

Bachelorarbeit

Application of Radio Tomographic Imaging to Sparse Systems Deployed in Indoor Environments

Sarah Viktoria Theußen
Matrikelnummer: 3003739



Networked Embedded Systems Group
Institut für Informatik und Wirtschaftsinformatik
Fakultät für Wirtschaftswissenschaften
Universität Duisburg-Essen

January 28, 2017

Erstprüfer: Prof. Dr. Pedro José Marrón
Zweitprüfer: Prof. Dr. Torben Weis
Zeitraum: 18. November 2016 - 10. Februar 2017

Contents

1	Introduction	3
1.1	Motivation	3
1.2	Problem description	4
1.3	Thesis structure	5
2	Materials and Methods	7
2.1	Wireless Sensor Networks	7
2.2	Radio tomographic imaging	8
2.2.1	Shadowing-based RTI	11
2.2.2	Variance-based RTI	12
2.2.3	Impact of device height	13
2.2.4	Impact of device placement	14
2.2.5	Impact of device density	15
2.3	Testbed	16
2.3.1	Nodes	18
3	Approach	21
3.1	General components	21
3.2	Generation of traces for RTI	22
3.2.1	Sampling of received signal strength	22
3.2.2	Data collection	23
3.3	Localization	23
3.3.1	Filtering of long links	23
3.3.2	RTI program	24
3.4	Room occupancy	25
3.5	Additional scripts	25
4	Implementation	27
4.1	Localization	27
4.1.1	Filtering of long links	27
4.2	Room occupancy	28
4.3	Additional scripts	29
4.3.1	Analysis scripts	29
4.3.2	Processing script	29

5	Evaluation	31
5.1	Evaluation metrics	31
5.2	Experiment set up	32
5.3	Localization	34
5.3.1	Influence of ellipse width	35
5.3.2	Influence of long links	36
5.4	Room occupancy	37
5.4.1	Influence of ellipse width	38
5.4.2	Influence of long links	38
6	Discussion	41
	Bibliography	43

Abstract

The function of the abstract is to summarize, in one or two paragraphs, the major aspects of the entire bachelor or master thesis. It is usually written after writing most of the chapters.

It should include the following:

- Definition of the problem (the question(s) that you want to answer) and its purpose (Introduction).
- Methods used and experiments designed to solve it. Try to describe it basically, without covering too many details.
- Quantitative results or conclusions. Talk about the final results in a general way and how they can solve the problem (how they answer the question(s)).

Even if the Title can be a reference of the work's meaning, the Abstract should help the reader to understand in a quick view, the full meaning of the work. The abstract length should be around 300 words.

Abstracts are protected under copyright law just as any other form of written speech is protected. However, publishers of scientific articles invariably make abstracts publicly available, even when the article itself is protected by a toll barrier. For example, articles in the biomedical literature are available publicly from MEDLINE which is accessible through PubMed. It is a common misconception that the abstracts in MEDLINE provide sufficient information for medical practitioners, students, scholars and patients[citation needed]. The abstract can convey the main results and conclusions of a scientific article but the full text article must be consulted for details of the methodology, the full experimental results, and a critical discussion of the interpretations and conclusions. Consulting the abstract alone is inadequate for scholarship and may lead to inappropriate medical decisions[2].

An abstract allows one to sift through copious amounts of papers for ones in which the researcher can have more confidence that they will be relevant to his research. Once papers are chosen based on the abstract, they must be read carefully to be evaluated for relevance. It is commonly surmised that one must not base reference citations on the abstract alone, but the entire merits of a paper.

Chapter 1

Introduction

Wireless sensor networks (WSN) are made of simple sensor nodes. When two nodes communicate with each other, there exists a link with a certain strength between them. If a person crosses this link, the communication becomes weaker. This concept can be used to locate the person in the network. This technology does not require persons to carry any additional device, meaning that any person walking through the network can be located without preparation. Furthermore, the technology can only locate obstructions in the network, not identify the person, ensuring their privacy.

1.1 Motivation

There are many uses for localization in private environments or for public services. One great advantage of this system is the lack of need for any additional device persons need to carry to be able to be located, called device-free localization (DFL). This allows the system to monitor an environment without much configuration time because only the network and its set up need to be known. Another benefit is that the privacy of any person is ensured. The system only allows to determine where a person is, to determine its identity another factor would be needed, e.g. an additional device like a smartphone or camera surveillance. Additionally, multiple persons can be localized instead of only one, resulting in many potential applications.

One possible application is the usage in emergencies. The wireless sensor network can be set up around the building, afterwards persons trapped during a fire or hostages and their captors in a hostage situation can be located. This enables to focus on certain locations which eases the rescue and lowers the risk for officers.

A second application option is a smart home system in which the concept of localization can be used to automatically adjust environment factors, e.g. lighting or temperature control. The knowledge in which room how many persons are makes it possible to bring these factors into line with known additional factors like outer temperature or daytime. For this option, especially when used in offices or public areas, privacy must be guaranteed.

For some indoor environments and application scenarios, the knowledge whether a certain room is occupied might be sufficient. This leads to a third option: determining room occupancy, e.g. for office environments. An interactive floor or building plan could show, which meeting room is currently used and which is free. A room occupancy system would provide real time data, easing the decision which meeting room to go to. Another possibility would be a notification option, so a person gets notified when one or even a certain desired meeting room is free again. In this way, a room occupancy system could help save time and increase efficiency.

An emerging technology called radio tomographic imaging (RTI) allows to locate persons or objects by imaging the received signal strength in wireless sensor networks (WSN). It has been proven promising as it can reach a high accuracy of less than one meter with 30 devices placed in an area of 70 m² [KBP12, pp. 259-261].

1.2 Problem description

The accuracy of radio tomographic imaging is influenced by three main factors: the placement, height and density of the devices forming the wireless sensor network. The number of links is crucial for an accurate localization, meaning that devices have to be placed in a way that over the whole covered area there is a sufficient number of links available. The devices should be placed at torso height, where a person causes the most impact on the links. Device density influences the number of links and many of the potential applications cover large environments such as whole buildings. Setting up a system with a similar device density as in the typical environments would require many devices. However, a high device density results in a complex system regarding the set-up and maintenance.

The goal of this thesis is to evaluate the applicability of an existing RTI solution to a larger area by examining the achievable localization accuracy in terms of accuracy in meters and room occupancy. In this thesis, RTI will be applied to a stationary testbed at the University of Duisburg-Essen. This testbed consists of 32 nodes in a wireless sensor network that covers an area of approximately 530 m², an area 7.5 times as large as typical environments with the same number of devices. Another difficulty is a sub optimal placement of the devices. First, there are rooms with a high device density but also areas in which are barely any nodes, resulting in a low link density. Second, the device height is not uniform, some nodes are set up at body height whereas others are placed at a height above the head of most persons. Comprehensively, in the testbed all factors that influence the accuracy of RTI are sub optimal, which makes it a challenging scenario worth investigating. The evaluation in this theses will include an analysis how the structure of the testbed influences the accuracy and applicability of RTI.

1.3 Thesis structure

This thesis first provides important background information in Chapter 2, covering wireless sensor networks, radio tomographic imaging and the stationary testbed. In Chapter 3, the basic concept of the performed solution will be explained, whereas Chapter 4 covers the concrete implementation. This includes problems that occurred during the implementation and their solution. The system evaluation is part of Chapter 5, as well as the description of the conducted experiments with their results and analysis. Chapter 6 contains a discussion with the interpretation of the evaluation results focusing on the initial problem. Finally, there is a conclusion in Chapter 6 that summarizes the results of this thesis.

Chapter 2

Materials and Methods

In this chapter, necessary background information is covered. First, wireless sensor networks are stated. Second, the concept of radio tomographic imaging is explained along with the specific tool used. Third, the stationary testbed used for the experiments is presented including the devices and their operating system.

2.1 Wireless Sensor Networks

Wireless sensor networks have some unique characteristics that distinguish them from other wireless networks. A wireless sensor network consists of multiple sensor nodes which are supposed to monitor a predefined area, e.g. its physical or environmental conditions. These nodes are small, but in order to be able to monitor a region of interest, equipped with sensors to measure temperature, humidity or other conditions. Additionally, they are able to communicate wirelessly, e.g. over radio transceivers, and to process data by using a microprocessor [ZJ09, p. 2]. The primary advantage of using radio frequency for communication is that no direct line-of-sight between two nodes is required and links are not only uni- but bidirectional [ZJ09, p. 9]. This makes it possible to use radio transmitters in obstructed areas as they can communicate through obstacles and walls. In many cases, all monitored data is sent to one particular node, often called the sink. All nodes collaborate to fulfill a defined task, e.g. monitor a certain area. To do that, all data is needed for the processing, for this reason all data is often sent to a sink. The characteristics and constraints explained hereinafter provide some challenges in the design and set-up of wireless sensor networks.

The nodes used for wireless sensor networks are typically under the following constraints [ZJ09, pp. 2-3]. First, there are power consumption constraints as the nodes are usually battery-powered. In some environments, it might be difficult to change the batteries regularly. This leads to the need of power saving implementations like event-driven programming which is possible i.a. on the embedded operating system TinyOS as explained in section 2.3.4. Second, there are computation and storage constraints. This is why the data is often collected at the sink and sent to a PC where it can be further processed,

stored and visualized. Third, the nodes are not 100% reliable as they are working autonomously. They can fail to monitor the environment or to communicate e.g. after being damaged or the energy gets low.

These constraints lead to some characteristics of a wireless sensor network [ZJ09, pp. 2-3]. The topology frequently changes due to unreliable nodes and the addition of new nodes or connection losses, e.g. because of obstacles in the network. Furthermore, the network architecture is typically the "many-to-one traffic pattern" [ZJ09, p. 3], where all data of interest is sent to one sink. In some cases however, there are multiple hops needed for the data to reach the sink because not all nodes can directly communicate with each other due to their distance or obstacles in their communication path. Another challenge is the interference of communication when multiple nodes send their information over the network [ZJ09, p. 35]. If two messages collide when two nodes send data at the same time, the data can get lost. This causes the need to implement schedules and routing algorithms to prevent interference and data loss.

2.2 Radio tomographic imaging

Radio tomographic imaging is a means to visualize the impact that persons or obstacles in the monitored environment cause on the propagation of the signals in a wireless sensor network. For each pair of nodes in the network, a uni- or bidirectional link between them exists. The link is not necessarily bidirectional. For communication over this link - when using radio frequency for transmission - there is no direct line-of-sight needed but the data can be transmitted through walls and obstacles. An object does not only absorb some of the signal strength, but also reflects, diffracts and scatters some of it [WP10a, p. 622], called attenuation. If a person walks into the monitored environment, the quality of the communication link decreases. This quality can be described as received signal strength (RSS). For the localization of physical objects, the goal of RTI is to determine where the change in RSS occurs due to objects in the environment instead of noise [WP10a, p. 622].

If we describe the number of nodes as K , the total number of links is

$$M = \frac{K^2 - K}{2},$$

meaning that only unique links are counted and communication must not be necessarily possible [WP10a, p. 623]. In Figure 2.1a, an RTI network is illustrated. The nodes are deployed in a squared area where each one can communicate with all others, leading to a fully linked network. In the following, the basic algorithm to create a radio tomographic image is explained as defined in [WP10a].

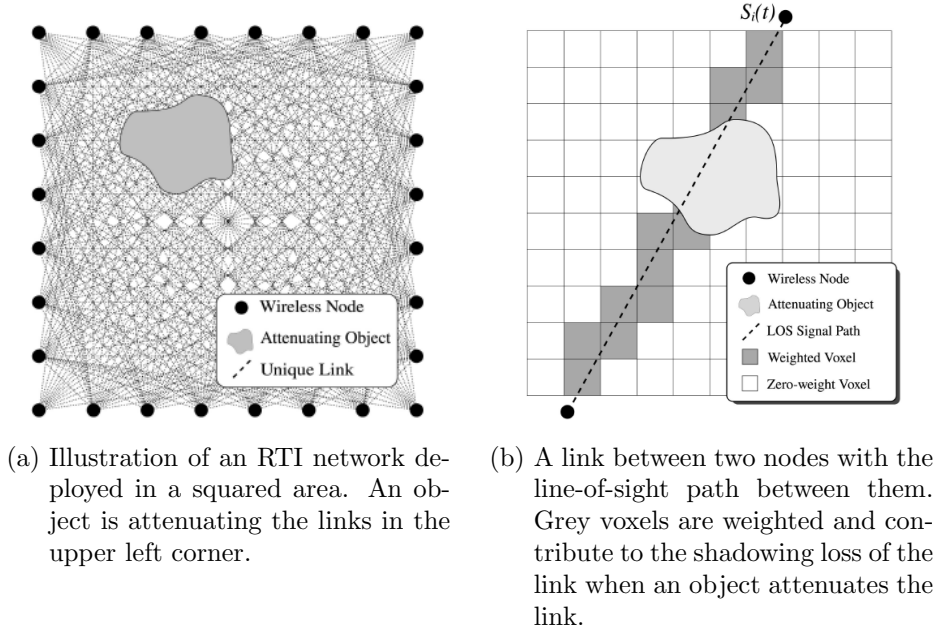


Figure 2.1: Illustration of (a) an RTI network and (b) the weighting of a single link.
Source: [WP10a]

To understand the way the image is created, the term voxel needs to be understood. A voxel is similar to a pixel: a pixel represents a value on a regular two-dimensional grid while a voxel represents a value on a three-dimensional grid. The generated image is a matrix that describes the amount of attenuation happening within a number of voxels N in a network [WP10a, p. 622]. With the knowledge where a voxel is located, the position of the attenuating object can be derived.

When a certain link is obstructed, the overall shadowing loss that a physical object infers on it is the sum of attenuation in each contributing voxel, see Figure 2.1b. To determine the contribution of voxels, they are weighted for each unique link separately. If a voxel is not in the line-of-sight (LOS) path, it is "removed" by applying a weight of zero. The weight of a certain voxel j and link i is w_{ij} , the attenuation in voxel j is $x_j(t)$. Summed up they describe the shadowing loss of a single link:

$$S_i(t) = \sum_{j=1}^N w_{ij} x_j(t).$$

To determine which voxels contribute to a certain link, an ellipsoid can be used to define the LOS path. The ellipsoid has foci at the location of the two nodes it is drawn around. If a voxel is in the ellipsoid, it is weighted, otherwise not. Longer links provide less information about an object's position due to the distance between the two nodes.

The distance results in a larger area affected by the attenuation and thus, the position could be anywhere in that area. For shorter links, the position is more reliable as the affected area is smaller. Furthermore, longer links have weaker signals and are thus less robust to noise and likelier to be affected by environmental changes [ABB⁺16, p. 2589]. Added to that, the RSS can be close to the so called sensitivity threshold of the radio modules [ABB⁺16, p. 2589]. If the measured RSS of a link is around that threshold, the link's connectivity is unpredictable and the RSS includes a higher noise and experiences more interference. A shorter link with a stronger signal will experience a more reliable attenuation when it is crossed by an object [WP10a, p. 624]. The weighting by distance tries to ensure that shorter links are taken stronger into account than longer links. The voxel is weighted by dividing it by the square root of the distance d :

$$w_{ij} = \frac{1}{\sqrt{d}} \begin{cases} 1, & \text{if } d_{ij}(1) + d_{ij}(2) < d + \lambda \\ 0 & \end{cases}$$

$d_{ij}(1)$ and $d_{ij}(2)$ describe the distance from the center of the voxel j to the locations of the two nodes of link i . λ is a parameter that defines the width of the ellipsoid. The width is typically very small as it then resembles the direct LOS path. On the one hand, if λ is too small, information about attenuation in voxels near the ellipsoid can get lost, although they experience changes. On the other hand, if it is too large, the information about where an object is present in the network can be distorted [WP10a, p. 630]. The width parameter must be set fittingly for the specific network to avoid loss in accuracy and detection rate.

We already know that the RSS from one node to another does not only consist of the transmitted power P_i , but is rather influenced by other factors. i describes the unique link and t a certain point in time. The factors are first, the shadowing loss $S_i(t)$ caused by obstructions; second, signals can interfere constructively and destructively, resulting in a fading loss $F_i(t)$; third, the so called static loss L_i caused by hardware and distance between devices and fourth, there is always a certain amount of measurement noise $v_i(t)$. Only the static loss is not influenced by time, it remains constant for a specific set up. When all factors are considered, the signal strength can be calculated as

$$y_i(t) = P_i - L_i - S_i(t) - F_i(t) - v_i(t).$$

If only the changes are included in the imaging, L_i and P_i can be discarded as they are constant over time. Additionally, fading and measurement noise can be grouped as noise n_i . When the change in RSS and attenuation is not calculated at one single point in time but as a difference between two points in time t_a and t_b , they can be written as

$$\Delta y_i = S_i(t_b) - S_i(t_a) - n_i(t_b) - n_i(t_a) \text{ and}$$

$$\Delta x_j = x_j(t_b) - x_j(t_a).$$

The result of RTI is then a system of RSS equations in matrix form, where M is the weight matrix, n the noise vector, Δy the vector with all RSS measurements and Δx the attenuation image. The calculation is as follows:

$$\Delta y = W \Delta x + n$$

The weight matrix is of dimension $M \times N$, where the weight for each voxel of each link is stored. The attenuation image to be estimated is a vector of size $N \times 1$, describing the amount of attenuation for each voxel. Both the vector with the RSS measurements and the noise vector are of size $M \times 1$ as they store the values for each link. All parameters are measured in decibels (dB).

The estimation of the image vector Δx results in an ill-posed inverse problem [KBP12, p. 256]. It is called an inverse problem, because the cause is calculated from the known results instead of the other way around. In this case, the result is the change in RSS as this is measured in the network. The cause is the location of an obstruction and calculated from the result. The same set of results, i.e. RSS measurements, can lead to different solutions, resulting in different images [KBP12, p. 256]. To solve this, regularization is required like the least-squares solution [WP10a, p. 626]. This minimizes the noise needed to fit the data to the estimated image and thus leads to a unique solution.

To estimate the location of the obstruction, the voxel with the maximum value in attenuation can be found [KBP12, p. 257]. The likeliest position of an obstruction is defined as

$$j = \arg \max_N \hat{x}.$$

This calculation of a position allows only the localization of one obstruction in the monitored area, no multi-target tracking. However, there are investigations in the localization of multiple targets.

The RTI program used in this thesis implements enhancements of this basic algorithm. It creates two different images for the same set of RSS measurements. One is shadowing-based RTI, where stationary people can be located by comparing the measured RSS to a base value. The other is variance-based RTI, which is able to track moving persons through an environment.

2.2.1 Shadowing-based RTI

Shadowing-based RTI, as explained in [KBP12], enhances the basic algorithm by using multiple channels. This increases the localization accuracy in heavily obstructed environments. The reason for this is, that the attenuation caused by a person in the environment cannot be easily distinguished from changes due to reflection or scattering of a signal. In a direct LOS path, only the links where a person stands and in a small

area around it are influenced. In obstructed environments, also links that are quite far away from the obstruction can be influenced. Furthermore, the RSS can increase for an obstructed link instead of an expected decrease. There are two different types of links: anti-fade and deep fade. Deep fade links are worse for RTI as they tend to increase for an obstructed LOS whereas anti-fade links tend to decrease and are more reliable [KBP12, p. 257].

By using multiple channels to measure the RSS for each link, the fade level can be maximized by selecting the best channels for each link after a calibration phase. In this way, the RSS is more reliable than using one single channel, as more anti-fade links can be used to create the image. Experiments have been conducted in an area of 70 m² with 30 devices placed around the walls of a lounge room with furniture, resulting in a device density of about 0.43 nodes per meter, converted 0.04 nodes per square foot (metric used in Figure 2.4 where the impact of device density on the average MSE is depicted). The best possible accuracy for an obstructed environment was 0.61 m compared to single-channel RTI with 1.81 m.

2.2.2 Variance-based RTI

Variance-based RTI (VRTI), defined in [WP10b], is able to track persons by monitoring the motion-induced variance in a network. The variance on a certain link caused by motion is higher than in an empty environment. Here, the vector Δy is named s and represents the RSS variance measurements on the links. The resulting image vector x shows the motion in a number of voxels N in the environment. The definition whether motion occurs in a voxel is as follows:

$$x(j) = \begin{cases} 1, & \text{if motion occurs in voxel } j \\ 0, & \text{otherwise} \end{cases}$$

The variance vector is also weighted by a variance weight matrix M and influenced by measurement noise and modeling error n . The voxels are weighted by distance as in the basic algorithm, because an ellipsoid is used again to determine the contribution of each voxel to the variance of a link. When all links are considered, the equation is similar to the one in the basic algorithm:

$$s = Wx + n.$$

The variance of a link is not calculated from a base value, but rather from a buffer of a certain number of previous samples. It is assumed that the environment remains constant for a short period of time. When the buffer size is too low, the image is influenced by more noise and modeling error, when too large, the image is blurred and the tracking lags [WP10b, p. 619]. Additionally, a Tikhonov least-squares regularization is used to solve the ill-posed inverse problem.

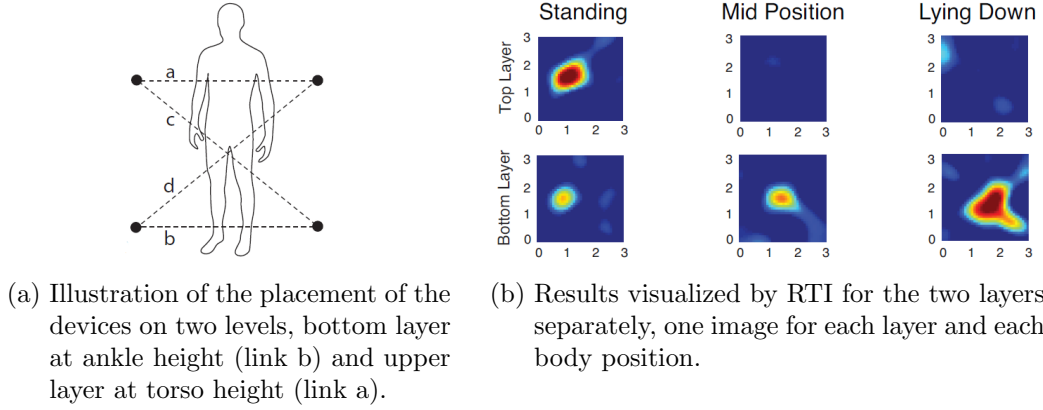


Figure 2.2: (a) Illustration of the placement of the devices on two levels. (b) Results visualized by RTI on the two layers in different body positions. Source: [MPB13]

Experiments have been conducted in a rectangular area around a typical home with one room including furniture. 34 nodes were placed around the four walls at approximately the height of the human torso. The nodes are placed around an area of approximately 75 m², resulting in a node density of 0.45 nodes per meter, converted 0.04 nodes per square foot (see Figure 2.4). A moving person causes a more reliable variance than a stationary [WP10b, p. 617]. The location of a person in the environment was estimated by using a Kalman filter instead of the maximum variance as opposed to the attenuation-based algorithm, resulting in an average error as low as 0.63 m.

When radio tomographic imaging is to be used to monitor the changes in RSS, the placement and height of the devices plays a great role in the localization accuracy. The impact these two factors have is explained in the following.

2.2.3 Impact of device height

The received signal strength needed for radio tomographic imaging is influenced i.a. by obstacles in their path, e.g. a person. The human torso influences the RSSI more than the ankles [MPB13, p. 2], from this can be inferred that the head also has a smaller influence because it is less dense as the torso, as well as the ankles are.

The correlation between the torso and the ankles was determined by experiments with two levels of devices in a 3 x 3 m square in which the RSS was measured [MPB13, p. 3]. The lower level is near the ground at 0.17 m and the upper level at torso height for most people, namely 1.40 m [MPB13, p. 3]. The levels are depicted in Figure 2.2a. The goal of these experiments was to evaluate whether falls, e.g. of older persons in residential care facilities, can be detected so help can arrive quicker [MPB13, p. 1]. For this, a

clear difference between both positions must be detectable. This can be realized by the two layers which can be used to determine the position of a body. To accurately get the position, there must be a significant difference between standing and lying down, which means that the ankles cannot cause the same impact on the lower layer as the torso on the upper layer when standing. Otherwise the difference to lying down, where only the lower layer is expected to be influenced, will be less significant.

The experiments covered various body positions, including walking, lying down, sitting on the floor as well as on a chair and falling down [MPB13, p. 3]. The measured RSS data is processed by generating two images with RTI for each of the two layers [MPB13, p. 3]. In Figure 2.2b the results regarding three different body positions are presented: standing, mid position (like bending down) and lying down. When a person is standing or walking in the environment, the impact caused by the torso, as seen on the upper layer, is much more significant than the impact on the lower layer caused by the ankles. In mid position and when lying down, the impact on the upper layer is barely visible and only recognizable on the lower layer.

For the testbed this means that in some areas where devices are placed at or above the height of the head, important data can get lost. The impact caused by walking or standing can be less significant or even get completely lost, which influences the accuracy and detection rate.

2.2.4 Impact of device placement

In addition to the height of the devices, the placement in and around the area of interest is a prominent factor to accurately localize persons with RTI. Typical environments cover a square or rectangular form, usually an area without walls or only one room. There are three different deployment geometries to place devices: square, front-back and random node deployment [WP10a, p. 626].

In Figure 2.3 the layout of the three geometries is shown. The number of devices is 28 in the squared geometry, which are placed evenly around the square area depicted the left image. In the middle image, the number of nodes is smaller, as for the front-back deployment the devices are placed only on two opposing sides of the square. The number of devices per side is the same as in the square deployment, namely 8, resulting in 16 nodes in total. In the right image, 22 devices are randomly placed inside the square for the random layout.

To assess the accuracy for these three layouts, the actual position and the estimated position need to be known. With these two factors, the mean squared error (MSE) can be calculated [WP10a, p. 630]. First, the difference between the actual and estimated value is calculated as error. Second, the average of the squares of the errors is computed,

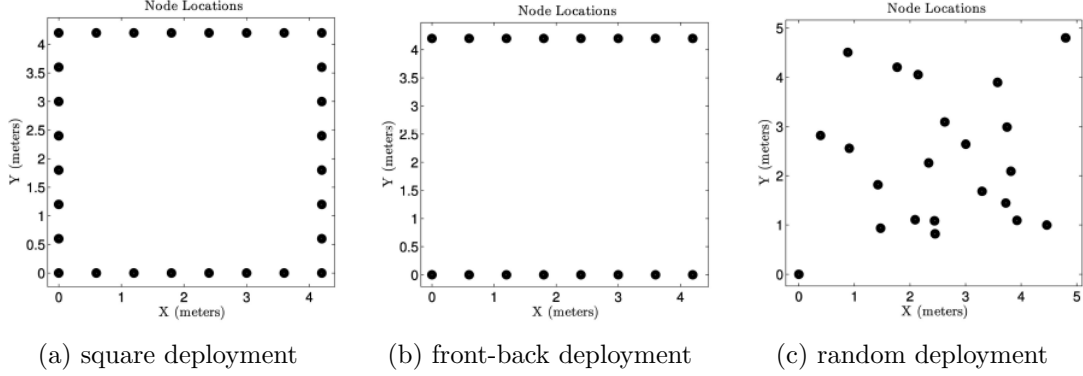


Figure 2.3: Layout of different deployment geometries: square, front-back and random.
Source: [WP10a]

which is called MSE. The MSE describes the quality of the estimated values: the smaller the MSE, the better the quality.

For evaluating the deployment geometries, the accuracy over the whole network area is determined by using the MSE [WP10a, p. 626]. The layout affects the available links and thus, which pixels (namely positions in the square) are estimated more accurately [WP10a, p. 626]. The accuracy increases with more links, that means the more links cross a certain pixel, the more accurate it can be estimated. In all geometries, there are areas that are crossed by no links at all and some that have a low link density. In the front-back deployment, these are the sides of the square where no nodes are placed; in the square deployment, the corners and in the random deployment, the low-density areas where few devices are placed. These areas have a higher MSE than other areas [WP10a, p. 627].

For many application cases, the square geometry is certainly a good choice, as it leaves few areas with a low link density (see figure 2.1 a) and thus, low accuracy, because the whole area is surrounded by nodes. This is also visible in Figure 2.4, where this geometry outperforms the others, no matter the node density. For other forms of the area of interest, a mixture of all layouts might be optimal, this might be a topic for future research.

2.2.5 Impact of device density

On the whole, not only the placement regarding device height and deployment geometry influence the accuracy and detection rate, but also the density of devices. First, the node density influences the link density. The more links, the more RSS information is available for a certain area that can be used to infer the events in that area without taking noise and smaller corruptions into account [WP10a, p. 626]. Second, the smaller

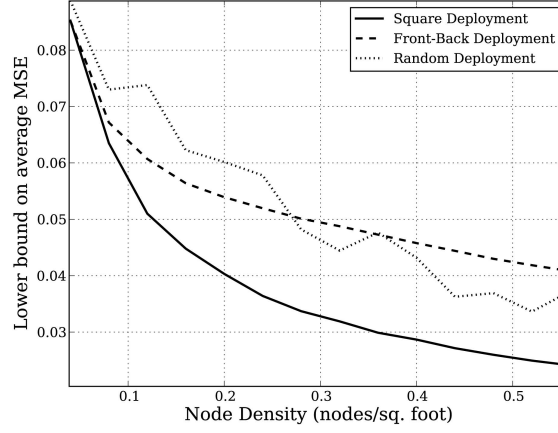


Figure 2.4: Effect of device density on the mean squared error (MSE) in different deployment geometries. Source: [WP10a]

the distance between nodes, the more accurate is the imaging expected to be, because the RSS information is more concentrated in an area [WP10a, p. 626]. This means, when there are many longer links, the image could be distorted.

To evaluate the effect of node density on the accuracy, the MSE is used again. The MSE is calculated for all three deployment geometries, while the density in nodes per square foot is increased. The results are illustrated in figure 2.5. When the node density is increased, the MSE decreases quickly, in the square and front-back deployment nearly exponentially. In the random form, there are some ups in between, probably caused by a more lucky placement, where there are fewer areas crossed by a small number of links.

However, not only the density has influence on the MSE. With a higher density comes not necessarily a higher accuracy, but also the positioning and environmental factors play an important role, e.g. wind and weather conditions in an outdoor environment [ABB⁺16, p. 2593]. In some cases, a higher density can even result in a lower accuracy, and a lower density in a higher accuracy when devices are positioned in optimal spots [ABB⁺16, p. 2593].

2.3 Testbed

The testbed is quite different to the typical indoor environments as all factors that influence the accuracy of RTI are sub optimal. First, the placement of the devices is not uniformly at torso height but some are placed higher. Second, there is a mixture of the three deployments. There are some areas crossed by no links at all, e.g. the sides of the main corridor and the corners of the lab on the southern side of the main corridor.

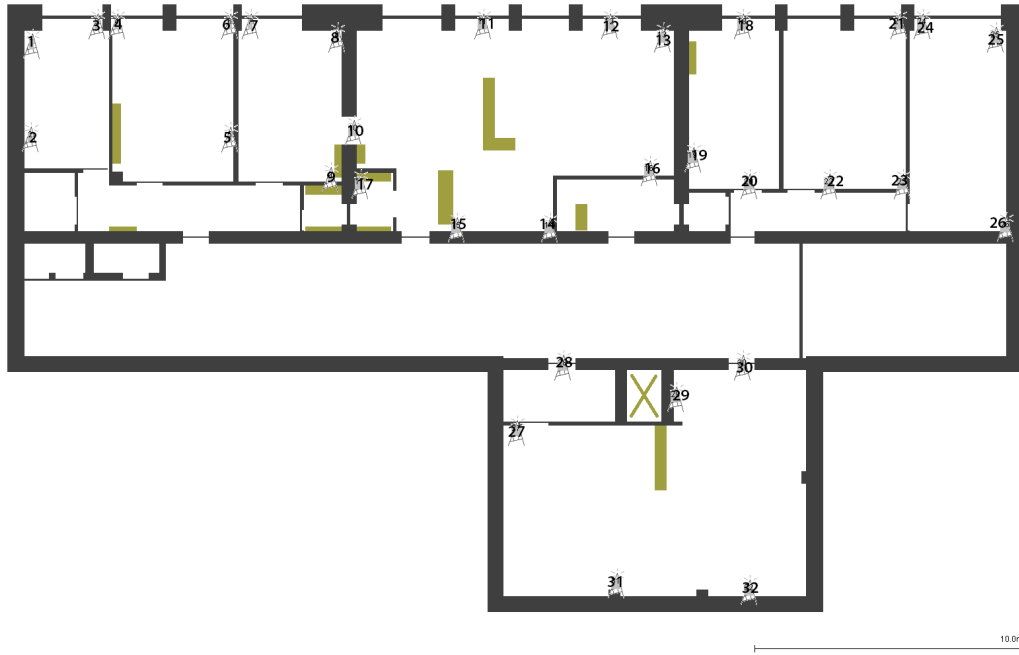


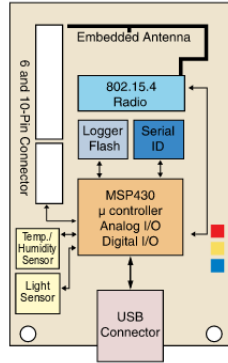
Figure 2.5: Map of the stationary testbed with the antennas representing the locations of the nodes with their unique ID.

Additionally, most areas have a low link density compared to typical environments, as the overall device density is low. Third, the density of devices is lower and the covered area is larger. The device density is 0.006 nodes per square foot. As Figure 2.4 shows, the MSE is quite high for this density regarding all three ways of deployment.

A wireless sensor network is set up as an indoor testbed at the University of Duisburg-Essen. It is located at campus Schützenbahn in Essen on the 3rd floor of the SA building. The testbed covers one half of the building and includes one main corridor, two laboratories and two smaller corridors with six offices in total. There are also seven smaller storage rooms and one server room. The main corridor can be entered from the left and the right side, where there are staircases, and from an elevator. This architecture leads to a heavily obstructed area with many obstacles, including walls, metal cupboards and desks. The total area of the floor is approximately 531 m², excluding the staircase on the right side of the main corridor.

Figure 2.5 shows the structure of the floor. The antennas represent the nodes that monitor the environment and the smaller room marked with an X is the elevator shaft.

The floor is monitored by 32 stationary low-power wireless radios [Adv]. The devices measure the received signal strength (RSS) to other nodes in their communication range, this indicator can be used to determine a person's location by using radio tomographic



(a) Block diagram of the TelosB mote (source: [Mem]). (b) Photo of the TelosB mote (source: [Zur]).

Figure 2.6: Block diagram and photo of the TelosB mote.

imaging explained in the section above. The received signal strength indicator (RSSI) is determined by communication via radio messages: when a message is received by a node, the RSSI from the sender can be measured. All RSS values in the network can then be used for RTI.

Comprehensively, in the testbed all factors that influence the accuracy of RTI are sub optimal, which makes it a challenging scenario worth investigating.

2.3.1 Nodes

The testbed is set-up with 32 devices, so called nodes. Together, they construct a wireless sensor network meant to measure the received signal strength between nodes. Each device is placed ascendingly over the area by its own unique ID, ranging from 1 to 32. This implies that the node with ID x can communicate at least with its two neighboring nodes with IDs $x - 1$ and $x + 1$.

Typically, sensor nodes consist of four units: sensing, processing, communication and power [ZJ09, p. 20]. The devices used here are Tmote Sky [Adv], equivalent to TelosB motes [Mem], with the operating system TinyOS. They are low-power devices with an integrated radio transceiver for communication. The power can be provided from a battery pack containing two AA batteries if used wirelessly or the USB port. The USB port can also be used to program the devices and to collect the measured data. The block diagram in Figure 2.6a shows the features offered by the mote. The mote offers temperature, humidity and light sensors [Adv].

TinyOS is a small, energy-efficient operating system specifically designed for sensor nodes with their power and computation constraints [LG, p. 5][Mem]. It is an event-driven


OS that does not need much RAM and also does not provide a file system, only static memory allocation [ZJ09, p. 11]. The advantage of an event-driven system is that the motes can sleep during the time they do not compute anything and are triggered when an event occurs, e.g. a message is received. This leads to less power consumption, one of the motes' constraints. TinyOS is written in nesC, a dialect of C, which helps reduce RAM and code size [LG, p. 5]. Furthermore, it provides APIs for message transmission and reading sensors [LG, p. 6].

In the testbed, each mote is connected to a Raspberry Pi which is connected to the local network. When a PC is connected to the local network, it can be used to program the devices and to collect the data measured by the motes. Using this non-wireless set up eases and quickens the data collection, as each mote can directly send its data to the PC where the data can be processed. The measurement code used to generate the traces for RTI is explained in Chapter 3.

Chapter 3

Approach

3.1 General components

The goal of this thesis is to evaluate, whether an existing RTI solution can be successfully applied to a large indoor environment with sub optimal device height, placement and density. In order to do so, the stationary testbed at the University of Duisburg-Essen  was used. This Chapter provides a description of the performed solution.

For the evaluation of RTI, two different approaches are used. First, the accuracy in meters and second, the accuracy in terms of room occupancy is calculated as the knowledge which room is occupied can be sufficient in indoor environments, see Chapter 1.1. As radio signals can penetrate walls, changes in RSSI could happen in two different rooms. This can lead to a small localization error, but the error in room detection can be larger as a room next to the occupied is detected. In this thesis, the correlation between both types of accuracy, namely accuracy in meters and accuracy in terms of room occupancy, is evaluated.

The system consists of three general components, as visible in Figure 3.1. First, the RSS needs to be collected in the testbed. These traces are then used for the second component, the RTI program, that calculates estimated positions of a person. The third component is the estimation of room occupancy, a program determines the occupied room from a given position.

The influence of long links and λ , the width of the ellipsoid in RTI, are analyzed. Long links can be filtered out by altering the traces with a script before running the RTI program. λ is a parameter that can be changed in the RTI program itself. In this way, the localization can be optimized as much as possible when all parameters are fittingly chosen.

Furthermore, a script is written to ease and quicken the analysis. It calculates the accuracy of both accuracy in meters and room occupancy. All components of the system are further explained in the following.

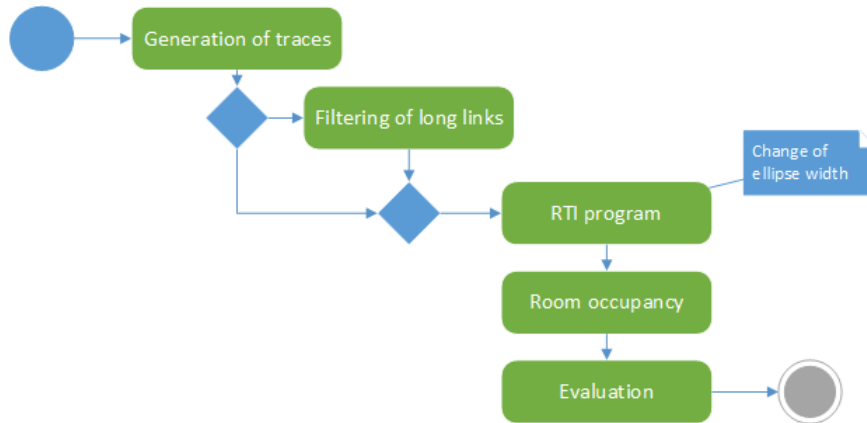


Figure 3.1: Activity diagram showing the components of the system: First, RSS traces need to be generated, long links can optionally be filtered out. Traces are processed by the RTI program, which produces estimated positions used to infer room occupancy. Finally, results are evaluated.

3.2 Generation of traces for RTI

The RSS does not need to be measured in a special way to be compatible with the RTI program, only a specific formatting is needed (described below). Here, one possibility is described that is used for the experiments. The traces are generated by measuring the received signal strength in the testbed. All measurements are sent to a PC which stores the data and creates a file for the RTI program. A nesC program running on the motes fulfills two tasks explained in the following: scheduling the messages and collecting the data at the PC.

3.2.1 Sampling of received signal strength

Scheduling the messages is necessary to avoid collisions. Typically, so called timeslots are used. Each node is assigned one certain timeslot, calculated by the node's ID multiplied with the worst case time a node needs to send a message. This method bears one disadvantage: it is quite slow as a node cannot send a message faster than its predefined slot, even if a message from the previous node was received. To create a faster schedule, a certain order was defined additionally to the timeslots. In the testbed, the motes are set up by their ID, meaning that node x can communicate with its two neighbouring nodes $x - 1$ and $x + 1$, so the order could be defined ascendingly using the node's ID. When a node receives a message, it can recalculate the time it is its turn at the latest, namely Δt .

$$\Delta t = (ID - ID_{received}) * timeslot,$$

where the ID of the node and $ID_{received}$, the ID of the node it received from, are known. $timeslot$ is the worst case time a node needs to send a broadcast. The node starts a timer with Δt , so that if a message gets lost, this timer fires and the node sends its message. In this way, lost messages are handled and the program continues to measure the RSS. This algorithm is typically faster than the timeslot based method, in the worst case, when all messages get lost, they are equally fast. This program was able to measure the RSS of the testbed in average in 247 ms.

ToDo: check!

3.2.2 Data collection

In the testbed, each mote is connected to a Raspberry Pi which is connected to the PC. This eases and quickens the data collection, as the measurements do not have to be sent over the network to a sink and thus, no scheduling is needed. Over the serial, there is no problem with collisions and data losses, as the USB controller buffers the messages and retransmits them if necessary. After each event, namely sending or receiving a message, a node sends a message over the serial with its ID and the following content. If it sent a broadcast, it includes a flag "SEND"; if it receives a message, it includes a flag "RECEIVED", the node it received from and the RSS to this node. In this way, a log file can be created at the PC, where all events are included. Afterwards, the log file can be used to create a new file whose structure equals the expected input in the RTI program, so the trace can be used for RTI. The expected structure of one line for RTI is as follows:

$$RSS_{1 \text{ to } 1} \dots RSS_{1 \text{ to } n} \dots RSS_{n \text{ to } 1} \dots RSS_{n \text{ to } n} \text{ timestamp}$$

Each line consists of the measurements for the whole network at a certain time defined by *timestamp*. The total number of nodes is described by n . For each node, the RSS to all other nodes is given, a non-existing link or a connection to itself, is given by a special value of 127.

3.3 Localization

3.3.1 Filtering of long links



Longer links provide less information about the position of an obstruction as shorter ones, explained in Chapter 2.2. To evaluate whether longer links worsen the accuracy, a



script can artificially delete links that pass a certain threshold length. The script takes two input files: the traces and the coordinates of the sensor nodes. First, it reads the coordinates of the nodes and calculates the distance of each connection to define which links are kept and which are "deleted". Afterwards, the trace file is read and for each link that will be discarded the RSSI is set to 127, which is parsed by the system as no connection. This script eases the evaluation of the impact of longer links, as a certain threshold can be defined and thus, different accuracy values can be compared.

3.3.2 RTI program

The RTI program used implements the two algorithms explained in Sections 2.2.1 and 2.2.2 and creates two images, one for shadowing-based RTI and one for variance-based RTI. After a calibration phase of a determined number of samples, the two algorithms started at the same time. The parameter λ to define the size of the ellipse was set to 0.1 distance units, which are related to the resolution of the images. As λ is very small, it resembles the direct line-of-sight path.

The program can take optional input so the actual position of a person in the environment is known for evaluation. To do that, it needs a number of coordinates a person can traverse, called pivot spots. Additionally, the path over a set of pivot spots and a start time, at which the person enters the environment, must be defined. The program takes a speed at which the spots are reached and can calculate the positions in between two spots at a given time, assuming a constant walking speed. This concept is known as "Ground Truth".

As input, the program needs the following files:

- Coordinates of the sensor nodes
- File with the traces to be processed, including a time stamp
- Optional: Coordinates of pivot spots
- Optional: Pivot path traversing pivot spots

The location estimate of a person in the environment is calculated by the maximum shadowing loss in the shadowing-based RTI, variance-based RTI is visualized but not taken into account for determining a position. If no location could be estimated, a special value of -99 is used. As output, the program creates a file with the estimated locations.

The following changes were made to the original RTI program. After some initial experiments, using shadowing-based RTI to estimate a person's location proved unusable as the shadowing loss was too small to be detected. Instead, the variance-based algorithm was used to determine a location. The method was not changed, still the maximum value



was used as the estimated position. Furthermore, the program was changed to output both the actual and the estimated position in the output file to ease the evaluation of the experiments.

3.4 Room occupancy

A program is used to evaluate the coordinates to determine which room is occupied. It takes an input file with all rooms in the testbed, it provides the coordinate of the bottom left corner as well as the height and width of the room. First, the program reads the file to know all rooms available. Afterwards, it takes the output file of the RTI program as input and reads it. The program outputs both the coordinates of the actual and the estimated position. Each line equals one point in time, for each line it evaluates the actual and estimated position. The x and y coordinate of each of the two positions is used to compare it with the properties of each room. If a coordinate is in the area of a room, this room is output as the occupied room. Two rooms are output in an output file to further process the accuracy.

3.5 Additional scripts

A processing script automatically runs the RTI program with a given trace file. With the positions output by the program, the accuracy in meters can be calculated, as well as other evaluation metrics explained in Chapter 5.1. Afterwards, it starts the program to determine the room occupancy with the output of RTI. With the output of this program, the accuracy of room occupancy can be calculated in percent, in addition to the other evaluation metrics. With this script, the evaluation is quickened as the program takes the traces of multiple experiments and outputs the evaluation metrics.

Chapter 4

Implementation

In this Chapter, information about the implementation of the system components, as visible in the activity diagram in Figure 3.1, is provided. The program to generate the traces and the RTI program existed beforehand, only the program to infer room occupancy and smaller scripts were written. The programs and scripts interact via files that are read and output.

4.1 Localization

For the localization, the generated traces and the RTI program are used. To analyze the impact of long links, the traces can be modified by a Python script to filter out long links. After running the RTI program with a set of traces, an analysis script explained in Section 4.3 can calculate the evaluation metrics described in Chapter 5.

4.1.1 Filtering of long links

The Python script is able to filter out links that are longer than a predefined threshold. To do this, it takes two files as input: the traces and the coordinates of the sensor nodes. The script outputs another file with the modified traces.

First, the file with the sensor nodes is read and the distance between each pair of nodes is calculated. The distance between two points $P1(p00, p01)$ and $P2(p10, p11)$ is calculated with the Pythagorean Theorem, where the distance is the length of the hypotenuse and the difference of their x- and y-coordinate denote the lengths of the other two sides:

Listing 4.1: Distance function

```
def distance(p00, p01, p10, p11):  
    return math.sqrt((p00 - p10)**2 + (p01 - p11)**2)
```

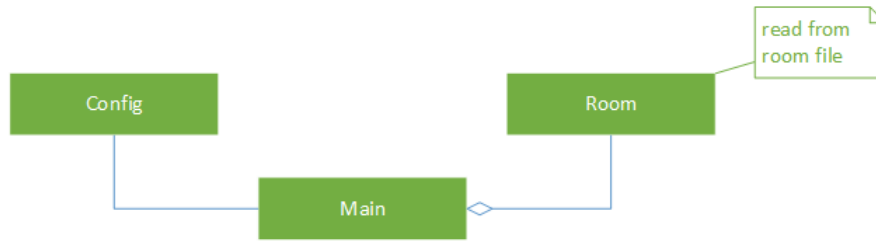


Figure 4.1: Class diagram showing the classes of the Java program. Class `Room` describes a room, `Config` contains static methods to read files and `Main` processes the traces.

The distance is then compared with the predefined threshold `MAX_DISTANCE`. If the distance is smaller than the threshold, the link is appended to a list called `links_to_keep` which stores all links that should be kept, otherwise it is discarded.

Afterwards, the trace file is read. Each RSS value of one line belongs to one link, then each link is looked up in the list `links_to_keep`. If the link is in the list, the RSS is copied to the output file; if not, a special value of 127 is written to the output file. The special value indicates that no connection is present between these two nodes, so the connection is artificially deleted when the link is longer than the threshold.

4.2 Room occupancy

The room occupancy is inferred by a Java program. The class diagram in Figure 4.1 shows the general architecture of the program. As input, a file containing the rooms of the testbed and a coordinate file to process are needed. The coordinate file is output by the RTI program and contains two locations: the estimated and the actual position of a person. The Java program then outputs the inferred estimated and actual room.

The class `Config` contains static variables indicating the required files and a list `ROOMS` containing the rooms of the testbed. Additionally, it provides a static function to read the room file and fill the room list with all rooms created from the room file. Each line of the room file must have the following structure:

$$x \quad y \quad width \quad height \quad ID$$

The class `Room` provides a constructor to create a room, taking five parameters: an x-Coordinate, a y-Coordinate, a width, a height and an ID. All parameters are stored in member variables. Furthermore, the class offers a function `isOccupied()` to determine whether a given point is in the room. It simply compares the given position with the data of the room. The following code snippet 4.2 shows how room occupancy is inferred.

Listing 4.2: Function to determine whether a given point is in the room.

```
public boolean isOccupied(float x, float y) {  
    if( (x >= this.x && x <= this.x + this.width) &&  
        (y >= this.y && y <= this.y + this.height) )  
        return true;  
    return false;  
}
```

The main class provides the main function which first invokes the function to read the room file. Afterwards, it reads the file with the coordinates. Each line contains two points which are processed by the function `processLocation()`. For each point, this function invokes `isOccupied()` until an occupied room is found and returned or 0, no occupied room, is returned. The inferred estimated and actual room are printed.

4.3 Additional scripts

4.3.1 Analysis scripts

Two analysis scripts written in Python calculate the accuracy and evaluation metrics (see Chapter 5.1) of the localization and the room occupancy. Both scripts take as input the file to process.

In case of the localization accuracy, this file contains two positions: the estimated and actual position of a person in the testbed. The script calculates the accuracy in meters, described by the mean error between the actual and estimated position, calculated by the Pythagorean Theorem (see Listing 4.1). Additionally, it calculates the standard deviation σ by using the function `pstdev()` of the package `statistics`. With the standard deviation, the amount of variance can be analyzed.

In case of the accuracy of room occupancy, the input file contains the actual and estimated room. The script calculates the accuracy in percent, indicating how often the correct room is inferred.

4.3.2 Processing script

A processing script, also written in Python, is able to run the RTI program with multiple given traces, execute the Java program to infer room occupancy and call the two analysis scripts. All output is stored in separate files, except the output of the RTI program as it is already output in a file. The general architecture is shown in Figure 4.2.



Figure 4.2: Activity diagram showing the steps of the processing script.

The script reads all trace files in a predefined directory. To start the RTI program, it needs an additional input file that defines the traversed pivot path and a start time for each trace. This file has the following structure:

name of trace pivot path start time

For each trace file, the script first looks up its pivot path and start time. With this data, the RTI program is started with `os.system()`, as this function waits for the command to complete. When the program is done, the script calls the analysis script for the localization. Afterwards, the Java program to infer room occupancy is started and when this is done, the second analysis script is called.

Using the processing script eases the analysis of multiple experiments, as they can all be given as separate traces and the script outputs the evaluation metrics to be analyzed without having to call each program and script for each trace file separately.

Chapter 5

Evaluation

This Chapter covers the evaluation of RTI in the stationary testbed at the University of Duisburg-Essen. First, the evaluation metrics used are explained. Second, the set up of the experiments is provided. Third, the accuracy of localization in meters and room occupancy is analyzed and evaluated. In both cases, the influence of the ellipse width λ and long links is examined.

5.1 Evaluation metrics

The actual accuracy is calculated for both localization and room occupancy. It can only be calculated, when there is an actual and estimated position available. The accuracy in meters is defined as mean of the error, calculated by the distance from the actual to the estimated position. To analyze the amount of variance of the error, also the standard deviation σ is calculated. The accuracy of room occupancy however is defined as the percentage how often the correct room is detected by the system.

To analyze the applicability of an RTI system in such a scenario, three additional values described in Table 5.1 are examined: false negative, false positive and the detection rate. False negative describes, how often the system does not detect a position but there is an obstruction in the environment. False positive on the other hand represents how often the system detects a position, but the environment is not obstructed. The detection rate displays, how often a position is detected when there is an actual obstruction. All three values are calculated for both accuracy in meters and room occupancy.

Table 5.1: Description of evaluation metrics. ✓: detection by the system / obstruction in the environment, ✗: no detection / no obstruction.

	System	Obstruction
False negative	✗	✓
False positive	✓	✗
Detection rate	✓	✓

The analysis of the influence of the ellipse width and long links is done by using the same set of traces and changing parameters. One parameter is the ellipse width λ , provided in the RTI program. The second parameter is the threshold of maximum link length, changeable in the script to filter out long links. The altered traces are then used for RTI. One of the two parameters is always held constant, while one is changed. In this way, the results can be compared to the original results.

5.2 Experiment set up

All experiments were conducted in the stationary testbed at the University of Duisburg-Essen, described in Chapter 2.3. To measure the RSS in the testbed, the sampling algorithm explained in Chapter 3.2 was applied. The generated traces were used to localize with a specific RTI program (see Chapter 3.1) which estimates a person's position by evaluating the change of RSS variance in the environment. Section 5.2 deals with the accuracy in meters, whereas Section 5.3 focuses on the accuracy in terms of room occupancy. In both sections, the ellipse width parameter λ and the influence of longer links were investigated.

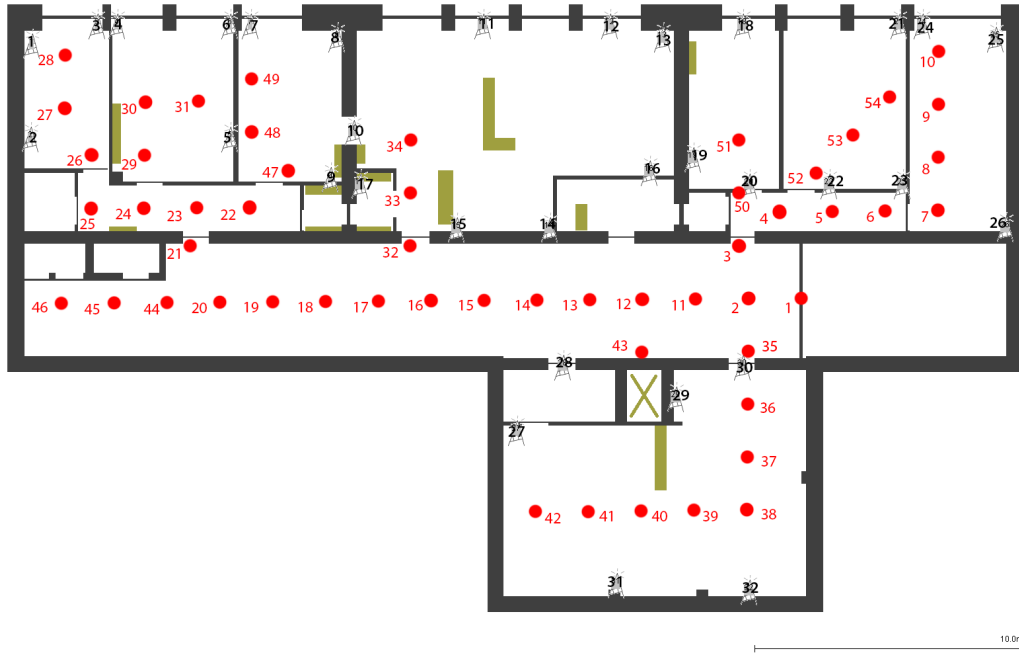


Figure 5.1: Red dots with a unique ID represent pivot spots established in the testbed, used for the concept of Ground Truth. The distance between two spots is 2 m. Paths for experiments were defined as a sequence of pivot spots.

To calculate the accuracy, the concept of Ground Truth, as explained in Chapter 3.1, was used. First, pivot spots with a distance of two meters were established over the area of the testbed, depicted in Figure 5.1. Each spot was provided with a unique ID. Second, different paths were defined. An experimenter walked down these paths with a speed of 1 m/s, equaling 3.6 km/h. Each walk represents one experiment. All different paths with their according sequence of pivot spots are visible in Table 5.2. The area is always entered and exited at the same spot which means that during an experiment, there is maximum one person present in the environment. This is important because the RTI system is only able to detect one person and multiple people could distort the results.

Table 5.2: Pivot paths with a description and their according sequence of pivot spots.

Path ID	Description	Pivot spots
1	Floor	1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 35 36 37 38 39 40 41 42 41 40 39 38 37 36 35 2 11 12 13 14 15 16 32 33 34 33 32 16 17 18 19 20 21 22 23 24 29 30 31 30 29 24 25 26 27 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 2 1
2	Corridor	1 2 11 12 13 14 15 16 17 18 19 20 19 18 17 16 15 14 13 12 11 2 1
3	Offices left side	46 45 44 20 21 22 47 48 49 48 47 22 23 24 29 30 31 30 29 24 25 26 27 28 27 26 25 24 23 22 21 20 44 45 46
4	Offices right side	1 2 3 50 51 50 3 4 52 53 54 53 52 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1
5	Elevator	1 2 11 12 43 12 11 2 1

Five paths were defined to analyze different areas of interest in the network, each area has its own difficulties that differ them from the typical environments. The main corridor is traversed by very few and mostly longer links, so the localization might be more inaccurate and the detection rate smaller than in the offices, where more and shorter links are available. In the offices on the right side, many devices are placed higher than in the offices on the left side. On the left side, devices are mostly placed at torso height or slightly above, whereas on the right, they are placed above the height of the head of most people. To analyze this, both areas are surveyed separately. As initial experiments showed, the elevator had a huge impact on shadowing-based RTI, drowning all other changes. As it probably influences variance-based RTI as well, a separate path was used to investigate its impact. The pivot spot in front of the elevator (ID 43) was traversed multiple times, as the experimenter waited until the called elevator arrived at the floor of the testbed. One path traverses all areas of interest, namely path one, to combine different aspects and analyze the overall accuracy.

ToDo: image?



5.3 Localization

The general localization accuracy is evaluated by analyzing the evaluation metrics calculated from the estimated and actual position output by the RTI program. Table 5.3 shows the metrics for the original traces with no changes applied to any parameter.

As the results show, the overall detection rate is low, ranging from 4.7 % to 19.6 % depending on the area, whereas false negative is high, ranging from 23.4 % to 53.5 %. This means, that when a person is walking in the environment, it is rather not detected. False positive is low, indicating that seldom the system detects a person which is not in the environment. The overall accuracy is worse than in typical environments but could still be sufficient, especially for room occupancy, as it ranges from 2.54 m to 2.80 m. The standard deviation σ shows, that the variance is in an acceptable range of 0.86 m to 1.47 m.

Regarding the different areas of interest, the corridor and the elevator provide by far the worst detection rate with 4.7 % and 4.9 %. Additionally, the ratio between false negative and detection rate is worse than in the other areas. This confirms that the corridor has a lower detection rate than the other areas and the arrival of the elevator drowns the changes in RSS triggered by the experimenter. However, the accuracy in both cases is not the worst, so if a position is detected, it is about as accurate as in the other areas. Especially the standard deviation confirms this, as it is lower than e.g. in the offices on the right side.

Comparing the two office areas, the left side actually came off worse than the right side. It has a lower detection rate and worse accuracy, only the standard deviation is better than on the right side. This is reasoned by the placement of the sensor nodes. Not only the density plays a major role, but also the placement. In the offices on the right side, the placement resembles a square geometry, whereas on the left side, most devices are placed on the three sides of the rectangular area. This placement can lower the accuracy. However, in the offices on the right hand side, the devices are placed quite high which results in a higher variance of the detected location.

Table 5.3: Accuracy in meters: mean values for the different paths traversed in the experiments with unaltered traces: $\lambda = 0.1$ and all available connections.

Area	False negative	False positive	Detection rate	Accuracy [m]	σ
Floor	0.535	0.001	0.196	2.57	1.47
Corridor	0.408	0.011	0.047	2.54	0.86
Offices left	0.333	0.000	0.124	2.80	1.03
Offices right	0.234	0.007	0.142	2.60	1.47
Elevator	0.275	0.019	0.049	2.64	1.11

Overall, when all areas are traversed without the impact of the elevator, the experimenter was detected in 19.6 % of the time. The ratio between false negative and detection rate is about 3 : 1, meaning that one of four positions is detected. The error between the actual and estimated position is mostly in a range from 1.10 m to 4.04 m, when the standard deviation is used.



5.3.1 Influence of ellipse width

In this section, the influence of the ellipse width is investigated. All experiments were run with the same set of traces and a larger ellipse width. The initial ellipse width of 0.1 resembled the line-of-sight path between two nodes, whereas now voxels in the nearer surrounding area are taken into account. In this way, more voxels experiencing change in RSS are regarded which could improve the accuracy and detection rate.

ToDo: explanation in 5.1?

The results in Tables 5.4 and 5.5 show, that the detection rate is lower and false negative higher the wider the ellipse, worsening the ratio between these two values. The accuracy and standard deviation mostly improve slightly for $\lambda = 0.2$, but for $\lambda = 0.3$ the standard deviation deteriorates slightly again. However, not all areas profit from a better accuracy, the offices on the left hand side are affected by a worse accuracy and standard deviation.

Table 5.4: Accuracy in meters: mean values for the different paths traversed in the experiments with $\lambda = 0.2$ and all available connections.

Area	False negative	False positive	Detection rate	Accuracy [m]	σ
Floor	0.538	0.001	0.183	2.40	1.43
Corridor	0.415	0.008	0.039	2.47	0.83
Offices left	0.342	0.000	0.116	2.94	1.11
Offices right	0.239	0.005	0.137	2.45	1.38
Elevator	0.277	0.015	0.047	2.57	0.84

Table 5.5: Accuracy in meters: mean values for the different paths traversed in the experiments with $\lambda = 0.3$ and all available connections.

Area	False negative	False positive	Detection rate	Accuracy [m]	σ
Floor	0.543	0.001	0.178	2.37	1.48
Corridor	0.417	0.009	0.037	2.43	0.70
Offices left	0.344	0.000	0.113	2.89	1.03
Offices right	0.243	0.005	0.132	2.45	1.60
Elevator	0.281	0.017	0.043	2.65	0.96

Comprehensively, increasing the width of the ellipse has a low impact on the accuracy and detection rate. The improvements are slight for some areas and for some there is no improvement at all, but the overall detection rate suffers from a larger ellipse.

5.3.2 Influence of long links

As long links provide less information about a person's location and are more influenced by noise, artificially deleting connections between nodes that have a distance larger than a predefined threshold could improve the accuracy and standard deviation.

The traces of all experiments were modified with the link filtering script explained in Chapters 3.3.1 and 4.1.1 and the RTI program run with the modified traces. The results in Tables 5.6 and 5.7 show the impact of longer links. The accuracy and standard deviation is improved for most areas while false negative and detection rate are worse than with all links. The values of the corridor and the offices on the left side are similar to the original.

Longer links distort the localization, as the impact of a person covers a larger area. When these links are filtered out, the accuracy can be improved. However, the detection rate deteriorates slightly, so the maximum link length needs to be chosen fittingly for the deployed system.

Table 5.6: Accuracy in meters: mean values for the different paths traversed in the experiments with $\lambda = 0.1$ and maximum link length 15 meters.

Area	False negative	False positive	Detection rate	Accuracy [m]	σ
Floor	0.546	0.001	0.175	2.45	1.21
Corridor	0.419	0.019	0.036	2.46	0.89
Offices left	0.335	0.000	0.123	2.80	1.01
Offices right	0.246	0.004	0.13	2.47	1.33
Elevator	0.283	0.014	0.040	2.55	1.05

Table 5.7: Accuracy in meters: mean values for the different paths traversed in the experiments with $\lambda = 0.1$ and maximum link length 10 meters.

Area	False negative	False positive	Detection rate	Accuracy [m]	σ
Floor	0.565	0.001	0.156	2.43	1.12
Corridor	0.428	0.002	0.026	2.54	0.85
Offices left	0.338	0.000	0.12	2.82	1.00
Offices right	0.258	0.004	0.118	2.43	1.40
Elevator	0.292	0.012	0.032	2.61	1.05

5.4 Room occupancy

Inferring room occupancy depends on the localization output by the RTI program. With the coordinates, the room the person is in can be estimated. The error in localization can on the one hand be larger, e.g. 3 m, and still the right room is detected. On the other hand, a small error could lead to a wrongly inferred room, indicating one or two rooms next to the actually occupied one. The accuracy indicates, how often the correct room was detected. Small differences of false negative, false positive and detection rate to the original values of localization can occur. The reason is a position estimate that is not in any room of the area, resulting in the fact that the room occupancy program outputs no occupied room.



The accuracy ranges from 67.3 % to 95.1 %. If the experiments with the elevator are left aside, the lower range is at 71.7 %. To evaluate the accuracy better, the localization accuracy and the standard deviation can be examined additionally. The corridor is mostly correctly detected, with a percentage of 95.1 %, as it is a large room and the estimated position has an error not large enough to indicate another room. Additionally, the offices on the left side are inferred correctly with 93.4 %. There the estimated position is mostly in the correct room. In the offices on the right side however, even a similar localization accuracy, when also the standard deviation is taken into account, results in a lower accuracy of room detection, namely 71.7 %. There a smaller error leads to a wrong room detection. This can be reasoned by the device height, so a person walking through these offices does not cause change in RSS at the actual position but also in a different room.

The overall accuracy is 74.8 %, meaning that 3 out of 4 rooms are inferred correctly. This indicates an applicability of RTI for room occupancy. When the rooms are not too small and devices are fittingly placed, this accuracy could even be improved to reach an accuracy similar to the offices on the left hand side, namely 93.4 %.



Table 5.8: Accuracy of room occupancy: mean values for the different paths traversed in the experiments with unaltered traces: $\lambda = 0.1$ and all available connections.

Area	False negative	False positive	Detection rate	Accuracy [%]
Floor	0.527	0.001	0.195	74.8
Corridor	0.408	0.011	0.047	95.1
Offices left side	0.333	0.000	0.124	93.4
Offices right side	0.234	0.007	0.143	71.7
Elevator	0.275	0.019	0.049	67.3

5.4.1 Influence of ellipse width

The results of RTI and room occupancy with a larger ellipse width are shown in Tables 5.9 and 5.10. With a wider ellipse, the accuracy improves while the detection rate slightly deteriorates. When regarding the overall accuracy, the improvement of accuracy is approximately anti-proportional to the loss in detection rate. For $\lambda = 0.2$, the accuracy is enhanced by almost 3 % with a loss of about 1.5 % in detection rate. For $\lambda = 0.3$, the accuracy is bettered by 4 % while the detection worsens of 2 %. As the elevator has a large impact on the RSS, this value is taken aside for the achievable accuracy with a larger ellipse width.

Widening the ellipse improves the accuracy slightly but is accompanied by scarcer detection of a person. In this case, an ellipse width of 0.2 might be a good choice, as the accuracy is improved somewhat while the detection rate is only slightly worsened.

Table 5.9: Accuracy of room occupancy: mean values for the different paths traversed in the experiments with $\lambda = 0.2$ and all available connections.

Area	False negative	False positive	Detection rate	Accuracy [%]
Floor	0.54	0.001	0.181	77.7
Corridor	0.415	0.008	0.039	94.3
Offices left side	0.342	0.000	0.116	96.8
Offices right side	0.241	0.005	0.135	75.2
Elevator	0.277	0.015	0.047	61.1

Table 5.10: Accuracy of room occupancy: mean values for the different paths traversed in the experiments with $\lambda = 0.3$ and all available connections.

Area	False negative	False positive	Detection rate	Accuracy [%]
Floor	0.544	0.001	0.176	78.8
Corridor	0.417	0.009	0.037	96.0
Offices left side	0.344	0.000	0.113	96.7
Offices right side	0.244	0.005	0.132	74.0
Elevator	0.281	0.017	0.043	64.6

5.4.2 Influence of long links

As longer links influence the localization accuracy, they might have a larger impact on room occupancy, as a small error could lead to an incorrectly inferred room. If long links are filtered out, the change in RSS could be condensed to the actual position of a person instead of distorting the changes over multiple rooms.

The results with different maximum link lengths are shown in Tables 5.11 and 5.12. It strikes that with a maximum length of 15 m, the accuracy improves somewhat but deteriorates again with a maximum length of 10 m. Restricting the number of links influences the amount of information in the network. When too many links are filtered out, useful data can also get lost, resulting in a lower accuracy. As the elevator has a large impact on the RSS, this value is taken aside for the achievable accuracy.



For a maximum length of 15 m, an improvement of up to 3.7 % can be noted. On the other hand, the detection rate worsens with a restriction of link length. Regarding the overall accuracy, the detection rate is 2 % smaller while the accuracy only improves 0.2 %. When taking both values into account, limiting the link length does not have a significant enough advantage to take the loss in detection rate.

Table 5.11: Accuracy of room occupancy: mean values for the different paths traversed in the experiments with $\lambda = 0.1$ and maximum link length 15 meters.

Area	False negative	False positive	Detection rate	Accuracy [%]
Floor	0.546	0.001	0.175	75.0
Corridor	0.419	0.004	0.036	97.8
Offices left side	0.335	0.000	0.123	93.3
Offices right side	0.246	0.004	0.13	75.4
Elevator	0.283	0.014	0.04	60.4

Table 5.12: Accuracy of room occupancy: mean values for the different paths traversed in the experiments with $\lambda = 0.1$ and maximum link length 10 meters.

Area	False negative	False positive	Detection rate	Accuracy [%]
Floor	0.57	0.001	0.156	75.3
Corridor	0.428	0.002	0.026	96.7
Offices left side	0.338	0.000	0.12	93.9
Offices right side	0.258	0.004	0.118	75.2
Elevator	0.292	0.012	0.032	56.4

Chapter 6

Discussion

The meaning of this paragraph is to interpret the results of the performed work. It will always connect the introduction, the postulated hypothesis and the results of the thesis/bachelor/master.

It should answer the following questions:

- Could your results answer your initial questions?
- Did your results agree with your initial hypothesis?
- Did you close your problem, or there are still things to be solved? If yes, what will you do to solve them?

Bibliography

- [ABB⁺16] ALIPPI, CESARE, MAURIZIO BOCCA, GIACOMO BORACCHI, NEAL PATWARI and MANUEL ROVERI: *RTI Goes Wild: Radio Tomographic Imaging for Outdoor People Detection and Localization*. IEEE Trans. Mobile Computing, 2016.
- [Adv] ADVANTICSYS: *MTM-CM5000-MSP*. <https://www.advanticsys.com/shop/mtmcm5000msp-p-14.html?zenid=8767fe0f4652c32f4e8b7ad33519f390>. Accessed: 2017-01-13.
- [KBP12] KALTIOKALLIO, OSSI, MAURIZIO BOCCA and NEAL PATWARI: *Enhancing the Accuracy of Radio Tomographic Imaging Using Channel Diversity*. 9th IEEE International Conference on Mobile Ad hoc and Sensor Systems, 2012.
- [LG] LEVIS, PHILIP and DAVID GAY: *TinyOS Programming*, Cambridge University Press. cs1.stanford.edu/~pal/pubs/tos-programming-web.pdf. 2017-01-28.
- [Mem] MEMSIC, INC.: *TelosB Datasheet*. http://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0094-02_B_TELOSB.pdf. Accessed: 2017-01-03.
- [MPB13] MAGER, BRAD, NEAL PATWARI and MAURIZIO BOCCA: *Fall Detection Using RF Sensor Networks*. IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications, 2013.
- [WP10a] WILSON, JOEY and NEAL PATWARI: *Radio Tomographic Imaging with Wireless Networks*. IEEE Trans. Mobile Computing, 2010.
- [WP10b] WILSON, JOEY and NEAL PATWARI: *See-Through Walls: Motion Tracking Using Variance-Based Radio Tomography Networks*. IEEE Trans. Mobile Computing, 2010.
- [ZJ09] ZHENG, JUN and ABBAS JAMALIPOUR: *Wireless Sensor Networks: A Networking Perspective*. John Wiley and Sons, ISBN 978-0-470-16763-2, 2009.
- [Zur] ZURICH, TIK WSN RESEARCH GROUP @ ETH: *Tmote Sky*. <http://www.snm.ethz.ch/Projects/TmoteSky>. Accessed: 2017-01-13.

Erklärung

German

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe.

English

I hereby declare that I have written this Bachelor thesis independently, using no other than the specified sources and resources, and that all quotations have been indicated.

Essen, January 28, 2017

(Place, Date)

Sarah
Theußen

Viktoria