaustively generating solutions, the system is able to find all possible layouts of rooms, given the pieces of data / constraints supplied. While successful at producing floor plans for traditional housing layouts, the method could fall short with modern architecture that often contains complex shapes (not squares) and unidentified empty spaces.

2.5 Summary

This chapter has now broken down the work surrounding building analysis into the data gathering, data localisation and floor plan generation steps and provided a variety of techniques that are applicable at each stage. When gathering data from users it has been shown that techniques exist to reduce the load placed on users and their devices, potentially increasing the amount of data that would be gathered in a crowd-sourced technique through more opted-in users. While users are of the utmost importance when performing any crowd-based operations, this report focuses mainly on applications of the data once collected.

When localising gathered data, a theme has emerged centring around the use of RSSI to estimate distances from access points. Despite its complex nature, the widespread use and coverage of wireless networks combined with the success of simple models make it an attractive choice for IPS. Techniques explored here covered the fundamentals of modelling radio signal propagation, then demonstrated more advanced and novel techniques inferring unknown parameters through machine learning for ease of deployment and increased accuracy alike.

Finally sets of constraints were identified for generating possible floor plans based on estimated dimensions. While restrictive in their application to walls in the four cardinal directions, this fits common traditional layouts of buildings and it is anticipated that more complex arrangements could be broken down into rectangular subspaces to build in. Through combinations of the described techniques this report will now explore ways in which a system might determine room usage from a set of simple crowd-sourced data.

3 Problem Analysis

Potential re-structure if you have time: Move advantages / disadvantages (critique) of each method to lit review, then you can make this section very brief, stating the main reason why you can't use it.

Break down goal of "begin framework" into the 4 stated in intro and discuss.

This chapter will explore the problem space involved with attaining the goals of the project. Primarily, this project is tasked with establishing a framework around which building analytic techniques might be developed or applied. This could include the development of any combination of the techniques covered in chapter 2 concerning crowd-sourcing the data, localising the data and modelling the space itself.

Developing a complete system however might prove intangible for a project of this size as the system would require implementation across multiple platforms and likely multiple languages, carrying with it the interoperability complications to navigate and technical difficulties as opposed to research challenges. Furthermore spreading implementation time so thinly might also prevent any deeper evaluation as it would be hard to compare systems at such a high level. As such it would be wiser to focus on implementing a single component around which a system can be built.

The first component to consider would be a modern crowd-sourcing tool-kit for specifying, gathering and collating some selection of data from sensors at minimal impact to the users. While implementing an application in Java using the techniques described in section 2.2 to enable gathering from a wide selection of Android devices would be beneficial to build from, it is known that a specific study into the factors surrounding crowd-sourcing data is running concurrently to this project [21]. Floor plan modelling using location data to feed a constraint solver such as that described by Charman [20] could also be an effective component. However, compiling such a complete set of data without implementing either of the previous components of the system would be very costly for such a small team, given the number of man hours it would take to exhaustively map out a significant area. Therefore, the most useful component to create as part of this project would be an Indoor Positioning System around which other techniques may be developed. As demonstrated by Torres-Sospedra et al. [6], there exists a high demand for open implementations of such components and thanks to the existence of such sample data sets, the cost of testing and comparing such techniques are greatly reduced.

As explored in section 2.3, there exist a multitude of techniques for localising users in an indoor space with varying degrees of accuracy. The goals of this project however, limit the range of applicable techniques as the model should be created from readily available sources of information. While this immediately rules out any infrastructured techniques as the installation of extra equipment severely limits the domain space that can be modelled, this project will also avoid those that require a large set of ground-truth, domain-wide test data as they are again limited in their application.With these restrictions in place the most interesting systems to study would be: RADAR [13], landmarked dead-reckoning [15] and EZ [17] from those evaluated in chapter 2.

RADAR provides one of the highest levels of accuracy of all reviewed techniques for relatively low localisation and model-building cost as it is all based around a pre-existing floor

plan. Using this floor plan, the technique estimates the RSSI of every AP by using a simple formula modelling the signal's attenuation through walls and across space. In situations where detailed floor plans are already available, using this method could be the most efficient means for determining building usage metrics as the model building process is skipped entirely and user positions can be determined immediately. However, floor plans are not always complete, universally accessible nor consistent with flexible spaces such as exhibition centres. Because of this limitation in combination with the lack of information generated by skipping the modelling process, this technique is not useful for the project.

Landmarked dead-reckoning appears to rectify most of these concerns as the technique requires no previous knowledge of the space (by dead-reckoning from a known point) and creates its map of landmarks that are potentially much less dependent on the flexible properties of the space (such as elevators and stairs). This enables the modelled space to change over time while preserving a relatively high level of relative location accuracy. Therein however lies the problem with using this technique to measure space utilisation as it can only detect relative movement as opposed to independent single-point localisation. This is not to say that the technique would be difficult to gather sample data as Torres-Sospedra et al. [22] provides the full complement of magnetometer, accelerometer and (in conjunction with future data to be provided by Torres-Sospedra [23]) RSSI readings necessary to implement such a system. The worry is more about the load placed on prospective user's devices and how to gather meaningful results without the storage requirements becoming infeasible. As shown in the UJIIndoorLoc-Mag database, the ~40,000 data points only correspond to ~300 short relative motion traces so it is not unimaginable for the data sets to quickly get out of hand, especially if a user were to store multiple day's worth of data before a storage opportunity arises. Solutions to these problems could involve modifying the sampling frequency to reduce the load on the user's device and the amount of data held for analysis, but it is unknown as to how this would affect the accuracy of the model due to the precise nature of the landmarks. Also, the test data and initial research do not cover the prospect of spending a large period of time walking around the same space with no significant landmarks. In these cases, the behaviour would return to basic dead-reckoning and continually stack up inaccuracies, potentially invalidating the data. For these issues in combination, landmarked dead-reckoning does not significantly work in the project's favour.

EZ however, implements a system requiring a very simple set of independent client-side measurements backed up by a server-side model that can adapt to a changing space as the set of observations grows, without human intervention (such as specifying a new floor-plan like RADAR). With this technique, RSSI measurements are provided along with GPS co-ordinates (where possible) to estimate the relative location and signal transmission properties of the access points being observed. This simplicity of the data required to build the model along with that of the model itself also makes it useful for research into building analytics as it reduces the effort required to crowd-source the data in the first place and reduces the complexity of any floor plan model wishing to reinterpret the EZ data.

Assuming that the crowd-sourcing programme or test data input provides a reading of all RSSIs available (as already required by a wide range of infrastructureless indoor positioning systems) alongside the occasional GPS location, EZ will be able to provide a model for localising those without position data and any future measurements made. Furthermore, assuming that the floor plan modelling technique is also able to make use of positioning data, the output of running EZ on the original data set will still be useful regardless of the technique's ability to interpret the AP parameters to some degree. Combining these assumptions with a large set of sample data such as that provided by Torres-Sospedra et al. [6], the EZ localisation component

can then be developed independently of any crowd-sourcing or floor-plan modelling strategies that may follow.

Does this need any more of a wrap up? Are there any points I have missed?

4 Method

This chapter will cover the development of an indoor positioning system from which positioning data and Wireless AP models can be extracted for use as part of a general building model. The model will follow the techniques as specified in the EZ localisation system [17] with measurements from the UJIIndoorLoc data set for testing and evaluation purposes [6]. This chapter will also identify some weaknesses in the specification of EZ and propose the Heuristic-EZ localisation system to compensate.

4.1 System Specification

Main point is that the implementation was not available and the ability to do it in matlab instead is a side note!

The original implementation of EZ by Chintalapudi et al. consisted of "about 7000 lines of C# and Python code" encompassing all of the client-server communications, data gathering and off-line modelling /on-line localisation operations. By separating out the data processing elements of EZ from the rest of the system, this implementation of EZ can take advantage of languages with more powerful built-in mathematical functionality without worrying about handling sensors and external communications. As such this project will pursue an implementation in Matlab 2015b consuming a .CSV file of pre-gathered measurements and outputting two .CSV files containing all localised points and the AP parameters separately. This allows future efforts into crowd-sourcing to take advantage of techniques available to other languages while still piping data into this implementation of EZ. It will also allow floor plan modellers to be selective as to which types of data they wish to use to generate their models, again in their language and platform of choice.

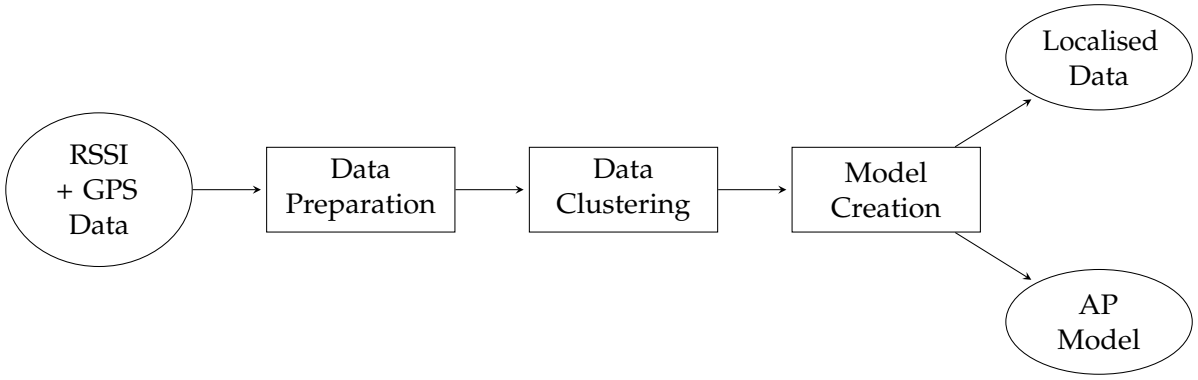


Figure 4.1: EZ Control Flow

EZ's specification then breaks the system down into a set of components, each of which is able to operate independently of the others or chained together for various effects. To this end, the project's implementation of EZ will take a pipe & filter approach allowing each component to affect the data in turn before passing it on to the next (see Figure 4.1). These components

will all operate without knowledge of the other's actions and upon the same data format, such that components may be changed for different versions or removed entirely without significantly impacting the program's flow.

The creation of EZ's model is the first and only mandatory component specified, but also comes as the final piece of any system variant that might be created. Given a set of data this component will attempt to create a model based on the idea that with enough already localised measurements of RSSI from an AP, some basic parameters can be estimated about the object (giving the AP model data). Then if a measurement without GPS data can view enough parameterised APs, then its location can be trilaterated using simultaneous equations (this will be covered in detail in section 4.4).

However, as is also specified, the data can be effectively refined and reduced before it is passed to the model creator through preparation and selection of the most useful pieces of data. The most prominent piece described for preparing data is that of gain estimation as they appeared to find gain differences of up to ~14dB which could significantly distort models generated using that data by making measurements taken at the same location on different devices appear very different by RSSI. To alleviate some of these problems, they propose the Relative Gain Estimation Algorithm (RGEA) which looks for small, consistent differences in RSSI measurements and creates a set of estimated differences in gain across devices. By solving these differences as a set of simultaneous equations, the gain values of each individual device can be estimated and used to correct the input data (covered in full in section 4.2).

Finally covered is the selection of specific data points to use when modelling through the use of hierarchical clustering. This clustering algorithm aims to group data points of 90% or greater similarity and then nominate a representative from each group to be passed to the model. The algorithm is run twice on the same data set, but once to select the most useful Access Points (APSelect) and once to select the most useful measurement locations (LocSelect) for different reasons. APSelect is run to reduce the number of APs modelled as the team observed up to "160 APs across a single office floor", potentially leading to an over-constrained model and excessive modelling time. LocSelect however, is necessary to prevent biases in the model where too many measurements from the same location may reduce accuracy in others.

The ordering of these components is of course fixed as clustering data without correcting for gain could miss potential groupings and correcting the gain of already localised points would lead to inaccuracies. The components can however be avoided entirely (except for modelling) as Chintalapudi et al. experimented with. This project will make comparisons of its implementation to their claims using their '*LARGE*' data set as its size is comparable to a single floor of a building in that of the work by Torres-Sospedra et al. [6].

How to segue from specification to implementation? Crude to just say "with the system now specified the implementation can begin"

4.2 Data Preparation

In order to compensate for artificial noise created by differences in gain across devices measuring RSSI at the same location, the RGEA estimates these differences and applies them to the data set. However, with only a subset of the data having a known location, the RGEA has to use RSSI to guess at 'proximate' locations. The specification states that measurements are deemed proximate if the average difference between RSSIs are less than 3dB. For each pair of devices (k_1, k_2), these pairs of measurements are then gathered into a set $M^{k_1 k_2}$ and compared to get the average difference (Equation 4.1). This relative gain is then plugged into

equation 4.2 to calculate the standard deviation of that relative gain from the difference in all comparable measurements.

$$\Delta G^{k_1 k_2} = \frac{1}{|M^{k_1 k_2}|} \sum_{(p_1, p_2) \in M^{k_1 k_2}} (p_1 - p_2) \tag{4.1}$$

$$\sigma(\Delta G^{k_1 k_2}) = \frac{1}{|M^{k_1 k_2}|} \sqrt{\sum_{(p_1, p_2) \in M^{k_1 k_2}} (p_1 - p_2 - \Delta G^{k_1 k_2})^2} \tag{4.2}$$

Because these relative gains are transitive (i.e. $\Delta G^{k_1 k_3} = \Delta G^{k_1 k_2} + \Delta G^{k_2 k_3}$), any pairs of devices with proximate locations can then be chained together with other device pairs to create a system of simultaneous equations. Finally a random device as part of this system is selected and chosen to have some value of gain so then the system can be solved. As this is an estimate however, the equations are weighted according to their standard deviation or in effect their likelihood of being accurate.

Implementing this short specification however, leads to incorrect results. The main discrepancy is the selection of RSSI readings to include when comparing a pair of measurements. For completeness, the UJIIndoorLoc data set includes every invisible AP as a measurement of 100dB to mark it as an invalid measurement. If all readings (including invisible markers) are compared for a pair of measurements, then in spaces where only a small proportion of APs are visible at any given location, all locations will be deemed proximate as all invisible APs register a difference of 0. The opposite however also causes problems, if only points visible at both locations are compared, then being equidistant from a single AP could cause the whole measurement to be deemed proximate. The solution is to compare all readings visible in either measurement, such if the measurement were to be localised using the set of points, they would be equal. However, including readings in this manner with the UJIIndoorLoc positive invisibility indicator causes any single difference in AP visibility, no matter how weak (i.e. -99dB), to be marked as not proximate in error. The simple fix was to change all 100dB readings to -100dB. Diagrams explaining these situations? Yes.

With the specification implementation producing feasible results, it was at this point that the technique was to be validated against some known values. With no provided gain values for the devices in use by UJIIndoorLoc however, it was necessary to attempt to calculate them separately. To this end another relative gain estimator was created using the ground truth location values specify exactly which pairs of measurements were proximate. The first major change was the addition of a general location translator to the data preparation phase as measurements were gathered within a few hundred metres of each other, but the latitude values started at ~4,865,000m. The other major difference was a factor of speed; as the location comparison only needed to compare the difference between latitude and longitude as opposed to the average difference of up to 520 APs, the ground truth calculator performed Profile RGEA and Ground Truth calculator for ballpark difference (i.e. 2 orders of magnitude faster).

Through experimentation and comparing the RGEA's output with the newly calculated values, several other faults of the implementation came to light. The first was that when calculating the average difference between measurements for inclusion in the set $M^{k_1 k_2}$, this was meant to be the average *absolute* difference being less than 3dB. Errors in light of this fault arose with measurements where readings balanced out with positive and negative differences (such as being opposite sides of a wall). For example the average difference between (-20dB, 20dB) for 2 APs is 0dB, causing the points to be falsely identified as proximate with no relative gain. Diagram?

- Fixed RGEA according to faults discovered found through experimentation (3)
  - Even with restrictions on APs tends to find 1 falsely proximate point, threshold so there is at least a significant overlap (sum(prox) > 10)
  - Changed to system of linear equations, underspecified for number of gains to be solved, but simulated annealing performs far worse (and modifies irrelevant parameters) (recognised gain estimation should be a deterministic process)
  - Relative gains prioritised by `sigma_deltaG`, but this causes exponential increase in priority for very little increase in probability through bell curve.
- Simplified RGEA compares only those where the strongest RSSIs were of the same APs. Massive speed gains as only comparing similar number of parameters to ground truth (also sparse). (2)

4.3 Heirarchical Clustering

- Explain EZ APSelect and LocSelect with goals and equations (1)
- Emphasize necessity specifically with UJIIndoorLoc as measurements with lots of different devices taken at the same location and EZ requires readings at different distances to prevent skew when modelling. (1)
- Original EZ Implementation (2)
  - EZ implementation requires *HUGE* amounts of comparisons because of its fundamentally simple nature (profiling shows > 10,000,000 * 520 comparisons for AP clustering alone!)
  - Renders LocSelect computationally intractible using the UJIIndoorLoc data set
  - Comparing all measurements drastically reduces average, comparing only those seen by both ignores absence of measurement. Must compare measurements seen by either to highlight differences.
  - Sparse matrix of measurements, but very difficult to get speed improvement using normal comparison
- Worth a brief investigation into set clustering techniques? (already pruning on equivalent set discovery) (1)
- Try implementing for raw speed in C called from Matlab? (1)
- Simplified version using similar techniques to simplified RGEA to speed the process along (2)

4.4 Model Creation

Explain main script and purpose of model.

4.4.1 Off-line Learning

- Briefly explain theory behind model; 5 RSSI measurements should be enough to uniquely place the AP and its parameters (steal diagrams and cite proof). Note inaccuracies associated with model (i.e. Wall attenuation from RADAR), but flexibility in AP placement allows it to simulate weakened strength (as proven in their experiments) (1)
- Basic EZ implementation iterates over all APs independently, identifying useful measurements, estimating probable bounds for AP parameters and performing simulated annealing (rather than GA, EXPLAIN) (2)
- 3D Localisation issues (3)
  - Explain why using 2D subspace - simplicity, existing techniques proven effective on these spaces, also allows space scaling (can model all 3 buildings simultaneously or just 1) Mention possibility of 3D!
  - Explain problem with using basic EZ - mapping weak signals from other floors in 2D space causes erroneous AP placements
  - Mention how UJIIndoorLoc doesn't have Ground truth AP parameters, so APs cannot just be pruned
  - Confirm development plans of UJIIndoorLoc (including Library and computer labs) so current heuristic work may be validated in the future
- Thresholding heuristic to split 3D into set of 2D subspaces (does eliminate edge-case measurements which are niche, but powerful in isolating a very particular area of the building such as directly below and AP on the next floor or in the nearest building) (1)
- Also threshold minimum number of measurements for an AP to be included to prevent adding APs that could easily skew the model. Another approach to thresholding could be to only use the strongest measurements at each location. (1)
- Discuss thresholding parameters (number of strongest measurements / threshold level) - could be determined through another GA, but runs the risk of over-fitting. State intention to experiment with parameters as they will affect the localisability of the space. (1)
- Implement ERSGA? - We seemingly have enough to discuss already. Explanation of algorithm would be necessary; allows estimation of APs without enough points already localised through GPS. Would also need to implement and explain AP propagation (i.e. determining an AP's parameters will allow more points to be localised and propagated etc.) (5)

4.4.2 On-line Localisation

- State localisation model; at least 3 *relevant* RSSI measurements to determine location through distance. Note how accuracy and amount of localisable areas changes depending on AP selection. (1)
- Compare simulated annealing and simultaneous equation solver (again not linear, so needs more implementation) (1)
- Simplified model uses only strongest measurements for localisation; avoids conflicting measurements, increases speed (1)

- Comparison with Horus methodology (create fingerprint of area using strongest APs then refine with lesser)? (1)

5 Evaluation

Save actual quantifiable results for this section! Evaluate against requirements / massive discrepancies in method.

- Parameter experimentation
- Use PlotFloor to show difference between models with various pieces of the system included / excluded and parameters tweaked
- Evaluate with noisy and partially localised data
- Independent vs. estimating every APs parameters at once?
- Note EZ specifically says it works best with a large amount of data distributed across the whole area whereas UJIIndoorLoc provides with lots of measurements from the same locations. Will be interesting to see accuracy when used on an actual crowd-sourced deployment / samples collected from randomly generated locations.
- Show validation as well as evaluation of new model - explain increasing threshold will remove potentially useful boundary cases, but remove a large amount of useless noisy data.
- Validation should mainly be shown through qualitative justification and explanation, but can be done over a large number of tests (multiple runs on multiple floors and buildings) in the absence of mathematical proof.
- Test same parameters across other data sets (library and computer lab) for consistent levels of accuracy

GOODPIECESWITHIN

Testing should show validation as well as evaluation of the new model. Validation could explain how increasing threshold level will remove APs from other floors, but also potentially remove small overlaps from nearby buildings (i.e. if an AP is visible only in a small area near the edge of the building, seeing that AP means the location must me in that space). A similar effect could potentially be observed across floors (i.e. the AP only being visible if the location is directly below it), but this would require a much wider domain for parameters, such as extremely low transmission powers or extremely high path loss rates. However, use of these extremes, could be identified and used to help place the AP on another floor with some defined loss across floors like the WAF used in RADAR.

6 Further Work

7 Conclusion

MAKE SURE YOU CHECK YOUR REFERENCES ARE CORRECT!

Bibliography

[1] G. Gao and K. Whitehouse, "The self-programming thermostat: optimizing setback schedules based on home occupancy patterns," in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. ACM, 2009, pp. 67–72.

[2] J. Stankovic, I. Lee, A. Mok, R. Rajkumar *et al.*, "Opportunities and obligations for physical computing systems," *Computer*, vol. 38, no. 11, pp. 23–31, 2005.

[3] C. Han, K. Wu, Y. Wang, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 271–279.

[4] ITS International, "Its international - wireless traffic management reduces costs and commute times," [Online]. Available: http://www.itsinternational.com/sections/cost-benefit-analysis/features/wireless-traffic-management-reduces-costs-and-commute-times/, [Accessed: Nov. 18, 2015].

[5] Ekahau, "How wi-fi rtls works," [Online]. Available: http://www.ekahau.com/real-time-location-system/technology/how-rtls-works, [Accessed: Nov. 18, 2015].

[6] J. Torres-Sospedra, R. Montoliu, A. Martınez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems," in *Proceedings of the fifth conference on indoor positioning and indoor navigation*, 2014.

[7] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 337–350.

[8] N. D. Lane, Y. Chon, L. Zhou, Y. Zhang, F. Li, D. Kim, G. Ding, F. Zhao, and H. Cha, "Piggyback crowdsensing (pcs): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 7.

[9] Google, "Android kitkat | android developers," [Online]. Available: http://developer.android.com/about/versions/kitkat.html#44-sensor-batching, [Accessed: Nov. 4, 2015].

[10] Microsoft, "Microsoft indoor localization competition - ipsn 2016," [Online]. Available: http://research.microsoft.com/en-us/events/msindoorloccompetition2016/, [Accessed: Nov. 4, 2015].

[11] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[12] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, 2007.

[13] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

[14] D. Madigan, E. Einahrawy, R. P. Martin, W.-H. Ju, P. Krishnan, and A. Krishnakumar, "Bayesian indoor positioning systems," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2. IEEE, 2005, pp. 1217–1227.

[15] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 197–210.

[16] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[17] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 173–184.

[18] J. Torres-Sospedra, R. Montoliu, A. Martınez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and J. Huerta, "Visualizer for indoorloc database samples," [Online]. Available: http://smartways.init.uji.es/indoor/, [Accessed: Nov. 4, 2015].

[19] J. Lu and K. Whitehouse, "Smart blueprints: automatically generated maps of homes and the devices within them," in *Pervasive Computing*. Springer, 2012, pp. 125–142.

[20] P. Charman, "A constraint-based approach for the generation of floor plans," in *Tools with Artificial Intelligence, 1994. Proceedings., Sixth International Conference on*. IEEE, 1994, pp. 555–561.

[21] I. Bate, private communication, October 2015.

[22] J. Torres-Sospedra, D. Rambla, R. Montoliu, O. Belmonte, and J. Huerta, "Ujiindoorloc-mag: A new database for magnetic field-based localization problems," in *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*. IEEE, 2015, pp. 1–10.

[23] J. Torres-Sospedra, private communication, December 2015.