Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Ryan Thomas Rybarczyk

Entitled

Managing Trust and Reliability for Indoor Tracking Systems

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Dr. Rajeev R. Raje
Chair

Dr. Mihran Tuceryan

Dr. Yao Liang

Dr. Fengguang Song

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Dr. Rajeev R. Raje

Approved by: Dr. Shiaofen Fang 11/28/2016

Head of the Departmental Graduate Program Date

MANAGING TRUST AND RELIABILITY FOR INDOOR TRACKING SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Ryan Thomas Rybarczyk

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2016

Purdue University

Indianapolis, Indiana

To my loving wife Amy

ACKNOWLEDGEMENTS

I would like to start off by thanking my professor, advisor, and mentor, Dr. Rajeev R. Raje, for allowing me the opportunity to conduct research under his outstanding guidance. His support through this process has helped shaped me into both the researcher and educator I am today. I am forever grateful for his encouragement through the process of preparing this thesis. I would also like to extend a special gratitude to my co-advisor, Dr. Mihran Tuceryan, as his guidance and expertise helped provide excellent insight during my research pursuits. I would like to thank both Dr. Yao Liang and Dr. Fengguang Song for their role in graciously serving on my committee.

I would like to thank my family, most notably my parents – for without their everlasting love and support I would not have been able to achieve this goal in my life. I would also like to thank my loving wife, Amy, for her support, guidance, and most notably her patience throughout the entire process. Often times my work consumed my time but she was always there for me and supportive – and for that I forever grateful.

I would like to thank the entire Computer Science Department at IUPUI. They provided a wonderful environment that allowed me to explore my various passions and helped foster my ability to learn and grow as an individual. The many members of the department who have supported this pursuit over the years are too numerous to count, but they all will hold a special place in my heart. I would be remiss if I did not mention

all of my fellow students and colleagues who have shared this journey with me – those that have both come before me in this pursuit and those still on the journey. We have all grown together and learned from one another. Thank you!

Finally, I would like to give thanks to God for blessing me with the vast multitude of wonderful people in my life that have helped me along this journey. He has blessed me beyond anything I could ever imagine. I am eternally grateful for all that He has provided in my life to make this possible.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

ITS	Indoor Tracking System
eDOTS	enhanced Distributed Object Tracking System
sKB	Sensor Knowledge Base
TA	Trust Agent
RA	Reliability Agent

ABSTRACT

Rybarczyk, Ryan Thomas. Ph.D., Purdue University, December 2016. Managing Trust and Reliability for Indoor Tracking Systems. Major Professor: Rajeev R Raje.

Indoor tracking is a challenging problem. The level of accepted error is on a much smaller scale than that of its outdoor counterpart. While the global positioning system has become omnipresent, and a widely accepted outdoor tracking system it has limitations in indoor environments due to loss or degradation of signal. Many attempts have been made to address this challenge, but currently none have proven to be the *de-facto* standard. In this thesis, we introduce the concept of opportunistic tracking in which tracking takes place with whatever sensing infrastructure is present – static or mobile, within a given indoor environment. In this approach many of the challenges (e.g., high cost, infeasible infrastructure deployment, etc.) that prohibit usage of existing systems in typical application domains (e.g., asset tracking, emergency rescue) are eliminated. Challenges do still exist when it comes to provide an accurate positional estimate of an entities location in an indoor environment, namely: sensor classification, sensor selection, and multi-sensor data fusion. We propose an enhanced tracking framework that through the infusion of QoS-based selection criteria of trust and reliability we can improve the overall accuracy of the tracking estimate. This improvement is predicated on the introduction of learning techniques to classify sensors that are dynamically discovered as part of this opportunistic tracking approach. This classification allows for sensors to be properly identified and evaluated based upon their specific behavioral characteristics through performance evaluation. This in-depth evaluation of sensors provides the basis for improving the sensor selection process. A side effect of obtaining this improved accuracy is the cost, found in the form of system runtime. This thesis provides a solution for this tradeoff between accuracy and cost through an optimization function that analyzes this tradeoff in an effort to find the optimal subset of sensors to fulfil the goal of tracking an object as it moves indoors. We demonstrate that through this improved sensor classification, selection, data fusion, and tradeoff optimization we can provide an improvement, in terms of accuracy, over other existing indoor tracking systems.

CHAPTER 1. INTRODUCTION

Tracking is a fundamental behavior that we, as humans, possess with respect to our environment and given locations. We have an innate desire to track objects as they move, or are moved, around within an environment. This movement takes place over a diverse scale of distances, durations, and settings. This idea of tracking serves many practical purposes that are often essential to our daily lives. Knowing where specific objects (e.g., people, places, and things) are located and how to find/discover these objects are often of great importance to us. Location awareness and tracking of objects can also yield additional, and often interesting, context regarding the current situation and the environment. For instance, one common form of tracking that has found itself as a mainstream necessity is for individuals to use the global positioning system (GPS) to track the movement of their vehicle as they drive in order to provide directions on how to reach a certain destination. This system has the ability to locate and then track a GPS sensor as it moves and because of this tracking ability the system can provide the shortest path and can help us to avoid potential “roadblocks” that would often hinder our ability to reach our goal.

The above example demonstrates not only the usefulness of tracking but also the need and reliance on such a system. One of the key points to note in the above provided and belief that the system will behave in expected manner. This trust has been built-out of positive evidences, or its reputation, that have been demonstrated over time and validated through extensive studies and use. At the same time, the individual also has a belief that the system will behave in a reliable fashion – meaning that the system will consistently provide the necessary information when requested and will not fail. In the example case provided, a failure of the system may not prove to be catastrophic but it could result in significant damage (both physical and reputation) if complete belief is placed with the system and its performance. This demonstration of the importance of trust and reliability with respect to the evaluation and use of a tracking system are vital for its wide spread use.

1.1 Motivation

Indoor tracking often comes with a significant cost attached to it. This cost is twofold: one aspect of cost is the financial cost associated with the purchase and then deployment of a physical sensing infrastructure to provide such tracking; secondly, the cost associated, with time and energy consumption. The second component of cost, in terms of energy consumption, is especially important since many sensors that are now being used for indoor tracking are mobile and/or running on battery power and thus have strict energy constraints. This problem is further compounded by the fact that the devices (containing the tracking sensors) are typically not dedicated solely for the use as tracking sensor, but instead are designed and used as personal mobile devices by the individual carrying them. As a result, careful attention must be paid to the cost associated with obtaining a positional estimate.

A secondary challenge associated with indoor tracking at the general level is that the tracking system itself may have little control over the behavior or movement of the sensors. Devices, and their associated sensors, may enter or leave the tracking environment, and thus the tracking sensor network, often may change without proper notice. This behavior may be planned (e.g., leaving a room) or unplanned (e.g., the battery of the mobile device has failed and thus has to shutdown). This unpredictability and dynamic mobility of the tracking infrastructure makes it challenging to identify which devices/sensors to make use.

Finally, as part of this tracking process there is the tradeoff between cost and gain that must be examined. The goal of a tracking system is to provide a maximal gain while minimizing the cost associated with obtaining this result. This process is not trivial and due to the dynamic nature of both the tracking environment and the sensors themselves it can be quite challenging. Optimizing this tradeoff is there a difficult problem and one that must be addressed in order to maximize the overall performance of an indoor tracking system.

1.2 Problem Statement

Existing indoor tracking approaches [1] are typically focused on single modal systems that do not consider the dynamic nature of the tracking environment. Hence, there are extreme limitations with what and where indoor tracking in these systems can take place. With the omnipresence of mobile and smart devices, a static single modal sensing infrastructure is not always required, nor necessary, in order to provide adequate indoor location tracking. Instead, an approach of opportunistic tracking, in whatever sensor infrastructure is available is used, is now possible in which indoor tracking systems may dynamically and opportunistically discover and take advantage of any sensor devices present in a given environment.

This thesis proposes the infusion of both trust and reliability as separate, and distinct, selection criterion for the purpose of improving the sensor selection and ultimately the accuracy of the tracking estimate. This improved selection will directly impact the data fusion process and ultimately the overall tracking estimate provided. As part of this work it is also necessary to explore, evaluate, and propose a solution to the tradeoff between the cost and the gain of such selection. This thesis will propose an optimization of this tradeoff with the intent of improving the overall performance of a prototypical indoor tracking system.

1.3 Hypothesis

The goal of this thesis is to demonstrate that indoor tracking performance, in terms of tracking accuracy, can in fact be improved through the infusion of new selection criteria (trust and reliability as separate criterion) and that this selection process can then lead to an improved performance while providing a tradeoff between the cost and the gains.

1.4 Indoor Tracking

Indoor tracking is a fascinating area of research due to its many significant challenges. Indoor tracking differs greatly from its outdoor counterpart due in part to the fact that the typical spaces, or areas, in which the tracking is taking place are much more confined and often times need a greater accuracy and responsiveness due to the close proximity of objects. An additional challenge that is magnified by indoor tracking is that of the presence and scale of obstacles that can obstruct or greatly negate the effectiveness and the use of certain popular sensing technologies (e.g., wireless signal propagation and thick concrete walls). Because of this, often a single sensor modality (e.g., satellite time distance arrival – GPS) will not suffice in providing the necessary coverage and associated accuracy needed to track in such indoor environments. Instead, a combination of different techniques and sensors are often needed to provide the desired or complete coverage with appropriate accuracy and end-to-end tracking time.

Existing static infrastructure [2] and single modal systems [3] are popular approaches to provide indoor tracking. These approaches are similar to that found in the outdoor tracking environment and attempt to leverage many of the same techniques while providing the tracking coverage. Such tracking infrastructure is often integrated into indoor environments to provide tracking through specialized installation and the use of tracking-specific sensing devices. One popular indoor tracking technology that is currently in use is that of radio frequency identification (RFID). RFID technology makes the use of specialized attached identifiers to help pinpoint an object's given location by exchanging of radio signals with a base station, otherwise known as a reader. This technology has proven to be extremely popular with respect to asset tracking. Another related technology is that of wireless frequency (Wi-Fi) fingerprinting and subsequent trilateration to determine an object's position. Each of these technologies allows for the tracking of objects in indoor spaces, however, both are tied to a specific modality and a specialized infrastructure.

With the current omnipresence of smart, often mobile, devices it is now possible to create *ad-hoc* networks of sensors within the indoor boundaries for tracking. These smart devices often carry many onboard sensors and can assist in tracking in areas where static infrastructure may not always be feasible. These mobile sensors can also help to enhance existing tracking infrastructure by providing additional sensing technologies that can be harnessed to better pinpoint an objects given location. The mobility of these devices also provides a distinct advantage in tracking as tracking itself is the task of "following" an object's movement. With the freedom of movement, the tracking system can evolve with the current needs of the application domain and provide a much wider area of coverage.

1.5 Sensor Classification

For indoor tracking, one of the key challenges is the processing of "discovering" the sensors in the environment. In a single modal system this is a trivial task as there is only one type of sensor to account for. However, in a multi-modal system this task can be quite a challenge and can require a substantial amount of time to complete. This is due to the inherent complexity of dealing with a wide range of sensors and their varying characteristics. The sensor classification problem is, therefore, represented as the attempt to label the modality of an identified sensor. This overall process can be broken down into two distinct phases: identification and classification.

In the first phase, the sensor must be properly identified as a consumable sensor. Here the term consumable is used to represent the ability of a software component to access the physical sensor itself. The tracking system must first recognize that a given sensor is present and establish communication with the device. This identification process is the initial step in making a connection with the device that contains the physical sensor itself. Once a connection and communication has been made with the sensor and that it is deemed that the sensor is indeed a consumable sensor, it is then possible to move on to the second phase of the process.

The second phase consists of the actual classification of the sensor modality. This is an important step as it separates the trackers, or those sensors that can provide data with respect to an object's location, from non-trackers. This phase of the classification activity can be defined as the attempt to match a sensors characteristics with known information, if/when this information is available. There are three unique cases in which this matching can be performed by the tracking system. Any one of these three cases can provide the necessary means classify a sensor. These three different cases are now described below in more detail.

The first case of sensor matching is built upon the idea that information regarding the identified sensor is publically available, either via an existing knowledge base or through manufacturers publications, and therefore the sensor can be matched based upon this previously acquired information. The second case of sensor matching is that some of the information may have been previously acquired, by *a priori* interaction with the tracking system, that can then serve as a base for matching the identified sensor with a given sensor template. This knowledge base can be constructed over time and be maintained by the system administrator. Finally, the third case of sensor matching is the instance where there is no information known about a given sensor. In this case, the tracking system will attempt to match with a generic template until such information can be collected and a new sensor identification created as part of the known world of tracking sensors.

This process of sensor classification relies heavily on the tracking system and its ability to learn about the presence of new sensors, be able to build/update sensor knowledge base, and to match the sensor with a given template. Once a sensor has been properly identified and classified it is now ready for use by the tracking system.

1.6 Sensor Selection

The problem of sensor selection can be defined as the process of selecting a sensor, S_i, from a given set of sensors S, so as to yield the highest benefit in indoor tracking. This process could be repeated until some criteria are met, ultimately resulting in an optimal subset of sensors.

In order to achieve this decision making process of the optimal subset selection, a set of selection criteria is often implemented to aid in this process. These selection criteria, discussed in the following subsection, allow for the selection process to evaluate and make a judgment based upon the comparison between the criterion and the sensor's performance. Other approaches that do not make an explicit use of specialized selection criteria can instead make use of counting or random selection techniques. In a counting technique the first/last *N*-number of sensors can be selected to serve as the subset. In a random-based approach, the system itself randomly selects a given subset of sensors. Sensor selection serves many roles in the overall indoor tracking process. The first role is that it provides the system with a set of sensors with which to work with in order to estimate a given position of an object at a moment in time. In the case of a multi-modal tracking system, sensor selection process must lean heavily on the sensor classification to know the behavior and expected performance of each sensor. This is important as a sensor may be very well suited for a specific environment or specific conditions but may provide unsatisfactory in terms of its performance in others. This decision making process has a ripple effect throughout the entire tracking system as the selection of one improper sensor can negatively impact the overall accuracy of the tracking system.

1.7 Sensor Criterion

A proper subset of sensors is often needed in order maximize the end goal of the tracking system. As previously mentioned, during the sensor selection process, a filter may be applied in order to select those sensors that meet a specified criterion and thus provide accurate tracking. This evaluation of selection criteria must be dynamic and evolve as both the system and the tracking requirements evolve.

As highlighted in the introduction, trust and reliability are often two important selection criteria during the process of selection. In an indoor tracking environment, there may be different sensors (either of the same modality or different) that the tracking system may need to interact with during the course of tracking an object. These sensors may be known *a priori*, however, this cannot be assumed to always be the case. Even in the case where the sensors are all known *a priori*, their specific behavior and subsequent performance and characteristics (e.g., communication rate, sensor life, etc.) may change, or be altered/manipulated, over time. We will now provide a brief introduction to the concepts of trust and reliability.

1.7.1 Trust

Trust can be defined [4] as the belief that an entity will behave in a certain specified fashion over a given period of time, t. This concept of belief is an important aspect of the trustworthiness of a specific entity. This definition can be applied to the domain of indoor tracking as a selection criterion for use in the sensor selection process.

Belief, and hence trust, in a sensor can be established through its reputation (e.g., other sensors opinions about the trustworthiness of a given sensor S_i) or through interactions with the sensor, by software trust agents, in which direct evidences of this behavior can be collected. In addition, while computing trust about a sensor, the reputation (i.e., the general held consensus as to the belief of an entity by a collective group) of the providers of the opinions must also be considered. Operators, such as consensus, to quantify trust associated with a sensor can be used to evaluate these collected opinions and evidences over a given time period. This trustworthiness can then in turn be used as a part of the evaluation of the selection criteria during the sensor selection process.

1.7.2 Reliability

Reliability can be defined [5] as the probability of failure-free operation for a specified period of time, t, for a given entity. In this definition, the key term of examination is failure-free. This term is the basis for distinguishing trust and reliability from one another. A failure-free operation does not ensure that the data provided by the sensor can be deemed as trustworthy; all that it ensures is that the sensor behaves without failing during the course of the tracking exercise. With respect to indoor tracking sensors failure can either be a hard or soft failure. In a hard failure, the actual physical device itself (i.e., the sensor) could mechanically malfunction and thus prevent the device from providing the necessary response. In a software failure, the software component of a sensor may fail either unintentionally or through malicious intent. This failure is not limited to just the sensor itself but can also include the communication network that is being utilized between the tracking system and the sensor.

1.8 Multi-Sensor Data Fusion

Data fusion can be defined [6] as the process of combining data from different sources into a single point of reference. In a multi-sensor tracking environment this can be further defined as the act of combining, or fusing, results from different sensor sources with the intent of tracking an object as it moves through an environment. The act of fusion is a multi-step process that includes: examination of the data and data sources, determining a singular point of reference, selecting a fusion technique, and presenting the final fused result.

In the first step, the tracking system must examine the data obtained from various data sources and decide how to use the data. This process is aided by sensor selection activity. Once the data has been selected, then these data items need to be combined. There is a degree of heterogeneity associated with various data items, for example, data results that provide two-dimensions of reference are not equivalent to results that provide three-dimensions of reference. Hence, such items need to be unified using a single point of reference. After a single point of reference is established the next step is to select a fusion technique to be applied. Techniques for data fusion range from an averaging technique to Kalman filtering. Simple techniques are efficient in terms of cost, both computational and time, but are often not sufficient as noisy sensor data can greatly skew the resulting estimate. Complex approaches such as Kalman-based Filtering are more costly; however, they can achieve much more precise and accurate results.

1.9 Tradeoff Optimization

Tradeoff optimization can be defined as the ability to measure and then optimize the tradeoff between gain and cost. In the case of indoor tracking the gain is typically measured by the accuracy obtained by the tracking system. For indoor tracking the cost can be measured via two parameters: the time and computational overhead associated with obtaining the positional estimate, as well as the cost associated with deploying sensors into a given indoor environment. As cost and gain are opposites of one another there is a need for optimizing this tradeoff with the intent of maximizing the gain while minimizing the cost.

Specifically for indoor tracking, the need to optimize this tradeoff serves two purposes. Firstly, we strive to provide the highest degree of accuracy with our positional estimate. This accuracy is a direct result of the sensors selected and the data fusion applied to the raw sensor data; second, we strive to minimize the cost, specifically time and network communication or bandwidth, to help minimize the impact when it comes to the sensors themselves. As previously mentioned, tracking is not a static process and as such the optimization must be dynamic. Learning techniques can be applied in order to adjust the optimization, as needed, during the course of tracking. This learning – and then application of the learned behavior – is essential as the dynamic nature of both the virtual and physical environments change over the duration of the tracking process.

1.10 Contributions

The formal contributions of this thesis are as follows:

1. We introduce and formalize the concept of opportunistic tracking. This proposed approach makes use of any available sensing device present in a given indoor environment. This proposed approach includes the discovery, identification, and communication with such devices and the formation of a network of tracking sensors.

2. We propose a classification algorithm and preliminary online repository for a knowledge base containing information regarding various sensor characteristics and performance ratings. This knowledge base will be constructed over time and referenced when needed in order to appropriately classify a sensor.
3. We propose a modified sensor selection technique in which trust and reliability are used as separate and distinct selection criterion. Trust and reliability are measured through the collection of evidences and are calculated using the concept of subjective logic to model the belief, disbelief, and uncertainty in the resulting value.
4. This thesis proposes an optimization tradeoff function in which we analyze the impacts on both cost and gain with respect to sensor selection and the ultimate performance of the tracking system. This tradeoff evaluation is then enhanced through the application of learning techniques that when applied can dynamically adjust the necessary function over the duration of tracking.
5. We apply each of these proposed enhancements and techniques to a prototype indoor tracking system for empirical validation and analysis. This is done with the aim of demonstrating that through the inclusion of the additional selection criterion of trust and reliability and the use of dynamic optimization of such selection that it is possible to improve the overall performance of a typical indoor tracking system and thus validate our hypothesis.

1.11 Organization

This thesis is organized into five chapters. The first of these chapters includes the introduction, an overview of the different aspects of indoor tracking, a discussion of the motivating factors behind this work, a problem statement outlining what problem this thesis document sets out to solve, and a hypothesis for the expected outcome of this work. The second chapter includes a comprehensive review of related works. The third chapter presents an outline of the proposed modifications to the sensor classification process, sensor selection process, infusion of trust and reliability, analysis of the data fusion process, and outlines the optimization function that will be used to evaluate the tradeoff between the cost and the gain in the tracking system. This chapter includes a discussion of the design of the indoor tracking system that will be used for experimentation. The fourth chapter describes the results from experimentation and an in-depth discussion of these results and how they relate to the problems and goals discussed in chapter one. The fifth and final chapter states conclusions of the work as well as some suggested areas of future work.

CHAPTER 2. RELATED WORK

There are many application domains [7-10] in which tracking of an object as it moves through an indoor environment is a requirement. In many cases, this tracking must take place with a high degree of accuracy and either at, or near real-time. There are many fundamental challenges that have been identified [11] with respect to the ability to track an object in an indoor environment. These challenges include: sensor classification, sensor selection, data fusion, accuracy, and cost (in terms of time, energy, and financial undertaking). Many solutions have been proposed that attempt to address these challenges; however, these challenges still exist and serve as the basis of this work.
We will now explore the specific areas of related work that this thesis focuses on addressing. We will examine work done in the following areas: sensor classification, sensor selection, trust, reliability, data fusion, and tradeoff optimization. We will begin with a general overview of indoor tracking systems and the techniques and technologies that they utilize for providing indoor tracking, thus setting the stage for the aforementioned discussions.

2.1 Related Work in Indoor Tracking

Many commercial indoor tracking systems [1] have been developed. Despite of many such efforts no single system, technique, or technology has gained widespread acceptance in the same fashion that the GPS has done for outdoor environments. Instead, a wide array of different technologies and techniques have been proposed [1] with the hope of improving the overall tracking accuracy for indoor environments.

Wireless technologies have been the popular focus of many of the existing proposed indoor tracking systems. This popularity stems from the omnipresence of such signals in our daily lives. Such signals are not subject to many of the problems or challenges that other tracking sensors face (e.g. line of sight) and thus are able to provide tracking where others sensors cannot. One such wireless technology that has been used for tracking is that of radio frequency used by RFID. RFID is a popular technology that is often found in asset tracking [12]. In this approach, “tags” are attached to an object and then are tracked based upon identification of these “tags” through a wireless radio signal. These “tags” can be either active or passive. Active tags require an on-board power source in order to transmit their signal to a corresponding base station. Passive tags do not require this on-board power source; instead they are able to transmit their signal only through close contact with a “reader” device. This close contact is necessary in order to retrieve the data from the sensor itself. Many popular systems [1] have been created that make use of both active and passive RFID tracking approaches. A few prominent ones are described below.

In [13], the authors have proposed an indoor tracking system that makes use of passive RFID tags in order to track an object as it moves through an indoor environment. They achieve this tracking through the deployment of passive tags throughout the environment and then affixing a RFID reader to the object that is being tracked. While this approach provides high accuracy, the overhead associated with obtaining and deploying the tags within the environment can be quite costly as active RFID tags can be

$15 USD or more depending on the quality of the tag. It may also not be feasible to deploy such tags throughout an environment. In order to complete this task, the tracking environment must be known *a priori* and the physical locations of each passive tag must be known for referencing a location. In [14], the authors propose a system in which active tags are used to estimate the positon of an object. This approach provides the ability to disperse objects, with active tags attached, into an environment and then use the feedback from the active tags to calculate the position of an object. The authors do however make note that the accuracy of this approach may not be suitable for all application domains (up to 45 meters of error). Also, due to the cost of active RFID tags, this approach may not always be feasible to deploy – in an indoor environment. In [15], the authors propose a system that uses a combination of both active and passive tags in order to enhance the overall tracking coverage possible. While this approach can provide improved coverage and accuracy over a standard active approach, noted in [14] – through the inclusion of passive tags, it presents the same challenges that both systems encounter independently (i.e., *a priori* knowledge and high cost of sensors).

The use of Wi-Fi technology has also proven to be a very popular technique for indoor tracking. A possible reason for its popularity and wide-spread usage is the pervasiveness of publically available wireless access points (AP) and devices that contain a wireless network cards. The use of Wi-Fi for indoor tracking has been widely commercialized over the years. The most prevalent of these developments has been the work conducted by Google, Inc. in their indoor mapping application, Google Indoor Maps [3]. Google Indoor Maps makes use of existing wireless infrastructure in order to map and then subsequently track wireless devices as they move about indoor environments. The process of tracking is through identification of known locations of stationary AP’s. This information regarding “discovered” AP’s is stored in an online database that can then be referenced by the application for the purpose of estimating the position of the wireless device.

When it comes to wireless-based indoor tracking, there are two prominent approaches that are used to determine an objects’ position: trilateration and fingerprinting. Distance-based trilateration, as described in [16], is the technique of obtaining a position of an object based upon the calculated distance the object is from at least three AP’s. Using these known locations of the AP, an estimate can then be made about the position of the object within the indoor environment by finding the area of overlap between the wireless signals.

This is positional estimate is achieved through the use of the following equations:

$$d_i = \sqrt{(x_i - X_1)^2 + (y_i - Y_1)^2 + (z_i - Z_1)^2}$$

$$A\vec{x} = \vec{b}$$

$$A = 2\begin{bmatrix} (X_2 - X_1) & (Y_2 - Y_1) & (Z_2 - Z_1) \\ (X_3 - X_1) & (Y_3 - Y_1) & (Z_3 - Z_1) \\ (X_4 - X_1) & (Y_4 - X_1) & (Z_4 - X_1) \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} (X_2^2 - X_1^2) + (Y_2^2 - Y_1^2) + (Z_2^2 - Z_1^2) - (d_2^2 - d_1^2) \\ (X_3^2 - X_1^2) + (Y_3^2 - Y_1^2) + (Z_3^2 - Z_1^2) - (d_3^2 - d_1^2) \\ (X_4^2 - X_1^2) + (Y_4^2 - Y_1^2) + (Z_4^2 - Z_1^2) - (d_4^2 - d_1^2) \end{bmatrix}$$

Where d_1, d_2, d_3, and $d_4$are the distances between the known AP and the wireless device and (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), (X_3, Y_3, Z_3), and (X_4, Y_4, Z_4) are the known coordinates of the AP.

$$\vec{x} = [x \quad y \quad z]^T$$

$$\vec{x} = (A^T\, A)^{-1}\, A^T \vec{b}$$

The above equations indicate the estimated position of wireless device with respect to the analysis between the signal strength and the wireless AP.

The second popular wireless-based tracking technique is that of fingerprinting [17]. The goal of this process is to create a radio mapping of the environment based upon the received signal strength (RSS) values of known AP and the locations at which these values are collected in terms of an (x,y) pair. This coordinate point is derived from a predefined origin that is established during this calibration process of the AP. During the tracking process, the received RSS values from the device are then compared with the values collected during calibration and a probability value is used to select the estimated position which best matches the current state. This two phase approach, consisting of the offline training and calibration and the online matching, is an expensive task in terms of time. This technique also requires substantial *a priori* mapping, or a known environment and infrastructure, and calibration in order to achieve accurate tracking and therefore is not always feasible.

In [18], the authors conduct an empirical study comparing the techniques of fingerprinting and trilateration. It is noted, by the authors, that the expensive cost of the offline phase of fingerprinting often makes it unfeasible for most indoor environments and as a result an impractical technique for most application domains. Trilateration, while still requiring the additional domain knowledge of known AP, allows for higher degrees of accuracy and confidence while reducing the overhead needed for calibration with the existing wireless infrastructure. The authors also note that in the average case, an accuracy of between three and five meters can be achieved through the use of wireless signals for location accuracy.

Bluetooth-based indoor tracking is an extension of the wireless-based tracking techniques. It makes use of a similar approach to that found when using Wi-Fi for calculating the estimated position of an object within an indoor environment. In [19], the authors provide a comparison, in terms of accuracy, with a Wi-Fi-based approach and a Bluetooth Low Energy (BLE) fingerprinting. The author's note that improvements can be made in terms of applying the BLE technique, however this approach is reliant upon additional computation (e.g., smoothing) as well as additional beacons, which are often necessary to obtain this increase in accuracy over Wi-Fi. In [20], the author found that the cost associated with obtaining the necessary accuracy, in terms of time overhead – due to the sampling rate of many Bluetooth enabled devices, was insufficient for many application domains. This work demonstrated the limitations of this as a viable tracking technique.

Vision-based tracking is another popular approach for indoor tracking. In this approach, cameras (with video processing capabilities) are utilized as tracking sensors in which the tracking object must be identified, visually, in order to estimate the objects location. This method of tracking can be broken down into two separate approaches: inside-out and outside-in. Inside-out tracking is a technique in which the location of the vision-based tracking sensor is stationary, and its position with respect to the environment is known, while the object being tracked and its position are unknown and therefore must be estimated. Outside-in tracking is a technique in which the location of the vision-based tracking sensor is unknown and must be estimated by using the known position of an object or landmark within the indoor environment. Traditionally, the inside-out tracking technique was the more popular of the two; however, with the emergence of mobile devices, containing cameras, it is now possible to build networks of vision-based trackers using the outside-in technique.

In [21], the authors propose a tracking system for use in the field of health informatics. The Information Technology for Assisted Living at Home (ITALH) project is a camera-based tracking system that has been designed to monitor the elderly who live without the need for a nurse. In this tracking system, occlusion, or the obstruction of the cameras view, is one of the primary problems outlined during the tracking process. The primary focus of this work is to resolve the issue of occlusion and to devise algorithms to handle such an occurrence while still providing the necessary tracking of an object.

In [22], the authors attempt to look at tracking various marker objects within a Vision-based distributed tracking system. The authors propose the use of a Kalman-based technique, a Kalman Consensus filter, which utilizes neighboring cameras in order to form a consensus as to the actual physical state of the marker object. It is suggested that through the use of this technique, the cameras within the system are able to be self-aware and self-organizing. This allows the cameras the ability to learn the network topology over the course of the tracking process. This, as a result, allows for improved tracking accuracy and the ability to handle dynamic changes within the environment.

The final set of sensor modalities that we will discuss are inertial sensors. Inertial sensors are sensors that continuously monitor the movement from a set location. These sensors are commonly represented on smartphones as accelerometers, gyro, etc. In [23], the authors propose an inertial-based tracking system which does not require any environmental infrastructure. The authors demonstrate, through a dead-reckoning technique in which check points allow for the recalibration and correct of the location estimate, that high accuracy can be obtained through the use of such sensors. This work demonstrates the feasibility of such a system and concludes by showing an average accuracy of between 1.5 and 2 meters. In [24], the authors use a similar approach by making use of a variety of sensors on board a typical smartphone in order to provide highly accurate tracking estimates.

In this thesis, we are not restricting our work to a single sensor modality or a single technique. Instead, we provide a framework that uses any available tracking infrastructure. We define this approach to be the concept of opportunistic tracking [25]; in which the tracking system opportunistically discovers available sensors and uses them to provide an estimate of an object's location in an indoor environment. This highly flexible and dynamic approach to tracking can help to reduce many of the previously mentioned challenges (e.g., cost of deployment, feasibility of sensor deployment, etc.) when it comes to a specific sensor modality or tracking technique.

2.2 Related Work in Sensor Classification

Classification is the task of attempting to properly identify an entity based upon a set of acquired knowledge. This task plays key roles within the scope of indoor tracking; from classifying the sensors that will perform the tracking, to the classification of the objects being tracked. Hence, achieving such classification in both an accurate and timely fashion is of the utmost importance. We will begin this discussion with an overview of classification techniques and then discuss how these techniques can be made applicable to the sensor classification problem for indoor tracking.

In [26], the authors define classification to be: "the problem of identifying to which of a set of categories (subpopulations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known." Using this definition, as a guide, it is then possible to produce two distinct steps in the process of classification: training and predicting. The authors discuss how, during the training phase, a mapping function can be applied to match features with defined labels. Once this training process is completed, the process of predicting can then begin. The role of predicting is then defined by the attempt to find a trained feature set that matches the actual entity.

There exist many classification methods that are described in relevant literature. We will focus on the classification methods that can be broadly categorized as Supervised Learning. These methods of classification include the following different described techniques: decision trees, linear classifiers such as Naïve Bayes and support vector machines (SVM), and neural networks. We will now briefly summarize these techniques and discuss how they can be directly related to sensor classification problem for indoor tracking.

A decision tree is a graph in which decisions and their associated consequences are modeled as nodes within the graph [27]. Based upon the choices made when traversing the graph, a conclusion will be made based upon a matching probability between the information found during the traversal. The strength of this process is the recurring nature of the choice selections. In [28], the authors propose a technique in which they apply decision tree classifiers to help in identifying RFID sensors. The purpose of this classification is to provide the set of RFID sensors that can provide the highest degree of accuracy when tracking for the purpose of providing indoor localization in typically “unfriendly” environments. The authors note that through the use of this technique, they are able to obtain an improvement of up to 98% of accuracy in terms of room location.

Closely related to decision trees are rule-based classifiers. Rule-based classifiers are algorithms that are closely related to decision trees. Instead of being represented as a tree they have been translate to a set of rules that can be then evaluated on [29]. An important feature of these algorithms is that while they are derived from the decision tree structure, they are capable of incorporating additional rules. This annotated approach can provide improved decision making on the application of the rules during the traversal process. The goal of this approach is to find the “best” rule that satisfies the given specification and then to classify the entity based upon the matching rule. If no rule can be found, that can satisfy this request, then a new identifier is added, through annotation, to the rule set for future reference.

In [30], the authors have proposed a fuzzy rule-based multi-classification system for topology-based Wi-Fi Indoor localization. This use of a fuzzy rule-based approach allows for the ability to model the uncertainty, and more specifically the unpredictable characteristics, of Wi-Fi signals in typical indoor environments. The authors note, that through this use of this rule-based approach the classification outperformed a comparable nearest neighbor algorithm with a lower execution time. In [31], the authors also make use of a fuzzy rule-based approach to classify the location that a specific device is currently located in. They have found that through the use of such techniques, it is possible to provide an accuracy of up to 90% as to the correct location of a device. This accuracy however does not reflect the actual physical location of the device but rather provides a general idea of the indoor space, or environment, in which the device is currently located (e.g. the mobile phone is in room SL 116).

A linear classifier makes a decision based upon the combination of different characteristics [29]. Characteristics are made up of the attributes of a given entity. Further features can be derived from this attribute set and matched through the combination of the various characteristics presently available. Two common methods are used for determining the parameters of the linear classifier: generative and discriminative. In the generative approach, the algorithm models the conditional probability distribution. A Naïve Bayes classifier is an algorithm that falls under the umbrella of generative. In the discriminative approach, the algorithm attempts to maximize the output of the classification process based upon the training set. A support vector machine (SVM) is an algorithm that falls under the umbrella of discriminative.

In [32], the author provides an empirical study of the Naïve Bayes classifier. The author demonstrates the effectiveness of the approach by highlighting the accuracy obtained in the classification process. In [33], the authors propose the use of a Naïve Bayes approach for classifying Wi-Fi fingerprints for indoor tracking. The purpose behind this approach is to correctly identify the proper set of wireless AP's for us in identifying the room in which the object is currently in.

A SVM is a supervised technique in which a model is constructed that assigns new examples into one category or the other. In [34], the authors propose a system in which they use SVM to classify the wireless devices in specific rooms within a typical home. They make note that the changing environment, in terms of wireless devices and possible interference make a supervised learning technique a good candidate for building a classifier for location awareness. In [35], the authors propose a new technique, built upon the paradigm of SVM, for determining the location of a wireless device. This approach attempts to address the many challenges that were outlined previously with the fingerprinting method of indoor wireless tracking. They demonstrate the ability to improve upon the existing state of the art and provide an improved selection of wireless points within the indoor environment.

Neural networks are a popular approach to machine-learned classification that attempt to model the behavior of the human brain [26]. The key component of this classification approach is an iterative process in which feedback from past classifications are used in order to provide a better "fit" at classification in the future. This makes the neural networks self-adaptive and data-driven, thus they avoid the often necessary manual intervention that other methods for classification require. The tradeoff with this approach is that often the accuracy of approximation for classification is low until sufficient data is collected to better predict and identify the proper classification.

In [36], the authors propose an approach of using particle swam optimization in conjunction with artificial neural networks (ANN) to improve the overall accuracy in highly dynamic indoor environments. This approach makes use of the fingerprinting technique for location tracking and highlights the inherent online and offline phases as primary candidates for the implementation of their proposed inclusion of ANN. In [37], the authors propose an approach that makes use of ANN to solve the problem of multi-sensor tracking in an indoor environment. This work focuses on identifying the sensors within the environment for use in tracking on a multi-floor exercise in a building. The authors demonstrate that through the use of such an approach that even in the presence of a small set of tracking sensors they are able to correctly identify the relative location, or region, that an object resides within the indoor space.

2.3 Related Work in Sensor Selection

One of the key challenges, specifically when focused on opportunistically discovering sensors, is that of sensor selection (i.e., how to determine whether or not the "best" set of sensors has been selected). "Best" can have many different means depending on the specific context in which the tracking is taking place. This process is of selecting a set of sensors is bounded by time and can be further constrained due to energy and communication limitations of a sensor device. This problem has been formally defined in literature as the subset selection problem. Much literature has been devoted to the subset selection problem and its specific impact on sensor networks.

Prior to proceeding on an overview of related work in the area of subset selection we must first define sensor selection, with respect to tracking – as the area and breadth of research is large, we instead only focus on the related subset of this work. This sensor subset selection is defined as the process of selecting the sensor(s) that will provide the tracking system with the ability to maximize the accuracy of the location estimate. The first step in this process is that of sensor classification, related work in this area can be referenced in the previous section.

In [38], the authors describe various methods for the sensor selection process with respect to wireless sensor networks. They highlight the significant challenges of attempting to maximize the gain, in terms of accuracy, for the system. They discuss the role that sensor selection plays in this overall tracking process and its identification of "desirable" trackers. The primary focus of the methods described in this work is that of sensor coverage rather than strictly of accuracy. In [39], authors describe a technique in which a Kalman-based filter is applied in order to reduce the impact of noise in the sensor selection process. This approach is non-deterministic, and does not make the assumption that the sensor set is known *a priori*. This is a key difference from many existing sensor selection techniques in which the sensing infrastructure must be known. This dynamic approach therefore differs from many of the related works in that the set of sensors can change over time and thus the problem of determining the "best" set increases in complexity. This uncertainty is an important attribute that must be identified as part of this process and taken into account when selecting a subset of sensors.

This thesis is proposing the infusion of two QoS-based selection criteria as part of an enhanced sensor selection process. These criteria provide additional information regarding the expected versus actual performance of a sensor that can be quantified and empirically validated. This infusion can be used in an attempt to find the optimal subset of tracking sensors.

2.4 Related Work in Trust

A key focus of this thesis is on the examination and impact that trust has on the selection of sensors for indoor tracking. Trust has been widely explored in literature; in the field of Computer Science it plays a prominent role in nearly all aspects of technological life. In this thesis, we have focused on the trust of both physical tracking sensors as well as the trust in the software services that are associated with these sensors.

In this section, we discuss related works that focus on the role of trust when it comes to sensors and tracking systems.

In Computer Science the concept of trust has often been associated with the notion of secure computing [40-42]. The notion of trust, by its inherent nature, is subjective and thus is heavily influenced, both positively and negatively, by formed opinions. These opinions serve as the basis for evidences, regarding an entity, which can be collected in order to evaluate the trustworthiness of such an entity. These evidences can be operated on, using various operators and operations, in order to ultimately determine the trustworthiness of an entity.

In [43], author introduces the concept of Theory of Evidence, commonly referred to as Dempster-Shafer (DS) Theory. This theory is built upon the idea that it is possible to combine evidences, collected from different sources, in order to arrive at a level of belief with respect to the trust associated with an entity. In [44], authors propose a trust model that implements the DS Theory for a wireless sensor network (WSN). They demonstrate the efficiency of calculating the trustworthiness of a sensor node within the WSN when compared to other existing trust-based techniques. In [45], the authors make use of the DS Theory in order to classify wireless access points based upon their trustworthiness with the goal of obtaining improved tracking accuracy. In this work, the authors demonstrate the effectiveness of the application of DS Theory by producing a location estimate of up to one meter of accuracy. In [46], the authors propose the use of the DS Theory in order to provide a belief probability as to the relative location in an indoor environment based upon proximity to wireless AP. They demonstrate the effectiveness of such inclusion of the DS Theory over other existing techniques for the classification of location based upon a specified wireless zone in an indoor environment. In [47], the authors examine the role that trust plays when composing systems. Specifically, they focus on the impact that trust plays when composing systems of families of related services. This work conducts a case study involving an indoor tracking system in which they evaluate the various software service components and apply a trust model to them.

Two of the unique features involved in the classification of trust are the temporal and subjective natures of this determination. Below we briefly cover related works in these specific areas as they are the primary focus of the work in this thesis.

In [48], the authors propose a method of evaluating the temporal nature of trust and how such evaluation is an alternative to the traditional evaluation of reputation and direct experiences. The authors identify that through this traditional use, there are both direct and indirect interactions that play a role in the determination of the trustworthiness of an entity. They are able to identify the impact that time plays when classifying the trust of a sensor. In [49], the author describes the impact that trust and its temporal nature plays on the role of selecting devices from the Internet of Things. This selection process is guided by the trust the selector has in the various different devices that are available. Related to the subjective nature, the author notes the importance that time plays in the evaluation of the trustworthiness of a device.

The notion of subjective logic, as first proposed in [4], introduces uncertainty in the evaluation of belief and disbelief about an entity. Applying this concept to the trust domain we can identify the nature that the uncertainty plays on the underlying calculation of trust of an entity. In [4], the tuple of Belief, Disbelief, and Uncertainty {B, D, U} is proposed to measure the trust in a specific entity. In [50], the authors apply this notion of subjective logic to a trust model. The proposed trust model determines the trust of an entity through the collection and evaluation of evidences that can then provide the corresponding values of the {B, D, U} tuple. This associated values then forms the basis for the trust decision.

These related works highlight the importance that trust plays in the selection process. This thesis is proposing to build upon this notion of trust through the construction of a trust model for indoor tracking systems. This thesis provides algorithms to categorize the trustworthiness of a sensor's data. In addition, this thesis provides the details for how this trust can be used in the sensor selection process. Through this infusion of trust as a selection criterion, we demonstrate an improvement in the overall accuracy of the system through enhanced sensor selection.

2.5 Related Work in Reliability

The notion of reliability has often been a closely related subcomponent of trust. Often, if an entity is determined to be reliable, it is in turn determined to be trustworthy and vice versa. The focus of reliability has largely been on the performance of mechanical features of devices and their ability to accomplish a given task. Reliability in the WSN domain has been widely studied in literature [5, 51-53]. Many definitions of reliability have been proposed across these various works, with the primary focus being on the ability for a sensor to provide fault-free behavior for a specified time frame.

In [51], the authors focus on the coverage and connection between sensor nodes in a WSN and the role reliability plays in both the selection and routing of messages throughout the network. They propose a hierarchical clustered WSN to handle the problem of coverage reliability. They specifically analyze the impact that common cause failures have on the perceived reliability of a sensor. In [52], the authors examine the event detection and its application within WSN and the role that reliability of sensor data plays in determining the likelihood of an event trigger. The definition of reliability provided by the authors is very similar to that found in the discussion of trust. The authors provide techniques for calculating the data generation rate and the failure probability of a sensor node.

In [53], the authors discuss the challenge of routing problems and how reliability of the various nodes involved in the routing can impact the overall performance of the WSN. This work is more focused on the network reliability rather than the actual performance of the sensors themselves. In [5], the authors discuss the tradeoff between power consumption and reliability in WSN. This work focuses on not only the behavior of the sensors themselves but also on the routing of their data and the subsequent performance impact that they receive in terms of this tradeoff. They propose a method for evaluating the reliability of a sensor while infusing power consumption as a factor in this determination. They then monitor the impact that providing reliability has on the power consumption, and subsequent the life of the sensor node.

In [54], the authors provide a survey on the notion of reliability within the realm of WSN. Their study focuses on the mechanisms necessary to handle faults or failures within a system. They outline various techniques and methods that have been applied in literature for the identification of both faults and failures in components. As previously mentioned, trust and reliability are often considered one in the same when it comes to evaluating the overall trustworthiness of an entity. In [55], the authors consider trust and reliability for constructing a trustworthy architecture for sensor selection in a WSN. Their motivation behind such a combination is that they define trust to be the essential reliable communication between the various sensor nodes in a WSN. Their proposed architecture focuses on the impact that trust has on sensor nodes and then how the reliability between the nodes plays a factor in their ultimate selection for a given task.

In [56], the authors propose a weighted averaging method for calculating the reliability of a sensor. The reason for the use of this approach as opposed to the popular DS Theory is that when evidences are highly contradictory the result may be inaccurate. Instead, they claim that through the use of their method they are better able to represent such contentious evidences when analyzing a sensor based upon its reliability. They empirically demonstrate that their proposed approach outperforms existing models, in terms of accurate fault diagnoses, by up to 90%.

This thesis proposes to leverage the work discussed in this section but adapt these concepts to fit the specific needs of indoor tracking. One of the key highlights of this thesis is the separation of trust and reliability and their consideration as separate attributes for indoor tracking. This separation is unconventional in the traditional sense of the evaluation of trust and reliability in related domains. We believe that this separation is necessary as a sensor may provide trustworthy data but unreliable service and vice versa. This infusion of trust and reliability as separate QoS-based selection criteria also makes use of the notion of subjective logic, as proposed in [25], in order to empirically evaluate the performance of the sensors for use in the sensor selection process.

2.6 Related Work in Data Fusion

Data fusion is the process of combining data from multiple sources into a single unified view. This process is necessary when multiple data sources are providing data during sampling. The concept of data fusion has been widely studied in literature [6, 57-59]. While this thesis does not directly address data fusion, it is a key component of the overall tracking process as it provides the key evaluation point for the success/failure of sensor selection, its impact on the overall end-to-end runtime of the system, and through its use it produces the location estimate which subsequently leads to the determination of accuracy. Below we focus on related work in the area of multi-sensor data fusion and their application for both WSN and tracking applications.

In [57], the authors describe various techniques and approaches specifically focused on target tracking applications and the fusion of sensor data for the purpose of state estimation. They provide a comprehensive background on the aspect and role that multi-sensor data fusion plays within the scope of tracking. The techniques proposed in this work, while having been improved by newer work in the years since, provided a foundation for identifying the key challenges that are encountered when attempting to fuse two different data sources to provide a unified estimate. In [6], the authors provide an overview on the various process models for multi-sensor data fusion. This survey is focused on the applications, specifically target identification, and how various models can be applied to each specific domain. This work highlights the importance of the identification and classification process.

The author, in [60], provides an overview of multi-sensor data fusion techniques from the domain of computer vision. We highlight this work due to the inclusion of vision-based sensors into the tracking environment as part of indoor tracking. The author describes the challenges associated with noise and identification as part of this process. The author then describes various methods and techniques that have been proposed to address the challenges and provides an unified estimate of an object being viewed. In computer vision, there has been significant literature devoted to the ability to provide data fusion to the many sensors located onboard a robot. An example of this work can be found in [61], in which the authors propose a technique for providing localization and the construction of maps for indoor environments based upon the exploration by a mobile robot. In [62], the authors discuss the improvements made through calibration when applied to existing techniques can yield benefits when attempt to fuse, or smooth the vision-based data.

In [63], the authors propose an approach that attempts to combine data estimates for providing localization in an indoor environment between different sensor modalities. These modalities include Wi-Fi, Inertial sensors, and the use of landmarks to determine the position of the device as it moves about an indoor environment. This work is similar to the approach described in [64] as it uses a Kalman-filter in order to combine the various data sources and their corresponding readings in order to estimate the position within the indoor environment. In [65], the authors describe a technique for combining data from Inertial, Magnetic, Pressure, and Wi-Fi signals. Their *ad-hoc* approach attempts to improve the overall accuracy in such tracking through minimizing the noise and drift of the sensors available. In [66], the authors discuss a multi-sensor collaboration technique between RFID tags and WSN for the purpose of providing real-time sensor notifications for fire detection. This approach discusses how the deployment of the sensors can impact the overall data fusion process due to the complexity of the communication network. They demonstrate a data fusion approach to handle this data, coming from these various sensors, based upon the characteristics of heterogeneity of the sensors.

As indicated above, one of the prevalent approaches found in multi-sensor data fusion is the use of Kalman-filters. A Kalman-filter uses recursion as a means to estimate the state of a process [67]. The goal of this process is to predict and then correct based upon the current estimates and the given state. This filter is designed to handle a linear process, in the case of a non-linear process the Extended Kalman Filter (EKF) has been proposed [67]. The authors in [68] make use of an EKF for the purpose of indoor localization using Wi-Fi. They cite improvement over existing fingerprinting techniques through the inclusion of the EKF by reducing the costly overhead of offline training phase. The EKF is able to predict and correct in an online fashion and thus improve the overall accuracy provided by the system. In [69], the authors propose an extension of the EKF in order to provide improved accuracy based upon time difference of arrival and RSS values from wireless points in an indoor environment. Their approach cites the low accuracy of RSS values for the use in tracking exercises. A similar finding was shown in [70], in which the unpredictability of RSS signals due to noise plays a significant factor on the overall accuracy that can be provided by the system. In order to alleviate this challenge, the authors, in [69], have proposed using the prediction/update process in order to smooth this data to be used in the localization estimate. The authors note a minor improvement over existing approaches and note that additional sensor modalities may be necessary in order to provide more accurate positional estimates.

In this thesis, we build upon the existing work that we have already done with respect to multi-sensor data fusion [64, 70, 71] and augment this work with the inclusion and comparison of other state-of-the-art techniques that have been proposed to meet this need. Specifically, we focus on the EKF and the ability to provide a unified data set to the fusion component of the tracking system. By capitalizing on the distributed nature of the tracking system these fusion components can be decentralized, thus reducing the workload on the system.

2.7 Related Work in Optimization

Optimization can be described as the process of attempting to find the "best" solution from the set of all possible solutions. In this section, we focus on optimization that attempts to measure the various tradeoffs involved in the sensor subset selection problem. Below we highlight related work in this area, and discuss related learning techniques that are necessary as part of the optimization process.

In [72], the authors propose the use of the Gaussian process global optimization for determining the "best" subset of sensors to be used in a monitoring process. This work applies various techniques related to this Gaussian process in order to determine the proper placement and location of sensors within an environment. Using historic temperature data they were able to demonstrate the effectiveness of this technique with respect to selecting the sensors that yielded the highest accuracy within the scope of the selection problem. The authors in [73], propose a technique in which they try to find the optimal placement of sensors within a condominium. In this work, they generate a model that highlights the indoor mobility patterns of humans living within this Smart-Condo and then attempt to predict where the optimal placement of sensors to track the individuals' movement should be placed within the home. Their technique outperforms other comparable techniques, as well as simply random deployment, in terms of accuracy obtained as a result of the sensor placement.

2.7.1 Related Work in Reinforcement Learning Algorithms

In this sub-section, we focus on reinforcement learning and techniques available for optimizing sensor subset selection. The reason for this optimization is due to the complexity of determining the optimal set of sensors during the sensor selection process.

In [74], the authors propose the use of a reinforcement learning based mechanism to perform filtering and load-balancing routing. The motivation behind this work is the high energy consumption and resource waste of sensors in WSN. This learning technique attempts to minimize this waste of both energy and resources by distributing the communication and resource allocation evenly across the various nodes in the network. The authors demonstrate the effectiveness of their approach through improved throughput of the system and prolonged longevity of the sensor nodes themselves.

In [75], the authors use a technique to minimize the energy consumption of sensor nodes in a WSN. Specifically, they examine the routing protocols used between various nodes and a sink. This work differs from the previously described work in [74] by proposing, through reinforcement learning techniques, an improved routing hierarchy. This improvement allows the sensor nodes to better utilize their resources and decides how nodes should conserve their energy during routing. This process is handled through feedback from the network and the sensor nodes themselves. The authors are able to demonstrate the effectiveness of their shortest path Q-routing algorithm to increase network lifetime over other approaches. This work proves significant benefits when dealing with variable sized networks and varying topologies.

In [76], the authors propose a novel reinforcement learning framework for the sensor subset selection problem. Their unique approach decentralizes the problem and makes it scalable with respect to large scale sensor networks – whether that WSN or for indoor tracking. By introducing the concepts of a game algorithm into their work they can provide a penalty or a reward for the process of learning and then provide an optimization function to model the tradeoff between energy consumption and accuracy. Their work demonstrates improvement in both the reduction of energy consumption as well the ability to maximize the accuracy provided. Their improved efficiency, in terms of their algorithmic performance, out performs other existing techniques.

In this thesis, we build upon these techniques (e.g., novel reinforcement learning algorithm, decentralized approach, etc.) through the creation of an optimization function in order to maximize the accuracy while minimizing the cost associated with acquiring a positional estimate for an indoor tracking system. Specifically, we borrow concepts from [76], to construct a framework to handle this optimized sensor subset selection.

This thesis provides the following new features that attempt to address the shortcomings of the related work mentioned in this chapter. Specific contributes include: a new framework that includes improved sensor discovery and classification as a result of the implementation of the opportunistic tracking approach in which the learning techniques and approaches described in this chapter are adapted and used; the introduction and quantification of trust and reliability as separate QoS-based selection criteria for use in the sensor subset selection problem; improved multi-sensor data fusion through the use of a decentralized EKF and pruning techniques as a result of the trust and reliability infusion; and a tradeoff optimization using a reinforcement learning technique, between cost and gain with respect to the selection of sensors, in an effort to find the optimal set of sensors for indoor tracking.

CHAPTER 3. DESIGN AND IMPLEMENTATION

In this chapter we will describe the design and implementation of our proposed enhanced framework for indoor tracking. We begin this chapter with an overview of the prototype ITS that we will be using as an experimental platform. Following this introduction, we will break the rest of the chapter into the following four subsections: Discovery and Classification, Sensor Subset Selection, Multi-Sensor Data Fusion, and Tradeoff Optimization.

In this chapter, we aim to propose solutions to the following challenges for ITS: (1) how to opportunistically discover and classify sensors in a previously unknown sensing environment, (2) how to quantify the QoS-based attributes of trust and reliability, (3) how to infuse these criteria as separate selection parameters within a sensor subset selection algorithm, and (4) how to find an optimal tradeoff between accuracy and runtime with respect to the data fusion process of the ITS.

3.1 Indoor Tracking System Overview

As noted in Chapter 2, there has been a significant amount of literature devoted to the topic of indoor tracking and the development of ITS to meet this need. In this thesis we focus on a prototype ITS, the enhanced Distributed Object Tracking System (eDOTS). The creation of this system was manifested out of a need to provide inexpensive tracking for indoor environments across a wide platform of application domains. In order to describe the techniques and approaches we are proposing to enhance indoor tracking; an introductory treatment on the structure of the eDOTS is necessary. Additional information can be found regarding this system can be found in [64, 70, 71].

The eDOTS is a prototype ITS that encapsulates physical sensors as virtual software services. These software services can then be queried in order to collect positional estimates, from each sensor, for an objects location in an indoor environment. The eDOTS is composed of four distinct layers as shown in Fig. 3.1. The first of these layers, the Sensor layer, consists of the physical sensors that exist within the indoor environment. Any and all sensors that can serve the purpose of tracking are included in

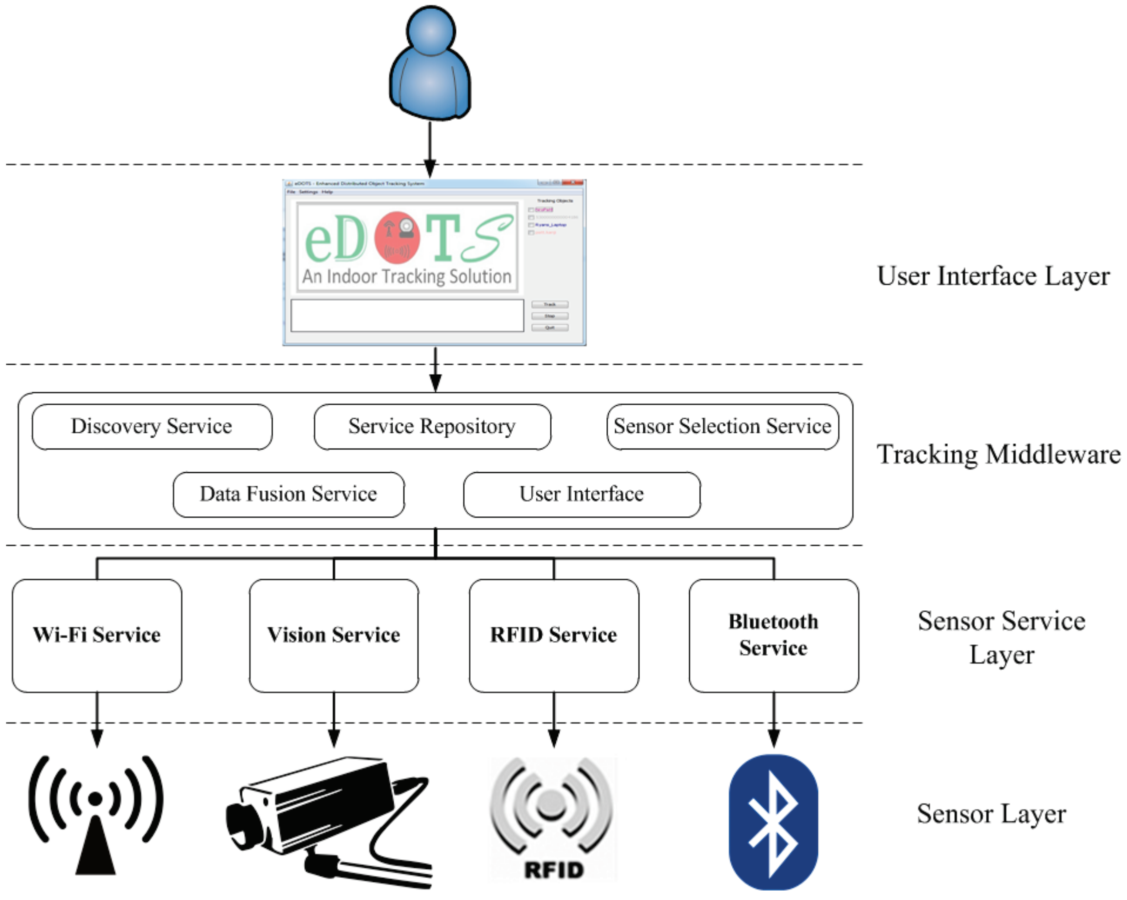


Figure 3.1 eDOTS Overview

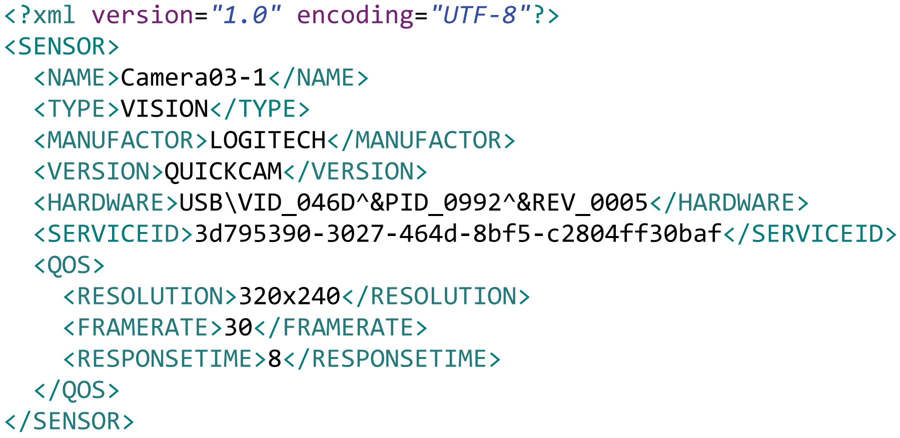
this set of sensors. These sensors are discovered in an opportunistic fashion, more on this discovery process is discussed later in this section. The second layer is that of the Sensor Service layer. In this layer, a virtual software service is created by the tracking system for each corresponding physical sensor discovered in the first layer. This software service, once created, is then registered with a software service repository where it is made ready for consumption. The third layer is that of the Tracking Middleware. This layer is the gateway between the physical sensors, their respective software services, and the user. This layer consists of the functionality, through the composition of various services, which is necessary to calculate an estimated position of an object. This is the core layer within the overall tracking system architecture and contains various services (e.g., discovery, sensor selection, data fusion) as part of this component. The final layer is that of the User Interface. This layer is responsible for visually displaying the results of the positional estimate provided by the Tracking Middleware layer to the user. The goal of this construction was to minimize the needed coupling between the various components, while maximizing the cohesion found in each respective layer.

We will now describe the role of each of these layers in more detail, discuss how each layer and component is related in the system architecture, and provide an overview of the various areas within the eDOTS in which extensions can be added to implement our proposed enhancements to improve the systems indoor tracking ability.

One of the key features of the eDOTS is its novel introduction and use of the concept of opportunistic tracking. This approach, as first defined in [77], revolves around the idea that a tracking system should not rely on a singular technique or sensor modality, but rather should make use of any and all sensors presently available in a given environment. The benefit of this approach is that it eliminates the need for a static sensing infrastructure. Instead, it provides an *ad-hoc* approach to indoor tracking through the dynamic discovery of any sensors present within the indoor environment. Opportunistic tracking does create additional challenges (e.g., dynamic sensor discovery, sensor classification, sensor selection, multi-sensor data fusion) that must be addressed in order to realize its full potential.

To opportunistically discover the sensing infrastructure, the eDOTS uses a multicast message to seek responses from available devices. Here, the assumption is made that there is a common communication channel present that allows for the exchange of messages between the tracking system and the sensor device. A secondary assumption is also made with respect to the ability to access the available devices. Under this assumption, we do not consider security policies placed on the individual devices and do not handle, as part of this work, any security mechanism to prevent malicious activity. We assume that only publically available, and accessible, devices are included in the tracking exercise.

When a physical sensor is discovered by the tracking system, the Tracking Middleware is responsible for creating a software service to serve as a virtual representation of the physical sensor. This Sensor service is responsible for facilitating the interactions between the physical sensor hardware and its software representation in an effort to obtain the necessary information for estimating an objects position. Once created, the Sensor service is registered with the software service repository. During the registration process a software service contract is created. This service contract serves as an agreement between the service provider and the client, in this case the tracking system, as to the expected performance, behavior, and characteristics of the service. An example of a sensor software service contract is shown in Fig. 3.2.



```
<?xml version="1.0" encoding="UTF-8"?>
<SENSOR>
  <NAME>Camera03-1</NAME>
  <TYPE>VISION</TYPE>
  <MANUFACTOR>LOGITECH</MANUFACTOR>
  <VERSION>QUICKCAM</VERSION>
  <HARDWARE>USB\VID_046D^&PID_0992^&REV_0005</HARDWARE>
  <SERVICEID>3d795390-3027-464d-8bf5-c2804ff30baf</SERVICEID>
  <QOS>
    <RESOLUTION>320x240</RESOLUTION>
    <FRAMERATE>30</FRAMERATE>
    <RESPONSETIME>8</RESPONSETIME>
  </QOS>
</SENSOR>
```

Figure 3.2 Sensor Software Service Contract

This software service repository is responsible for maintaining a directory of all of the tracking-related services that are currently available. The process of software service registration and discovery is achieved through the use of the JINI framework [78] . The service repository in this case is provided as a JINI Lookup Service. One of the reasons for use of the JINI framework is its ability to allow for the concept of leasing. A lease is created when a software service registers itself with the JINI Lookup Service. This lease is specified for a period of time, and as part of the JINI framework [79]; this lease is periodically examined and either renewed or terminated. This process of leasing prevents inactive software services from maintaining a presence in the service repository and ensures that “dead” services are properly removed to avoid cluttering the repository and adding unneeded overhead to the sensor selection process.

When a tracking request has been issued from a user, the User Interface forwards this request to the Tracking Middleware. The Tracking Middleware must then query the service repository to see if any current tracking sensors are available and are able to track the requested object. If at least one service is found the Tracking Middleware then issues a tracking request to each identified service, this begins the active tracking process. At the start of this process the Tracking Middleware is responsible for starting another service, the Filter service. The role of this service is to provide a proxy between the sensor software service and the Tracking Middleware components. If tracking is not possible, the user is notified that tracking is currently unavailable due to lack of tracking sensors. In this case, the Tracking Middleware will continue searching, as long as the tracking request is active, for tracking services capable of fulfilling the given request.

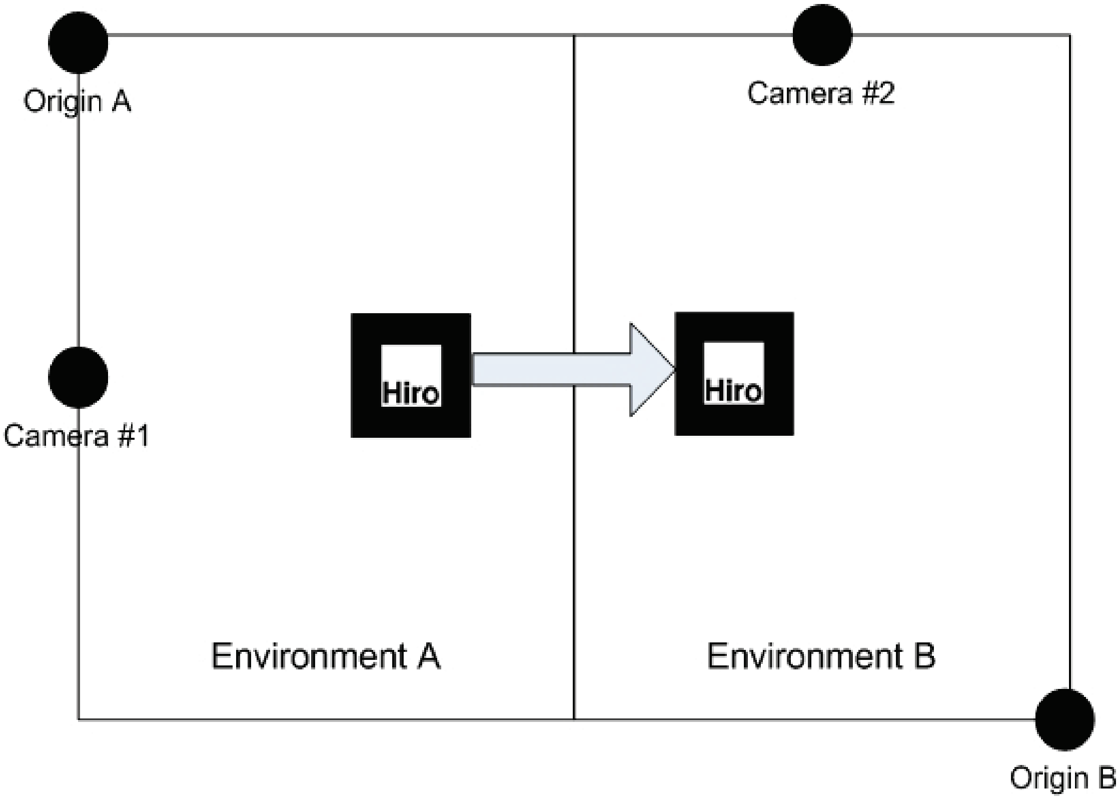


Figure 3.3 Handoff & Transformation

When active tracking begins, the Filter service begins the process of polling each identified Sensor service for its data that provides to the estimated physical location of the requested object. The first step in this process is handling the sensor subset selection. In certain cases, it may not always be feasible or desirable to take data from every reporting sensor service, as the quantity or quality of data may exceed the limitations imposed by the user in terms of runtime. The second step in this process is to then pass along the collected data, if necessary, to the Data Fusion service; otherwise the positional estimate is simply passed to the User Interface for display. The Filter service is responsible for providing any necessary data adjustments, or data smoothing. Data adjustments, in this context, refer to the handoff and transformation between two coordinate systems. This handoff is necessary as there is often no notion of a global coordinate system for indoor environments.

An example, as shown in Fig. 3.3, would be the case of two sensors in two distinct environments in which they both have their own respective point of origin local to their environment. The use of Spatial Relation Graphs (SRG) has been proposed [70] as a method for solving this problem of handoff and transformation between coordinate systems for indoor tracking. These graphs provide the means to allow for the handoff between coordinate systems by adjusting the local reference environment of a sensor. In order to achieve this transformation, the following approach, as shown in Fig. 3.4, is utilized. In this figure the problem is defined as follows: if we know the distance between node B from origin A, and we know the distance from node C to an origin D; through the common position that both nodes B and C are sharing we can then calculate the distance

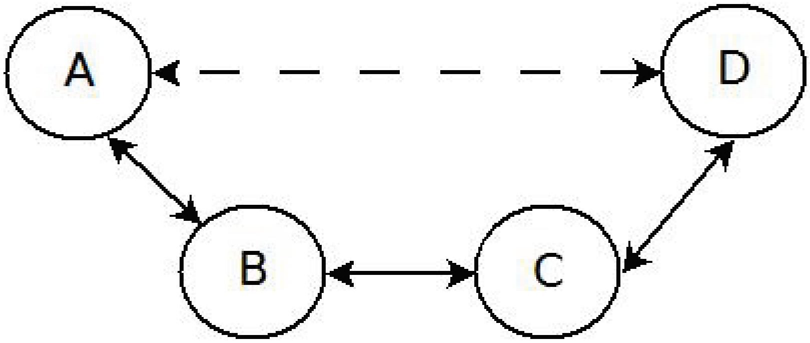


Figure 3.4 Spatial Relation Graph

between the two origin points A and D within the environment. The full details of this approach are described in [70].

In order to initiate this handoff process, the Filter service is responsible for collecting the necessary coordinate information from the sensors that it has selected as part of sensor selection. It is unnecessary to collect such coordinate data from multiple sensors, all with the same point of origin – this is a time consuming process of querying the sensor services when many sensors reside in the same environment using the same coordinate system. Instead, a single representative sensor service is needed for each coordinate system identified by the Filter service. Here we assume the sensors are aware of the environment they are in; this can be achieved through a calibration process (in the case of a static sensor) or through proximity to other known sensors (in the case of a mobile sensor).

The single representative sensor service is determined based upon a ranking hierarchy of the sensor services. This ranking is achieved by the evaluation of the sensor services' Trust level (this notion of Trust is discussed later in this chapter) with the highest ranked sensor service serving as the representative for the given environment.

This “leader” is then used by the Filter service to establish the handoff and transformation between two coordinate systems. In the case that handoff is not possible (e.g., the object is not identified and tracked within two adjacent environments – then a transformation is not necessary, as the respective local coordinate system will suffice).

The process of data fusion is necessary when two or more sensors are actively tracking the same entity at a given moment in time. When the Filter service identifies the presence of two or more sensors actively tracking the same entity, it is responsible for passing this data along to the Fusion service. The Fusion service is a component of the Tracking Middleware layer of the eDOTS. The goal of the Fusion service is to fuse the results from multiple sources into a single positional estimate. As noted in Chapter 2, there exist many different techniques and approaches to provide multi-sensor data fusion. By providing the data fusion mechanism as a software service-based component, it allows for the ability to have available a wide range of different techniques for inclusion in the ITS. In the existing prototype of the eDOTS, the Fusion service uses two different data fusion techniques: simple averaging and a Kalman-based technique. The choice of sensor selection can be specified by the Filter service (based upon the requirements of the user) and provides the flexibility in terms of weighing the cost versus gain tradeoff. More discussion regarding the data fusion for indoor tracking can be found in the Multi-Sensor Data Fusion subsection in this chapter.

Once a positional estimate is received by the Filter service, either as a result of data fusion or the sensor service directly, this information is then passed along to the User Interface level for display purposes. In the current prototype of the eDOTS, this information is displayed in a graphical user interface that marks, via a colored point on

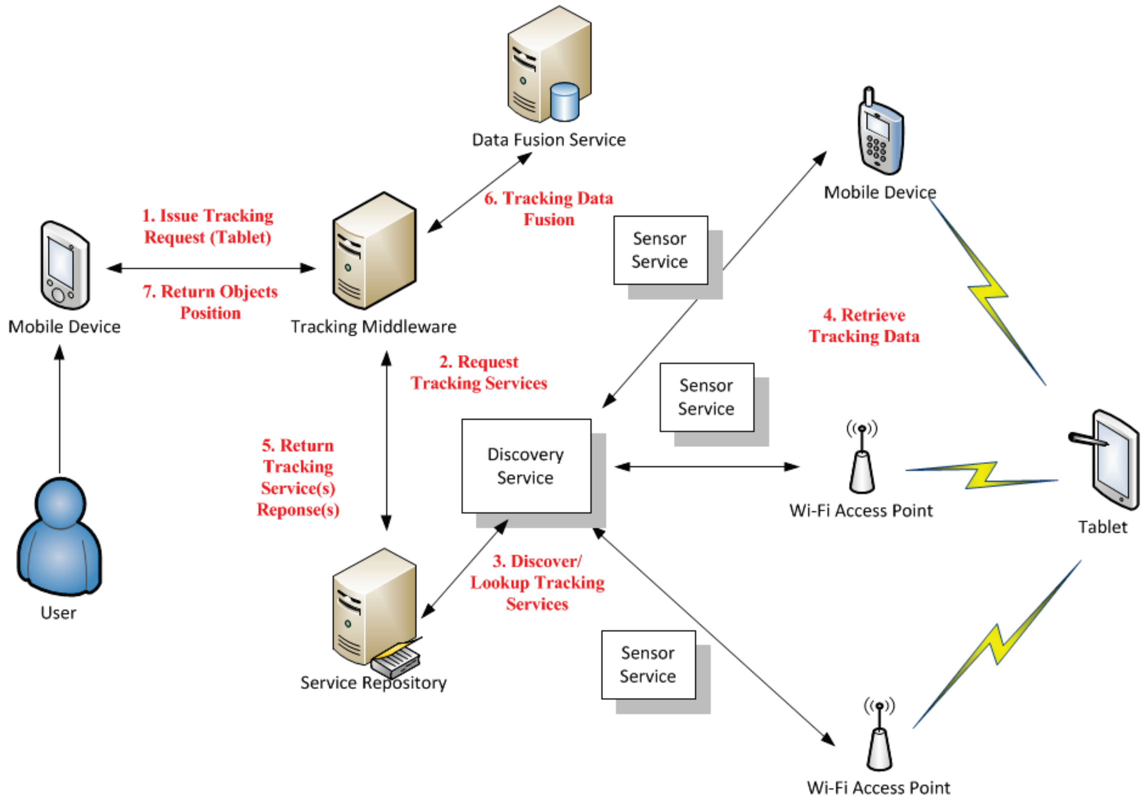


Figure 3.5 Tracking Overview

the screen, the estimated position of the tracked object on an overlay of a map of the indoor environment. These maps are stored in a repository and are accessible by the system for use in tracking. The maps used in this process are obtained via publically available building schematics or could be handcrafted offline by the application domain. This entire end-to-end tracking process, as described in this subsection, is shown in Fig. 3.5.

In the existing prototype, there are three significant challenges that must be addressed in order to provide such opportunistic tracking and ultimately improve the overall accuracy of indoor tracking. These challenges are: sensor discovery and classification, sensor selection, and sensor fusion. These challenges are not unique to the eDOTS, but instead are challenges that are applicable across all ITS. While we use the eDOTS in our case study, it should be made known that the description of the various techniques and approaches that we are proposing here as part of this thesis can easily be generalized such that they apply to any ITS.

In the existing implementation of the eDOTS, sensor discovery is handled by the JINI Lookup Service. This discovery process uses a multicast protocol that allows for the discovery of registered services. These services must be handcrafted by the individual service providers and registered accordingly. This is a manual process that requires knowledge of the respective Tracking Middleware to craft the proper sensor specification. Due to the nature of this discovery the sensors are known *a priori* and thus no classification of the sensors is needed. In the opportunistic tracking approach this process should be dynamic to handle any and all sensors that are discovered. As the sensors may or may not be known *a priori* the tracking system must be able to properly classify the sensors in order to create their respective sensor service for use in tracking.

In the existing implementation of the eDOTS, sensor selection is handled via a simplistic ranking mechanism in which the selection criteria are handcrafted prior to tracking. This mechanism is not agile, in that it does not adapt to any new sensors or changing conditions of the sensors. This mechanism also does not consider the historical behavior of the sensors in its determination for selection. The simple pruning provides only a basic approach to sensor subset selection which is insufficient due to the dynamic nature of opportunistic tracking.

The eDOTS, as previously stated, makes use of two techniques to accomplish data fusion: simple averaging and a Kalman-based technique. Each of these techniques has merit for inclusion in an ITS as each provides a benefit over the other. In simple averaging the benefit is the cost (in terms of time) associated with obtaining the estimate – often at the expense of accuracy; while in the Kalman-based technique the benefit is accuracy – at the expense of time. Both of these techniques are directly impacted by the quality of the sensor selection process. In the existing eDOTS, these techniques are constrained by the quality of subset selection process and have been implemented in such a fashion as to maximize their output. In an opportunistic approach these implementations must be dynamic and be able to properly handle the agile behavior of the sensor selection process. Through improved classification and sensor selection the benefit of these two techniques can be fully realized.

3.2 Sensor Classification

One of the key challenges in an opportunistic tracking scenario is that of sensor discovery and classification. As described in the previous subsection, when a sensor is discovered by the Tracking Middleware a corresponding Sensor service is created as a virtual encapsulation of the physical sensor. In most ITS, the modality of the tracking sensors is known *a priori* and therefore the system can be crafted to properly discover and classify the sensors appropriately with this respective knowledge. In an opportunistic approach this is not the case – and instead the discovery and classification must be dynamic to handle the unknown environment. In order to achieve this dynamic discovery and classification, a new approach is needed. This thesis is proposing the inclusion of a supervised learning technique for the classification of sensors based upon their specific modality during the discovery process. This in turn will allow a previously manual process to be automated and the ability for the ITS to take full advantage of the opportunistic tracking approach.

The first step in the process of classification is building an appropriate sensor knowledge base (sKB). To build this sKB, information regarding the characteristics of common sensors is collected, categorized, and inserted as records for each sensor. This process requires collecting information from publically available specifications along with benchmarks provided in other literature. This information can then serve as the ground truth regarding the expected performance and characteristics of the physical sensors. Using a domain expert, these sensors can then be categorized based upon their specific modality. As we are proposing this sKB to be an online artifact, the initial creation will only need to take place once. Over time the domain expert(s) can monitor the sKB to ensure that all of the information contained in it is indeed correct and up-to-date. We expect this online sKB to evolve over time and essentially serve as a living knowledge base for sensor modality information.

Once the sKB has been created, it can then be used during the sensor discovery process in order to properly classify a newly discovered sensor. Classification is a challenging step that often requires the presence of rules or manual intervention in order to properly arrive with the correct outcome. To address this challenge of sensor classification, we are proposing the inclusion of a supervised learning technique that can make use of the existing sKB to properly classify a sensor upon discovery. As part of this work, we examine three different supervised learning techniques: Naïve Bayes classifier, Rule-based classifier, and a Decision Trees classifier. The reason behind the selection and analysis of these three techniques is their prevalence in related work, as described in

Chapter 2, with respect to sensor classification and use in similar wireless sensor networks. We now describe how each technique can be integrated into the existing sensor discovery process in order to properly classify a given physical sensor.

A Naïve Bayes classifier is a classification approach that is based on the Bayes' Theorem and the maximum posteriori hypothesis [80]. The role of the classifier in this approach is to produce a probability that a given entity, *e*, belongs to a specific class, *c*. In order to accomplish this classification, we make use of the following Bayes Theorem formulas ultimately leading us to our Naïve Bayes classifier.

$$P(H \mid X)$$

Here P is the probability that our maximum posteriori hypothesis (H) holds for the given evidence defined by *X*. By using this approach, we are attempting to find a probability that, given our collected evidences regarding *e*, we can find the specific class, *c* , that it belongs to. For example, if we have the attributes *frame_rate* and *resolution* and we have an *X* such that *X* is a sensor with frame rate of 30 frames per second and a resolution of 320x240 pixels, our goal is to find which class of sensor *X* belongs to. In this case, we can make the hypothesis that this particular sensor, *X*, is a Vision-based sensor – and can take this hypothesis one step further in that not only is sensor, *X*, a Vision-based sensor but more specifically it is a Web Camera. In this example, our $P(H \mid X)$ is the posteriori probability that sensor *X* is a Web Camera based upon the sampled attributes of *frame_rate* and *resolution*.

$$P(H)$$

Here $P(H)$ is defined as the prior probability of our hypothesis *H*. Continuing with our example: given our previously proposed hypothesis, this would represent that any sensor, regardless of the aforementioned attributes (*frame_rate* and *resolution*), would be classified as a Web Camera.

$$P(X \mid H)$$

Here we define the posteriori probability of our sensor, *X*, with respect to our hypothesis, *H*. An example of this would be the probability of sensor *X* having a *frame rate* of 30 frames per second and a *resolution* of 320x240 pixels given that we know that sensor *X* is a Web Camera and that these are the known attributes of a Web Camera as obtained through collected evidences.

$$P(X)$$

This represents the prior probability of *X*. In our example, this would be represented by the probability that a sensor, *X*, from our given set of sensors, S, has a frame rate of 30 frames per second and a resolution of 320x240 pixels. As a result of these definitions we are left with the following formula for computing the probability of the classification of our sensor, *X*:

$$P(H\,|X) = \frac{P(X\,|H)\,P(H)}{P(X)}$$

Using these formulas we can formally define our Naïve Bayesian Classifier as follows:

1. Given a training set, with class labels, in which there are *k*-classes and each respective evidence is represented by an *n*-dimensional vector that consists of *n* attributes, we can create the necessary hypothesis for classification.
2. The classifier component will then attempt to predict, via the highest a posteriori probability, the class to which a given sample *X* belongs to. This

process can be shown through the following formula in which *C* is the class for which *X* belongs:

$$P(C\ |X) = \frac{P(X\ |C)\ P(C)}{P(X)}$$

3. As the denominator portion of this formula is the same for all classes we only need to concern ourselves with maximizing the part of the formula that concerns itself with the various classes within our training set: $P(X\ |C)\ P(C)$.
4. In the case of sensor classification, each class, *C*, consists of many attributes. The nature of this problem allows the naïve assumption to be made as part of this classification through the notion of class conditional independence.

$$\prod_{k=1}^{n} P(x_k|C)$$

Here $x_k$represents the value of a specific attribute for *X*. In the case of sensor classification, $P(x_k|C)$ is the number of samples found in the training set that have the attribute x_k divided by the number of samples of *C* found in the training set.

5. Each class in *C* is evaluated, and a classification is made if and only if the given class maximizes the following: $P(X\ |C)\ P(C)$.

In order to apply this to our given domain, ITS, the following algorithms have been implemented into our indoor tracking framework's classification module:

Training Algorithm:

```
TRAIN(DATASET D, PRIORPROBABILITY P)
CATEGORY C → D
for each c ∈ C
do DOCUMENT D ← C
F ← SELECTFEATURES(D)
COUNT(F)
if P is EMPTY
        ESTIMATEPROB(F)
else
        for each TERM t ∈ D
        do LIKELIHOOD ← $\frac{T_{ct}+1}{\sum T(_{ct}+1)}$
return D, P, LIKELIHOOD
```

Classification Algorithm:

CLASSIFY(DATASET S, LIKELIHOOD L, DOCUMENT D)

```
CATEGORY $C \rightarrow S$
**for each** $c \in S$
**do if** $\mathrm{CONTAINS}(c, D)$
        $\mathrm{LOGPROB}\ [c] += L * \mathrm{SCORE}[c]$
        **else continue**
**return** $\mathrm{MAXSCORE}[\mathrm{SCORE}[c]]$
```

Update Classification Algorithm:

UPDATE(DATASET S, LIKELIHOOD L, DOCUMENT D)

```
CATEGORY $C \rightarrow S$
**for each** $c \in S$
**do if** $\mathrm{CONTAINS}(c, D)$
        $\mathrm{LOGPROB}\ [c] += L * \mathrm{SCORE}[c]$
        **else return** NULL
**return** $\mathrm{MAXSCORE}[\mathrm{SCORE}[c]]$
```

In this implementation, we have used sentiment analysis [81] in the form of a Boolean Multinomial Naïve Bayes model to evaluate the sKB. This model provides the ability to scan a document (a pre-constructed knowledge base or specific software service contract) to find the presence of specific attributes or a keyword that can then be used to match with a training set based upon likelihood probability. The reason for the use of this particular approach is that a specific sensor may fit under multiple different sensor classifications and may only differ based upon runtime performance. Therefore, the probability that a given sensor is in that specific class may change over the course of the tracking exercise. Since a sensor may be unknown, we cannot assume that a sensor will always behave or act in the same manner that it is originally classified as. For instance, an optical sensor may transit from a passive video stream to one that just provides details regarding ambient light due to battery/power consumption. Here, the sensor would transit from one class of sensor (tracker) to a sensor that can no longer provide this information (non-tracker). This provided our motivation for the selection of a Bayesian-based model. We have implemented this approach and technique into the existing discovery component of the Tracking Middleware.

A decision tree is a classification approach in which a tree-structure is constructed based upon a series of conditions [82]. This approach is simpler in nature than that of a Bayesian classifier. These conditions make up the nodes of the tree and the decisions decorate the branches, or links, between the nodes in the tree. The construction of the tree therefore dictates how the various decisions are made in order to reach the classification of a sensor. Based upon this series of decisions, the goal is to reach a leaf node in the tree that contains the proper classification of a given entity. During the traversal of the tree, a probability can be provided as to the belief that given the set of attributes for a given entity, *e*, the current path of nodes in the tree will lead to the proper classification. An example of a simple sensor-based decision tree is shown in Fig. 3.6.

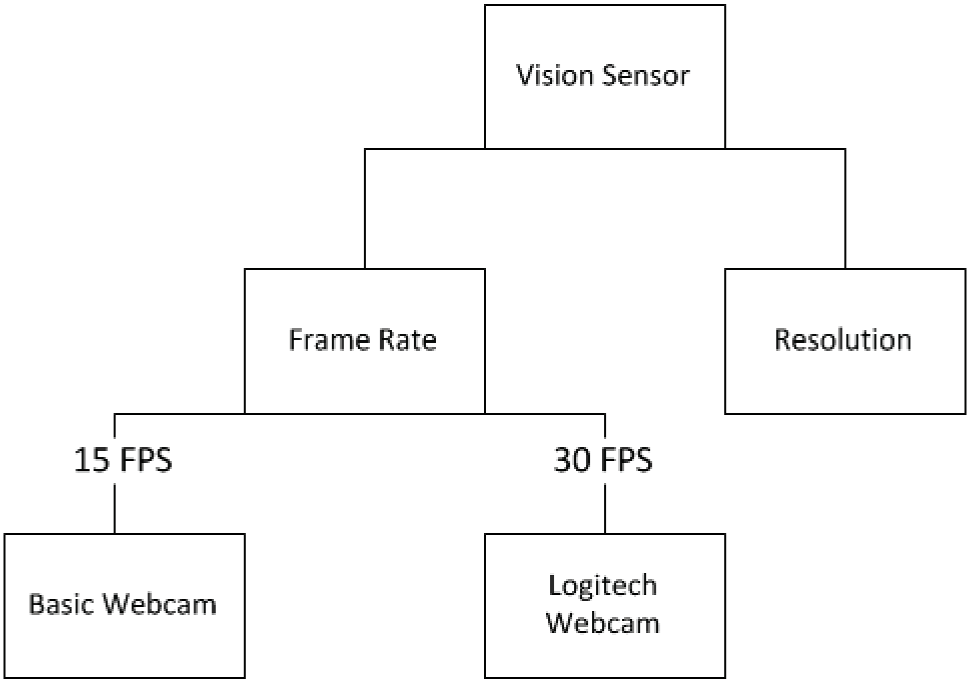


Figure 3.6 Sample Decision Tree

In our classification framework, we begin the decision tree process by constructing such a tree making use of the sKB. Each node in the tree represents a sensor modality class in our sKB. Each condition, or branch decision, in the tree is based upon the different attributes of the existing classes in the training set. In order to build the decision tree, we utilize the ID3 algorithm as proposed in [82]. This algorithm makes use of a top-down approach in analysis and a greedy search technique through the space of possible branches. Each branch in the tree is traversed until a classification can either be met, by the satisfaction of all given conditions, or that no present solution meets the given set of attributes. One of the key features of this approach is the use of entropy and information gain to provide confidence in the classification process. Here entropy is defined as the measure of the homogeneity of a given sample. Information gain is defined as the result of a decrease in entropy of a dataset when it is split on a specific attribute.

Here, we desire to find the maximum information gain possible (e.g., the most homogeneous branch in the decision tree).

A rule-based classifier is an approach similar in nature to that of the decision tree classifier, but instead of the tree structure dictating the traversal that is necessary to reach a conclusion, a set of rules is applied with the goal of determining a proper classification based upon successfully meeting their criteria. This process of constructing rule-based classifiers can be reduced to the problem of converting a decision tree into a series of rules that are iteratively evaluated.

Rules can be constructed in a binary fashion in which an entity either meets (1) or does not meet (0) the respective attribute value. Applying this to our classification framework, this leads to if-then statements within the decision making process. These statements are dynamically created, in an *ad-hoc* fashion, based upon the data and information contained in the sKB. These rules can then be stored by the Tracking Middleware for the duration of the tracking exercise. In the same fashion of the decision tree approach, the process of classification is met through a recursive process in which all of the rules are evaluated with the hopes of finding a matching rule and subsequent classification decision. If a series of rules can be found such that a classification can be made – then a successful classification can be achieved, otherwise, the sensor is given a default classification and the attributes that it contains are entered as new rules as part of the classification process. Once the sensor has been classified the process of analyzing its performance with respect to its expected behavior can begin in the form of trust and reliability analysis.

3.3 Trust

For the purpose of indoor tracking we define trust to be the belief that a sensor will behave in a specified fashion, based upon known information. Here, the notion of trust does not only reflect on the behavior/performance of the physical sensor, but rather on all the factors associated with encapsulating the sensor as a software service and the ability to access it. In this work, we have chosen to focus on the performance of a sensor within a given indoor environment. Trust can therefore be classified as the data trust. We define the data trust to be the trust that the client has in the accuracy of the data provided by the sensor. In this thesis we do not attempt to tackle the issue of security in trust, and thus our notion of trust does not include any analysis of malicious intent nor do we provide any discussion on the aspect of trust and the communication layer.

In this thesis, we use the concept of subjective logic, as proposed in [4], to quantify the trust associated with a particular sensor. This concept makes the use of the idea that an opinion about an entity is subjective and therefore the trust in the entity should be modeled with this subjective nature in mind. This application of subjective logic is modeled in the form of a tuple that contains the values of {B, D, U}: belief, disbelief, and uncertainty. This tuple, when summed, equals the value 1.0. Each of the values in the tuple is a measure, on a scale of 0 to 1.0, trust or lack thereof in a given entity.

In order to calculate this tuple, for a sensor, we use the collection of available evidences. We consider a positive evidence for a sensor reading if it meets or exceeds the sensor’s given specification with regards to its performance. We consider a negative evidence if a reading does not meet sensor’s given specification with regards to its performance. We consider the parameter of uncertainty to be the lack, or insufficient, number of evidences present at a given time *t*. Through these evidences we can then build the trust tuple for a given sensor.

In order to collect these evidences, we have created the concept of a Trust Agent (TA). The role of the TA is to examine the performance of each sensor. When a sensor is discovered by the Tracking Middleware it is subsequently classified and a virtual software service is created and registered with the service repository. Once this has been done the Tracking Middleware will create a new TA for this sensor. This TA will then be responsible for monitoring the performance of the sensor throughout its lifetime. The TA will observe the data, as provided by the sensor service, and provide an analysis of the sensor's performance. Each data reading provided by the sensor service will be treated as an evidence by the associated TA. Using the algorithm described above for determining the nature of the evidence (positive, negative, uncertain), the TA will then calculate the corresponding {B, D, U} for the respective sensor. This approach and placement of the TA eliminates the need to poll the sensor and instead simply examines the data provided to the Filter service. This prevents any unnecessary communication between the sensor service and another component. Thus the overall of the runtime is not negatively impacted by this analysis.

The primary reason for the evaluation of the trust of a sensor is about the accuracy that it provides. This accuracy is subject to the characteristics of the physical sensor and how a positional estimate can be provided by the sensor software service. In order to determine this accuracy we use the following equation:

$$\sum_{i=1}^{N}((L^{*}(S_i) - L(S_i))^2$$

In this equation, the accuracy of the sensor (S_i) is determined based upon the estimated positional estimate (L^*) and the actual position (L). The process of determining the actual position is achieved by using a ground truth sensor for analysis purposes. This ground truth sensor is established by feedback from a sensor injected into the environment in which its position is known. If no sensor can be found that meets this criterion, then the sensor with the highest existing trust belief will be designated as the ground truth sensor.

A secondary point of evaluation of the trust of a sensor is the associated response time. Response time is defined as the amount of time between when a request for tracking data is issued and the time that the response has been received from the sensor. Often this is specified as part of the sensor characteristics and is included as part of the sensor software service contract, as shown in Fig. 3.2, that is registered with the repository and included in the sKB. It is the role of the TA to evaluate the performance of the sensor as this information is logged through the inclusion of timestamps as part of every tracking data. The expected and actual response times are then analyzed by the TA, and a corresponding evidence is formed.

In order to then calculate the trust of the sensor we apply the following equation:

$$S_i \subseteq\ S \text{ and } O_j \subseteq O, t_D\big(S_i, O_j\big) >\ \delta_D\ \rightarrow\ \delta_D\ = Avg(\sum_{i=1}^{N} L^{*}(S_i))$$

In this equation, $S_i \subseteq\ S$ represents each sensor S_i in the set of sensors S involved in tracking. $O_j \subseteq O$ represents each object that the tracking system is currently tracking. The trust is then calculated, via collected evidences, and compared with the trust threshold value denoted by δ_D .

Here we assume that there is a pairing between a sensor and a tracking object in which a positional estimate can be obtained. We define an evidence to be an individual data reading for a given sensor/object pairing when requested by the tracking system. During the tracking process (as indicated earlier) each TA is responsible for collecting these evidences and then evaluating the sensor's performance based upon the data obtained as part of these evidences. In order to determine whether an evidence is trustworthy or not, the TA must analyze the expected versus the actual behavior. Once this has been done, an evidence is then created as to whether or not it is positive or negative. In order to provide a trustworthy assessment, there must be a sufficient amount of evidences available. We define the sufficient amount of evidences to be ten data samples from a given sensor. If sufficient evidences are not available then the uncertainty of the trust classification is present – as evident by the U in the {B, D, U} tuple.

One of the key questions that must be addressed here is the quantity of evidences required in order to provide proper trust analysis. This quantity plays a role in terms of determining the uncertainty of the trust analysis tuple. There are two distinct options available to achieve this: static and dynamic. In the static approach an *a priori* value is assigned before the tracking exercise. This value then serves as the analysis for the uncertainty with a sensor in terms of respective trust. In the dynamic approach, an initial value is provided and then over a period of time the TA learns the proper value that should be associated with the sensor. In both approaches an initial value is provided as the basis for comparison – in the static approach this value does not change over the course of tracking.

A second key question to be addressed as part of the infusion of trust is that when using the notion of subjective logic each of the values in the respective trust tuple have the ability to be evaluated independently. This means that there is the ability for the tracking system to select only sensors that meet or exceed a pre-defined threshold value for each value in the tuple. This threshold then becomes the key point at which a trustworthy decision is made with respect to a sensors performance. In this work this threshold is determined by averaging the trust values of the available sensors. This average is then used to evaluate the trust tuple values of each sensor.

Another aspect that we considered during this trust infusion was how to deal with fault tolerance and bias by the TA's. To address these two concerns we let the Tracking Middleware create *N* number of TA to be associated with a given sensor. In this fashion, if a single TA fails then other TA can provide the necessary support to maintain the analysis of the sensor. As a side effect of this approach we address the concern of bias by the TA. We accomplish this through the fact that each TA is unaware of how many other TA are presently monitoring a given sensor, S_i. We prevent TA from communicating with one another directly as part of the framework, thus no TA is aware of the behavior of the other TA associated with the sensor. To further prevent unwanted bias, a consensus can be formed, by the Tracking Middleware, such that a biased TA information could be discarded. This concern is not completely alleviated unless the TA have no knowledge of what technique is being used for the calculation of the trust value.

Once the TA has collected the necessary evidences, it can then provide the resulting trust tuple to the Filter service which will then use this information as part of the sensor selection process. The role of the TA extends past the monitoring and evaluating of the sensors performance based upon evidences. It is also responsible for updating these trust-related values in the sKB. This is an important step as the inherent temporal nature of trust plays a vital role in the determination of whether an entity is trustworthy or not. When an evidence is collected and the trust tuple calculated the TA is responsible for writing this trust tuple and its associated timestamp into the sKB.

Integration of the TA approach into the existing eDOTS framework was simple due to the agile nature of the software infrastructure. Using the concepts found in good software design and software design patterns a new component in the form of an Agent factory was created. This concept of a factory allows for the creation of agents for both trust and reliability respectively. This new software component can directly interface with the existing Sensor Selection component as part of the system. By using this approach our implementation can easily be adapted to any tracking system by providing a hook into the respective sensor selection mechanism.

To accurately assess each of these trust tuple readings a weight factor must be introduced to distinguish between those ratings that are less relevant to the current tracking exercise. In order to achieve this, we include a weighted trust approach in which both an exponentially weighted moving average algorithm [83] and a penalty/reward system [76] is used to properly assess the trust value. An exponentially weighted moving average is an algorithm that applies weight factors as means for evaluating a time series of data.

We use the following definitions to describe this algorithm:

$$S_1 = Y_1 \; ; \; t > 1$$

Here, we initialize the initial state, *S*, to the first value in our time series of sensor data, represented by *Y*. We also make the assumption that that value of *t* will be greater than 1 in any subsequent evaluations.

$$S_t = \omega Y_t + (1 - \omega) S_{t-1}$$

In this equation, ω represents the weight associated the given data observation. This value is between 0 and 1. The higher the weighted value, the greater discounting of older data observations is achieved. This provides a weight associated with each trust value that is stored in the sKB. Over the course of interaction with the sensor, this value will be updated by TA and evaluated appropriately by the Tracking Middleware as part of the sensor subset selection process.

3.4 Reliability

A key aspect of our sensor selection framework is the separation of trust and reliability as separate selection criteria. We define reliability to be the belief that a sensor will behave, fault-free, over a given period of time. Here, the notion of reliability does not only reflect on the physical sensor, but rather on all the factors associated with encapsulating the sensor as a software service and the ability to access it. This definition allows for a sensor to provide trustworthy responses but be unreliable and vice versa. For the evaluation of reliability, as a QoS-based criterion, we create the concept of reliability agent (RA). This RA will behave in much the same manner that the TA does for trust analysis. Therefore, a majority of the discussion found in the Trust subsection of this chapter applies here to the concept of an RA. We will highlight the differences between the two.

As noted in Chapter 2, there have been many approaches proposed for quantifying the reliability of sensors. The common theme found across these various approaches in literature is that of fault-free operation. A fault can be defined as either a mechanical fault

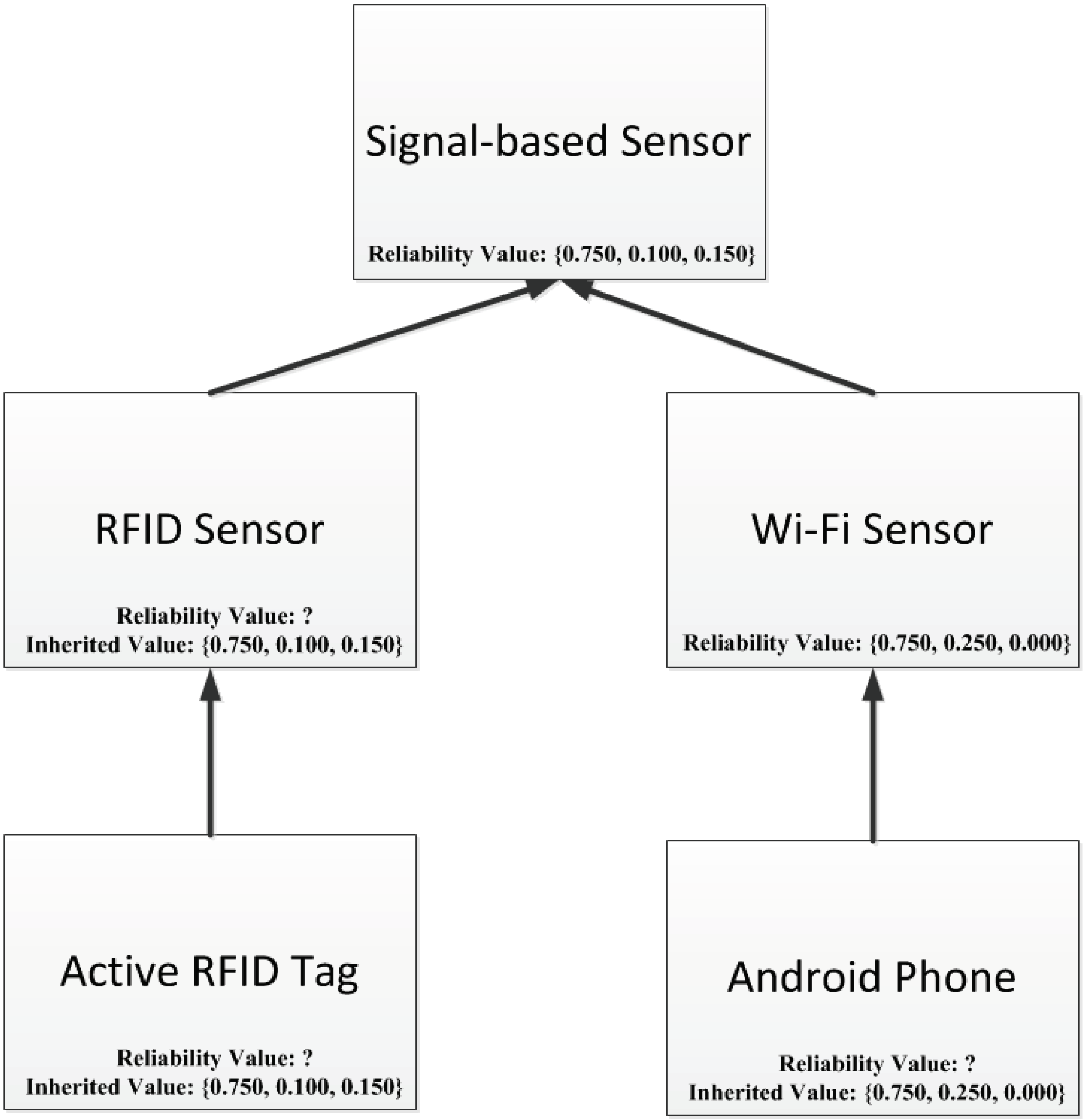

Signal-based Sensor
Reliability Value: {0.750, 0.100, 0.150}
RFID Sensor
Reliability Value: ?
Inherited Value: {0.750, 0.100, 0.150}
Wi-Fi Sensor
Reliability Value: {0.750, 0.250, 0.000}
Active RFID Tag
Reliability Value: ?
Inherited Value: {0.750, 0.100, 0.150}
Android Phone
Reliability Value: ?
Inherited Value: {0.750, 0.250, 0.000}

Figure 3.7 Reliability Hierarchy

of the sensor itself or in the software component associated with the sensor. These faults may be temporary or may cause an ultimate failure in the sensor rendering it useless for the tracking process. Identifications of these faults are key to the overall selection of a sensor.

One of the features that we use as part of reliability determination is the notion of transitive reliability between abstract sensor class and concrete sensor. In Fig. 3.7 this is shown through the use of a reliability hierarchy. The concept of an abstract sensor we define as the high level of classification assigned to a sensor (e.g., Signal-based Sensor) as we traverse this hierarchy we next encounter child nodes of the parent that contain further sensor classifications (e.g., Wi-Fi, RFID, Bluetooth, etc.). Eventually, our traversal leads us to a leaf node in which a concrete sensor is contained (e.g., Samsung Galaxy S6 IMEI: 357754075488264). This hierarchy allows for the sensor to inherit the reliability from the parent class that it is a member of. One of the benefits of this approach is that when a new sensor is added it can inherit and start with a baseline reliability value associated with its expected behavior.

Each value of the reliability tuple is determined based upon collected evidences, or interactions, with the sensor and its produced behavior. The primary goal of this approach is to model the uncertainty that comes with ascertaining the level of reliability of an entity. A lack of sufficient evidences makes it impossible to make a judgement as to the reliability of an entity. By modeling this uncertainty, it demonstrates the full spectrum of confidence one has with evaluation of a given entity.

For implementing the RA into the existing eDOTS framework, we use the Factory pattern. This pattern allows for an Agent factory to create RA necessary to evaluate each given sensor service. When a new sensor is identified and a respective sensor service is created the Tracking Middleware is responsible for creating a corresponding RA. The RA behaves in the same fashion as the TA does, as discussed in the Trust subsection. We will omit a discussion here as the general behavior is the same.

In order to quantify the reliability of a sensor we use the following equation.

$$S_i \subseteq S, r(S_i) > \delta_R \rightarrow \delta_R = Avg(\sum_{i=1}^{N} r(S_i))$$

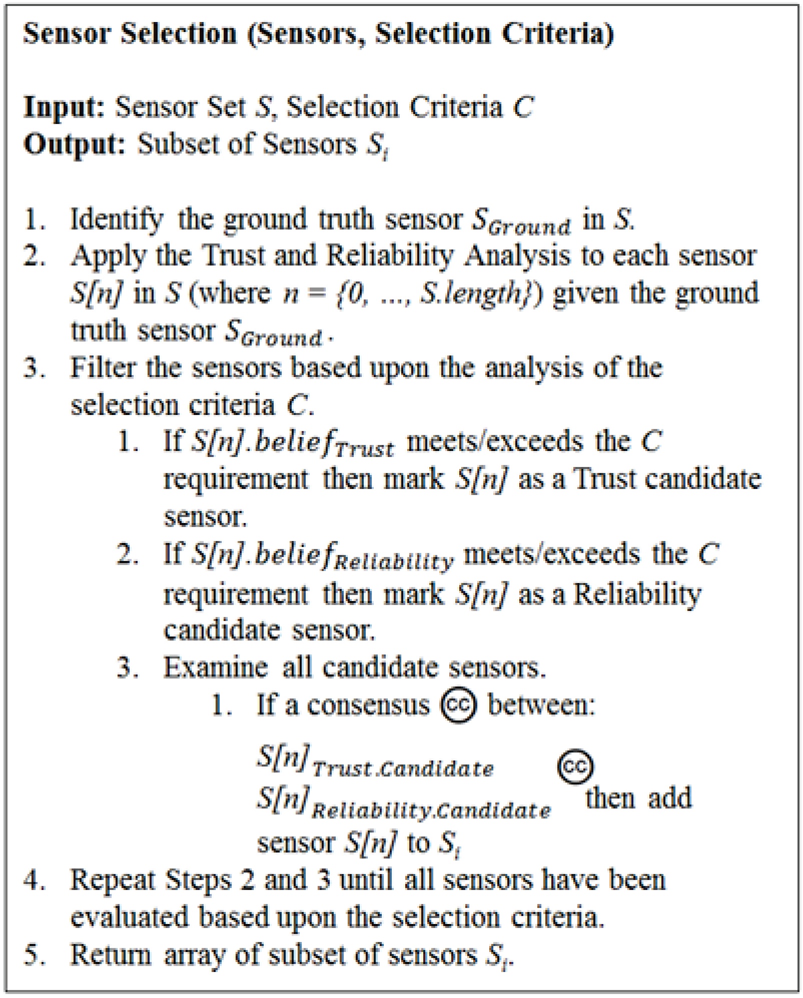
Here, the reliability of a sensor, $r(S_i)$, is determined based upon whether or not it meets or exceeds the reliability threshold, δ_R , as either specified by the client or calculated as a result of computing the averaging reliability values for all sensors involved in the tracking process. This calculation is then used to determine whether or not, based upon collected evidences, a sensor is determined to be reliable or not.

3.5 Sensor Subset Selection

As stated earlier, one of the key challenges with indoor tracking is that of sensor selection. With regards to the approach of opportunistic tracking this process becomes an even greater challenge as the sensors are not known *a priori* and therefore an offline approach cannot be used. The problem of sensor selection is the process of finding the "best" subset of sensors to be used in the tracking process. While it has been shown to be an NP-hard problem [84], we use, as heuristics, trust and reliability as separate quality-of-service (QoS) criteria to aid in this process.

As previously mentioned, in Chapter 2, there have been many algorithms and techniques proposed to address the challenge of sensor selection. We define a simple selection technique in this subsection and then discuss a more advanced technique in last subsection, Tradeoff Optimization. This simple technique uses sensor pruning in order to accomplish the goal of finding a subset of sensors for the tracking purpose. This pruning is guided by the QoS-based selection criteria, as proposed, of trust and reliability.

Sensor selection is an important step in the determination of a positional estimate in an ITS. For indoor tracking the key feature that is to be considered as part of this selection process is that of tracking accuracy. Here the overall tracking accuracy is a direct result of the quality of the sensor selection process. The sensor selection process



```
Sensor Selection (Sensors, Selection Criteria)

Input: Sensor Set $S$, Selection Criteria $C$
Output: Subset of Sensors $S_i$

1. Identify the ground truth sensor $S_{Ground}$ in $S$.
2. Apply the Trust and Reliability Analysis to each sensor
   $S[n]$ in $S$ (where $n = \{0, ..., S.length\}$) given the ground
   truth sensor $S_{Ground}$.
3. Filter the sensors based upon the analysis of the
   selection criteria $C$.
     1. If $S[n].belief_{Trust}$ meets/exceeds the $C$
        requirement then mark $S[n]$ as a Trust candidate
        sensor.
     2. If $S[n].belief_{Reliability}$ meets/exceeds the $C$
        requirement then mark $S[n]$ as a Reliability
        candidate sensor.
     3. Examine all candidate sensors.
          1. If a consensus 🅭 between:
             $S[n]_{Trust.Candidate}$ 🅭
             $S[n]_{Reliability.Candidate}$ then add
             sensor $S[n]$ to $S_i$
4. Repeat Steps 2 and 3 until all sensors have been
   evaluated based upon the selection criteria.
5. Return array of subset of sensors $S_i$.
```

Figure 3.8 Sensor Selection Algorithm

can be defined as given a set of sensors, S, the goal is to produce a subset of these sensors, S_i, that will yield the most gain for the system. In order to provide an evaluation of this gain to the system it necessary to have a ground truth as to which to compare to.

The first step in the Sensor Selection algorithm, shown in Fig. 3.8, is to identify the ground truth sensor in the given environment. Once this has been established, we begin the process of selection by pruning sensors based upon their QoS-based criterion of trust and reliability. The ground truth sensor serves as the basis for the trust and reliability analysis in that it provides the threshold value as to which both criteria are evaluated by. This threshold value serves as the filtering decision for the pruning of sensors.

In order to filter the sensors, each criterion is evaluated independently. The first of

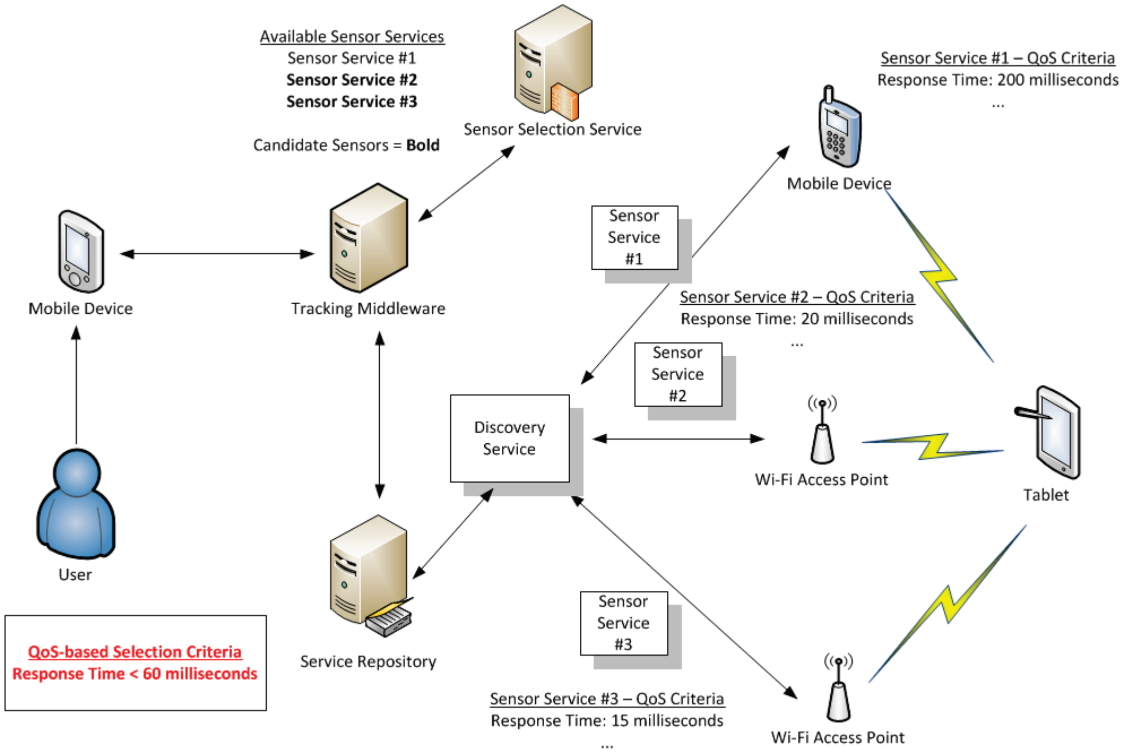


Figure 3.9 Sensor Selection Overview

these criteria to be evaluated is trust. To evaluate and filter based upon the trust of a sensor each of the values of the {B, D, U} tuple is evaluated and compared with the defined threshold value. If the value of the sensor meets or exceeds the necessary threshold value for the belief portion of the tuple then we continue the evaluation based on the other two values. If the disbelief portion of the tuple is below or meets the given threshold value, then we continue on to the final value of uncertainty. If the value of the sensors uncertainty is below or meets the given threshold value as provided by the ground truth analysis, we mark the sensor as a candidate sensor for the purpose of sensor selection.

In order to filter the sensors based upon the reliability criteria the same process is followed as used by the trust-based selection. The outcome of this process is a set of candidate sensors that have been identified to have met the requirements based upon the reliability threshold. At this point in the process we are left with two candidate sets (one for trust and one for reliability). In order to resolve this process, we apply the subjective logic operator of consensus. This operator attempts to resolve any conflicts by ensuring that a sensor in question appears in both lists. If a consensus is met then the sensor is added to the sensor set, S_i. This process is repeated until all sensors in the sensor set, S, have been evaluated. The outcome of this process is a subset of sensors that is now ready for the data fusion process. An overview of this process is visually represented in Fig. 3.9.

Our approach is different than those approaches listed in Chapter 2 in that infuse of the QoS-based attributes of trust and reliability as separate selection criterion. We hypothesize that this infusion will lead to improved sensor selection, specifically when it comes to the accuracy of the ITS. This hypothesis is empirically validated as described in Chapter 4.

3.6 Multi-sensor Data Fusion

In this section we focus on the role that sensor selection and the infusion of trust and reliability have on the sensor data fusion process and how modifications to the existing approach yield potential benefits as a result of this improved sensor selection. The role of the extended Kalman-based filter is to provide a predict and correct mechanism that attempts to smooth potentially noisy data. Once a subset of sensors has been identified, it is now the responsibility of the Filter service to passed to the Data Fusion service component of the eDOTS. This process is only necessary if there are two or more sensors that have been identified during the sensor selection process.

One of the key challenges associated with any multi-modal environment is determining the proper process to combine, or fuse, the data results. This process is typically computationally expensive and serves as a bottleneck for the system. Furthermore, this costly data fusion may not always be possible or feasible on mobile devices without further pruning or assistance. In order to address these challenges, we use a combination of techniques in order to provide accurate multi-sensor data fusion. The eDOTS currently provides the ability for averaging or an extended Kalman Filter technique (as discussed in subsection 1) as its primary methods of multi-sensor data fusion. Prior to the data fusion process, the data must be streamlined and prepared for the data fusion method. This pruning process currently relies heavily on the ranking system and other environmental collected heuristics. We first fuse data of similar classes of sensors based upon their ranks. However, if the ranks are cut across various sensor modalities, these modalities and their associated data must be unified prior to being passed to the data fuser.

In previous work [70] we have discussed, the typical data fusion process and have provided details and analysis regarding the techniques used. These techniques have been adapted to fully realize the distributed approach to the Kalman-based filter in order to meet the needs of mobile sensors and the incorporation of the ranking algorithm. In this adaptation we allow for the fuser, as part of the data fusion process, to be distributed locally or remotely as part of the Fusion service. In order to accomplish this and satisfy the strict requirements of multi-sensor data fusion, we include the sensor subset selection algorithm, described in the previous subsection. This process provides the necessary data, in the proper format, for the Fusion service to use.

When incorporating mobile devices into the tracking infrastructure, careful attention is needed when performing data fusion, due to its computational complexity. Through prior analysis of the techniques and methods used for performing the data fusion it was demonstrated that it would not be feasible to perform this data fusion on mobile sensing devices [20] in the same fashion as that of the static system. This was due to the unpredictable nature of the mobile devices including their connectivity and availability on the local network. By using a distributed approach in the data fusion is spread across various software components is the basis of the distributed Kalman-based filter technique as described in [70].

We will now describe how we apply an extended Kalman filter to the problem of multi-sensor data fusion.

The model for the extended Kalman filter [85] is as follows:

$$x_k = Ax_{k-1} + Bu_k$$

- Where x_k is the current state of the system, x_{k-1} is the previous state of the system, and A and B are scaling constants.

$$z_k = Cx_k + v_k$$

- Where z_k is the current observation of the system, $v_k$is the current noise associated with the observation, and C is a scaling constant.

The predict portion of the extended Kalman filter using the following formulas:

$$\hat{x} = A\hat{x}_{k-1} + Bu_k$$

- Where $\hat{x}$ is the estimate of the predicted current state of the system and $\hat{x}_{k-1}$ is the estimate of the previous state.

$$P_k = AP_{k-1}A^T$$

- Where P_k is the prediction error and P_{k-1} is the predicted error of the previous calculation.

The update portion of the extended Kalman filter is achieved through the use of the following formulas:

$$G_k = P_kC^T(CP_kC^T + R)^{-1}$$

- Where G_k is the current gain and R is the average noise of the measurement.

$$\hat{x} = \hat{x}_k + G_k(z_k - C\hat{x}_k)$$

$$P_k = (1 - G_kC)P_k$$

If we apply this extended Kalman filter to our tracking exercise we find that the following:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Here, the initial assumption is that each coordinate value is accurate and that there is no noise. This matrix indicates the impact that each of the coordinates has on the other – in this case, the measurement used to obtain the Y coordinate has no impact on the X or Z calculation.

This approach can also be expanded to the fusion between the trust and reliability values with respective sensors and by the Tracking Middleware. Another possibility is the infusion of the existing X, Y, Z coordinate values with the trust and reliability values as produced by their respective analysis.

3.7 Tradeoff Optimization

In the discussion of the sensor selection process, we made references to the tradeoff between the gain and the cost of such selection. The gain, with regards to tracking, is the accuracy provided by the system. The cost, with respect to tracking, is the runtime overhead associated with obtaining such accuracy. As a result of this process, there exists a tradeoff in which the tracking system desires to maximize the accuracy obtained while minimizing the cost associated with obtaining this accuracy. Many attempts have been made at modeling such a tradeoff between gain and cost, as discussed in Chapter 2. Here we describe the use of an optimization function, one that models that tradeoff with associated weights that allow for the optimization to take place.

In order to quantify the accuracy of the positional estimate provided by the system we, again use a ground truth sensor that can be selected as part of this supervised process. In order to quantify the cost associated with the calculation of this tradeoff the end-to-end runtime of the system is collected. During an offline process of calibration, the end-to-end runtime of the system can be calculated and then used for comparison.

The proposed optimization function models the utility versus cost via a weighted sum approach:

$$F_A = W_{Accuracy} * F(Accuracy) + W_{Cost} * F(Cost)$$

$$\text{Subject To: } F(Accuracy) = \sum_{i=1}^{N} \Big((L(S_i) - L^*(S_i))^2 * \big(t_{D_i}\big) \Big)$$

$$F(Cost) = \sum_{i=1}^{N} (1 - E(S_i))^2$$

$$\text{Where: } W_{Accuracy} + W_{Cost} = 1$$

Here $W_{Accuracy}$ is defined as the weighted value associated with the accuracy of sensor S_i position estimate at a given time T. W_{Cost} is defined as the weighted value associated with the cost, in terms of system runtime, in order to provide a tracking estimate from sensor S_i. F_A to be the objective function for the tradeoff between accuracy and cost when performing tracking action A. L to be the location of the object. L^* to be the estimated location of the object. t_{D_i} to be the trust related to the location estimate.
If the weighted value associated with accuracy is greater than that of the weighted value associated with the cost – the accuracy generated as result of the sensor selection process will dominate. If the weighted value associated with cost is greater – the runtime of the system will dominate the tracking process. The goal of this function is to provide a way

to find the optimal set of sensors in order to optimize the function. In order to realize this goal we make use of reinforcement learning techniques in order to learn the optimal set of sensors to satisfy this function.

In [76], the authors propose a novel reinforcement learning framework for the sensor subset selection problem. Here we adapt that approach to fit our optimization framework to model the tradeoff between accuracy and the runtime of the system. In the work described in [76], the authors focus on the tradeoff between accuracy and energy consumption. Their approach makes use of a decentralized pursuit learning game algorithm. In order to focus on the impact of time on the tradeoff optimization, we modify the existing proposed approach [76]. Our proposed objective function describes two conflicting goals: (1) to produce a highly accurate positional estimate and (2) to minimize the runtime cost in terms of time. We now focus on the differences between the existing approach described in [76] and our modified approach for the problem of sensor subset selection.

Each tracking sensor is represented as a learning automaton. The set of active trackers that we have identified serve as the basis for the formation of a team of automata that attempt to converge towards the optimization of our tradeoff function as previously described. Here the action of the tracking sensor is directly related to its QoS-based performance. If the tracking sensor is evaluated by using these criteria, performs the selection analysis and sends its data on to the Tracking Middleware, at the cost of the runtime performance hit that is required for such analysis. If the tracking sensors action is to not perform this analysis, its data is not forwarded on to the Tracking Middleware at the possible expense of the accuracy of the estimate, however, a performance hit is not taken as the added runtime with the evaluation process is not performed.

As part of this process, there is the impact of the related weighted values associated with both accuracy and cost. Here the user of the system can specify which of these criteria are more important in their given application domain. For instance, in certain cases it may be more important to have a real-time response generated by the ITS at the expense of the accuracy provided; while in other cases it may be more important to have as highly accurate positional estimate at the expense of the time required to produce such an estimate. The cost is measured as the end-to-end runtime of the selection process. We can then formulate a penalty probability mechanism denoted by the specific action selected with respect to the cost. Thus, $(1 - F(Cost))$ gives us the reward associated with the specific action selected.

We will now empirically evaluate these proposed enhancements through the eDOTS prototype.

CHAPTER 4. EXPERIMENTATION AND ANALYSIS

In this chapter, we provide the details about how the various components, described in Chapter 3, have been implemented into the eDOTS and experimented on. This experimentation was carried out to empirically validate the proposed framework and to demonstrate its effect on our hypothesis. We begin with an overview of the tracking environment and our experimental setup before proceeding to discuss each area of enhancement, as described in Chapter 3, in more detail and provide the empirical results from the experimentation.

To empirically validate our proposed enhancements to indoor tracking, we made use of the eDOTS described in Chapter 2. The experiments discussed in this chapter were conducted in a variety of indoor environments, with the primary environment being the research lab located in the Science building (SL) on the campus of Indiana University-Purdue University Indianapolis (IUPUI). This laboratory consists of a wide range of equipment that has the potential to be tracked as it is moved about the indoor environment.

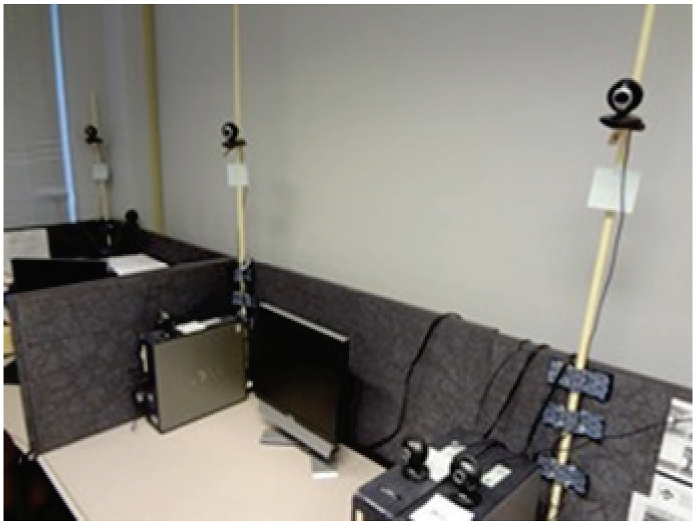


Figure 4.1 Laboratory Setup

This specific environment makes use of two primary classes of sensor: Wi-Fi-based and Vision-based. These sensors provide a static tracking infrastructure that allows for precise measurements to be taken based on the accuracy of the system. We have made the decision to focus on the two primary classes of Wi-Fi and Vision due in part to the following three criteria: a) their popularity as indoor tracking approaches, b) their accuracy that they provide, and c) their ability to be greatly impacted by both sensor failure and other environment variables that could affect their tracking performance. Other sensor modalities were present within the environment such as: inertial sensors, Bluetooth sensors, and RFID (both active and passive). These sensors, along with mobile Wi-Fi-based and Vision-based sensors provide a dynamic flavor to the tracking exercise.

The typical environment in which we conducted our experimentation in consisted of at least twenty unique sensors. In these experiments, we make use of the opportunistic tracking approach. With this approach, tracking sensors may enter or leave the environment and therefore we do not rely on any static infrastructure but rather simply take advantage of whatever sensors are currently present. This use of opportunistic tracking introduced a unique challenge to our experimentation, in that each tracking exercise was unique and therefore made it impossible to reproduce organically. Validation could be achieved but in this case it was handcrafted in order to empirically validate our results.

Our primary tracking environment consisted of twenty static web cameras, shown in Fig. 4.1, as part of the physical existing infrastructure, which could be used for Vision-based tracking. Due to this existing infrastructure and its static nature we could always ensure that at all times there were always at least twenty sensors present, although in most experiments there were considerably more sensors present. Based upon a preliminary study of typical indoor environments, we determined that twenty sensors would provide an adequate representation. The primary room in which we tracked objects is 54.81 square meters in size. This room is on the first floor of a multi-story building, and thus it was often possible for sensors from the adjoining floors to be discovered as their signal, or sensing capabilities, propagated into our tracking environment. This potentially unknown sensing infrastructure added the necessary components to evaluate the performance of our proposed enhancements for indoor tracking.

We divided the experiments up into each specific area that we have proposed enhancements to in this thesis: sensor classification, trust, reliability, sensor selection, and data fusion. The experiments in each area were conducted to test the effectiveness of our approach; we conclude this chapter with a description of an end-to-end test of the tracking system to demonstrate the overall effectiveness of our proposed approach and the cost incurred through such inclusion through analysis of our tradeoff optimization function.

4.1 Sensor Classification

For the purpose of evaluating sensor classification, as indicated earlier, we examined three prevalent techniques: a Decision Tree Classifier, a Rule-Based Classifier, and a Naïve Bayes Classifier. The goal of this exercise was to evaluate the performance of these techniques with respect to the usage and the creation of the tracking sensor knowledge base (sKB) and the accuracy of such classification.

The initial step in this exercise was the creation of the sKB. This involved an in-depth search for sensor specifications published by manufactures that were publically available. Once a sensor specification was found, it was then translated into an XML format and stored as part of the sKB. An example of a sensor specification in XML format can be seen in Fig. 4.2.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SENSOR>
  <NAME>Camera03</NAME>
  <TYPE>VISION</TYPE>
  <MANUFACTOR>LOGITECH</MANUFACTOR>
  <VERSION>QUICKCAM</VERSION>
  <HARDWARE>USB\VID_046D^&PID_0992^&REV_0005</HARDWARE>
  <SERVICEID>48e2f0f1-2f0f-4ca1-9c20-9daed3e251ba</SERVICEID>
  <QOS>
    <RESOLUTION>320x235</RESOLUTION>
    <FRAMERATE>30</FRAMERATE>
    <RESPONSETIME>12</RESPONSETIME>
  </QOS>
</SENSOR>
```

Figure 4.2 Sensor Specification

The above specification provides the characteristics of how the given sensor modality should behave during its use. This serves the basis for sensor evaluation (both in terms of classification and in the case of its performance). These specifications serve as living documents and are both modified and removed as necessary during the course of interaction with a sensor.

To conduct the evaluation of the three classification techniques, we provided the system with a total of ten sensors – of which we knew the proper classification for each sensor. This will allow us to compare the actual versus the expected classification by the three different approaches in a typical indoor environment. Five of the sensors were selected to be provided with complete specification data, while five others contained limited specification data. This limited specification data was created manually for validation purposed by removing existing knowledge from the sKB. The motivation behind this decision was to empirically validate the accuracy that the classification technique could obtain in both the presence of a complete and incomplete training set. The results of this experiment are shown in Table 4.1.

Table 4.1 Sensor Classification

Classification Technique	Accuracy (Known)	Accuracy (Unknown)
Decision Tree Classifier	100%	90%
Rule-Based Classifier	100%	90%
Naïve Bayes Classifier	90%	80%

As can be seen in Table 4.1, we found that each method was able to provide a high accuracy (all sensors over 80%) with respect to the classification of the sensors – for many application domains (e.g., asset tracking) this accuracy may suffice. As shown, the accuracy with respect to the sensors in which information was not available was higher for both the Decision Tree and Rule-Based approaches, while the Naïve Bayes Classifier approach appeared to have a more difficult time properly identifying the sensor based upon the lack of information. We believe this is the case due to the training data set available and the ability of the Rule-Based and Decision Tree approaches to simply add a new sensor modality based upon a sensor template. We had to conduct this experiment multiple times as we struggled initially with the problem of overfitting. The cost associated with the implementation of each of these techniques into the prototype system, in terms of time, was an added mean of 2.5 milliseconds.

The second exercise we conducted, with respect to classification, was to validate that the sKB was updated appropriately based upon the information collected. In this experiment, we introduced a new sensor into the tracking system that had no *a priori* specification within the sKB. This sensor should, based upon the implementation, be given a generic classification. Over the course of interaction with the sensor, the classification should evolve based upon the evidences and additional information and thus allow a proper classification to take place.

For this exercise, we began with the following sensor specification from the sensor service contract, as shown in Fig. 4.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<SENSOR>
  <NAME>Device001</NAME>
  <SERVICEID>90bd3f05-f45e-4fe2-941b-eb343caf6e43</SERVICEID>
</SENSOR>
```

Figure 4.3 Sensor Specification (Unknown)

Based upon this given information, a classification of the sensor, to a concrete modality or type, can be made. Instead a generic classification is given, of SENSOR. At this point the Tracking Middleware must begin to make queries of the service to determine the following: 1) is the sensor a tracking sensor, and 2) what is the expected behavior of the sensor with respect to its physical characteristics. Over the course of interaction with the sensor, additional information in the form of FRAME_RATE and RESOLUTION were obtained. Based upon our sKB, we find that these attributes are common to Vision-based sensors. Furthermore, the classifier could attempt to match the provided FRAME_RATE and RESOLUTION with known examples of Vision-based sensors within the sKB. In our sKB, we have two different types of Vision-based sensors: one with a FRAME_RATE of 30 frames per second, and one with a FRAME_RATE of 15 frames per second. With this knowledge, the classifier can attempt to provide a match to the previously unknown sensor. The results of this predictive classification are highlighted in Table I and the subsequent updated sensor specification, as maintained in the sKB, is shown in Fig. 4.4.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SENSOR>
  <NAME>Device001</NAME>
  <TYPE>VISION</TYPE>
  <MANUFACTOR>LOGITECH</MANUFACTOR>
  <VERSION>QUICKCAM</VERSION>
  <SERVICEID>90bd3f05-f45e-4fe2-941b-eb343caf6e43</SERVICEID>
  <QOS>
    <FRAMERATE>30</FRAMERATE>
  </QOS>
</SENSOR>
```

Figure 4.4 Sensor Specification (Classification Approach)

4.2 Trust Analysis

For the purpose of evaluating the trust of a sensor, we conducted a series of experiments to evaluate the sampling of evidences. The goal of this exercise was to evaluate the performance of the TA and its role in determining the trustworthiness of a sensor. In this exercise, a random distribution of sensors was introduced into the environment. The purpose behind this distribution was to evaluate how the trust of the sensor evolved over the course of the tracking exercise and the interaction between sensor and tracking system.

The initial set of experiments was conducted in order to focus on the integration of trust-based accuracy into the tracking system. Each sensor upon registration was assigned a corresponding trust agent (TA) that collected the specifications, per the service contract, and sampled the location data when available. This TA then reported the data back to the Tracking Middleware layer for analysis and ultimately form a trust-based decision to be used in the sensor selection process. These accuracy-related experiments were split into three categories based upon initial trust assignment: optimistic, pessimistic, neutral. In the optimistic approach, the tracking system made the assumption that all sensors, upon registration, were trustworthy – and thus, had a {B, D, U} tuple value of {1.0, 0, 0}. In the pessimistic approach, all of the sensors were assumed to be untrustworthy – and thus, had a tuple value of {0, 1.0, 0}. Finally, in the neutral approach, the tracking system assumed that insufficient data (less than 10 evidences for a given sensor) was available for the sensors and thus, a level of uncertainty persisted – and hence, a value of {0.33, 0.33, 0.33} was assigned for each sensor.

Table 4.2 Sensor QoS-based Comparison

Sensor ID	**Timestamp**	**Actual Response Time (ms)**	**Expected Response Time (ms)**
V001	2016-09-21 12:53:43	7	8
V001	2016-09-21 12:53:43	7	8
V001	2016-09-21 12:53:43	12	8
V001	2016-09-21 12:53:44	7	8

Table 4.2 highlights a sampling of a collection of evidences for a sensor. In this example, the QoS-based evaluation is about the response time of the sensor. This performance serves as the basis for a trust decision as to the trustworthiness of the sensor's data. As indicated in the previous chapter, if the sensor meets or exceeds the expected response time then the evidence is recorded as a positive evidence. If the sensor does not meet the expected response time then the evidence is recorded as a negative evidence.

The first experiment, in this set of exercises to evaluate this trust-based accuracy, was to verify that the trust tuple associated with the accuracy was indeed being properly set and maintained for an individual sensor. To validate the existence of such tuples for each of the different categories, we identified a sensor that we knew to be trustworthy, in terms of its accuracy, and one that we knew to be untrustworthy, in terms of its accuracy, and ran our algorithm against these sensors. We achieved the identification of sensors through offline calibration of the sensor devices that allowed us to collect evidences and evaluate them manually. In this test, only stationary sensors were used to mitigate the opportunity for additional error in regards to the location estimate into the final result. For each category and each sensor, we ran 100 data points through the algorithm and then examined the resulting trust scores. Tables 4.3, 4.4, and 4.5 highlight our findings for both the sensors in their respective categories – sensor A being the predefined trustworthy sensor and sensor B being the predefined untrustworthy sensor.

Table 4.3 Empirical Accuracy Analysis (Optimistic)

Sensor Name	**Belief**	**Disbelief**	**Uncertainty**
Sensor A	0.824	0.167	0.010
Sensor B	0.175	0.815	0.010

Table 4.4 Empirical Accuracy Analysis (Pessimistic)

Sensor Name	**Belief**	**Disbelief**	**Uncertainty**
Sensor A	0.813	0.176	0.010
Sensor B	0.098	0.892	0.010

Table 4.5 Empirical Accuracy Analysis (Neutral)

Sensor Name	**Belief**	**Disbelief**	**Uncertainty**
Sensor A	0.819	0.171	0.010
Sensor B	0.152	0.838	0.010

From Tables 4.3, 4.4, and 4.5, we can see that the algorithm appropriately determined the {B, D, U} tuples for the respective sensors. This initial analysis confirms the ground truth that we knew about each sensor going into the experiment regarding its trustworthiness, in terms of its accuracy. In each case, the algorithm provided a probability regarding the sensor's performance at 0.810 or higher.

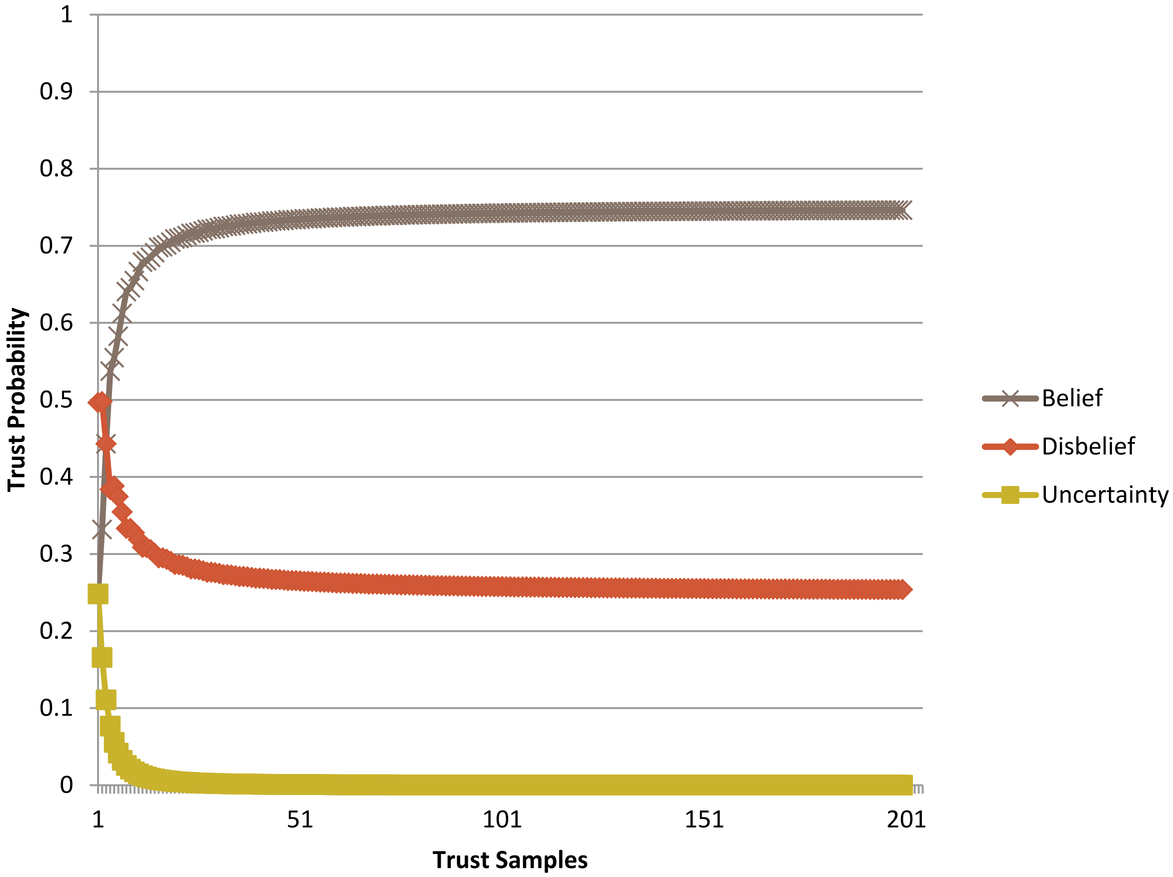


Figure 4.5   Trust Analysis

Fig. 4.5 shows the results of this exercise. The evolution of the trust tuple can be seen in this figure. As additional evidences are collected, as part of the tracking process, this sample sensors trust is modified accordingly due to the presence of the TA. Through these evidences it is shown how the uncertainty converges to 0, as data estimates are provided; while the disbelief decreases through the collection of positive evidences in favor of the sensor.

Table 4.6 Historical Trust Values

Timestamp	Belief Tuple	Weight
2016-09-21 16:11:17	{0.819, 0.171, 0.010}	0.9
2016-09-18 13:11:54	{0.813, 0.176, 0.010}	0.6
2016-09-17 15:23:02	{0.780, 0.210, 0.010}	0.5

Table 4.6 indicates the historical values associated with trust determination of a sensor. In this table, we demonstrate the importance of the exponentially weighted moving average into the determination of the trust of a sensor. In order to evaluate this concept, we conducted a series of experiments that collected 200 evidences from a sensor.

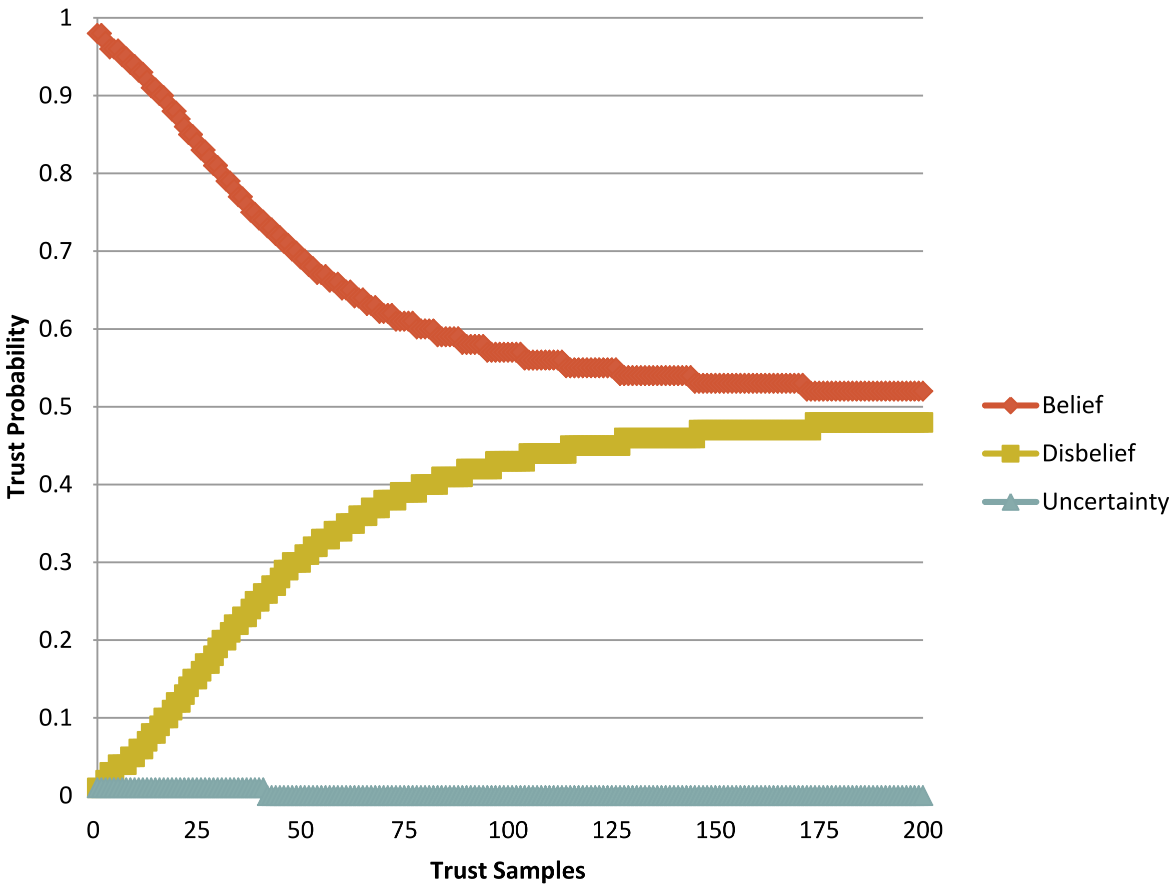


Figure 4.6 Optimistic Initial Trust

The duration of these experiments was to highlight the effect that historical trust values have on the overall trust determination. Fig. 4.6 shows the effect that our trust calculation has on a sensor with an initial trust value that is optimistic. Here each evidence and each subsequent trust determination is evaluated and then compared with the existing historical trust values. Once this has been done the final updated trust value is provided.

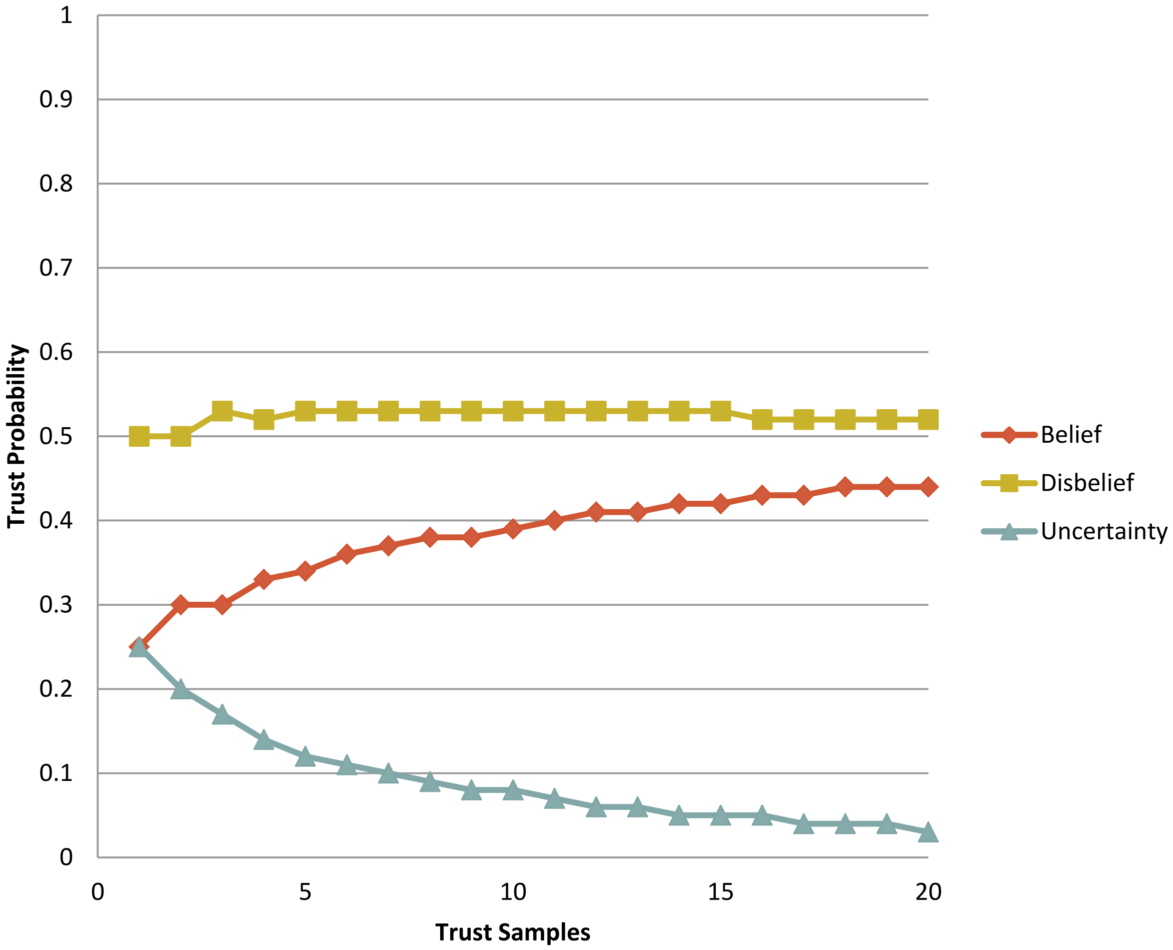


Figure 4.7 Trust Snapshot

Due to the nature of the data samples (e.g., 200 evidences collected for each sensor) it is impossible to show the actual fluctuation of the trust tuple values. In Fig. 4.7 we take a snapshot of an example trust calculation of a sensor. Here, we see that the sensor in question proves to be dominated by untrustworthy or negative evidences. The next experiment, we wanted to evaluate, was that of trust decay over time. In this case, we evaluated a Vision-based sensor. The reason for this type of sensor is the original motivation behind the infusion of trust. One of the primary limitations of Vision-based

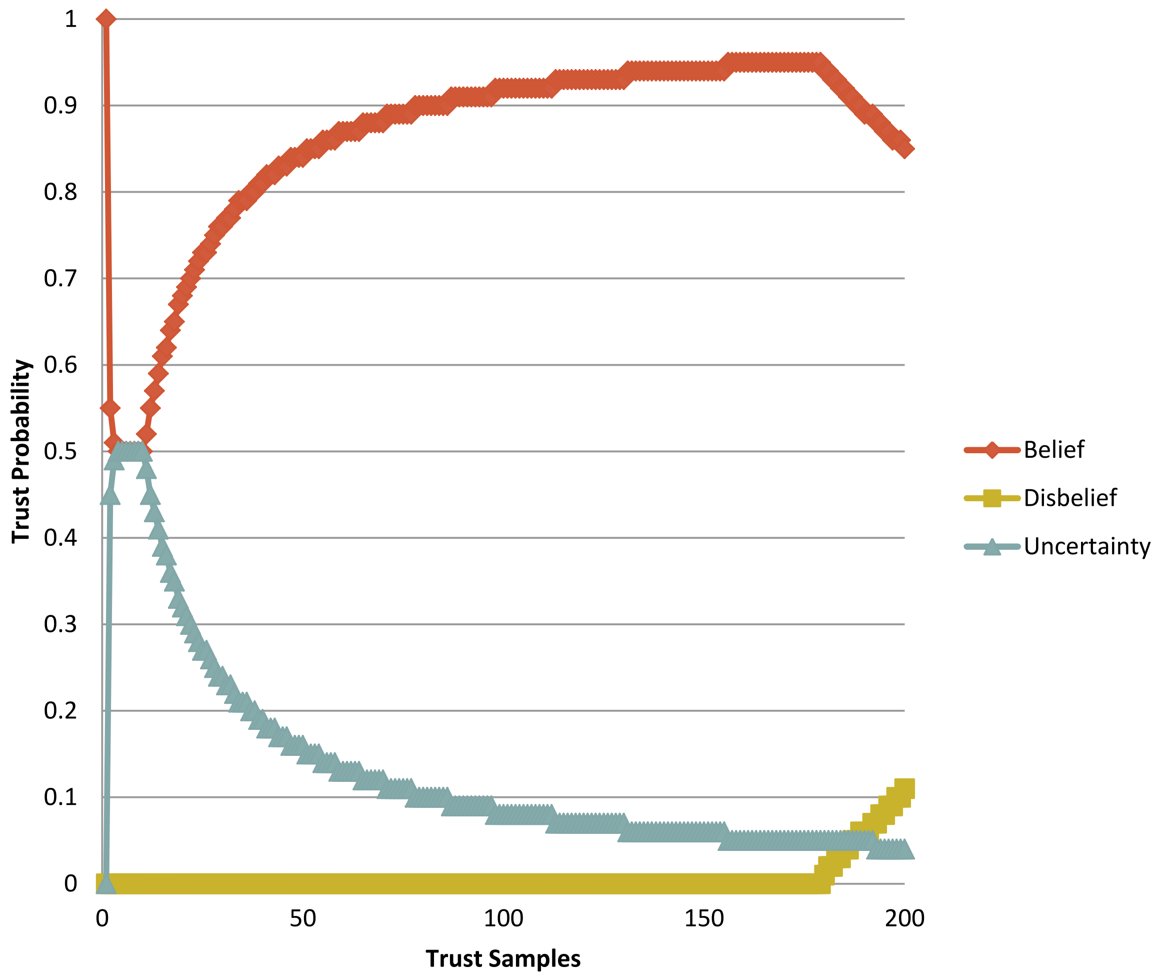


Figure 4.8 Optimistic Trust Fluctuation

sensors is that of occlusion. Occlusion occurs when the view between the sensor and the object being tracked is obscured. A second, but equally common, limitation is that of incorrect identification of the object being tracked. In our experimental setup, we used a common augmented reality pattern for visual recognition. In certain scenarios, it was possible for the sensor to incorrectly identify the wrong object. In this case, the data trust should appropriately reflect this. In Fig. 4.8 this is shown at the tail end of the chart in which the belief decreases and the disbelief increases. During this experiment, we documented the actual location of the object being tracked and then determined that the vision sensor was incorrectly identifying a monitor in the background due to the contrast in the color (black and white). Due to this, the positional estimate provided was inaccurate because of the misidentification. This is correctly reflected in the decrease in the belief in the sensor’s data.

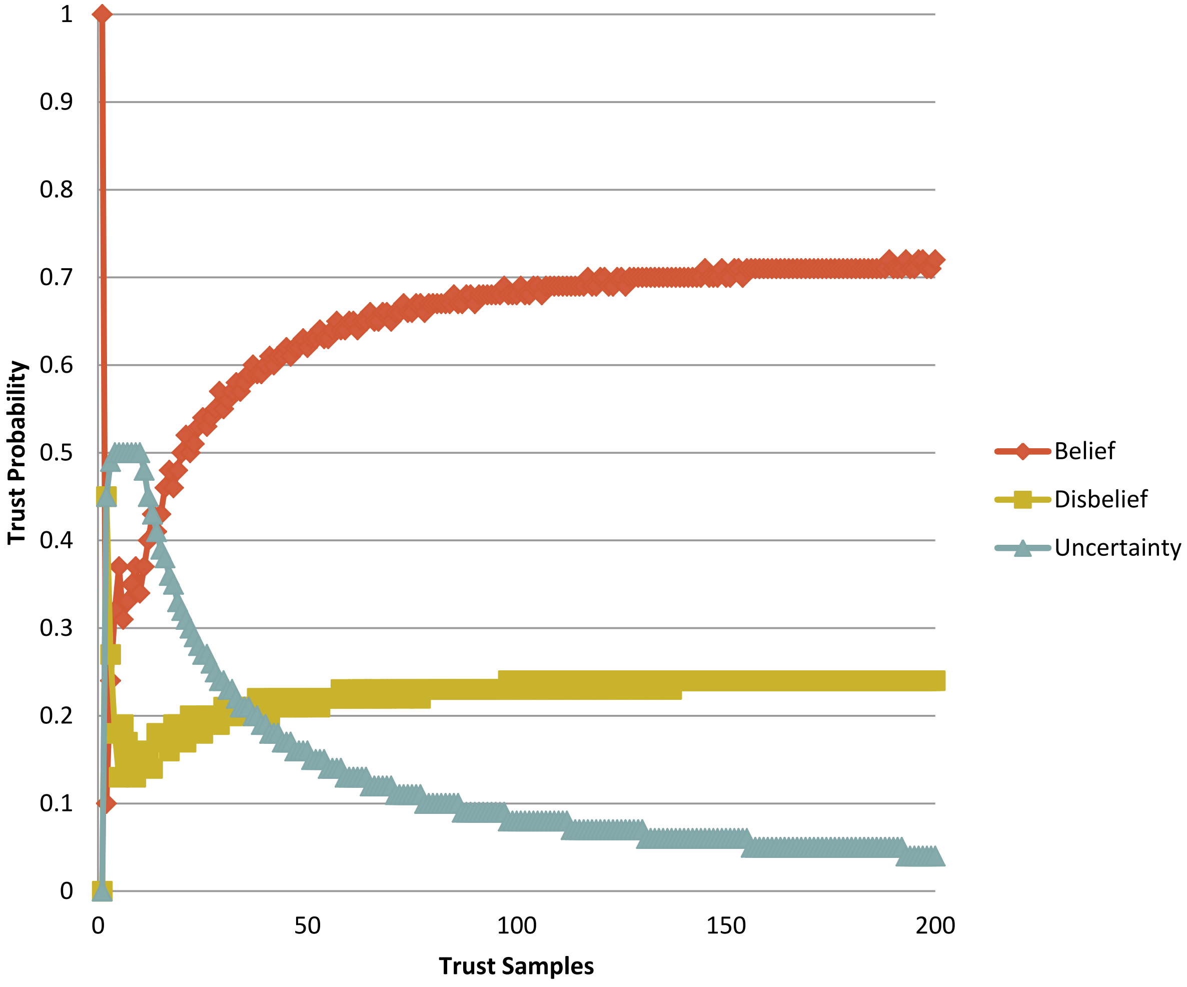


Figure 4.9 Trust Uncertainty Impact (Optimistic)

Fig. 4.9 demonstrates the impact that the value of uncertainty plays on the trustworthiness of a sensor. In this figure, it is seen how due to the lack of evidences the uncertainty dominates, even in the case where an optimistic approach is taken, until sufficient evidences can be obtained from the sensor. In this case, the experiment was continued for the duration of tracking, but as is seen in the first 25 data samples the uncertainty dominates the overall perception of the sensor and based upon the trust threshold established the sensor would be excluded from selection due to its {B, D, U} score.

Fig. 4.10 demonstrates the pessimistic initial approach in which the sensor is deemed untrustworthy. In this figure, the trust of the sensor is initially pessimistic – this may be a user defined attribute or may be based upon past historical trust data associated with the given sensor modality. In this case, until sufficient positive evidences are acquired, the sensor maintains this high level of disbelief. During this period of time, the sensor is not a candidate for sensor selection.

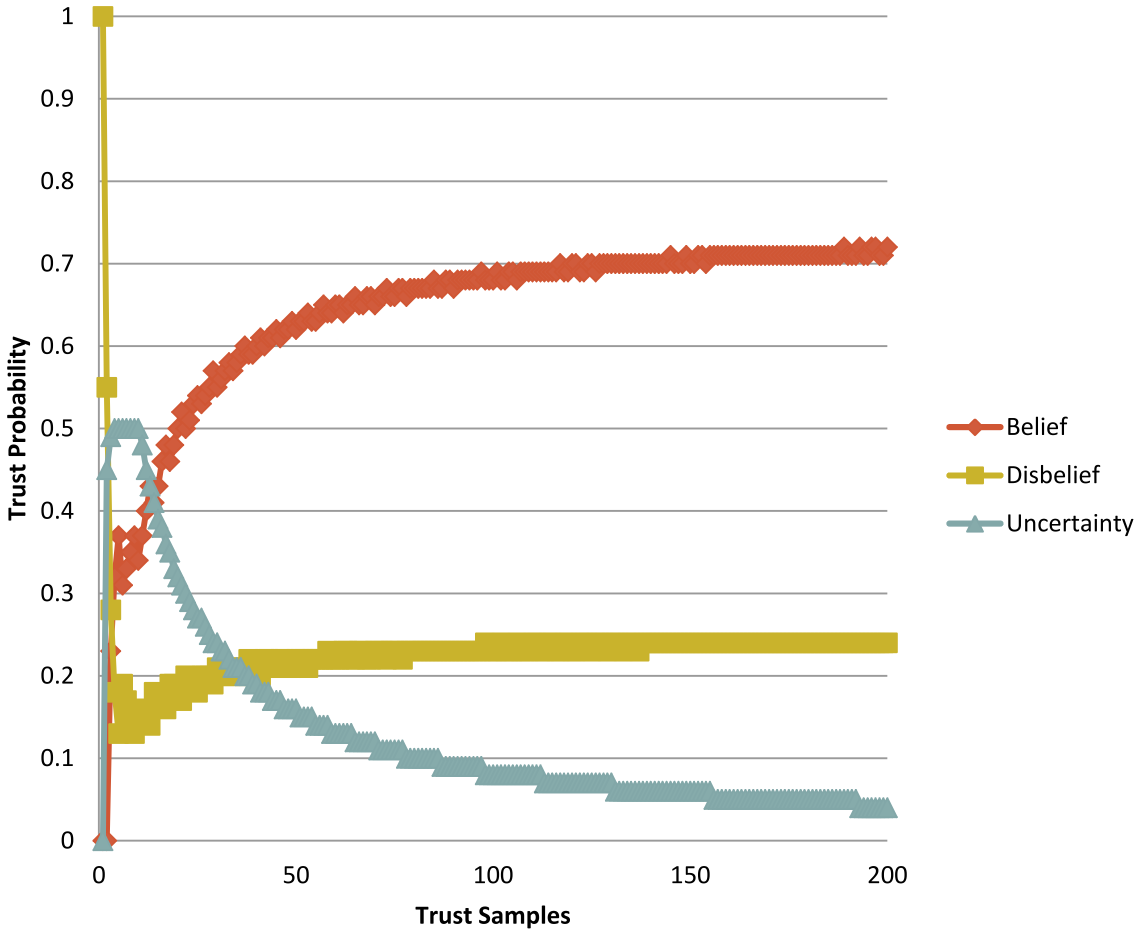


Figure 4.10 Trust Uncertainty Impact (Pessimistic)

In Fig. 4.11, we see a sensor that over the collection of evidences shows the evolution of the trust tuple over a period of time and its ability to dynamically adjust to changes in the sensor's performance. A fluctuation in the belief is then seen due to the presence of misidentification – once the object is correctly identified the belief value grows.

In this set of experiments, we have empirically validated the infusion of trust in the eDOTS. We have shown the impact that the initial assignment of trust has on a sensor's trustworthiness during its use. We have demonstrated that our trust algorithm

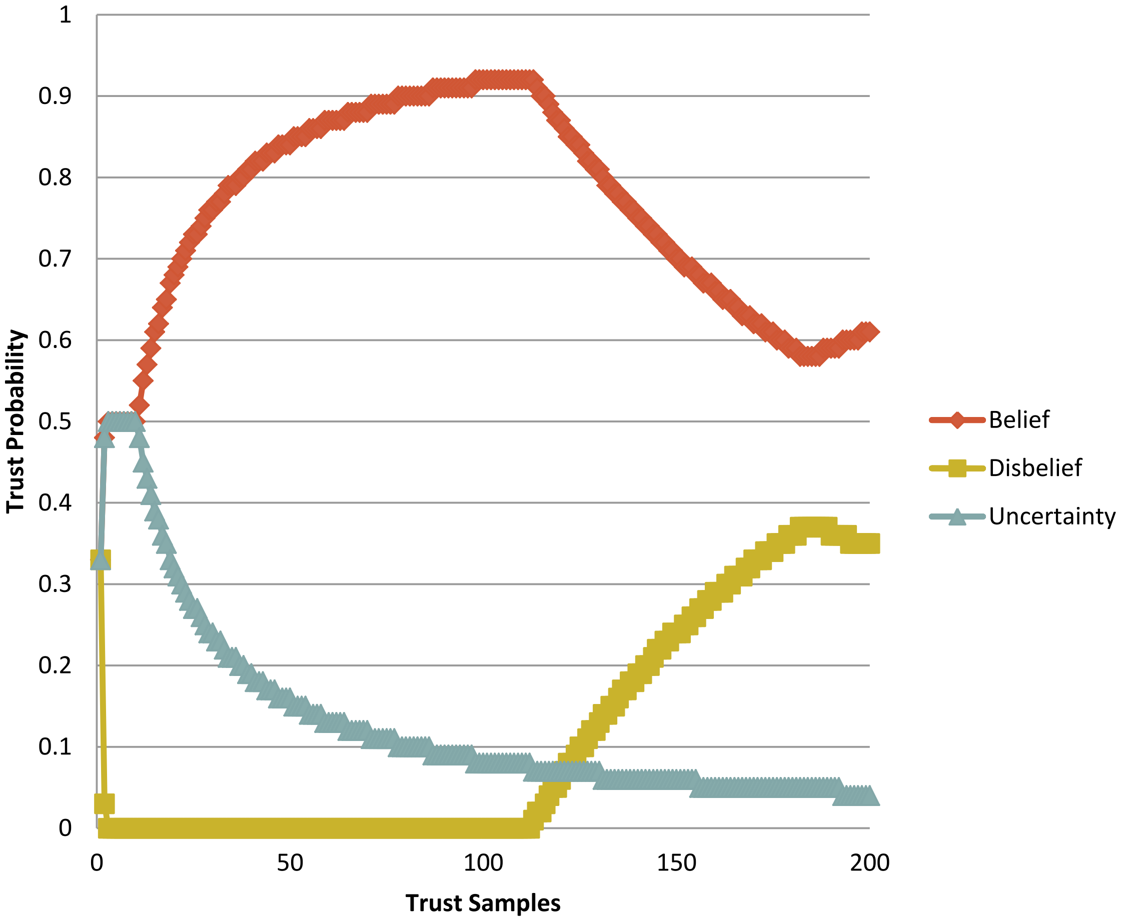


Figure 4.11 Trust Adjustment

provides an agile approach to handling the evolution of trust over a period of time and the impacts that evidences have on the determination of the trust tuple.

4.3 Reliability Analysis

For the purpose of evaluating the reliability of a sensor, we conducted a set of experiments that would highlight the role of the RA within the eDOTS. The first of these experiments was to evaluate the overall performance of the RA in classifying a sensor to be reliable or not. For this exercise, we included sensors that had a previous history of being highly reliable. These sensors were identified as highly reliable due to their performance in past exercises. This historical data was obtained from the reliability hierarchy as described in Chapter 3, as part of the sKB. Here we focus on the classification of responsiveness of each sensor with regards to its specification.

Similar in nature to the trust-based experiments, the reliability-based experiments were split into three categories based upon initial reliability assignment: optimistic, pessimistic, neutral. In the optimistic approach, the system made the assumption that all sensors, upon registration, were reliable – and thus had a tuple value of {1.0, 0, 0}. In the pessimistic approach, all of the sensors were assumed to be unreliable – and thus, had a tuple value of {0, 1.0, 0}. Finally, in the neutral approach, the system assumed that insufficient data was available for the sensors and thus, a level of uncertainty persisted – and hence, a tuple value of {0.33, 0.33, 0.33} was assigned as the initial reliability value to each sensor. Here we discarded the inherited reliability values from the base sensor modality; the reason for this omission was in order to strictly validate the application and assignment of the reliability tuple for each sensor based upon collected evidences.

To empirically validate the proper assignment of reliability, for each of the different categories, we again identified a sensor that we knew to be reliable in terms of its performance (lack of failures – e.g., responsiveness) to serve as our ground truth and one that we knew to be unreliable in terms of its unpredictable responsiveness and ran our algorithm against the sensor. For each category and each sensor, we ran a tracking exercise in which we sampled data points throughout, in order to demonstrate sufficient communication between the Sensor Service and the Tracking Middleware layer. Tables 4.5, 4.6, and 4.7 highlight our findings, for both the sensors in their respective categories - sensor C being the predefined reliable sensor and sensor D being the predefined unreliable sensor.

Table 4.7 Empirical Reliability Analysis (Optimistic)

Sensor Name	**Belief**	**Disbelief**	**Uncertainty**
Sensor C	0.95	0.01	0.04
Sensor D	0.48	0.48	0.04

Table 4.8 Empirical Reliability Analysis (Pessimistic)

Sensor Name	**Belief**	**Disbelief**	**Uncertainty**
Sensor C	0.86	0.10	0.04
Sensor D	0.00	0.96	0.04

Table 4.9 Empirical Reliability Analysis (Neutral)

Sensor Name	**Belief**	**Disbelief**	**Uncertainty**
Sensor C	0.93	0.02	0.04
Sensor D	0.28	0.68	0.04

From Tables 4.5, 4.6, and 4.7, we see that the algorithm appropriately, per our prior knowledge of each sensor and a collection of 100 evidences, determined the {B, D,

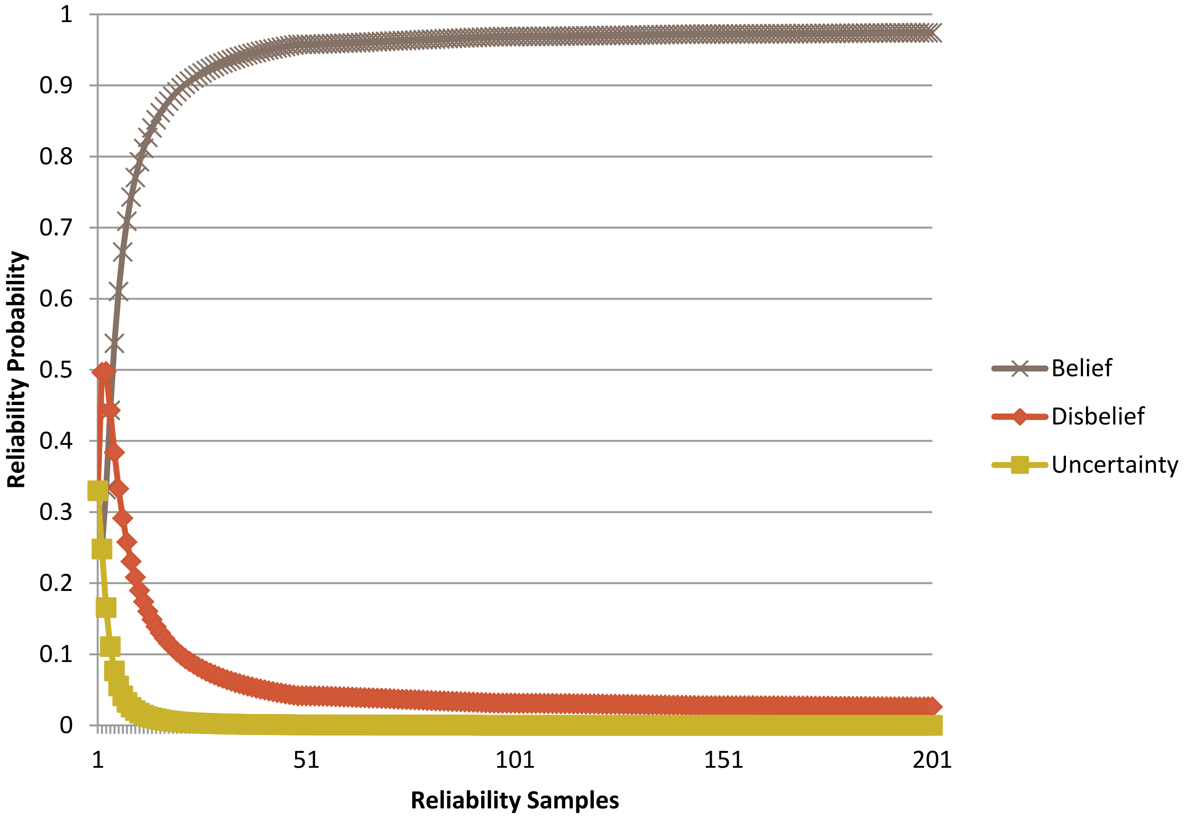


Figure 4.12 Reliability Analysis

U} tuple for the respective sensors. This confirms the expectations we had regarding the performance of each sensor with respect to its reliability.

Fig. 4.12 highlights the findings of the results when historical reliability data is inherited as part of the initialization process of a sensor during registration. In this figure, it can be seen how the reliability of the individual sensor changes over time due to the increased information, or evidences, collected regarding its performance. The sample sensor, in the graph, begins with a higher reliability probability score based upon the pre-existing knowledge stored in the sKB regarding that specific class of sensor. This score is then factored into the collected evidences through sampling of the sensors performance during the tracking exercise. Here the sensor has been determined to be reliable based upon its high belief level as the result of its performance. This indicates a lack of failures by the sensor during the tracking exercise.

Fig. 13 highlights the impact of an initial neutral reliability opinion on the determination of the reliability tuple. Here, it is shown how the belief of the sensor, and the corresponded plotted line tends to jump, indicating potential missed communication points. Another point of note in this figure is the high initial uncertainty due to lack of evidences and the initial assumption of the unknown behavior of the sensor.

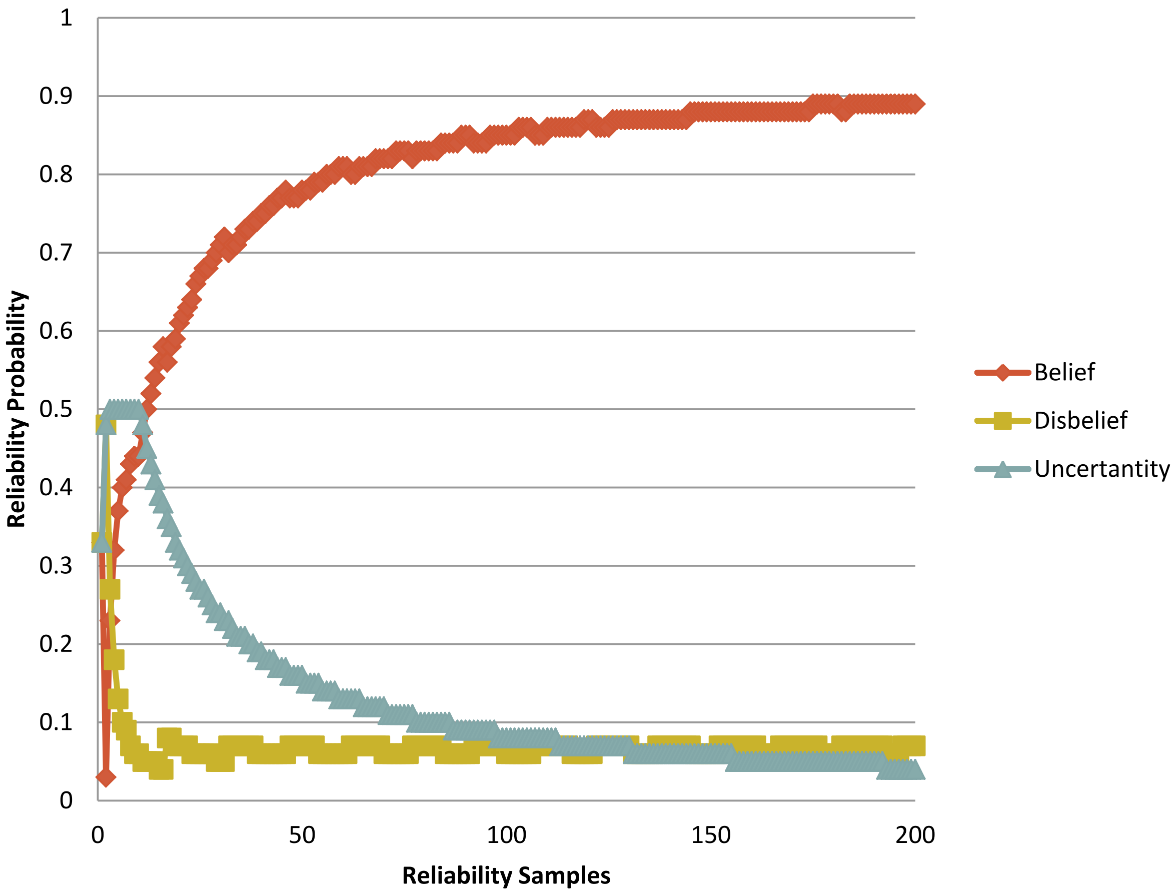


Figure 4.13 Neutral Reliability Opinion

Fig. 4.14 highlights the reliability values for a sensor that proves to be unreliable due to collected evidences. In this example, the sensor in question responds to every other tracking inquiry and thus, provides a reliability of 50%. Upon examination of the data logs recording during tracking, this sensor would block any incoming requests while attempting to process the data. As a result of this action, the sensor would only respond to half of the inquiries that were requested of it for data. This sensor was actually a prime

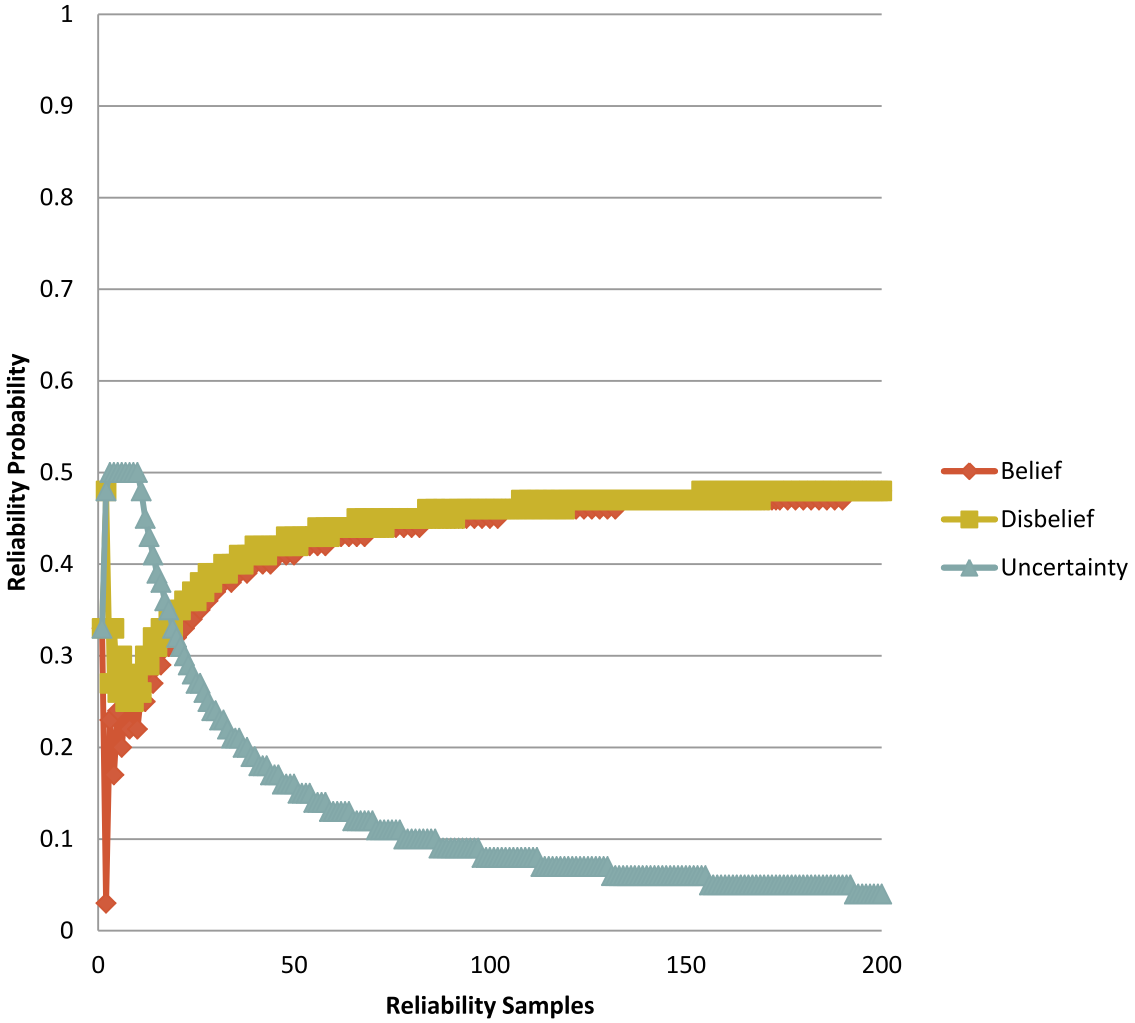


Figure 4.14 Unreliable Sensor

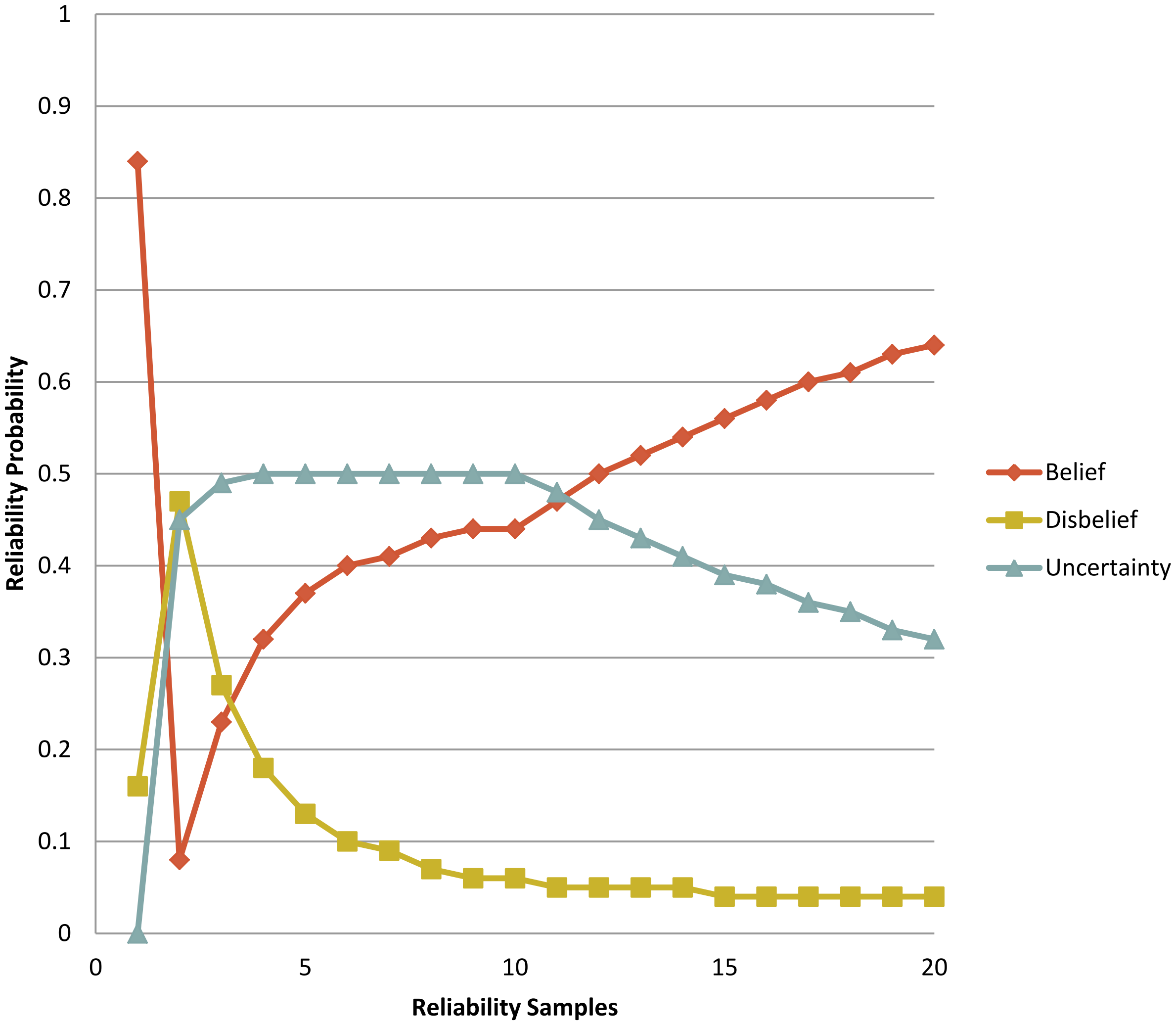


Figure 4.15 Reliability Snapshot

candidate for the separation of trust and reliability as it provided very trustworthy data responses.

Fig. 4.15 provides a snapshot view of the reliability of a sensor. In this view, the initial inherited reliability value is shown. Due to the lack of evidences and after the initial collection of evidences the reliability belief begins to normalize due to sufficient evidences. It is clear in this figure the impact that uncertainty has on the calculation of the reliability tuple until sufficient evidences are present. This snapshot also shows the role that each respective evidence has on each value of the reliability tuple.

In this set of experiments, we have empirically validated the infusion of reliability in the eDOTS. We have shown the impact that the initial assignment of reliability values has on a sensor's reliability during its use. We have demonstrated that our reliability algorithm provides an agile approach to handling the evolution of reliability over a period of time and the impacts that evidences have on the determination of the reliability tuple. Finally, we have shown the role that both inherited reliability values and uncertainty play in the overall determination of whether a sensor is deemed reliable or not.

4.4 Sensor Selection

For the evaluation of the enhanced sensor selection process, we wanted to evaluate not only the accuracy of the system, with this selection technique applied, but also to empirically validate the overall cost associated with the application of this new approach. The first experiment was designed to illustrate the improvement of accuracy obtained through this enhanced process. The accuracy was verified through physical measurements recorded as a sample object moved through the environment. These measurements were recorded by hand and given timestamps in order to provide an offline comparison of the systems performance. The results of this experiment are shown in Table 4.10.

As part of this exercise, we considered the results from the previous evaluation of the Trust and Reliability. During the experimentation phase, of the previous discussion, we evaluated the sensor selection based upon these QoS-based criteria. Here we took actual physical measurements as we moved a tracking object around our indoor environment. During this process, we manually recorded each measurement with a timestamp (computed digitally) and then compared these physical measurements with the estimated measurements. We also, examined the log files of each sensor to examine which sensors were used in the sensor selection process to match the further evaluate the reliability and trust.

Table 4.10 Tracking Accuracy (Meters)

	With QoS-aided Sensor Selection	**Without QoS-aided Sensor Selection**
Accuracy (meters)	0.97	1.35

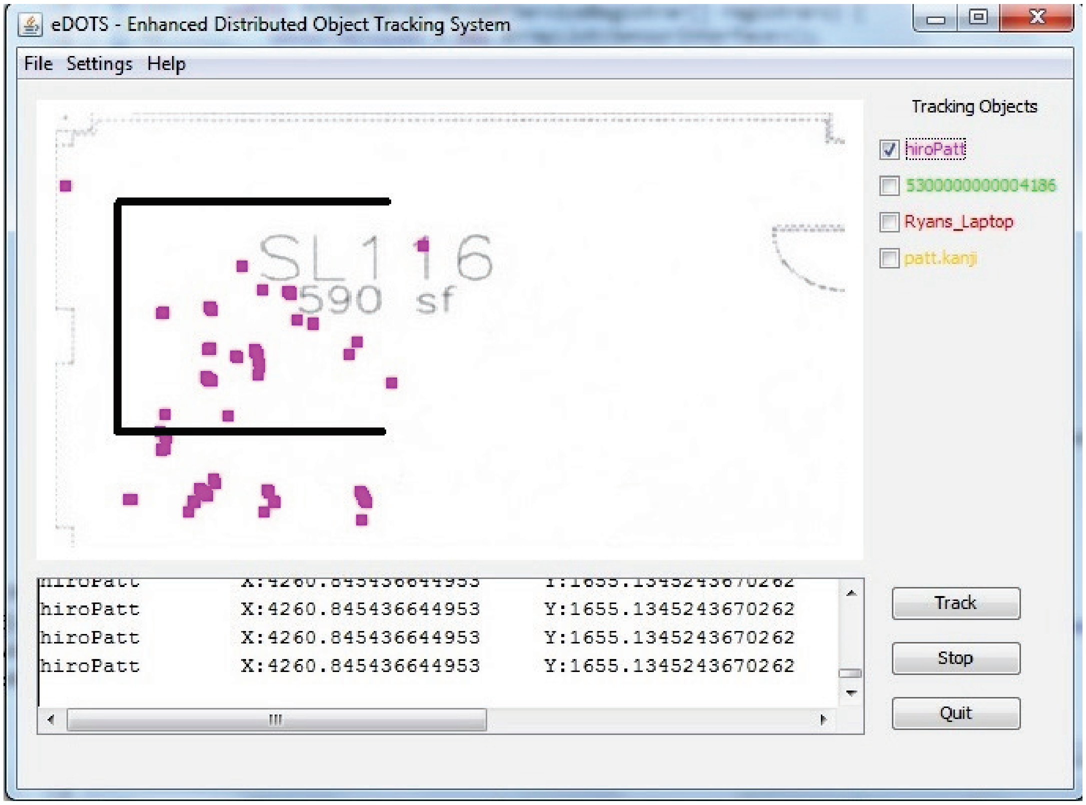
As shown in Table 4.10, through the inclusion of QoS-based sensor selection process we have demonstrated, empirically, that improvement can indeed be made with respect to the overall accuracy provided by the tracking system. This accuracy improvement is a direct result of the pruning of either unreliable or untrustworthy data sources. We then compare this accuracy, as shown in Table 4.11, with three other related and prominent ITS approaches: Google Indoor Maps [3], UnLoc [24], and Ekahau [86].

Table 4.11 Tracking Accuracy Comparison (Meters)

ITS	**Mean Accuracy**
eDOTS*	0.97
Google Indoor Maps	1.22
UnLoc	1.50
Ekahau	0.91

* Opportunistic eDOTS using QoS-aided Sensor Selection

Here, we demonstrate that the accuracy obtained using the Opportunistic eDOTS with QoS-aided Sensor Selection outperforms two approaches (Google Indoor Maps and UnLoc) in its tracking accuracy. While the Ekahau system does slightly (0.06 meters) outperform our approach their system is built upon a static infrastructure and requires



Tracking Objects

☑ hiroPatt
□ 5300000000004186
□ Ryans_Laptop
□ patt.kanji

```
hiroPatt       X:4260.845436644953      Y:1655.1345243670262
hiroPatt       X:4260.845436644953      Y:1655.1345243670262
hiroPatt       X:4260.845436644953      Y:1655.1345243670262
hiroPatt       X:4260.845436644953      Y:1655.1345243670262
```

Figure 4.16 Without QoS-aided Sensor Selection

proprietary equipment for the purpose of tracking. In the case of Google Indoor Maps and UnLoc no special tracking infrastructure is necessary for tracking.

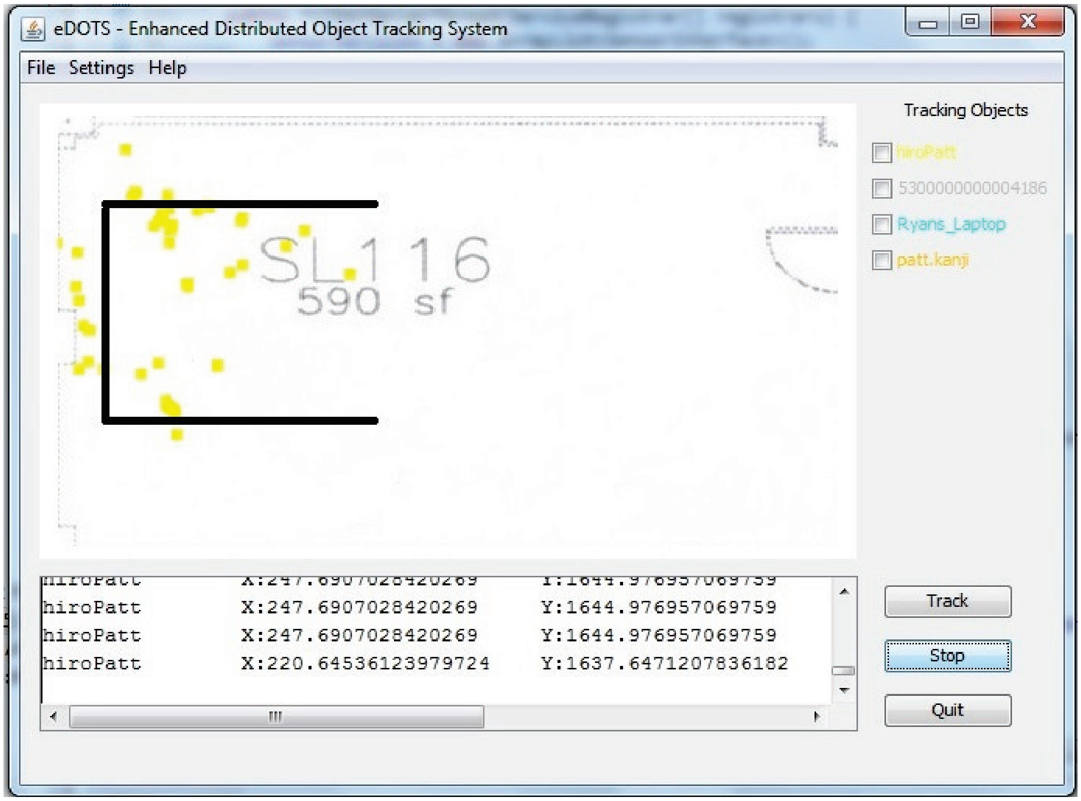
A sample of the effects of the pruning is shown in Fig. 4.16 and 4.17. Fig. 4.16 indicates the use of the traditional sensor selection approach (of ranking and pruning) in the eDOTS. Using this approach, two sensors involved in the tracking process were providing untrustworthy data due to misidentification of a pattern. This skewed the data estimates as is shown in the figure. In Fig. 4.17, the use of the QoS-aided sensor selection eliminated these untrustworthy sensors and thus the accuracy was improved.

The second experiment that we conducted in this set was to evaluate the overall cost, in terms of run time, associated with this enhanced sensor selection on the indoor tracking system. In order to evaluate the performance of the system, we empirically quantified the end-to-end run time of the system both with and without the QoS-aided sensor selection. We define the end-to-end run time to be the time from the request is issued to track an object to the point where the tracking system displays the location estimate to the user. The results of this test are shown in Table 4.12.

Table 4.12 Mean End-to-End Runtime (Milliseconds)

With QoS-aided Sensor Selection	Without QoS-aided Sensor Selection
53	49

One of the obvious points to note with this experiment is the impact of the number of sensors and the perceived quality of the sensors involved. In order to mitigate



```
hiroPatt        X:247.6907028420269     Y:1644.976957069759
hiroPatt        X:247.6907028420269     Y:1644.976957069759
hiroPatt        X:247.6907028420269     Y:1644.976957069759
hiroPatt        X:220.64536123979724    Y:1637.6471207836182
```

Figure 4.17 With QoS-aided Sensor Selection

this impact, we used the same set of sensors for both tests. We restricted the environment to only the sensors currently present thus mitigating the negative impact that a dynamic environment could play on the overall outcome of this run time comparison. This design choice was necessary in order to compare the two side-by-side for performance analysis as previously noted as a challenge associated with the opportunistic tracking approach. Other external factors (e.g., data fusion technique, number of sensors, which sensors are currently tracking, etc.) have a significant impact on this end-to-end runtime evaluation. A tradeoff therefore exists between the cost incurred for the enhanced sensor selection and the gain obtained with a higher accuracy, in terms of location estimate. We discuss and empirically examine this tradeoff later in this chapter with the optimization function described in Chapter 3.

4.5 Multi-Sensor Data Fusion

For multi-sensor data fusion, we wanted to evaluate the impact on the improved sensor selection as part of this process. Prior empirical evaluations, regarding the performance of both averaging and an Extended Kalman Filter approaches, can be found in [64]. We use these previous studies as a baseline for the purpose of comparing the existing approach with our proposed enhanced sensor selection process. In this exercise we just compared the runtime of the data fusion component – the full end-to-end runtime of the tracking system is shown in the previous subsection.

Table 4.13 Mean Data Fusion Runtime (Milliseconds)

Technique	**Runtime (Existing)**	**Runtime (Enhanced)**
Averaging	12 milliseconds	< 0 milliseconds
EKF	52 milliseconds	11 milliseconds

As shown, in Table 4.13, the runtime of the existing data fusion approaches are reduced by around a factor of five with respect to their performance. The reason behind this improvement is due to the improved sensor selection process (as described in the previous subsection). This improved process allowed for better filtering of the sensor data prior to arrival at the Fusion service. This filtering, by the selection component, allowed less overhead, in terms of work, by the fuser and instead allowed the data to be processed in a more streamlined and efficient fashion. This table represents just the runtime required for the data fusion process to complete, the complete end-to-end runtime of the system is shown in Table 4.12.

In the previous subsection, we covered the outcome of the improved sensor selection, with respect to the accuracy obtained by the system. While this process is greatly impacted by the selection of the "right" set of sensors, it is also greatly impacted by the data fusion process selected. In the case of the accuracy, described in Table VIII, the data fusion technique used was the Extended Kalman Filter approach. This approach has been shown, in [64], to provide a more accurate positional estimate when compared to simple averaging.

4.6 Tradeoff Optimization

For this set of experiments, the goal was to evaluate the tradeoff function that we proposed in Chapter 3. Here, we made use of the decentralized pursuit learning game algorithm as described in [76]. This tradeoff between cost (time) and gain (accuracy) is evaluated and the optimal sensor selection is determined based upon the optimization function proposed in Chapter 3.

For the purpose of analyzing the convergence of the decentralized pursuit learning game algorithm, we tested the performance on a set of five sensors. The reason for this controlled cutoff is due to the average number of sensors ever tracking one object at one given instance of time. This was found through extensive study over the course of the work of this thesis to be the mean number for a typical environment. Using this experimental setup, we then utilize the action set of either to send the data from the sensor or to not send the data from the sensor. As shown in previous work [64], the end-to-end runtime of the tracking system is primarily dominated by two components: sensor selection and data fusion. Therefore, by reducing the number of sensors the goal is that the runtime of the system will be improved.

The goal of this exercise was to evaluate the performance of the algorithm as it converged to the value of 0.85. We ran this algorithm during the course of a regular tracking exercise involving the five sensors as specified. The results of this evaluation are shown in Fig. 4.18.

Here, we focus on the performance of the algorithm, in terms of runtime, as this is a primary factor in the use of an ITS for the role of tracking. This runtime is represented by the cost component of the optimization function. The runtime here is the time required for the algorithm to converge to the optimal set of sensors for selection. The results of this experiment are show in Table 4.13.

Table 4.14 Optimization Performance

	Decentralized Pursuit Learning Game Algorithm
Average Runtime (ms)	6.5

Here, we see the mean runtime cost associated with the convergence of the algorithm. This cost when factored into the existing data fusion process using the Extended Kalman Filter, outperforms the existing approach while providing a higher degree of positional accuracy with the tracking estimate. This demonstrates that the use of this algorithm improves the overall tracking ability of the eDOTS.

In this chapter, we have discussed how we empirically validated the proposed implements as provided in Chapter 3. The findings in this chapter reinforce our hypothesis that the inclusion of Trust and Reliability as separate selection criteria can improve the overall accuracy of the system. We have shown that as a result of this

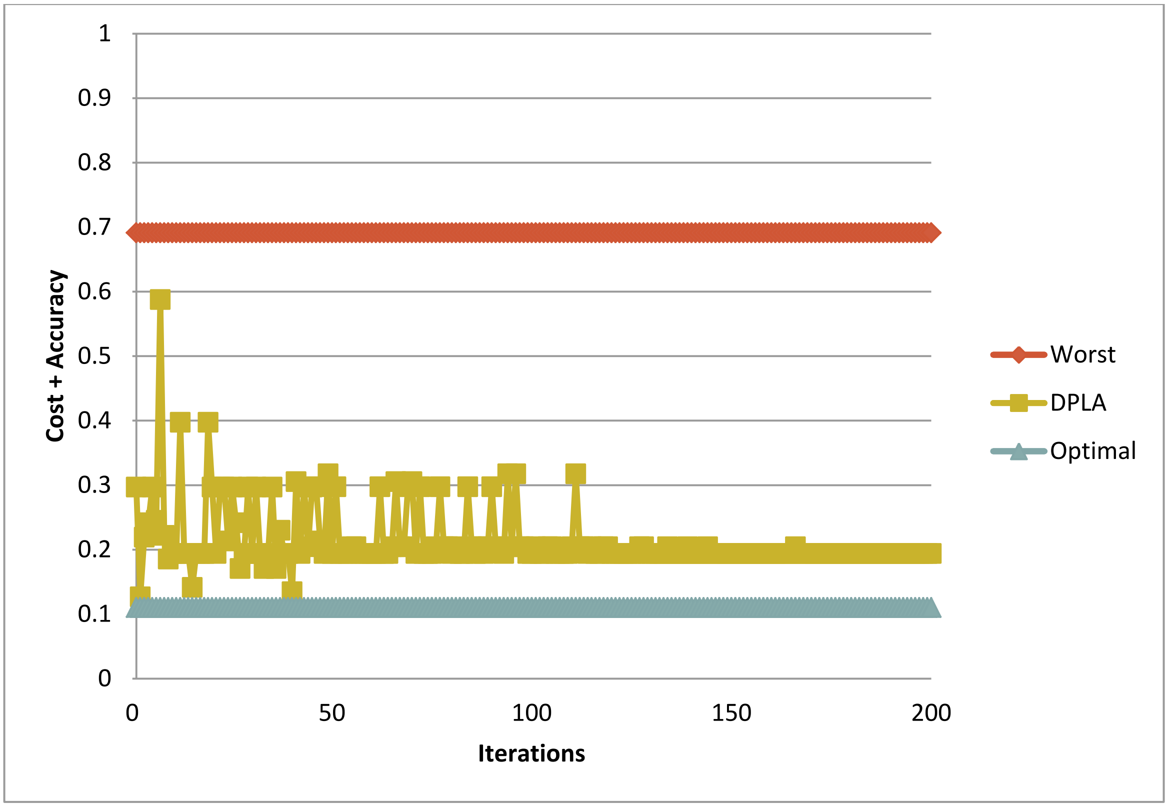


Figure 4.18 Optimization Analysis

infusion of these two QoS-based selection criteria that we can also, through optimization and learning algorithms, find the optimal tradeoff between cost and gain when it comes to providing this improved accuracy.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

In conclusion, this thesis has shown that sensor classification is a vital first step in the discovery process of sensors for indoor tracking. This step is especially important in an opportunistic setup, as the tracking infrastructure is not known *a priori* and cannot be handcrafted. We have provided a comparison of various classification techniques (Rule-based Classifiers, Decision Tree Classifiers, and Naïve Bayes Classifiers) and shown their respective accuracy (80% or higher), in terms of appropriate classification, when provided data that contains samples both complete and incomplete. This use of a classification process allows for improvement in the sensor selection process by providing additional information regarding a sensors' performance.

In this thesis, we have also demonstrated the ability to classify a sensor based upon its trust and reliability, through collected evidences for the use in the sensor subset selection process. We have proposed that these two QoS-based criteria should be evaluated and treated independently of one another. We have empirically demonstrated that through the infusion of trust and reliability, as separate QoS-based selection criteria, we are able to make improvements to the problem of subset selection. This improvement in selection aids in the improvement of overall accuracy, in terms of the positional estimate, that the tracking system can provide.

Through this work, we have explored the role that this QoS-infused sensor selection has on the data fusion component. Specifically, we have demonstrated how selecting the “right” set of sensors reduces the time required to complete the fusion process. This improved process also directly relates back to the accuracy obtained when integrated into an ITS. Through this improvement in efficiency, with respect to computational time, and accuracy we have demonstrated that the main benefactor from this improved sensor selection is the data fusion component of an ITS.

Finally, we have provided a tradeoff optimization function that attempts to maximize the accuracy of the tracking system while minimizing the overhead, in terms of time, associated with obtaining such a measurement. This tradeoff provides a benchmark for customizing the ITS to the needs of a specific application domain. All of this was shown and empirically validated on a prototype ITS, the eDOTS. We believe that this work has provided an improvement to the overall challenge of sensor subset selection through the infusion of Trust and Reliability as separate selection criteria. This improved subset selection then can directly aid the data fusion component of a system which ultimately leads to improved results. In the case of an ITS, this improved process leads to improved tracking accuracy which address the problem, as identified in Chapter 1, of accurate indoor tracking. We believe that the methods and techniques proposed as part of our framework can help to advance the state of the art with respect to indoor tracking, through improved sensor selection, and further improve the overall accuracy that such systems can produce in practice.

5.2 Future Extensions

Future extensions of this work could include but is not limited to:

- A scalability study involving the sensor selection (including trust and reliability models) process.
- The inclusion of malicious sensors into the existing framework to evaluate the security impact that such sensors would pose to the tracking accuracy and how the trust and reliability infusion would deal with such malicious sensors.
- Integration and analysis of other sensor modalities include the impact of non-tracking sensors and their ability to add contextual awareness to the tracking exercise.
- Integration the proposed framework into other existing commercial tracking systems and adoption in mainstream use in a variety of application domains (e.g., asset tracking, medical tracking, emergency rescue, etc.).

LIST OF REFERENCES

LIST OF REFERENCES

[1] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology,* vol. 64, pp. 1263-1278, 2015.

[2] Skyhook. (2016, 2016). *Skyhook Precision Location*. Available: http://www.skyhookwireless.com/, Accessed: 2016.

[3] Google. (2016, 08/18/2016). *Google Indoor Maps*. Available: https://www.google.com/maps/about/partners/indoormaps/, Accessed: 2016.

[4] A. Jøsang, "An Algebra for Assessing Trust in Certification Chains," in *NDSS*, 1999, p. 80.

[5] A. Dâmaso, N. Rosa, and P. Maciel, "Reliability of wireless sensor networks," *Sensors,* vol. 14, pp. 15760-15785, 2014.

[6] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE,* vol. 85, pp. 6-23, 1997.

[7] A. E. Abo-Elhassab, S. M. A. El-Kader, S. Elramly, and H. S. Eissa, "Localization for Fire Rescue Applications by Using Wireless Sensor Networks," *International Journal of Computer Science Issues (IJCSI),* vol. 12, p. 45, 2015.

[8] S.-C. Kim, Y.-S. Jeong, and S.-O. Park, "RFID-based indoor location tracking to ensure the safety of the elderly in smart home environments," *Personal and ubiquitous computing,* vol. 17, pp. 1699-1707, 2013.

[9] A. Pietrabissa, C. Poli, D. G. Ferriero, and M. Grigioni, "Optimal planning of sensor networks for asset tracking in hospital environments," *Decision Support Systems,* vol. 55, pp. 304-313, 2013.

[10] H. M. Khoury and V. R. Kamat, "Evaluation of position tracking technologies for user localization in indoor construction environments," *Automation in Construction,* vol. 18, pp. 444-457, 2009.

[11] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 37, pp. 1067-1080, 2007.

[12] H. Hakim, R. Renouf, and J. Enderle, "Passive RFID asset monitoring system in hospital environments," in *Proceedings of the IEEE 32nd annual northeast bioengineering conference*, 2006, pp. 217-218.

[13] S. S. Saab and Z. S. Nakad, "A standalone RFID indoor positioning system using passive tags," *IEEE Transactions on Industrial Electronics,* vol. 58, pp. 1961-1970, 2011.

[14] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless networks,* vol. 10, pp. 701-710, 2004.

[15] R. Tesoriero, J. A. Gallud, M. Lozano, and V. M. R. Penichet, "Using active and passive RFID technology to support indoor location-aware systems," *IEEE Transactions on Consumer Electronics,* vol. 54, pp. 578-583, 2008.

[16] S. Chan and G. Sohn, "Indoor localization using Wi-Fi based fingerprinting and trilateration techiques for LBS applications," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 38, p. 4, 2012.

[17] 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*, 2000, pp. 775-784.

[18] E. Mok and G. Retscher, "Location determination using WiFi fingerprinting versus WiFi trilateration," *Journal of Location Based Services,* vol. 1, pp. 145-159, 2007.

[19] R. Faragher and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA*, 2014.

[20] A. M. Phadke, "Designing and experimenting with e-DTS 3.0," Purdue University, M.S. Thesis, 2014.

[21] J. M. Eklund, T. R. Hansen, J. Sprinkle, and S. Sastry, "Information Technology for Assisted Living at Home: building a wireless infrastructure for assisted living," in *Proceedings of the 27th Annual International Conference of the IEEE Engineering In Medicine and Biology Society (EMBS)*, 2005, pp. 3931-3934.

[22] C. Soto, B. Song, and A. K. Roy-Chowdhury, "Distributed multi-target tracking in a self-configuring camera network," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1486-1493.

[23] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 421-430.

[24] 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*, 2012, pp. 197-210.

[25] R. Rybarczyk, R. Raje, and M. Tuceryan, "Infusing trust in indoor tracking: poster," in *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, 2016, pp. 374-377.

[26] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Classification: Algorithms and Applications,* p. 37, 2014.

[27] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems,* vol. 21, 1991.

[28] L. Calderoni, M. Ferrara, A. Franco, and D. Maio, "Indoor localization in a hospital environment using Random Forest classifiers," *Expert Systems with Applications,* vol. 42, pp. 125-134, 2015.

[29] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," ed, 2007.

[30] K. Trawiński, J. M. Alonso, and N. Hernández, "A multiclassifier approach for topology-based WiFi indoor localization," *Soft Computing,* vol. 17, pp. 1817-1831, 2013.

[31] N. Hernandez, M. Ocana, S. Humanes, P. Revenga, D. P. Pancho, and L. Magdalena, "A WiFi-based software for indoor localization," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 2345-2351.

[32] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, pp. 41-46.

[33] M. Kessel and M. Werner, "SMARTPOS: Accurate and precise indoor positioning on mobile phones," in *Proceedings of the First International Conference on Mobile Services, Resources, and Users, MOBILITY*, 2011, pp. 158-163.

[34] C.-L. Wu, L.-C. Fu, and F.-L. Lian, "WLAN location determination in e-home via support vector classification," in *Networking, sensing and control, 2004 IEEE international conference on*, 2004, pp. 1026-1031.

[35] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks,* vol. 47, pp. 825-845, 2005.

[36] N. Li, J. Chen, Y. Yuan, X. Tian, Y. Han, and M. Xia, "A Wi-Fi Indoor Localization Strategy Using Particle Swarm Optimization Based Artificial Neural Networks," *International Journal of Distributed Sensor Networks,* vol. 2016, 2016.

[37] M. Moreno-Cano, M. A. Zamora-Izquierdo, J. Santa, and A. F. Skarmeta, "An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings," *Neurocomputing,* vol. 122, pp. 116-125, 2013.

[38] N. Cao, S. Choi, E. Masazade, and P. K. Varshney, "Sensor Selection for Target Tracking in Wireless Sensor Networks with Uncertainty," *arXiv preprint arXiv:1510.01993,* 2015.

[39] X. Shen and P. K. Varshney, "Sensor selection based on generalized information gain for target tracking in large sensor networks," *IEEE Transactions on Signal Processing,* vol. 62, pp. 363-375, 2014.

[40] D. Artz and Y. Gil, "A survey of trust in computer science and the semantic web," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 5, pp. 58-71, 2007.

[41] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks,* vol. 76, pp. 146-164, 2015.

[42] Y. Ruan and A. Durresi, "A survey of trust management systems for online social communities–Trust modeling, trust inference and attacks," *Knowledge-Based Systems,* 2016.

[43] G. Shafer, *A mathematical theory of evidence* vol. 1: Princeton university press Princeton, 1976.

[44] K. Yang, S. Liu, and J. Shen, "Trust Model Based on DS Evidence Theory in Wireless Sensor Networks," in *China Conference on Wireless Sensor Networks*, 2014, pp. 293-301.

[45] D. Bhatt, S. R. Babu, and H. S. Chudgar, "A novel approach towards utilizing Dempster Shafer fusion theory to enhance WiFi positioning system accuracy," *Pervasive and Mobile Computing,* 2016.

[46] D. AlShamaa, F. Mourad-Chehade, and P. Honeine, "Zoning-based localization in indoor sensor networks using belief functions theory," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2016, pp. 1-5.

[47] L. S. Gallege, D. U. Gamage, J. H. Hill, and R. R. Raje, "Understanding the trust of software-intensive distributed systems," *Concurrency and Computation: Practice and Experience,* vol. 28, pp. 114-143, 2016.

[48] L. Longo, P. Dondio, and S. Barrett, "Temporal Factors to evaluate trustworthiness of virtual identities," in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, 2007, pp. 11-19.

[49] A. Ramanathan, "A Multi-Level Trust Management Scheme for the Internet of Things," 2015.

[50] L. S. Gallege, D. U. Gamage, J. H. Hill, and R. R. Raje, "Trust contract of a service and its role in service selection for distributed software systems," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, 2013, p. 16.

[51] A. Shrestha, X. Liudong, and H. Liu, "Modeling and evaluating the reliability of wireless sensor networks," in *2007 Annual Reliability and Maintainability Symposium*, 2007, pp. 186-191.

[52] E. S. Elmallah and H. S. Hassanein, "On the reliability of wireless sensor networks," in *2006 IEEE International Conference on Communications*, 2006, pp. 3455-3460.

[53] P. R. Pereira, A. Grilo, F. Rocha, M. S. Nunes, A. Casaca, C. Chaudet*, et al.*, "End-to-end reliability in wireless sensor networks: Survey and research challenges," in *EuroFGI Workshop on IP QoS and Traffic Control*, 2007, pp. 67-74.

[54] M. A. Mahmood, W. K. Seah, and I. Welch, "Reliability in wireless sensor networks: A survey and challenges ahead," *Computer Networks,* vol. 79, pp. 166-187, 2015.

[55] V. S. Dhulipala, N. Karthik, and R. Chandrasekaran, "A novel heuristic approach based trust worthy architecture for wireless sensor networks," *Wireless personal communications,* vol. 70, pp. 189-205, 2013.

[56] K. Yuan, F. Xiao, L. Fei, B. Kang, and Y. Deng, "Modeling sensor reliability in fault diagnosis based on evidence theory," *Sensors,* vol. 16, p. 113, 2016.

[57] E. Waltz and J. Llinas, *Multisensor data fusion* vol. 685: Artech house Norwood, MA, 1990.

[58] P. K. Varshney, "Multisensor data fusion," *Electronics & Communication Engineering Journal,* vol. 9, pp. 245-253, 1997.

[59] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion,* vol. 14, pp. 28-44, 2013.

[60] J. K. Aggarwal, *Multisensor fusion for computer vision* vol. 99: Springer Science & Business Media, 2013.

[61] J. A. Castellanos and J. D. Tardos, *Mobile robot localization and map building: A multisensor fusion approach*: Springer Science & Business Media, 2012.

[62] J. Veitch-Michaelis, J. Muller, J. Storey, D. Walton, and M. Foster, "Data Fusion of LIDAR Into a Region Growing Stereo Algorithm," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 40, p. 107, 2015.

[63] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization," *Sensors,* vol. 15, pp. 715-732, 2015.

[64] R. T. Rybarczyk, "e-dts 2.0: A next-generation of a distributed tracking system," Purdue University, M.S. Thesis, 2010.

[65] R. Minutolo and L. A. Annoni, "Indoor localization with multi sensor data fusion in ad hoc mobile scenarios," in *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*, 2014, pp. 403-408.

[66] F. Alshahrany, M. Abbod, J. Alshahrani, and A. Alshahrani, "Intelligent networks data fusion web-based services for ad-hoc integrated WSNs-RFID," 2016.

[67] G. Welch and G. Bishop, "An introduction to the kalman filter. Department of Computer Science, University of North Carolina," ed: Chapel Hill, NC, unpublished manuscript, 2006.

[68] J. Yim, C. Park, J. Joo, and S. Jeong, "Extended Kalman Filter for wireless LAN based indoor positioning," *Decision support systems,* vol. 45, pp. 960-971, 2008.

[69] J. Lategahn, M. Muller, and C. Rohrig, "Tdoa and rss based extended kalman filter for indoor person localization," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, 2013, pp. 1-5.

[70] R. Rybarczyk, R. R. Raje, and M. Tuceryan, "eDOTS 2.0: A Pervasive Indoor Tracking System," in *SEKE*, 2013, pp. 429-434.

[71] R. Rybarczyk, M. Tuceryan, and R. Raje, "e-DOTS: AN INDOOR TRACKING SOLUTION,", IUPUI Research Day , 2012.

[72] R. Garnett, M. A. Osborne, and S. J. Roberts, "Bayesian optimization for sensor set selection," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 209-219.

[73] I. Vlasenko, I. Nikolaidis, and E. Stroulia, "The smart-condo: Optimizing sensor placement for indoor localization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 45, pp. 436-453, 2015.

[74] R. Huang, X. Chu, J. Zhang, and Y. H. Hu, "Energy-efficient monitoring in software defined wireless sensor networks using reinforcement learning: A prototype," *International Journal of Distributed Sensor Networks,* vol. 2015, p. 1, 2015.

[75] M. A. B. Patel and H. B. Shah, "Reinforcement Learning Framework for Energy Efficient Wireless Sensor Networks," 2015.

[76] O. Tilak, S. Mukhopadhyay, M. Tuceryan, and R. Raje, "A novel reinforcement learning framework for sensor subset selection," in *Networking, Sensing and Control (ICNSC), 2010 International Conference on*, 2010, pp. 95-100.

[77] R. Rybarczyk, "Proposal for managing sensor selection through the integration of trust for indoor tracking systems," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, 2015, pp. 239-241.

[78] W. K. Edwards, *Core Jini with Book*: Prentice Hall Professional Technical Reference, 1999.

[79] J. Newmarch, "Jan Newmarchs Guide to Jini Technologies," ed, 2010.

[80] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia,* 2006.

[81] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval,* vol. 2, pp. 1-135, 2008.

[82] J. R. Quinlan, "Induction of decision trees," *Machine learning,* vol. 1, pp. 81-106, 1986.

[83] S. Roberts, "Control chart tests based on geometric moving averages," *Technometrics,* vol. 42, pp. 97-101, 2000.

[84] V. Isler, S. Khanna, J. Spletzer, and C. J. Taylor, "Target tracking with distributed sensors: The focus of attention problem," *Computer Vision and Image Understanding,* vol. 100, pp. 225-247, 2005.

[85] S. Schmidt, "Computational techniques in Kalman filtering," *Theory and applications of kalman filtering,* p. 65, 1970.

[86] Ekahau. (2016). *Ekahau Solutions*. Available: http://www.ekahau.com/, Accessed: 2016.

VITA

VITA

Ryan Thomas Rybarczyk
Department of Computer and Information Science
Indiana University-Purdue University Indianapolis
Website: https://www.cs.iupui.edu/~rrybarcz
Email: rrybarcz@iupui.edu

a. Professional Preparation

- Ph.D. in Computer Science, Purdue University, Indianapolis, IN, 2016.
- M.S. in Computer Science, Purdue University, Indianapolis, IN, 2010.
- B.S. in Computer Science, Butler University, Indianapolis, IN, 2007.

b. Appointments

- CIS Department, IUPUI, Indianapolis, IN: Visiting Scientist, 2016-Present.
- CIS Department, IUPUI, Indianapolis, IN: Research Assistant, 2010-2015.
- CIS Department, IUPUI, Indianapolis, IN: Lecturer, 2010-2015.
- CIS Department, IUPUI, Indianapolis, IN: Department Tutor, 2009-2010.
- Sallie Mae Inc., Fishers, IN: Senior Programmer Analyst, July 2009- August 2009.
- Sallie Mae Inc., Fishers, IN: Programmer Analyst II, 2008-2009.
- Sallie Mae Inc., Fishers, IN: Programmer Analyst I, 2007-2008.
- CSSE Department, Butler University, Indianapolis, IN: Senior Department Tutor, 2006-2007.
- CSSE Department, Butler University, Indianapolis, IN: Teaching Assistant, August 2006 – December 2006.
- CSSE Department, Butler University, Indianapolis, IN: Department Tutor, 2005-2006.

c. Recent Publications

Relevant

i. Rybarczyk R, Rajeev R, Tuceryan M, Infusing Trust in Indoor Tracking, The 10th ACM International Conference on Distributed and Event-Based Systems (DEBS '16). Irvine, CA. 2016.

ii. Rybarczyk R, Proposal for Managing Sensor Selection Through The Integration of Trust for Indoor Tracking Systems. PhD Forum. 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom '15). St. Louis, MO, 2015.

iii. Phadke A, Rybarczyk R, Raje R, Tuceryan M, Incorporating Mobile Devices in Indoor Tracking, International Conference on Network Infrastructure Management Systems (Interface 2014), Mumbai, India, 2014.

iv. Rybarczyk R, Raje R, Tuceryan M, eDOTS 2.0: A Pervasive Indoor Tracking System, Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE'13), Boston, MA, 2013.

v. Gamage D, Rybarczyk R, Raje R, A Practical Approach to Adaptive Service Composition, In: Software Engineering: An International Journal (SEIJ), 2013.

Other Papers

i. Acheson L, Rybarczyk R, Integrating Career Development into Computer Science Undergraduate Curriculum, 2016 11th International Conference on Computer Science & Education. Chikusa-ku, Nagoya, Japan, 2016.

d. Synergistic Activities

i. Participated in the development of courses, titled "Principles of Software Design," "Software Testing," and re-development of course, titled "Object-Oriented Design & Programming."