A HIGH-PRECISION INDOOR TRACKING SYSTEM

by

Ishar Pratap Singh

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Location tracking is of paramount importance to many applications such as healthcare, retail and navigation. Outdoor tracking can be easily implemented using the Global Positioning System (GPS). However, indoor tracking has been a difficult problem to tackle because GPS requires the line of sight to the satellites and therefore it does not work well in indoor environments. In this thesis, a high-precision indoor tracking system is proposed to identify, locate and track a person in an indoor room at a low cost.

The proposed tracking system consists of three components: Stepscan™ tiles, RFID and Kalman-filter based prediction. The Stepscan™ tiles can generate precise location information. However, using Stepscan™ tiles only in an indoor tracking system is too expensive because the manufacturing cost of each Stepscan™ tile is very high. In the proposed system, Stepscan™ tiles are deployed to cover a part of the indoor floor while RFID provides a full coverage. The location information from Stepscan™ tiles and RFID is then used as inputs for our innovative prediction algorithm based on the Kalman filter, which consequently generates high-precision tracking results.

The performance of the proposed system is investigated through extensive simulations. Our simulation results indicate that the proposed system increases the capability to track and locate a person by at least 24% (more than 50% in some cases), with errors ranging from 2.5% to 15%. Furthermore, the proposed system helps to reduce the cost of indoor tracking significantly. In terms of the number of Stepscan™ tiles deployed in the system, a reduction of 7 to 25 tiles can be achieved in the scenarios under investigation. In terms of monetary cost, $21,000 to $75,000 can be saved for an indoor tracking system considered in our research.

LIST OF ABBREVIATIONS USED

GPS	Global Positioning System
RFID	Radio Frequency Identification
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
RF	Radio Frequency
IR	Infra-red
PRAT	Passive Reader Active Tag
ARPT	Active Reader Passive Tag
ARAT	Active Reader Active Tag
HIS	Hospital Information System
PPI	Positive Patient Identification
CM	Constrained Mobility
TIMM	Tactical Indoor Mobility Model
RWP	Random Waypoint
GMM	Gauss Markov Mobility
KF	Kalman Filter
EKF	Extended Kalman Filter
PF	Particle Filter
α-β	Alpha-Beta
DR	Dead Reckoning
DFS	Depth First Search
BFS	Breath First Search
IDD	Iterative Deepening Algorithm
A^*	A-Star
UHF	Ultra High Frequency
PDF	Probability Density Function
DHCP	Dynamic Host Configuration Protocol

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Qiang Ye for his expert guidance, support, valuable feedback and wealth of knowledge which has contributed immensely towards the research work. I am equally thankful to Dr. Srinivas Sampalli for introducing me to ViTRAK and providing me with an opportunity to work towards my goals. Their counsel, support and encouragement helped me immeasurably throughout the research.

I would also like to thank ViTRAK Systems Inc. for their product specific knowledge and their kind co-operation to the finishing point of my project work.

Importantly, I would like thank my wife, parents, sister, brother and all my friends for their constant encouragement and support. I would not have been able to complete this thesis without their unconditional love.

CHAPTER 1 INTRODUCTION

Location tracking is of paramount importance to many applications such as healthcare, retail and navigation. Outdoor tracking can be easily implemented using the Global Positioning System (GPS). However, indoor tracking has been a difficult problem to tackle because GPS requires the line of sight to the satellites and therefore it does not work well in the indoor environments. In this thesis, a high-precision indoor tracking system is proposed to identify, locate and track a person in an indoor room at a low cost. In this chapter, we first presents the motivation of this study. Then we describe the overview of the proposed system and the outline of the thesis.

1.1 MOTIVATION

Location tracking has gained much interest in the field of healthcare, retail, navigation, emergency detection and entertainment industries. To detect the position of an object, a typical navigation system generally comprises either an indoor positioning system or an outdoor positioning system (or both). In general, the technology used for location tracking depends upon the environment (indoors, outdoors, etc.) and the application (patient tracking, asset tracking, security applications, etc.). In large outdoor environments, the Global Positioning System (GPS) is capable of determining the position of the object with excellent accuracy [1]. The system proposed to enable blind people to identify landmarks [2] helps to determine the location of an object using a beep sound in a small-scale outdoor environment. For indoor environments, as the satellite's radio signals weaken, the GPS system often fails to determine the position with desirable accuracy [3].

For the indoor environment, the tracking techniques generally involve sensors such radio-frequency (RF), infrared (IR), ultrasound sensors and Bluetooth [4-6]. Although, these techniques can be used to track indoor targets, each have their own limitations. Techniques involving radio-frequencies generally rely upon received signal strength (RSS) measurements [7] and the fingerprinting methods to estimate the position. In general, as the receiver moves away from the transmitter, there is a gradual loss in the signal strength. The signal strength measurement may be used to determine the location

of the receiver. In an ideal environment, a direct relationship holds between RSS and the distance of the receiver from the emitter. However, in a real environment, severe fading may occur due to noise, multipath characteristics and physical changes. Therefore, the RSS measurements themselves are not enough to estimate the position of the receiver. To overcome the RSS uncertainty (caused by noise and multipath effect), multiple RSS samples could be collected and averaged to obtain more accurate positioning with fingerprinting positioning methods such as weighted k-nearest neighbors method and maximum likelihood. Other localization techniques such as triangulation and angle of arrival (AOA) can also be negatively affected by multipath effect.

RADAR [5] uses RF with fingerprinting to estimate the location of a user. It compares the user's runtime RSSI observations with a set of stored signal strength measurements known as fingerprints at each of the base stations to identify the user's coordinates. It uses weighted k-nearest neighbors method for fingerprinting in order to determine the closest match between the observed RSSI and the stored data. The major disadvantage of RADAR system includes extensive training coverage for fingerprinting and poor extrapolation for areas not covered during training phase.

Indoor tracking systems [8][9] based upon ultrasound or infrared technologies are limited by the line-of-sight requirement. Active Badge [4] is one of the earliest location estimating application that uses infra-red to determine the location of the user. The location is determined by triangulation using periodic IR waveforms. The Active Badge suffered from poor IR scalability and high maintenance drawbacks. The Cricket [6] application uses RF with ultrasonic sensors. The system uses time-of-fight difference between RF and ultrasonic pulses in order to locate a person. The system resulted in improved accuracy and stability, but it requires high maintenance and significant calibrations.

To estimate the indoor location, a number of probabilistic approaches have been studied. *Bayesian inference* [10]*, Particle filter* [11] and *Extended Kalman filter* [12] are a few of them. Bayesian filters for location estimation using ultrasound and infrared have been also been surveyed [13][14]. The Kalman filter and its variants are most efficient in terms of memory and computation whereas particle filter converge to true posterior state for non-Gaussian cases. The Kalman filter has been widely used in the field of robot

tracking and navigation. Nonlinear Particle filters are widely used for real time tracking with good estimation efficiency, but at a high computational cost [14][15].

Most of the existing indoor tracking systems does not have important features such as ease of deployment, low cost, high tracking accuracy and excellent scalability. In our research, we attempt to propose an indoor tracking system that incorporates these features.

1.2 SYSTEM OVERVIEW

The major objective of this research is to propose a high-precision indoor tracking system that is capable of tracking a moving object precisely at a low cost. This objective is achieved by integrating Stepscan^{TM} tiles with RFID and estimating the location using a Kalman-filter based approach.

The Stepscan^{TM} technology from ViTRAK Systems Inc. [16] provides a reliable, secure, scalable, accurate tracking and localization solution. However, the cost of a tracking system based only upon the Stepscan^{TM} tiles is very high. The major objective of this research is to reduce the overall system cost by decreasing the number of tiles deployed in the tracking system without seriously sacrificing the precision. The proposed system not only tracks a person walking on the tiles, which are deployed to cover a part of the indoor floor, but also attempts to predict the current location of the person when he/she is not stepping on the tile.

We tackle the problem of reducing the number of tiles in the tracking system by designing and developing a system that feeds the Stepscan^{TM} tile readings to *Discrete Kalman filter* [17] to estimate the user location. The location estimation results are further improved by filtering the Kalman estimations using the RFID system. The resulting system not only reduces the number of tiles, but also helps to track multiple users at the same time.

The Discrete-Kalman-filter based method used in the proposed system employs a probabilistic approach to estimate the user location. The behavior of the method is dependent upon the availability of the location information from Stepscan^{TM} tiles. In the cases that the tiles can locate the person, the method simply uses the precision information to track him/her. In the scenarios that no location information from the tiles

is available (i.e. the person is not stepping on the tiles), the method attempts to estimate the user location using previous location information. To further improve the estimation results, the zone information from the RFID system is compared with the estimated zone data and, within each RFID zone, the estimated position's RSSI values are examined using the real location's RSSI values. This helps to avoid false estimates and improves the location estimation effectiveness.

The performance of the proposed system is evaluated through extensive simulations. In our simulations, a person walks on an indoor floor with an area of 20x20 ft^2. The floor is partially covered by a few deployed Stepscan^{TM} tiles. In addition, the experimental area is almost fully covered by the RFID system.

1.3 THESIS OUTLINE

The rest of the thesis is organized as follows. Chapter 2 provides the background and related work. It describes various indoor tracking systems, provides an overview of the RFID system and its use in different domains, discusses various filters (*Alpha-Beta filters, Alpha-Beta-Gamma filters, Discrete Kalman filter, Extended Kalman filter* and *Particle filter)*, gives a brief introduction to mobility models and compares varied path finding algorithms that could be used in the simulations. Chapter 3 presents the system design of the proposed system. It includes the details about Discrete Kalman filter, various mobility models, Stepscan^{TM} tiles, RFID system and the integrated system design. Chapter 4 explains the simulation design and provides the detailed experimental results. Chapter 5 includes the conclusion and future work.

CHAPTER 2 RELATED WORK

Indoor tracking has been an active research area over the past years. Many different research groups have been attempting to locate the position of a person within the indoor environments using various concepts and technologies. This chapter describes the work that has been completed in the field of indoor tracking and points out the advantages/disadvantages of the existing systems. In addition, this chapter presents an overview of several important concepts and technologies that are closely related to our simulations and the proposed indoor tracking system.

2.1 INDOOR TRACKING

Indoor tracking has wide range of applications in healthcare, retail and entertainment industry. The main approaches for indoor tracking are categorized as *fingerprinting, trilateration,* and *dead reckoning (DR).* Both *fingerprinting* and *trilateration* are based upon the received signal strength (RSS). The physical position of target is determined based on signal parameters such as *TOA (Time of Arrival), TDOA (Time difference of Arrival), RSS* and *AOA (Angle of Arrival)* using technologies such as *Radio Frequency Identification (RFID), Infrared (IR), Sensor Networks, Ultra-Wideband (UWB), Global Positioning System (GPS), Standard Wi-Fi based Positioning,* etc.

The *fingerprinting* based methods are used to localize the user using wireless technology. Fingerprinting localization approach is a two-step approach: the off-line phase (training) and the on-line phase (localization). In the training phase, the map of the environment is built. The map is based upon the unique parameter (for example: RSS) at specific points within the environment. In the localization phase, the run time RSS values of the user being monitored are recorded and compared to the values stored in the database in the previous training phase. The user location is estimated based upon the best match between the user's current RSS value and training database values [18][19]. Fingerprint based localization delivers satisfactory localization accuracy, but requires extensive pre-configurations (databases creation) and the map is vulnerable to environment changes due to change in room layout. In [20], neural network based model is used to determine the position of a mobile user inside a working area. RADAR [5] uses

WLAN RSS in combination with fingerprinting method and the theoretical propagation model to establish an indoor positioning system.

Trilateration is the process of localization by measuring the distance of the user from at least three stationary observation points with known locations. In 2- dimension, if a point lies on two curves (boundaries of circles), then given the radii and the center of the circles, its location can be estimated with some accuracy. The estimation accuracy is further improved to a single point using one more reading. This is the concept of trilateration. The number of beacon nodes required to accurately estimate the position of the user increases with increase in the indoor area, in turn incurring high infrastructure cost. Radio frequency based range measurements with high accuracy require high sampling rate, this limits the transmission power and the distance of a beacon nodes in order to save power and limit interference. For the same reason ultra sound based measurements are also limited in distance [21].

Dead reckoning is the process of position estimation. The current position estimate is based upon the previously measured position and advancing that position with the estimated velocity in that direction. Typically, a pedestrian dead reckoning system relies on sensors such as accelerometers and magnetometers to estimate the displacement of the object (user) in the time elapsed from a known initial location [21]. Since the estimation of position is based upon the measurements by sensors, which can be noisy, the errors in dead reckoning system can accumulate over time, irrespective of the methods of computing the displacement. For example, the displacement can be computed using double integrating the acceleration measurements [22], or by step detection [23]. To solve the problem of estimation error that accumulates with time, techniques such as Kalman filter [24], zero-velocity-update [25] and spectrum control [26] are used.

In [27], to establish indoor tracking, a foot-mounted inertial sensor module was used to encode the steps of the user, and additionally used sensors such as gyro-meter and magnetometers to measure the orientation of the user. Table 2.1 [28][29][30] summarizes various localization/tracking systems.

Table 2.1 Localization Systems

System Name	Technology	Accuracy	Advantage and Disadvantage
WhereNet [83]	RFID – TDOA	2m to 3m	• Distinctively identify the person and equipment. • High infrastructure cost.
RADAR [5]	WLAN – Triangulation	2.26m	• Uses the existing WLAN infrastructure. • Low level of accuracy and no consideration of privacy.
COMPASS [84]	WLAN – Fingerprint	1.65m	• Handles user orientation. • System design for single user.
Active Badge [4]	Infrared – RSS	Room level	• Address privacy • Low accuracy, long transmission period and influenced by fluorescent light and sunlight.
EKAHAU [85]	WLAN – RSSI	1m	• Low cost and low power consumption. • Provides 2-D localization with low level of accuracy.
Cricket [6]	Ultrasound, RF – TOA and triangulation	10cm	• Address privacy, low cost, decentralized administration. • High power consumption.
Ubisense [86]	UWB - TDOA and AOA	Tens of Centimeters	• Provides 3-D localization with high accuracy without line of sight requirement for a large coverage area. • High system cost.

In comparison to the existing indoor tracking solutions, ours is a high precision indoor tracking solution. Along with reduction in the cost of the tile system, it also, helps to track multiple users with good accuracy using the passive RFID system hence, making

the system scalable. The system design helps to track and estimate the location of user with high precision. The tracking and location estimation effectiveness is measured in units of feet. For most of the estimations, the user's location prediction is within one feet of the actual location.

2.2 RFID OVERVIEW

Radio Frequency Identification (RFID) uses radio frequency for data transfer to automatically identify and track the objects attached to the tags. RFID technology is considered an innovative solution for automatic data collection and asset tracking. RFID has advantage over bar coding identification technology as the RFID tags do not necessarily need to be within the reader line-of-sight and can be enclosed/embedded within the objects being tracked. In healthcare, RFID technology is usually used for identification and tracking.

The RFID system generally includes tags, readers, antennas and the software system. Usually the RFID systems are classified as *Passive Reader Active Tag* (PRAT), *Active Reader Passive Tag* (ARPT), *Active Reader Active Tag* (ARAT) and *Passive Reader Passive Tag* [31]. PRAT consists of passive reader that reads signals only from active tags. ARPT consists of active reader that transmits interrogator signals and receives authentication replies from passive tags. ARAT consists of active reader and active tags. A variation of ARAT uses Battery Assisted Passive (BAP) tags which acts like passive tags but has a small battery to power the tag's return reporting signal. As the name suggests, *Passive Reader Passive Tag system* consists of passive RFID reader and passive RFID tags.

The tags can be passive or active depending upon the power techniques. The passive RFID tags does not have battery power, and can communicate with the reader only when it is within the electromagnetic field of the reader. Whereas, the active RFID tags, using battery power can power the integrated circuits and can broadcast the response signal to the reader. Tags can be read only (the serial number is factory assigned) and read/write (object-specific data can be edited).

Tightly controlled interrogation zones can be created using fixed reader. The zones created by the reader are the reading areas for the tags whenever one comes into

the proximity of the reader zone. Based upon the frequency of operation, RFID technology is regulated (refer Table 2.2 [31] [32][33]).

Table 2.2 RFID Frequency Bands

Frequency Band	Regulation	Range	Data transfer rate	Remarks	Tag cost
120–150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collection	$1
13.56 MHz (HF)	ISM band worldwide	1 m	Low to moderate	Smart cards	$0.50
433 MHz (UHF)	Short Range Devices	1–100 m	Moderate	Defense applications, with active tags	$5
865-868 MHz (Europe) 902-928 MHz (North America) UHF	ISM band	1–2 m	Moderate to high	EAN, various standards	$0.15 (passive tags)
2450-5800 MHz (microwave)	ISM band	1–2 m	High	802.11 WLAN, Bluetooth standards	$25 (active tags)
3.1–10 GHz (microwave)	Ultra wide band	Up to 200 m	High	requires semi-active or active tags	$5 projected
120–150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collection	$1

The RFID reader scans the tags and transmits the information to the back-end database system. The database system filters, analyzes and saves the data. Here-on, the data is passed to enterprise application systems to process it as per the application logic. The database system can collect data from multiple readers located at various sites via wired or wireless networks. In healthcare, RFID systems are usually used in combination with technologies such as sensors and alarms, mobile device and Bluetooth for various

purposes. Active RFID tags are primarily used for the tracking purpose whereas, passive RFID tags are mostly used for patient identification and drug authentication [34].

In healthcare, RFID has been applied in a variety of practices like tracking, identification and verification, sensing, interventions, alerts and triggers, etc. Apart from asset and equipment tracking, RFID technology is also used for patient localization [35] and tracking psychiatric patients [36], children in intensive care unit [37], etc. Further, RFID is used to track and manage the location of patients, as well as manage the waiting-list of patients at medical office [38]. RFID is used to improve the transportation performance of trauma patients [39]. MASCAL [40] tracks patients, staff and equipment in mass causality scenarios using RFID. The Harvard hybrid system [41] uses RFID and bar code for tracking equipment, patient beds and volunteering staff. In comparison to asset tracking, people tracking is more difficult since it involves patients, doctors, medical knowledge, confidentiality and social issues [34].

Along with tracking, RFID can be used for identification purposes. Positive patient identification (PPI) applications uses a smart patient wristband that can be used to identify patient's information such as name, date of birth, insurance information, etc. [42]. RFID has been proposed to identify surgery patients [43]. Aarhus context-aware application [44] identifies the patient lying in the bed. Galway RFID/handheld application [42] identifies patients using wearable RFID wrist bands. Intel transfusion system [45] enhances the blood fusion safety by identifying the patients and staff using secure RFID enabled wristbands and badges that contain encrypted data.

As already mentioned, the tags are classified as passive and active. For the research purpose we will use the passive tags because of its advantages over the active tags, such as longer life, light weight, small size, flexible shape and low tag cost. The reader being used is also passive RFID reader as the indoor area being covered is small.

2.3 PREDICTION FILTERS

It is a common practice to model a practical system with mathematical equations. The general idea of filtering is to form some sort of 'best estimate' for the accurate value of a system, given some observation and potential noise of that system [46]. The system could either be linear or non-linear, depending upon the relationship between the input

values and the output data. The linear system is the system in which the relationship between input and output is linear [47] i.e. if $y_1(t)$ is response for input $x_1(y)$, and $y_2(t)$ is response for input $x_2(y)$, then scaled and summed input produces scaled and summed output i.e. $ax_1(y) + bx_2(y) = ay_1(t) + by_2(t)$. A non-linear system is a system whose output is not directly proportional to its input i.e. a system that does not satisfy the superposition principle. Generally, systems are often modeled as linear since it makes the design and analysis task mathematically tractable. If the system is either inherently linear or the degree of nonlinearity is negligible, the behavior of the system is as expected. Otherwise, there could be significant deviation from expected behavior and the performance of the system could degrade severely. In such cases, it is essential to apply nonlinear methods that properly characterize the system behavior.

In general, the filters can be categorized as time-invariant filters and adaptive filters. The time-invariant filters are filters for which the internal parameters and the structure of the filter are fixed. Time invariance [47] means that output of the system is identical except for a time delay of T seconds, given the input is same for each T seconds delay. For example, if $y(t)$ is the system output due to $x(t)$ input, then the output for input $x(t-T)$ will be $y(t-T)$, where T is the delay in input. Hence, output is independent of the particular time the input is applied. Adaptive filters are required when either the specifications cannot be satisfied by time-invariant filter or the fixed specifications are not known. In order to meet a performance requirement the adaptive filters continually change their parameters [48].

In the theory of stochastic processes [49] (stochastic process or random process is a concept of probability theory and refers to collection of random variables, where random variables are variable whose value changes with each experiment outcome), the filtering problem is a mathematical model for a number of problems in the field of signal processing. To estimate the kinematic components such as position, velocity and acceleration of a moving target, stochastic estimation approach is the predominant paradigm in target tracking [47]. Conventionally, *Kalman filter (KF)* [50] and its derivatives are used for tracking of a stochastic process. *Discrete Kalman filter* is a linear optimal filtering approach. The nonlinear filtering methods being used for target tracking are classified as point based filtering method and the density based filtering method. The

point based method includes *Extended Kalman filter (EKF)* [51][52] and the density based method includes *Particle filter (PF)* [52]. The deterministic approach for tracking target includes *Alpha-Beta filters* [53] and its extensions.

2.3.1 Alpha-Beta (α-β) and Alpha-Beta-Gamma (α-β-ϒ) Filters

Alpha-Beta (α-β) [53][54] filters were designed to minimize the mean square error in estimating position and velocity. The filter assumes that the velocity remains more or less constant over the small time period (the sampling rate). Thus, *α-β* filters have little capacity to track accelerating or maneuvering (changing direction) targets. The filter presumes that the system is sufficiently approximated by a model having two internal states. The two states are position *X* and velocity *V*. Hence, mathematically, the first state is obtained as a result of integration of the second state over time. Assuming the velocity change to be small between data sampling interval *T*, the position state *X* is projected forward to estimate its value at the next sampling time as.

$$X_p^k = X_s^{k-1} + T * V_s^{k-1}$$

As velocity remains approximately constant between data sampling interval, the projected value for velocity state at the next sampling time equals the current value as given by equation:

$$V_p^k = V_p^{k-1}$$

As the filter name suggests, the alpha-beta filter takes selected alpha and beta gains. The filter uses alpha times the deviation (difference between the measured position and estimated position) to correct the position estimate, and uses beta times the deviation (difference between the measured position and estimated position) with a normalizing factor T to correct the velocity estimate as given by equations:

$$X_s^k = X_p^k + \alpha(X_m^k - X_p^k)$$
$$V_s^k = V_s^{k-1} + (\beta/T)(X_m^k - X_p^k)$$

Where,

X_s is the estimated position

X_p is the predicted position

V_s is the estimated velocity

V_p is the predicted velocity

α, β are filter gains

T is the time between the samples (sampling time)

As long as the time between the observations (sampling time) T, or the target acceleration or their combination is small, *alpha-beta* filter will perfectly track a constant-velocity target moving in a straight direction without any errors [54]. For targets moving with constant acceleration, the *alpha-beta* filter will generate a constant prediction error for the position estimations, which is proportional to the acceleration [53]:

$$bx_p^k = \text{X}*\text{T}^2/\beta$$

For constant acceleration, there will also be a steady state constant delay errors for the estimated target position X_s^k and the estimated target velocity V_s^k given respectively by [53]:

$$bx_s^k = \text{-acceleration}*\text{T}^2((1-\alpha)/\beta)$$
$$bv_s^k = \text{-acceleration}*\text{T}((2\alpha-\beta)/2\beta)$$

To account for the constant acceleration, the *Alpha-Beta-Gamma* filter was designed. The filter design was extension of *Alpha-Beta* filter, and presumes that the second state (velocity) can be obtained by integrating the third (acceleration) state, analogous to the way state one (position) and state two (velocity) are related in *Alpha-Beta* filter. An equation is formulated to account for constant acceleration and a multiplier, *gamma* (γ) is selected for applying corrections to the new state estimates. The alpha beta gamma equations [55] are given as:

$$X_s^k = X_p^k + \alpha(X_m^k - X_p^k)$$
$$V_s^k = V_s^{k-1} + (\beta/T)(X_m^k - X_p^k)$$
$$A_s^k = A_s^{k-1} + (\gamma^2/T)(X_m^k - X_p^k))$$

The *Alpha-Beta* filter being a deterministic model, does not account for errors in process and measurement. The filter assumes that the measured position is entirely true without any errors, and the target is moving in one direction without any change in velocity. Even though the *Alpha-Beta-Gamma* filter is designed to handle changes in velocity due to constant acceleration, still it does not accounts for process and measurement errors. As the actual systems usually have process noise and measurement noise associated, we cannot rely on deterministic filters such as *Alpha-Beta* and *Alpha-Beta-Gamma* to provide a reliable and accurate tracking and prediction solution.

2.3.2 Kalman Filter (KF)

Kalman filter is the most widely used state estimator [17]. It is an optimal recursive data processing algorithm [56] that estimates the state of a linear dynamic system from a series of random measurements, generally co-related with noise. They are modeled on a Markov chain, built on linear operators perturbed by Gaussian noise [57].

The *Kalman filter* has been widely used in the application of guidance, navigation and vehicle controls, aircrafts and rockets. It is widely applied concept in the field of signal processing and econometrics [57]. The *Kalman filter* estimates the state of the system that is governed by the linear stochastic difference equation:

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

With measurement update

$$z_k = Hx_k + v_k$$

Where, the known constant matrix are: A is *nxn* state transition matrix, B is *nxm* control matrix, H is *qxn* measurement matrix. The random variables w_k and v_k represent the process noise and measurement noise with covariance Q_k and covariance R_k respectively [12] and are zero-mean $(E(w_k) = E(v_k) = 0)$ white Gaussian noise.

$$P(w) \sim N(0, Q)$$

$$P(v) \sim N(0, R)$$

Also, the initial state x_0, noise vectors $w_{1\ldots k}$ and $v_{1\ldots k}$ at each step are uncorrelated.

The algorithm is a two-step approach [12][57][58] to estimate the state of the system. The first step, known as the prediction step, the *Kalman filter* estimates the current state of the system along with noise. The second state, that is, the measurement update, updates the estimates of prediction step with the measurement data (generally corrupted with some amount of error (noise)). The current estimates are updated using weighted average; higher the certainty, more is the weight associated with the estimate. As the filter is recursive, it incorporates only the current input measurements and the previously estimated state, without the need of the past inputs of estimations.

2.3.3 Discrete Kalman Filter (KF)

As mentioned earlier, the *Kalman filter* is a recursive estimator, that is, the estimation of the current state is dependent only upon the last time step estimated state

and the current measurement. The *Discrete Kalman filter* is generally conceptualized in two distinct phases, *'Time update'* also referred as prediction phase and *'Measurement update'* also known as correction phase. The prediction phase estimates the current state of the system using the previous state of the system. The prediction phase is also known as *a-priori* state as even though the current state estimate is available, it does not include the observation information from the current time. In the measurement update phase the a-*priori* prediction is updated with the measured date (observed information) to refine the state estimation. The improved estimate is known as *posteriori* state estimate. Usually, the two phases work in an alternate manner with the prediction stage advancing the system state until the next observation is made available. However, this is not entirely true, if the observation data is not available, multiple prediction steps may be performed without the measurement update [58][59].

Predict (Time Update):

The time update phase projects the state and covariance estimates from time step k-1 to step k as shown in equations 2.3.1 and 2.3.2

Predicted (a-*priori*) state estimate $$x_{k|k-1} = A_k x_{k-1} + B_k u_{k-1} \quad (2.3.1)$$

Predicted (a-*priori*) covariance estimate $$P_{k|k-1} = A_k P_{k-1} A_k^T + Q_k \quad (2.3.2)$$

Measurement Update (Correction):

The measurement update phase calculates the Kalman gain, estimates the posterior state using the actual measurement and generates the posterior covariance estimate as shown in equations 2.3.3, 2.3.4 and 2.3.5 respectively.

Optimal Kalman gain $$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (2.3.3)$$

Updated (*posteriori*) state estimate $$x_k = x_{k|k-1} + K_k (z_k - H_k x_{k|k-1}) \quad (2.3.4)$$

Updated (*posteriori*) covariance estimate $$P_k = (I - K_k H_k) P_{k|k-1} \quad (2.3.5)$$

After each predict and update phase, the process is repeated with the previous *posteriori* estimates to predict the (current) new *a priori* estimates [12]. It is to be noted that the

filter is both deterministic and stochastic. The state matrices are deterministic part of the filter and the process and measurement noise are stochastic.

The *Kalman filter*, with adjustment to the gain *K* make optimal use of the measurement. Specifically, the gain *K* will give more weight to more accurate measurement. If the measurement data had been missed for a step, then also the Kalman gain adjusts itself optimally. In addition, the filter parameters also adjust to allow for non-equal times between the measurements. The *Discrete Kalman filter*, being linear can easily represent the physical system in mathematical form. Also, it accounts for both process noise and measurement noise of the system. It is an optimal choice for estimating the position of a target in 2-D plane.

2.3.4 Extended Kalman Filter

The *Extended Kalman filter* is an extension of Kalman *filter* for the processes that have a non-linear relationship for measurement and/or estimation. A *Kalman filter* that linearize about the current mean and covariance is known as extended *Kalman filter* (EKF) [12][60][61].

In a non-linear filter, the current state estimate and measurement have a non-linear relationship with previous state estimate and measurement respectively. This relationship is given as:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \qquad \textit{Equation 1}$$

$$z_k = h(x_k) + v_k \qquad \textit{Equation 2}$$

Where, w_k and v_k represent the process and measurement noise with covariance Q_k and R_k respectively as in *Kalman filter*. The non-linear function *f(.)* relates the predicted estimate with the previous step estimate. The non-linear function *h(.)* relates the measurement with the predicted estimate. u_k is the control function. To apply the non-linear functions to the covariance, at each step the Jacobian is computed with current predicted state. This process essentially linearize the non-linear function around the current estimate. The non-linear relationships can be expressed in linear forms as.

$$x_k = \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Ww_{k-1}$$

$$z_k = \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k$$

where,

x_k and z_k are the actual state and measurement vectors,

$\tilde{x}_k$ and $\tilde{z}_k$ are the approximate state and measurement vectors.

$\hat{x}_k$ is an a posteriori estimate of the state at step k.

w_k and v_k represent the process noise and measurement noise.

A, W, H, V are the Jacobian matrices defined as:

$$A_{[i,j]} = \partial f_{[i]}/\partial x_{[j]}(\hat{x}_{k-1}, u_{k-1}, 0)$$

$$W_{[i,j]} = \partial f_{[i]}/\partial w_{[j]}(\hat{x}_{k-1}, u_{k-1}, 0)$$

$$H_{[i,j]} = \partial h_{[i]}/\partial x_{[j]}(\hat{x}_k, 0)$$

$$V_{[i,j]} = \partial h_{[i]}/\partial v_{[j]}(\hat{x}_k, 0)$$

The state estimation equation can be simplified as

$$\hat{x}_k = \tilde{x}_k + K_k \tilde{e}_{z_k}$$

where

$\tilde{e}_{z_k} = z_k - \tilde{z}_k$ is the measurement residual.

The predict and measurement updates for *Extended Kalman filter* (EKF) are defined as:

Predict:

As with *Kalman filter*, The EKF predict stage projects the state and covariance estimates from time step k-1 to step k as shown in equations 2.3.6 and 2.3.7

Predicted state estimate $$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_{k-1}) \quad (2.3.6)$$

Predicted covariance estimate $$P_{k|k-1} = A_{k-1} P_{k-1} A_{k-1}^T + Q_{k-1} \quad (2.3.7)$$

Measurement Update:

As with Kalman filter, The EKF measurement update stage calculates the Kalman gain, estimates the *posterior* state using the actual measurement and generates the *posterior* covariance estimate as shown in equations 2.3.8, 2.3.9 and 2.3.10 respectively.

Near-optimal Kalman gain $$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (2.3.8)$$

Updated state estimate $$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1})) \quad (2.3.9)$$

Updated estimate covariance $$P_k = (I - K_k H_k) P_{k|k-1} \quad (2.3.10)$$

2.3.5 Particle Filter

As with EFK, *Particle filter* is also a non-linear filter used for state estimation. It is known by several names for example, sequential importance sampling (SIS), bootstrap filtering, Monte Carlo filtering, sequential Monte Carlo filtering, etc. *Particle filter* foundation is based upon the Bayesian approach to estimation [11][62]. Similar to *equation 1* and *equation 2*, suppose the non-linear system is defined by equations.

$$x_{k+1} = f_k(x_k, w_k) \qquad Equation\ 3$$

$$y_k = h_k(x_k, v_k) \qquad Equation\ 4$$

Where, *k* is the time sequence index, x_k is the state to be estimated and y_k is the received measurement, w_k is the process noise and v_k is the measurement noise. The functions $f_k(x_k, w_k)$ and $h_k(x_k, v_k)$ are time-varying, non-linear state estimation and measurement equations. Both w_k and v_k are uncorrelated and white. It is to be noted that *equation 3* is defined as first order Markov process [62], and an equivalent probabilistic description of the state is $(x_{k+1}|x_k)$, which is known as transition density [62]. Similarly for *equation 4*, the equivalent probabilistic description of the state is $p(y_k|x_k)$.

The objective of Bayesian approach is to estimate the posterior probability function (pdf) of the state $'x_k'$ based upon measurements $y_1, y_2, y_3, \ldots, y_n$. The *posterior* pdf is denoted as $p(x_k|Y_k)$, where $Y_k = y_i, i = 1, \ldots, k$. The Bayesian recursive filter consists of prediction and update operations [62]. The prediction operation propagates the *posterior* pdf of the state from time step k forwards to time step *k + 1*. Suppose that $p(x_k|Y_k)$ is available, then $p(x_{k+1}|Y_k)$, the *prior* pdf of the state is obtained by the use of the dynamics model.

$$\underbrace{p(x_{k+1}|Y_k)}_{Prior\ at\ current\ step} = \int p(x_{k+1}|x_k) \underbrace{p(x_k|Y_k)}_{Posterior\ from\ previous\ step} dx_k \qquad Equation\ 5$$

The *prior* pdf is updated to incorporate the new measurements y_k to give the required *posterior* pdf as per the Bayes rule.

$$p(x_{k+1}|Y_{k+1}) = p(y_{k+1}|x_{k+1})p(x_{k+1}|Y_k)/p(y_{k+1}|Y_k) \qquad Equation\ 6$$

Where $p(x_{k+1}|Y_{k+1})$ is the *posterior* pdf (corrected estimate after measurement)

$p(y_{k+1}|x_{k+1})$ is the likelihood pdf

$p(x_{k+1}|Y_k)$ is the *prior* (predicted estimate)

$p(y_{k+1}|Y_k$ is the normalization factor, given by $p(y_{k+1}|Y_k) = \int p(y_{k+1}|x_{k+1})p(x_{k+1}|Y_k)dx_{k+1}$

Equation 5 and *equation 6* define the Bayesian recursive filter with initial condition given by the specified *prior* pdf $p(x_0|Y_0)$ with no measurement data. In particular, if '*f(.)*' and 'h(.)' are linear functions and, $'w_k'$ and $'v_k'$ are additive, independent and Gaussian, then the solution is the Kalman Filter.

The *Particle filter* was invented to numerically implement the Bayesian estimator [11].

1. Generate N state vectors (also called particle or sample), based upon the initial pdf $p(x_0)$, which is assumed to be known. The particles are denoted as $x_{0,i}^+ \; i = 1,2,\ldots.,N$, where i is the particle index. The parameter N is selected as a tradeoff between computational effort and estimation accuracy.

2. For each time step k = 1,2,3,...N, perform the following:

a. Propagate the particles to the next step to obtain a *priori* estimation $x_{k,i}^-$ of the N particles using the process function $x_{k,i}^- = f_{k-1}(x_{k-1}^+, w_{k-1}^i)$ where each noise vector is randomly generated on the basis of the known pdf of w_{k-1}^i

b. With the measurement data, compute the relatively likelihood for each particle $x_{k,i}^-$. To do this, evaluate pdf $p\left(y_k \middle| x_{k,i}^-\right)$, on the basis of the measurement equation (*equation 4*).

c. Scale the relative likelihood, such that the sum of all the likelihoods is equal to one. $q_i = q_i \,/\, \sum_{j=1}^{N} q_j$

d. Based upon the relative likelihood, generate the set of *posterior* particles, $x_{k,i}^+$.

e. Resample (with replacement) the N particles, based upon normalized weights to produce a new set of particles. The re-sampling effectively weights each *prior* particle of the state with respect to the current measurement. Hence, the re-sampling operation is biased towards the more probable *prior* samples. As a result, more heavily weighted samples may well be chosen frequently.

This results in a set of particles that are distributed according to their pdf's $p(x_k|Y_k)$.

As the physical system can easily be defined using linear mathematical equations, i.e. $p_t = p_{t-1} + v_{t-1}t + \frac{1}{2}a_t t^2$ and $v_t = v_{t-1} + a_t t$, where p_t is the position, v_t is the velocity, a_t is the acceleration and t is the time step. We prefer to use *Discrete Kalman Filter* in 2-D for the purpose of tracking and prediction. To model for *Extended Kalman Filter* or *Particle Filter*, the system needs to be defined by non-linear equations. To do so, we need the displacement in position and angle of orientation (polar coordinates) thus, making the system complex and difficult to estimate. As the Cartesian coordinates are easily attainable using the ViTRAK's Stepscan™ tiles, we can easily model the physical system using liner mathematical equations defined above.

2.4 MOBILITY MODEL OVERVIEW

Mobility models are used to represent the movement of mobile users, and how their velocity and acceleration changes with time and location. Such mobility models are often used for simulation purpose to investigate new communication or navigation techniques [63]. In recent years, several mobility models have been proposed and are used to evaluate the performance of both indoors and outdoors networks [64].

The approaches addressing the indoor movement are *Constrained Mobility Model (CM)* [65] and *Tactical Indoor Mobility Model (TIMM)* [66]. In CM, the nodes move along edges of a graph representing valid paths inside the building and vertices represent the possible destinations. The movement is accompanied by traversing the edges which constitutes shortest path, resulting in movements through hallways and doors [65]. The TIMM model uses graph based approach similar to CM. The model represents the building as a graph, where each room and door or passageway corresponds to a vertex. Edges are formed by connecting the vertices with a straight line [66]. The 3-D mobility model proposed by Kim, assumes boundary conditions and vertical motion through elevators for indoor mobility model [67]. These models do not account for movement of a person within a room or a hall and are constrained by the floor plan, passageway and doors. Therefore, we cannot use these models to simulate a person walking in a room/hall.

In random based mobility models the nodes move around randomly, more specifically, the destination, speed and direction are all chosen at random [68]. Models

like *Random-Waypoint-Mobility-Model* (RWP) and *Gauss-Markov-Mobility-Model* (GMM) are mainly used random mobility model [66], but RWP model because of its simplicity and ease of use is more frequently incorporated for simulations. In RWP model, each node randomly selects a waypoint and moves towards that waypoint (Waypoint is a reference point in physical space [69]) with a constant speed chosen randomly from (0; Vmaximum), where Vmaximum is the maximum allowable speed for the node. Once the mobile node reaches that waypoint, it becomes stationary for a predefined pause time. After the pause interval, it selects another random waypoint within the simulation region (predefined area) and moves towards it. The whole process is continuously repeated until the end of simulation [70].

Biologists have studied that animals including birds and sharks abandon Brownian motion, the random motion seen in swirling gas molecules [71], for Levy flights in search of food. A flight is defined to be a longest straight line trip from one location to another that a particle makes without a directional change or pause time [72]. Such a mobility model is known as Levy Walks. Injong [73], showed that human walk in outdoor settings contain statistically similar features as Levy walks including heavy-tail flight and pause-time distributions. Still, Levy walks cannot be used for the simulation purpose as our simulation scenario is confined to a 20x20 ft^2 indoor area.

For the simulation of person walking on a floor of size 20x20 ft^2, we modify the *Random-Waypoint-Mobility-Model* to generate four different simulation models. To the best of our knowledge, there is no existing realistic mobility model for a person walking on a small size area.

2.5 PATHFINDING ALGORITHMS

Pathfinding refers to plotting of the shortest route between two points with the help of computer application. Generally, the pathfinding algorithm is used with intent to find the shortest path or route. The route is generated by searching the graph for a node and exploring the neighboring nodes until the destination node is reached. The primary purpose of using pathfinding is to find a path between two nodes and also to generate optimal shortest path in the graph [74]. Several algorithms have been proposed and written to answer the problem of pathfinding. For example *Depth First Search algorithm,*

Breadth First Search algorithm, Iterative Deepening algorithm, Best First Search Algorithm, Dijkstra's Algorithm and A-Star(A) Algorithm.*

The *Depth First Search* (DFS) algorithm progresses the search tree by expanding the first child node and diving deeper and deeper until a destination node is found, or until no node remains to be traversed. Thereafter, it backtracks, returning to the most recent node which has not been explored completely [75]. Eventually the algorithm transverses the entire graph. Similar to DFS, the *Breadth First Search algorithm* is an exhaustive search algorithm. Breadth First Search algorithm iterates the graph in levels. Starting at a given vertex considered to be at level 0, the system visits all the vertices at level 1. Once level 1 is searched, vertices of level 2 are iterated. The level 2 vertices are adjacent to level 1 vertex, and so on. The search terminates once all the vertices at all the levels are transverse. The BFS algorithm returns the shortest path, but with a poor time performance as it needs a lot of memory for queuing [76]. Both DFS and BFS [74][76][77][78] can be used to determine a path between two nodes using exhaustive search, but may not be able to determine the shortest optimal path.

*Iterative Deepening algorithm (*IDD*)* repeatedly runs the depth-limited search, increasing the search depth with each iteration until the complete depth of the graph is reached. IDD is similar to breath first search algorithm, but with much less memory utilization, thus making it more responsive [14].

Although *Breadth First Search algorithm* and *Iterative Deepening algorithm* finds the path given enough time, other algorithms tends to generate a shortest optimal path with less processing time. In general, a person walks in the direction of the destination and deviate at a minimum, only to avoid obstacles [12]. Hence, we avoid using the *Breadth First Search, Depth First Search* and *Iterative Deepening algorithms* for the simulation purpose to determine the shortest path.

Best first search algorithm uses heuristics to rank the nodes based on the estimated cost from current node to the goal node. "Heuristics" can be defined as a useful guide for problem solving. It refers to a set of problem-solving rules that do not guarantee a solution. The algorithm maintains an OPEN list and a CLOSED list. The OPEN list maintains a list of candidate nodes yet to be visited and the CLOSED list maintains a list of visited candidate nodes. The OPEN list maintains all unvisited successor nodes for

each visited node i.e., the algorithm is not restricted to only neighbors, but also selects the best of unvisited nodes. It is this property which distinguished *best first search algorithm* from *depth first algorithm* and *breadth first algorithm* [79][80]. The *best first algorithm* search may provide a fast solution, but it may not be an optimized solution because the heuristics function might not be very accurate.

Dijkstra's Algorithm and A-Star(A) Algorithm* are classical algorithms for path finding and many variations of each have been proposed [81]. *Dijkstra's Algorithm*, for a given (starting) vertex, can be used to find the shortest path from the starting vertex to all other vertices in the graph or from starting vertex to the destination vertex. The algorithm is similar to *Breadth first search* algorithm, but assigns weights to the edges. Whereas, in BFS each edge is assumed to have a standard weight of 1. The basic process of *Dijkstra's Algorithm* is to assign each node (vertex) a distance value. Initially the value is zero for initial node and infinity for rest of the nodes [82].

In *Dijkstra's Algorithm*, the current node's neighbors are examined and their distance '*D*' from the starting node to current node is calculated i.e. distance '*D*' is equal to sum of distance of current node to the initial node and distance of current node to the neighbor node. The newly calculated distance value replaces previously calculated distance value for that node if the new distance '*D*' is less than the previously calculated distance value. Once all the neighbors of the current node are examined, the current node is marked as visited and will not be examined again. The neighbor node with the new lowest distance value is marked as the new current node and the process repeats until the target is marked as visited, or all nodes are marked as visited without the target being found. Once the target is marked as visited, path is traced from the destination node to the starting node [82].

A good variation of *Dijkstra's Algorithm* is *A-Star(A*) Algorithm*. *A* Algorithm* assigns weights to each node which is equal to the sum of weight of the edge to current node and the approximate distance between current node and the target node. The approximate distance is establish by heuristics and represents a minimum possible distance between the current node and the target node. *A* Algorithm* improves upon the behavior of Dijkstra's algorithm using the heuristic values [74].

For each node being examined, *A* Algorithm* maintains three values: *f(x), g(x),* and *h(x). f(x)* is the sum of *g(x)* and *h(x),* where *g(x)* is the distance from the initial node to the node currently being examined through all the previous nodes traversed to get to that point and *h(x)* is the heuristic distance from the current node to the target node. The heuristic can be calculated in several ways, and for the purpose of simulation we used Manhattan distance. Manhattan distance is simple to calculate and is described as the distance between the two points along the *x*-axis plus the distance between the two points along the *y*-axis. To calculate Manhattan distance, subtract the target node's *x*-value from current node's *x*-value and take its absolute value. Add this to the absolute value of the difference between target node's *y*-value and current node's *y*-value. Calculation of heuristics using Manhattan distance is based upon the assumption that mostly the distance from the current node to the target node deviates from a straight line as it passes through more than one node. Therefore, a horizontal displacement plus vertical displacement formula is likely to result in value closer to the actual distance that will be traveled, than an estimate using plain Euclidean distance [82].

A Algorithm* also maintains an OPEN list and a CLOSED list, which is a list of all unvisited nodes and list of all visited nodes respectively. At the start of search, all nodes are in the OPEN list with starting node marked as current node. The values of *g(x), h(x),* and *f(x)* are calculated for each of its neighbors. If the latest *f(x)* value of a node being examined is less than the earlier *f(x)* value for that node, the new f-value replaces the old f-value. There onwards, the current node is moved from the OPEN list to the CLOSED list and the neighbor node with the lowest *f(x)* value is marked as the new current node. The search process repeats until the target node is added to the CLOSED list, or there are no more nodes on the OPEN list [82].

As already mentioned, *A* Algorithm* is equivalent to *Dijkstra's Algorithm* if the heuristics for *A* Algorithm* evaluates to zero. The heuristic value plays an important role in *A** algorithm. As the heuristics approximation gets closer to the true distance, *A* Algorithm* examines fewer nodes and resulting in faster generation of optimal paths. *A* Algorithm* examines fewest nodes (hence fastest) when the heuristic approximation is exactly equal to the true distance. However, generally it is not possible to compute the true distance using heuristics function [44]. *A* Algorithm* generates pathways which are

not smooth. As the simulation graph is 20x20 matrix (for calculating the path using pathfinding), we rely on heuristics and calculate the path using *A* Algorithm*. Another drawback with A* algorithm is the memory requirement as the algorithm scans and builds the entire path in the open list before the path is actually traced by the user, as the A* algorithm is being used for non-linear path generation for the simulation models, this drawback is of no significance in the system design. To smooth the path returned by A* *Algorithm*, gradient descent algorithm is applied over the path generated.

CHAPTER 3 SYSTEM DESIGN

In this chapter, we present the proposed high-precision indoor tracking system. As mentioned previously, the Stepscan™ tiles [16] can be used to locate a person in an indoor environment with excellent accuracy and reliability. However, the cost of a system consisting of only Stepscan™ tiles is very high. The proposed system is designed to lower the cost by integrating passive RFID technology with Stepscan™ tiles and filtering the collected data using Discrete Kalman Filter [87]. With the proposed system, we are able to reduce the number of tiles required to track the person without sacrificing the tracking precision significantly.

3.1 PRELIMINARIES

3.1.1 Discrete Kalman Filter

The Discrete Kalman filter can estimate the position of point in 2-dimensional plane. To estimate the position of a point in a plane, we first need to define the physical motion of the point and describe the same in mathematical form. In this section, we will explain the physics of motion, white noise, Kalman filter's discrete time model and error covariance matrices.

Physics of motion

Assume $p(t)$ be the position of target that varies with time *t*. Derivate to yield Taylor series.

$$p(t) = p\,(t_n) + \Delta t\dot{p}(t_n) + \frac{\Delta t^2}{2!}\ddot{p}(t_n) + \frac{\Delta t^3}{3!}\dddot{p}(t_n) + \ldots\ldots \quad (3.1.1)$$

where, $p(t)$ is the position, $\dot{p}$ is the velocity and $\ddot{p}$ is the acceleration of target in motion. The above Taylor series expansion hold true for small value of t (i.e. Δt). If acceleration is constant, the third derivate of the Taylor series reduces to zero and the series is reduced to:

$$p(t_{n+1}) = p\,(t_n) + \Delta t\dot{p}(t_n) + \frac{\Delta t^2}{2!}\ddot{p}(t_n) \quad (3.1.2)$$

This holds true for very small variations in acceleration and for small time intervals Δt.

If velocity is constant, the second derivate of the Taylor series reduces to zero and resulting series is as follows:

$$p(t_{n+1}) = p\,(t_n) + \Delta t\dot{p}(t_n) \quad (3.1.3)$$

As a result of equation F= m.a, where *a* is the acceleration, equation 3.1.2 suggests that the force *F* resulting in the movement of the target of mass *m*, is constant. This holds true for motion of technical systems, but for humans, the force keeps on changing. The motion of equations 3.1.2 and 3.1.3 holds true only for constant acceleration and constant velocity motion.

Humans generally don't walk with constant acceleration or constant velocity. Their motion under consideration is associated with white noise. So, human motion for constant acceleration becomes:

$$p(t_{n+1}) = p\,(t_n) + \Delta t\dot{p}(t_n) + \frac{\Delta t^2}{2!}\ddot{p}(t_n) + white\ noise$$

And motion with constant velocity becomes:

$$p(t_{n+1}) = p\,(t_n) + \Delta t\dot{p}(t_n) + white\ noise$$

For motion with constant acceleration, the acceleration component of the motion is the source of white noise, (i.e. the deviation from constant acceleration). For motion with constant velocity, a variation in velocity (i.e. acceleration) is the source of white noise.

The below equations describe the relation between velocity at time *t+1* and time *t* for constant velocity and constant acceleration:

$$v(t_{n+1}) = v\,(t_n) \quad (3.1.4)$$

$$v(t_{n+1}) = v\,(t_n) + \Delta t a(t_n) \quad (3.1.5)$$

White noise

A random vector is said to be a white noise vector if each of its components are statically independent (i.e. have zero mean probability distribution and finite variance). Two variable are statically independent if they are statically uncorrelated (i.e. have zero covariance [94]).

The white noise vector's *'w',* covariance matrix with *n* elements is *nxn* diagonal matrix, where each diagonal element represents the *variance*. In addition, if each element of *w* has zero mean normal distribution and same variance(σ^2), then *w* is known as Gaussian white noise vector having multivariate normal distribution.

The power spectrum P of a random vector w can be defined as $Pi = E(|W_i|^2)$, where, W, is Fourier transform coefficients. With this definition, Gaussian white noise vector will have a perfectly flat power spectrum (i.e. $P_i = \sigma^2$ for all i).

Kalman filter discrete time model

As mentioned in Section 2.3.2, the *Kalman filter* estimates the state of the system that is governed by the linear stochastic difference equation:

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

If no input vector is available, the state of the system can be expressed as:

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

And the measurement update as:

$$z_k = Hx_k + v_k$$

To estimate the location of a particle in 2-dimension, the vector $x_k = [p_k, v_k]^T$, known as *process state vector* consists of elements representing position p_k and velocity v_k. Matrix A relates the previous state estimate to the current state estimate x_k, i.e. how the current position at time t is related to previous position and velocity at time t - 1.

The state matrix A describes the motion of object as per the physical model equations 3.1.2 and 3.1.3. For 1-dimention motion, the matrix A is written as

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

The process noise vector w_k associated with the random process, as per [87] is defined as:

$$\begin{bmatrix} w_{p,k} \\ w_{v,k} \end{bmatrix} = \begin{bmatrix} \int_{t_k}^{t_{k+1}} (t_{k+1} - \tau) X_a(\tau) d\tau \\ \int_{t_k}^{t_{k+1}} X_a(\tau) d\tau \end{bmatrix}$$

where $w_{p,k}$ is the process error related to the position, $w_{v,k}$ is the process error related to the velocity. τ is the time difference 't_2 -t_1'. $X_a(t)$ is a random variable describing the acceleration (having the characteristics of white noise).

As $X_a(t)$ is a random variable, the integrals $w_{p,k}$ and $w_{v,k}$ cannot be determine. Only, moments like expected value and variance can be determined from the random variable $X_a(t)$.

Assuming $w_k = 0$, the equation $x_k = Ax_{k-1} + w_{k-1}$ reduces to $x_k = Ax_{k-1}$. This part of equation is the deterministic part. Using the above information, this equation can be rewritten as.

$$\begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ v_{k-1} \end{bmatrix} \tag{3.1.6}$$

Equation 3.1.6 is the detailed deterministic equation, representing equation of motion with constant velocity in 1-dimension. The first component of x_k (i.e. p_k) is the result of $p_{k-1} + \Delta t\, v_{k-1}$ and the second component of x_k (i.e. v_k) is the result of $v_k = v_{k-1}$. Solving the above equation using matrix calculations yields equations similar to equations 3.1.3 and 3.1.4.

For motion with constant acceleration, the equation $x_k = Ax_{k-1} + Bu_k + w_{k-1}$ can be rewritten as:

$$x_k = Ax_{k-1} + Ga_k + w_{k-1} \tag{3.1.7}$$

Where Ga_k is the control due to acceleration (a_k) in motion. For motion of particle in 1-dimension, the position is related to acceleration as $\frac{\Delta t^2}{2!}$ and velocity is related to acceleration as Δt. Hence, matrix G can be defined as

$$G = \begin{bmatrix} \frac{\Delta t^2}{2!} \\ \Delta t \end{bmatrix}$$

Assuming $w_k = 0$, equation 3.1.7 reduces to $x_k = Ax_{k-1} + Ga_k$, which is deterministic. Substituting the values, we obtain:

$$\begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} a_{k-1} \tag{3.1.8}$$

The equation 3.1.8 represents the motion of particle with constant acceleration in 1-dimension. Here, p_k is the result of $p_{k-1} + \Delta t\, v_{k-1} + \frac{\Delta t^2}{2!} a_k$ and v_k is the result of $v_k = v_{k-1} + \Delta t a_k$. Equation 3.1.8 can be further solved using matrix calculations and will result in equations similar to equations 3.1.2 and 3.1.5.

The second part of Kalman filter deals with the measurement data according to the linear equation:

$$z_k = Hx_k + v_k$$

This equation relates the measurement vector z_k with state vector x_k using the measurement matrix *H*. For motion in 1-dimension or motion in 2-dimension, there is no need to measure the velocity of motion as that can be calculated by Kalman filter internally. To estimate the position and velocity using the state vector x_k, only the measurements of position is enough. As we measure only the position, the state matrix *H* links the position measurements with the state matrix 'x_k' position estimates. This results in a non square matrix *H*. In 1-dimension, the matrix *H* can be written as:

$$H_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

This results in the following measurement equation with zero noise:

$$p_{zk} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} \tag{3.1.9}$$

The above section explains the Kalman matrices and vector values for motion in 1-dimension. Modeling of motion in 2-dimension is just an extension of motion in 1-dimension. In 2-dimensional motion, the position and velocity are both a function *of x-coordinates* and *y-coordinates.* As the position is measured in Cartesian coordinates, it can be referred with notation (p_x, p_y). Here, p_x is the position in *x-coordinates* and p_y is the position in *y-coordinates*. Similarly the velocity can be represented in *x-y plane* as (v_x, v_y).

The process state vector defines 2-dimensional position and velocity. Using $p_{k,x}$, $p_{k,y}$, $v_{k,x}$, $v_{k,y}$, equation 3.1.6 can be written as:

$$\begin{bmatrix} p_{k,x} \\ p_{k,y} \\ v_{k,x} \\ v_{k,y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1,x} \\ p_{k-1,y} \\ v_{k-1,x} \\ v_{k-1,y} \end{bmatrix} \tag{3.1.10}$$

Where,

$$\begin{bmatrix} p_{k,x} \\ p_{k,y} \\ v_{k,x} \\ v_{k,y} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} p_{k,x} \\ p_{k,y} \end{bmatrix} \\ \begin{bmatrix} v_{k,x} \\ v_{k,y} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} p_k \\ v_k \end{bmatrix} = x_k$$

$$\text{And } \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} p_{k,x} \\ p_{k,y} \end{bmatrix} \\ \begin{bmatrix} v_{k,x} \\ v_{k,y} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} p_{k,x} + \Delta t v_{k,x} \\ p_{k,y} + \Delta t v_{k,y} \end{bmatrix} \\ \begin{bmatrix} v_{k,x} \\ v_{k,y} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k,x} \\ p_{k,y} \\ v_{k,x} \\ v_{k,y} \end{bmatrix}$$

Similarly for equation of motion with constant acceleration, the equation 3.1.8 can be written as:

$$\begin{bmatrix} p_{k,x} \\ p_{k,y} \\ v_{k,x} \\ v_{k,y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1,x} \\ p_{k-1,y} \\ v_{k-1,x} \\ v_{k-1,y} \end{bmatrix} + \begin{bmatrix} (\Delta t/2)^2_{k-1,x} \\ (\Delta t/2)^2_{k-1,y} \\ \Delta t_{k-1,x} \\ \Delta t_{k-1,y} \end{bmatrix} a_{k-1} \tag{3.1.11}$$

Where,

$$\begin{bmatrix} (\Delta t/2)^2_{k,x} \\ (\Delta t/2)^2_{k,y} \\ \Delta t_{k,x} \\ \Delta t_{k,y} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} (\Delta t/2)^2_{k,x} \\ (\Delta t/2)^2_{k,y} \end{bmatrix} \\ \begin{bmatrix} \Delta t_{k,x} \\ \Delta t_{k,y} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} (\Delta t/2)^2_{k} \\ \Delta t_{k} \end{bmatrix} = G_k$$

Furthermore, the measurement equation 3.1.9 can be written as:

$$\begin{bmatrix} p_{k,x} \\ p_{k,y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{k-1,x} \\ p_{k-1,y} \\ v_{k-1,x} \\ v_{k-1,y} \end{bmatrix} \tag{3.1.12}$$

Solving the above equations will result in position measurement in *x* and *y coordinates.* As per [87], in equations 3.1.10, 3.1.11 and 3.1.12, the components of *x*-coordinates are independent of the components of y-coordinates. Also the process error vector w_k contains elements $[w_{p,k,x}, w_{p,k,y}, w_{v,k,x}, w_{v,k,y}]^T$.

Error covariance matrices

Being a stochastic model, Kalman filter is able to handle the noise associated with the physical model. As mentioned earlier, the different types of white noise associated with Kalman filter are process noise and measurement noise.

$$E[w_i w_j^T] = \begin{cases} Q_k & if\ i == j \\ 0 & if\ i \neq j \end{cases}$$

$$E[e_i e_j^T] = \begin{cases} R_k & if\ i == j \\ 0 & if\ i \neq j \end{cases}$$

$$E[w_i e_j^T] = 0$$

As the error is induced in the system due to acceleration and as $X_a(t)$ has zero correlation for two different times, $E[w_i w_j^T] = 0 \; \forall \; i \neq j$. Furthermore, the acceleration at time t_k is not influenced by acceleration at other times. The measurement error $E\left[e_i e_j^T\right] = 0 \; \forall \; i \neq j$.

The covariance matrices handling the noise are Q_k and R_k. Both P_k and state vectors are updated during the recursion process, but the process error Q_k and the measurement error R_k are constant. As the values for process error and measurement error are not changed/updated by Kalman filter, it becomes essentially important to correctly determine the noise in the system before the execution of the Kalman recursion.

Measurement error covariance matrix

The measurement error covariance matrix is denoted by R_k. It accounts for the errors in the measurement. For 2-dimensional system, we are measuring the position in *x-y* plane in terms of *x,y* coordinates. The covariance matrix R_k accounts for the errors in measuring the position in *x-y* plane.

Let $X_{e,x}(t)$ and $X_{e,y}(t)$ denote the random variables describing the measurement error such that $e_{k,x} = X_{e,x}(t)$ and $e_{k,y} = X_{e,y}(t)$. The covariance matrix R_k is described as:

$$R_k = E[\left(X_{e,x}(t_k), X_{e,y}(t_k)\right)^T \left(X_{e,x}(t_k), X_{e,y}(t_k)\right)]$$

$$R_k = \begin{bmatrix} E[X_{e,x}(t_k), X_{e,x}(t_k)] & E[X_{e,x}(t_k), X_{e,y}(t_k)] \\ E[X_{e,y}(t_k), X_{e,x}(t_k)] & E[X_{e,y}(t_k), X_{e,y}(t_k)] \end{bmatrix} \quad (3.1.13)$$

As the measurement in x-coordinate is independent of measurement in y-coordinate, the random variables $X_{e,x}(t)$ and $X_{e,y}(t)$ have zero-mean and are uncorrelated. The integration of $X_{e,x}(t_k), X_{e,y}(t_k)$ over some time interval will result in zero value [87]. Therefore, the expression $E[X_{e,x}(t_k), X_{e,y}(t_k)]$ and $E[X_{e,y}(t_k), X_{e,x}(t_k)]$ will become zero. The expressions $E[X_{e,x}(t_k), X_{e,x}(t_k)]$ and $E[X_{e,y}(t_k), X_{e,y}(t_k)]$ are the squares of standard deviation (variance) of the random variables $X_{e,x}(t)$ and $X_{e,y}(t)$ and are known as variances. Hence, expression 3.1.13 becomes:

$$R_k = \begin{bmatrix} E[X_{e,x}(t_k), X_{e,x}(t_k)] & 0 \\ 0 & E[X_{e,y}(t_k), X_{e,y}(t_k)] \end{bmatrix} \quad (3.1.14)$$

Measurement errors are always associated with a system [87]. The measurement errors are generally calculated or known and sometimes estimated. For the model, we assume that the tiles generate the *x,y* coordinate measurements with ± 0.5 feet error with equal probability. This deviation in measurement is a result of the human feet size (assuming the biggest feet size to be 1 foot and hence, the maximum measurement error). We consider the *x,y*-coordinate location to be the mean of the human feet, but as there are several *x,y*-coordinates locations related to each step, we calculate the error in measurement with feet's mean *x,y* measurement and 0.5 ft deviation in each direction. Hence, resulting variance is:

$E[X_{e,x}(t_k), X_{e,x}(t_k)] = \frac{((-0.5)^2 + 0^2 + 0.5^2)}{3} = .167$ giving the measurement covariance matrix as:

$$R_k = \begin{bmatrix} .167 & 0 \\ 0 & .167 \end{bmatrix}$$

In general, the measurement error covariance matrix for 2-dimensional motion can be written as:

$$R_k = \begin{bmatrix} \sigma_{e,x}^2 & 0 \\ 0 & \sigma_{e,y}^2 \end{bmatrix}$$ where, $\sigma_{e,x}^2$ and $\sigma_{e,y}^2$ are the variance in x-coordinate measurement and y-coordinate measurement respectively.

Process error covariance matrix

For constant velocity model, the process noise is a result of deviation in constant velocity and is associated with random variable X_a. As per [87], the random acceleration $X_{a,x}$ and $X_{a,y}$ in *x,y-plane* are orthogonal and independent and given by:

$$E[X_a(u)X(v)_a^T] = \begin{bmatrix} E[X_{a,x}(u), X_{a,x}(v)] & E[X_{a,y}(u), X_{a,y}(v)] \\ E[X_{a,y}(u), X_{a,x}(v)] & E[X_{a,y}(u), X_{a,y}(v)] \end{bmatrix}$$

$$= \begin{bmatrix} a^2\delta(u-v) & 0 \\ 0 & a^2\delta(u-v) \end{bmatrix}$$

where a is the white noise amplitude and δ is Dirac delta impulse.

As $Q_k = G.E[X_a(u)X(v)_a^T].G^T$, with matrix G equal to identity matrix, the resulting process noise covariance matrix according to [87] is:

$$Q_k = a^2 \begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix}$$

For constant acceleration model, the acceleration in motion is incorporated using the equation:

$$G_k = \begin{bmatrix} (\Delta t/2)^2_{k,x} \\ (\Delta t/2)^2_{k,y} \\ \Delta t_{k,x} \\ \Delta t_{k,y} \end{bmatrix}$$

Using $Q_k = G.G^T a^2$, the process noise covariance matrix is:

$$Q_k = a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \frac{\Delta t}{2} & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \frac{\Delta t}{2} \end{bmatrix}$$, where a^2 is the spectrum amplitude.

For the Kalman filter's first iteration, in addition to state vectors, state matrices and noise co-variances, we need to determine the initial state vector $x_{k-1|k-1}$ for equation $x_{k|k-1} = A_k x_{k-1} + B_k u_{k-1}$ and initial process noise covariance P_{k-1} for equation $P_{k|k-1} = A_k P_{k-1} A_k^T + Q_k$. For simplicity and also because both $x_{k|k-1}$ and $P_{k|k-1}$ are iteratively corrected by the Kalman filter, the x_{k-1} and P_{k-1} are set to initial measurement and process error covariance matrix respectively.

Kalman Recursion

As mentioned in Section 2.3.2, the Discrete Kalman filter is conceptualized as time update phase and the measurement update phase. Once the vectors, state matrices, noise co-variances, etc. are determined, the Kalman filter recursively predicts and corrects the estimated states.

Predict (Time Update):

The time update projects the state and covariance estimates from time step *k-1* to step *k* as:

$x_{k|k-1} = A_k x_{k-1} + B_k u_{k-1}$ estimates the current state from previous estimated state.

$P_{k|k-1} = A_k P_{k-1} A_k^T + Q_k$ estimates the current error covariance from previously updated error covariance.

Measurement Update (Correction):

The correction update calculates the Kalman gain, estimates the *posterior* state using the actual measurement and generates the *posterior* covariance estimate.

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$

The Kalman gain controls the effect of measurement error. Larger the variance in measurement, smaller will be the Kalman gain.

$$x_k = x_{k|k-1} + K_k(z_k - H_k x_{k|k-1})$$

The equation updates the current state estimate with current measurement. The Kalman gain weights the difference between the measurement and the state estimated before measurement (i.e. $K_k(z_k - H_k x_{k|k-1})$).

$$P_k = (I - K_k H_k) P_{k|k-1}$$

The equation updates the error covariance estimate with current measurement. The more informative the measurement ($K_k H_k$), better will be the error covariance estimate.

3.1.2 Mobility Models

To simulate the movement of a person walking in an area of 20x20 ft^2, a few mobility models were used in our research. The models were based upon *Random Waypoint model* **(**as explained in Section 2.4). The random generated paths account for the person's movement and provide data on how the person's location, velocity and acceleration changes over time.

The preferred human walking speed ranges from nearly 0 ft/s to maximum of 8.5 ft/s (9.0 km/h; 5.6 mph) with typical walk speed limiting to 4.5 ft/sec [95]. Factors that contribute to speed selection are mechanical, energetic, physiological and psychological. Motivation, destination time factor (time to reach the destination) and metabolic efficiency usually results in faster walks whereas ageing, joint pain, instability and decreased metabolic rate can cause people to walk slowly [95].

To simulate the human movement in the 20x20 ft^2 area, we proposed four simulation models namely: *Constant Rate Linear Walk, Variable Rate Linear Walk, Constant Rate non-linear Walk* and *Variable Rate non-linear Walk*. All of these models are based upon Random Waypoint model and are independent of each other.

3.1.2.1 Constant Rate Linear Walk

Constant rate linear walk model randomly selects the destination (random waypoint). Once the destination is selected, the user moves along a straight line towards the destination from the current position. The user walks with a step size of approximately 1 foot. Once, the user reaches destination, the model again randomly locates a new destination and the process repeats resulting in constant velocity linear paths. The walk speed is randomly selected from 0.5 ft/sec to 4.5 ft/sec (the normal human walk speed range) for each destination and remains constant for that transit.

3.1.2.2 Variable Rate Linear Walk

Variable rate linear walk model is based upon the *constant rate linear walk model.* Similar to *constant rate linear walk model*, the variable rate linear walk model randomly selects the destination inside the simulation area of 20x20 ft^2. Once the destination is selected, the user moves along a straight line towards the destination point from the current position and changes its velocity with every step. In other words, acceleration and de-acceleration are incorporated for the variable rate linear walk model. The user moves along a straight line with an initial velocity selected randomly from a range of 0.5 ft/sec to 4.5 ft/sec. Thereafter, for the first half of the distance, with each step, the velocity of the user increases so as to reach a maximum velocity which is selected randomly (from range: initial speed to 8.5 ft/sec). For the remaining half of the distance, the velocity decreases with each step such that the velocity at the final location (destination) is equal to the initial velocity at the beginning. Once at the destination, the process repeats for a new destination, considering the current destination as the current position. Hence, *variable rate linear walk model* incorporates acceleration and de-acceleration in motion along a straight line.

3.1.2.3 Constant Rate Non-linear Walk

Both *constant rate linear walk model* and *variable rate linear walk model* consider the motion to be along a straight line. However, in case of obstruction (for example a table) in the path from current position to destination, the human walk/motion will bend around the obstruction to reach the destination with shortest covered distance. To determine the shortest path when an obstruction is present we adopt the A* search approach mentioned in Section 2.5.

To adopt the *A* search*, we divided the floor area of 20x20 ft^2 into a grid of 400 squares each of size 1 foot. This square size is important because it helps determines the step size, which we desire to be approximately 1 foot. At first, the destination is randomly selected. Once the destination is selected, random tagging of the 400 squares as walkable and non-walkable follows. As the name suggests, walkable squares are those on which the user can walk and non-walkable square are the ones representing an obstruction and hence cannot be walked upon. Following is the criteria for square selection (refer to Figure 3.1).

- Grid squares that are on the boundary of the floor are never tagged as non-walkable.
- The squares surrounding the current position square and destination position square are never tagged as non-walkable.
- Each grid square is surrounded by at least six walkable squares

The above selection criteria ascertain that the non-walkable tagged squares never block away the path between the current position and the destination position. Iterating the above logic several times results in walkable and non-walkable grid squares. After the grid squares are tagged as walkable and non-walkable, *A* search* is used to determine the shortest path from current position to the randomly selected destination.

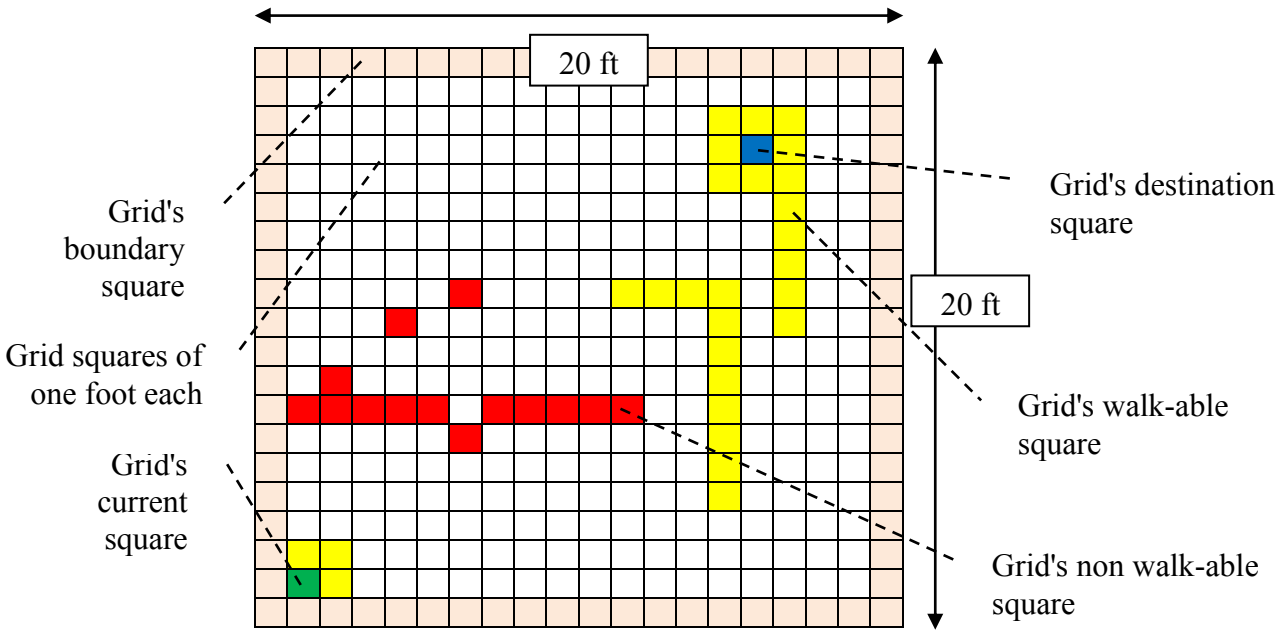

Figure 3.1 Grid Representation Of 20x20 ft^2. Floor Area

The path generated by *A* search* though is shortest, but have sharp bends and turns. Assuming that a user (human being), while walking, avoids unnecessary bends and follows a path in a smooth route, the path generated by *A* search* is smoothened. *Gradient descent smoothening algorithm* is used to give smooth curves to the sharp turns and bends.

To transform the planned path into a smooth path, the points (coordinates on which the user steps) on the path are iteratively updated. Please refer to Figure 3.2 [96] for a description of the smoothening process. Here x_i's represent the original points on the path and y_i's represent the smooth points for the x_i's. To smoothen the path, the two functions $(x_i - y_i)^2$ and $(y_i - y_{i+1})^2$ are minimized. If only $(x_i - y_i)^2$ is minimized, the smooth path is equal to the actual path because equating the first derivate of $(x_i - y_i)^2$ to zero yields $x_i = y_i$. Also, if only $(y_i - y_{i+1})^2$ is minimized, the smooth path is equal to a point because equating the first derivate of $(y_i - y_{i+1})2$ to zero yields $y_i = y_{i+1}$. By assigning appropriate weights to both functions $(x_i - y_i)^2$ and $(y_i - y_{i+1})^2$, a smooth path can be obtained.

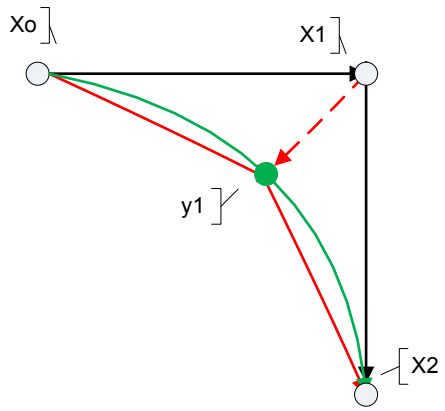

Figure 3.2 Path Smoothing

Considering Figure 3.2, the functions $(x_i - y_i)^2$ and $(y_i - y_{i+1})^2$ are minimized iteratively using gradient descent. For all iterations, considering the difference between x_0 and x_1, and x_2 and x_1, the function $(y_i - y_{i+1})^2$ can be written as:

$x_1 = x_1 + \alpha((x_0 - x_1) + (x_2 - x_1))$.

$\Rightarrow y_1 = y_1 + \alpha(y_0 + y_2 - 2y_1)$ (3.1.15)

where, $y_1 = x1$, $y_0 = x_0$ and $y_2 = x_2$, in order to retain the x_i's values for the next function and for next points in the path. Here α determines how close y_1 is to new points y_{1-1} and y_{1+1}. Similarly for each iteration, the function $(x_i - y_i)^2$ can be written as:

$y_1 = y_1 + \beta(x_1 - y_1)$ (3.1.16)

where, β determines the displacement of y_1 from x_1.

As already mentioned, the equations 3.1.15 and 3.1.16 are iteratively minimized using gradient descent. The weights α and β control how smooth or how similar to the original path the new path ends up being. The gradient descent is an optimization algorithm that determines the minimum of a function and is defined as:

$$b = a - \gamma \nabla F(a)$$

Here, b is the new point (i.e. y_1 in our case), a is the old point (i.e. x_1 in our case) and $\nabla F(a)$ determines the steepest descent with the weight gamma (γ). Consider the following function to be minimized:

$$F(y_i) = \alpha/2(y_i - x_i)^2 + \beta/2(y_i - y_{i+1})^2 + \beta/2(y_i - y_{i-1})^2$$

Its derivate is $\frac{\partial y}{\partial x} F(y_i) = \alpha(x_i - y_i) + \beta(y_{i+1} + y_{i-1} - 2y_i)$ which is similar to equations 3.1.15 and 3.1.16. Figure 3.3 shows an example of the path generated using A^*

search in red and the same path after smoothening using *gradient descent smoothing* [96] in blue.

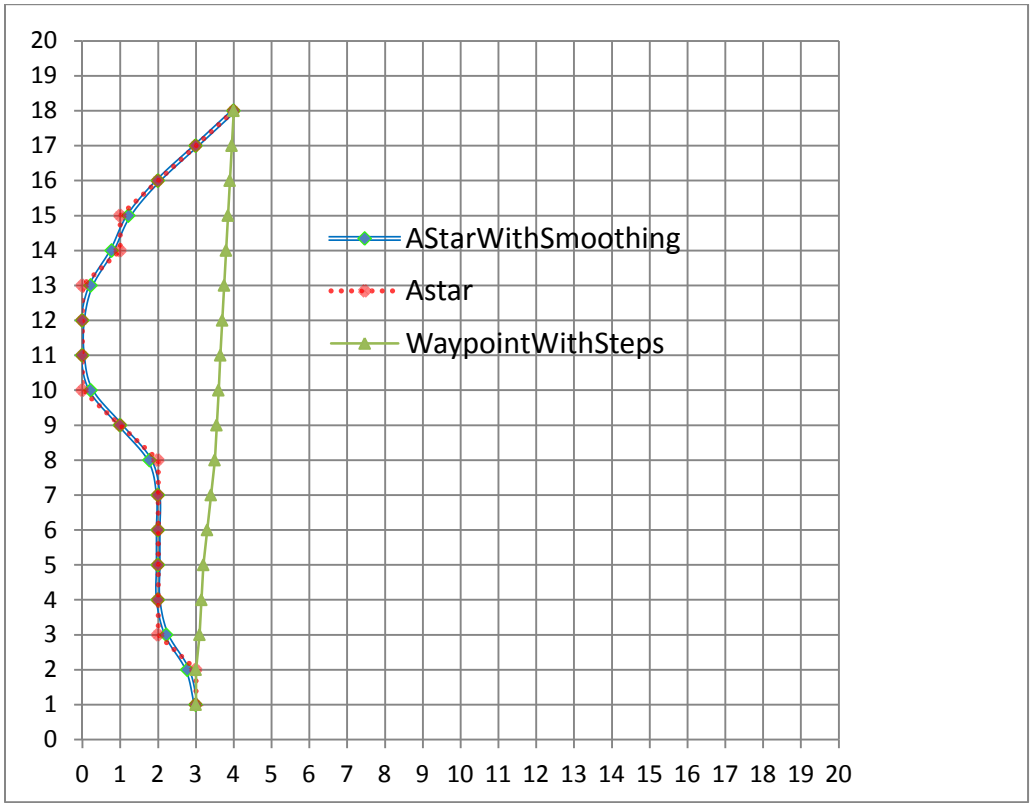

Figure 3.3 Non-linear Path With Smoothing

Once the *A* search* generates a path, the grid squares are converted back to points and are smoothened by *Gradient descent smoothening algorithm*. This results in a smooth non-linear path with step size of approximately 1 foot. Thereafter, the speed of motion is selected in a manner similar to *constant rate linear walk model*. Once the user reaches the destination, a new random destination is selected. The current destination becomes the current position and the process repeats resulting in constant velocity non-linear paths.

3.1.2.4 Variable Rate Non-linear Walk

Variable rate non-linear walk is similar to *constant rate non-linear walk* but incorporates acceleration and de-acceleration (as explained for *Variable Rate non-linear Walk model*). Once the *A* search* generated path is smoothened by *Gradient descent*

algorithm, the velocity parameters are selected. The initial velocity (i.e. the velocity at the current location) is randomly selected from the range 0.5 ft/sec to 4.5 ft/sec. With each step, the velocity of the user increases and attains a maximum velocity by half of the distance. The maximum velocity is randomly selected from a range between randomly initial velocity and velocity of 8.5 ft/sec. Depending upon the maximum velocity selected, the increase in velocity with each step is determined. Thereon, the velocity decreases with each step such that the velocity at the destination is equal to the initial velocity at the beginning. After reaching the destination, again, a new destination is selected and the process repeats resulting in *variable velocity non-linear paths.*

3.2 STEPSCAN™ TILES

Stepscan™ is designed by ViTRAK Systems Inc. Stepscan™ tiles use proprietary sensor floor technology patented by ViTRAK Systems Inc. The tiles are capable of not only identifying and locating the person, but can also detect early signs of dementia and underfoot pathologies such as diabetic foot ulcers. The Stepscan™ tiles are capable of tracking multiple subjects, and are expandable and customizable. Each tile has a size of 2x2 ft^2 and comes in various designs. As we are interested in tracking and predicting the location of a person, we utilize the localization capability of the tiles.

3.3 RFID TEST BED

As already mentioned in Chapter 2, RFID technology can be used to identify targets with RFID tags within the reader's read range. It can be divided into active and passive types depending upon the RFID reader and tags being used. For our experimental setup, we used RFID passive tags with passive RFID reader. The Gao UHF RFID reader is ideal for indoor tracking. It can read up to 430 tags per second and connects to four monostatic antennas with maximum receive sensitivity of -82 dBm. Figure 3.4 [88] shows Gao RFID UHF Gen 2 RFID Reader/Writer with 4 Antenna Ports. For RFID reader specifications, refer to Table 3.1 [89]. The RFID reader is attached to four circular polarization Gao RFID antennas. The antennas operate in the 902 MHz to 928 MHz frequency range and are compatible with Gao RFID UHF Gen 2 RFID Reader/Writer.

Figure 3.5 [90] shows UHF 902 MHz RFID Antenna Circular Polarization Indoor. For RFID antenna's specifications, refer to Table 3.2 [91]. The UHF RFID Clothing tags are suitable for use in textile industry and are shown in Figure 3.6 [92]. The clothing tags can withstand printing, dyeing and washing. For tag specifications, refer to Table 3.3 [93].

Figure 3.4 GAO RFID UHF Gen 2 RFID Reader/Writer With 4 Antenna Ports

Table 3.1 RFID Passive Reader Specifications

RFID Reader	Goa RFID: UHF Gen 2 RFID Reader/Writer with 4 Antenna Ports
Model	236015
Air Interface protocol	EPC Global Class 1 Gen 2/ISO 18000-6C
Typical Throughout	Approx. 430 tags/s
Supported Regions	• US, Canada and other regions following US FCC Part 15 regulations • Europe and other regions following ETSI EN 302 208 v1.2.1 without LBT regulations • Brazil
Antennae	4 high performance, monostatic antenna ports
Transmit Power	• 10.0 to 30.0 dBm (PoE) • 10.0 to 32.5 dBm (external universal power supply)
Max Receive Sensitivity	-82 dBm
Max. Return Loss	10 dB
Network Connectivity	10/100BASE-T auto-negotiate (full/half) with auto-sensing

<table>
<tr><td></td><td>MDI/MDX for auto-crossover (RJ-45)</td></tr>
<tr><td>IP Address Configuration</td><td>• DHCP
• Static
• Link Local Addressing (LLA) with Multicast DNS</td></tr>
<tr><td>Time Synchronization</td><td>Network Time Protocol (NTP)</td></tr>
<tr><td>Management Interfaces</td><td>Management Console using serial management console port, telnet or SSH</td></tr>
<tr><td>Management Console</td><td>RS-232 using a standard Cisco-style management cable (DB-9 to RJ-45)</td></tr>
<tr><td>Power Source</td><td>• Power over Ethernet (PoE) IEEE 802.3af
• 24 V DC, 800 mA via external universal power supply with locking connector-sold separately</td></tr>
<tr><td>Power Consumption</td><td>
<table>
<tr><td></td><td>Idle</td><td>Typical</td><td>LDC</td></tr>
<tr><td>PoE at 30 dBm</td><td>3 W</td><td>11.5 W</td><td>6 W</td></tr>
<tr><td>Power Supply at 6</td><td>3 W</td><td>13.5 W</td><td>6 W</td></tr>
</table>
</td></tr>
<tr><td>Operating Temperature</td><td>-20 °C to 50 °C</td></tr>
<tr><td>Dimensions</td><td>7.5 in × 6.9 in × 1.2 in (19 cm × 17.5 cm × 3 cm)</td></tr>
<tr><td>Weight</td><td>1.5 Ibs (24.5 oz)</td></tr>
</table>

Figure 3.5 UHF 902 MHz RFID Antenna Circular Polarization Indoor

Table 3.2 RFID Antenna Specifications

Model	326003
Polarize style	Circular polarization
Operating Frequency	902 MHz to 928 MHz
Center Frequency	915 MHz
Gain	7 dBi ± 1 dBi
VSWR	<1.3:1 (902 MHz to 928 MHz)
Natural impedance	50 Ω
Maximum Power	10 W
Dimensions	245 mm × 235 mm × 40 mm
Weight	470 g
Case Material	Glass fiber reinforced plastic
Operating and Storage Temperature	-20 °C to 70°C
Compatibility	Compatible with any 902 MHz - 928 MHz Reader

Figure 3.6 UHF RFID Clothing Tag

Table 3.3 RFID Tag Specifications

Model	116060
Physical Parameters	
Inlay dimension	125 mm x 7 mm
Package material	coated paper

Operating temperature	-40 °C to 65 °C
Storage temperature	-40 °C to 85 °C
Functional Parameters	
RF frequency	860 to 960 MHz
Read distance	5 to 7 m
Compliant standard	ISO/IEC18000-6C

For the RFID test bed (refer to Figure 3.7 below), RFID interrogation zones were created to read and identify the tags. The test bed, as shown in Figure 3.8 below, consists of four RFID antennas and one RFID reader. The RFID antennas can transmit the RFID signals (radio waves) and receive the data transmitted by the excited tags. The reader collects the data from the antennas and converts it into useful information, which can later be processed using application logic. All the four antennas are connected to the reader using N-type female connector.

These four antennas form four RFID zones, namely *zone1, zone2, zone3* and *zone4* which are represented by green, pink, orange and blue colors respectively (refer to Figure 3.8 and Figure 4.1). In case of no obstruction in the path of RFID signals, the zones formed are circular with a radius of 5 feet each. Please note that the circular shape of the interrogation zone attains a right angle at the corners of the room due to interference (i.e. reflection of RFID signals) by the walls. Hence, resulting in RFID zones shape as shown in Figure 3.8 and Figure 4.1. In total, the four RFID antennas cover a square area of 20x20 ft^2. To achieve precise zonal boundary, the application logic uses the RSSI value (signal strength) and the tag detection frequency within each zone. Using the RSSI value and the tag detection frequency within each zone, we were able to create the interrogation zones with sharp boundaries.

The tag detection frequency within a RFID zone is directly proportional to visibility of the tag to the RFID transmitted signals from the antenna. Two passive tags were placed on both the shoulders of the person being tracked. This arrangement of tags helps to avoid the RFID signal interference due to the human head. In order to backscatter the signal from the tags, it was important to keep the tags at a height of

approximately 1 inch from the shoulders. This resulted in better detection and read accuracy of the RFID tags.

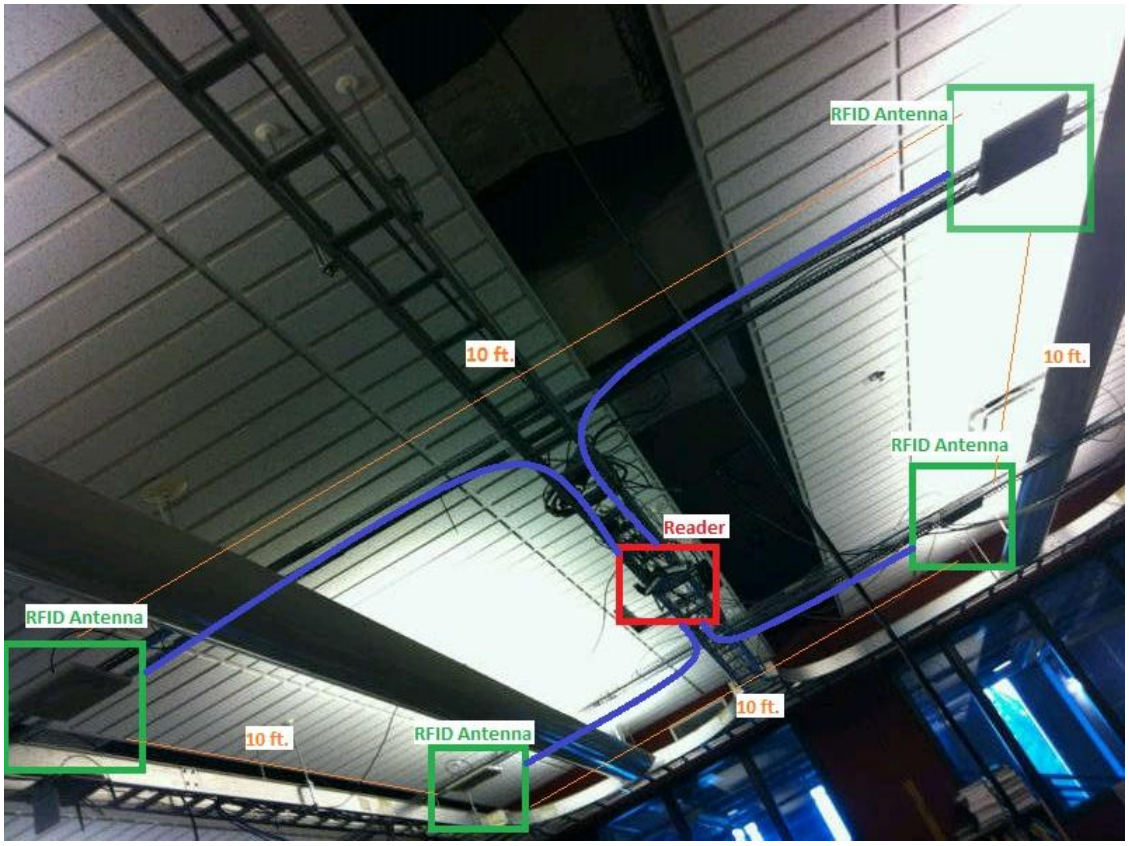

Figure 3.7 RFID Test Bed

The RFID reader connects and collects the data from all four RFID antennas. Furthermore, the reader also connects to a computer that runs the application logic and controls the working of the RFID reader and RFID antennas. The application running on the computer (it may also be integrated to run on the RFID reader itself) is used to configure the reader and antenna parameters as per the requirements. For our purpose, we configured each of the four antennas as per data in *Appendix A: RFID antenna and reader configuration*.

The RFID system generates target's zonal information and gives the received signal strength (RSS) of the RFID tag several times per second. This data is used to improve the tracking performed by Kalman filter.

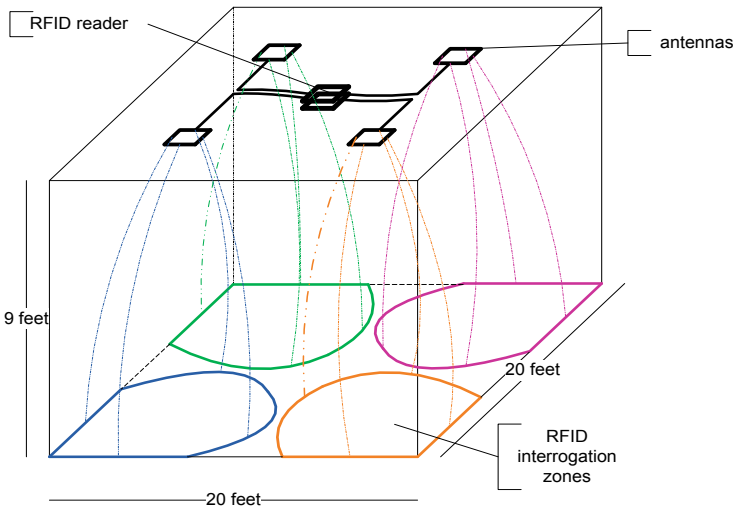

Figure 3.8 RFID Reader And Antenna Setup

3.4 HIGH-PRECISION INDOOR TRACKING

The simulation area which combines STEPSCAN^{TM} tiles and RFID is described in Figure 3.9. The experimental area (for the purpose of simulation) is overlaid with partially-deployed STEPSCAN^{TM} tiles and RFID interrogation zones. The area is divided into four RFID interrogation zones. Within each zone, sub-zones are formed using the RSS information.

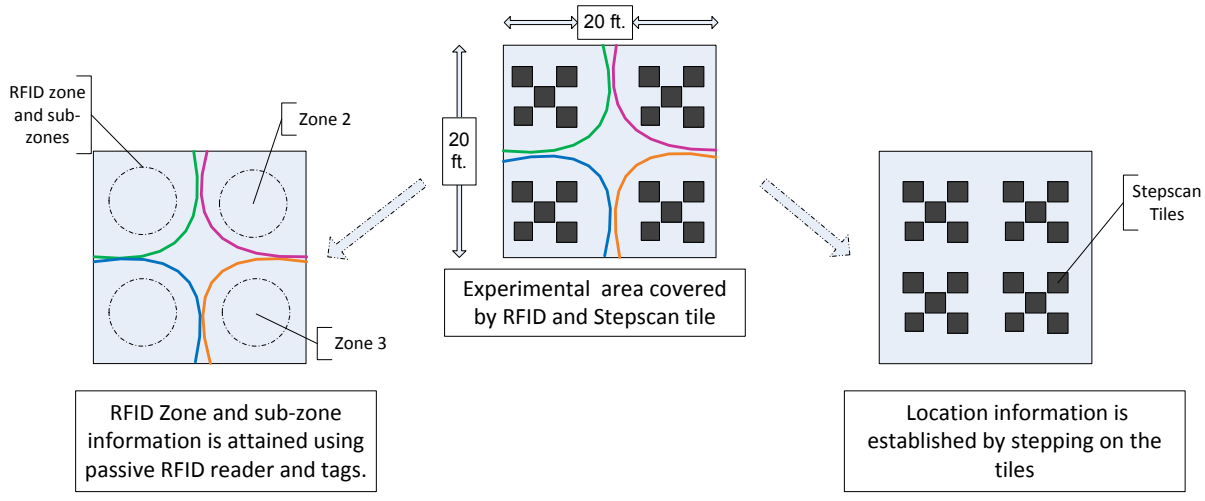

Figure 3.9 Experimental Area

The STEPSCAN^{TM} tiles can generate the person's precise location information whenever he/she steps on a tile. In addition, the RFID system can generate some location-related information (i.e. the person's current RFID interrogation zone and the sub-zone information) as the person walks around within the RFID's interrogation zones. The information from the tiles and RFID is utilized by the Kalman filter to track the person precisely when he/she steps on a tile and estimate his/her location when precise location information is not available (Refer Figure 3.10). Without loss of generality, we assume that time is divided into timeslots and the area of the experimental field is 20x20 ft^2.

Here are the detailed steps that the proposed system uses to track a person effectively:

- When the person's precise location is found by a tile (i.e. when the person steps on a tile) during a timeslot, the Kalman filter is initiated. The precise location is considered to be the current location and it is used as the first input for the Kalman filter. The Kalman filter will be used to estimate the future location of the person during the following timeslots.
- For each of the following timeslots:
  - If again the person's location is found by a tile during the timeslot, the precise location is considered to be the current location, which is also used as the input for the Kalman filter for the purpose of estimating future locations.
  - If the person does not step on the tile during the timeslot (i.e. no precise location information is generated by a tile), the Kalman filter will estimate the current location of the person using the precise or estimated location information obtained during the preceding timeslots.

    Note that the precision of the estimations generated by the Kalman filter (i.e. the location information generated by the proposed system when the person does not step on a tile) could fluctuate seriously. In the extreme cases, the estimated location could be very far from the real spot. To guarantee that the output of the proposed tracking system does not diverge significantly from the real locations, our system uses the RFID readings to filter the estimated results. Specifically, in the simulations, for each of the

Kalman filter's estimated location, the RFID interrogation zone and RFID sub-zone information corresponding to the estimated location is compared to the person's zonal and sub-zone information from the RFID test bed. In the cases where there is a significant discrepancy, the Kalman filter estimation process will stop until the person's precise location is obtained through a STEPSCANTM tile (i.e. the person steps on a tile).

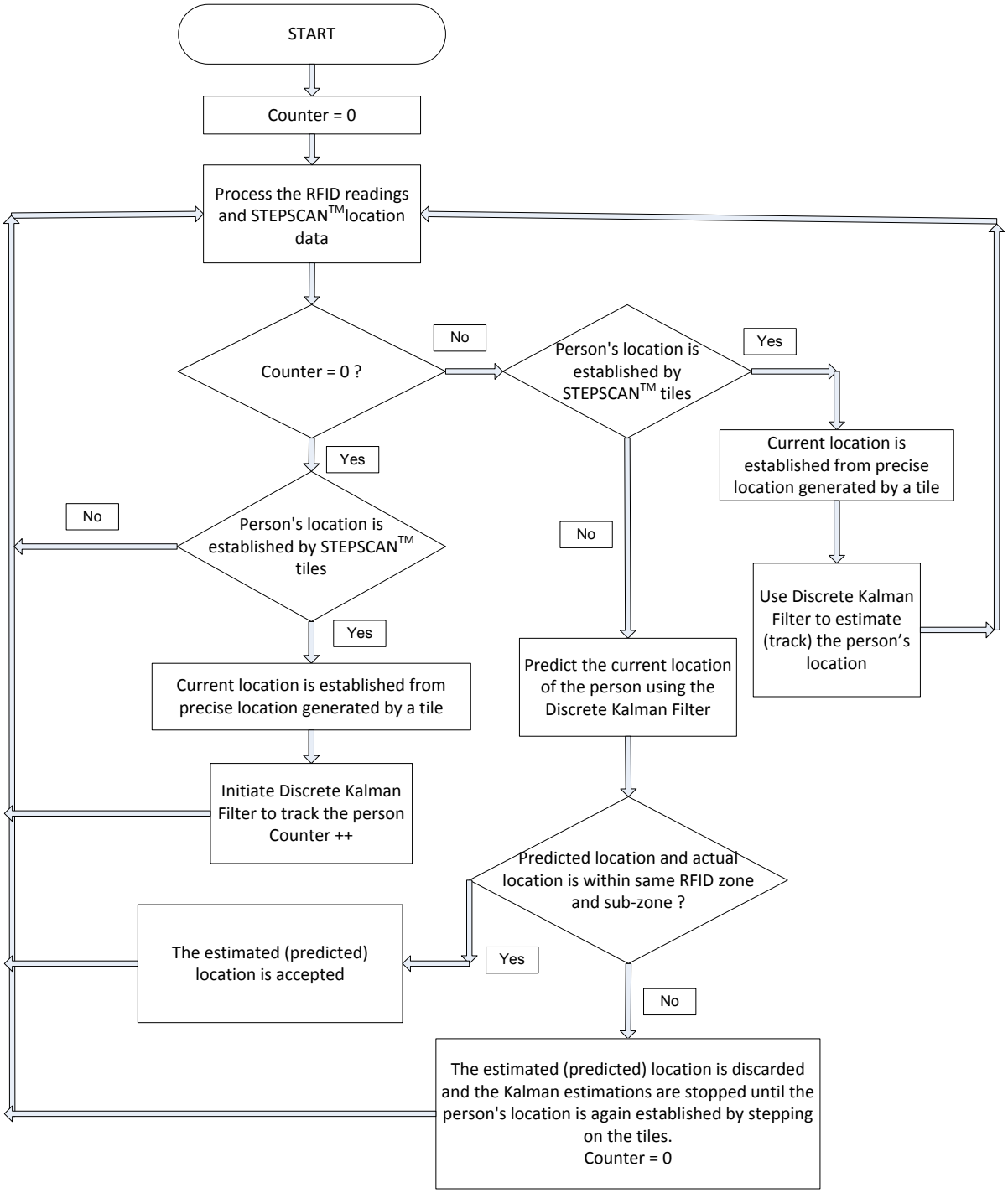

Figure 3.10 System Design and Location Estimation

CHAPTER 4 EXPERIMENTAL RESULTS

This chapter presents the details of the simulations and the experimental results. The primary objective of the study is to reduce the number of STEPSCAN^{TM} tiles used for tracking and locating a person. The Kalman filter utilizes the location information generated by the tiles to estimate the position of the person in an indoor environment.

4.1 SIMULATION DESIGN

This section describes the simulation model that integrates RFID, Stepscan^{TM} tiles and the Kalman filter. The simulation application was developed using Java and Matlab. The model simulates the movement of a person walking in a room with floor size of 20x20 ft^2 that is overlaid by Stepscan^{TM} tiles and RFID interrogation zones. As the person moves around the room (using the mobility models described in Section 3.1.2), the data generated by stepping on the tiles and the information available from the RFID reader is combined and used as an input for the Kalman filter. The integration of RFID, Stepscan^{TM} tiles and the Kalman filter not only allows to track the movement of the person (when the user steps on the tiles), but also predicts the current step location when the person is not stepping on the tile as per the application logic. The RFID readings are used to improve the Kalman filter estimations by stopping the estimations whenever necessary.

As described in Section 3.1.1, the Kalman filter parameter selection depends upon the physical model of the system, the noise associated with the process and measurement errors. For the mobility models described in Section 3.1.2, the parameters for Kalman filter are described in Table 4.1.

Table 4.1 Kalman Parameter For Walk Models

Kalman parameters	physics of interest	state transition matrix and time phase update state estimation	process error covariance Q_k	measurement error covariance R_k
Constant rate linear walk	$p(t_{t+1}) = p(t) + \Delta t v(t)$ $v(t_{t+1}) = v(t)$	$\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $x_{k\|k-1=} A_k x_{k-1}$	$\begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix}$	$\begin{bmatrix} .167 & 0 \\ 0 & .167 \end{bmatrix}$
Variable rate linear walk	$p(t_{t+1}) = p(t) + \Delta t v(t) + \frac{\Delta t^2}{2} a(t_n)$ $v(t_{t+1}) = v(t) + \Delta t v(t_n)$	$\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $x_{k\|k-1=} A_k x_{k-1} + G a_{k-1}$	$\begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \frac{\Delta t}{2} & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \frac{\Delta t}{2} \end{bmatrix}$	$\begin{bmatrix} .167 & 0 \\ 0 & .167 \end{bmatrix}$
Constant rate non-linear walk	$p(t_{t+1}) = p(t) + \Delta t v(t)$ $v(t_{t+1}) = v(t)$	$\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $x_{k\|k-1=} A_k x_{k-1}$	$\begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix}$	$\begin{bmatrix} .167 & 0 \\ 0 & .167 \end{bmatrix}$
Variable rate non-linear walk	$p(t_{t+1}) = p(t) + \Delta t v(t) + \frac{\Delta t^2}{2} a(t_n)$ $v(t_{t+1}) = v(t) + \Delta t v(t_n)$	$\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $x_{k\|k-1=} A_k x_{k-1} + G a_{k-1}$	$\begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \frac{\Delta t}{2} & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \frac{\Delta t}{2} \end{bmatrix}$	$\begin{bmatrix} .167 & 0 \\ 0 & .167 \end{bmatrix}$

In brief, as explained in Section 3.1.1, the Kalman filter model for motion with constant velocity holds true for *Constant Rate Linear Walk model* and *Constant Rate Non-linear Walk model*. Similarly, Kalman filter model for motion with constant acceleration holds true for *Variable Rate Linear Walk model* and *Variable Rate Non-linear Walk model*.

The zones created by the RFID system are further divided into sub-zone using the RSS information as shown in Figure 4.1. The RSSI value for each zone ranges from -82 dBm (at the centre of the zone) to -55 dBm (at the boundary of each zone). The sub-zones are created using the variation in the RSS level from centre of the zone towards its boundary. Within each zone, a sub-zone is formed at -62 dBm which divides the RFID zone into two sub-zones at around 4 ft from the centre of each zone. The inner sub-zone is of radius 4 ft from the centre of the zone and the area with internal radius of 4 ft and outer boundary as the RFID's zone boundary forms the second sub-zone. The subdivision helps in improving the person's location estimations when the measurement data is not attainable (i.e. when the person is not stepping on the tile). In particular, the Kalman estimation stops when the estimated location's sub-zonal and zonal information mismatches with the person's actual position sub-zonal and zonal information.

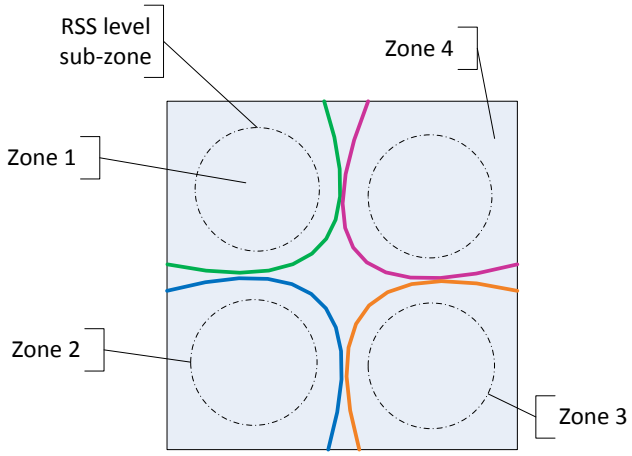

Figure 4.1 RFID Subzones

In order to integrate the tiles onto the simulation area, six floor plans are proposed. In each of the six floor plans, the tiles are deployed to cover a part of the indoor floor. Table 4.2 provides details on the number of tiles in each plan and the area

being covered. Each tile size is 2x2 ft^2 and generates a 2-dimensional location in Cartesian coordinates. Please refer to *Appendix B: Tile Deployment Patterns* for floor plans snapshots.

Table 4.2 Tiles Layout

Floor Plans	Number of Tiles	Area Covered (Total area 400 sq. ft)
Layout24	24 tiles	96 sq. ft
Layout28	28 tiles	112 sq. ft
Layout32	32 tiles	128 sq. ft
Layout36	36 tiles	144 sq. ft
Layout40	40 tiles	160 sq. ft
Layout48	48 tiles	192 sq. ft

4.2 DETAILED EXPERIMENTAL RESULTS

4.2.1 Indoor Tracking Performance

Tables 4.3 to 4.6 includes the details of our experimental results for floor plan layouts mentioned in Table 4.2. In general, it is observed that the estimation rate as well as the estimation accuracy is more for *Constant Rate Linear Walk Model* and *Variable Rate Linear Walk Model* in comparison to *Constant Rate Non-linear Walk Model* and *Variable Rate Non-linear Walk Model*. For all the four simulation models, the estimated location is mostly within 1 ft of the actual location. As we move from linear models to non-linear models, the divergence of the estimated location from the actual location increases, but still the estimated location is mostly within 1 ft of the actual location. *Appendix C: Simulation Execution Results* section shows the execution results for each of the floor plans for various simulation models. Mathematically, the columns for the tables are defined as:

Tile Trace Coverage *= (no. of steps on tiles/total no. of steps on the floor)*100*

Kalman Trace Coverage with Half ft. accuracy *= (no. of steps predicted using Kalman Filter with deviation <= 0.5 ft from the actual location/ total steps on the floor)*100*

Kalman Trace Coverage with accuracy greater than Half ft. & less than One ft. *= (no. of steps predicted using Kalman Filter with deviation >0.5 ft and <= 1.0 ft from the actual location/total steps on the floor)*100*

Kalman Trace Coverage with accuracy greater than One ft. & less than Two ft. *= (no. of steps predicted using Kalman Filter with deviation >1.0 ft and <= 2.0 ft from the actual location/total steps on the floor)*100*

Kalman Trace Coverage with accuracy greater than Two ft. *= (no. of steps predicted using Kalman Filter with deviation >2.0 ft from the actual location/total steps on the floor)*100*

Trace Gain for Half ft. accuracy *= ((Kalman Trace Coverage with Half ft. precision)/Tile Trace Coverage)/100*

Trace Gain for accuracy greater than Half ft. & less than One ft. *= (Kalman Trace Coverage with accuracy greater than Half ft. & less than One ft. / Tile Trace Coverage)*100*

Trace Gain for accuracy greater than One ft. & less than Two ft. *= (Kalman Trace Coverage with accuracy greater than One ft. & less than Two ft. / Tile Trace Coverage)*100*

Trace Gain for accuracy greater than Two ft. *= (Kalman Trace Coverage with accuracy greater than Two ft./Tile Trace Coverage)*100*

Table 4.3 describes the results for *Constant Rate Linear Walk Model*. The estimation rate as well as the accuracy increases with an increase in the number of tiles. As we move from a 24 tile floor plan to a 48 tile floor plan, the location estimation increases from 38.80 % to 40.07 % for estimation precision within half feet of the actual location. Also, the error (i.e. the location estimation with more than 2 ft. deviation from person's actual location) decreases from 7.07 % to 2.40 % as we shift from 24 tiles plan to 48 tiles plan.

Table 4.4 describes the results for *Variable Rate Linear Walk Model*. The observations described for *Constant Rate Linear Walk Model* holds true for *Variable Rate Linear Walk Model* with a slight decrease in the location estimation rate. For estimations with precision within half feet of the actual location, the gain in trace ranges from 29.0 %

to 33.00 %. In addition, the estimation error decreases significantly from 6.85 % to 2.57 % with an increase in the number of tiles.

Table 4.5 and 4.6 provide data for the *Constant Rate Non-linear Walk Model* and *Variable Rate Non-linear Walk Model* simulation execution. As expected, the estimation rate and accuracy decreases with increase in the simulation complexity. As both of the non-linear simulation models are more complex and random in comparison to linear simulation models, their system accuracy and location estimation capacity decreases in comparison to the linear models. For *Constant Rate Non-linear Walk Model* the trace gain varies from 15.10 % to 22.20 % for precision within half feet of the actual location. Also, the error in estimation decreases with an increase in the number of tiles. The *Variable Rate Non-linear Walk Model* shows a somewhat random behavior (in terms of trace gain and error generated) with changes in the floor plans. The fluctuation in estimations are reasonable as the effectiveness of the Kalman filter to predict the person's location is influenced by the placement design of the tiles. We can expect a change in the observed simulation results with a change in tile's layout designs.

Table 4.3 Constant Rate Linear Walk Model Statistics

Tile Configuration	Area Covered by tiles (in sq. ft)	Tile Trace Coverage (in percentage)	Kalman Trace Coverage (in percentage)				Trace Gain With Respect to Tile Trace Coverage (in percentage)			
			Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.
24 Tiles	96	30.40	11.80	4.65	3.50	2.15	38.80	15.30	11.50	7.07
28 Tiles	112	39.30	15.70	5.45	4.10	2.85	39.95	13.86	10.43	7.25
32 Tiles	128	40.50	13.10	6.40	3.85	2.90	32.35	15.80	9.50	7.15
36 Tiles	144	42.70	17.30	5.25	4.60	3.40	40.50	12.30	10.77	7.95
40 Tiles	160	48.00	17.80	6.50	3.60	1.40	37.10	13.54	7.50	2.90
48 Tiles	192	52.15	20.90	5.90	3.75	1.25	40.07	11.30	7.20	2.40

Table 4.4 Variable Rate Linear Walk Model Statistics

Tile Configuration	Area Covered by tiles (in sq. ft)	Tile Trace Coverage (in percentage)	Kalman Trace Coverage (in percentage)				Trace Gain With Respect to Tile Trace Coverage (in percentage)			
			Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.
24 Tiles	96	31.40	9.10	5.60	4.40	2.15	29.00	17.83	14.00	6.85
28 Tiles	112	40.80	13.40	6.23	3.95	3.10	32.85	15.27	9.68	7.35
32 Tiles	128	43.10	13.50	6.90	5.80	2.90	31.32	16.00	13.45	6.73
36 Tiles	144	42.40	13.60	5.63	5.55	3.30	32.10	13.28	13.10	7.78
40 Tiles	160	50.60	16.25	6.20	4.55	2.33	32.10	12.25	9.00	4.60
48 Tiles	192	52.40	17.30	5.80	4.65	1.35	33.00	11.05	8.87	2.57

Table 4.5 Constant Rate Non-linear Walk Model Statistics

Tile Configuration	Area Covered by tiles (in sq. ft)	Tile Trace Coverage (in percentage)	Kalman Trace Coverage (in percentage)				Trace Gain With Respect to Tile Trace Coverage (in percentage)			
			Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.
24 Tiles	96	29.20	4.43	3.65	4.43	4.43	15.10	12.5	15.10	15.10
28 Tiles	112	36.83	5.60	5.45	6.60	3.95	15.20	14.80	17.90	10.70
32 Tiles	128	37.60	6.30	4.80	5.30	3.50	16.75	12.75	14.10	9.30
36 Tiles	144	41.50	8.77	5.75	5.50	3.53	21.13	13.85	13.25	8.50
40 Tiles	160	44.37	9.85	6.25	4.60	2.35	22.20	14.08	10.35	5.30
48 Tiles	192	49.20	9.80	6.00	5.17	4.00	19.90	12.20	10.50	8.13

Table 4.6 Variable Rate Non-linear Walk Model Statistics

Tile Configuration	Area Covered by tiles (in sq. ft)	Tile Trace Coverage (in percentage)	Kalman Trace Coverage (in percentage)				Trace Gain With Respect to Tile Trace Coverage (in percentage)			
			Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Half ft. accuracy	>Half ft. & < One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.
24 Tiles	96	36.55	6.80	3.35	4.60	3.90	18.60	9.15	12.60	10.67
28 Tiles	112	36.70	5.40	3.55	4.75	3.70	14.70	9.67	12.95	10.10
32 Tiles	128	38.70	5.50	3.95	4.80	4.90	14.20	10.20	12.40	12.65
36 Tiles	144	40.90	6.50	4.50	5.40	4.30	15.90	11.00	13.20	10.50
40 Tiles	160	48.00	7.30	4.40	4.20	3.55	15.20	9.15	8.75	7.40
48 Tiles	192	51.70	9.25	5.65	4.75	2.90	17.90	10.90	9.18	5.60

4.2.2 Tile Saving Performance

The Kalman filter helps to estimate the location of the person. It is observed that by incorporating the Kalman filter location estimations, considerable savings in term of number of tiles can be achieved. In addition to the simulation models and the number of tiles being used for simulation, the results described in Table 4.3 to Table 4.6 are dependent upon the manner in which the tiles are laid upon the floor. Therefore, the savings in terms of number of tiles (as described in Table 4.7) is an estimate that may fluctuate with the change in the tiles layout design. The savings in terms of number of tiles is deduced by using simple linear mathematics as:

Savings in term of number of tiles = |*(No. of Tiles in Tile Configuration/Tile Trace Coverage)* Kalman Trace Coverage*|

From Table 4.7, it is observed that for location estimations with precision within 1 ft. of actual location, the savings in tiles increases with the increase in number of tiles being used for the simulation. For the least complex model (i.e. *Constant Rate Linear Walk Model*) the savings in terms of number of tiles ranges from 13 to 25 tiles. For *Variable*

Rate Non-linear Walk Model the savings in terms of number of tiles ranges from 11 to 21 tiles. For non-linear simulation models, the savings in terms of number of tiles remain almost similar (ranging from 7 tiles to 15 tiles). The number of tiles being reduced ranges between 7 and 25 tiles for the worst case and the best case scenarios respectively. For location estimations with divergence greater than 2 ft. from person's actual position (i.e. error in location estimation), is minimal and remains somewhat constant for floor patterns.

Considering the cost of a tile to be approximately $3,000, savings in terms of cost ranges from $21,000 to $75,000 for floor designs that ideally costs approximately $72,000 to $144,000. The RFID system cost is generally low. For the RFID test setup, the system cost was around $3,000 which is relatively low in comparison to the overall savings.

Table 4.7 Savings In Terms Of Number Of Tiles

Tile Configuration	Savings in Term of Number of Tiles |(No. of Tiles in Tile Configuration/Tile Trace Coverage)* Kalman Trace Coverage|											
	Constant Rate Linear Walk Model			Variable Rate Linear Walk Model			Constant Rate Non-linear Walk Model			Variable Rate Non-linear Walk Model		
	Up to One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Up to One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Up to One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.	Up to One ft. accuracy	>One ft. & <Two ft. accuracy	> Two ft.
24 Tiles	13	03	02	11	03	02	07	04	04	07	03	03
28 Tiles	15	03	02	13	03	02	08	05	03	07	04	03
32 Tiles	15	03	02	15	04	02	09	05	03	08	04	04
36 Tiles	19	04	03	16	05	03	13	05	03	10	05	04
40 Tiles	20	03	01	18	04	02	14	04	02	10	03	03
48 Tiles	25	03	01	21	04	01	15	05	04	14	04	03

Figures 4.2 to 4.7 compare the savings in terms of number of tiles for each of the floor plans. As expected, the saving increases with the increase in number of tiles and decreases with the increase in the complexity of the motion.

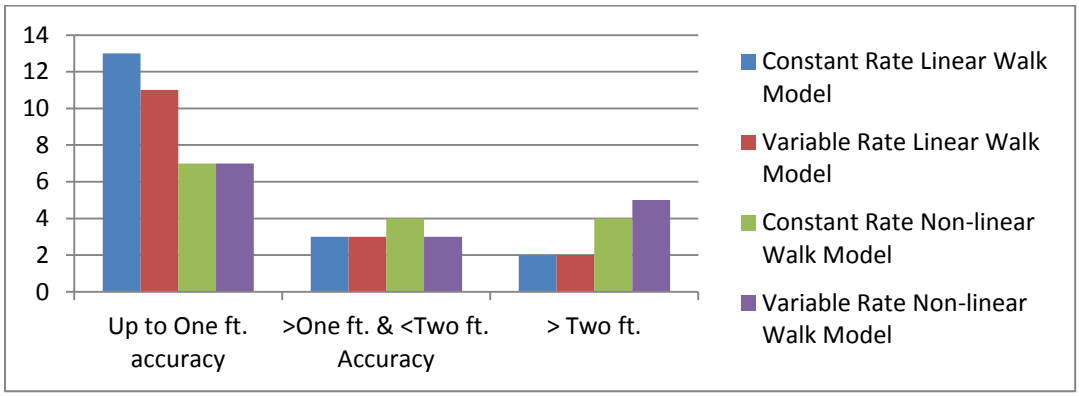

Figure 4.2 Savings Comparison of 24 Tiles Floor Plan

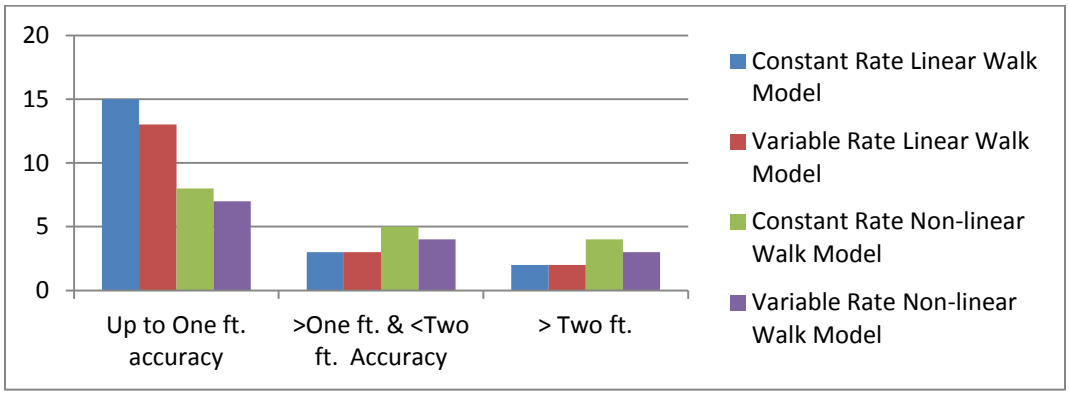

Figure 4.3 Savings Comparison of 28 Tiles Floor Plan

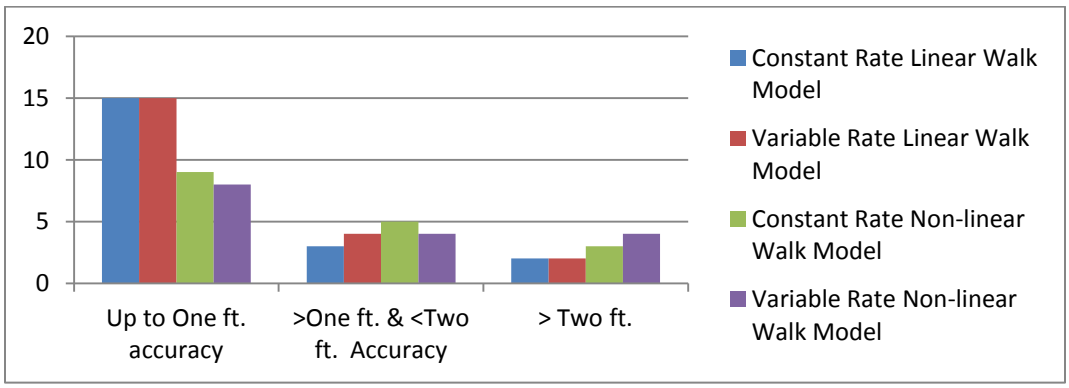

Figure 4.4 Savings Comparison of 32 Tiles Floor Plan

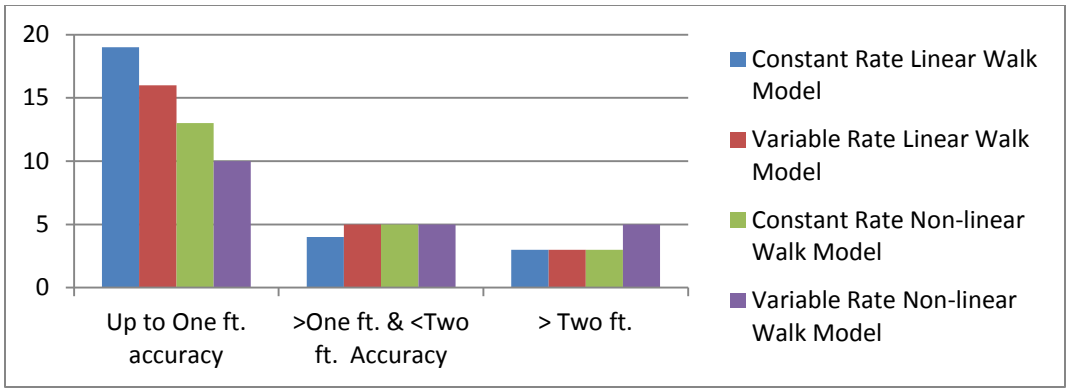

Figure 4.5 Savings Comparison of 36 Tiles Floor Plan

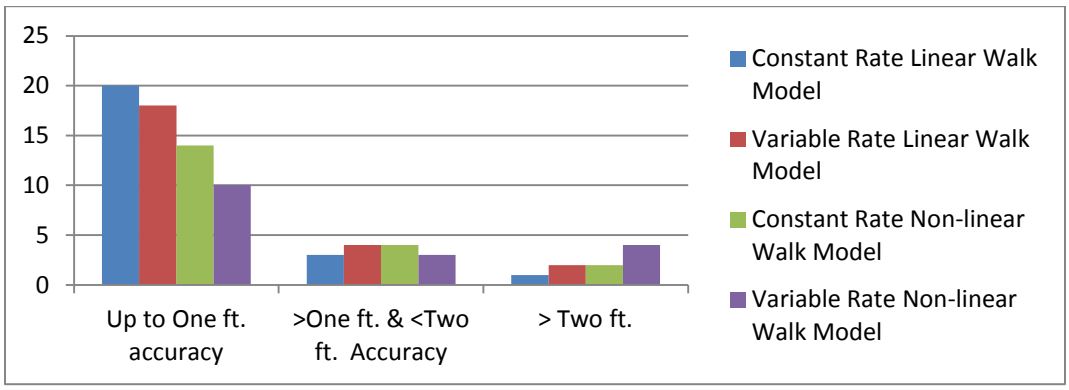

Figure 4.6 Savings Comparison of 40 Tiles Floor Plan

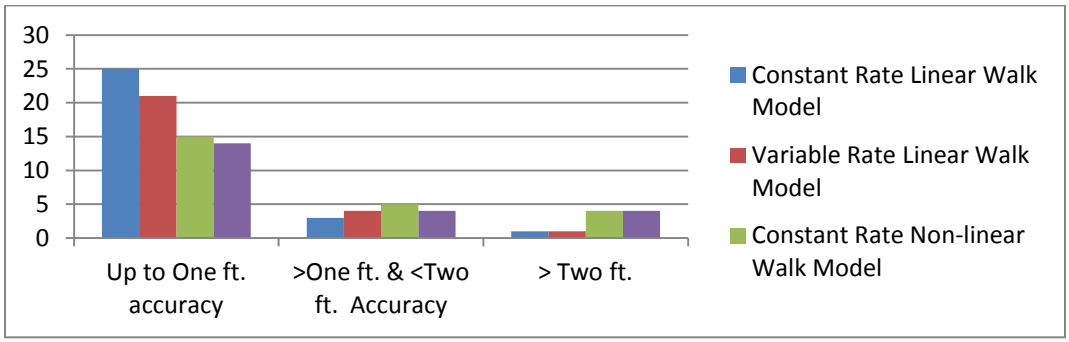

Figure 4.7 Savings Comparison of 48 Tiles Floor Plan

CHAPTER 5 CONCLUSION AND FUTURE WORK

This chapter concludes the thesis by summarizing the design of the high-precision indoor tracking system and the performance of the proposed system. The future work that is worth exploring in the future is also included.

5.1 CONCLUSION

In our research, we propose a high precision indoor tracking system which utilizes the location information from the Stepscan^{TM} tiles to track and locate a person in an indoor environment floor with the assistance of the Discrete Kalman filter. The RFID zonal readings and RSSI values are used to improve the Kalman estimations. The proposed system tracks the person precisely when the location information can be obtained through the tiles and attempts to estimate the person's position whenever the person's location is not available (i.e. the person is not stepping on any tile during the timeslot).

In order to evaluate the effectiveness of the proposed system, extensive simulations were carried out using java and Matlab. Our simulations include the precise models of the Stepscan^{TM} tiles, RFID system and Discrete Kalman filter. Specifically, the tiles provide the location data in terms of Cartesian coordinates whenever the user steps on the tiles placed onto 20x20 ft^2 simulation area. The placement of the tiles on the floor yields floor patterns described in Table 4.2. The RFID system provides the zonal information and the RSS level data whenever the user is in the RFID interrogation zones. The entire floor is divided into four RFID zones, each of which has a radius of 5 ft. as described in Section 3.3. Furthermore, each zone is divided into subzones based upon the RFID RSSI values as mentioned in Section 4.1. As each user can be uniquely identified using the RFID tag id, an additional benefit of multiple user tracking is provided by RFID system making the system scalable. The Discrete Kalman filter tracks and estimates the location of a user on an area of 20x20 ft^2 using the data from the tiles and the RFID system.

In order to simulate the movement of a user on the floor, we designed four mobility models based on the *Random Waypoint model: Constant Rate Linear Walk*

Model, *Variable Rate Linear Walk Model, Constant Rate Non-linear Walk Model* and *Variable Rate Non-linear Walk Model*. For simulation purposes, each mobility model was integrated with six different floor patterns described in Section 4.1. The simulation results detail the effectiveness of the proposed system. Specifically, the proposes system not only tracks the person precisely while he/she steps on a tile, but also estimate his/her location when the location information cannot be obtained through the tiles.

Our simulation results indicate that the proposed system increases the capability to track and locate a person by at least 24% (more than 50% in some cases), with errors ranging from 2.5% to 15%. Furthermore, the proposed system helps to reduce the cost of indoor tracking significantly. In terms of the number of Stepscan^{TM} tiles deployed in the system, a reduction of 7 to 25 tiles can be achieved in the scenarios under investigation. In terms of monetary cost, $21,000 to $75,000 can be saved for an indoor tracking system considered in our research.

5.2 FUTURE WORK

A future direction based upon the proposed approach is to extend the system design to include doorways, corridors and staircases. Even though both the RFID system and the Stepscan^{TM} tiles are capable of identifying, tracking and locating multiple persons at the same time, the simulation models described are used for single person simulations. As the system accuracy for multiple person tracking is not known, understanding and evaluation of both the RFID system and the Stepscan^{TM} tiles for multiple person tracking is necessary to simulate multiple user behavior. The assumption of uniform distribution of space in the simulation models may not stand correct for real world scenarios, and hence, simulation models giving weights to space within the indoor environment where the probability of the user being present is more should be considered.

The current approach of placement of the tags on each of the shoulder by stitching them to the user clothes may not feasible for certain indoor environments and hence an alternative to the current tag placement design is required. As the efficiency of the proposed system is influenced by the RSS values, more rigorous RSS screening should be done in order to reduce the deviation of the estimated location from the actual location of

the person. It will be interesting to deploy the proposed system in real tracking scenarios and compare the observed results with the simulation results.

BIBLIOGRAPHY

[1] Balachandran, W.; Cecelja, F.; Ptasinski, P., "A GPS based navigation aid for the blind," *Applied Electromagnetics and Communications, 2003. ICECom 2003. 17th International Conference on* , vol., no., pp.34,36, 1-3 Oct. 2003

[2] Blenkhorn, P.; Evans, D.G., "A system for enabling blind people to identify landmarks: the sound buoy," *Rehabilitation Engineering, IEEE Transactions on* , vol.5, no.3, pp.276,278, Sep 1997

[3] Guolin Sun; Jie Chen; Wei Guo; Liu, K. J R, "Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs," *Signal Processing Magazine, IEEE* , vol.22, no.4, pp.12,23, July 2005

[4] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 40, no. 1, pp. 91.102, January 1992.

[5] Bahl, P.; Padmanabhan, V.N., "RADAR: an in-building RF-based user location and tracking system," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* , vol.2, no., pp.775,784 vol.2, 2000

[6] N. Priyantha, A. Charraborty, and H. Balakrishnan, "The Cricket location-support system," In *Proceedings of the 6th annual international conference on Mobile computing and networking* (MobiCom '00), July 2000.

[7] Besada, J.A.; Bernardos, A.M.; Tarrio, P.; Casar, J.R., "Analysis of tracking methods for wireless indoor localization," *Wireless Pervasive Computing, 2007. ISWPC '07. 2nd International Symposium on* , vol., no., pp.,, 5-7 Feb. 2007.

[8] Ran, L.; Helal, S.; Moore, S., "Drishti: an integrated indoor/outdoor blind navigation system and service," *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on* , vol., no., pp.23,30, 14-17 March 2004

[9] Ertan, S.; Lee, C.; Willets, A.; Tan, H.; Pentland, A., "A wearable haptic navigation guidance system," *Wearable Computers, 1998. Digest of Papers. Second International Symposium on* , vol., no., pp.164,165, 19-20 Oct. 1998

[10] Paul, Anindya S.; Wan, Eric A., "Wi-Fi based indoor localization and tracking using sigma-point Kalman filtering methods," *Position, Location and Navigation Symposium, 2008 IEEE/ION* , vol., no., pp.646,659, 5-8 May 2008

[11] D. Simon, "Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches," *1st ed. Wiley-Interscience*, 2006.

[12] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Technical Report. University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[13] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE pervasive computing*, 2003.

[14] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 425 - 435, 2002.

[15] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitious computing: A case study," in *Proceedings of International Conference on Ubiquitious Computing (UBICOMP)*, 2004.

[16] Stepscan, "Stepscan," 16 May 2012. [Online]. Available: http://stepscan.com/index.php?page=home [Accessed 15 April 2013].

[17] Kalman R. E., ”A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, Vol 82, No 1, pp 35 – 46, 1960.

[18] Rodrigues, M.L.; Vieira, L.F.M.; Campos, M. F M, "Fingerprinting-based radio localization in indoor environments using multiple wireless technologies," *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on* , vol., no., pp.1203,1207, 11-14 Sept. 2011

[19] Lei Yu; Laaraiedh, M.; Avrillon, S.; Uguen, B., "Fingerprinting localization based on neural networks and ultra-wideband signals," *Signal Processing and Information Technology (ISSPIT), 2011 IEEE International Symposium on* , vol., no., pp.184,189, 14-17 Dec. 2011

[20] Shih-Hau Fang; Tsung-Nan Lin, "Indoor Location System Based on Discriminant-Adaptive Neural Network in IEEE 802.11 Environments," *Neural Networks, IEEE Transactions on* , vol.19, no.11, pp.1973,1978, Nov. 2008

[21] Jin, Y.; Soh, W.; Motani, M.; Wong, W., "A Robust Indoor Pedestrian Tracking System with Sparse Infrastructure Support," *Mobile Computing, IEEE Transactions on* , vol.PP, no.99, pp.1,1, 0

[22] Y. K. Thong, M. S. Woolfson, J. A. Crowe, B. R. Hayes-Gill and R. E. Challis, “Dependence of inertial measurements of distance on accelerometer noise,” *Measurement Science and Technology*, vol. 13, no. 8, pp. 1163-1172, 2002.

[23] Lei Fang; Antsaklis, P.J.; Montestruque, L.A.; McMickell, M.B.; Lemmon, M.; Yashan Sun; Hui Fang; Koutroulis, I.; Haenggi, M.; Min Xie; Xiaojuan Xie, "Design of a wireless assisted pedestrian dead reckoning system - the NavMote experience," *Instrumentation and Measurement, IEEE Transactions on* , vol.54, no.6, pp.2342,2358, Dec. 2005

[24] Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," *in Proc. International Technical Meeting of the Satellite Division of The Institute of Navigation* (ION-GPS), pp. 1735-1740, 2000

[25] Sagawa, K.; Inooka, H.; Satoh, Y., "Non-restricted measurement of walking distance," *Systems, Man, and Cybernetics, 2000 IEEE International Conference on* , vol.3, no., pp.1847,1852 vol.3, 2000

[26] Yan Wu; Mugler, D.H., "A robust DSP integrator for accelerometer signals," *Biomedical Engineering, IEEE Transactions on* , vol.51, no.2, pp.385,389, Feb. 2004

[27] Krach, B.; Roberston, P., "Cascaded estimation architecture for integration of foot-mounted inertial sensors," *Position, Location and Navigation Symposium, 2008 IEEE/ION* , vol., no., pp.112,119, 5-8 May 2008

[28] Yanying Gu; Lo, A.; Niemegeers, I., "A survey of indoor positioning systems for wireless personal networks," *Communications Surveys & Tutorials, IEEE* , vol.11, no.1, pp.13,32, First Quarter 2009

[29] Hui Liu; Darabi, H.; Banerjee, P.; Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* , vol.37, no.6, pp.1067,1080, Nov. 2007

[30] Da Zhang; Feng Xia; Zhuo Yang; Lin Yao; Wenhong Zhao, "Localization Technologies for Indoor Human Tracking," *Future Information Technology (FutureTech), 2010 5th International Conference on* , vol., no., pp.1,6, 21-23 May 2010

[31] Wikipedia, "Radio-frequency identification," 31 March 2013 . [Online]. Available: http http://en.wikipedia.org/wiki/Radio-frequency_identification. [Accessed 12 March 2013]

[32] D. Sen, P. Sen, and A. M. Das, “RFID For Energy and Utility Industries,” *PennWell*, 2009, pp. 1–48

[33] S. A. Weis, “RFID (Radio Frequency Identification): Principles and Applications” Available: www.eecs.harvard.edu/rfid-article.pdf [14 March, 2013].

[34] Wen Yao; Chao-Hsien Chu; Zang Li, "The use of RFID in healthcare: Benefits and barriers," *RFID-Technology and Applications (RFID-TA), 2010 IEEE International Conference on* , vol., no., pp.128,134, 17-19 June 2010

[35] Cully, W. P L; Cotton, S.L.; Scanlon, W.G.; McQuiston, J.B., "Localization algorithm performance in ultra low power active RFID based patient tracking," *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on* , vol., no., pp.2158,2162, 11-14 Sept. 2011

[36] Ming-Hua Tsai; Chieh-Ling Huang; Pau-Choo Chung; Yen-Kuang Yang; Yu-Chia Hsu; Shu-Ling Hsiao, "A psychiatric patients tracking system," *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on* , vol., no., pp.4 pp.,4053, 21-24 May 2006

[37] Gentili, G.B.; Dori, F.; Iadanza, E., "Dual-Frequency Active RFID Solution for Tracking Patients in a Children's Hospital. Design Method, Test Procedure, Risk Analysis, and Technical Solution," *Proceedings of the IEEE* , vol.98, no.9, pp.1656,1662, Sept. 2010

[38] Kim, Do-Sung; Kim, Jungchae; Kim, Seung-Ho; Yoo, S.K., "Design of RFID based the Patient Management and Tracking System in hospital," *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* , vol., no., pp.1459,1461, 20-25 Aug. 2008

[39] Yuh-Wen Chen; Guan-Jie Wang; Tzung-Hung Li; Tzung-Min Yang; Ting-Wei Shiu; Meng-Shian Tasi; Ya-Wen Jeng; Chao-Wen Chen, "A RFID model of transferring and tracking trauma patients after a large disaster,"*Service Operations, Logistics and Informatics, 2009. SOLI '09. IEEE/INFORMS International Conference on* , vol., no., pp.98,101, 22-24 July 2009

[40] E. Fry, L Lenert, “MASCAL - RFID tracking of patients, staff and equipment to enhance hospital response to mass casualty events”, *Amia Symposium, Bethesda (USA), American Medical Informatics Association*, 2005.

[41] J. Halamka(2007, January), "Early experiences with positive patient identification," *Journal of Healthcare Information Management*, [Online] 20(1):25–27. Available: http://mthink.com/article/early-experiences-positive-patient-identification [25 March, 2013]

[42] A. Aguilar, W. vander Putten, and F. Kirrane, "Positive patient identification using RFID and wireless networks," *InHISI 11th Annual Conference and Scientific Symposium, Dublin, Ireland*, November 2006.

[43] Jeong, B.H.; Cheng, C.Y.; Prabhu, V.; Yu, B. J., "An RFID Application Model For Surgery Patient Identification," *Advanced Management of Information for Globalized Enterprises, 2008. AMIGE 2008. IEEE Symposium on* , vol., no., pp.1,3, 28-29 Sept. 2008

[44] J. Bardram, "Applications of context-aware computing in hospital work: examples and design principles," *In ACM SAC*, pages 1574–1579, 2004.

[45] D. Joseph and R. Silvano, "Using RFID technologies to reduce blood transfusion errors," *White Paper by Intel Corporation, Autentica, Cisco Systems, and San Raffaele Hospital,* 2005.

[46] Wikipedia, "Filtering problem (stochastic processes)," 12 October 2012. [Online]. Available: http://en.wikipedia.org/wiki/Filtering_problem_(stochastic_processes) [Accessed 16 march 2013].

[47] Wikipedia, "LTI system theory," 26 February 2013. [Online]. Available: http://en.wikipedia.org/wiki/LTI_system_theory [Accessed 16 march 2013].

[48] Paulo S. R, Diniz, "Adaptive Filtering: Algorithms and Practical Implementation," *Springer, softcover reprint of hardcover 3rd ed*. 2008 edition.

[49] Wio, S. Horacio, Deza, R. Roberto & Lopez, M. Juan, "An Introduction to Stochastic Processes and Nonequilibrium Statistical Physics," *World Scientific Publishing*. ISBN 978-981-4374-78-1, 2012

[50] Ma Di; Er Meng Joo; Lim Hock Beng, "A comprehensive study of Kalman filter and extended Kalman filter for target tracking in Wireless Sensor Networks," *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on* , vol., no., pp.2792,2797, 12-15 Oct. 2008

[51] Costa, P.J., "Adaptive model architecture and extended Kalman-Bucy filters," *Aerospace and Electronic Systems, IEEE Transactions on* , vol.30, no.2, pp.525,533, Apr 1994

[52] Farina, A.; Ristic, B.; Benvenuti, D., "Tracking a ballistic target: comparison of several nonlinear filters," *Aerospace and Electronic Systems, IEEE Transactions on* , vol.38, no.3, pp.854,867, Jul 2002

[53] Eli Brookner, "Tracking and Kalman Filtering Made Easy," *Wiley-Interscience*, April 1998

[54] Mayiatis, D. E., "Comparison of an Alpha-Beta and Kalman Filter in Track While Scan Radars," *NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Diplomarbeit*, 1976. http://www.stealthskater.com/Documents/Radar_02.pdf [21 March, 2013]

[55] Wikipedia, "Alpha beta filter," 6 February 2013. [Online]. Available: http://en.wikipedia.org/wiki/Alpha_beta_filter [Accessed 22 march 2013].

[56] M. Isabel Ribeiro and P. Lima, "Introduction to Kalman Filtering," *Instituto Superior Técnico / Instituto de Sistemas e Robótica*, 2008.

[57] Wikipedia, "Kalman Filter," 6 March 2013. [Online]. Available: http://en.wikipedia.org/wiki/Kalman_filter [Accessed 22 march 2013].

[58] M. I. Ribeiro, "Kalman and extended Kalman filters: Concept, derivation and properties," *Institute for Systems and Robotics,* 2004.

[59] C. K. Chui and G. Chen, "Kalman Filtering with Real-Time Applications," *Springer-Verlag New York, Inc.*, New York, NY, USA, (1987),

[60] Wikipedia, "Kalman Filter," 23 March 2013. [Online]. Available: http://en.wikipedia.org/wiki/Extended_Kalman_filter [Accessed 25 march 2013].

[61] A. S. Patkar, "Localization in Noisy Environment Using Extended Kalman Filter," *ProQuest*, 2007.

[62] D. Salmond, "Introduction to Particle Filters for Tracking and Guidance," *in Advances in Missile Guidance, Control, and Estimation*, vol. 20121297, B. White, Ed. CRC Press, 2012.

[63] Wikipedia, “Mobility model,” 3 July 2012. [Online]. Available: http://en.wikipedia.org/wiki/Mobility_model. [Accessed 14 march 2013].

[64] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, “A Survey on Mobility Models for Performance Analysis in Tactical Mobile Networks,” *Journal of Telecommunications and Information Technology (JTIT)*, vol. Vol. 2, pp. 54–61, 2008.

[65] Cavilla, A.L.; Baron, G.; Hart, T.E.; Litty, L.; de Lara, E., "Simplified simulation models for indoor MANET evaluation are not robust," *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on* , vol., no., pp.610,620, 4-7 Oct. 2004

[66] Aschenbruck, N.; Ernst, R.; Martini, P., "Indoor mobility modelling," *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* , vol., no., pp.1264,1269, 6-10 Dec. 2010

[67] Tai Suk Kim; Jae Kyun Kwon; Dan Keun Sung, "Mobility modeling and traffic analysis in three-dimensional high-rise building environments," *Vehicular Technology, IEEE Transactions on* , vol.49, no.5, pp.1633,1640, Sept. 2000

[68] F. Bai; A. Helmy, "A SURVEY OF MOBILITY MODELS in Wireless Adhoc Networks," *University of Southern California*, U.S.A, 2006.

[69] Wikipedia, “Waypoint,” 28 February 2013. [Online]. Available: http://en.wikipedia.org/wiki/Waypoint [Accessed 14 march 2013].

[70] Alshanyour, A.; Baroudi, U., "Random and realistic mobility models impact on the performance of bypass-AODV routing protocol," *Wireless Days, 2008. WD '08. 1st IFIP* , vol., no., pp.1,5, 24-27 Nov. 2008

[71] Wikipedia, “Levy flight,” 28 February 2013. [Online]. Available: http://en.wikipedia.org/wiki/L%C3%A9vy_flight [Accessed 15 march 2013].

[72] Injong Rhee; Minsu Shin; Seongik Hong; Kyunghan Lee; Song Chong, "On the Levy-Walk Nature of Human Mobility," *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* , vol., no., pp.924,932, 13-18 April 2008

[73] Injong Rhee; Minsu Shin; Seongik Hong; Kyunghan Lee; Seong Joon Kim; Song Chong, "On the Levy-Walk Nature of Human Mobility," *Networking, IEEE/ACM Transactions on* , vol.19, no.3, pp.630,643, June 2011

[74] Wikipedia, “Pathfinding,” 24 February 2013. [Online]. Available: http://en.wikipedia.org/wiki/Pathfinding [Accessed 15 march 2013].

[75] Tarjan, Robert, "Depth-first search and linear graph algorithms," *Switching and Automata Theory, 1971., 12th Annual Symposium on* , vol., no., pp.114,121, 13-15 Oct. 1971

[76] Khantanapoka, K.; Chinnasarn, K., "Pathfinding of 2D & 3D game real-time strategy with depth direction A∗ algorithm for multi-layer," *Natural Language Processing, 2009. SNLP '09. Eighth International Symposium on* , vol., no., pp.184,188, 20-22 Oct. 2009

[77] Wikipedia, “Breadth-first search,” 8 March 2013. [Online]. Available: http://en.wikipedia.org/wiki/Breadth-first_search [Accessed 15 march 2013].

[78] Wikipedia, “Depth-first search,” 9 March 2013. [Online]. Available: http://en.wikipedia.org/wiki/Depth-first_search [Accessed 15 march 2013].

[79] Dechter, R. & Pearl, J. (1985), "Generalized Best-First Search Strategies and the Optimality of A*," *Journal of the Association for Computing Machinery*, 32, 505-536.

[80] Wikipedia, “Artificial Intelligence/Search/Heuristic search/Best-first search,” 27 October 2012. [Online]. Available: http://en.wikibooks.org/wiki/Artificial_Intelligence/Search/Heuristic_search/Best-first_search. [Accessed 15 march 2013].

[81] Xiang Liu; Daoxiong Gong, "A comparative study of A-star algorithms for search and rescue in perfect maze," *Electric Information and Control Engineering (ICEICE), 2011 International Conference on* , vol., no., pp.24,27, 15-17 April 2011

[82] K. Nikhil, "Comparison of Efficiency in Pathfinding Algorithms in Game Development," (2009). Technical Reports. Paper 10.

[83] Zebra Technology Company, “Zebra Technology,” [Online]. Available: http://www.wherenet.com/ [Accessed 25 march 2013].

[84] T. King, S. Kopf, T. Haenselmann, C. Lubberger and W. Effelsberg, “COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses”, *ACM WiNTECH*, Los Angeles, USA, Sept. 2006, Pages: 34-40.

[85] Ekahau, “Ekahau,” 25 March 2013. [Online]. Available: http://www.ekahau.com/ [Accessed 25 march 2013].

[86] Ubisense, “Ubisense,” 28 March 2013. [Online]. Available: http://www.ubisense.net/en [Accessed 25 march 2013].

[87] M Kohler, "Using the kalman filter to track human interactive motion modelling and initialization of the kalman filter for translational motion, " 1997

[88] GAO RFID Inc. , "RFID Reader," 29 March 2013. [Online]. Available: http://www.gaotek.com/images/RFID/UHF%20Gen2%20RFID%20Reader%20with%204%20Ports.JPG [Accessed 29 march 2013].

[89] GAO RFID Inc., "RFID Reader," 20 June 2012. [Online]. http://www.gaorfid.com/index.php?main_page=product_info&cPath=128&products_id=859 [Accessed 29 march 2013].

[90] GAO RFID Inc. , "RFID Antenna," 29 March 2013. [Online]. Available: http://www.gaorfid.com/images/RFID-Antenna/Antenna%20Circular%20Polarization%20Indoor.JPG [Accessed 29 march 2013].

[91] GAO RFID Inc., "RFID Antenna," 20 June 2012. [Online]. http://www.gaorfid.com/index.php?main_page=product_info&cPath=90&products_id=626 [Accessed 29 march 2013].

[92] GAO RFID Inc. , "RFID Tags," 29 March 2013. [Online]. Available: http://www.gaorfid.com/images/RFID-Tags-902-to-928MHz-(UHF)/UHF%20RFID%20Clothing%20Tag.JPG [Accessed 29 march 2013].

[93] GAO RFID Inc., "RFID Tags," 20 March 2012. [Online]. http://www.gaorfid.com/index.php?main_page=product_info&cPath=136&products_id=981 [Accessed 29 March 2013].

[94] Wikipedia, “White noise,” 30 April 2013. [Online]. Available: http://en.wikipedia.org/wiki/White_noise [Accessed 30 April 2013].

[95] Wikipedia, “Preferred walking speed,” 23 April 2013. [Online]. Available: http://en.wikipedia.org/wiki/Preferred_walking_speed [Accessed 25 April 2013].

[96] John D. Kelleher, "Robot Path Smoothing," 23 April 2013. [Online]. Available: http://www.comp.dit.ie/jkelleher/rtf/classmaterial/week10/PathSmoothing.pdf [Accessed 25 April 2013].

APPENDIX A: RFID Antenna and Reader Configuration

Refer to SDK documentation (available with the hardware) for detailed information on RFID antenna and reader configuration.

```
<AntennaConfiguration>
<AntennaID>1</AntennaID>
<RFReceiver>
      <!--Set the antenna receiving sensitivity as per the
      current regulatory region. An Index setting of '1' =
      ReceiveSensitivityValue of '0' which = -80dBm. Up to index 42 =
      ReceiveSensitivityvalue of 50 = -30dBm (the highest, least
      sensitive setting)-->
      <ReceiverSensitivity>2</ReceiverSensitivity>
</RFReceiver>
<!-- Set the Antenna RF transmit power -->
<RFTransmitter>
      <!-- HopTableID value is same for all four antennas -->
      <HopTableID>1</HopTableID>
      <!-- ChannelIndex is same for all four antennas. Set the
      channel from those allowed in the current regulatory region -->
      <ChannelIndex>1</ChannelIndex>
      <!-- TransmitPower can be configured uniquely for each antenna,
      but we set it same for each antenna. Set the TransmitPower as per
      the region and frequency of operation. With Reader powered by
      adapter, the PowerIndex ranges from 1 to 91. 1 correspond to
      transmitPowerValue:1000(minimum) and 91 correspond to
      transmitPowerValue:3250(maximum. With each index value the
      transmitPowerValue is changed by 25-->
      <TransmitPower>1</TransmitPower>
</RFTransmitter>
<C1G2InventoryCommand>
      <TagInventoryStateAware>false</TagInventoryStateAware>
      <!-- C1G2RFControl controls the reader performance, set this
      parameter as per the environment and RFID usage -->
      <C1G2RFControl>
            <!--0: Max Throughput; 2:Dense Reader; 1000: AutoSet Must
            be same for all four antennas -->
            <ModeIndex>0</ModeIndex>
            <!-- Set Tari to false to enable ModeIndex-->
            <Tari>0</Tari>
      </C1G2RFControl>
      <C1G2SingulationControl>
            <!-- Session value is same for all four antennas -->
            <Session>2</Session>
            <!-- TagPopulation value is same for all four antennas -->
            <TagPopulation>32</TagPopulation>
            <!-- TagTransitTime is same for all four antennas. It
            measures of expected tag mobility in the field of view -->
            <TagTransitTime>0</TagTransitTime>
      </C1G2SingulationControl>
      <Impinj:ImpinjInventorySearchMode>
```

```
      <!--Need Dual-target for lots of low level data Low, for medium
      tag count or low-throughput applications where  repeated tag
      observation is desirable-->
      <InventorySearchMode>Dual_Target</InventorySearchMode>
      </Impinj:ImpinjInventorySearchMode>
      <Impinj:ImpinjLowDutyCycle>
            <!--Eanble LowDutyCycleMode as per the    current
            regulatory region -->
            <LowDutyCycleMode>Enabled</LowDutyCycleMode>
            <!-- EmptyFieldTimeout specifies in milliseconds the  time
            the Reader will wait before entering low duty cycle -->
            <EmptyFieldTimeout>10000</EmptyFieldTimeout>
            <!-- In low duty cycle mode, the Reader will rescan every
            FieldPingInterval milliseconds, checking for tags -->
            <FieldPingInterval>200</FieldPingInterval>
      </Impinj:ImpinjLowDutyCycle>
</C1G2InventoryCommand>
</AntennaConfiguration>
```

APPENDIX B: Tile Deployment Patterns

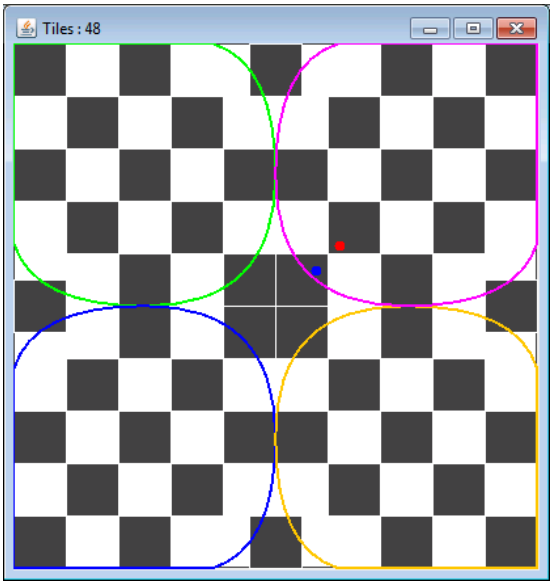

Figure 1 48 tiles pattern

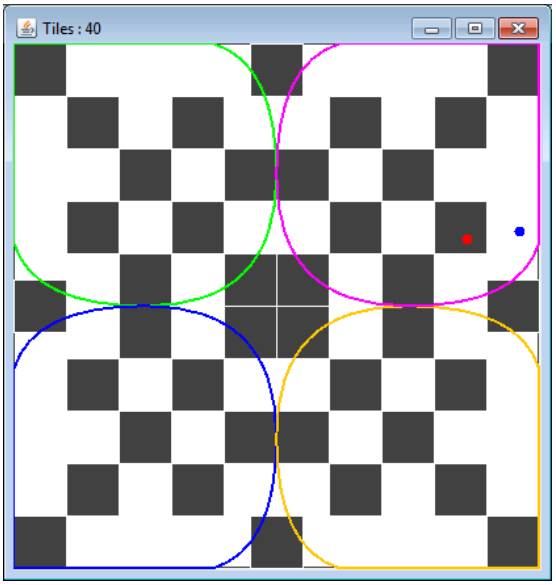

Figure 2 40 tiles pattern

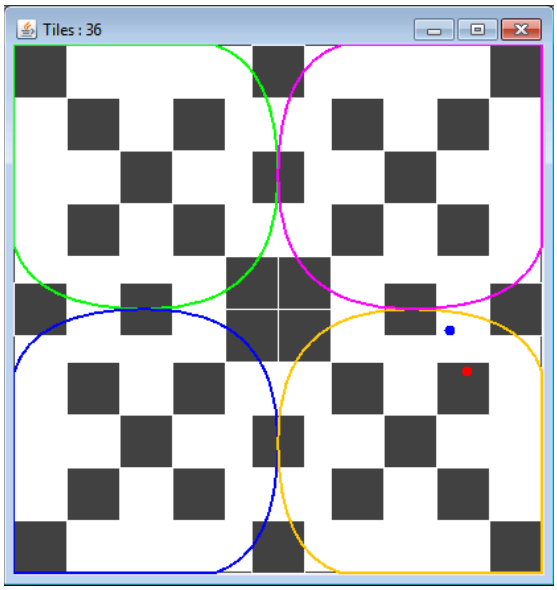

Figure 3 36 tiles pattern

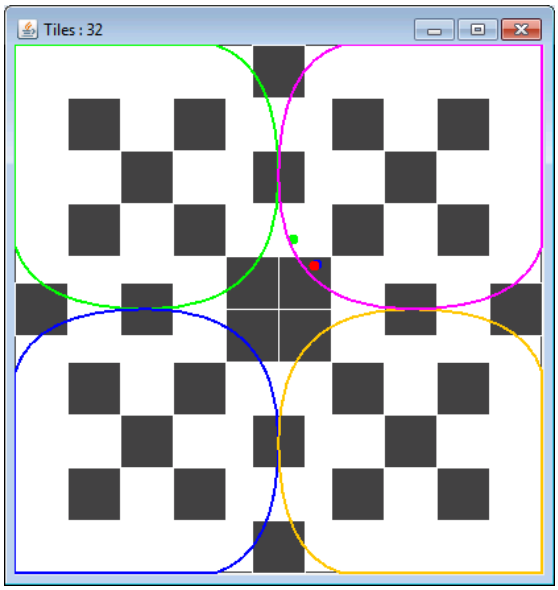

Figure 4 32 tiles pattern

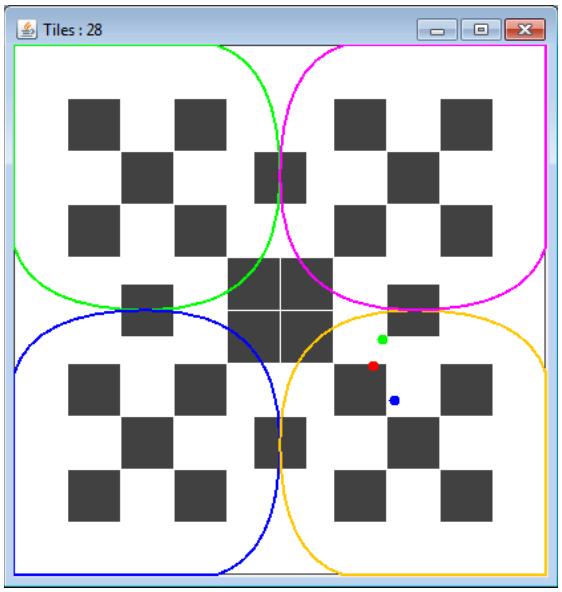

Figure 5 28 tiles pattern

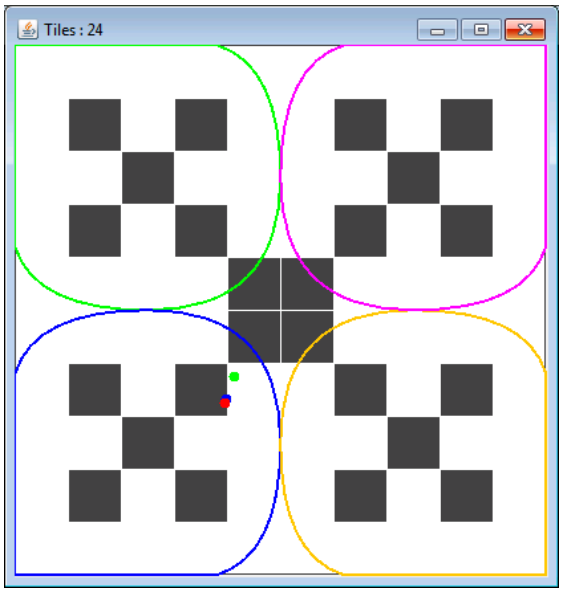

Figure 6 24 tiles pattern

APPENDIX C: Simulation Execution Results

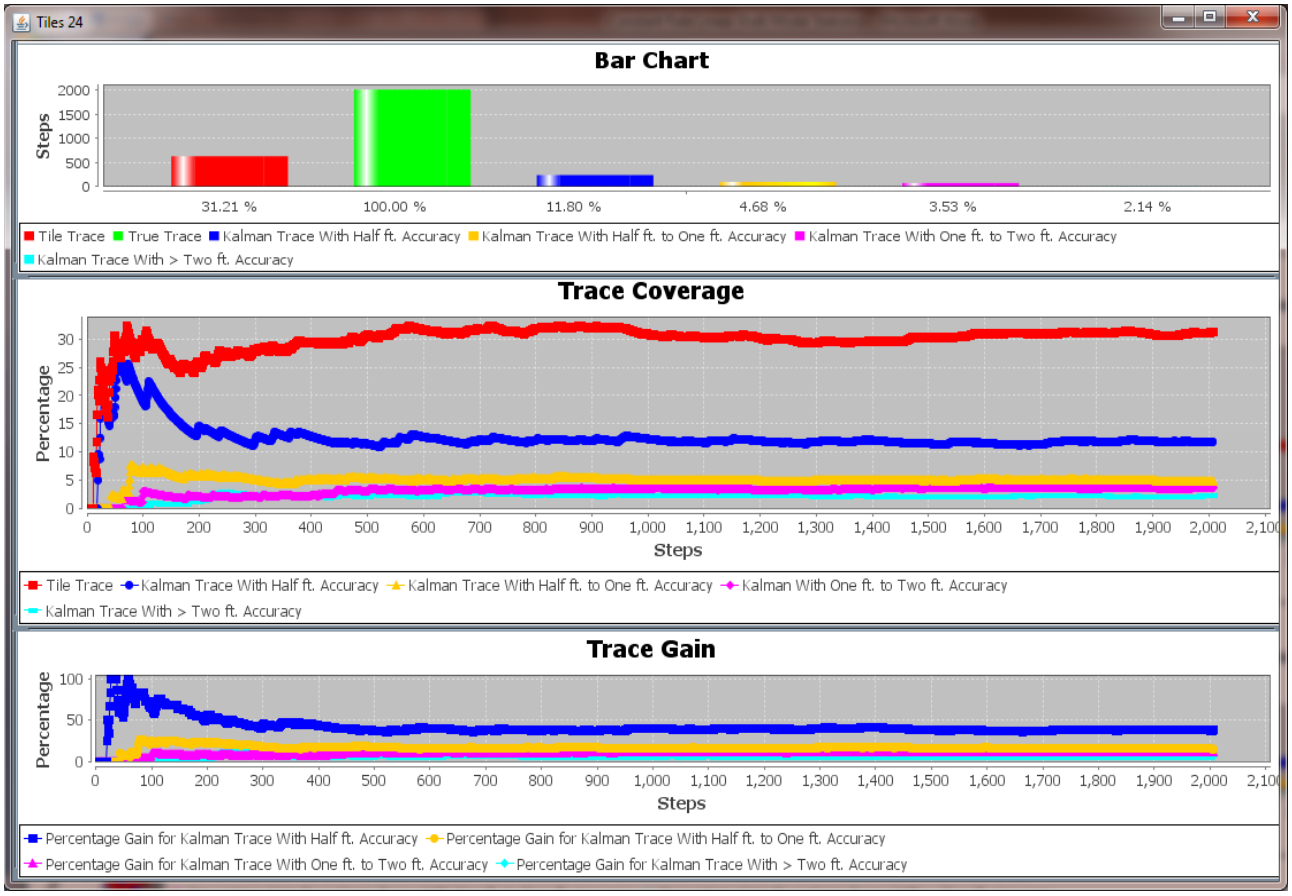

Figure 7 Constant Rate Linear Walk Model - 24 Tiles Floor Plan

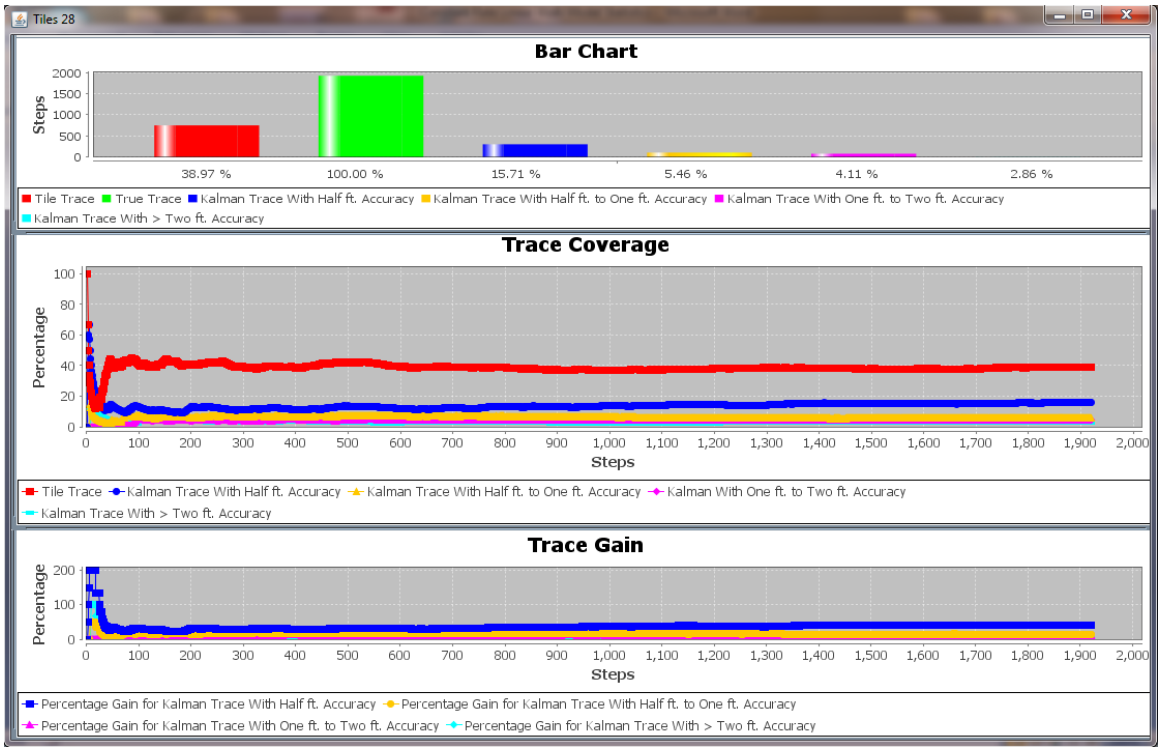

Figure 8 Constant Rate Linear Walk Model - 28 Tiles Floor Plan

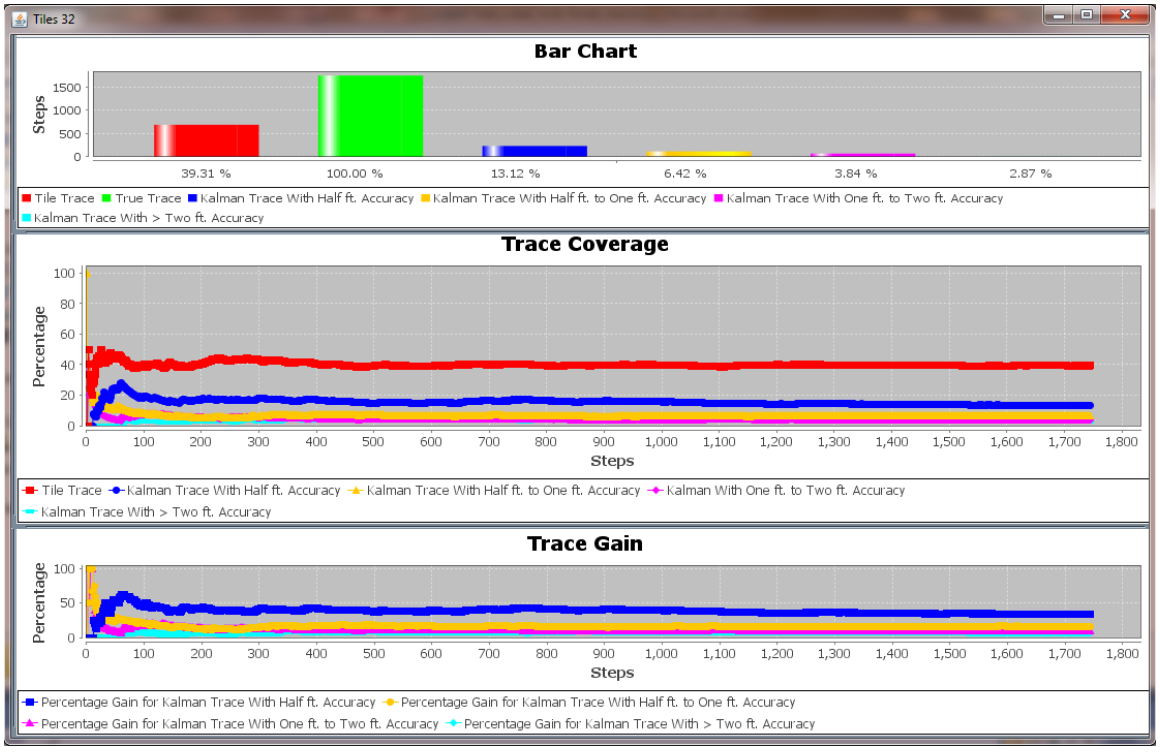

Figure 9 Constant Rate Linear Walk Model - 32 Tiles Floor Plan

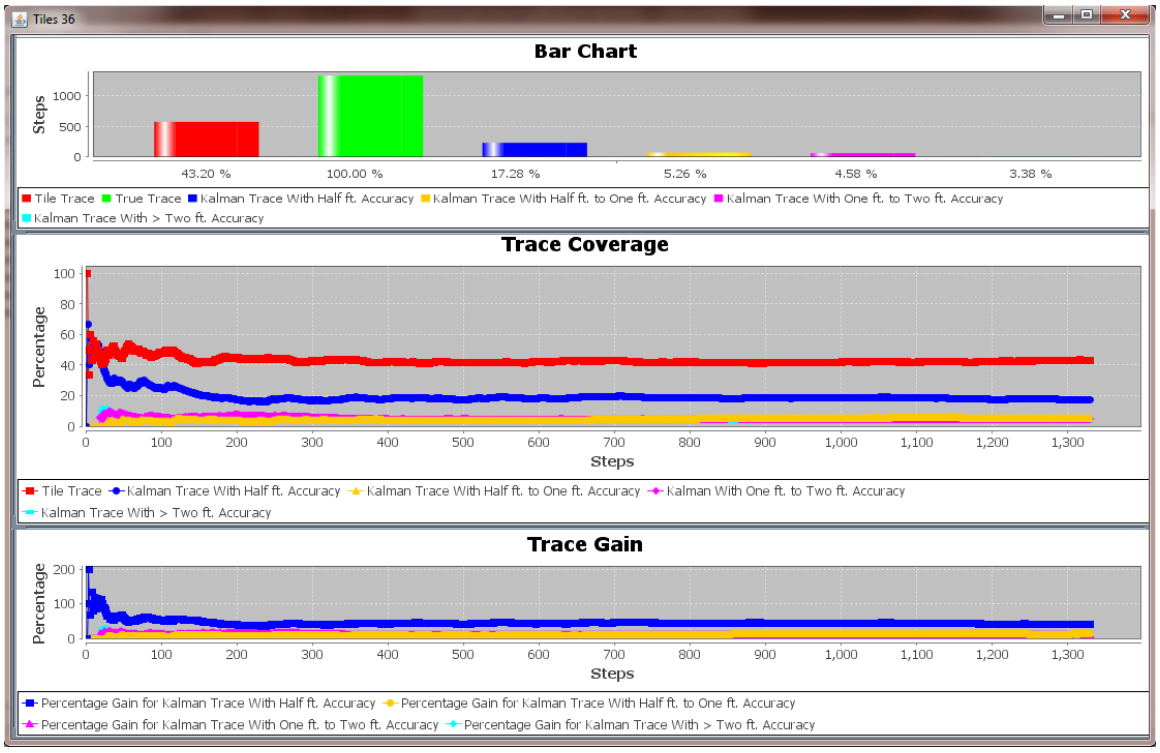

Figure 10 Constant Rate Linear Walk Model - 36 Tiles Floor Plan

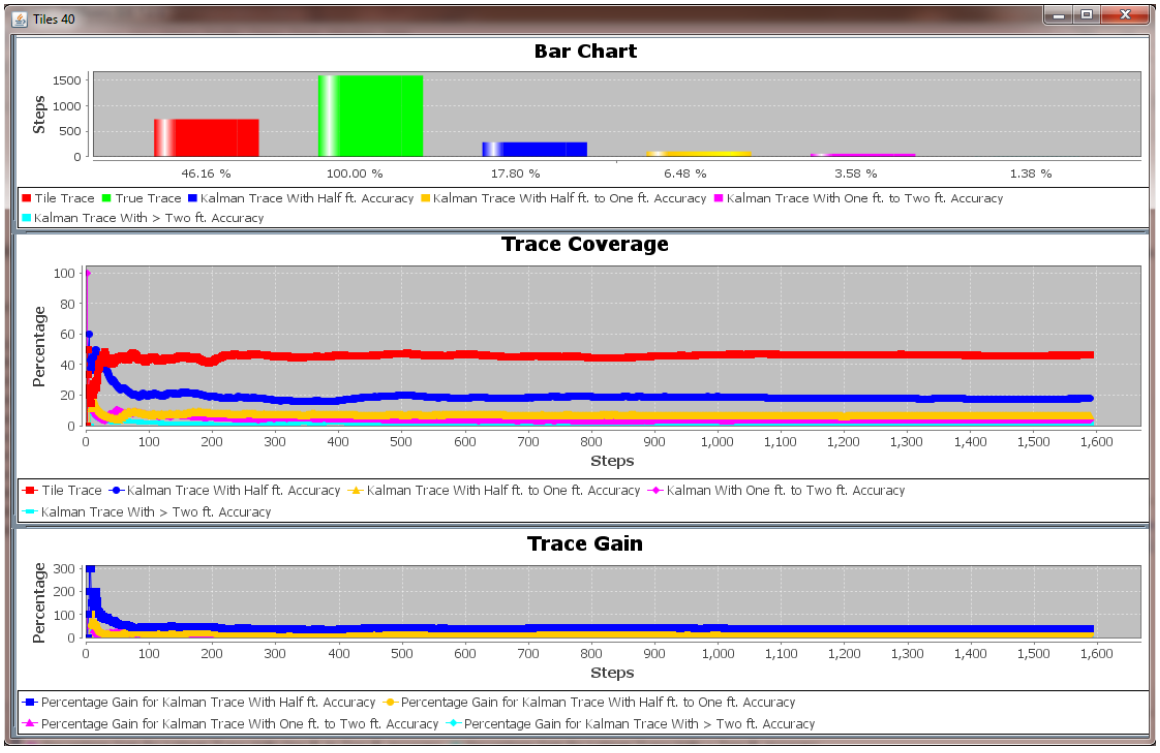

Figure 11 Constant Rate Linear Walk Model - 40 Tiles Floor Plan

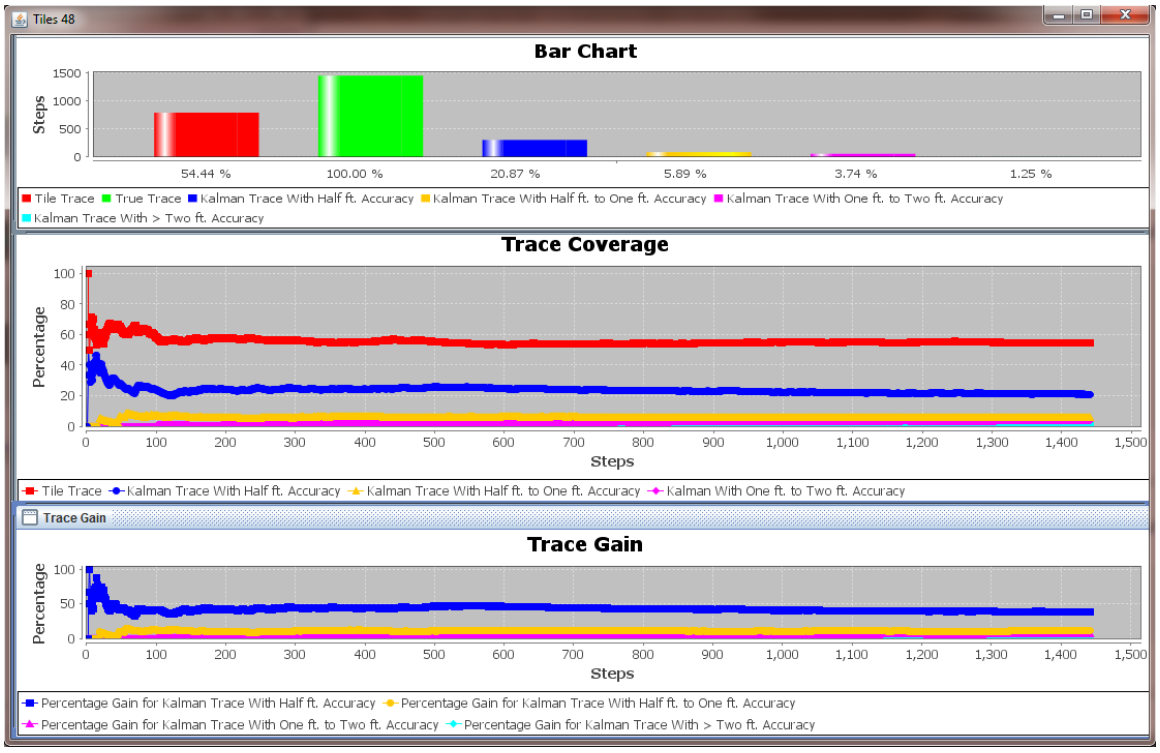

Figure 12 Constant Rate Linear Walk Model - 48 Tiles Floor Plan

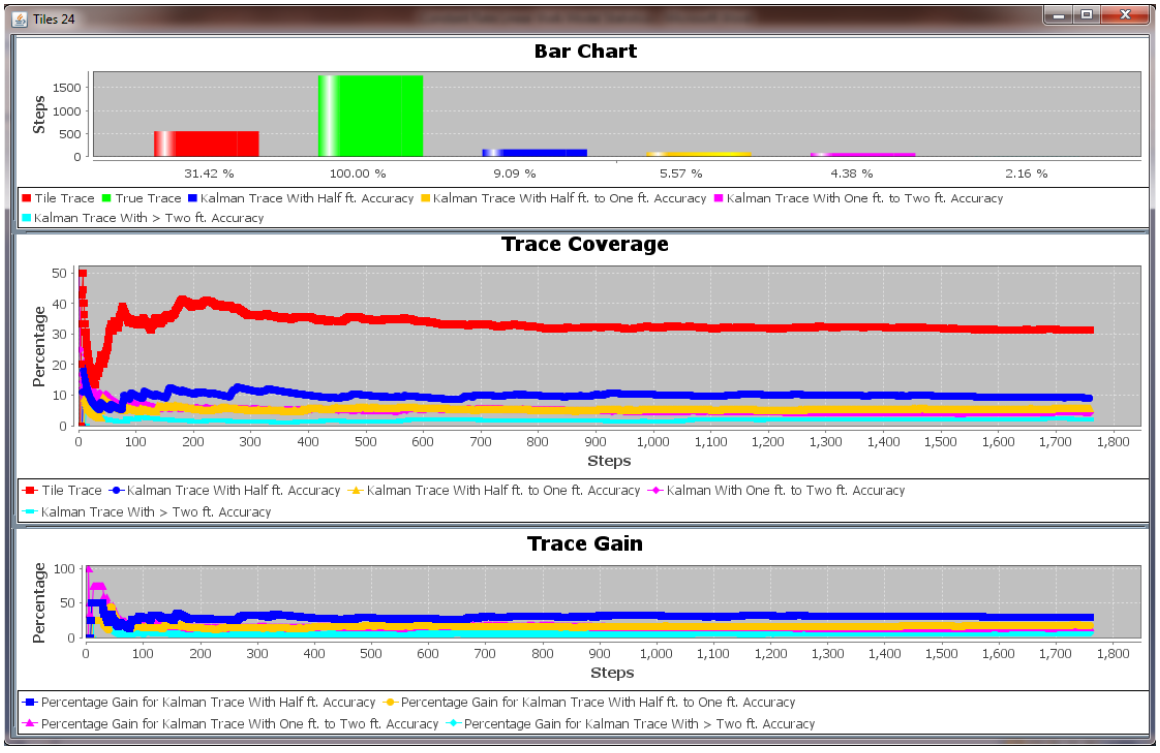

Figure 13 Variable Rate Linear Walk Model - 24 Tiles Floor Plan

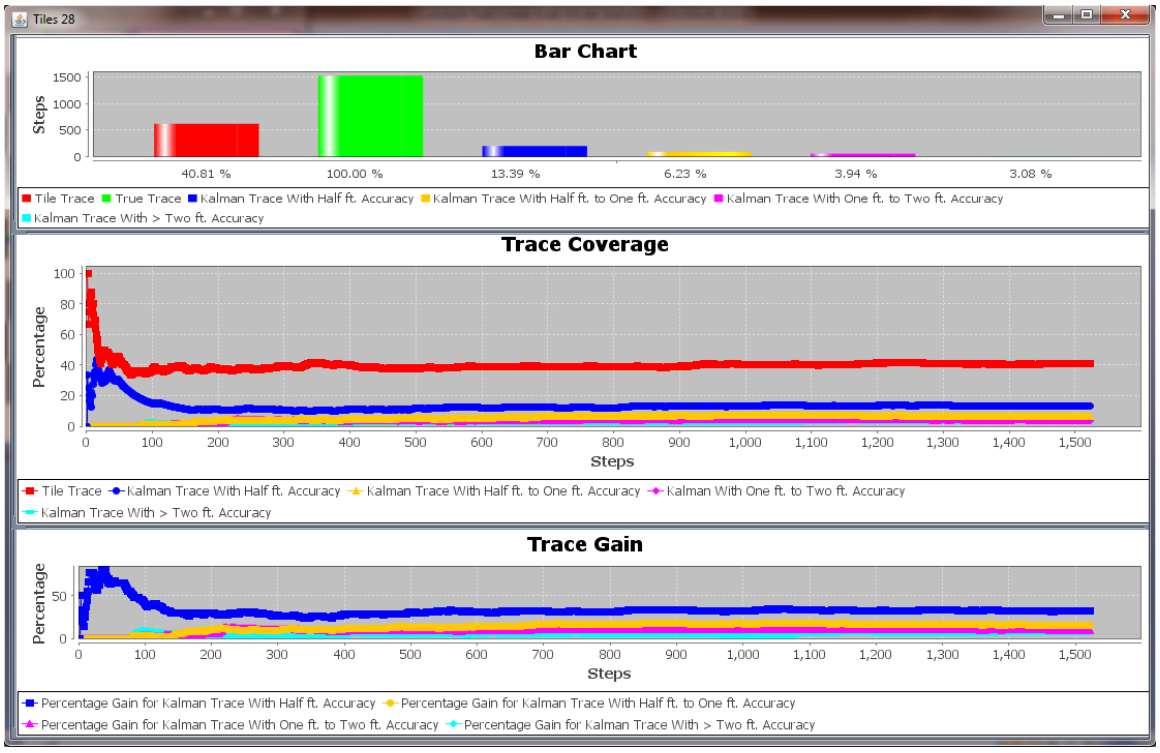

Figure 14 Variable Rate Linear Walk Model - 28 Tiles Floor Plan

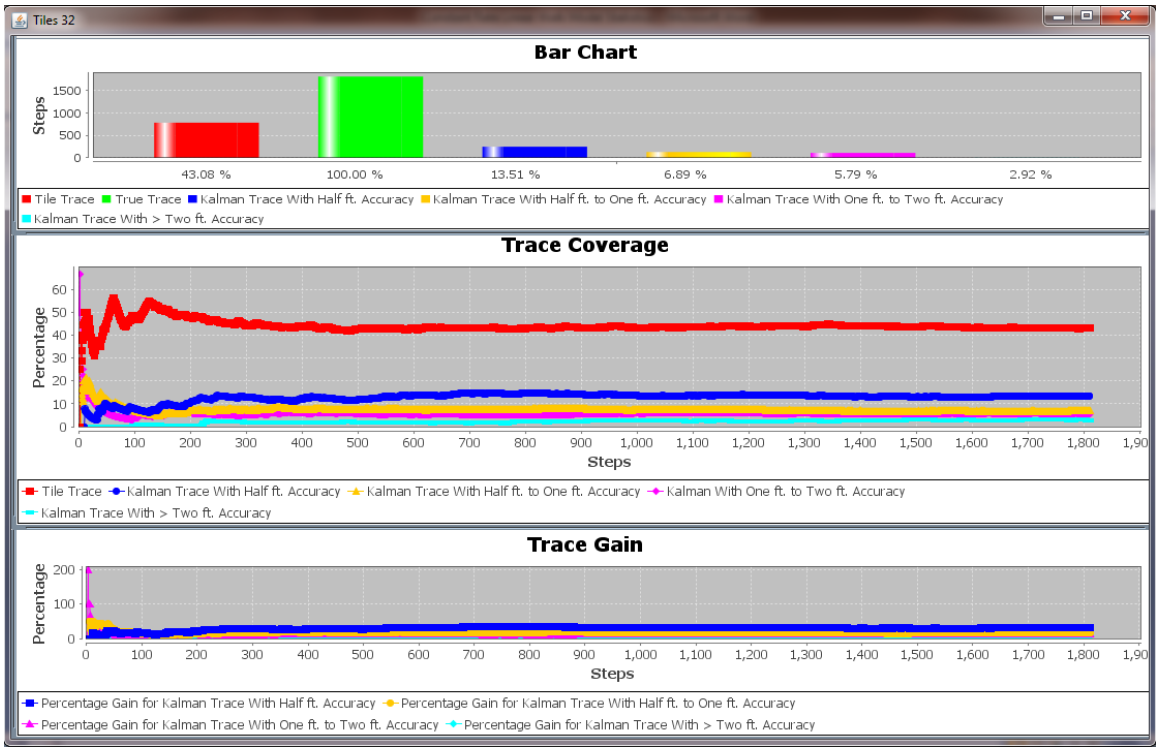

Figure 15 Variable Rate Linear Walk Model - 32 Tiles Floor Plan

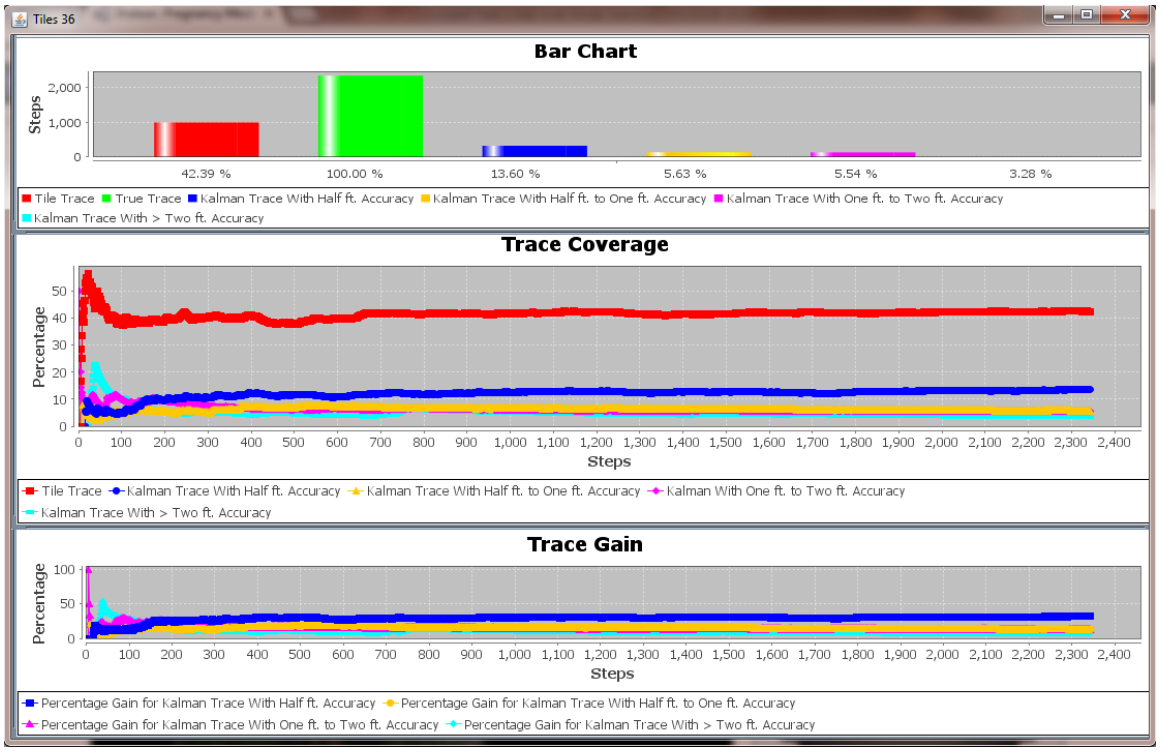

Figure 16 Variable Rate Linear Walk Model - 36 Tiles Floor Plan

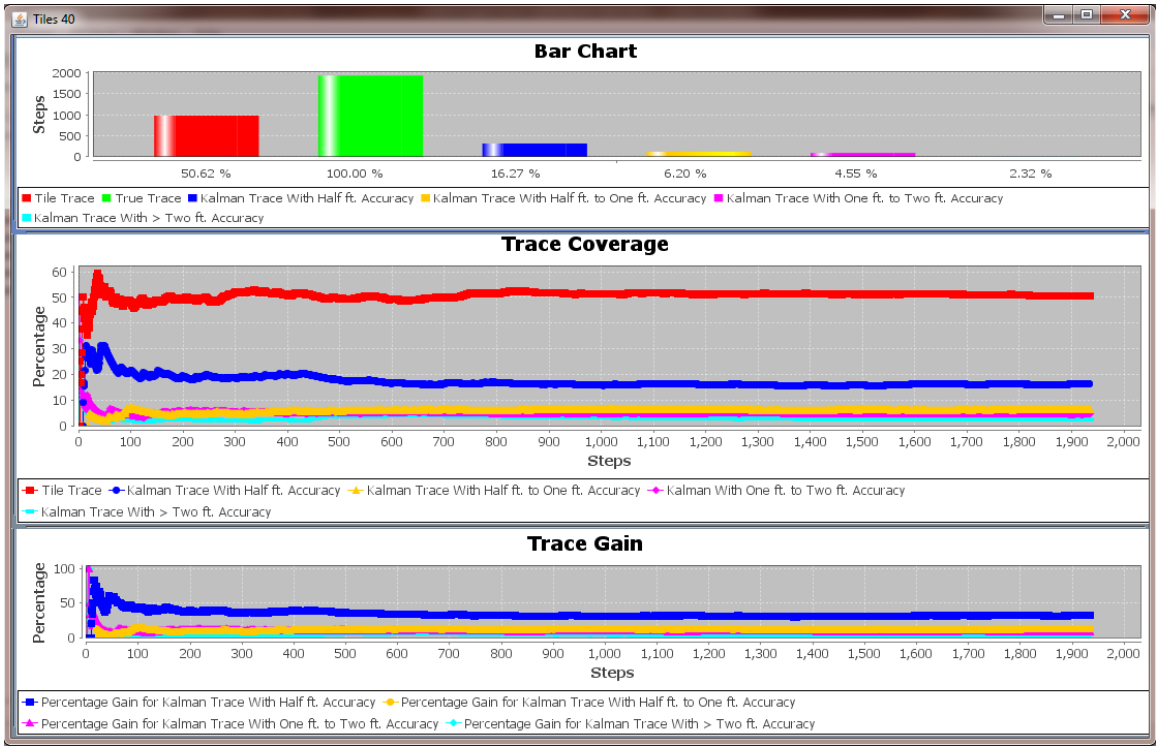

Figure 17 Variable Rate Linear Walk Model - 40 Tiles Floor Plan

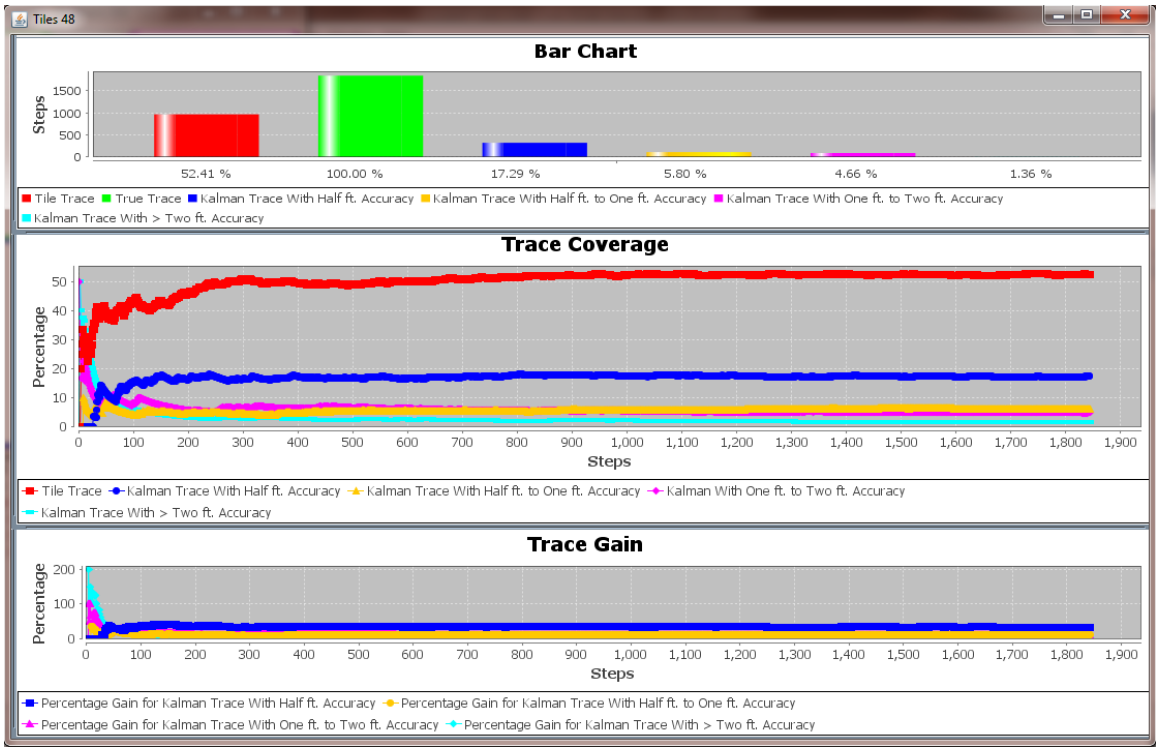

Figure 18 Variable Rate Linear Walk Model - 48 Tiles Floor Plan

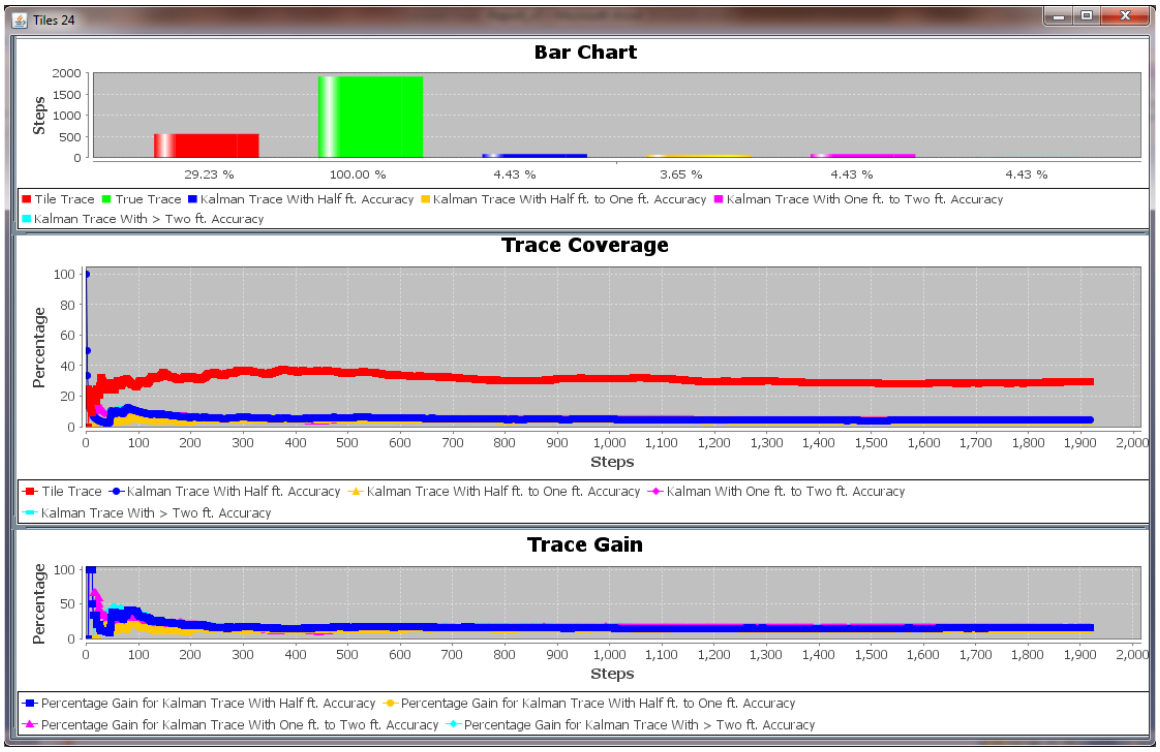

Figure 19 Constant Rate Non-linear Walk Model - 24 Tiles Floor Plan

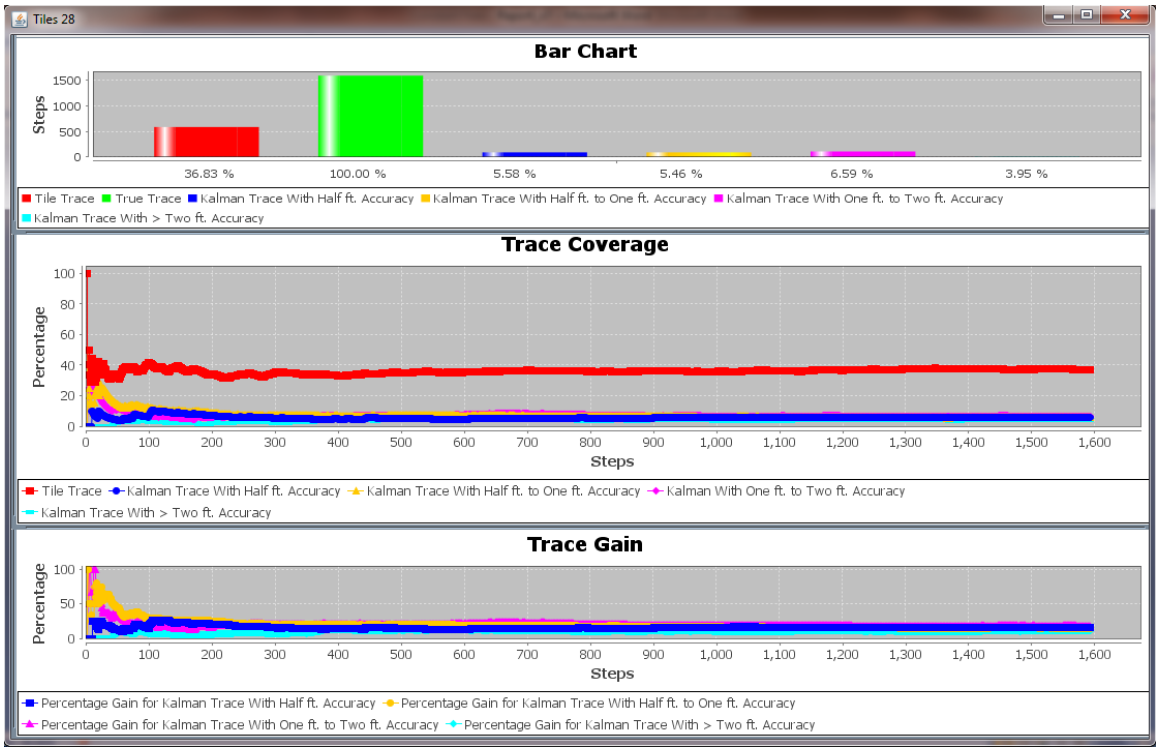

Figure 20 Constant Rate Non-linear Walk Model - 28 Tiles Floor Plan

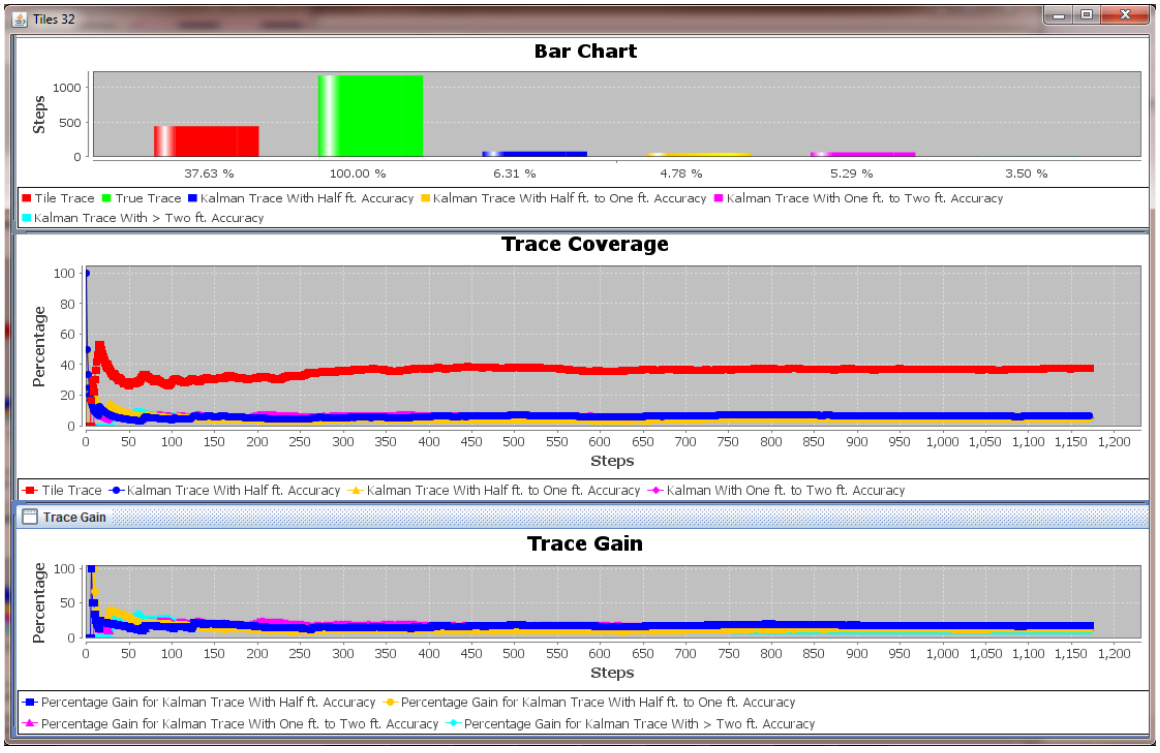

Figure 21 Constant Rate Non-linear Walk Model - 32 Tiles Floor Plan

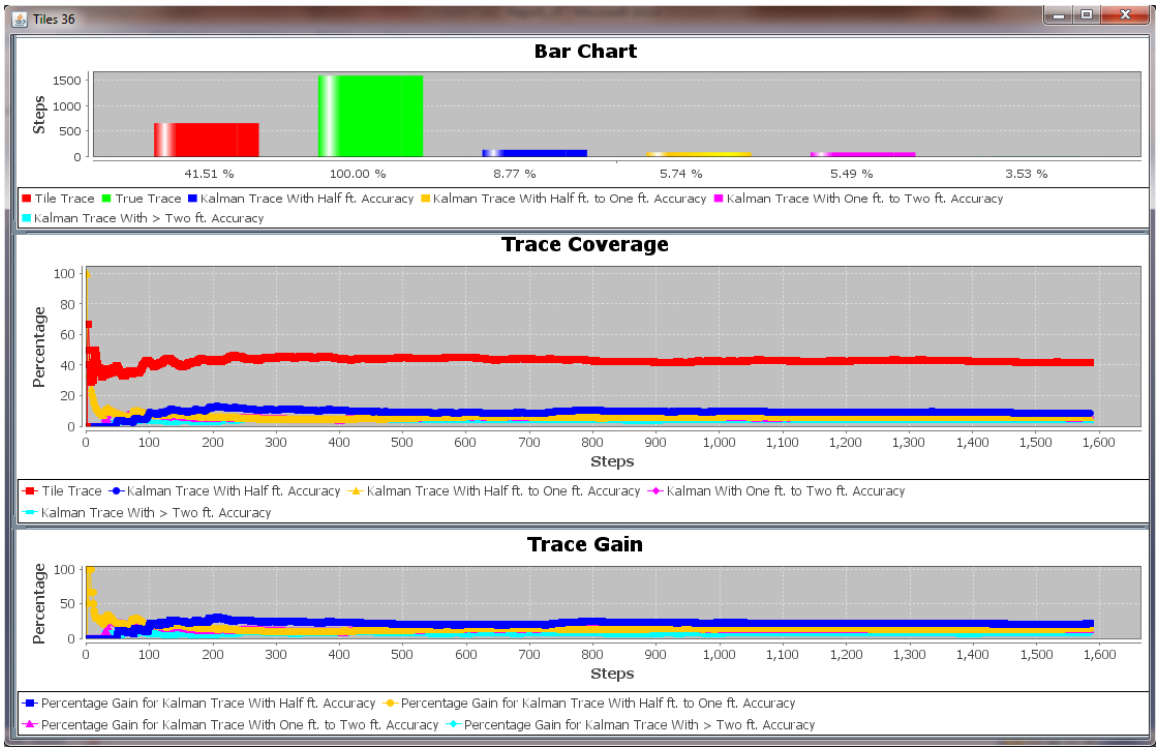

Figure 22 Constant Rate Non-linear Walk Model - 36 Tiles Floor Plan

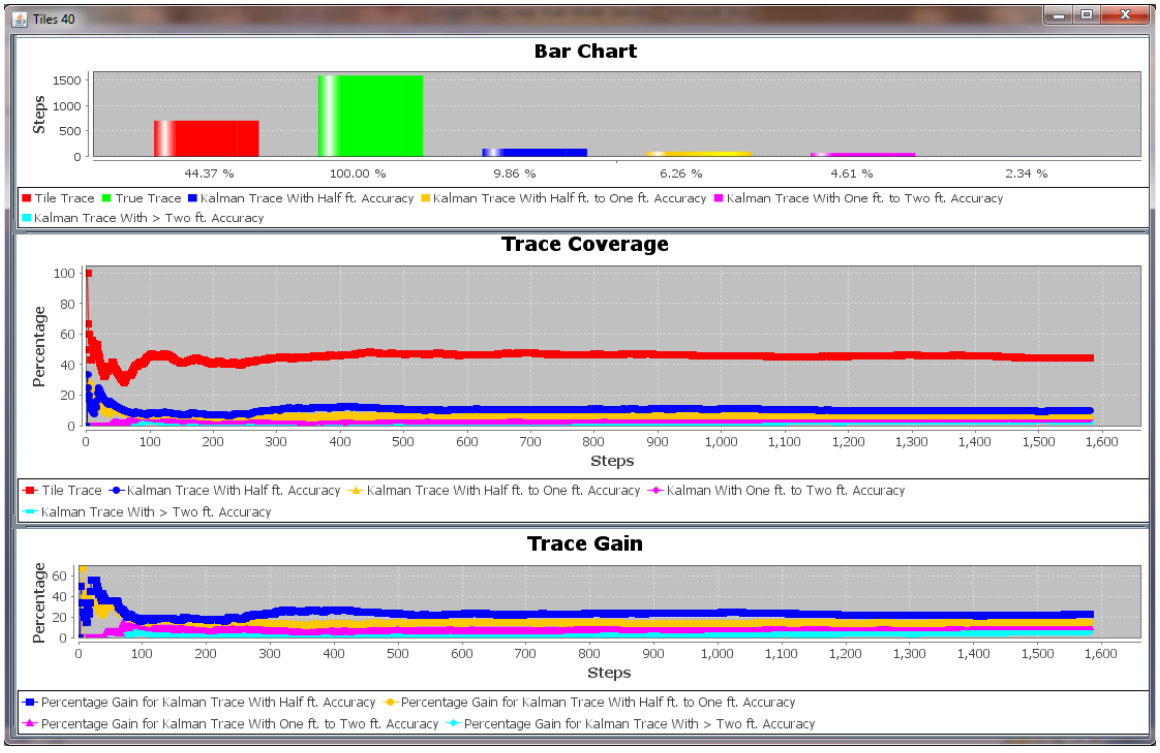

Figure 23 Constant Rate Non-linear Walk Model - 40 Tiles Floor Plan

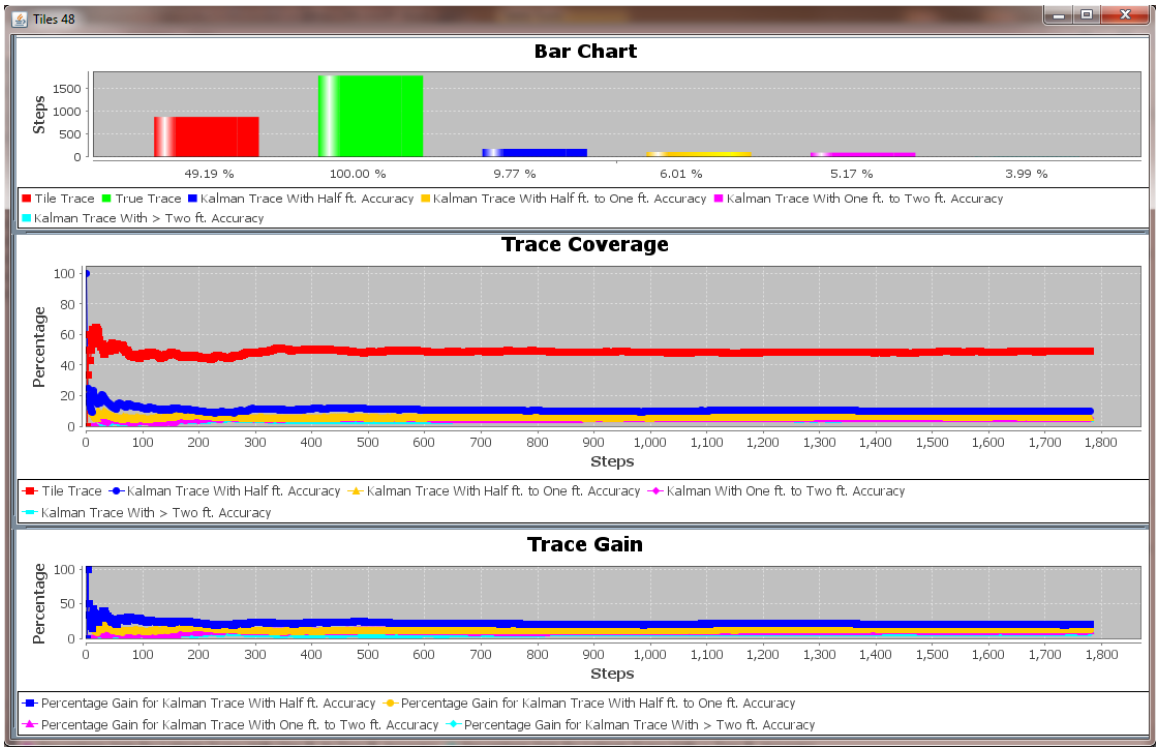

Figure 24 Constant Rate Non-linear Walk Model - 48 Tiles Floor Plan

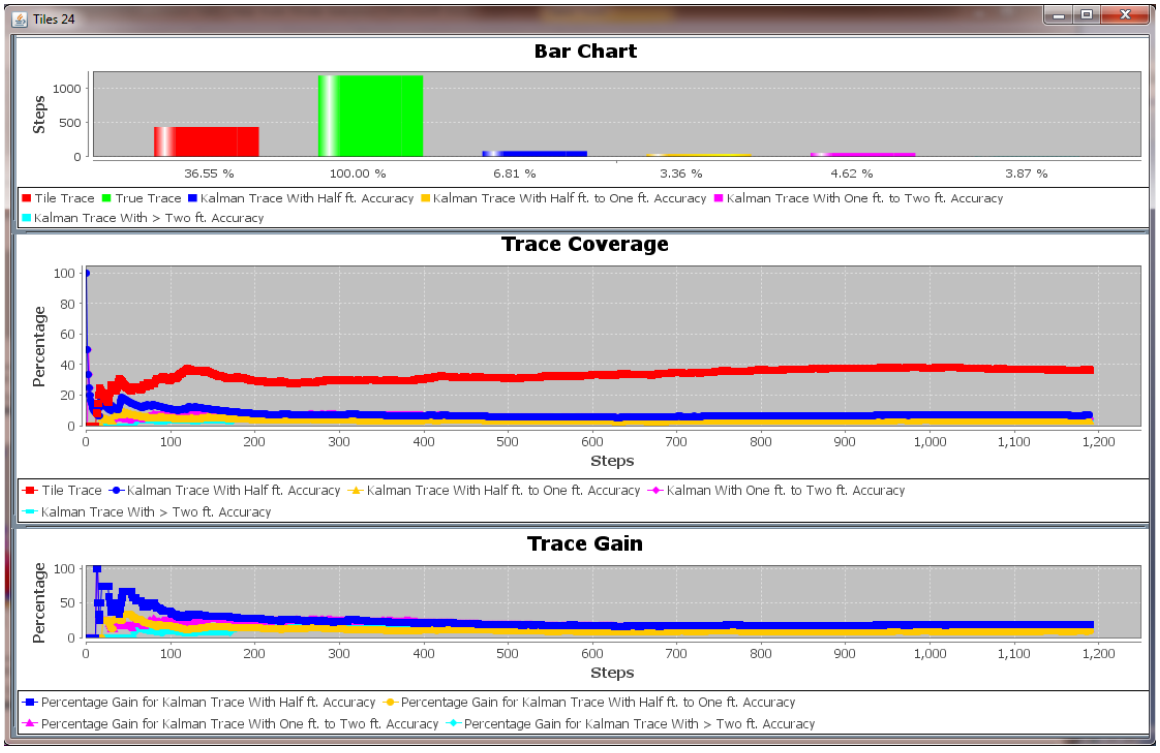

Figure 25 Variable Rate Non-linear Walk Model - 24 Tiles Floor Plan

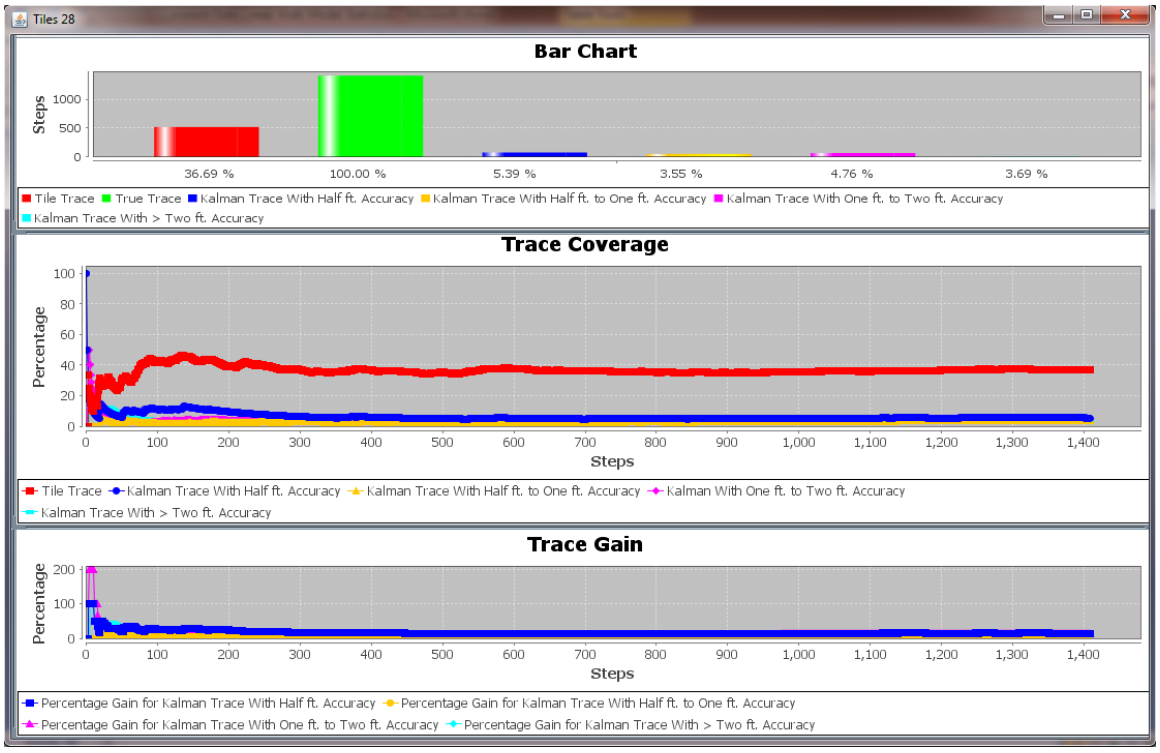

Figure 26 Variable Rate Non-linear Walk Model - 28 Tiles Floor Plan

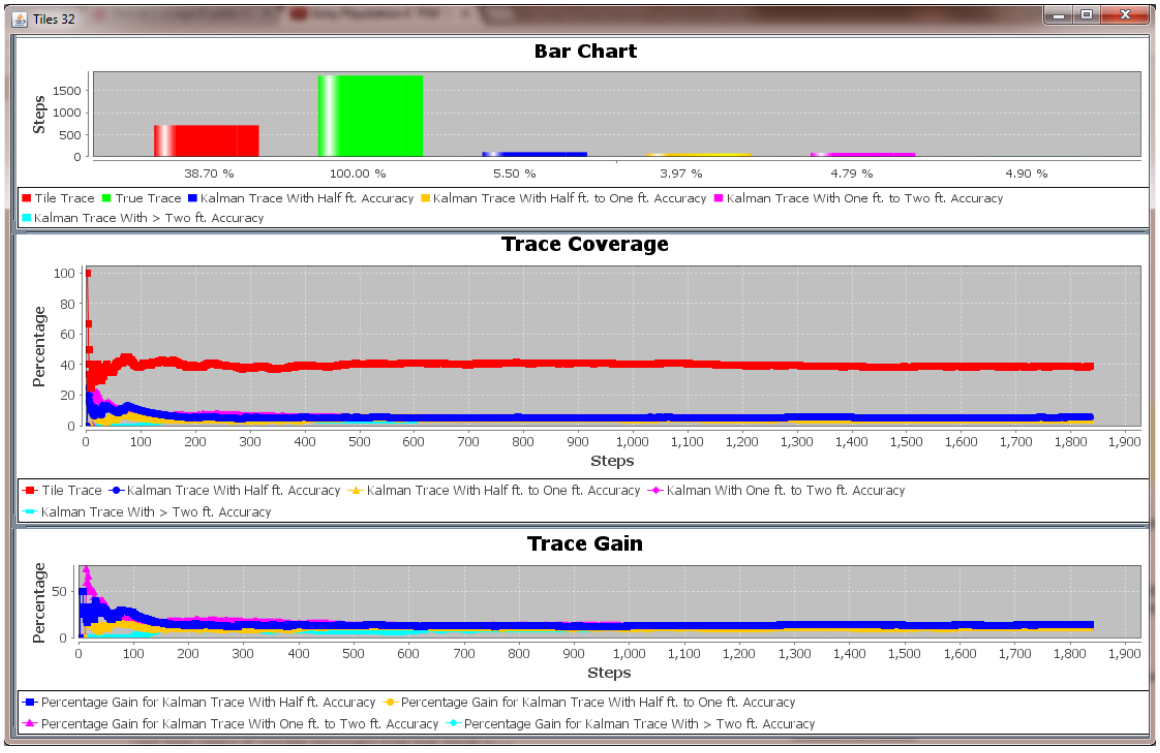

Figure 27 Variable Rate Non-linear Walk Model - 32 Tiles Floor Plan

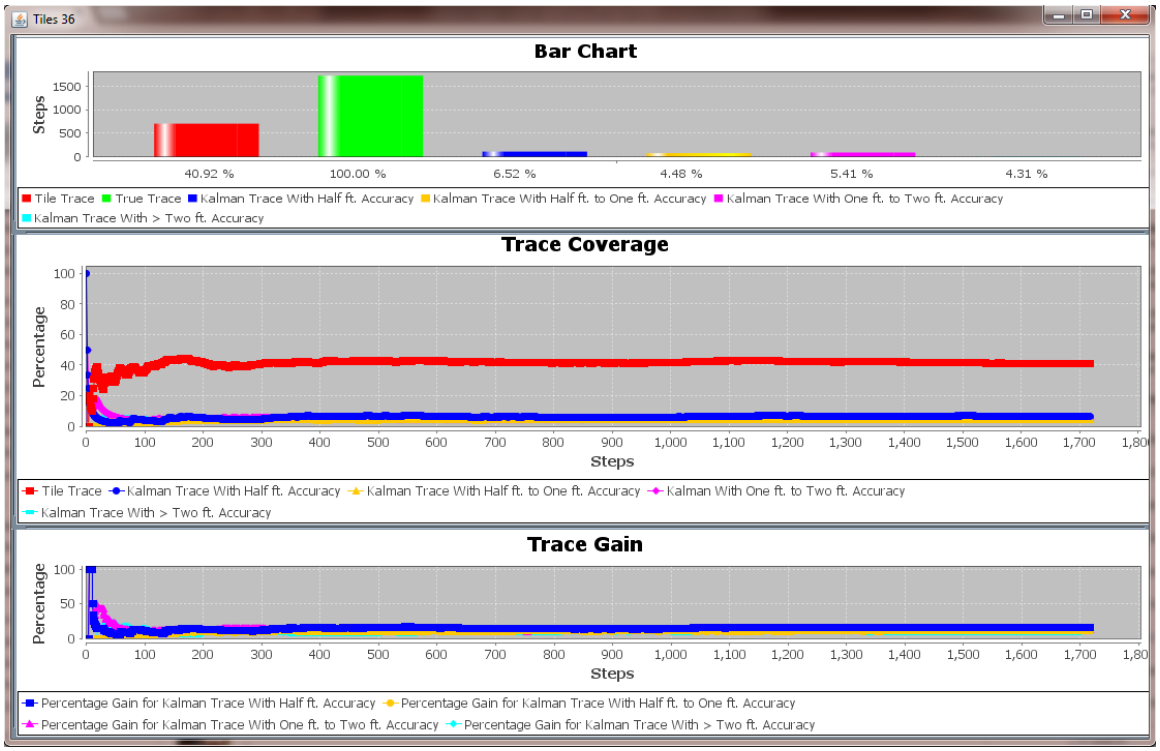

Figure 28 Variable Rate Non-linear Walk Model - 36 Tiles Floor Plan

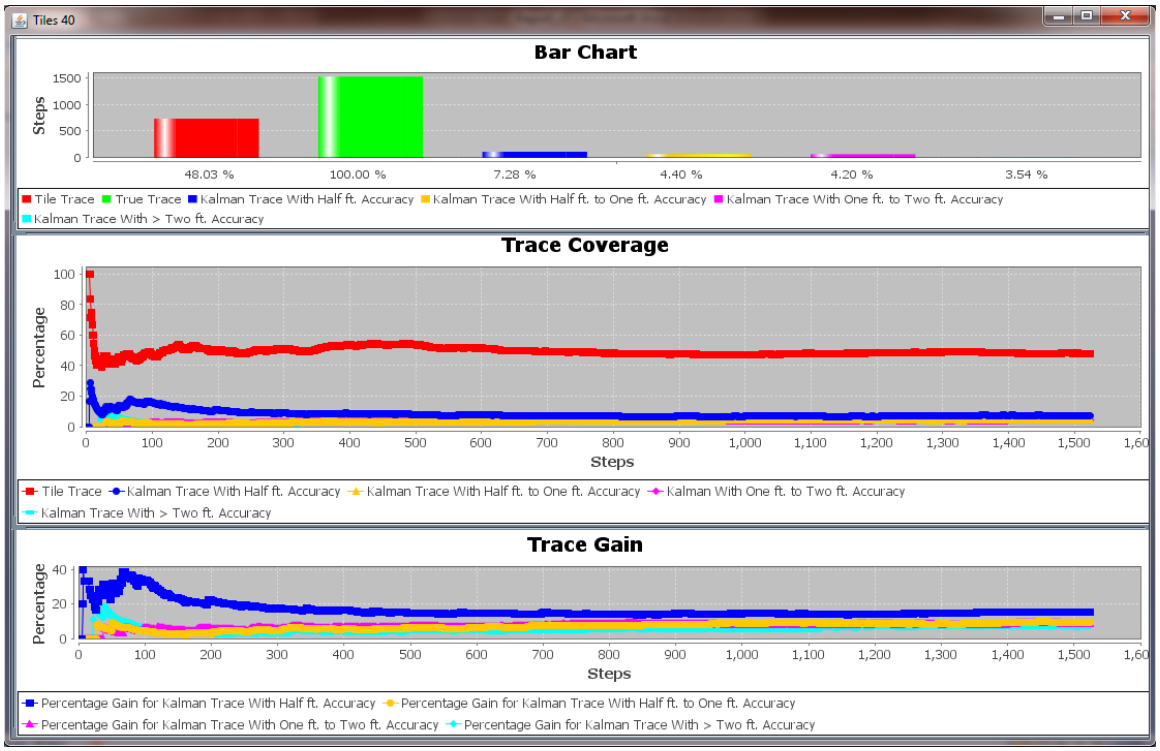

Figure 29 Variable Rate Non-linear Walk Model - 40 Tiles Floor Plan

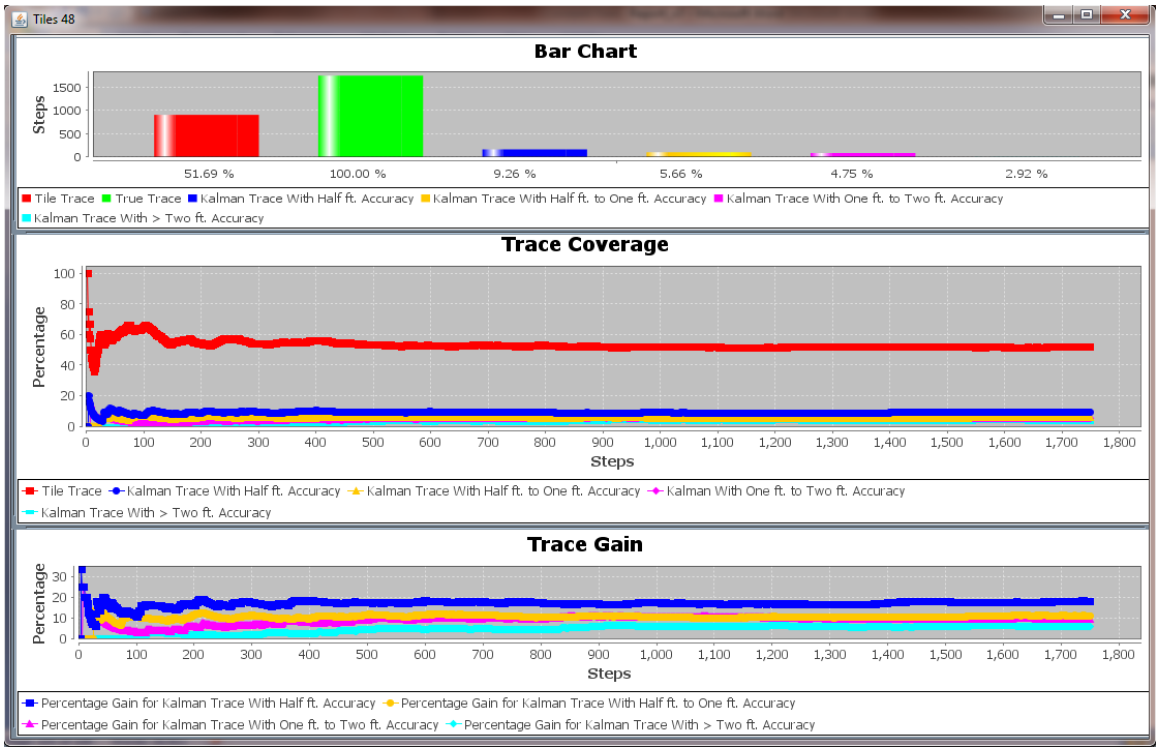

Figure 30 Variable Rate Non-linear Walk Model - 48 Tiles Floor Plan