I N A U G U R A L - D I S S E R T A T I O N

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht - Karls - Universität

Heidelberg

Diplom Informatiker Domnic Savio Benedict
aus: Coimbatore, Indien
Tag der mündlichen Prüfung: 19.05.2008

Tracking Biped Motion in Pervasive Environment

1. Gutachter: Prof. Dr. habil Thomas Ludwig
2. Gutachter: Prof. Dr. habil Gerhard Reinelt

Ich versichere, dass ich die vorliegende Dissertation selbst und ohne unerlaubte Hilfe angefertigt habe. Es wurden alle in Anspruch genommenen Quellen und Hilfsmittel in der Dissertation angegeben.

....

Abstract

Textiles are ubiquitous to humans since ages. Transistors made of silicon have made a deep impact in modern industry. A new field of research called wearable electronics integrates both these worlds to provide intelligent new services. Based on modern technologies of textile manufacturing, a carpet is embedded with a network of computing devices. One of their applications is to sense, when someone walks over them. This carpet was used to track the path a person took on his walk.

When a person steps on the carpet, embedded sensors in the carpet get activated. These activations are stored at a monitoring PC as a log file. This data is processed and carefully viewed by data mining algorithms to identify hidden patterns that reveal the trails of the subject on his motion over the carpet.

Different methods for validating the data mining algorithms are presented. These methods are perfected to produce an ideal reference in a format that can be directly compared with the estimated results of the algorithms. The evaluation results show a better performance for the new approach compared to the state-of-the-art technologies.

Veracious testing, discussions, suggestions and their impact after implementation, are discussed in detail. The concepts used in the data mining algorithms are structurally sound and maintainable. Suggestions are given for further work on this system as whole. The footsteps of the person walking on the carpet are identified. The trajectory of walk is traced. The carpet can be used in a variety of domains. Rich examples on usage, assisted with augmented literature conclude this work.

Zusammenfassung

Textilwaren sind dem Menschen seit langem allgegenwärtig. Aus Silikon gefertigte Transistoren beeinflussen sehr stark die moderne Industrie. Ein neues Forschungsgebiet nennt sich "Tragbare Elektronik". Dieses Gebiet integriert Silikon und Textilien, um intelligente neue Dienstleistungen zur Verfügung zu stellen. Beruhend auf modernen Technologien der Textilherstellung wird ein Teppich in ein Netz von Rechengeräten eingebettet. Dieser Teppich kann dazu verwendet werden, um die Berührung eines darüber laufenden FuSSes zu erkennen. Er kann somit den Weg ausfindig machen, den eine Person auf dem Teppich gegangen ist.

Wenn eine Person auf dem Teppich geht, werden eingebettete Sensoren im Teppich aktiviert. Diese Aktivierungen werden am Überwachungsrechner in einer Protokolldatei abgespeichert. Die Daten werden durch Data Mining- Algorithmen bearbeitet, um versteckte Muster zu identifizieren, die durch die Bewegungen über den Teppich verursacht wurden.

In der vorliegenden Arbeit werden verschiedene Methoden zur Validierung der Data Mining-Algorithmen vorgestellt. Diese Methoden sind dazu geeignet eine Referenz in einem Format bereitzustellen, das direkt mit den Ergebnissen der Erkennungsalgorithmen verglichen werden kann. Die Ergebnisse dieser neuen Algorithmen erreichen eine höhere Genauigkeit als bekannte Technologien.

Des weiteren wird auf zuverlässige Prüfungen, Diskussionen, Vorschläge und deren Einfluss nach der Implementierung ausführlich eingegangen. Die Konzepte, welche in den Data Mining-Algorithmen verwendet werden, sind gut strukturiert und erweiterbar. Die Schritte einer Person, die auf dem Teppich geht, können identifiziert und deren Laufbahn verfolgt werden. Der Teppich kann in verschiedenen Bereichen angewendet werden. Hierfür werden abschlieSSend Beispiele genannt und Vorschläge für weitere zukünftige Arbeiten an diesem gesamten System unterbreitet.

Acknowledgements

This work would not have been possible without the help of good hands and I like to mention a few of them here.

I sincerely thank Prof. Thomas Ludwig for accepting me as a candidate in his research group although the nature of the thesis was not his research focus. His interest on this topic had encouraged a lot of my efforts. I must thank Dr. Werner Weber at Infineon Technologies for giving me the idea first to do this thesis. It all started at the old Lab for Emerging Technologies at Infineon. Dr.Weber had been of credential help and motivated me in all the phases of the thesis.

Prof. Dr. Dr. Thomas Sturm at the Bundeswehr Universität, Munich help me crystallize certain vague ideas without which I could have easily deviated. His immense knowledge and experience had been a lamp on the dark ally. Sincere thanks to Dr. Armin Stolze who took care of supervising my thesis at difficult times and supported me to complete the work.

I wish to thank all at former Lab for Emerging Technologies at Infineon, especially Rupert Glaser, Guido Stromberg and Markus Schnell for their motivation, critics and feedback. I thank them for making the past three years a wonderful time to remember.

Tanja and Enya for helping me afford my time with the laptop although we missed each other a lot. They put up with me in all the decisions I had taken and I owe them for their support.

Domnic Savio Benedict, Munich, 10th June 2007

Contents

1. Introduction

The desktop computer has rapidly changed the way we do things. A typesetting program enables the secretary to edit a document any number of times, before the final version is printed which was not the case a few decades before. Not only the typewriter was replaced by the desktop, the process of editing, typesetting, and proof reading consequently added comfort in authoring, set the liberty of illustration, and enhanced accuracy on the final document. This was the aftermath of one desktop computer, which hosts a microcontroller as its seat of thought. Looking around the number of gadgets that hang around our desk today, the printer, fax machine, cell phone, calculator, the PDA, and notably the bluetooth headset for example all have similar microcontrollers with different degrees of computing power and communication capabilities adding ease, flexibility, intelligence on the execution of our thoughts, and activities in our daily routine. What is now being influenced is the way we do things, for example papers and post are packed in bits and bytes, transported on the information highway and the delivery time is less than a finger click.

There is another set of interesting changes that have taken place. The mouse and the keyboard that used to be the input devices of the standard desktop have now a series of extended family members. The interface devices are changing shape and incorporate friendly features. The integrated keypad in the mobile phone doesn't have 101 keys but still can be used to type almost every key of a regular keyboard. And the mobile phone can hear us and dial a number when we just command it. The mobile phone can understand our language. How would it be if our computer could understand our walking! Close a door and switch off the lights when we leave the room! Making the computer understand our footsteps and identify the path of our walk is a demanding intention, focused in this thesis. The applications that benefit from this understanding are classified under the banner of context aware computing. When trying to examine walking and developing an interface for our footsteps, this thesis focuses precisely on tracing the path taken by the subject while walking by identifying where the feet were placed. By tracking the feet and their movements the methods in this thesis plot a trajectory of walk, the subject took when he or she moved from one place to another. Based on the plot subsequent actions can be taken by the desktop, decorated with applications.

Tracking human motion is an ongoing area of research attracting rich experiments. The attempted formal methods can be broadly classified as intrusive and non-intrusive tracking techniques. In an intrusive technique, sensoric devices are embedded on the subject. Here, the subject has to wear or carry some sensors while being in motion. The sensors then activate a series of controllers, wired or wireless. Data collected from these sensors are then processed off-line or real time. With non-intrusive methods, sensors are placed in a fixed location. The data is collected from these sensors to identify movement. A fixed camera or a motion detection sensor for example could be used to extract features of the subject to locate and calculate the position and displacement of the subject.

1.1. Observing Walking

Although we consider both the cases, non-intrusive methods of tracking have an advantage as the subject does not need to carry or wear any sensors. But to recognize objects in both cases, location of the sensors have to be optimized, features have to be selected to determine what is being tracked, and the result of the tracking system should have an acceptable tolerance. The methods used could depend on the desired result. However, the science of observing walking could make tremendous contributions. For example, in neuroimaging studies, researchers investigated biological motion and its activity in certain regions of the brain (STS - superior temporal sulcus) [RS04]. They observed that normal biological motion activates STS. By simulating non-biological motions and carefully coordinating them, they were able to understand how intensively the brain could process non-biological motion, which is a huge resource for physically disabled persons. Pervasive means of monitoring health has tremendous impact in providing quality health care. In the U.S. about 1.4 million lower extremity fractures are occurring annually. Researchers show that the patients recover faster with limited weight bearing programs. But gauging how much pressure to apply on the injury, before doing harm, is difficult. A new E-foot was developed to address this, by the students in the Johns Hopkins University [Uni01]. A force sensor is placed under the sole where the feet lands. As the patients walk, it measures the pressure and alerts patients if they apply too much pressure on their injured leg. FitSense a Massachusetts based sports wear company, has developed footwear that can monitor speed, pace and distance of an athlete during training. This confirms the fact, walking has been observed for quite some time. The observations can be used in context aware computing to enhance safety and performance. However, the above examples are intrusive ones which require the user to wear a part of the system which has sensors. It would make more sense if the subject didn't need to be burdened with wearing such apparel.

1.2. Extracting Gait

Observing a human walk is a part of identifying the physical characteristics of the person during his motion. Methods used to identify persons based on their physiological or behavioral characteristics are called biometrics. Proven biometric methods range from fingerprint and hand geometry tangenting to more sophisticated techniques like face recognition and iris scanning. If we consider these methods under the ambiance of context aware computing, the methods need direct contact to the sensoric elements or more precisely they need to get the attention of the subject to proceed an action. Although face recognition and iris scanning do have advantages over it, they need illumination and often cannot be hidden from the view of the subject. An infra-red camera can of course reveal the presence of a person, but to extract facial features, perception is diminished in an IR image. Biometrics that can address these shortcomings is human 'gait' or the walking style of an individual. Gait is more attractive because it does not require a cooperative subject and non intrusive methods can be extensively employed. Early attempts of gait recognition date back to mid 70s. A subject with dark attire was attached with small light bulbs at his joints [Joh75]. When he was still, the lights formed a star constellation to indicate the subject was standing. When he moved, the lights formed curves and cues to indicate motion. Research over the years gathered evidence that there exists identity information in gait and motivated development of computer vision based algorithms for gait based human recognition. Movement of thighs were fitted in pendulum motion models to identify gait of a person (Fig. 1.1) [DCC95].

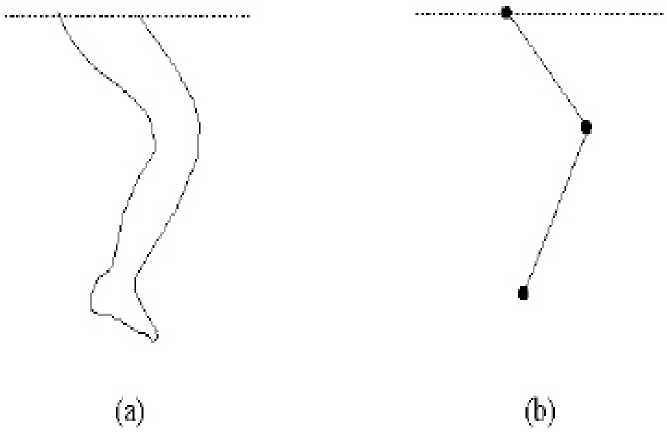


Figure 1.1.: Pendulum model of a leg

Binarized silhouette of a person was taken and fitted with numerous circles (Figure 1.2) and ellipses [LG02]. Then the centroids, tangents and foci from the shapes were taken to identify a gait signature.

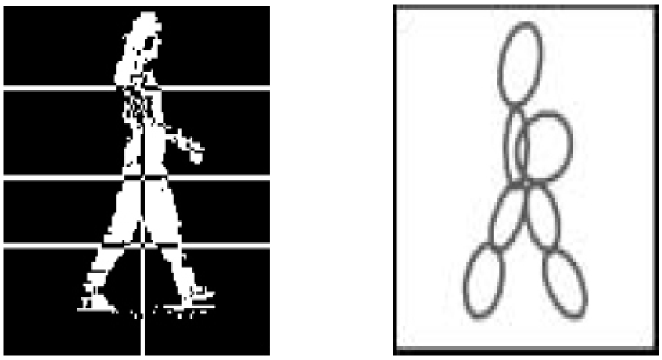
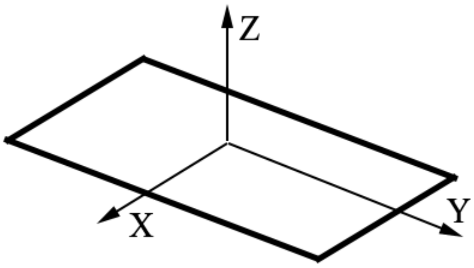


Figure 1.2.: Fitting Ellipses on Silhouette

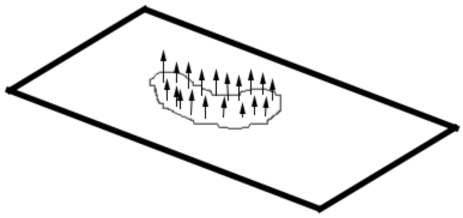
Most of these works have been carried out based on images and videos taken by observing walking. However, background detection, low illumination levels and objects in the background contribute a lot of noise that is still posing a major problem for gait identification of a person by visual means. Secondly the presence of a camera before a person always gives the psychological feeling, every step he or she takes is being watched.

1.3. Ground Reaction Forces(GRF)

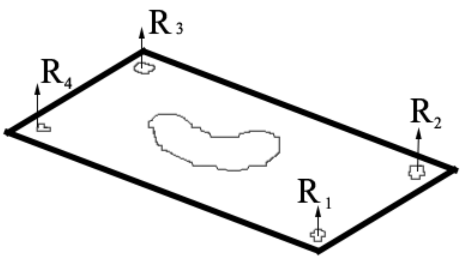
While walking, as a person moves, due to gravity he constantly maintains contact with the ground. In this process there occurs interactions between the body and the ground. At each foot step, the body exerts certain forces on the ground. According to Newton's 3rd law of motion, there is an equal and opposite reaction. This reaction force supplied by the ground is specifically called the ground reaction force (GRF). It is an important external force affecting human motion.



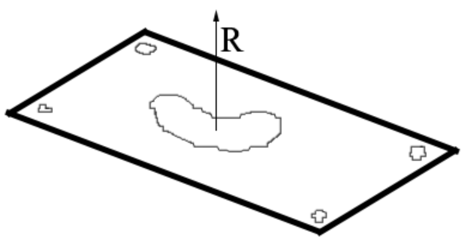
(a) Force components



(b) Ground reactions



(c) Reaction on force sensors



(d) Equivalent force

Figure 1.3.: Ground reaction forces

The ground reaction force (Fig. 1.3(a)) has three components: X-, Y- and Z-components (R_x, R_y and R_z). Among these, the Y-component is along the direction of the motion which reflects the propulsion or the braking force. The Z-component supports the body to prevent it from collapsing, and also to thrust the body upward in jumping motion. Figure 1.3(b) shows the reactions from the ground to the foot. The sum of all the reactions from the ground shown in figure 1.3(d) is equivalent to the sum of the forces measured by the four force sensors in a force plate (R_1 + R_2 + R_3 + R_4) as illustrated (Fig. 1.3(c)). A force plate consist of force sensors placed at the four corners of a stable flat plate. The force sensors can be connected to an analog to digital converter or an oscilloscope to measure the signals.

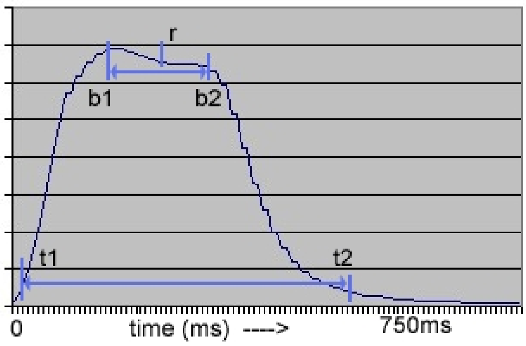


Figure 1.4.: GRF on an oscilloscope

Figure 1.4, shows the observed oscilloscope signal of a step during a normal walk. The first peak *b1* is the result of the heal strike on the ground. Shoe designers often observe this part to know the softness and the grip of the material while designing their performance wears. As the person lands the palm and tends to parallel the foot to the ground the transfer of body weight takes place. This is the valley at *r*. As the person transfers the weight on the foot, he or she moves forward and tends to lift the foot applying a small pressure at the toe, this is the second peak at *b2*. As the person removes the foot from the ground the slope *b2* to *t2* takes place. This complete profile is observed from the four force sensors placed at the edges of the force plate. Constructing such sensors on the floor would help to detect the person's weight, speed of walking, stride length and furthermore, by making a GRF profile a person's gait signature can be constructed to identify a footstep.

1.4. Hiding Gait Extraction

To construct this gait information, force sensors with signal processing circuits are needed. An analog to digital converter parses the signals coming from the force sensors and constructs a digital information of the person's foot stamp.

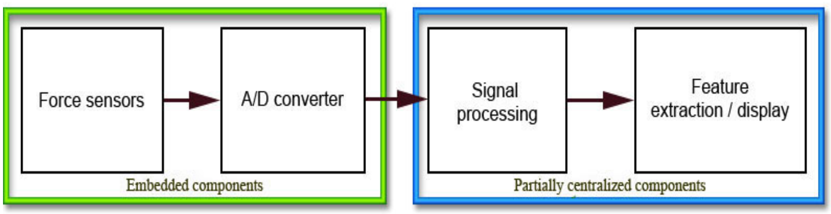


Figure 1.5.: Blocks for footstep recognition

This data is then passed through signal processing algorithms running on a computer to analyze and detail features of the footstep. This requires a lot of computing effort in the embedded level. A desktop computer and the computing time cannot be avoided unless fast signal processing circuits are miniaturized to fit in a sensor. Alternatively certain blocks of the process can be embedded near the sensors and the rest can be routed to a centralized processing machine (Fig 1.5 on the previous page). For example the force sensors and the A/D converters can be placed together with certain signal processing capabilities on a single printed circuit board. Such a board can be placed on a flat plate or a tile big enough to cover a foot stamp. Placing such tiles on the ground would avoid the visibility of the sensor.

By observing walking in this method, the sensor that captures the motion can be completely hidden from an intruders knowledge by simply embedding the sensor in the floor. This allows not only the subject to walk comfortably, but irrespective of the illumination, night or day, a gait profile can be constructed by this method. A person's footsteps can be detected without any background noise which dominates in video based gait detection. In areas where surveillance is required, the struggle to conceal monitoring cameras is often an issue based on the environment. The best hidden eye can at times hide information, too. But in this approach of embedding sensors in the ground new possibilities of observing and using human motion can be realized.

1.5. Pervasive Environment

Observing human motion by integrating sensors and associated electronics in the environment to add comfort and enhance services is a new area of research. Sensors are embedded in a normal carpet and the observed data is then sent to a desktop computer which processes the data coming from the carpet. Although processing algorithms are the main contribution of this thesis, this work also provides an insight into technologies that can understand the environment of walkers and provide services relevant to the context. Human motion, observed by embedding computing power in an existing environment like a floor carpet, is presented in this work. By this approach, the presence or absence of a foot press is detected and reported to a desktop where the data is analyzed and relevant services can be rendered. This not only strikes the sticks to disappear the computing behind a walk, it also paves the way for new services. Imagine Bob, with a busy schedule has such a carpet installed in his office room. As he leaves the office in the evening, the carpet senses his footsteps and calculates his path when he leaves the room. If he exits at the main entrance and is not reentering, perhaps for a few minutes, the lights can automatically switch off. If installed in an old age home, it could activate an alarm if Bob's dad fell on the ground, was helpless and could not move. By embedding sensors and their electronics in a carpet the computing is well hidden from the user's point. It also enables a next generation of interacting devices and user interfaces. It offers the possibility for a new vision of applications which are based on how and where we walk.

1.6. Summary

We live in a world where personal electronics try to understand our requests. This development has over the years caused machines to become outdated. As devices become smaller, research is trying to make them interactive. One such context aware computing is trying to provide services based on human motion. As a person moves, the walk triggers an equivalent ground reaction force which can be trapped by force sensors. Converting this information enables new technologies to identify the location of a foot step. As a subject walks, permanent contact to the ground is always maintained in the path, which helps tracking algorithms to identify it. Tracking human motion promises new applications in the field of biomedical engineering, surveillance, surveying and health care to name a few. The next chapter briefs the prototype of such an ubiquitous concept with details of actual hardware. It is then followed by an overview of contents of each chapter in this thesis.

2. Human Motion Tracking and The Smart Carpet Concepts

Enabling things around to compute and communicate, we create an environment where information is available anywhere and anytime also known as Ubiquitous Computing. Under this context, common devices such as appliances, gadgets and toys are given computational sensing and communication abilities. This technological movement has been amplified by research initiatives like MIT Media Laboratory's Things that Think projects and Xerox PARC's concepts on Ubiquitous Computing. The fact that humans enjoy clothing as a universal body language has contributed to new areas of research where textiles and computing are integrated. A system stitched inside a carpet as a sensor layer would hide the existence of computation in a carpet. Such a carpet could be used to capture human motion.

2.1. State-of-the-Art for Pervasively Identifying Human Motion

There has been an increasing interest in pervasive means of sensing and identifying foot pressure. Some of the pioneering works towards this goal are worth mentioning at this point. Early attempts of tracking human motion can be traced back to Olivetti and Oracle Research laboratories [MAS97]. A floor tile was considered as a force plate. Load sensors were integrated to form a sensor tile. In the early stages one load cell was considered for one tile. Although the weight of an object could be precisely measured, many times the weight of the object would be spread across the tiles. It was not easy to interpret spread loads. Hence four load cells were taken and placed on the edges where tiles join (Fig. 2.1). In this way one load cell could collect load from different cells and the load spread is dealt uniformly to form an Active Floor.

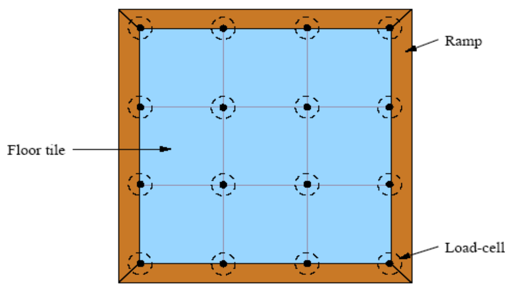


Figure 2.1.: Construction of Active Floor

An array of load cells supporting four adjacent floor tiles in the middle and two at the edges is placed to form the Active Floor. Steel plates about 500 by 500 mm in size and 3 mm thick are shaped together with an 18 mm thick plywood of similar size to form a tile. Carpet tiles were fitted on top of the floor tiles to give the appearance of a normal office floor. A wooden ramp is placed on all four sides of the Active floor to provide a smooth transition.

A full bridge circuit provided 10 V DC excitation voltage to the load cells. A VXI based strain gauge data acquisition equipment with a 64 channel, 16-bit ADC board converts 16 full bridged load cell signals to a Hewlett-Packard's Virtual Engineering Environment (HPVEE) software. Sampling the signals of walking and running people at different sample rates, it is found that most of the load cell signals are less than 250 Hz. The experiment was split into three different stages. In the initial stage, data from the center load cell was taken and sampled at 500 Hz. In the second stage, 819 samples of each sensor's data set are averaged and the result is deducted from the data set to remove systematic errors due to the static weight of the floor. The data is then converted from volts to kilograms. This instruments in finding the weight of any object placed on the Active Floor which is directly proportional to the vertical GRF components. In the final stage the previous data set is analyzed and the time traces of the footstep's vertical GRF components are extracted. The experiment is repeated for different subjects in a regular pattern. Each subject was requested to step over the center tile. A database logged the foot signatures for a group of 15 people.

Hidden Markov models (HMM) are constructed and the data is passed on to these models. The weight estimator calculated the average of the GRF over a step interval. This resulted in error rates greater than 50 % obtained by the HMM. Hence the direct application of HMM over the entire signal was inadequate. Although the results are not that practical, there are certain useful properties that distinguish this experiment from alternative approaches:

- It is non inversive
- Objects cannot be hidden
- Pervasive in nature
- Can be hidden from the subject under observation
- Can be combined with other sensor system

It is evident from this experiment that sensors embedded in the floor can reveal the distribution of vertical GRF components. These signals can be captured to identify human motion.

Since Active Floor used HMM which averaged the amplitude of the GRF signal over the step interval which resulted in more than 50% errors, a further development was made in Smart Floor [OA00]. Here, similar experimental setup is used at the Georgia Institute of Technology in building the Smart Floor. Instead of an array of tiles, only one tile with four load cells placed in the four corners of a custom made force plate are used. The captured signals are also treated differently. Instead of using an HMM model a more computationally easy method is chosen where footstep profile features are drawn out of the footstep interval (Fig. 2.2).

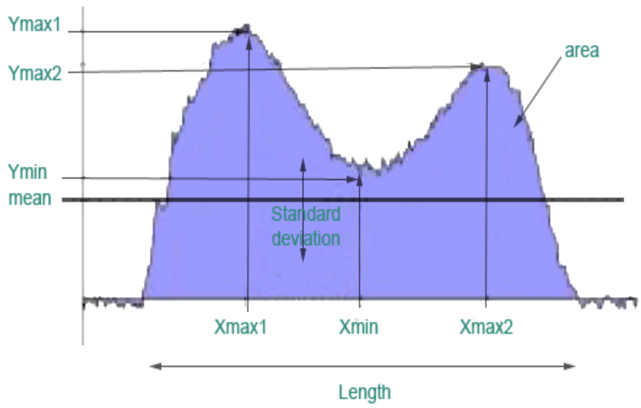


Figure 2.2.: Footstamp profile using Smart Floor

The features included the mean value of the profile; the standard deviation of the profile; the length of the profile; the total area under the profile curve; the coordinate of the maximum point in the first half of the profile; the coordinate of the maximum point in the last half of the profile and the coordinate of the minimum point between the two maxims. The footsteps are placed in a ten-dimensional feature space and a nearest-neighbor search is performed to match unidentified footsteps with the identity of the closest cluster in the training set. The Euclidean distance is then calculated. The identity of the cluster with the least average distance is taken as the identity of the unknown footstep.

Subjects were made to walk in straight lines and make a single step on this floor tile. The walk was repeated over different shoes and various foot sizes. The Smart Floor is able to identify 88 % of the footsteps correctly. It is concluded from this research that the footwear does not greatly affect the ability of this approach to identify user by his footstep. The Smart Floor was intended to be deployed in the Aware Home of the Georgia Tech ubiquitous experiment project [Ini01, Ess00].

In another approach [SPR03b] new materials are applied in constructing a sensor based floor. Electro mechanical film (EMFI) is a thin, flexible low cost conductive material which consist of cellular, biaxially oriented polypropylene film coated with metal electrodes. In manufacturing this film, an internal vacuum layer is created in the polypropylene layer where a large permanent charge can be stored by electric fields that exceed the dielectric strength of the EMFI (Fig. 2.3 on the following page).

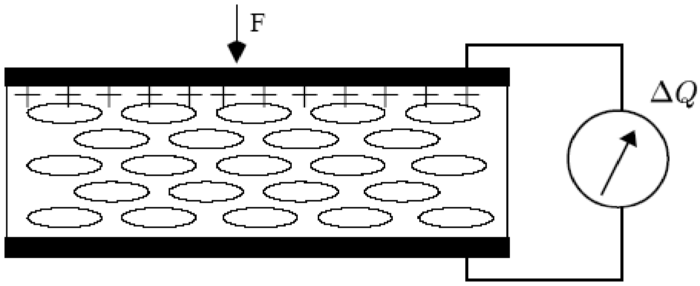


Figure 2.3.: Cross section of a EMFI sensor stripe

An external force affecting the EMFi surface causes a change in the film's thickness, resulting in a change in charge between the conductive metal layers. This charge can be detected as a voltage signal. The EMFi floor consists of 30 vertical and 34 horizontal EMFi sensor strips. Each strip is 30 cm wide and 10 m long. They are installed under normal flooring. Each stripe produces a continuous signal which is sampled at a rate of 100 Hz using an analog to digital conversion card - PCI6033E from National Instruments. This data acquisition device handles upto 64 analog channels simultaneously. The digital output is streamed to a PC on the PCI bus.

As a subject walks on these stripes, the majority of the footsteps land in the middle or between two adjacent stripes. This results in a heel strike profile on one channel and the tow push off profile in the one next to it. Secondly, when consecutive footsteps land on the same stripe, the baseline of the EFMI-signal starts to fluctuate. As a signal amplitude based simple thresholding method done in HMM is inadequate for footstep pattern detection, a statistical pattern matching method [SPR03a, KKR04] based on Segmental Semi Markov Model (SSMM) is chosen.

This method contains two major components: an explicit state duration distribution and a segmental observation distribution. Unlike standard HMM, where a state generates a single observation yt, a state in a SSMM generates a segment of observations $yt_1 \dots yt_2$. The duration of this segment in time is modeled by a specific distribution with a mean duration and some variability around that mean. This segmental observation model brings the aspect of *shape variability* into the detection process. This helps to generate a specific footstep pattern for every kind of footstep. Once this footstep profile is generated by combining observed segments from adjacent stripes using a piecewise linear segmentation algorithm, then a Viterbi based algorithm is used to detect similar waveforms in the data generate by the EMFi-sensor (Fig. 2.4 on the next page). This experiment has proven that footsteps can be more reliably detected than with the previously developed HMM based detection techniques [SPR03a].

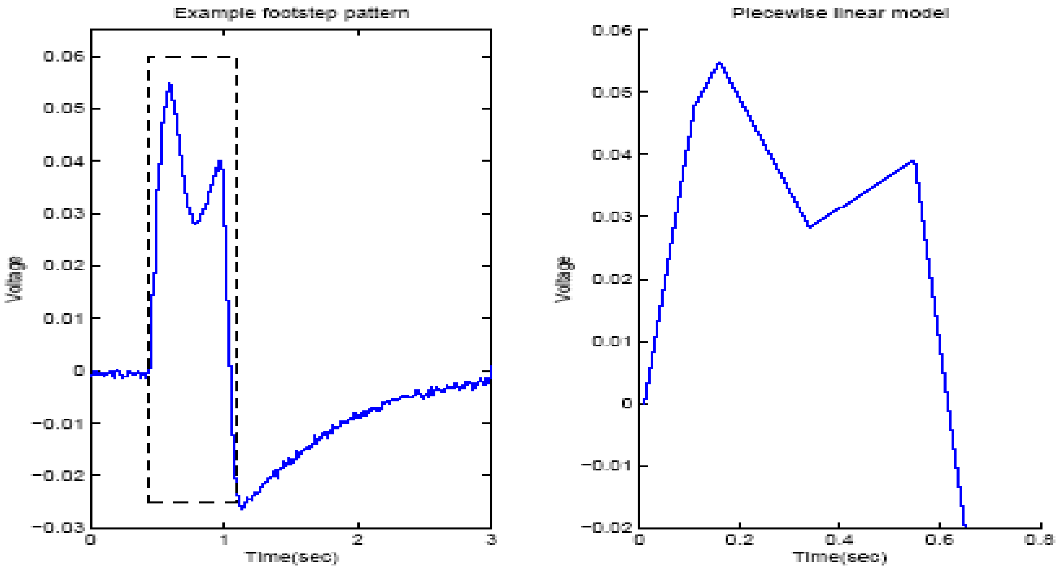


Figure 2.4.: Foot stamp profile segmented in PLS

In an interactive approach, Magic Carpet [JPR97] produces a soundscape which generates music and notes to the subtle movements of a performer when he dances on this carpet. A cable that can sense foot pressure is laid out in the carpet. The cable consists of piezoelectric wires with an inter wire pitch of 4 inches and an insulating layer of piezoelectric co-polymer material. The cable is shielded at the outer layer to give the construction of a coaxial cable. Such a cable, working like a transducer, produces a voltage when pressed or flexed anywhere along its length. The cable is a more rugged, off the shelf product with a higher dynamic range in pressure response. Signals from each wire are buffered by a high-impedance operational amplifier and the footstep profile is being detected by an envelope detector circuit.

A custom made circuit scans 64 signals at 60 times a second. Every time a new pressure peak is detected, the circuit sends out a MIDI note with the note number corresponding to the particular wire generating the data, along with a 7 bit pressure value. As the subject lifts the feet, the pressure on a particular wire drops. This is also sensed by the circuit to turn off the corresponding music note that was switched on. Additional to sensing foot Doppler radars are also implemented to observe the dynamics and position of the body and arm swing. The Doppler-shifted reflections of a beam from a moving performer in its vicinity is gathered in a 4 element micro patch antenna. These radars respond to motion within a range of at least 15 feet. Instead of processing the Doppler signals in the Fourier domain, a simple analog signal conditioner produces three analog signals for each radar. These signals are sampled at 50 Hz and represented as 8 bits. The digitized data is used by a music generation algorithm and fed to a sound bank.

As the floor uses a grid of wires, it is a relatively coarse grained matrix when considering the translation of steps into a choreographic representation. The response of the entire system is rather slow, approximately 0.5 seconds. Although the floor is scanned at high sample rate, the effective responsive rate is hampered by the sensitive noisy wires and the bandwidth limits of the low pass filter that conditions the analog signals of the radar. The Magic Carpet cannot guarantee the location of steps if more than one step is present at an instance.

A LiteFoot [RPM95] Floor is developed to overcome these disadvantages. It has an area of $1.76\,\mathrm{m}^2$, made of a 10 cm thick wooden slab with a matrix array of optical proximity sensors. When a subject keeps the foot on the floor, the location of contact points between the floor and the foot is detected by the sensors. An embedded controller scans through the sensors every 10 ms. It considers each sensor position as a pixel and identifies changes in the pixels. The changes are then transmitted to the PC (Fig. 2.5).

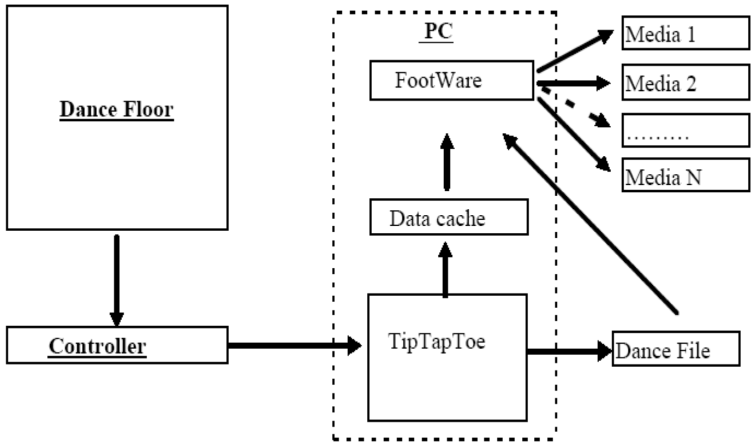


Figure 2.5.: Blocks of the LiteFoot

The LiteFoot Floor operates in two modes. In the *reflective* mode, the performer would have to wear a shoe that has light reflective soles. The footsteps are detected by the proximity of an object causing a reflection of light back to the sensor. In the *shadow* mode, the floor is flooded with light and the footsteps that stop light entering the sensors are detected. The software running on the PC displays where the feet are located. Another set of libraries map the floor positions and a simple MIDI device to a pentatonic scale. This resulted in a very sensitive floor subject to subtle inflections of foot movement. The dancer's steps became the music with the rhythm and tempo of the dance augmented by additional dimensions of pitch and timbre.

Tracking human motion and enabling applications to use the movements have gathered much attention in recent years. However the identifying footsteps and estimating its locations still remains a technical challenge. The methods suggested so far are pervasive in nature but have to be considered as a single piece of floor equipment with a predefined shape and is difficult to reinstall or transport to another place. Hence the design considerations started to demand modularity and flexibility in installation.

A new approach was needed following the footsteps of the Active Floor. Hence modular units with sensors were considered, which can form blocks of a compound system. In this new development, the modular blocks which could be interlocked or disassembled even with a new size or shape were designed. Each modular unit is required to provide sensing capabilities. In order to provide the widest possible range of information to the external application, the size of the sensor is reduced to 40 mm diameter. Twenty such sensors made of force sensitive resistors are fused to form a tile known as Z-Tile [RRF04].

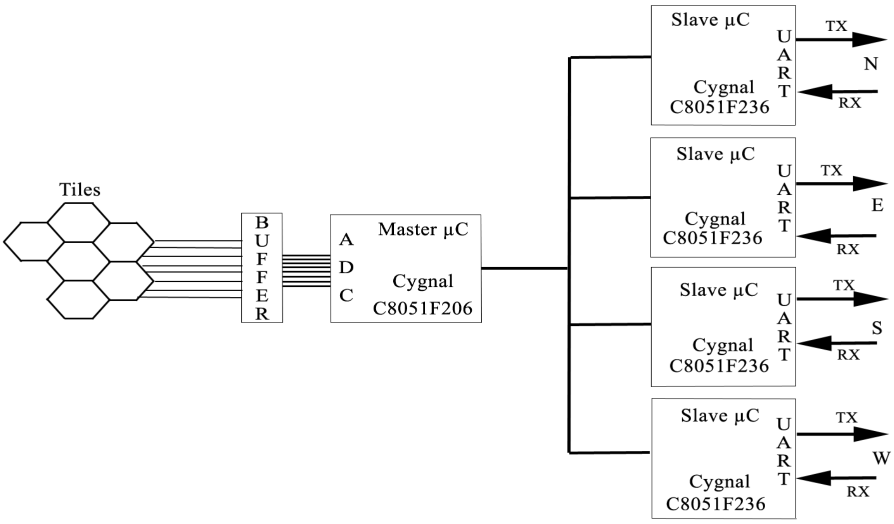


Figure 2.6.: Internal logical structure of Z-Tile

Each module has a unique shape that can interlock with each other in a regular pattern and can hold firmly. The modules are protected by a 2 mm plastic material on their surface. The control circuitry for each tile is contained within the tile casing and consists of 5 Cygnal microcontrollers in each tile. The controllers can read the sensor information, provide data communication and self organization for a network of interconnected tiles. The tiles communicate with each other using the universal asynchronous receiver/transmitter (UART) interface. Since the tiles can be interlocked, interfaces between the tiles eliminate wiring between the tiles. A light weight network protocol transports data to a central PC which governs the activities on the Z-tiled floor (Fig. 2.6). Wired connections offer the advantage of high speed, direct communication between the nodes in a network and a lightweight communication protocol can be used, as compared to the wireless counterparts.

The Z-tiles are targeted towards music and dance control applications. Pressure sensing can be used to give the musicians control over existing musical instruments. The tiles have been tested as an input device for control of computer games using virtual reality.

To emphasis on modularity and transportability Geta Sandals [KOC05] are developed in a intrusive way. Geta Sandals are targeted further in reducing system infrastructure and cost. This system can identify footsteps due to the sensors and computing devices installed in a wooden sandal. The system could also compute the location of the user. The sandal detects the heel strike and toe off stances. As the user walks from location A to B, he would leave a series of footprints. The system continuously measures a *displacement vector* formed between two advancing footprints. To track a user's current location relative to the starting point, the system sums all previous *footprint displacement vectors* (Fig. 2.7) leading to his current location.

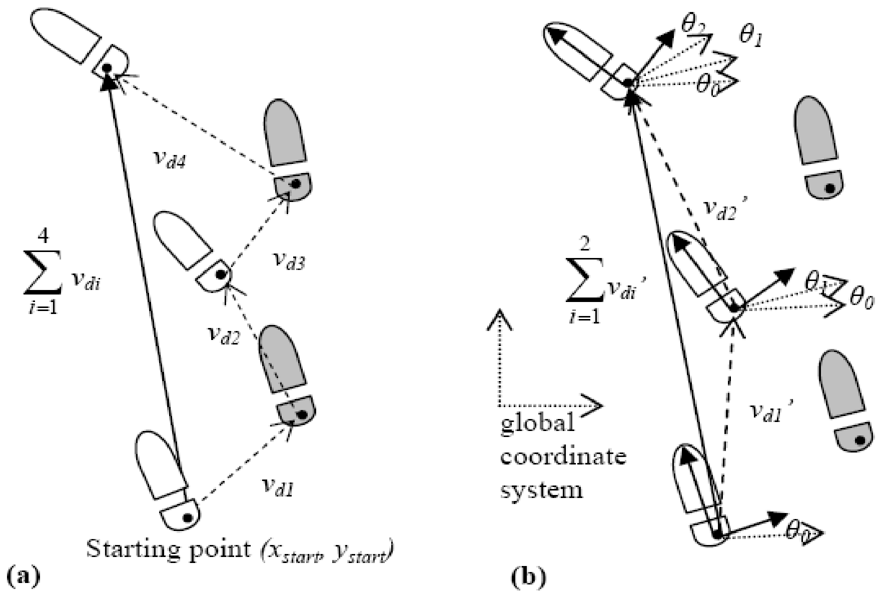


Figure 2.7.: Displacement vectors corresponding to footsteps

To measure the displacement vector for each footprint, two ultrasonic infrared combo receivers in the left sandal and two similar transmitters on the right sandal are installed respectively at the inner sides of the sandals. The coordinates of these two transmitters are measured relative to the local coordinate system of the left sandal. However, the sandals could not provide good position accuracy due to the signal interference of the two transmitters. Hence the receivers calculate incorrect coordinates. Secondly, the initial location of the user is assumed to be at a starting point which needs to be calibrated to the physical location of the user in a room. As the user walks any error introduced in the displacement vector gets accumulated over each step. To eliminate the signal interference problem, one of the transmitters was removed. To reduce the error accumulation in the displacement vectors, passive RFID tags are laid in the floor in a regular grid fashion. Each tag has a location information. Hence, as the user walks the displacement vector error is reset to the corresponding location of the RFID tag in the floor. However, this is the second experimental setup which is aimed in tracking human motion. The first one could be considered as the Active Floor which was more targeted towards human identification.

In a recent development towards tracking human motion, sensor blocks about 18 by 18 cm in size, consisting of binary sensors are integrated in the Human Tracker [TMI04]. The blocks are laid to cover an area of 37 m^2 to track human motion. The signals are then processed using Markov Chain Monte Carlo method to treat lossy data. The system could track a person within 58 cm of his true position.

The possibility of identifying footsteps has interested researchers to find methods to improve performance and feasibility of the final product. Summarizing the important works that have been developed so far, we could see where more attention has gathered. The following tree shows the developments over time (Fig. 2.8). Active Floor, Smart Floor, Ubi Floor are targeted towards footstep identification. Meanwhile Magic carpet, LiteFloor, and Expressive foot wear are targeted towards musical floors based on foot stamps at specified locations. But all of these are single block units which lack modularity and cannot be targeted towards a commercial implementation. It is because each of the systems need an expensive ADC board for a set of sensors (maximum 16) which is not economical and cannot be embedded. Geta Sandals and Z tiles do have a modular approach. Geta sandals offer the possibility of tracking people while Z tiles does not provide human motion tracking (there is no publication in this direction using Z tiles till the time this thesis was written). Geta sandals are intrusive in nature and the accuracy is largely attenuated by the accumulating displacement vector error. Human tracker is an improved version of Ubifloor with human motion tracking. Again the system lacks modularity. There is a huge gap between tracking human motion, a modular concept, and a commercial implementable solution.

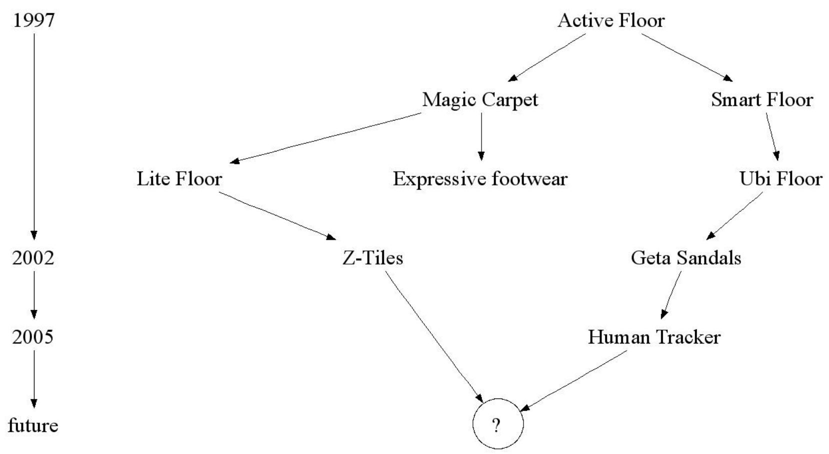


Figure 2.8.: State-of-the-art comparisons with respect to technological advance

It could be observed that Z tiles have developed towards a more robust, easily transportable, self organizing system to identify foot pressure. On the other hand Human tracker offers the possibility to track walkers at a tolerance of 58 cm. What is missing, is the gap in between, where there is a need for a system, that can be shaped to the size of the environment, still modular in architecture and easily transportable. The system should also offer the possibility of commercial manufacturability with components off the shelf. This would be the next step from research stepping out of the laboratories into commercial implementation. The system should not only identify footsteps, but also locate them in any event and provide means of tracking the footsteps of the subject in the targeted environment.

A Smart Carpet was developed to address these requirements.

2.2. Smart Carpet Internals

The Smart Carpet is made up of different layers of polymers, cotton and electronics. In the bottom layer is a black sheet of sturdy polymer to hold the above layers quite stiff. On top of this layer is a thin glue layer. The glue layer is double stick synthetic gum that glues the upper and the bottom layers firmly. On top of this layer is the conductive textile layer - textile fabric which is interwoven with copper wires (Fig. 2.10 on the following page). They are woven in a regular grid fashion. Small printed circuit boards (PCB) are integrated to this layer. The boards have electrical connection thru the copper wires that interconnect them thru this layer. The complete setup is then insulated by a transparent double sided hair crow layer. The velcro layer houses a normal floor carpet making it a uniform structure of textile with sensors of computing capability.

2.2.1. Hardware Construction

A capacitive sensor was constructed by placing a thin insulating material between two pieces of a 225 cm^2 conductive foil. This sensor is placed on top of the conductive textile layer. The plates of the capacitor are terminated to a (General Purpose Input/Output) GPIO pin of the microcontroller. Additionally the microcontroller has 4 UARTs. The H8S2249 microcontroller with 256 KByte of in-build RAM is used. The microcontroller with additional circuitry for power distribution and operation is built in a thin protected PCB of $5^2\,\text{cm}^2$ forms a node (Fig. 2.9).

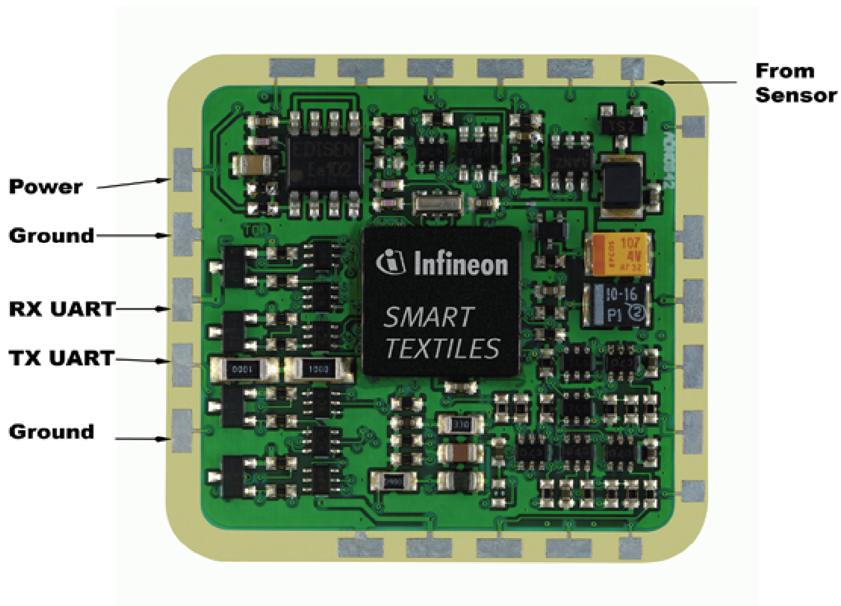


Figure 2.9.: A Microcontroller with 4 UARTS

120 such identical nodes with sensors are serially interwoven at the UARTs to form a regular

mesh network. The interconnections between the nodes were recognized using conductive textile based wires. Similar wire fibers were provided for power and ground signals separately. A polyester fabric with interwoven silver-coated copper wires with a line resistance of 0.4 ohm/m was used as a textile cloth. One of the nodes from the network was terminated to a PC. A 12 V DC distribution circuitry powers a few nodes. These few nodes route power to the whole network using MOSFETs and a power routing algorithm.

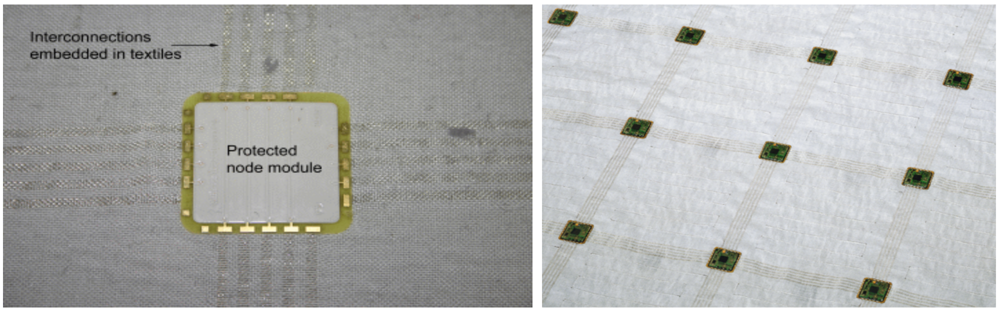


Figure 2.10.: Textile interconnect and the conductive textile layer

This algorithm checks for powered neighbors and open or short circuits. At each node, power consumption is approximately 10 mA per module in active and 6 mA per module in standby mode.

A special bonding adhesive called Anisotropic conductive adhesive (ACA) [Doe06] was used to integrate the electronic module to the copper wires in the textile. The ACA is a bonding material which consists of metal particles dispersed in an epoxy adhesive. This allows electrical current to flow along a single axis. The conductive particles are aligned under a magnetic field during curing, therefore, controlling the spacing of the metal particles within the material. Separately-spaced columns that conduct between opposing conductive surfaces are formed without requiring pressure. The ACA bonding is a low-temperature process (around 120 degree Celsius) compared with soldering. Hence, it is possible to interconnect the textile wires and the solder pads on the node module (Fig. 2.10).

Since the textile wires are interwoven in as a single fabric, by integrating the nodes on them provided a single piece of a smart textile cloth. This piece of cloth is cold-laminated between two layers of textile material to reduce the mechanical stress and equalize the height of the nodes with casing (2 mm). The top layer is a tufted carpet. The bottom layer consists of a 1000 gram polymer material. The size of the carpet is 240 by 200 cm. The power consumption of this network with 120 nodes was 8.4 watts.

Since each node in the network has four UART's, data communication is possible in all four directions sequentially. Communication between the nodes was performed by an algorithm CORP [Sav04], based on the serial communication protocol. In this algorithm, UART's are chosen based on the current requests and priorities at a node. The priorities of the UART interrupts are rotated in a regular cyclic way so the whole network has equal probabilities of sending and receiving data at any instance. The protocol does a software handshaking before it starts to send the data. Data is transferred dynamically using an inbuilt dynamic data transfer controller (DTC) [Ren03]. The protocol also supports bidirectional and fault tolerant data transfer.

2.2.2. Software Description

Currently, one node is interfaced to the RS232 port of the PC at a data rate of 115200 bps. A self organizing network operating system, ADNOS (Algorithmic Device Network Organization System) [SS00, Stu00, FSS02] is built in the nodes. ADNOS takes care of certain network features that are quite crucial to the network, like fault tolerance, self booting, error reporting and sensor status. Certain parts of the carpet can be removed and the whole network can still be reorganized to work in a regular mesh topology. The carpet undergoes a series of phases , until the whole network becomes operational (Fig. 2.11).

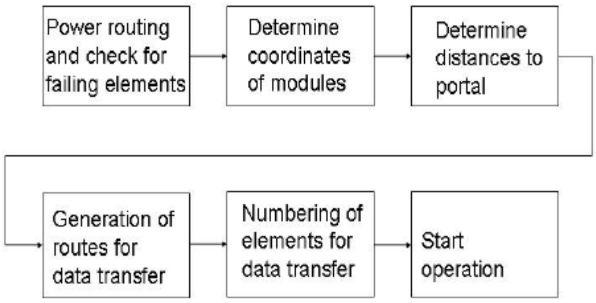


Figure 2.11.: Stages of software flow

Power routing is the first phase, where starting from the PC, each node checks its neighbors for current limits, disables defective neighbors and switches the power through to all functional nodes until the whole network is powered up. During the second phase, the network is organized. A boot frame is passed from the PC which contains the initial coordinates for that particular node interfaced to the PC. Based on this information and the number of available neighbors the spatial coordinated for the neighbors are calculated and the neighbors are booted by this node. No prior knowledge of the spatial coordinates is used except from the ones provided in the boot frame.

The orientation of the nodes is also determined using the reference to the UARTs. In this fashion the boot frame is recalculated at each node as it travels till the end of the network. As the frame passes on, the distance to the PC is calculated at each node using hop counts. With this information a backward organization algorithm performs data route determination at each node. Hence each node has a specific data routing path to send its sensor signals and error codes to the PC. Then the PC issues a static identification number also referred to as the processor number to the node to which it is terminated. This node with the knowledge of all its neighbors, calculates the processor numbers for its neighbors. In this way, each node on the network has its spatial coordinates identified by a unique processor number. After this phase the network in the carpet is ready for operation.

2.2.3. Operation

The whole network is kept alive by constantly sending ping frames between the nodes using the data routes calculated during the initializing phase. The ping frames originate from the PC at the rate of 5 seconds which is also configurable. If there is a failure in the ping message, the PC is informed about the exact node where this message failed. If there is any electrical failure at a node or if there was any physical tear at a place on the carpet which led to wearing off of a node, either the interrupts at the neighbors get triggered or failure of consequent ping messages generate a deactivation frame to the PC with its identity.

A monitor routine runs on the PC which takes care of displaying the status at each node, the structure of the network, and all the administration tasks of the ADNOS. Figure. 2.12 shows a snapshot of the SCM in the operation mode. This monitoring routine records each frame that is sent and received from the carpet. At each instance a time stamp of the transmission or reception is also recorded. This data is presently available as a text file referred to in the rest of the chapters as the log file. When the whole network is organized, the carpet is then activated by sending an activation frame. After this frame is being received at the end of the nodes in the network the carpet can be used as a smart sensor floor.

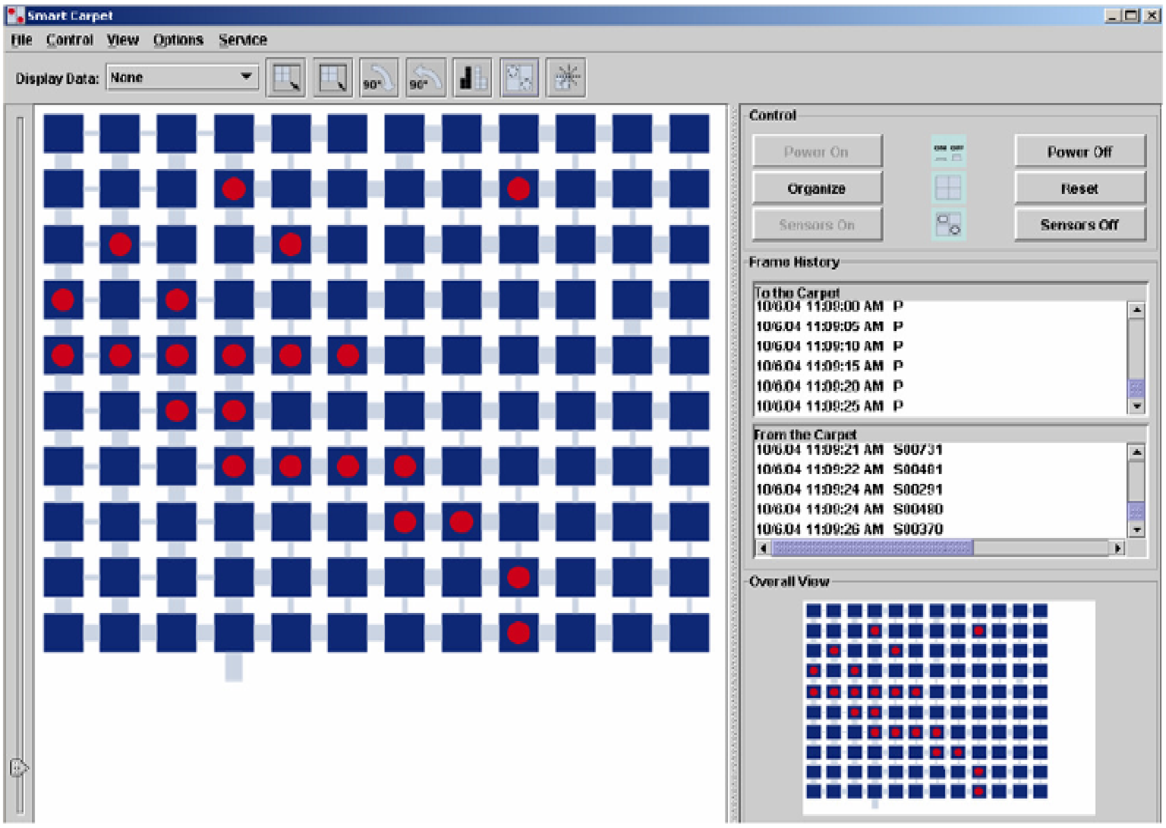


Figure 2.12.: Snapshot of the Smart Carpet Monitoring routine

When a person steps on the carpet the capacitive sensors get activated due to the foot pressure and send a sensor ON frame to the PC along with a processor number. When the person moved or kept the foot off from that particular sensor, a sensor OFF frame is sent from that node to the PC through the data path calculated during booting.

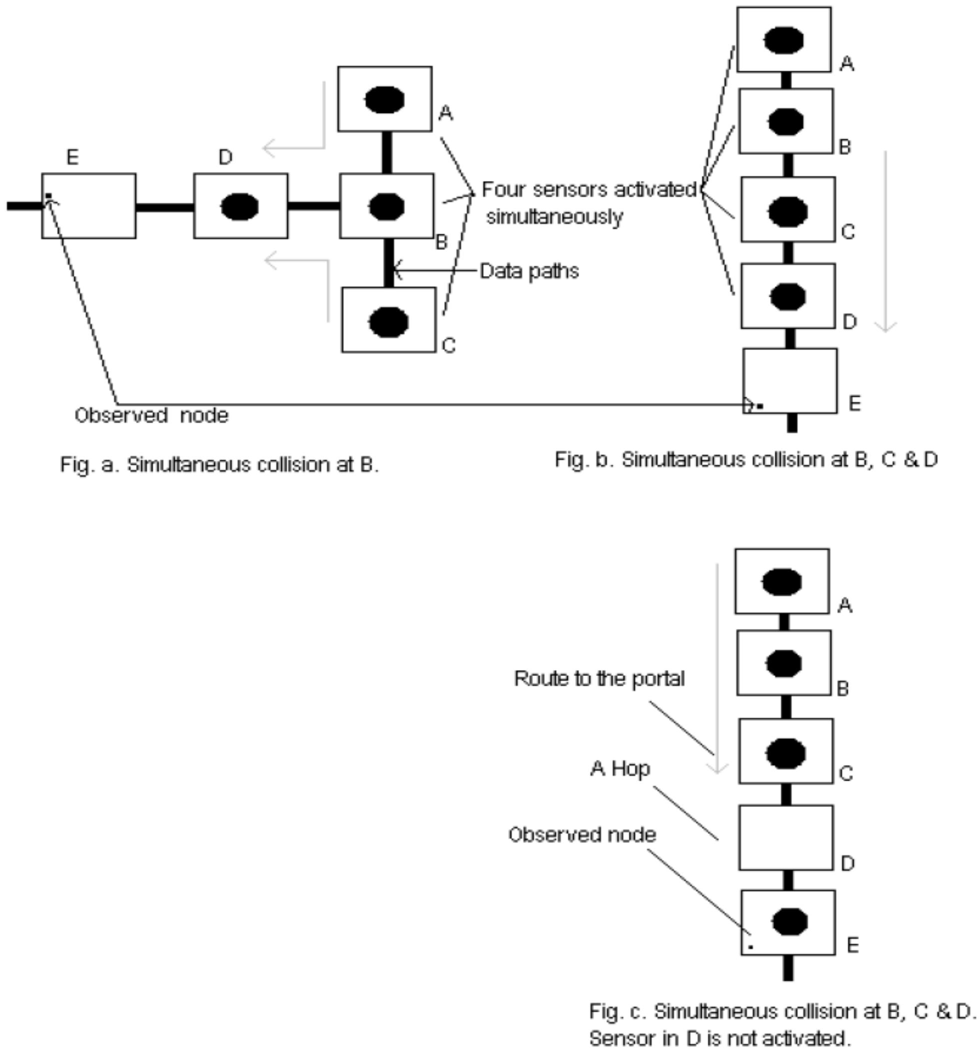


Figure 2.13.: Test configuration for observing hop delay and data collision

2.2.3.1. Performance

As the PC receives each frame, a time stamp is recorded along with the frame in a log file. The time taken for one sensor information to be recognized, processed and transmitted to its neighbor takes approximately 1.5 ms. Figure 2.13 shows configuration details and results of data exchange during hopping and collision. As the frame tries to find its way to the PC, it passes at a speed of 1.5 ms at each hop. When two nodes need to send data simultaneously to each other data is mutually exchanged on a negotiation (Fig. 2.14 on the following page).

This avoids collisions and increases throughput.

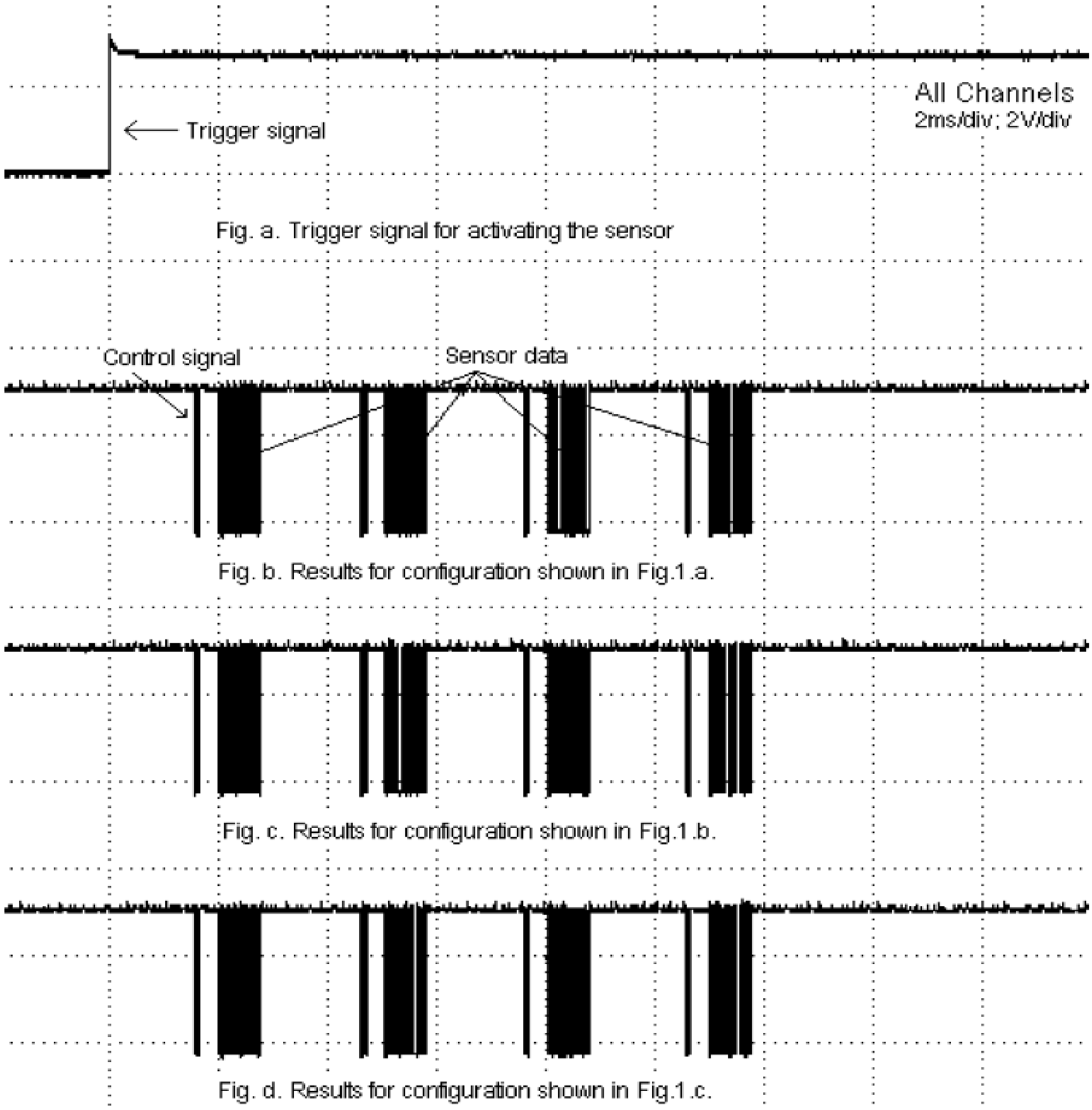


Figure 2.14.: Oscilloscope snapshot showing data exchanging avoiding collision

The typical area occupied by a normal person while walking is 2.3 m^2. Comparing this to the traffic that would occur in a network of 120 sensors spread across an area of 7.6 m^2, we can expect a maximum of three people walking without touching each other [Fur71]. They would activate a maximum of 24 sensors per second (6 feet by 4 sensors per foot, if their foot lands at borders of two sensors).

If there is a stampede a maximum of 6 people could congest and if they are running, they would trigger 48 sensors per second (12 feet by 4 sensors per foot). If tap dancers are dancing on the carpet, three dancers in the given area, they would generate 480 sensor packets (6 feet by 4 sensors by 20 taps per second). These situations are simulated on the carpet for the given area. Results of the measurements taken are shown in Fig. 2.15. They show that, the Smart Carpet is able to effectively handle data packets with minimal losses even at extreme situations(Fig. 2.15).

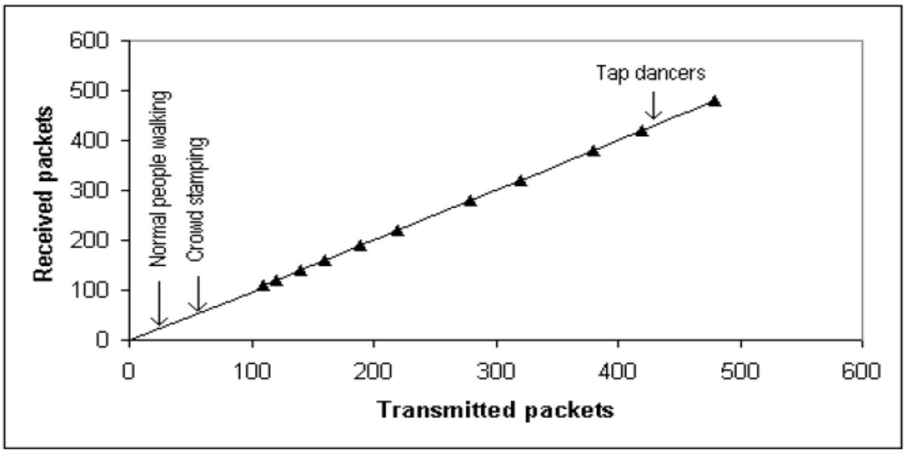


Figure 2.15.: Transmitted vs received packets

Z tile has done exclusive research on reducing data loses in smart sensor floors. Their solution was to eliminate data loses by data blobbing [Ric04]. This method improves network performance and increases the throughput. Similar principles can be applied to the communication protocol and the messages delivered by the Smart Carpet. Another important factor that contributes to the data loss is the operating system on the PC. Unsynchronized processes occurring at random intervals do cause delays on threads running on the RS232 protocol. Secondly, the GUI that is getting updated also proximates delay in the threads synchronized with the RS232 interface at the PC.

2.3. Thesis Motivation

Considering human motion estimation using video analysis techniques, most of the proposed studies involve extracting parameters from the pose of the human body and represent them as kinematic chain of parts [BM98, HSS02, RR97] or as spatial arrangement of blobs [WP98] or point features [YSP00]. A general overview of such analysis [Dje05] can be brought in under a framework for tracking human motion based on four main tasks [OB80]: prediction, synthesis, image analysis, and state estimation. However, these tasks can be true for camera based tracking approaches where the prediction task considers previous states up to time *t* to make a prediction for time *t+1*. It allows the integration of knowledge into the tracking process. The synthesis component translates the prediction from the state level to the measurement (image) level. The synthesis allows the image analysis task to selectively focus on a subset of regions and look for a subset of features. Finally, the state estimation task computes the new state using the segmented image. This framework can be applied to any model-based tracking problem, whether involving a 2D or 3D tracking space. However, they suffer from the fact that as number of cameras increases, the complexity of processing all the videos and images increases, too.

Secondly, video analysis cannot guarantee exact location of a person in spatial coordinates. When he or she is not cooperative or if the shape is modulated while carrying a load for example, the subject's silhouette is modulated. The shadows need to be extracted from the neighbor or a dark object behind the subject. The camera field has to be mapped to fit the area of the observation. This means for each camera the mapping has to be modified which is also dependent on the location and surroundings. Feature extraction is affected by the lightning effects on the camera and the perspective distortions. Ubiquitous computing had opened new possibilities in sensing and processing observed data. Extensive research had been conducted in this area where sensors based on state-of-the-art materials are embedded in diverse environments to sense foot pressure and identify foot steps. However, most of the systems except for Z tiles lack a modular design. None of the systems are available for commercial productivity. Very few of them provide footstep tracking which is of major interest for security and health care segments. Hence tracking people using natural means is still an unsolved problem.

Table 2.1.: Table of comparison for state-of-the-art of pervasive environments and human motion tracking

Project	*Identification*	*Tracking*	*Algorithm*	*Accuracy*
Active Floor	Yes	No	Standard Mean	50 %
Magic Carpet	Yes	No	Proprietary	N.A.
Lite Foot	Yes	No	Proprietary	N.A.
Smart Floor	Yes	Yes	HMM	93 %
Z Tiles	Yes	No	Proprietary	N.A.
Geta Sandals	Yes	Yes	Proprietary	error 158 cm
Human Tracker	Yes	Yes	MCMM	error 59 cm

2.3.1. Challenges

The table 2.1 summarizes the different innovative research attempts towards a pervasive environment intended to sense foot pressure and identify footsteps. However, a smart floor system should have the properties of self organizing, less power consumption and sources, detecting newly added parts of the floor or the shape of the floor space. It should be fault tolerant, e.g. able to detect when some of the sensors got damaged, should be able to recognize a best route for data to reach an external controller like a PC. It should be easily transportable, commercially marketable and easily manufactured. All of these above properties qualify the Smart Carpet.

Although the Smart Carpet does offer the possibility to locate a footstep at any particular location, a new set of rules has to be identified and realized as an algorithm to identify the sensor activations as a person walks on the carpet, detect the sequential footprints of a person and identify the spatial coordinates that can be mapped to a physical location in a room. It should be able to identify time at each instance of motion. The rest of this thesis is contributed to this part of the research where the data from the Smart Carpet is explored, analyzed and the results are discussed.

2.4. Contribution of this Dissertation

This research attempts to analyze the data that is generated from the Smart Carpet. It tries to empirically model the nature of a footprint on the Smart Carpet. It attempts to build data mining models that can use the structured data from the Smart Carpet. The models are adapted to patterns of actions rendered by human motion. The models deliver a set of spatial coordinates and time stamps that describe the trajectory of the human motion on the carpet. This thesis also presents novice methods to validate the models. The models are compared with motion patterns a subject took while walking on the Smart Carpet and the state-of-the-art technologies. The models together with the Smart Carpet system are targeted towards a most accurate human motion tracking system.

2.4.1. Data Definition

Before we go into the details of the models, the data that is generated by the Smart Carpet is carefully examined. The Smart Carpet consists of touch sensors which are capacitive in nature. These sensors are terminated to the GPIO pins of a microcontroller. When a person keeps the feet on the carpet the sensor triggers an interrupt signal in the microcontroller to intimate the presence of foot pressure. On reception of this interrupt signal, the microcontroller sends a sensor ON packet to the Smart Carpet Monitoring (SCM) computer. The SCM programs identify the sensor signal and display an activated blob at the location of the micocontroller in the carpet. This data is then recorded in a log file along with a time stamp in the SCM. When the subject lifts the feet to move over the carpet, the sensor again interrupts the microcontroller which in turn sends an OFF signal. This data is recorded in the log file. The models described in the subsequent chapters use the recorded data from the log files to identify the patterns of human motion. The data available from the Smart Carpet consist of foot step activations and deactivations along with time stamps recorded at a single point of entity, the SCM computer.

2.4.2. Modeling

A systematic approach is considered in designing the models and treating the data. This approach is derived from standard data mining techniques. Figure 2.16 on the next page shows the flow of operations during the entire search phase for an efficient data mining algorithm. Although different data mining algorithms are examined, the overall process remains the same, except for the methodology applied in treating the data in a model.

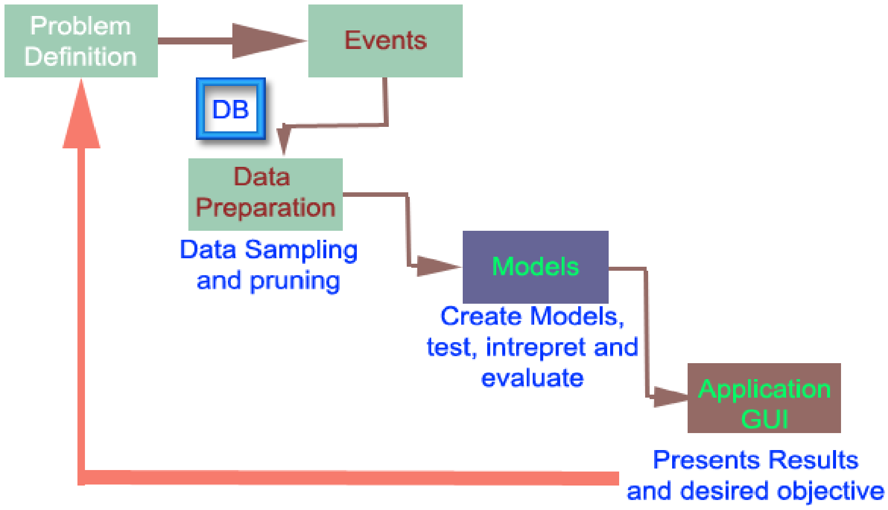


Figure 2.16.: Data mining process flow

As the subject walks on the Smart Carpet, the sensor activations are recorded in the log file. Simultaneously an independent source keeps track of the events as the person moves from one location to another. The data recorded by the independent source has to be error free and has to truly reflect each footstep in its accuracy of location and time. This reference would be used to evaluate the performance of the models.

The recorded data is then examined for consistency and a specified syntax is formed that can serve as input for the models. The result of the models and the reference are then compared. A common format has to be agreed on for this comparison. The result is observed for its significance. If there are ways for further improvement in the calculations for example; if there is a possibility to improve the performance using another parameter, the model is then further refined. This process is done until an creditable result of observed. Different models can be created by implementing the mathematics in the model block. Hence, a common framework is maintained for different methods of mathematical exploration in data mining.

2.5. Road Map

The precluded chapters and sections have been intended towards a brief introduction for identifying human motion using pervasive methods. The methods studied so far have been briefly mentioned. The pros and cons of different observations have been explained. However, the transfer phase between laboratory experiments and industrial productivity lacks a modular implementation in tracking human motion. To increase the performance and enhance commercial productivity the Smart Carpet has been described sequentially. A broad description is presented in exploring the data that comes out of the Smart Carpet. Data loses in the Smart Carpet need to be addressed possible approaches have to be outlined. Data needs to be structured before an algorithm can process it. This structured data has some patterns which can contribute immensely for building a relevant model. A set of models based on the nature of data are presented in chapter 3. The models implement data mining techniques to process the structured data. These models are reasoned, a detail description of each model, its implementation, possible effects are predicted and state of art technologies relevant to the nature of data from the Smart Carpet are discussed.

Techniques and drawbacks for evaluating the results generated by the models are quite important in assessing the developed models. The built models have to be evaluated by an independent source. Possible approaches for developing an independent source with reasons are discussed in chapter 4. Advantages and disadvantages of each approach are briefed. A suitable reference method is chosen and reasoned. Methods used to generate a reference, must produce results that can be directly compared to the estimation of the data mining methods. These results must contain analytical data. Challenges faced in evaluating the generated results and the reference sources are discussed in Chapter 5. These results are then compared with the state-of-the-art results. Each model is built based on certain assumption on the nature of the data and the patterns found in the pruned data set. The results of the models are discussed in Chapter 6. This chapter condenses the work presented in this research. It also identifies some areas that consent to improve. The words model and methods are frequently used as a synonym for a procedure which implements the data mining algorithm. The words, walk, trail, path and trajectory are used inter changeably to mean the motion of a subject by his or her feet.

2.6. Summary

The things around us that can make decisions and offer services by observing our actions have been briefly described in the early chapters of this work. The driving interests of some leading research institutions have focused research in Ubiquitous computing. Textiles - the harmless friend we embrace over the centuries seem to be a native target for penetrating intelligence in our daily activities. Possibilities of services that can be rendered by observing human motion are enormous. Pioneering experiments are carried out towards this goal of observing human motion by smart means. However the lack of a modular development and commercial production techniques paved the way for a Smart Carpet. By treating the data from the Smart Carpet with data mining techniques, location and motion of subjects as they walk on the carpet can be accurately plotted. In the next chapter, the data that is vital for identifying the footsteps is briefly dealt. It is then followed by the different models that can be used to treat the data to plot a trajectory of human motion.

3. Describing Sensor Data in Models

The Smart Carpet is initiated into active mode by the host PC, a normal personal desktop computer running Windows 2000. The host PC sends small packets of data to the carpet. The packets contain commands to initiate different phases of the carpet in a sequential manner. The size of the data packets varies based on the command and its arguments that are sent to the carpet. In the initial phase the carpet is organized with dynamic addresses for each node. Based on the node which is terminated to the host PC, spatial coordinates are calculated at each node. This phase determines the data route between each node and the carpet. After this phase, the carpet is put on an active mode were it can sense footstep activations. In the active mode, the status of each node is identified by a ping command or the "P" frame, that is sent in regular intervals from the host PC. In between the ping commands is the sensor activation data that comes from the carpet. This data is packed in a frame which has a header, followed by the data and then a footer.

3.1. Data Management Considerations

These frame transactions are recorded in the log file, instantly with the time stamp. This log file serves as a primary data, that is available for the data mining models, which are presented in this chapter. In the first phase of the experiments, the log file is read in off-line mode, where the subject had finished the walk. This requires the models to read the network information from the log file and re-simulate the network layout, which was available when the subject had walked. Then, it has to read sensor signals without the ping frames or any other frames that had been exchanged between the carpet and the host PC.

3.1.1. Data from the Carpet

Each node passes its identity and its location along with information about its neighbors to the host PC. This information is stored in the "V" frame (Fig. 3.1). Details of various other frames used in the exchange are briefed in the description over ADNOS [SS00].

V - Header	Orient Offset	Y Coord	X Coord	Distance to host	Source ID	Throughput
Processor Number	Port 0 Data	Port 1 Data	Port 2 Data	Port 3 Data	Footer	

Figure 3.1.: Structure of V frame

The V frame has a heavy payload. Each node calculates this V frame based on the initialization frame it had received from the host PC. The nodes send only one V frame to the host PC although it would have been initialized by four of its neighbors. The host PC is not involved in calculation of the parameters in this frame rather it just initiates the calculation at one node. This frame contains the header followed by the orientation of the node with respect to the north of the carpet. This north is identified while the carpet is produced. It also contains the spatial coordinate information which is calculated by the node after it was initialized by the host PC. As the initialization frame was passed to this node, it hopped over some processors. The hop count is then sent to the current node by its neighbors. Sometimes the node might have been directly terminated to the host PC. In this case the hop count would have been zero. The lowest hop count is used to calculated the distance to the host and the throughput. The neighbor who initialized with the lowest hop count also becomes the source for the current node. Based on the hop counts and the current spatial coordinates, the processor number is calculated at this node. Similar data from the neighbors are gathered and packed in the V frame and then sent to the host PC. This data is vital for recalculating the entire network.

The methods that would implement the data mining algorithms should have the possibility to read this V frame and simulate the network which was available at the time the log files were created. Once the network is generated, the ping frames have to be discarded. This can be done while processing the sensor frames, too.

S - Header	Processor Number	Flag	Footer

Figure 3.2.: Structure of Sensor frame

The Sensor frame is a very light weight frame with the structure shown in Figure 3.2. The flag field contains the status of the node, if its sensor was activated or not. If the sensor is activated the flag shows 1 and a 0 when someone lifts the foot off this particular sensor. The methods have to read this sensor data, interpret the frame to identify the processor number and the status of activations. Taking a close look at this log file, the methods have to have a plug-in which does the following operations, before the data mining algorithms are applied;

- Read the V frames
- Reconstruct the network
- Ignore the Ping frames
- Identify the Sensor frames
- Check for activation and deactivation flags
- Retrieve the processor number and spatial coordinates of each Sensor frame

From the processor number, we derive the spatial coordinates of the sensor stored in the log file in the V frames. As the Sensor frames reached the host PC, each frame was recorded with a time stamp. The time difference between activation and a deactivation of a particular Sensor frame would be equal to the time a person paused on that sensor. By using the average time differences between two sensor activations, the speed of the person's walk can be calculated. The spatial distance between two sensor activations would contribute the stride length in the walk.

3.2. Data Pruning

When a sensor gets activated or turns ON, an ASCII 1 is added as data to the end of the sensor frame. When the sensor turns OFF, a 0 is updated in this field. As a sensor turns ON there could me more than one sensor that turns ON at the same or nearby instances. A sensor ON frame is not immediately followed by its OFF frame in most cases. Figure 3.3 shows at the first time instance (column id 0), sensor 20 got activated and its deactivation frame reached the monitoring software at the 6th time instant. The time is recorded in millisecond format in the monitoring software. In certain instances, the deactivation frame does not reach the monitoring software at all (the 7th time instance). Taking a close look at the sensor registrations in the log file (a snapshot is show in Fig. 3.3), data registration in the log file is not consistent to the sequence of actual events; rather it is dilated by the traffic and the responsiveness of the operating system on the host PC. This necessitates data sprucing. The spruced data has to be retrieved often and recalculated. A stable database management system is required. The algorithms are intended to be developed using Java. Hence, a Java friendly DBMS was appreciated. HSQLDB or MySQL RDBMS were good candidates. The former is a 100

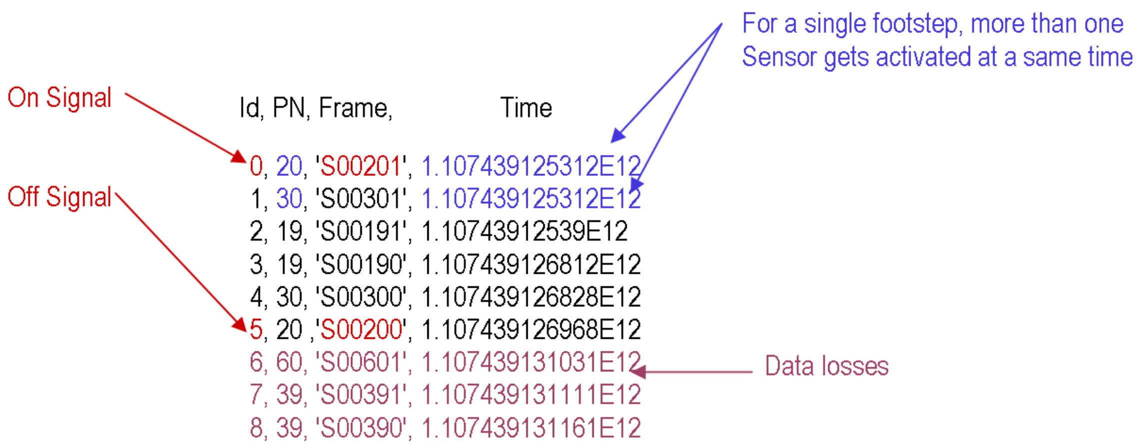


Figure 3.3.: Snapshot of the sensor data in the log file

The log file is read and the useful data is extracted from the file and stored in the database tables. The tables were ordered based on sensor activation and its corresponding deactivation flags, with time stamps. Figure 3.4 on the following page shows a snapshot of how cleaned data looks like in a base table. The spatial coordinates for the respective processors were extracted from the V frame's, X and Y coordinate fields. These entities are stored in a base table. Looking more closely the network inside the carpet did lose data because of the reasons discussed in the previous chapter. These missing holes were also stuffed with duplicate data that were derived based on probability of existing neighbors around the holes.

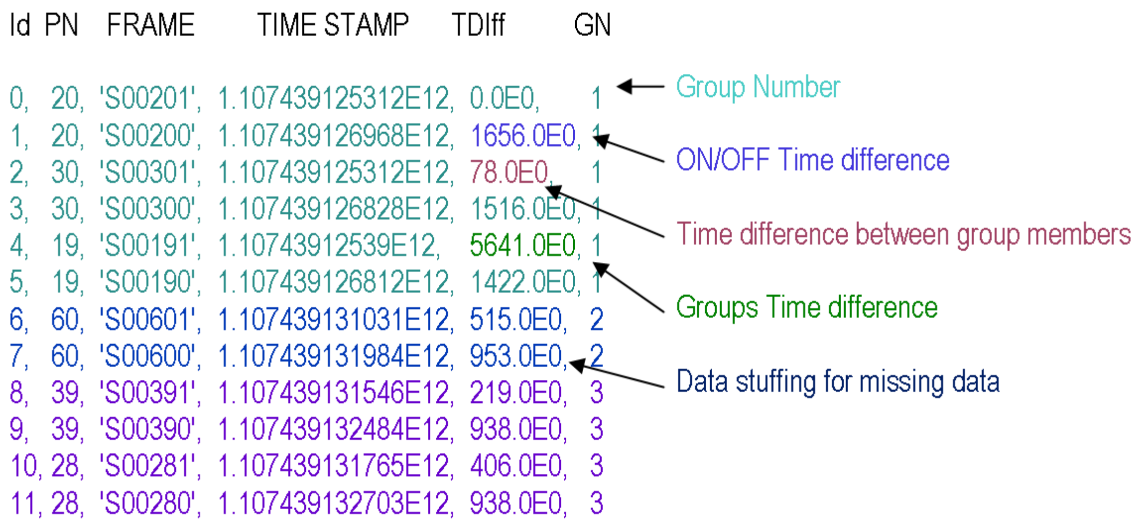


Figure 3.4.: Cleaned data ordered with time difference showing group numbers

3.3. Walking Patterns

The cleaned data of this example presents certain interesting features. For each foot press on the carpet, more than one sensor was activated. There existed a distinctive time difference between a group of sensors that belonged to one foot press and the other. These sensors are located under a radius of 15 cm. This explains the fact that there exist clusters in sequence of activations that represented a person's walk. These clusters have date for each foot step. A cluster centroid of this cluster would be then the person's approximate foot center while the midpoint between two centroids would be an approximate center of a person's body. An interconnection of these midpoints would determine the actual trajectory of the person's walk. The most time consuming task was to identify these clusters. When more than one sensor got activated at a same instance, some belonged to a person while some were caused by a sensor malfunction or when there was an obstacle dropped nearby or when the sensors were activated not within the radius of the feet but at the same instance. It was challenging to identify the cluster outliers and centroids. One of the main contribution of this thesis, was to identify methods that can cluster footprints and classify the methods.

Since clustering data is a traditional data mining task, this thesis falls under the broad spectrum of data mining research in embedded sensor networks. Although several established clustering algorithms exist, an appropriate one for clustering sensor data with a small amount of available sensor data has to be identified. Then, the algorithm has to be fairly verified. Most of the clustering algorithms classify clusters using an estimated parameter. Alternatively, a series of conditions could also satisfy a cluster. In the data mining world, this algorithm, which governs the tasks of clustering, is often referred to as the model [MR05]. A model briefly describes the characteristics of the data in the database, although not that big as the database itself. It can be used with other databases or datasets of similar characteristics to determine the patterns in the datasets.

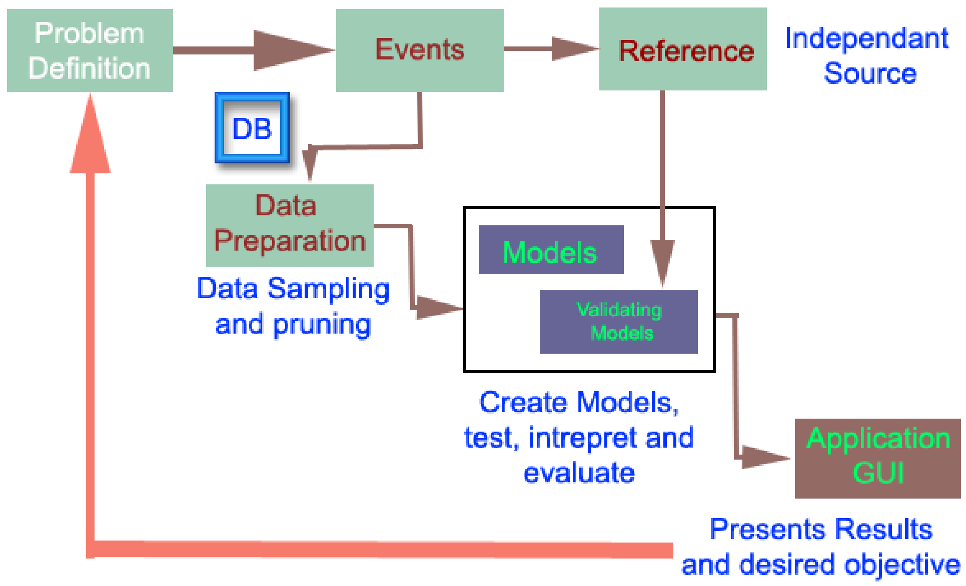


Figure 3.5.: An overview of the data mining process on the log file

3.4. Model Features

The data that is fed as input to the model has to be pruned. Figure 3.5 shows a brief overview of the processes involved in generating the trajectory of a persons walk based on the data generated by the Smart Carpet. As models process the data, the results of the models are evaluated. There would always be some improvements that can be made to fit the model more precisely. A perfect ideal model never exists. The deviation of these results in the models would explain its accuracy. Hence an independent source can observe the data as the log files are registered with entities from a person's walk. A systematic structure has to be maintained in the development of the models that offers the possibility of fairly disciplined changes. It was decided to separate the model files from the rest. Hence each process was considered as a module of object oriented classes with individual tasks represented in the above overview. Each model can be split further into a set of objects and classes that implements various algorithms that cumulatively describe the characteristics of the model.

Table 3.1.: Description of CARPET table in the database

Field	Type	Null	Key	Default	Extra
procnum	int(11)	NO	PRI	NULL	
yCoord	int(11)	YES		NULL	
xCoord	int(11)	YES		NULL	
through	int(11)	YES		NULL	

3.4.1. The Model Input

The walk of the person is represented as the events in the block diagram (Fig. 3.5 on the preceding page). The events contain footsteps of the subject which are in turn represented by sensor activation and deactivations. This data is available along with frame time stamps in the log file. The spatial coordinates of each node in the network are also available in the log file. The model needs an algorithm which can retrieve the complete network details and store them in a database. The details include the spatial coordinates and the respective processor numbers. The ADNOS system takes care that each node in the network gets a unique processor number and a spatial coordinate. The coordinates represent the location of the nodes with respect to the working neighbors at the initialization of the network. Later, when the Sensor frames are encountered before passing it to the models for determine the trajectory, the frames have to be read for the processor numbers. An algorithm has to search the database for the spatial coordinates of the corresponding processor number. To store and retrieve these data, a table with complete network information is created in the database. This table, referred to as CARPET table, is frequently used to recover the spatial coordinates of the Sensor frames. The table is created while reading the log file for network information.

Each network 'V' frame entity in the log file corresponds to a node. For example if there are 180 V frames in the log file, each node would have sent a frame and this explains that there were 180 active nodes which formed the network. The *procnum* field in the CARPET table (Tab. 3.1) corresponds to the processor number of each node in the network. This field is not auto incremented and serves as a primary key for this table. This field cannot be null and has to have a non duplicate processor number. The *procnum* field is of integer type of size 11 bytes. This field can be compared to the IP address of a node in its network. The fields *yCoord* and *xCoord* store the Y and X coordinates of the particular processor number. These three fields are unique fields which contain the same information present at each node while the network was created and operational. In the following tables that would be created for processing the sensor information, the *procnum* field is used as a selection criteria for retrieving the spatial coordinates of the given processor number.

Each sensor frame ('S') entity in the log file corresponds to a sensor activation or a deactivation. As a subject walked on the carpet, the events that were registered during the walk are represented in the BASETABLE. The *id* field in the BASETABLE (Tab. 3.2 on the following page) corresponds to the number of events during the walk. This field is auto incremented and serves as a primary key for this table. It cannot be left empty, but it is allowed to start with

Table 3.2.: Description of BASE table in the database

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto inc
procnum	int(11)	YES		NULL	
tFrame	varChar(256)	YES		NULL	
tStamp	double	YES		NULL	
diffTime	double	YES		NULL	
groupNum	int(11)	YES		NULL	
yCoord	int(11)	YES		NULL	
xCoord	int(11)	YES		NULL	
weight	int(11)	YES		NULL	

Table 3.3.: Description of CLUSTERTEST and SORTED tables in the database

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI		
procnum	int(11)	YES		NULL	
yCoord	int(11)	YES		NULL	
xCoord	int(11)	YES		NULL	

zero as its first entity. It is followed by the *procnum* field which has an integer type of size 11 bytes. This field stores the processor number of the node. This field is retrieved from the 'S' frame. The $tFrame$ field stores the 'S' frame upon its arrival at the host PC. It contains the processor number and the status flag which indicates an activation or a deactivation of the sensor corresponding to this processor number. As the frame arrives at the host PC, the time stamp is registered in the log file. This time stamp is recalculated to the millisecond format and stored in the *tStamp* field. The field $diffTime$ calculates the time difference of two consecutive time frames. The field *procnum* is used as a condition on the select statement, while searching for the spatial coordinates on the CARPET table, which fills the fields $yCoord$ and $xCoord$ - the spatial coordinates for this particular processor number.

The input for the models is currently available as raw data in the database. This data has to be cleaned before the models process them. Hence two tables CLUSTERTEST and SORTEDTABLE are used to rearrange the data.

The *id* field in the CLUSTERTEST and the SORTED tables (Tab. 3.3) represents the events and they are identical to the events in the BASE table. This table contains only the activations and the deactivations of sensors. It is obvious that each activation frame has to have a corresponding deactivation frame sooner or later. The CLUSTERTEST stores the data as it is registered in the log file. But in the SORTED table, data is so arranged that each activation frame is followed by its corresponding deactivation frame. If there are any missing activation frames, the time stamp is taken from the deactivation frame. Hence the difference time for this particular sensor becomes nullified. This is repeated in the case when a deactivation frame is missing for an activation frame.

The complete process of creating the database tables, storing, retrieving and sorting the data is done before the data mining tasks begin. The basic information that can be given as input for the models are,

- p_n - the processor number
- X_n, Y_n - corresponding spatial coordinates
- t_{ON} - active ON time of the sensor
- t_{diff} - the time difference between two consecutive sensor activations

This data is now available in the tables. The data mining algorithms can use this readable data to interpret the hidden trajectory of the subject's walk.

3.4.2. The Model Output

Briefly studying the log file, it was observed that each foot step activated more than one sensor. As a result two or three Sensor frames contributed to a footstep. The data mining algorithm in its first task should be able to group these frames to represent a footstep. In data mining terms, *clustering* is a technique which applies to a case when there is no classification of instances, but rather when the instance are to be divided into natural groups [WF05]. These clusters presumably reflect some mechanism at work in the domain from which instances are drawn, a mechanism that causes some instances to bear a stronger resemblance to each other than they do to the remaining instance. Clustering naturally results different techniques to the classification and association learning methods. This resemblance is seen in the time difference field t_{diff} and the radius of the activated sensors. This explains the fact that the sensor activations need to be clustered and the clusters represent a foot print. Considering a single sensor, its state can be represented as:

$$S_n = S_{n(ON)}, S_{n(OFF)} \tag{3.1}$$

Due to the resemblance of few sensors based on certain conditions, a cluster can be defined as:

$$C_n = \{S_1, S_2, S_3 \cdots, S_n\} \tag{3.2}$$

A person's position on the carpet can be represented by describing the center of the body in spatial coordinates. The trajectory of motion can, then be described as the line joining these coordinates. But, it is observed that more than one sensor represents a subjects foot. Which means, a cluster of spatial coordinates has to be refined to its center. The *centroid* of a cluster is the average point in the multidimensional space defined by its dimensions. In a sense, it is the center of gravity for the respective cluster. A cluster can have one or more sensors in its member list and the list varies depending on the location of the foot and the availability of the sensor edges under the foot. Hence, the size of a cluster cannot be determined in advance. The cluster size has to be used in determining the centroid of the cluster. This method is also known as *weighted pair-group method using the centroid average* [SS73]. The centroid of a cluster $A(C_n)$ can be calculated using the spatial coordinates and the time differences of the cluster members. This can be given as,

$$A(C_n) = AC_{n(xy)},\, AC_{n(t)} \tag{3.3}$$

The equation (3.3) refers to the centroid of a single cluster. This could be either the left or the right foot. A subject's center can be represented as the midpoint (m_j) of two consecutive cluster centroids. A subject's walk would then contain a set of these midpoints with time instances. It can be represented as:

$$T(n) = \subset \{\, m_j, \frac{A(t)_j}{dt} \,\} \tag{3.4}$$

where

$$m_j = [A(C_j) + A(C_{j+1}) \cdots] \,/\, j \tag{3.5}$$

After the clusters, their centroids, and midpoints are identified, the data are stored in a table GROUPMIDS. This table describes the trajectory of the subject's walk on the carpet.

Calculations based on cluster centroids for the spatial coordinates are stored in the fields *yGroupc* and *xGroupc*. The corresponding time stamp for the centroid is also stored in the *groupTime*. The *groupNum* stores the cluster number. The spatial difference between two groups would give the stride length of the subject and the time difference between two groups can be used to calculate the cadence of the subject's walk. The fields *ymidc* and *xmidc* give the location of the subject on the trajectory. This point is the position of the subject on the carpet in a multidimensional space. This point is always referenced by the *ymidTime* field which describes the time instance of the subject at this point. These mid points refer to the position of the subject during his or her motion on the carpet. After the data mining algorithms calculate these midpoints, the information is available in the database as discrete quantities.

The motion of the subject in a multidimensional space can be visualized using the information obtained by the data mining algorithms. The spatial coordinates can be mapped to a scaled image for visualization. There must be a possibility to compare the actual data and the

Table 3.4.: Description of GROUPMIDS table in the database

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto inc
procnum	int(11)	YES		NULL	
tFrame	varChar(256)	YES		NULL	
yGroupc	double	YES		NULL	
xGroupc	double	YES		NULL	
ymidc	double	YES		NULL	
xmidc	double	YES		NULL	
groupNum	int(11)	YES		NULL	
seriesNum	int(11)	YES		NULL	
groupTime	double	YES		NULL	
ymidTime	double	YES		NULL	

estimated ones. The output of the models should provide an easy and most accurate means for estimating the deviation of the results.

3.4.3. Model Specification

Considering the data that has to be processed and the relevant details that can be extracted from the data, it is obvious that a set of specifications needs to be laid out before the actual data mining algorithms could be analyzed. The algorithms can be separated into three categories. One set of algorithms would prepare the data that would be processed by the data mining methodology. Another set of algorithms would describe the results. While the third category of algorithms can interpret the data, that the applications can use it meaningfully. In this thesis, the application is considered as a visual display of the trajectory. The algorithms presented here, process the data, try to understand the data, and manifest the hidden details of the subject's walk. These three activities have to be automated. Hence details have to be outlined for the preprocessing phase, the data mining algorithms and the post processing phase.

We consider the preprocessing phase, before the data mining algorithms process the data:

- Initially, we have to read the data. At the same time a database would be opened with a persistent connection. Data from the file handle of the read operation should be available as *stringtokens* to understand each word.

- Fixed definitions should be used to identify particular frame headers from the log file. As the frames are read from the log file, the input data stream has to be sampled fast enough to recognize the tokens that are needed, for example the "V" and the "S" frames.

- Each field description of the "V" frame has to be read. The data has to be stored at the database in the CARPET table. The data should be simultaneously used to rebuild the network topology and this must be available in a visual interface. The visual interface should provide the possibility to look into each node and display the spatial coordinates and the processor number that was give to a particular processor at the instance the subject walked on the Carpet.

- When reading the "S" frames, the activation and the deactivation of the sensors should be visually displayed. This data has to be stored in the BASE table. At this point, missing frames should be recognized visually. The algorithm should also facilitate data base extraction using standard SQL commands. A facility can be provided for user interaction. Here the user must be able to initiate the database access. A suggestion would be to use a "Sensor ON" button on the GUI which could start the whole process of displaying the sensor signals. Data retrieval should stop when the end of the file is encountered.

- After the sensor data is extracted from the log file, sensor data in the database should be cleaned and rearranged. The file handler can be deleted at this time. Each activation should be followed by the corresponding deactivation frame. On the other hand, a table should also maintain the original sequence of events. This data might be vital for further preprocessing. After the data is arranged, the time difference between two sensor activation frames should be calculated and the time difference between an activation and a deactivation frame should be calculated. These data should be filled in the database tables CLUSTERTEST and the SORTED tables.

- Missing data has to be treated in the least error prone manner.

When deactivation frames are missing, the time difference between the activation and the corresponding deactivation frame can be nullified. This is because, each foot stamp would trigger more than one sensor and all these sensors would generate a deactivation frame when the foot is removed. If one of the frames is missing, the rest of the frames, from this cluster describe the pause, the subject's foot made on the cluster. When an activation frame is missing, the frame either belonged to a previous cluster or to the one after it. If the time stamp of this missing frame is filled with that of the previous frame, then the missing frame would belong to the previous cluster. But if that is not true, the particular frame must be considered as a member of the next cluster, which may or may not be true. Hence there has to be a possibility to identify this frame as a missing active frame.

Now the data is available in the table and ready for the data mining algorithms to process them. The data mining algorithm should be able to:

- Read the sensor activations and calculate the clusters. This is the main problem which has to be dealt with extensively. The clustering algorithms should be relevant to the nature of the data that is available at the database. This is not a classification problem where all the instances have to be classified, rather the instances have be divided into natural groups or clusters based on their neighbor information. Hence the algorithm has to provide a means to calculate an estimate that can identify the cluster borders.

- The estimate has to fit every instance of sensor entities and in all use cases of entities. Which means, when the subject walks, the path might involve more curves or trail only a straight line. The method that does the estimate should be consistent in its result, irrespective of the variations in the subject's trajectory.

- After the estimates are determined and the clusters are identified, the clusters have to be verified for overlaps. It might be possible that two clusters could overlap due to a weak cluster estimate. These clusters have to be further splitted and simplified. Then the cluster centroids have to be calculated.

- The cluster centroids must be used to calculate the midpoints. The midpoints reflect the subject's physical location on the carpet. These midpoints lie on the trajectory the subject took while walking on the carpet.

The estimated midpoints are available in the database for further processing or can be used by an application. In this work the GUI was considered as the application. The trajectory has to be visually displayed on the network that lies under it.

- The calculated data has to be converted to the spatial coordinated that correspond to the physical location of the carpet.

- The trajectory should be the line joining the spatial coordinates and this line has to be visual on the display.

- The application should offer the possibility to evaluate the models. This means the trajectory must be comparable to a reference which describes the actual path the subject took during walking. The reference and the result of the post process must be directly comparable.

3.4.3.1. Miscellaneous Considerations

There are various other parameters that must be considered when estimating the trajectory. For example, the current size of the sensor used is $15\,\text{cm}^2$. At an average three sensors can cover an adult foot. Hence the boundary of the foot cannot be identified. This means, the estimated centroid rarely accords the real center of the person's foot. The boundaries can be granulated by reducing the size of the sensors.

Since the carpet is of a specified size, a person's walk is modulated by the boundary conditions of the carpet area. His or her walking behavior can be different in an area quite large in size [RS03]. The person might not slow down while encountering the edge of the carpet although the room is larger. The pace of the walk is not altered during curves that exist at certain intervals, whereas in areas with short curves, the walking pace is modulated by the limited area. This modulation cannot be summed up as a vector of numerals as we have one more important parameter that does consistently affect the walk of a person.

The mood of a person briefly affects the number of steps a person takes and the speed of walking [New06]. The walking speed of a relaxed office guy is considerably different than of a timely pizza delivery boy.

Considering all the above facts, the input data that is fed to the model is taken as in an ideal condition whereas the walking pattern of the person on the carpet is considered as nonlinear.

3.5. State-of-the-Art for Pervasively Tracking Human Motion using Smart Floors

Video based gait identification methods can be considered as a natural way to model human walking. Most of the existing algorithms for motion estimation are confined to video based data sources. In this approach each frame in the video sequence is treated separately through the algorithm. Changes from each frame are processed and treated using a fitting function, implicitly or explicitly. Various features from the images are extracted to calculate the posture, the limb positions of the subject, the frequency, and phase presentation of walking [LB96]. Eigenspace transform with canonical space transformations and continuous HMM have been applied for modeling mode dynamic properties of moving objects [HHN98, KRCK02].

An interesting work worth mentioning at this point is the real-time human tracking using N-ocular stereo [SIT00]. In this system four omni directional vision sensors (ODVS) were placed 1 meter above an average person's height. A standard image capture card (Matrox Meator, 640 by 480 pixels) on a personal computer was used to capture the image of the room. The ODVS camera offers a wide screen and observation errors of azimuth angles should be considered when measuring persons locations. Since the human body deforms every moment and is widely projected on the ODVSs, the subject is represented with a circle of a constant radius, and the location of a person is represented as the center of the circle.

The system has a better target recognition with a maximum error of 0.17 m. It was able to completely track a single person. However, the system suffers from false identifications. Binocular stereo using ODVSs has a low-precision problem with respect to targets locating along the baseline of the sensors. Secondly, the result of background subtraction becomes noisy if the color of subjects clothes is similar to that of the background. General noise filtering techniques such as the Kalman filter may not be able to successfully eliminate the background noise, since the noise is different from white noise. A more effective method would be to add sensors additionally. But this again increases computing complexity and cost.

Although vision based systems provide reliable methods, which can capture the dynamic properties of human walking, they suffer from the differences in light conditions and the background movements. Obviously a vision based system is not always hidden. This evokes a feeling that the subject is being monitored. A floor based motion estimation system accords the requirement of a transparent computing in the ubiquitous world.

In Active Floor [OA00] and Smart Floor [MAS97], footstep identification was accomplished by utilizing small force plates. A floor tile was fitted with four force sensors in the corners to realize a tile in the Active Floor. The analog sensors were then assisted by a 4 channel analog to digital converter to recover the signals on the computer. Nearest-neighbor classifier and HMM methods were used in the research to identify foot steps. Besides identification, force plates have been used to detect and classify simple activities of human movement, such as crouches and jumps as well as standing and sitting [SSvL+02].

Although the system can be argued as an advanced system, it lacks a commercial implementation and the cost is rather high. Each tile needs a 4 channel analog to digital converter and a personal computer to extract the features. This system cannot be foreseen as a low cost embedded solution on one tile. Most of the other systems [SPR03b, RPM95] do fall in the same category of high infrastructure installations.

Active Badge [WHFG92], Active Bat [HHS+99, WJH97], and Cricket [PCB00] location systems require the installation of infrared/ultrasonic transmitters/receivers at fixed locations in the environment. These systems only locate, but cannot track the subject. In order to attain high location accuracy and good coverage the system infrastructure requires a large number of transmitters and receivers to be installed in the deployed environments. Audio Location system [SD05] requires the subject to finger click, that the subject could be located, which cannot be expected in reality.

Target classification and trajectory tracking are intensively explored in wireless sensor networks [LWHS02, BGF02]. Most of these experiments are focused on detecting Combat vehicles; however, Geta Sandals [KOC05] and Human tracker [TMI04] are focused on low infrastructure, easy to implement footstep tracking systems. Geta Sandals use Radio Frequency (RF) based location identification techniques with an accuracy of 158 cm. However the system needs additional installation of RFID tags on the floor in order to avoid the incremental displacement error that is caused by the accelerometer installed in the sandals. However this research is focused on tracking a subject's walk and is manufacturable compared to the systems mentions in the above discussion.

The Human tracker [TMI04] is quite similar to the EMFi floor but the methods used for trajectory tracking in this experiment have better performance than the Geta sandals. The Human tracker has an error deviation of 58 cm on similar straight line walking experiments. To remove lossy signals the sensor data was processed by Markov Chain Monte Carlo method which implements two estimation methods. A linear Gaussian model that requires less computation is used to estimate the trajectory of the subject. The error estimate could be improved by a non linear bipedal model. In the Gaussian model, the history of past six footsteps was taken and these coordinates along with their velocities were considered as a block. A normal distribution of the blocks was computed to estimate a trajectory. In the bipedal method a truth table was used with different conditions. In a straight line experiment the system can track people with a mean error of 20 cm.

The systems discussed so far are confined to laboratory experiments and cannot be commercially implemented. The cost of the systems run to hundreds of dollars. The Geta sandals can be considered as the one which can be brought to the retailer shelf. On the other hand, this system is error prone. However, a low cost human motion tracking system with commercial manufacturability, that is easily deployable is still not available. Using the Smart Carpet as the base, this thesis focuses on tracking human motion. The main contribution of this thesis are the conceptual methods towards attaining this goal. The data from the Smart Carpet is analyzed and refined to a useful form which is extensively analyzed by the methods that are described in this chapter. This chapter proposes three methods that describe the data and analyze the hidden patterns in it. In the first method the classification based on mean, time differences between the arrival of activated sensors are treated using a mean based model. The same data is then experimented using a sophisticated mathematical model in the second method. In the third approach, the data is treated logically using a truth table. In all the methods, the criteria for clustering is treated extensively.

3.6. Model 1 - Classification based on Mean

The data stored in the database has to be analyzed for patterns that match the subject's walk. The task of the data mining model is to identify these patterns. The identification process must be fully automated and the results of the identification should be consistent for all use cases of data available in the database.

3.6.1. Model 1 - Rationale

To achieve this objective an algorithm that can briefly describe the underlying data in the database must be constructed. The method "Classification based on Mean" is inspired from the early works done in Active Floor [OA00]. Here, the researches tried to consider the complete GRF profile and took an average on the amplitude between a limited time frame. This method did yield considerable results, although it was further improved by the Smart Floor [MAS97] where parameters were considered from the same GRF profile.

Although the methods used in Active Floor and the Smart Floor cannot be directly compared to the methods used here in the first instance, the differences are worth to be explained. In this approach we consider the entire walk instead of a single footstep. Considering only the activated footsteps, parameters are extracted from the spatial coordinates and the speed of walking. Whereas, in Active Floor only the amplitude of the GRF profile is considered and in the Smart Floor, this profile is sliced or sampled and the sampled amplitudes are considered as parameters.

The advantage of this method is that the estimated parameters reflect the characteristic of the subject's entire walk and not just a single footstep which is not the case in the other methods. Another consideration for this approach is the computing time of the algorithm, which is rather short compared to that of complex signal sampling and processing work done in the Smart Floor.

3.6.2. Description

Looking into the database at the SORTED table which contains the sorted sensor information (Tab. 3.5), a resemblance among sensors can be seen in the *tDiff* field. This field (records 0,2,4,6,8 and 10) shows the time difference between two successive sensor activations and the time difference (records 1,3,5,7 and 9) between the active frame and its corresponding deactivation frame.

Table 3.5.: Snapshot from SORTED table in the database

id	procNum	tFrame	tStamp	tDiff
0	20	S00201	1,10743912531E12	0,0E0
1	20	S00200	1,107439126968E12	1656,0E0
2	30	S00301	1,107439125312E12	78,0E0
3	30	S00300	1,107439126828E12	1516,0E0
4	19	S00191	1,10743912539E12	5641,0E0
5	19	S00190	1,107439126812E12	1422,0E0
6	60	S00601	1,107439131031E12	515,0E0
7	60	S00600	1,107439131984E12	953,0E0
8	39	S00391	1,107439131546E12	219,0E0
9	39	S00390	1,107439132484E12	938,0E0
10	28	S00281	1,107439131765E12	406,0E0

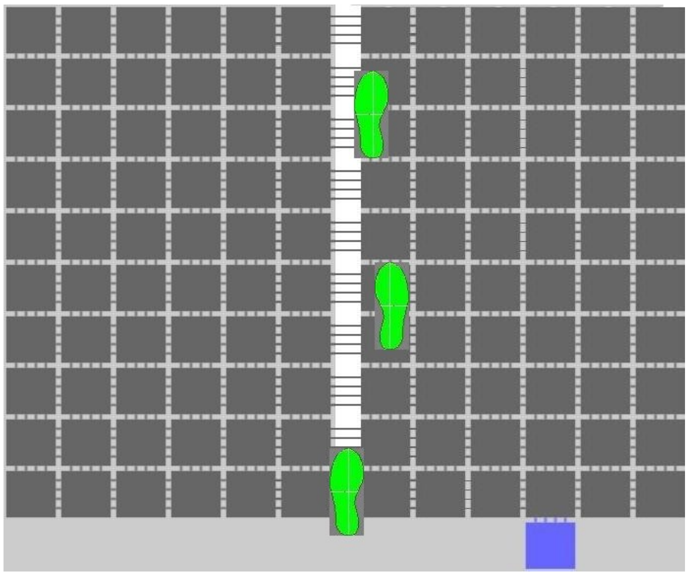


Figure 3.6.: Impression of the subject's walk on the Smart Carpet

The activated sensors time stamps and the time differences are of importance. Comparing the records of $0, 2$ and 4 the *tdiff* field has a value where the 4th record is extremely higher than the other. Comparing the records 6, 8 and 10 it can be observed that 6 is greater then 8 and 10. But looking at the time stamp field *tstamp*, it can be observed that records 8 and 10 lie very close in the time instance.

While this log file was recorded, the carpet was monitored for activated sensors. In reality it was seen that when the subject kept the first foot on the carpet, three sensors (20, 30 and 19) got activated. In the next step, it was observed only sensor 60 got activated, while consecutive steps activated sensors 39 and 28 (fig 3.6 on the previous page). This does correspond to the patterns that are visible in the database table. Now the real problem lies in how to automate the grouping of the records based on their resemblance.

To understand the data, its relationship with its neighbors, and to group the most relevant group members, a straight walk of the subject was considered. Manual clustering was observed. On the first instance the recorded data and the observed data were compared. Hence as the subject walked, a camera was observing each footstep and the time instance of the steps. The image was taken and each foot was plotted on the exact location on the Smart Carpet. This image can be used to compare the approximate location of the subject's feet on the carpet and the exactness of the clustering algorithm. The subject had placed 3 footsteps and the walk is shown in figure 3.6 on the preceding page. As the subject walked on the Smart Carpet, the log file was recorded. After the log file was processed and sorted, the SORTED table had the contents shown in table 3.6.

Table 3.6.: Contents of SORTED table for the straight walk

id	procNum	tFrame	tStamp	tDiff
1	55	S00551	1116506697224	47
2	55	S00550	1116506697818	594
3	56	S00561	1116506697271	594
4	56	S00560	1116506698036	765
5	59	S00591	1116506697865	0
6	59	S00590	1116506698036	71

The subject kept the first foot on a non sensoric area. Since there was no sensor at this location, the log file does not have this data. The second and the third foot landed on two sensors. If all the sensors were triggered, there must have been 4 sensor activations and corresponding deactivations. This implies, 8 records in the database. From the log file, it was seen that instead of 4 sensors, only three had been activated. Considering the available data it is seen that two of the subject's footsteps were captured by the Smart Carpet in this walk. Comparing the contents of the SORTED table and the location of the subject's feet, it is seen that sensors 55 and 56 got triggered in the first instance and sensor 59 got triggered in the subject's final step. The time difference between the activation of sensor 56 and sensor 55 was 47 milliseconds while the time difference between sensor 59 and 56 was 594 milliseconds. Hence it is clear that sensors 55 and 56 belong to a cluster while the second cluster does have only sensor 59 in its member list.

3.6.3. Algorithm Description

To automate this clustering process, the average of the time differences for all activated sensors can be taken. The estimate can be checked against the time difference for each activated sensor. Hence the model has to loop the contents of the SORTED table, select only the activated time differences and find an average. In the second loop, the estimated time difference has to be checked against the activation time difference. If the time difference of the activated sensor is large, then the sensor belongs to a new cluster. If it is less, then the sensor is the member of a current cluster. Thus in the above example, the average of the three activated sensors is 213.667 milliseconds. This clearly splits the first two sensors 55 and 56 from the sensor 59. Hence the time difference can be used as a parameter to cluster the instance. This can be give as the condition

$$T_e = \frac{\sum T_{d1} + T_{d2} + \ldots T_{dn}}{n} \tag{3.6}$$

On the other hand sensors 55 and 56 have their spatial coordinates located at 497, 503 and 497, 504 for x and y respectively. Meanwhile the spatial coordinates of 59 are 497, 507. It is also important to consider this parameter in automating the clustering. The radius of a foot print could cover a range between 12 and 25 cm for adults. This could cover an average of 4 sensors at one instance. Hence a cluster can have a maximum of 4 members at any instance. This parameter R can be considered to determine cluster members with the following condition

$$R = [X_n - X_{n+/-1}] + [Y_n - Y_{n+/-1}] \tag{3.7}$$

$$R \leq 2 \tag{3.8}$$

After these parameters are estimated, each record can be validated for condition in eq. (3.9). If both the conditions are satisfied it can be assumed that the entity is a cluster member. If any of the condition, does not satisfy, it can be concluded that the entity belongs to a new cluster.

$$C_m = T_{diff} \leq T_e, R(true) \tag{3.9}$$

From equation 3.9, we can determine if the activated sensor entity in the SORTED table belongs to a cluster or forms a new cluster. The cluster centroids can be calculated by averaging the members of the cluster. This method was chosen, not only because it was quite fast to implement such an approach and the computing is quite easy, but this approach described the underlying data in a efficient way. Similar methods were also done in the Active Floor to identify footsteps [OA00]. But in this approach, the conditions are implemented using estimation parameters which consider the complete walk of the subject.

The conditions are implemented and shown in algorithm 1. In the first instance, all sensor details are sorted and arranged in their order of instance. This is done by a sort function. Then the time difference estimate is determined considering all instances of the activated sensors of the particular walk. Next the estimated time difference is validated for each instance of the sensor activations. If the conditions do not hold it is most probable that the entity belongs to a new cluster. On the other hand if the entity adheres to the condition (eqn. 3.9), the entity belongs to the current cluster.

After the clusters are identified, the centroids of the clusters are calculated. The cluster centroids would locate the approximate center of the subjects foot. Upon location of the subjects foot, the time difference between two centroids would describe the speed the subject took between two steps.

Algorithm 1 Algorithm implementation for Classification using Mean

for all activated sensor[i], $i < n$ **do**
  sort (processor num, time stamp)
  sum_{td} = sum of sensor s_n(timestamp) - s_{n-1}(timestamp)
  $est_{td} = sum_{td}/n$
end for
for all $sensor_n$ **do**
  if (($s_{n-1}(x)$ - $s_n(x) \leq 2$) and ($s_{n-1}(y)$ - $s_n(y) \leq 2$)) **then**
    cluster member
  else
    new cluster
    $N \Leftarrow N + 1$
  end if
end for
for all $sensor_n$ **do**
  if $N == tN$ **then**
    calculate centroid $(x, y, time)$
  end if
end for
for all centroid **do**
  if $N \neq tN$ **then**
    calculate midpoint (centroid)
  end if
end for
for all midpoint **do**
  plot trajectory
end for

This can be used to calculate the cadence of the walk. On the other hand, the spatial difference between two cluster centroids would give the stride length. The centroids are stored in the GROUPMIDS table in the database. This data is then processed again. The midpoints of two centroids are calculated to determine the location of a person on a thin trajectory. The center of a subject is assumed to be an arbitrary point between the feet. Calculating the midpoint of two centroids would locate the center of the person. The data is then presented to the application layer where the trajectory is then drawn using the midpoint coordinates.

3.7. Model 2 - Classification based on Maximum Likelihood Estimation

Using the Model 1 classification method, it was possible to construct the trajectory of the person's walk on the Smart carpet. The experiments were made where the subject was walking on a straight line. Each walk was specified to a particular path that reflected a pattern. The straight walk was defined as pattern I. More details of patterns are given in appendix A. The subject was walking on more complex patterns which involved short curves and turns which were classified under different patterns. All subjects were requested to walk the same set of patterns.

3.7.1. Model 2 - Rationale

As the subject made curves within the carpet edges, the speed was less during the turning compares to the speed taken during a straight trajectory. As a result the speed of the subject was changing during the curves compared to the straight line walks. This was observed for all subjects when they walked a particular set of patterns (Fig. 3.7) .

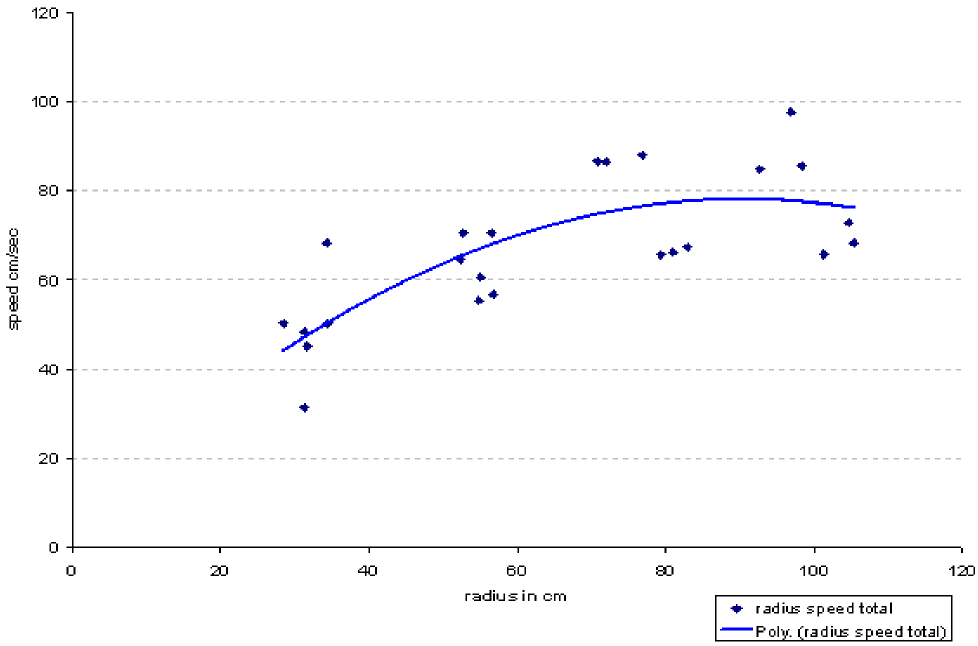


Figure 3.7.: Walking speed in cm vs radius in cm

For example in pattern S where the subject has to make the trajectory like an S on the carpet, one has to curve twice and walk straight in between. During the curves, the subject was slow compared to the speed during the straight line. This was observed for different age groups and the results are plotted in figure 3.7 on the preceding page.

For a dataset of a subject who walked pattern S, model 1 derived an estimated time difference as 289,035 ms for the data given in table 3.7. Based on the cluster criteria for the model 1, the classified clusters are given in field *groupNum*. It can be clearly seen for subject A that the entities 5 to 13 fell in a single cluster. In the second column for subject B, at events 27, 29, and 31, the clusters were merged together although in reality they were two footprints. The spatial radius was also within the conditions. As the subject walks slower during the curves, the estimated time difference was a large value, many non-members were merged to a cluster. Hence the method of estimating the time difference has to be much more refined to accommodate the dynamic behavior of the subject's speed. The parameter estimation method for determining the time difference can be more precisely realized.

Table 3.7.: Model 1 results for pattern S of two subjects

Subject A				Subject B			
id	procNum	diffTime	groupNum	id	procNum	diffTime	groupNum
1	30	750	1	1	90	453	1
3	42	640	2	3	99	250	2
5	68	47	3	5	97	406	2
7	69	297	3	7	86	63	3
9	79	203	3	9	85	156	3
11	89	47	3	11	84	687	3
13	98	453	3	13	62	625	4
15	86	453	4	15	55	63	5
17	65	203	5	17	48	31	5
19	56	469	5	19	49	781	5
21	39	406	6	21	51	594	6
23	13	188	7	23	51	188	7
25	11	47	7	25	31	1156	7
27	14	578	7	27	27	312	8
29	4	594	8	29	21	235	8
31	43	46	9	31	19	547	8
33	46	641	9	33	11	781	9
35	73	63	10	35	2	31	10
37	83	0	10	37	3	0	10
Est. Time diff.		322.3 ms		Est. Time diff.		387.3 ms	

3.7.1.1. Maximum-Likelihood Estimation

After the model is specified with its parameters and the data being analyzed, one can evaluate the quality of fit between the model and the data. The previous model described the data efficiently on certain use cases only. To improve the quality of fit, the estimated parameter is tuned to fit more precisely. But a fixed parameter cannot be defined as a perfect fit for all the use cases of subjects walk. Hence for each pattern of the subject walk a precise parameter has to be estimated that can describe the model to that particular set of data. There are two common methods of parameter estimation. The Least-Squares estimation (LSE) and the Maximum-Likelihood estimation (MLE).

LSE requires no or minimal distributional assumptions. Summarized observed data can be well described using LSE, but it has no basis for testing hypotheses or constructing confidence intervals. MLE [Fis20] is a standard approach to parameter estimation and inference in statistics. It has many optimal properties in estimation:

- sufficiency (complete information about the parameter of interest contained in its MLE estimator)

- consistency (true parameter value that generated the data recovered asymptotically, i.e. for data of sufficiently large samples)

- efficiency (lowest-possible variance of parameter estimates achieved asymptotically)

- and parameterization invariance (same MLE solution obtained independent of the parametrization used).

In contrast, no such things can be said about LSE. As such, most statisticians would not view LSE as a general method for parameter estimation, but rather as an approach that is primarily used with linear regression models. Further, many of the inference methods in statistics are being developed based on MLE. For example, MLE is a prerequisite for the chi-square test, the G-square test, Bayesian methods, inference with missing data and modeling of random effects.

3.7.1.2. Likelihood Function

Considering the time difference of all the sensors of a particular subject, one foot would have a data set where $t_d = (t_{d1}, \ldots, t_{dn})$ is a random sample from a collection of feet or a walk which can be said in statistical terms as a population. The goal of data analysis is to identify the population that is most likely to have generated the sample. In statistics, each population is identified by a corresponding probability distribution. Associated with each probability distribution is a unique value of the model parameter. As the parameter changes in value, different probability distributions can be generated. Formally, a model is defined as the family of probability distributions indexed by the model parameters.

Let $f(\frac{t_d}{w})$ denote the probability density function (PDF) that specifies the probability of observing data vector t_d given the parameter w. Given a set of parameter values, the corresponding PDF will show that some data are more probable than other data. In reality, however, we have already observed the data. Accordingly, we are faced with an inverse problem: Given the observed data and a model of interest, find the one PDF, among all the probability densities that the model prescribes, that is most likely to have produced the data. To solve this inverse problem, we define a likelihood function by reversing the roles of the data vector t_d and the parameter vector w in $f(\frac{t_d}{w})$, the likelihood function can be given as equation 3.10

$$L(\frac{w}{t_d}) = f(\frac{t_d}{w}) \tag{3.10}$$

The important difference between the likelihood function and the PDF is the fact that the PDF is a function of the data, given a particular set of parameter values defined on the data scale. On the other hand, the likelihood function is a function of the parameter given a particular set of observed data defined on the parameter scale. The principle of MLE, originally developed by R.A. Fisher in the 1920s [Fis20], states that the desired probability distribution is the one that makes the observed data "most likely", which means that one must seek the value of the parameter vector that maximizes the likelihood function $L(\frac{w}{t_d})$: The resulting parameter vector, which is sought by searching the multi-dimensional parameter space, is called the ML estimate, and is denoted by $w_{MLE} = (w_{1_{MLE}}, \ldots, w_{n_{MLE}})$. According to the MLE principle, this is the population that is most likely to have generated the observed data: To summarize, maximum likelihood estimation is a method to seek the probability distribution that makes the observed data most likely. The values of these parameters that maximize the sample likelihood are known as the Maximum-Likelihood Estimator.

Although Bayesian parameter estimation can be used to identify estimation for the time difference, the latter views the parameters as random variables having some known distribution [DHS00]. But in the MLE approach the parameters are fixed and they are unknown. Since we have only one fixed parameter (time difference) and a varying estimate (the spatial coordinates), the time difference alone can be identified using the MLE and the spatial coordinates can be estimated subsequently during the criterion check. The data that is used to estimate the time difference has two properties. Their sources are consistent. This is due to the fact that a subject moves over a period of time from place A to place B. During this motion a set of sensors get activated and deactivated sequentially. Secondly, they are affected by the delay in the network as data tries to reach a central processing unit, passing over an unknown random time delay. They can be considered as noise. A Kalman filter could be used to eliminate this noise. But since the noise is random and it is of a very negligible value of 2.4 milliseconds per hop compared to 750 milliseconds for an average walking speed of an adult, this noise can be eliminated [TMI04].

The data further represents two distinctive features. One is the time between two foot steps which can be said as the Euclidean distance between two clusters. The other is the inter cluster distance which is the time difference between groups of atleast one sensor that represent a single foot step. The speed of the subject's walk differs randomly. Hence the data also has small values which represent inter cluster distances for members which belong to a cluster and large values that represent inter cluster distance for two clusters. A generic likelihood equation would suffice to derive an estimate that would separate these two distances. Since the time difference is a quantity from the speed of the subject, a harmonic mean has to be considered while calculating the mean for the variance function in the maximum likelihood estimate [Cho70]. A set of data samples can be taken to validate the likelihood function. The same set of data could be given as an input to standard MLE software and the estimates can be compared to see if the equations do fit. $mleV2.1$ is a free Maximum likelihood estimate software for building MLE models. It is implemented with a generic model for calculating estimates. The generic model is made to fit the input data based on statistical estimated parameters [Hol03]. This software is used for testing the likelihood equation used in this thesis. The results are compared to see the accuracy of the estimate.

In the first step to parameter estimation using MLE, the harmonic mean is calculated by equation (3.11)

$$\mu = \frac{n}{\sum_{i=0}^{n}(\frac{1}{t_{di}})} \tag{3.11}$$

its variance is given by (3.12)

$$\sigma^2 = \sum_{i=1}^{n} \frac{(t_i - \mu)^2}{n-1} \tag{3.12}$$

the Likelihood estimate of the time difference can be calculated by (3.13)

$$o(\mu, \sigma^2) = \frac{n}{2}\ln(2\pi) - n\ln(\sigma^2) - \sum_{i=1}^{n} \frac{(t_{di} - \mu)^2}{2\sigma^2} \tag{3.13}$$

For computational convenience, the MLE estimate is obtained by maximizing the log-likelihood function, ln $(L(\frac{w}{t_d}))$. This is because the two functions, ln $L(\frac{w}{t_d})$ and $L(\frac{w}{t_d})$ are monotonically related to each other so the same MLE estimate is obtained by maximizing either one. Assuming that the log-likelihood function ln $(L(\frac{w}{t_d}))$; is differentiable, if w_{MLE} exists, it must satisfy the following partial differential equation known as the likelihood equation:

$$\frac{\partial \ln(L(\frac{w}{t_d}))}{\partial w_i} = 0 \tag{3.14}$$

3.7.2. Algorithm Description

In the first instance, the equations are tested for their correctness by a standard mle software. The likelihood estimation was calculated by iterating the likelihood equation 3.13 for time difference taken for each sensor entity in a subject's walk. This equation was implemented in a separate Java class file. The result of this class file was validated against the estimates determined by mle. A set of walks with various stomp numbers were taken. The stomp numbers can be related to the number of samples iterated in the likelihood equation.

Table 3.8.: Comparison between automated and derived *mle* results

Samples (N)	Automated	Derived	Difference
18	128.4682	129.0703133	0.602113316
17	116.8582	117.1635553	0.305355274
12	78.35338	78.01594305	-0.337436945
28	199.6215	199.635659	0.014159012
21	143.4148	143.5573307	0.142530696
19	131.2204	131.7338661	0.513466084
18	123.4628	123.4924133	0.029613261
19	133.228	134.063405	0.835405012
13	89.54553	89.24861289	-0.296917106
3	21.04786	21.0140921	-0.033767903
Mean difference			0.177452070 ms

The results (tab. 3.8) show a very minor deviation of 0.177 ms from the mle software and estimate calculated by the likelihood equation in the Java class file. The table further shows that for large samples, the deviation is highly negligible. This accords the basic nature of the maximum likelihood estimation. In the implementation, the complete dataset is first passed thru eqn.3.13, until the MLE is estimated for this particular sequence of data. Then the estimate is used as a criteria to group the clusters.

Each model is implemented in a separate Java file with interfaces. The core algorithms that process the database and handle the log file call the data mining models using the interface classes. This provides a complete abstract layer for the implementation and testing of the data mining models. The data mining models do not call any abstract methods of the core modules. Rather only references for the database results are returned in the handle. The modules mentioned here are the Java classes which are organized in packages that contribute to different functions of handling data from the log file and rendering them to the Smart Carpet Monitoring software. This software is internally constructed in packages which describe the ADNOS algorithm for monitoring the Smart Carpet. Further models can be added in this way using abstract classes and the thus a modularized coding is implemented. Using this current model the estimates for subject A is 134 ms while for subject B is 137 ms. Now the SORTED table looks like this:

It is observed in the database that entities 5 to 13 for subject A are grouped in different clusters. In the second column for subject B, at events 27, 29 and 31, the clusters were further split to reflect the real data. The parameter estimation method using MLE has considered every event in the whole walk and the estimation has been derived based on the all events of the subject's walk.

Table 3.9.: Model 2 results for pattern S on two subjects

Subject A				Subject B			
id	procNum	diffTime	groupNum	id	procNum	diffTime	groupNum
1	30	750	1	1	90	453	1
3	42	640	2	3	99	250	2
5	68	47	3	5	97	406	3
7	69	297	3	7	86	63	4
9	79	203	4	9	85	156	4
11	89	47	5	11	84	687	4
13	98	453	5	13	62	625	5
15	86	453	6	15	55	63	6
17	65	203	7	17	48	31	6
19	56	469	8	19	49	781	6
21	39	406	9	21	51	594	7
23	13	188	10	23	51	188	8
25	11	47	11	25	31	1156	9
27	14	578	11	27	27	312	10
29	4	594	12	29	21	235	11
31	43	46	13	31	19	547	12
33	46	641	13	33	11	781	13
35	73	63	14	35	2	31	14
37	83	0	14	37	3	0	14
Est. Time diff.		134 ms		Est. Time diff.		137 ms	

3.8. Model 3 - Classification based on Rank Regression

When the subject was walking straight, the whole walk had a linear speed and did not vary much, within the walk. As the subject tried to make turns while walking, the speed decreased at the curves. It was observed that at certain foot transitions, the subject kept the foot on the floor during the curves a bit longer than the time when the transitions were during a straight walk. This resulted in quick time transitions between the walk while he was on the straight path and a very slow time transition during the curves. Model 1 estimated this transition as a parameter to group the clusters for each footstep. In that process, the method used calculated an average considering all events of the time difference in one instance. This estimate was not a problem when the subject was walking straight, but during curves, the estimate was quite large that it swallowed a few clusters where the subject had walked straight. This resulted in a single footstep (events 5 till 13 in tab. 3.10 on the next page)). To solve this problem, the second method used the MLE estimate on the time difference to refine the clustering.

3.8.1. Model 3 - Rationale

The MLE considered every event and determined an estimate that would most likely suit all the foot transitions. The estimate was nearly half the value of the estimate determined by model 1 (see $timediff$. field in tab. 3.10). In the subject's walk that involves curves, the subject has short time differences while walking straight and large time differences between the feet during the curves. But the ML estimate was so fine grained that some events are splitted within the clusters. Comparing columns M1 and M2 under $groupNum$ in table 3.10, for events between 13 to 23, ML estimate split the cluster members, where the subject walked slowly, into individual clusters.

Table 3.10.: Cluster results for subject A on pattern S using model 1 and model 2

id	procNum	diffTime	groupNum	
			M1	M2
1	30	750	1	1
3	42	640	2	2
5	68	47	3	3
7	69	297	3	3
9	79	203	3	4
11	89	47	3	5
13	98	453	3	5
15	86	453	4	6
17	65	203	5	7
19	56	469	5	8
21	39	406	6	9
23	13	188	7	10
25	11	47	7	11
27	14	578	7	11
29	4	594	8	12
31	43	46	9	13
33	46	641	9	13
35	73	63	10	14
average diff time			322 ms	134 ms

Using MLE, the estimated parameter is more biased towards the areas where the subject had walked faster. In model 1, the estimate was just the opposite. Hence it is quite difficult to rate all time stamps of a walk at the same instance. A more adaptive method needs to be found which can identify clusters with a varying estimate. A fixed estimated parameter cannot improve the clustering problem that reflects the dynamics of a persons walk [Gup06].

3.8.2. Bradley-Terry Model

In the previous attempts, the models were estimating a fixed parameter by considering the entire walk at one instance. The result was an estimate which was compared with the time difference of each event to see if the event belonged to the current cluster or belongs to a new cluster. Carefully observing, the decision to be taken on an event is, if the entity belongs to the current cluster or not. This decision is based on the estimated parameter against the inter cluster distance. The same decision can be made, if we consider two or three entities and identify which has the largest inter cluster distance. In this approach we do not depend on an estimated parameter, rather we consider the dynamics of the subject's walk and decide on each entity.

Consider i and j as two entities, the possibility that i is greater than j can be given as,

$$\pi_{ij} = \frac{exp(td_i)}{exp(td_i) + exp(td_j)} \tag{3.15}$$

$$\pi_{ij} = \frac{exp(td_i - td_j)}{1 + exp(td_i - td_j)} \tag{3.16}$$

Therefore, the Likelihood function of a paired comparison model is,

$$L(td_1, td_2 \dots td_n) = \prod_{(i-j)\epsilon A} \pi_{ij} \tag{3.17}$$

where A is the sample collection of all entities in the walk and td is a positive values parameter which can be said as the ability of i or j. In this model it can be referred to as the time difference. For comparison, if td_i is a member or non member to td_j, let the vector d_i be such that, $d_i = (d_1, d_2 \dots d_n)$

$$d_{ik} = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.18}$$

The likelihood for the Barry-Terry model is identical to a binary logistic model with d_i as covariates, no intercept, and a constant response, although the time differences vary randomly. It is more easy to rank the ability of an entity that possesses a large inter cluster distance compared to that of its counterpart. At the first instance, the work of the ranker can be evaluated for all entities in a walk. In this approach, it must be remembered that the result is not a single entity rather, it is a classification of entities into different groups based on their sequence and neighbors.

The radius of a foot can occupy more than two sensors. Hence the entity td_i can belong either to td_{i+1} or to td_{i-1}. Therefore, to decide if an entity belongs to a previous cluster or forms a new cluster requires three entities to be compared for the inter cluster distance at the same instant. A better way would be to find the maximum and the minimum value of three entities, defining ranks for each, take the middle value and continue to compare and rank with the consecutive non-ranked entities. The comparison vector d_i can be given as,

$$d_{ik} = \begin{cases} 1 & \text{if } k > i-1, i+1, \\ 0 & \text{if } k < i-1, i+1, \\ 2 & \text{otherwise.} \end{cases} \tag{3.19}$$

Table 3.11.: Truth table for ranking entities

$i-1$	$i+1$	Rank	Comments
1	1	0	member
1	0	0	member
0	1	0	member
0	0	1	new cluster
2	0	X	condition 1
0	2	X	condition 1
2	1	X	condition 1
1	2	X	condition 1
2	2	X	condition 1

A rank of 1 defines the entity as a limit or rather the entity has the greatest inter cluster distance. The consecutively following entities would then belong to a new cluster. The rank 0 defines an entity as a member of the current cluster. A rank of 2 defines the entity to be re-evaluated. This entity can be a part of the current or the next cluster. In the second iteration, the ranks can be passed through a truth table (tab. 3.11) based on the nature of the data registered in the log file. In this iteration, the entities that have a rank of 2 can be re-ranked as either a 0 or a 1. For condition 1, the greatest time difference from the three consecutive entities is ranked 1 and if there are clusters with more than 4 entities, the cluster is split by choosing the entity with a greatest time difference.

3.8.3. Algorithm Description

This method of classification requires more iterations compared to the previous models. The outline of the algorithm is presented in listing 2 on page 67. The parameters from the log file are extracted previously and stored in the database. The table is sorted based on the sequence of events. The result of an SQL query on this sorted table is then passed to this algorithm. The algorithm attempts to classify cluster limits by ranking three entities at each instance. The ranking is done on the time difference parameter. Hence the parameter is not a fixed estimate as compared to the previous models, rather it is a floating estimate based on a comparison of events at past and future values. The results of the ranker are binary values for each entity which describe if that entity has the largest inter cluster values or not, typically a 1 or a 0 as its rank. All entities which have a rank of 1 and the consecutive entities which its rank as 0 are decided as a cluster. If the iteration meets a 1, then the following entities would be considered as a new cluster.

Table 3.12.: Clusters results for subject A on pattern S using model 1 and model 2

id	procNum	diffTime	groupNum		
			M1	M2	M3
1	30	750	1	1	1
3	42	640	2	2	2
5	68	47	3	3	2
7	69	297	3	3	2
9	79	203	3	4	3
11	89	47	3	5	3
13	98	453	3	5	3
15	86	453	4	6	4
17	65	203	5	7	4
19	56	469	5	8	4
21	39	406	6	9	5
23	13	188	7	10	5
25	11	47	7	11	5
27	14	578	7	11	6
29	4	594	8	12	6
31	43	46	9	13	7
33	46	641	9	13	7
35	73	63	10	14	8
Time diff			322 ms	134 ms	N.A.

Algorithm 2 Algorithm implementation for Classification using Rank Regression

```
for all activated sensor[i], i < n do
  sort (processor num, time stamp)
  SQL query to check cluster criteria.
  Check first part of criteria based on eqn.3.4.
end for
for all sensor_n do
  if cluster members greater than 4 then
    do Rank Regression(Result set)
  end if
end for
Rank Regression
1st Iteration
for all forallentities do
  select first three entitites
  find greatest and the least of the three
  rank 1 and 0 respectively, rank 2 for the middle
  consider current 2 and consecutive non-ranked entities
  ...continue...
end for
2nd Iteration
for all forallentitites do
  use truth table. 3.11 on page 65
  if cond.1 then
    repeat 1st iteration, rank rest as 0.
  end if
end for
3rd Iteration
for all for all entities do
  check for 1s.
  if rank = 1 then
    cluster number ++
  end if
  if clustermembers ⪰ 4 then
    check largest tdiff
    split cluster
  end if
end for
```

The results of model 3 can be compared to that of the other previous models in the same data from subject A on pattern S in table 3.12 on the previous page. It could be observed that, events 15 to 23 had been evenly splitted.

3.9. Summary

A model as an algorithm is actually the compressed representation of the input data in the data mining world. This model contains the essential knowledge extracted from the data and it is described as an algorithm [MR05]. Before defining the model, the data pattern is briefly explored. In this process of understanding the data, it was observed that the footprints on a walk are represented as clusters in the sensor data. This leads to the problem of identifying these clusters. In identifying the clusters, a suitable method has to be found to group the clusters and it is treated not as a classification problem. Based on this, the specifications of a model are detailed. While implementing, the models would differ by the clustering methods employed.

State-of-the-art approaches use traditional clustering algorithms like K-means and Gaussian mixtures to find a best model for the available data. Considering a trade off between the two models, a new model is specified here to identify the trajectory of a subject's walk. The model employs an estimated parameter and a moving neighborhood criterion for clustering. The model is improved by implementing a likelihood function for parameter estimation. In the first model, the estimated parameter was large and could not identify certain cluster members when the subject was walking on curves. In the second model, the estimated parameter was too fine grained, that it split the clusters. Considering a more rational approach of rank regression, the problem was solved. This model is supported by a truth table which is extremely biased on the nature of the data available from the log files.

The next task is to check the consistency of the models and identify the best model. The following chapter proposes possible methods to generate an accurate reference to evaluate the models. From the available methods, a reasonable approach can be used to validate the models.

4. Concepts of Model Validation

To trace the trajectory of a subject's walk, sensor data from the Smart Carpet is analyzed and fitted in a model that describes the nature of the data. To make a meaningful use of the results derived out of these approaches, the accuracy of the results is of most concern. This concern can be addressed by verifying the model for its quality. This chapter focuses on dedicated methods that could be used to evaluate the results of the models.

4.1. Introduction

The accuracy and consistency of a model can be determined only when the model is compared to the actual sequence of events. The actual events have to be represented in a form that most accurately reflects every details of its real-time happening. This representation has to be later reproduced in way that can be compared with the models that have been presented in the previous chapter. Model validation is usually defined to mean "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" [Sch79] and is the definition used here.

A systematic comparison of model predictions with reliable information has to be done in model validation. Validating a model over a broad variation of events would qualify the efficiency of a model. It could help to articulate different models over different natures of events. A reliable data source that describes the events as accurately as possible should be used to generate a data set which can serve as a reference. It should describe the same meaning of events and time as the data in the log files. The reference should be considered as ideal data with a minimum of errors, in ideal case none at all. This reference should be available as the ideal representation of a subject's walk.

A model can be accredited when it is questioned for an accreditation criteria according to a specified process. Credibility of a model is also of concern, if a potential application would have to develop sufficient confidence when it depends upon the results of the model. A model is developed for a specific purpose. It has different meanings based on the purpose. Hence the model has to be validated for that purpose. If not all a variety of use cases has to be forethought and either simulated or experimented to define the domain of the models intended application. These use cases have to focus on certain primary goals common throughout the domain. The model may be validated for its accuracy in this domain, but they can be invalid for any other domain. These use cases are represented as patterns. Each walk on the Smart

Carpet is a pattern, and each pattern has its own unique character and a common goal: an undisturbed walk of a single person on the sensor floor. A model can be considered valid for this domain, if its results are within an acceptable range. The model can be said to have better performance, if its accuracy is better than the counterparts. The best model, must produce valid results in most of the use cases if not in all, and should have the best performance.

4.2. Video Analysis

As the results of the models are specified, the acceptable range of tolerance in the results is also defined. In Chapter 3 when the specification of the models was defined, each model was focused to deliver the coordinates of a trajectory the subject took while walking on the carpet. This partly considered the validation and verification as a part of the model development process. Since the results of the model consist of two facets: spatial and its corresponding time information, the reference should contain each of these quantities as accurate representations of its real-time sequence. Firstly, the nature of the model output is determined. It is an array of numerical quantity, based on the foot locations on the carpet. The size of this array would be pretty small, since the size of the carpet is confined to the available prototype, and the subjects walking on this prototype will have very few steps registered on the sensors.

A model that can predict 90 % for given 10000 entities accurately is far more reliable than a model that is tested for 100 entities. Comparing all the patterns a subject would walk on the carpet, each pattern could generate 11 footsteps at an average. This would create an array that holds 11 spatial coordinates and corresponding time stamps for a walk. This is a very small quantity to evaluate a model. A practical approach that can generate more data has to be realized. One way to do that could be: taking an image of the trajectory the model plots, extracting the spatial coordinates from the trajectory alone, and comparing that to a reference. This part is crucial and has to be decided before the models are developed. Each footstep is a single entity in the reference. However, the footsteps might trigger more than one sensor during the walk. This might generate clusters that might not be equal to the number of footsteps. Hence a common base has to be identified between the results of the model and the reference to be compared with.

4.2.1. Overview

If the results of the models are plotted on an image, a straight walk on a normal VGA resolution image would contain between 480 by 640 pixels in one plane. We could consider the pixels as entities. This is definitely larger than just 11 entities of foot step locations. It is possible to represent the results of the models as an image that directly reflects the dimensions of the space occupied by the actual Smart Carpet. To start with, the sizes of the sensors, the carpet and the foot of the subject are scaled to a uniform scale and an image is generated based on this scale. The midpoints between the clusters are then plotted on this image using affine transformation on each spatial coordinate. Now this image can be used as the result of a model, that has to be evaluated.

As we now have the results, we can specify how the reference can be. The reference must be an image, too. Numerous approaches have been made towards interpreting gait and algorithms for gait recognition and understanding, using video sequences [KRCC95]. In the majority of approaches, the camera is placed on a plane vertically or horizontally parallel to the subject. Gait information is then extracted from a set of feature points in each video frame. They are plotted or used to train a neural network based model. Since the models need an image from which spatial data has to be extracted, a video sequence of a subject's walk can be condensed to this single image. Since the footsteps of the subject were the target of observation, a camera on the horizontal plane above the subject's head pointing vertically down, could be used to observe the subject's foot steps. The result should be a single image that has a thin line of the subject's trajectory on the carpet.

To plot the trajectory of a subjects walk, the center of the body has to be identified on the walk and all the centers have to be joined to form a line. A simple experiment was done to verify this task. A camera was fixed on a tripod facing down. A red can was rolled under the camera. As the can rolls, each captured frame is individually extracted out as shown in figures 4.1(a) and 4.1(b). Then the complete set of figures was collectively added together to form a single image (fig. 4.1(c)). This provided a single stripe of line or rather the trajectory the can took on its motion. Image addition was done using Java Advanced Imaging libraries.

(a) Can at position 1

(b) Can at position 2

(c) Path of can roll over

Figure 4.1.: Testing image addition

The same experiment can be repeated for a subject by placing a red cap on the head and observing the walk on the carpet using a camera one meter above the head. The frames can be removed and the image can be added (like in fig. 4.1 on the previous page) to retrieve the trajectory the subject took. This would provide an exact real time trajectory of the subject during the motion.

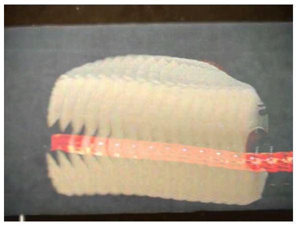
4.2.2. Camera Angles and Distortion

A suitable height was taken to cover the subject and the Smart Carpet in the background. This height was increased to fit the corners of the carpet. However, it was possible to fit only one side of the edges. At a height of about 75 cm above the subject, the camera was fixed on the ceiling. It was controlled remotely. At a command the camera was switched to record. At the same instance a subject, wearing a dark cap with a red dot in its center, was made to walk on the carpet (fig. 4.2(a), 4.2(b), and 4.2(c)).

(a) Subject at position 1

(b) Subject at position 2

(c) Subject at position 3



(d) Trajectory of Subject

Figure 4.2.: Implementing Image addition

The figures were extracted and then the final image was constructed by adding each frame of the motion. The resultant image (fig. 4.2(d)) did have the information needed to evaluate the models: the trajectory of the subject during the motion on the carpet.

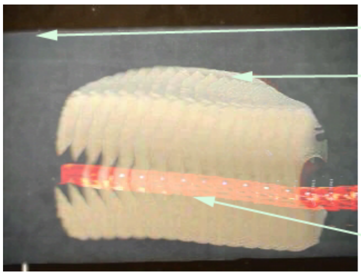


Figure 4.3.: Errors in capturing trajectory

The timing information is available in the video frames. A PAL coding was used with 25 frames per second. The subject initiated the walk at the sound of a start signal. Although this process was an automated task, with minimum human intervention, it had lots of errors and this method has some drawbacks,

- The edges of the Smart Carpet and the screen of the camera did not fit properly. This leads to wrong spatial coordinates on the trajectory. Further, the camera lens is fixed for a VGA resolution which is not the size of the Smart Carpet. Shaping the carpet to fit the size of an observing camera would be a very expensive reference. An alternative approach would be to hide some parts of the carpet with a dark cloth or a color. The background distortion comes into picture due to light reflections on the floor and surrounding surfaces. A uniform background color cannot be constructed unless the entire background, including the carpet, floor, surrounding walls and the subject is completed painted in white.

- Due to the altitude of the camera mount, the distance between the camera and the subject increases. This is not the case on the red can experiment where the can was relatively in a viewing distance. Between the camera and the can, light reflections were minimum but still shadows casted on the image were clearly observed (fig. 4.1(c) on page 71). Due to the increased distance between the camera and the subject and also due to the restricted range of the focal lens, perspective distortion is introduced. Although error correction algorithms can be used to recover the distortion, they modulate the spatial coordinates to a large extent.

- The trajectory shown in figure 4.3 is quite thick to serve as a reference. The thickness of the trajectory can be maximum of one pixel to determine the exact center of the subject. When the red line was reduced to a thin line, some of data was lost during the addition of the frames.

This method can be used to generate references for a subject walking on a straight line. But when the subject walks in curves, the trajectory looks like something piled on top of the carpet. This is largely due to perspective distortion. Hence a better approach must be identified.

4.3. Video Mapping

This method is used to derive a source that can represent the real time sequence of events. The representation must be in a discrete form that it could be used to compare the results of the models directly. Enough data points should be identified to increase the confidence level of the methods. To achieve this purpose more than one camera can be used to observe the subject while walking on the carpet.

4.3.1. Overview

One camera is fixed on the ceiling to monitor the horizontal plane (C). This camera is referred to as the horizontal camera. Two more cameras, one on the x-axis (A) and the other on the z-axis (B) were used to closely monitor every step the subject took on the carpet. These cameras are referred to as the vertical cameras. Figure 4.4 shows the position and the angles of the cameras placed. In these camera positions, the exact locations of the feet can be identified and the angle of the foot with respect to any reference along the planes can be observed.

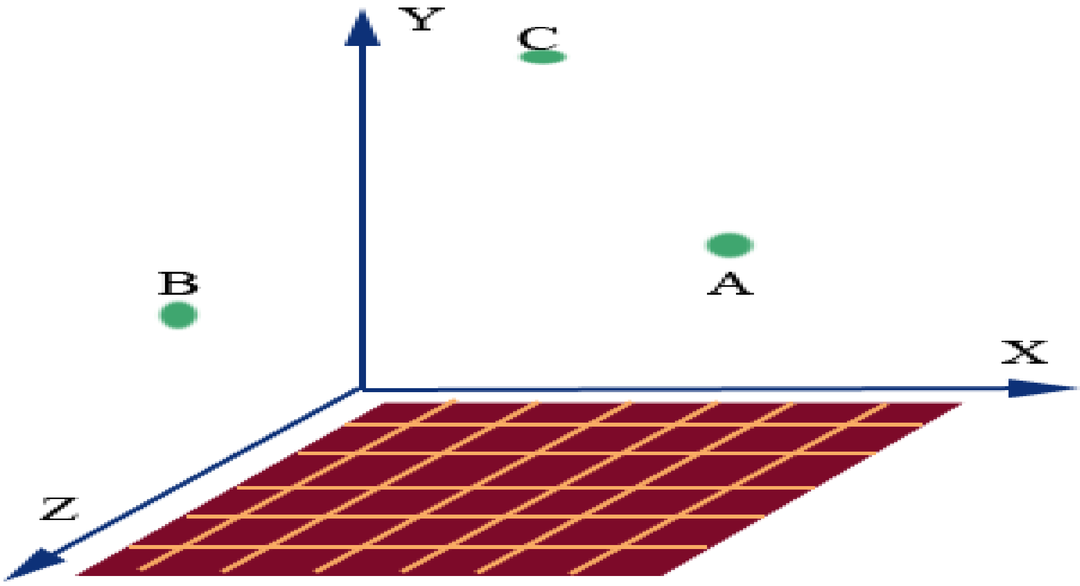


Figure 4.4.: Position of cameras at points A, B and C

The vertical cameras were focused only on the foot of the subject. This approach gives a precise view of where exactly the subject placed the foot and how long it was on the carpet. It reduces the errors of perspective distortion since the camera is held closely to the target and focused on a particular object. To enhance further accuracy, the carpet was painted with lines of regular intervals in X and Z directions to produce a visible grid like structure on its surface. Packing tapes of 5 cm width were stuck on the carpet to produce these painted lines on the surface. The tapes were stuck exactly on the place where the power and communication wires pass thru between the sensors. The size of the grid was 15 cm. This reflects the underlying sensor layout structure on the surface of the carpet. Hence if a subject kept the foot on the middle of the grid, it is like keeping it exactly on the middle of the sensor. Further the grids were marked with lines in between the cross overs points of the tapes (fig. 4.5). The lines had a gap of 5 cm providing a resolution of 5 by 5 cm grid on the whole carpet.

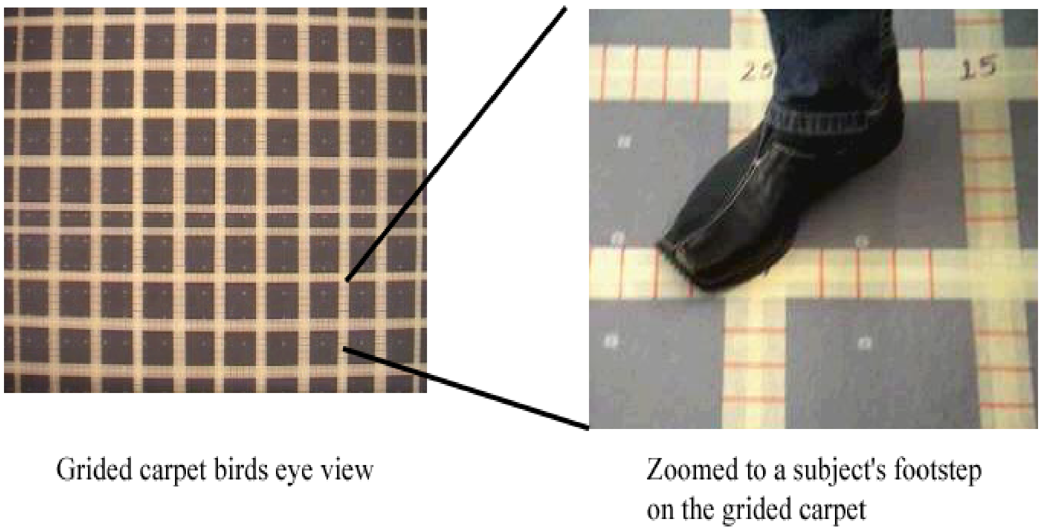


Figure 4.5.: Griding the carpet

The goal is to obtain an image that ideally reflects the orientation, location, and time of each foot placed on the carpet, when a subject walks on it.

4.3.2. Process

Since there are three cameras to be operated simultaneously, they have to be synchronized to a common point of reference for a uniform time scale. This common point of reference, could be used as a starting reference to identify timing information for each foot step. The time difference between the start of the video clip and this reference point can be neglected. But the millisecond after this reference point would be the start of the time line for the rest of the video sequence. Hence all the cameras should be turned on before the capture of this reference and the cameras can start to record irrespective of their turn-on delay. The reference point was the sound of a hand clap that the subject has to make before starting the walk. This sound would be recorded in all the cameras at their respective time stamp. Considering the speed of sound and the vicinity of the cameras, the delay in propagation of this reference can be neglected and the time stamp of this sound in each camera can be considered as a common reference. Should a camera fail, or the carpet develops any abnormalities or any sensor fails or the subject is not feeling comfortable to walk, the experiment can be terminated before this point until the required steps are taken to restore a calm comfortable working atmosphere.

At the instant, when the cameras are turned on, the vertical cameras are focused to the subject's foot that would make the first contact on the carpet. The subject does not start directly on the carpet, rather, a few meters away from the carpet and the walk finishes after the subject completes the required trial of motion on the carpet. The camera mounted horizontally would cover the entire carpet in its field of view. This camera can be used to capture the frame when the subject made the first and last foot stamps on the carpet. Although the same aspect is observed on the other cameras, this view is a good reference to know on which sensor block was the footstep started. Consecutive foot steps can be traced on the number of block the subject's transition lapped on the foot swing. As the vertical cameras are focused on the subject's foot, a close view would cover the entire foot and the sensor around it. To track on which sensor the subject has kept the foot, the horizontal camera is used as a reference. Alternatively, the grids were numbered at their crossover junctions and their numbers are just a reference to know which part of the carpet is the subject on at any moment. Although these numbers directly correspond to the sensor number they are not the processor numbers of the network as they are not intended to reference them.

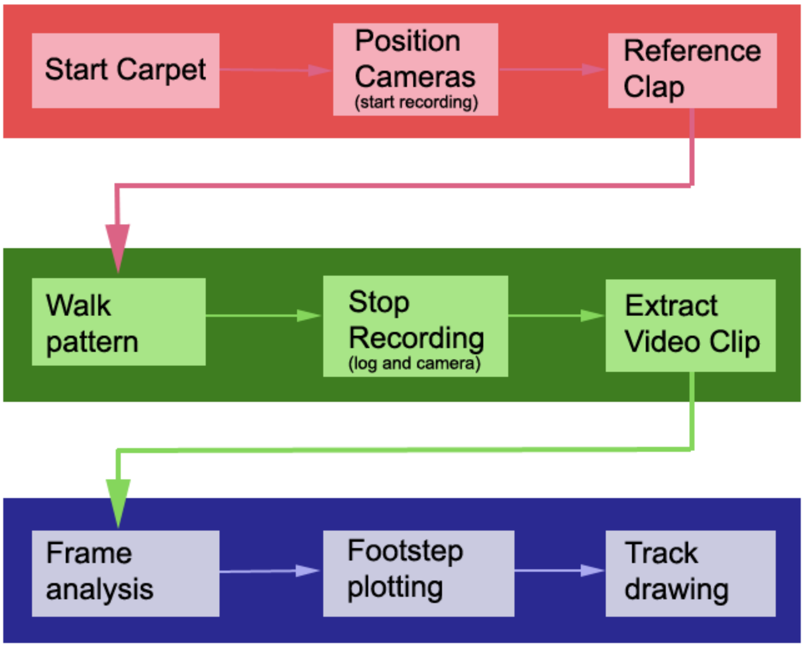


Figure 4.6.: Video mapping process overview

After the Smart Carpet is turned ON, i.e. in operational mode, the cameras are positioned and the recording can start. During start up of the network inside the Smart Carpet, a log file is opened by the system to record events with time stamps. The cameras can start to record now. After the reference clap the subject walks on the carpet for a specified pattern. As the subject leaves the carpet, the cameras and the carpet are turned off. This ensures a complete observation of the subject's motion by two independent sources: the camera and the carpet. This video clip is then extracted from the recorded medium. The time before the clap and after the subject's last footstep on the carpet is not considered as they do not involve the motion of the subject on the carpet. Hence this part in the video clip is removed. In the next step (fig. 4.6 on the preceding page), each frame is analyzed from all the three cameras simultaneously to observe the location of the foot and the time stamp. An image with the size of the subject's foot is embedded on an image that represents the Smart Carpet. This image of the carpet is the same captured by the SCM software without the subject on the carpet. It is of the same size as the one produced by the data mining algorithms and serves as a reference source image. The footage of the subject is placed carefully on this image. Section 4.3.2.2 details briefly this process. A thin track is drawn joining the footage, which is the trajectory of the subject on the carpet. Although this process involves laborious observation of each footstep from the subject on three angles, it is worthy to generate an ideal reference. The timing information is calculated from the frames and tabulated in an excel sheet.

4.3.2.1. Scaling Feet

Considering the structure of the carpet, the size of the sensor is 15 by 15 cm and the gap between two sensors is 5cm in all its sides. This is directly reflected in the markings on the surface of the carpet. The tapes are stuck at a distance of 15 cm between them and the width of the tape is 5 cm. The carpets were manufactured in modules having a width of 125 cm and 195 cm length each. They where joined together to form one uniform network. At the joining of the carpets, there are no sensors present. This junction is having a 10 cm sensor-less carpet connection. This part is also identified and marked on the surface of the carpet. The image generated by the SCM software considers these dimensions while displaying the carpet on the monitoring GUI. This image is a uniform image without any joins. It shows one uniform image of 120 nodes with an inter nodal gap of 5 cm. This image has to be modified to consider the physical structure of the carpet. Hence the SCM was modified to display an image that accommodates the junction between the carpets.

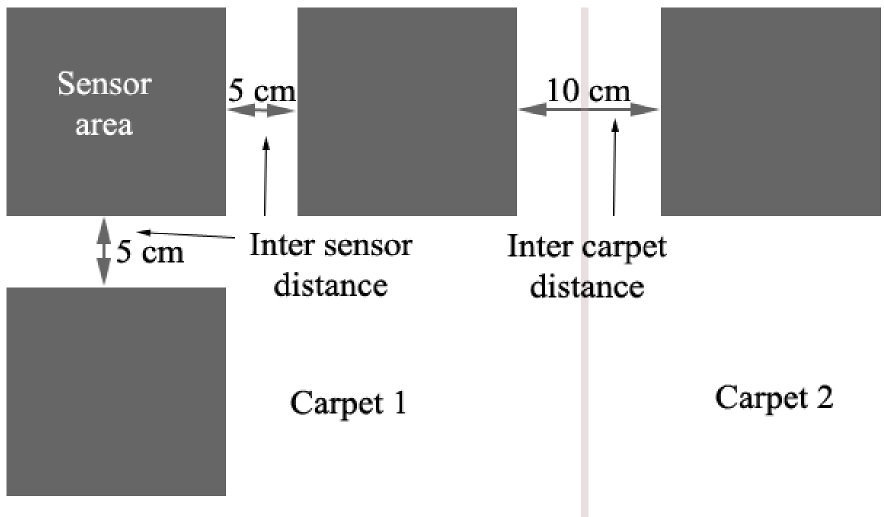


Figure 4.7.: Scaling the carpet

The new image generated by the SCM consists of 15 cm for each sensor. There were 6 sensors placed in the width of the carpet. The inter sensor gap is 5 cm. There were two carpets joined and the inter carpet distance is 10 cm (fig. 4.8). Considering these physical dimensions, the width of the carpet was 240 cm and the length of the carpet was 195 cm. The SCM algorithm was modified to consider these details and draw an image representing two carpets joined at the inner edges. This generated a picture of size 694 pixels in width and 616 pixels in length. Considering the inner edges the virtual carpet on the SCM is 240 cm in width and 195 cm in length. This means an average of 3 pixels for 1 cm.

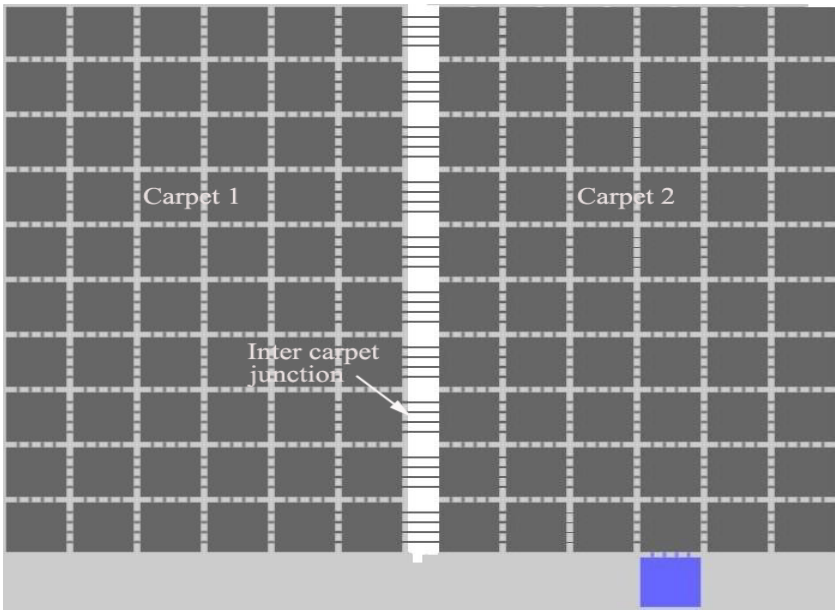


Figure 4.8.: Modified SCM image

This was important to have a uniform scaling from the physical size of the carpet and the virtual image seen on the GUI. After this image was generated by the SCM, the foot of a subject is imposed on this image by hand (fig. 4.9 on the following page). This was done by observing the location of each foot in three cameras, frame by frame and identifying the most possible location and patching the foot of the subject on this image. To do this, the subject's foot has to be measured and the image of the shoe has to be scanned. It was scaled to a size that 1 cm corresponds to 3 pixels. The shoe was carefully drawn with a uniform color. Then the center of this virtual shoe was identified by a different color. This was done to calculate the actual center of the subject. The center between two feet centers would give the approximate center of the subject. Joining these centers would produce the actual trajectory of the subject's walk.

This image was drawn for all subjects who took part in the walks. A walk at an average would have 5 to 10 feet on this image depending on the pattern the subject took. As indicated in figure 4.9 on the next page, the center of a foot has a distinctive pixel color. This is used to calculate the center of the subject. After the center of the subject is identified at all the foot transitions, a straight line is drawn between midpoints to identify the trajectory the subject took during the motion on the carpet.

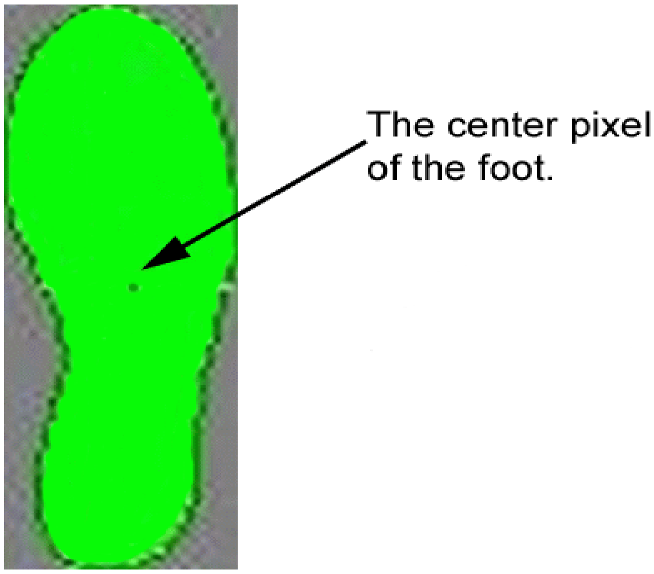


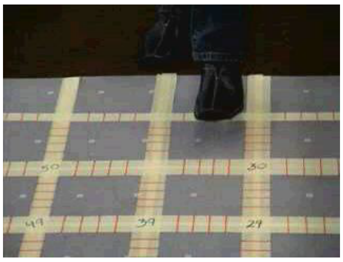
Figure 4.9.: Scaled right foot of a subject

4.3.2.2. Tracks Extraction

When a subject walks, the first contact to the ground is the heal strike. As the body moves forward, the foot becomes parallel to the ground and the transfer of body weight takes places. After the video clips are extracted for the subject's walk, each frame is then closely observed for this time instance where the foot of the subject lies parallel to the ground. The time at this state is considered as the instant, when the sensor is triggered as a result of foot stamp. In a video sequence of a subjects walk (fig 4.10 on the following page), the frames 1 to 9 show the different locations of the feet, during the motion. The video is taken using a PAL codec. The frame rate for a PAL video is 25 frames per second which leaves 40 msec for each frame. The frame number, when the clap reference was initiated, is noted. This frame number is the starting frame number. In a simple walk, considering the clap reference occurred at frame 31, it can be seen in figure that at frame 2 the subject had the feet parallel to the ground. The next foot that was kept parallel is in frame 5. The time difference between the two can be calculated as (5 - 1) by 40 msec which is 160 msec. In reality there are a lot of frames between frame numbers 5 and 2. They are edited out for this presentation (fig 4.10 on the next page). This time difference is used to calculate the mean error distance between the estimated time difference of the models and the real time difference between two consecutive foot steps.

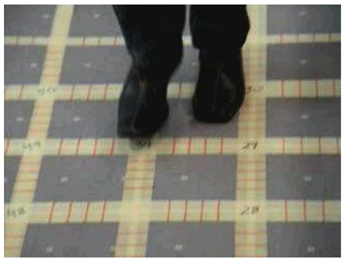
The time instance of these foot steps were updated on a spreadsheet with corresponding frame numbers, subject identity, the pattern on walk he had taken, and a video clip identification tag. The time difference is then used to calculate the cadence and the speed of the subject's walk. It was observed that subjects walked slowly in curves compared to their walk in straight lines, which was consistent with the parameter estimation problem for a fixed time difference on the entire walk.

The subject had made 10 steps in this walk. The first three steps of the video sequence are extracted and shown in figure 4.10. The time information for this sequence is noted like in table 4.1 on the next page. Since the reference clap is recorded in all three cameras, the location of the foot step while comparing the time stamp is consistent in all the cases. Hence a single time stamp is considered as there exists no difference when the cameras are synchronized.

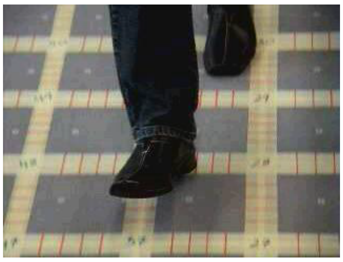


(a) Frame #1

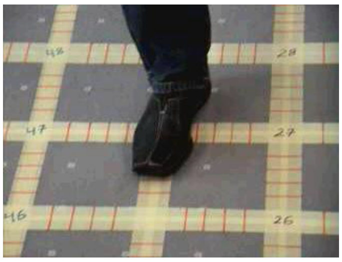
(b) Frame #2



(c) Frame #3



(d) Frame #4



(e) Frame #5

(f) Frame #6

(g) Frame #7

(h) Frame #8

(i) Frame #9

Figure 4.10.: Sequence of first three steps in a walk

This procedure was repeated for each walk of the subject. After the foot size of the subject is scanned, the image is skewed to the scale of the reference carpet image. This image is then placed on the reference carpet image observing the video clips. The resulting image for the complete sequence of the walk is shown in figure 4.11.

Table 4.1.: Recording time information for a walk

Total Steps	Duration (s)	Cadence (st/mm)	Stride (m)	Step (cm)	Speed (cm/S)	Average Time Dif (s)
9	6.2667	86.170	0.8922	51.91	64.069	0.696

Step	X pixel	Y pixel	Mid X pixel	Mid Y pixel	Step dist (cm)	Stride (cm)	Frame nr	Time (s)	Time Dif (s)
Left	520	0	495	82	57.15	112.574	50	1.66667	0.767
Right	470	164	470	249	56.666	73.4133	73	2.43333	0.633
Left	470	334	388	322.5	55.201	104.469	92	3.06667	0.600
Right	306	311	240	271	51.45	62.301	110	3.66667	0.600
Left	174	231	211.5	182	41.135	76.829	128	4.26667	0.733
Right	249	133	286.5	94.5	35.829	76.369	150	5	0.700
Left	324	56	400	110	62.153	100.788	171	5.7	0.800
Right	476	164	470	244	53.483	107.0332	195	6.5	0.700
Left	464	324	441.5	402	54.12		216	7.2	0.733
Right	419	480					238	7.93333	

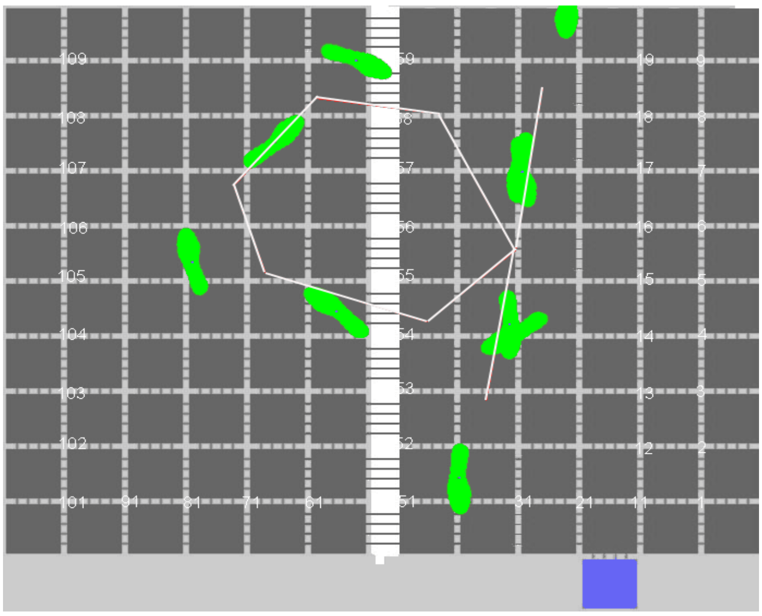


Figure 4.11.: Locating footsteps on the reference image

The footsteps of the subject are carefully observed for their location by observing the adjacent sensor number, the scaled markings on the girded tape. This image is compared with all three cameras and finally the scaled image of the foot is then pasted on this location. The image is rotated to the near exact angle at which the subject kept the particular foot.

After the walk is constructed, the midpoints of X and Y coordinated of each foot are noted in the table 4.1 on the previous page. Then the midpoints between two foot steps are calculated by using the following equation (4.1),

$$X_{Mid} = \frac{X_2 - X_1}{2}, \quad Y_{Mid} = \frac{Y_2 - Y_1}{2} \tag{4.1}$$

A thin line of 1 pixel width is drawn joining these midpoints. This line is the trajectory, the subject took while walking on the carpet. Since the plotting of feet was done using Adobe Photoshop, it was possible to save this track alone as a single layer image without any background deduction (fig. 4.12). Hence a single image without any background noise or perspective distortion was attained to serve as an accurate reference.

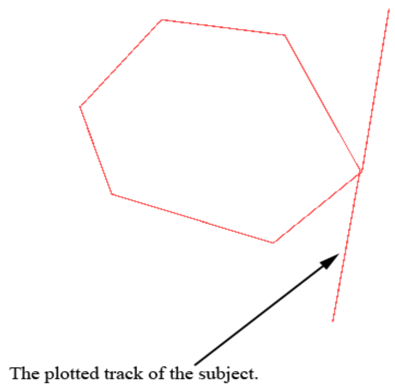


Figure 4.12.: Reference trajectory of the subjects motion

4.3.2.3. Pixel Mixture

This reference image has to be compared with the image that is generated by the data mining models. In order to do that, the track alone has to be extracted out from the image. The extracted information will consist of X and Y coordinates of the trajectory. The reference image does not have any background noise and hence it is possible to extract the trajectory alone. This would result in a set of pixel location information. This process has to be done on both the reference as well as estimated images. The difference between these two data sets would give the mean square error which determines the accuracy of the data mining models.

Since the images are having a background of uniform color, the trajectory is plotted by a specific color for the complete path. During extraction, the X and Y coordinates of this particular color in all its occurrences would describe the location of the trajectory in the image. This would then be a data set that can be numerically analyzed. Usually the image is scanned from the left to the right by any algorithm that processes the pixel information of an image. Imagine the subject had been walking a pattern "6" i.e. the subject traced the number "6" while walking on the carpet; the subject had the possibility to start the walk either from the right and finish at left or the subject could start from the down and finish on the top right. In either case, the sequence of the extracted pixel locations would not be true to the timing information.

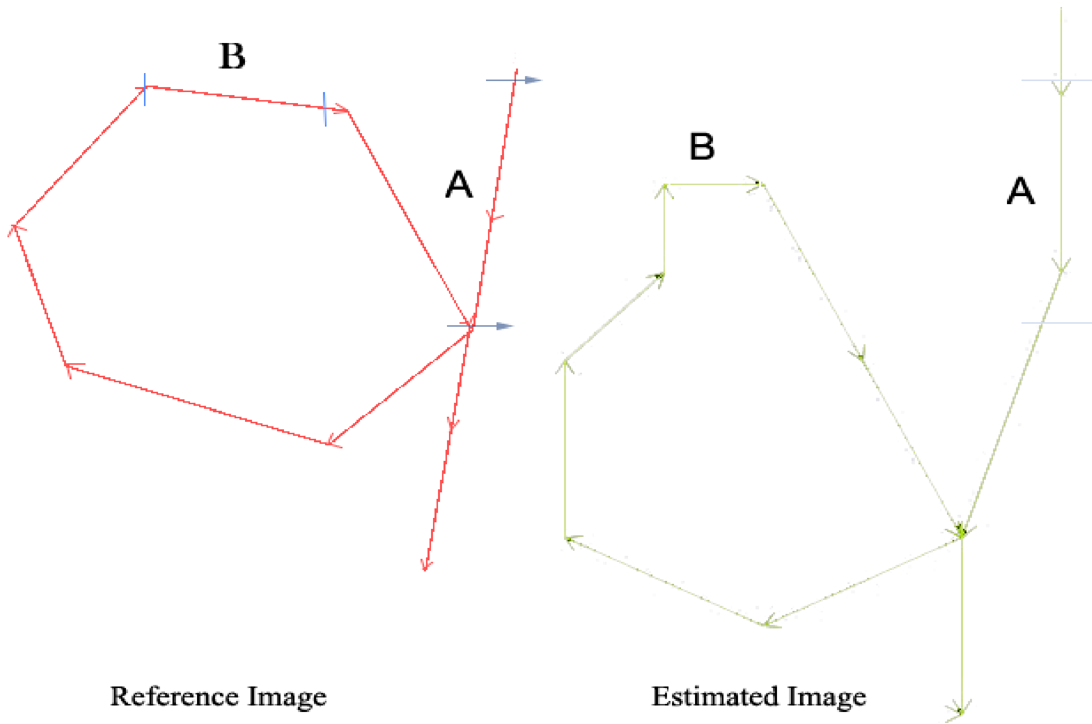


Figure 4.13.: Problems on comparing Image Pixels by direct scanning

When comparing the reference and the estimated image, in most cases it was observed that the data set was not consistent. Considering the pattern 6 for example, the reference image of a subject's walk on the carpet and the estimate image by the data mining models are shown in figure 4.13. Here it can be seen that the subject started from top right and finished the walk on the bottom right. When the image is scanned from the left to right, X and Y coordinated of the pixels in section A and the section B of the track would be mixed up together. When a horizontal or a vertical scan is done either from left to right or from the opposite side, data from different segments of the track are always mixed up resulting in a pixel spaghetti. Hence a more suitable approach has to be considered when extracting the comparable data set of pixels.

4.3.3. Comparison

One possibility to compare the two images would be to apply a regular grid on both the images. Since the reference image and the predicted images do have equal resolution and size it is possible to fix a regular grid on the images. Then the grids can be numbered. Now information can be extracted from a particular grid, either manually or automatically by an algorithm. The result would look like in figure 4.14. The crossover points of the X and Y axes of a particular grid can be taken from the reference image and compared to that of the predicted image. This leads to sampling of the image based on the grid size. To sample the image a consistent window size for the grids is identified by a trial and error method. The grid has to have a window size. The number of pixels in the size of the grid has to provide an acceptable error tolerance.

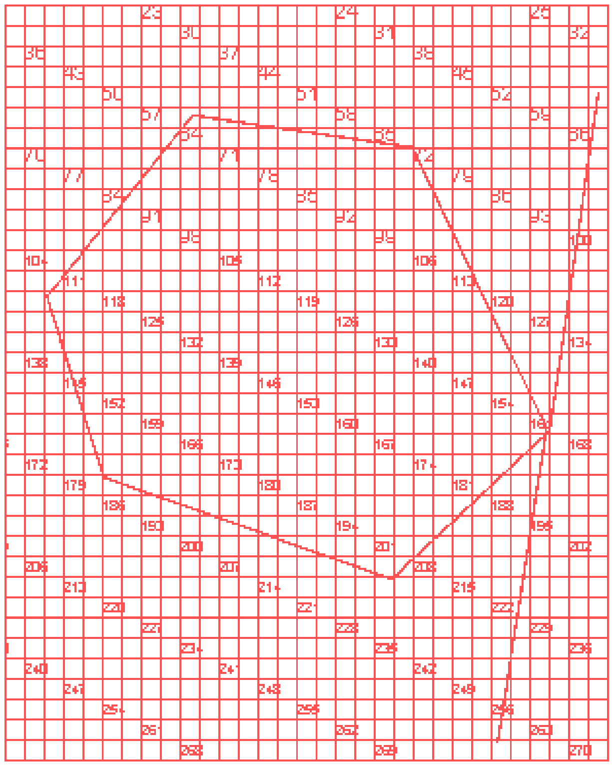


Figure 4.14.: Girding the Reference image

The reference and the estimate images were girded. The cross over points between the grid and the track were taken for different grid sizes and the mean square error between the images. This process was repeated for a number of images. It was observed (fig. 4.15 on the next page) that the deviation of the MSE tends towards a consistent value when the grid size is reduced from a large number.

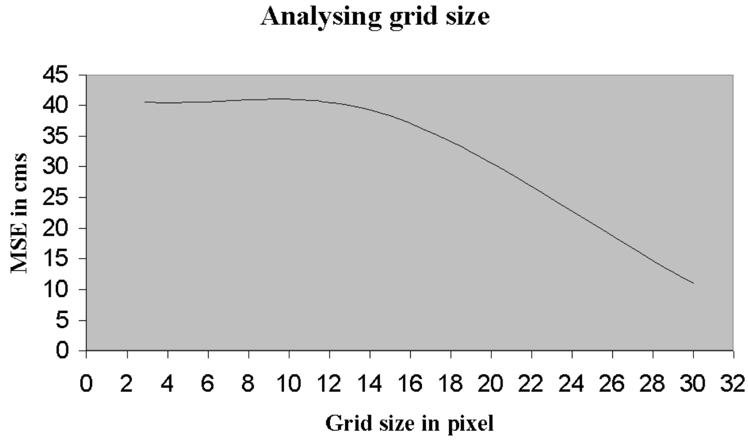


Figure 4.15.: Grid size for suitable comparison

Hence an acceptable grid size of 15 pixels was taken between the X and Y coordinates. The pixels at the crossover points of the trajectory within a grid are then extracted to an Excel sheet. This approach can be used as a stable accurate reference.

4.4. Simulation

The observation done so far had been in real time with subjects walking on the carpet. Using the methods described above, it was possible to observe the walk of a subject and determine the trajectory of the subject's walk in a well referenced environment. However the consistency of the observed results can tell more about the stability of the developed models.

4.4.1. Overview

To determine the validity of the models, stable references were generated that reflect the real time data most accurately for later analysis. Different methods were tested and a most reasonable one can be applied. In the above methods described so far, the observations were done on a carpet which was 240 by 190 cm which is about 4.5 m^2 in size. At any instance a subject can walk a minimum of 5 steps either straight or more while making turns and patterns during the walk on the carpet. But in order to prove the consistency of the models, it would be much favorable to have a carpet of a large area.

Currently the construction of the carpet was intended as a prototype demonstrator to prove the feasibility of a smart surface which can detect foot steps. This setup was used to develop means of identifying the trajectory of the subject while walking on the carpet which is the main contribution of this thesis. In order to continue further research on this topic a large carpet has to be constructed and various methods of motion tracking can be enhanced.

The current size of the carpet attenuates the natural walking patterns of a subject due to its restricted size. While performing the previous experiments, the videos were further analyzed to determine the walking behavior of the subjects for particular patterns. This was repeated for a range of subject's heights and ages. It was observed that in certain walking patterns like 8, S, C and I, where the curves in these patterns start from a radius of 20 cm and extend towards infinity (for a straight line), tall subjects tend to be much slower in the curves compared to short ones and kids (fig. 4.16).

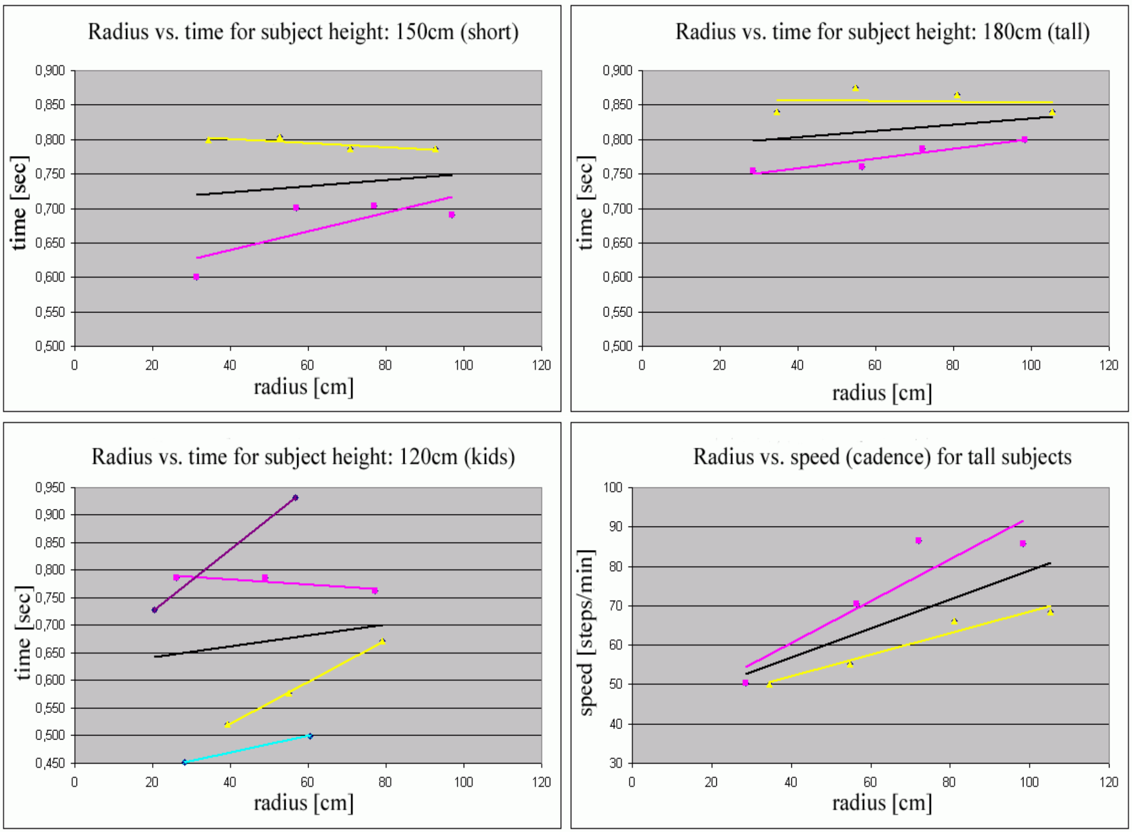


Figure 4.16.: Walking behavior of different subjects on curves

Can this be possible because of the finite structure of the carpet and the room [RS03] ?. If the area of the carpet is a problem then an alternative means of simulating this area can be addressed. The carpet and the network inside the carpet can be simulated virtually by a simulation environment. By simulating, the carpet can be of a desired size and thus allows testing the data mining models for its consistency.

4.4.2. Requirements of a Simulation Environment

Considering simulation, the target of simulation can be much simplified in its representation. Which means, the carpet can be considered as having only sensors and the sensors having a predefined number and a location which is starting from a specified end of the representation space. This is not the case in the real carpet. Some of the sensors are not numbered in a sequential manner from top to bottom uniformly. Rather, some are numbered horizontally and some are vertically routed towards the host PC. This randomness is not of importance for testing the data mining models as the models are considering only the sensor data. Hence on a simulation environment the carpet can be numbered in any direction uniformly.

The simulation environment should also provide the option of recording a log file for the sensor activation. The network information can be identified on the log file. This format has to be compatible with the existing data mining environment where the models can directly use the data from the simulation programs. The simulation program should be able to reflect the walking behavior of the subject. Enough parameters should be provided to replicate the exact patterns and behaviors of the subject. The carpet and the subject are two independent entities. The events of the subject on the carpet should be well controlled and recorded by two independent methods.

4.4.3. Parameter Extraction

By simulating the entire system, we identify two independent entities, the carpet and the subject, that have to be characterized. The carpet need not be entirely simulated, rather, the sensors underneath, with a unique identification number for each sensor as the most important characteristic features of this entity. Hence a GUI should be available that determines the number of sensors, the length and the width of the carpet. Using this data, the sensors have to be simulated. That can be easily displayed with the normal working performance of the PC where the simulation is done.

Considering the subject as an entity, there are a variety of parameters that have to be considered that directly reflect the walking behavior. However, complete and accurate characterization of gait dynamics requires knowledge of the kinematics of tens, if not hundreds of body landmarks (such as joints and extremities) [DK97]. But in this simulation, only the sensors need to be activated. Hence the distance a foot moves, the time the foot remains at a locations, the angle it turns at curves and the speed of the foot at various radius of curves are of prime importance. These parameters can be observed from the videos that had been taken for specified patterns. Each person exhibit individual gait features for these patterns, that have to be parameterized and applied when the subject is simulated to walk a particular pattern.

Observing the height, a subject has a rhythmic up and down movement of the upper-body while walking. This bobs the head of the subject between a range. It can be said that the height of the subject is a time-dependent quantity. Because the subject bends or droops few centimeters while moving forward. Stride length is a function of this height. This height is different from the stance or the height of a person when the subject is not in motion. To extract this parameter from the video sequences automatically is quite difficult as it involves a lot of background noise that has to be detected and the result is not an accurate estimate rather a rough estimate. Hence stance height can be considered and varied within a certain limit for the sake of realism.

Different walking models of pedestrians have been thoroughly researched and simulated. In most models in the literature, the important parameters influencing the walking behavior are the desired speed and speed regimes. These are very important when we simulate the foot steps walking on the carpet. In the social force model [HM95], the desire to adapt the actual velocity $v_i(t)$ to the desired speed v_t within a certain *relaxationtime* τ_i is reflected by the *accelerationterm*:

$$[v_i e_i - v_i(t)]/\tau_i \tag{4.2}$$

where $(v_i e_i)/\tau_i$ is referred to as a driving term and $vi(t)/\tau_i$ is referred to as the friction term. The Cellular Automata model [Klu03] addresses the problem using dynamics of the floor field. Here, the movement of a pedestrian is considered as a particle movement, that crosses a field with its own dynamics of diffusion and decay. In the lane-based approach, Blue and Adler [BA02] design the walking model to account for variations in walking speeds observed in the real world. Each time step is one second and they consider walking speed varying among pedestrians, using distribution of walking speed with a cell size of 0.21m^2:

- fast walkers: maximum speed of 4 cells per time step (about 1.8 m per time step)
- standard walkers: maximum of 3 cells per time step (1.3 m per time step)
- slow walkers: maximum of 2 cells per time step (0.85 m per time step)

In their experiments, they use a population or a crowd of people, composing 5 % of fast, 90 % of standard, and 5 % of slow walkers. In an overview the speed of the individual is limited by the physical conditions and near by individuals or obstacles. There is a set of admissible velocities defined as:

$$V_a(t, x) = v :\| v \| \leq v_0(t, x) \subset R^2 \tag{4.3}$$

where t and x describes the changing of the maximum speed stemming from the change in flow conditions as well as differences in maximum speed between different parts of the walking infrastructure. The maximum speed of the subject also depends on the individual's characteristics (e.g. age, gender, trip-purposes, luggage etc . . .).

4.4.4. Decision Making

An important aspect often not visible was the decision making process of the subjects when they walk. In determining the route, a subject would take to trace a pattern, they pause, calculating in their thoughts and arrive at a decision, during the curves or before the curves.

This decision involves, which is the most comfortable route to take, or attributes related to path distance and time [HBD02]. This contributes to the slowing down of walking speed during the curves. Although this is in millisecond range, it directly affects the estimation of time difference which instruments in clustering the members.

For a rough estimate of the subject's walk, the height of the person, the radius of the curves, the speed limits of the walk during curves and straight lines, the age, sex and foot size of the subject were all taken as parameters and fed to a simulation program.

4.4.5. Simulation Environment

Here are a few words shedding some light on simulating gait in various contexts. As already observed, there are two entities that have to be simulated: the carpet and the subject. There is a behavior for each entity. There are a variety of simulations that range from individual gait to context based simulations. However, agent based simulation programs are of considerable interest in this context to represent pedestrians in a walking mile or a shopping complex. Majority of simulations are aimed at finding the shortest path and subject behavior while exercising emergency tasks such as evacuation from a building under fire. Some of the existing simulation programs are mentioned here:

- PEDSIM [Glo05] is a software tool for microscopic pedestrian and crowd simulation. It implements the social force model.

- NOMAD [Daa04] is a microscopic simulator developed at the Transport & Planning department of the Delft University of Technology. It is based on microscopic characteristics of pedestrians, such as walking speed, pedestrian size, etc....

- Simulex [TM94] is the evacuation simulation part of the IES Virtual Environment, a set of tools developed to aid in the design and evaluation of buildings.

Although these programs are highly focused on infrastructure based approaches, they can be used to model the carpet. But the simplicity and the use of the carpet sensor does not require programs and environments targeted towards evacuation or movement of the subject towards a particular goal by choices of his own based on his character, rather, the subject has to walk in a predefined path based on a characteristic behavior.

NetLogo is an agent based simulation system where the network in the Carpet can be easily realized. It offers the possibility to model the sensors in the carpet by using agents. Agents in Netlogo [Wil06] are referred to as Patches and Turtles. There can be any number of patches and turtles at the same instance. Each agent can be configured individually by a characteristic behavior or shared a set of behaviors. In this usage, the sensors all have the same property of sensing and reporting the sensed data, but still each sensor has a unique representation or an identity which suits either of the agents as a well for this task.

On the other hand new behavior or parameters can be introduced to the agent to customize a character or a subject. These parameters can be either the subject's height, foot size, average step width, speed, etc. The patches were considered as the sensors and the turtles were considered as the feet of subject. The specifications of the simulator can be summarized as follows:

- The simulator should simulate the layout of the nodes or sensors
- Offer a possibility to alter the size of the network
- Create log files for recreation of the network
- Simulate subjects
- Profile subjects
- Describe a pattern which specifies the characteristic of a path for the subject
- Log sensor information based on activation

The simulator need not simulate:

- The ADNOS algorithm, rather provide a log file based on the specifications of the ADNOS algorithm, to regenerate the network on the Smart Carpet Monitor application where the data mining algorithms are available.
- The Biomechanical gait information, rather uses standard profiles of individuals and visualize their foot impressions on a floor based on speed and stride length.

There are three main components that have to be treated independently:

- The input control panel to configure the sensors and the subjects
- The display of the network with subjects and
- The log files written in the background

Apart from the above specification, the simulation algorithm or any other method have to be applied to independently record the events happening in the simulator to validate the models for their accuracy and consistency.

While developing the simulation software in NetLogo, inspite of the flexibility offered by the simulation environment, there are important parameters that have to be considered as fixed and do not directly reflect a subject's walk as in real time. To cover all aspects of these gait dynamics is deviating a lot from the core work of this thesis as it involves a large amount of video processing just to simulate a range of artificial parameters. Hence a simulation of the carpet and the gait cannot be considered as a stable reference to validate the model. As many parameters have to be calibrated and need to be assumed for the realism sake, simulation is not considered as a valid instrument for proving the consistency of the models. But on the other hand, simulation can be done for improving the carpet infrastructure where the parameters such as data rate, hop count, shape, and size are more static compared to the dynamics of a subjects gait.

4.5. Summary

The accuracy and the consistency of the models can be determined by comparing the estimated results and the actual sequence of events. Hence alternative approaches in recording the real time events of the subject's walk are important and carefully considered. It was observed that certain features e.g. exact location of the foot on the carpet while walking, have to be recorded as accurate as possible. In developing the methods, some of the discrepancies observed give a better understanding of what is more important in representing the real time events. The location of the pixels that represent the real trajectory can be a considered as a quantity for comparison. In the first method, the subject's motion was captured in a camera placed on the horizontal plane. The video frames were extracted from the sequence and summed up to a single image producing a streak of motion patterns. The distortions introduced by the camera angle and the thickness of the streak makes the image less ideal for consideration.

Alternatively the subject can be observed using three cameras at different angles and foot steps of the subject can be laid on a reference image of the carpet. This image is scaled to the physical size of the carpet and can serve as a reference. The estimated results of the data mining models are also plotted on an image which is exactly the size of the reference image. The size of the carpet and the shoe size of the subject were skewed to a single scale. Each frame in the video sequence was observed and the subject's foot was laid on the reference image. The image of the foot has a center pixel marked with a different color. A trajectory was drawn connecting all the midpoints between the center pixels. This trajectory serves as the reference trajectory. The pixels of the reference and the estimated trajectory are then extracted and exported to a spreadsheet where further analysis is done. Timing information was recorded in a separate file.

Although this method was quite accurate in observation, the size of the carpet was not that big to ensure the consistency of the models. A bigger version of the carpet is needed. Hence the carpet was simulated on an agent based simulation software NetLogo. It was observed that simulating the subject's motion contains dynamics that have to be considered with static values and the speed of the subject was depending upon various parameters like walking height which varies, thought of decision while walking and environment around where the subject is walking. These are some parameters which have to be assumed under an arbitrary range and severely attenuated the genuineness of the subject's motion. To simulate the subject's motion considering factors that influence the above parameters is beyond the scope of this thesis and the result is not accurate as it lacks reality.

Hence, to validate the results of the data mining models, the video mapping technique was considered as an ideal representation of the real time events. After the pixel locations are extracted from their images, a spreadsheet calculates the mean error distance between the original and the estimated tracks. The observations and the results are presented in detail in the next chapters.

5. Evaluation

The video mapping technique provided a unique method of capturing and representing the real time data of the trajectory the subject took while walking on the carpet. It was observed that the representation of tracks in an image can be compared to the estimated image produced by the data mining models. To do this, a detailed process of applying a grid to both the images and extracting pixel information in each grid had been explained in the previous chapter. After this data is extracted from the reference and the estimated images, the data has to be analyzed to evaluate the accuracy of the employed data mining model.

5.1. Overview

Considering the applications were the trajectory estimates of the data mining models are used, it can be seen that the results of these models are of serious concern to develop trust on a particular model in a specific usage of the model. This concern is addressed by evaluating the model with respect to certain conditions. A model can also be accredited, provided the model satisfies a specified accredited criteria according to a specified process. Model credibility, although closely related to accreditation, is the confidence that potential applications develop to use the model or the results obtained by the model. These are certain factors which can be considered while evaluating the data mining models.

As far as the trajectories of the subject are concerned, the accuracy of the estimated track contributes to the credibility and trust on the particular data mining model. Accuracy can be estimated by finding the difference between the actual trail of the subject's walk and the estimated track of the data mining model for a particular pattern or a series of patterns. This distance has to be observed in latitude and longitude. This can be compared as a combination of X and Y coordinates at random equal points.

5.1.1. Variables for Accuracy

The data mining models estimate the footsteps as clusters. The midpoints between two clusters explain the coordinates in the trajectory of the subject's walk. The trajectory is a thin line of one pixel width. The tolerance of this trajectory can be considered as a range anywhere between the outer edges of the subjects feet. This distance depends upon the subject, however we can still determine this range considering the number of sensors and their size. The data mining models were tested on the data that was generated on a Smart Carpet that has a sensor size of 15 cm^2. While the subject is walking and if the foot lands on a sensoric area which covers 98 % of this carpet, at least one sensor gets triggered. Hence the trajectory can lie between +/- 7.5 cm from the midpoint of this sensor. One centimeter is scaled to 3.3 pixels by using the video mapping technique. The estimated trajectory can have a tolerance of 24.75 pixels. If the size of the sensor is decreased, the tolerance range will reduce.

This tolerance range determines the acceptable level of accuracy that might be required for a particular application. When the results of the trajectories are within this range of the acceptable tolerance, the data mining model can be accredited to the particular tolerance criteria. When this condition can be met under all possible use cases, the model under test can be declared as validated under these conditions. A walk would consist of a variety of smooth curves, sharp turns and quick twists with straight, diagonal or spiral trails. We tried to collect some possible combinations of these configurations as patterns. For example a pattern with a straight walk, a quick twist and a diagonal walk ending with a twist on the opposite side can be said as pattern "N" (fig. 5.1). The subject would have to make this trail on the smart carpet. His starting and finishing line is usually outside the carpet boundaries. The main pattern is trailed inside the carpet boundaries.

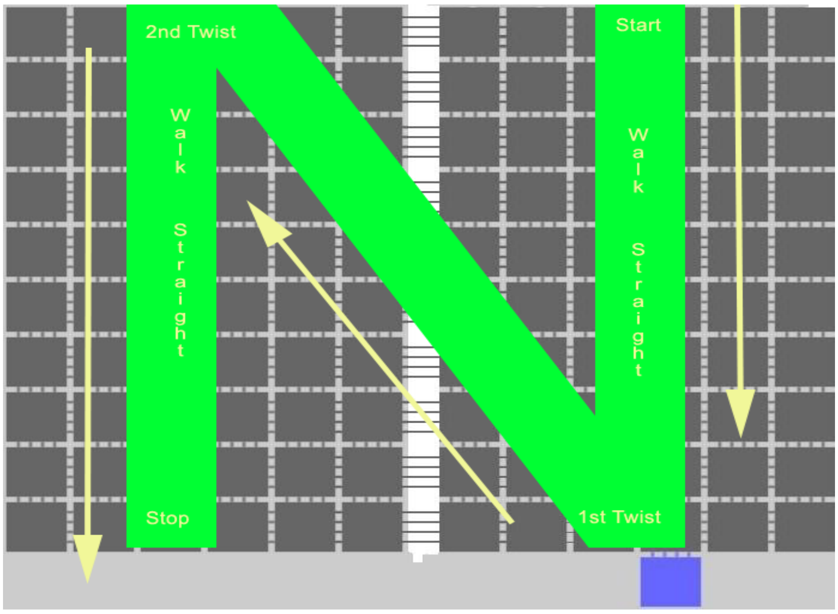
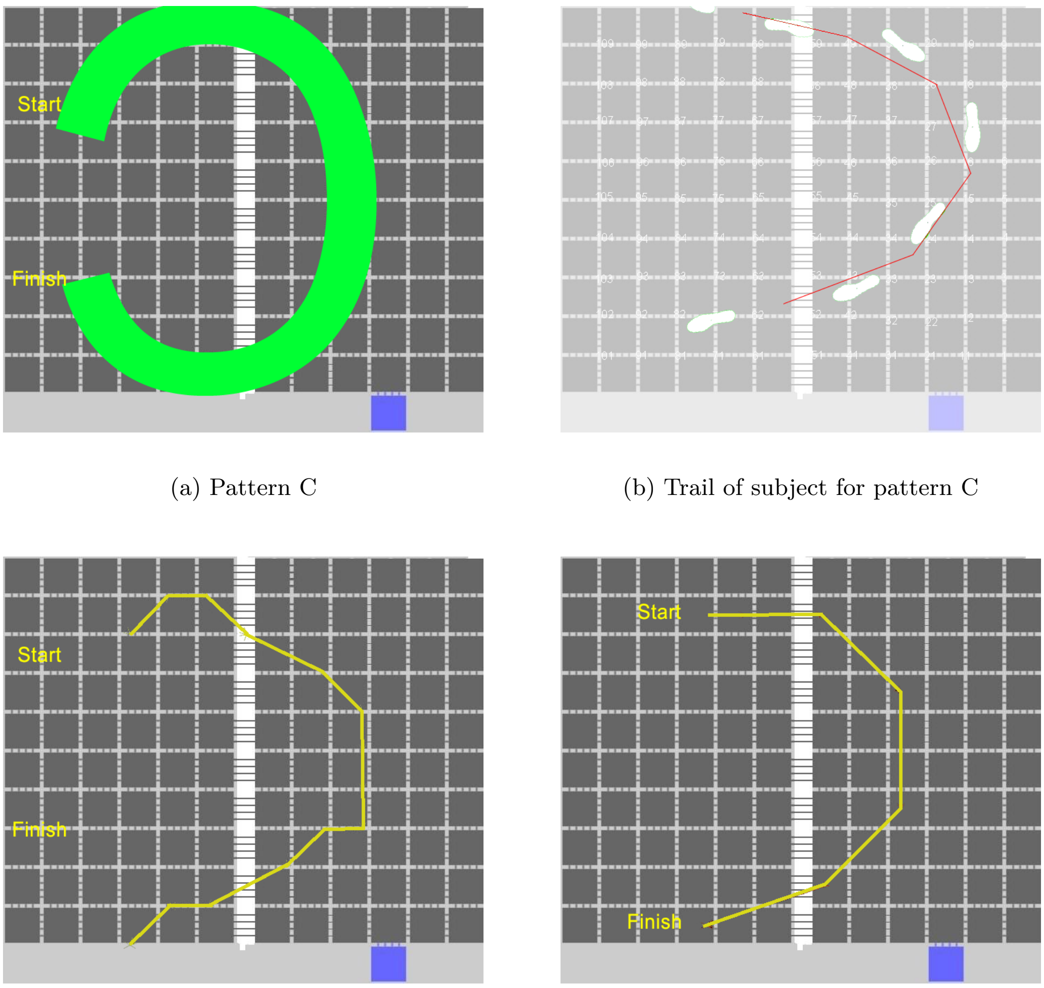


Figure 5.1.: Pattern N on the Carpet

Similar patterns with different set of curves and straight lines were defined. Details of each pattern can be found in the appendix A section. The subjects were requested to trail these patterns on the carpet. The experiments were carried with 16 adults (10 men and 6 women) with different shoe sizes (35 cm to 44 cm), heights (tallest person was 182 cm) and weights. All the walking was performed on the sensor floor, starting from outside the carpet meanwhile the main pattern was trailed on the sensors.

5.2. Actual vs. Estimated Tracks

Before the subject is walking on the carpet, a list of patterns are given and the subject is requested to study the shape of the pattern and its location on the carpet. The starting and the finishing edges are discussed. The camera positions are then adjusted and the software routines are initialized. The subject then initiates the starting signal by a clap. A few seconds before this sound, the cameras are starting to record, they capture the clap sound and point to the feet of the subject. The pattern the subject walks might be a smooth curve. As the subject traces the trail on the carpet, the sensors being in large regular squares, do not guarantee the smoothness of the curve. But they explicitly show the nature of the curve with sharp edges.



(a) Pattern C

(b) Trail of subject for pattern C

(c) Estimation of pattern C by method 1

(d) Estimation of pattern C by method 3

Figure 5.2.: Pattern C, trail, and estimations

The trail of the subject in pattern C has the same features as that of the requested pattern.

These features of a smooth curve can be observed also in the results of the data mining models 1 and 3, except for the fact that the estimations have sharp edges along the curves. This is due to the size of the sensor. When the sensor size is smaller, the curve would become more fine and the edges would turn smoother. Optimization algorithms can also be used to smoothen the curve at the cost of introducing new errors.

5.2.1. Location Accuracy

The size of the existing sensor is 15 cm by 15 cm. Between the sensors are strips of non sensoric area which lie at 5 cm in width on the perimeter. When the subject places the feet in any part where underneath there is a sensor area, the corresponding sensor or the midpoint of its cluster is considered to be lying in the trajectory of the subject's trail. The coordinates of the trajectory are calculated between the midpoints of two clusters or sensors if the cluster member is only one. Since the size of the sensor is 15 by 15 cm^2, it can be activated by pressure or touch at any point on its area. This can be any edge of the sensor, too. On the other hand, the foot that triggers this sensor can have the point of contact made at any of its edges, too. On the contrary, the whole sensor area is always considered with a single coordinate in the data mining algorithms.

The estimated tracks (Fig. 5.4 on the following page) and the original trail (Fig. 5.3 on the next page) are compared . It can be observed that the trail is plotted by locating the midpoints of the feet. The midpoints are calculated by scaling the image to a fixed size that corresponds to the size of the carpet and the sensors. The midpoint between two feet would approximately determine the center of the subject's body. The same is attempted by the data mining models. The activated sensors are clustered to one foot and the midpoints of the clusters are considered. The midpoint between two clusters determines the center of the subject where the trajectory lies.

While calculating the center of the cluster, the coordinates of the sensors are taken. These coordinates are the center of the sensor and the foot of the subject might have been located or landed on the center of the sensor or at any one of its edges, which cannot be determined by the sensors currently used. By considering the coordinates of the sensor, the whole area of 15 by 15 cm^2 has a single coordinate which introduces quite a considerable location error compared to the exact pixel location of a foot. But this error is not propagating, it is rather localized to the particular sensor. Hence the estimated trajectory and the actual trail would cross over at certain areas. This effect can be observed at the points B and C where the tracks are displaced due to the error introduced by the size of the sensor and its coordinate, compared to that of the trail at point A. The cross overs can be observed at point D.

Since the location of the foot on a sensor cannot be predicted with the currently used sensor, it is quite difficult to improve the accuracy of the trajectory recording. To increase the accuracy, the resolution of the carpet needs to be increased. When the size of the sensors is reduced and each sensor is given a coordinate, the resolution of the carpet increases. This would give a more accurate estimate which can lie close to the center of the foot.

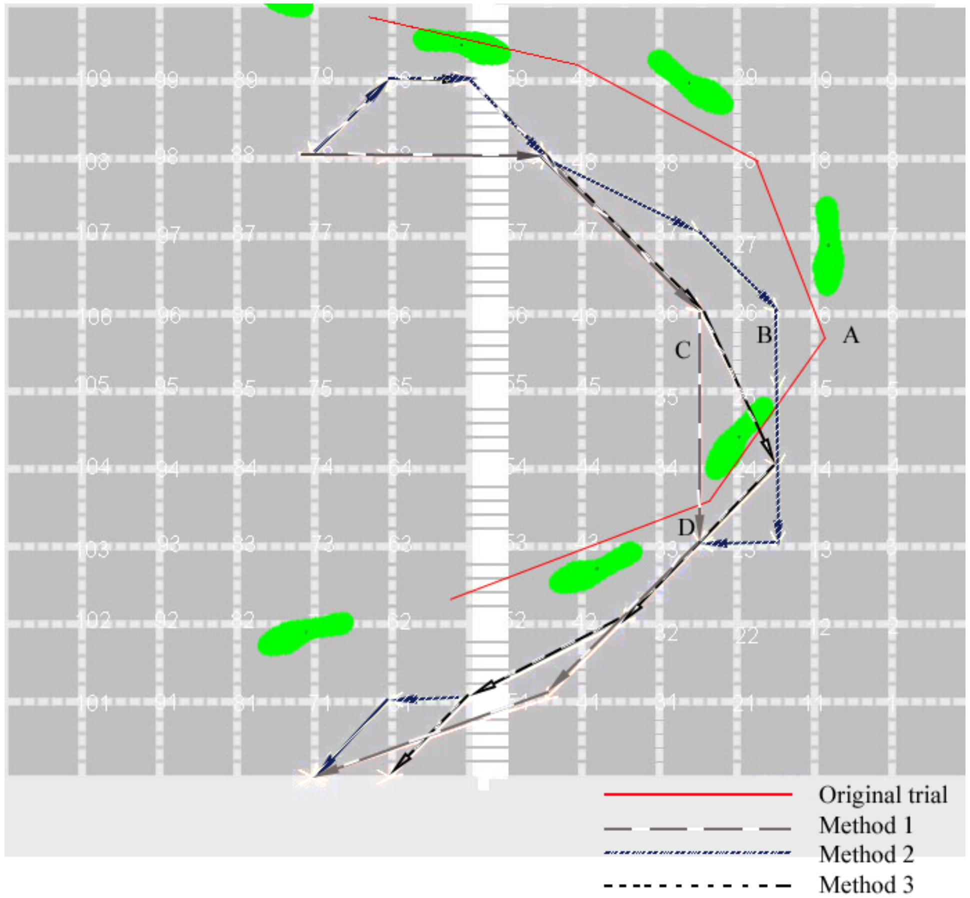


Figure 5.3.: Comparision of estimates with original trail

Figure 5.4.: Model 1 estimate

Figure 5.5.: Model 2 estimate

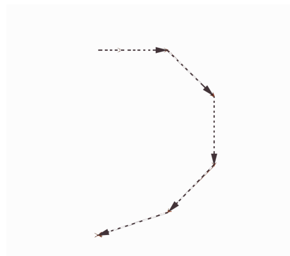


Figure 5.6.: Model 3 estimate

The displacement of the estimated tracks from the original trail occurs in two directions, transversal or longitudinal depending upon the direction the subject took on a trail. In order to estimate this error, a cumulative combination of transversal and longitudinal mean error (**E**) would determine the average displacement on the estimated tracks. Considering the transversal direction as Y and the longitudinal direction as X, the eqn. 5.1 can be used to determine the mean error distance between the estimated and the original tracks. The X coordinates of the original tracks are considered as a vector of m entities and the corresponding vector with n entities is considered to be that of the estimated tracks.

$$E = \frac{m+n}{2mn} (\sum_{m=0}^{m-1} \sum_{n=0}^{n-1} (X_m + X_n)^2 + (Y_m + Y_n)^2) \tag{5.1}$$

The video mapping technique suits best to determine these two quantities effectively. Since the whole picture is segmented into grids, certain segments of the tracks can be chosen, if not the whole track. The deviation is calculated in *pixels*, and later converted to *centimeter*. They describe the spatial deviation. This deviation determines the accuracy of the data mining models, given the condition that the sensor size is $15\,\text{cm}^2$.

5.2.2. Timing Deviation

Along with the spatial information, the time deviation has to be determined on the location of the tracks. It is important to understand the timing errors, too, which become quite critical when there is more than one person walking on the carpet. As the subjects walk on the carpet, the activated sensors get recorded on the log file with a time stamp generated from the PC. While the sensor data travels thru the network, it has a 2.5 ms delay at each hop as it passes thru before reaching the PC. Considering a small network of 120 nodes, the time delay of 10 ms is comparatively negligible. This time stamp recorded in the log file is used by the data mining model to predict the time instance when the subject was at a particular location. Since this time instance is having a delay which depends upon the random status of the PC at that instance and the activity on the carpet, it is rewarding to analyze the conditions surrounding an activated sensor.

When one particular sensor is getting activated and there are no other sensors getting activated around it and there is no application running on the PC, then any allowable time delay would not be a noise to the data mining models.

When there is a lot of activity on the carpet, there is considerable amount of time delay in the network because the current implementation uses only one PC and this becomes a bottleneck. It is worth considering the data paths for a group of sensors which are affected by the time delay.

In the later case, a group of sensors might have either one or two data paths towards the PC. The time delay is uniformly affected on all the sensors in this group. If the group is associated with another cluster or a group, the time delay becomes an issue that has to be handled carefully. The alternatives are that two clusters are related to each other or that the two clusters have to be separated. Such a condition can occur when there is more than one person walking on the carpet. Imagine a subject A is crossing a point P1 and subject B is just one foot away from this point. If there is a delay in the network, all the sensors that are getting activated in the next few foot steps might be affected. Now the problem lies in associating the clusters either to subject A or to subject B. This problem can be handled easily by considering the history of the trail and the cadence of the subject.

In this thesis, only a single subject is considered in all the patterns of walking. Hence network delay does not play a major role as noise or distortion of the observed data. When a sensor is activated, the instance of the sensor data packet, generated at the microcontroller can be said as $t1$. The time delay in the network due to hop count and the PC latency can be said as $\Delta t1$. Now $t1 + \Delta t1$ lies somewhere between points $t1$ and $b1$ in figure 5.7, which is recorded in the log file. In the video mapping technique, the time stamp is recorded when the foot of the subject lies parallel to the ground. At this instance, the body weight of the subject gets transfered. This can be anywhere between the points $b1$ and $b2$. Considering the fact that an average speed of an elderly subject is 125.27 cm/second and 151 cm/second for a young subject, the average stride length is about 76 cm. If the distance between two points is 151 cm, a subject would have, say lifted his right leg, placed it on the floor, moved his left leg a distance or 76 cm and lifted his right leg towards the destination and while the entire travel was about to be 151 cm, one second would have elapsed. The time interval between a heal strike ($t1$) to toe-lift ($t2$) is approximately 750 ms at normal walking speed [KPN96].

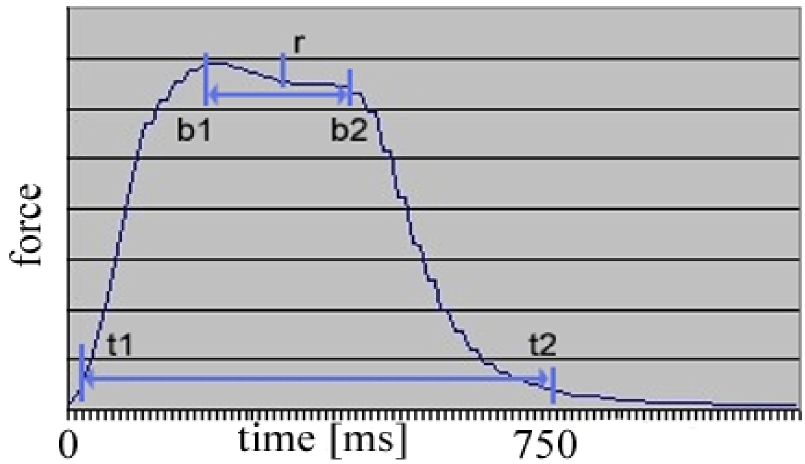


Figure 5.7.: Time profile of a foot stamp

The sensor activation would have been recorded between $t1$ and $b1$, meanwhile the time reference taken from the video mapping technique is between the points $b1$ and $b2$. If the sensor has sent the data packet with a 2.5 ms time delay with a worst case senario of 10 ms as hop count and PC latency, the data packet would be recorded at $t1$ + 12.5 ms. If the video mapping technique observed the foot stamp at the last possible time instance which is two thirds of the foot stamp which is 480 ms and if $t1$ was zero, the maximum allowable time error can be considered as 467.5 ms. If the time difference between an estimated coordinate and the corresponding pixel coordinate in the original trail exceeds this limit, the timing accuracy of the predicting model can be criticized.

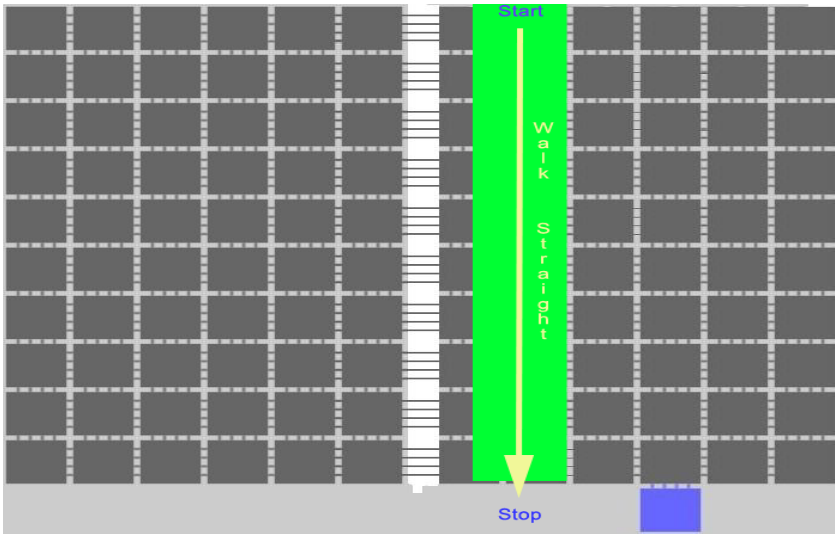


Figure 5.8.: Pattern I on the Carpet

5.3. Results

The walking patterns can be split in three different categories for single subjects walking on the carpet. One walking in straight line, another making a smooth curve, and the third case when subjects are walking in zig-zag patterns. The curves can have some varieties like short curves and combination of curves and straight lines. Zig-zag patterns can be combined with sharp turns and straight walks and combination of curves. More features and complex trails can be built, but answering the question of relevance, it is important to consider the trails, subjects take frequently in their daily walks.

Walking on straight lines is the easiest pattern to process. In this case, the subject is made to walk on a straight line on the sensors from point A to finish at point B. These are arbitrary points lying outside the carpet. Figure 5.8 shows the pattern "I" which is illustrated for a straight line walk. Whenever there was a sensor activated by the subject's feet, the log file recorded these foot stamps and the data mining algorithms plotted the trajectory. At the same instance, the video mapping technique was employed to capture the events independently and process the data. The spatial difference between the estimated and the original trails of the subject are plotted in figure 5.9 on page 102 to 5.12 on page 103. The Euclidean distance is calculated using eqn. 5.1.

The actual size of the physical carpet and the virtual image are related in their dimensions by a 1:3 ratio. This means, one centimeter in the physical carpet is equivalent to three pixel values either in the X or the Y directions on the represented image. This ratio was kept constant in representing the carpet, the foot of a subject and also the trajectory the subject took while walking. The difference between the estimated and the actual trajectories are calculated in pixels because the trajectories are represented on the X and Y coordinates of the image in pixels. By obtaining the difference between these spatial coordinates and plotting them in a spreadsheet, the difference between the trajectories results only in pixel values. This difference is converted into centimeters using the 1:3 ratio.

In the first instance, the reference image was taken and using the free hand selection tool a section of the track is selected. Then a java plug-in was developed to extract this selected segment of the track. The extracted pixel information is then exported into a spreadsheet automatically. This process is repeated for both the reference image and the estimated image, until all the features of the track are segmented. This reduces errors and speeds up the segmentation process. Both the estimated and the actual images are mapped to grids of a common size. The spatial coordinates for each grid lying along the trajectory are identified. The coordinates of the points where the trajectory intersects the grid is considered as segmentation points. When the subject was walking for a longer time the number of segmentation points from the images increase, too. The difference between the X coordinates of the reference and that of the estimated trajectories of the trail were treated separately. The procedure was repeated for the Y coordinates. In the end, the Euclidean distance is then estimated between these spatial coordinates and plotted on a graph.

The error difference in centimeter is plotted for models 1 and 2 which implement the mean and MLE estimates (Fig. 5.9 on the following page and 5.10 on the next page). It can be seen at point B in figure 5.9 on the following page and 5.10 on the next page, the estimate of the deviation tend to stabilize when the number of segments increase. They tend to get more accurate or rather the error difference is less biased as the number of segments are increased. If the segment points are less, they tend to be heavily biased or the deviation from the original values is quite large compared to that of model 3 (point A in Fig. 5.9 on the following page and 5.10 on the next page). This condition truly satisfies the nature of these models [NIS06]. In model 3, as the number of segment points increases, accuracy decreases (Fig. 5.11 on page 103). This is because the model clusters the known members in a propagating fashion. If an entity is grouped to a wrong cluster, the consecutive member might loose one member or add a wrong member while clustering. This effects the consistency of the model. This effect can be seen at point A in figure 5.12 on page 103.

The average performance of all the models for pattern "I" is plotted in Fig. 5.12 on page 103. This plot indicates that the models 1 and 2 behave stable when the number of segment points are increased. When the subject is walking for long distance in a large carpet, it would make sense to use either model 1 or 2 to estimate the trajectory of the subject. But the error deviation of model 3 is much less compared to that of the model 1 and 2. Since the size of the sensor is 15 cm in length and breath, the estimated error rate would have a tolerance of +/- 7.5 cm.

The accuracy of model 1 has a mean standard error of 6.8 cm while that of model 2 was 4.9 cm. Model 3 performed well with 1.86 cm. But the number of clusters obtained and the deviation when number of segment points increased, did increase the error estimate for this walking pattern. However, in a practical application, all these models can be employed and the compared for a particular use case. The optimized result can then be used by the application. The results presented are however still better than the results obtained by any of the other state-of-the-art technologies available at the time of writing this thesis.

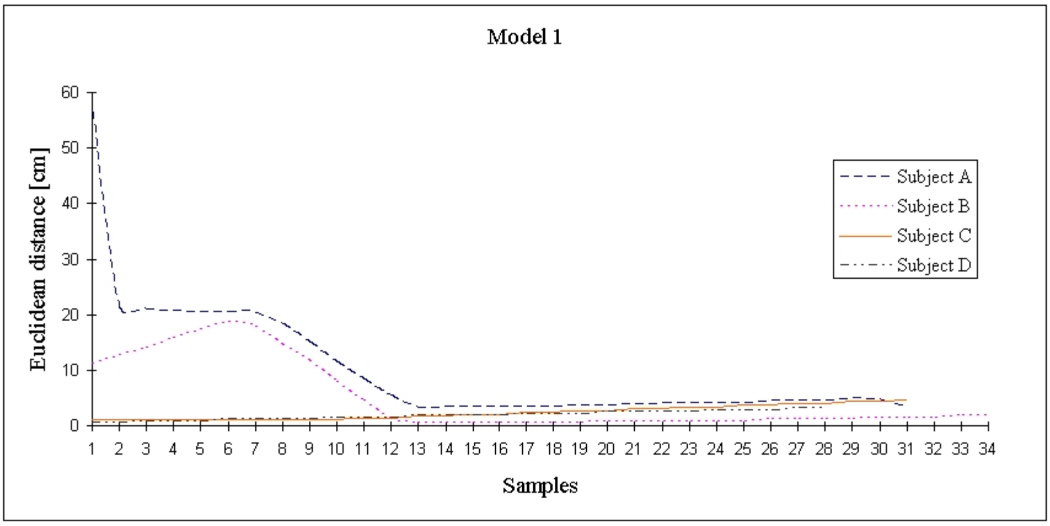


Figure 5.9.: Error difference using model 1 for pattern "I"

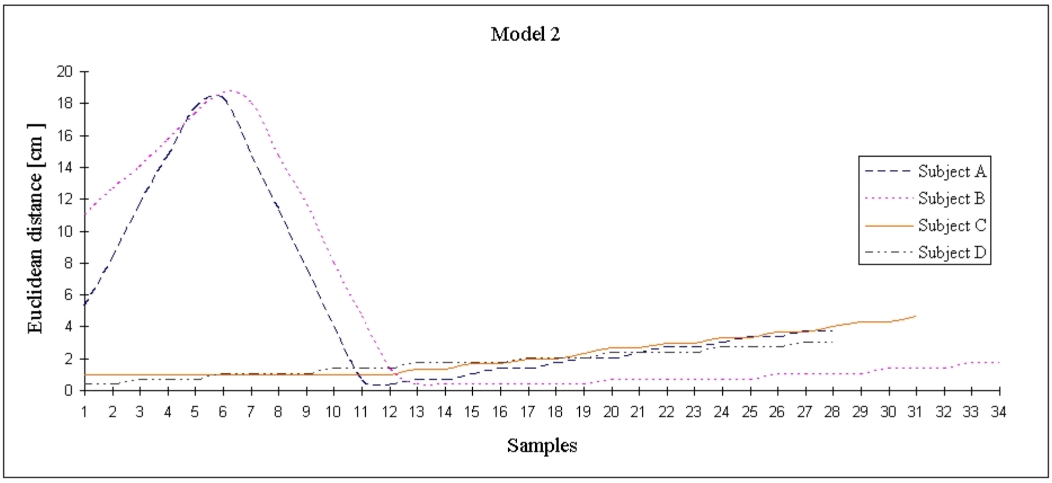


Figure 5.10.: Error difference using model 2 for pattern "I"

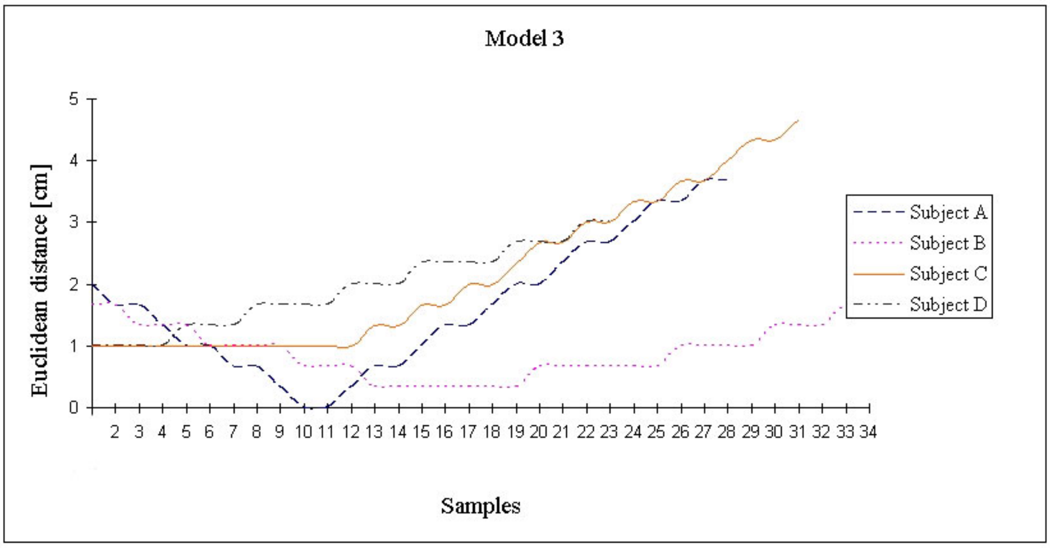


Figure 5.11.: Error difference using model 3 for pattern "I"

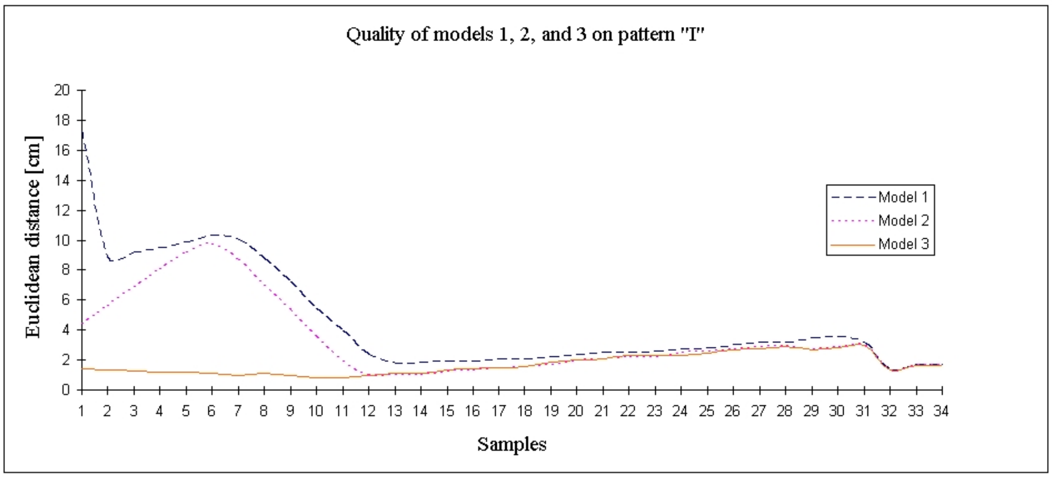


Figure 5.12.: Analysis of models 1, 2, and 3 for pattern "I"

Normal walking also involves walking in curves. A walker does make smooth curves walking inside a room, compared to walking on the corridors where sharp turns are made. It is interesting to observe the performance of the algorithms when the subject is trailing a smooth curve. Figure 5.2(a) on page 95 shows the pattern "C" which is illustrated for a walk involving a smooth curve on its trail. The subject had the starting point outside the carpet, the main curve was trailed on the sensor area and the walk was terminated outside the carpet. It was observed that the shape of the trail is maintained by all the estimates of the methods for different subjects. The estimated trajectory did correspond with the trail of the subject at curves and straight lines. The distance between the original trail and the estimated trajectory differed, based on the methods used.

Interesting features were observed when the subjects where walking in curves. At the start of the trajectories the difference is quite small when compared to the differences near the finish line or in the middle of the curve. The difference is quite high during the end of the curve. This feature has been observed in all the experiments. A steep peak is seen in the difference just as the subject finishes the curve and tends to finish the walk.

The results of the particular pattern were observed as the ones with large errors compared to the results of all other patterns. Results of models 1 and 2 can be seen with similarities (fig. 5.13 on the following page and 5.14 on the next page). This is due to the nature of the models previously discussed. However, model 3 was significantly unstable in this pattern. Every use case had a very nonlinear profile and similarities between different subjects were not found (fig. 5.15 on page 106). It was observed that model 1 had an average mean error of 25.5 cm meanwhile model 2 was 27.63 cm. Model 3 was observed with a mean error of 43.11 cm which was the largest average distance compared to other models and use cases. Although method 3 had a large difference in this pattern, the shape of the pattern in the trajectory was well maintained.

Considering the coding complexity, model 3 does not involve any floating point manipulations. A truth table reflects the working of this algorithm. It reduces computing and coding complexities. The algorithms can be directly embedded in the microcontroller which works as a sensor node itself. From the observation of subjects walking pattern "C", during the curves, the center point of the subject often falls outside the vertex of the subjects legs. This results is a considerable error difference seen in the valley A on figure 5.13 on the next page and 5.14 on the following page. The subject tends to be slower during the curves compared to the strides in the straight line trails. Another parameter that affects the nature of the walk is the boundary line conditions. As the subject tends to make turns, the error difference is large at the curves compared to the areas where the subject had made straight lines (point B in (fig 5.13 on the next page and 5.14 on the following page). However, it can be observed that as the subject did converge to a line from the curve, the error difference had also dropped towards a low value in most of the cases in models 1 and 2 (part of the plot after point B).

In figure 5.16 on page 106, it is observed that the models 1 and 2 at an average, maintained a consistent error difference throughout the trajectory of the subject. It can be also seen that these two models were able to follow the trajectory of the subject. Although the models did have an error rate of 25.5 cm and 27,63 cm respectively, the distance of 27 cm is approximately the size of an average foot of the subject. If the size of the sensors are reduced, this error difference would definitely reduce. Comparing the performance of the models with pattern "I", it is observed that the models are biased by the nature of the patterns (fig 5.12 on the preceding page and 5.16 on page 106). In the pattern "I", the difference is largely due to the properties of the models in analyzing small segment entities. This together with the nature of the pattern "C" where there is a semi continuous curve, adds to the effect of maintaining the error difference. Since there are a lot of parameters that have to be considered when comparing the original trail of the subject with respect to the estimated trajectory, the acceptance level of the algorithm heavily depends on the demanding application. However, the important fact to consider here is that such walking patterns were not discussed or presented so far in any other state-of-the-art technologies and such patterns must be emphasized and explored in further research.

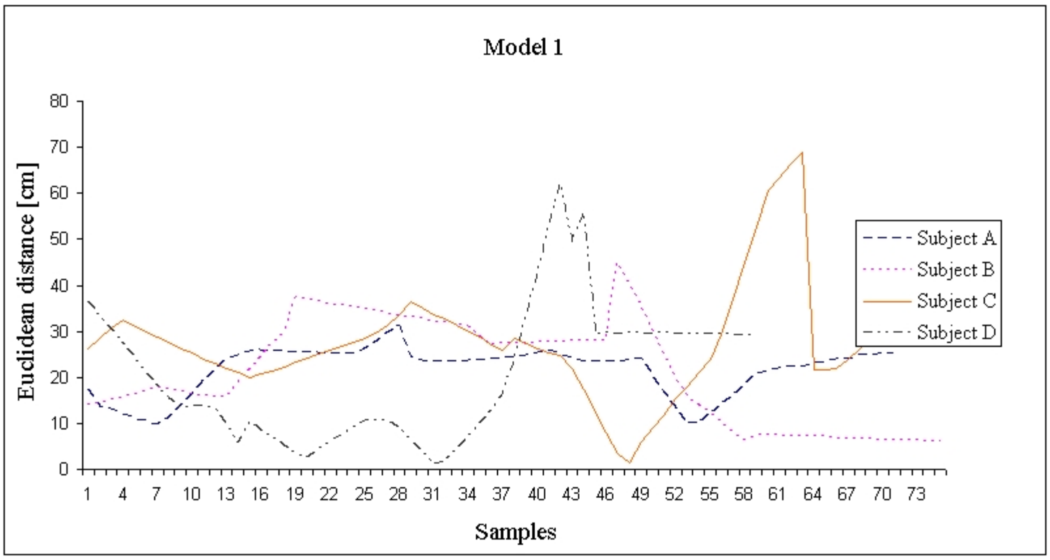


Figure 5.13.: Error difference using model 1 for pattern "C"

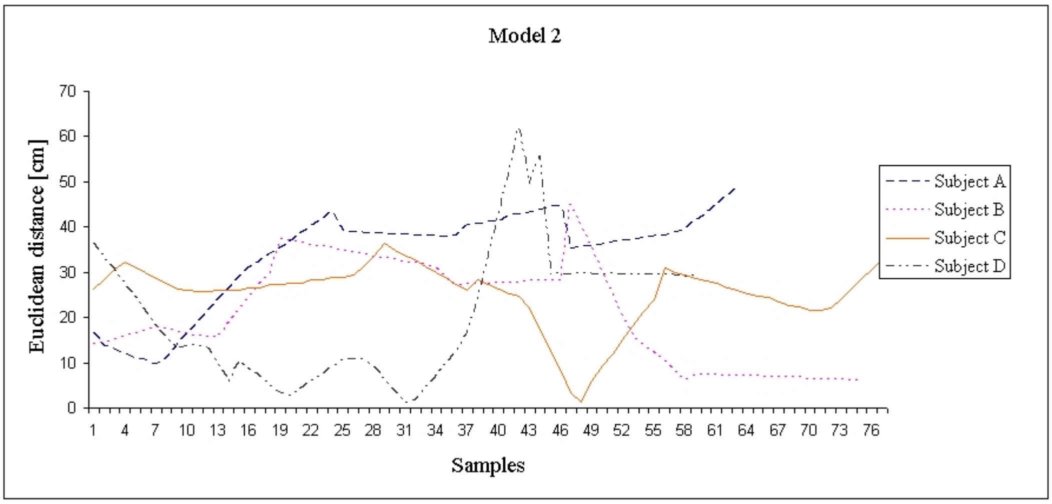


Figure 5.14.: Error difference using model 2 for pattern "C"

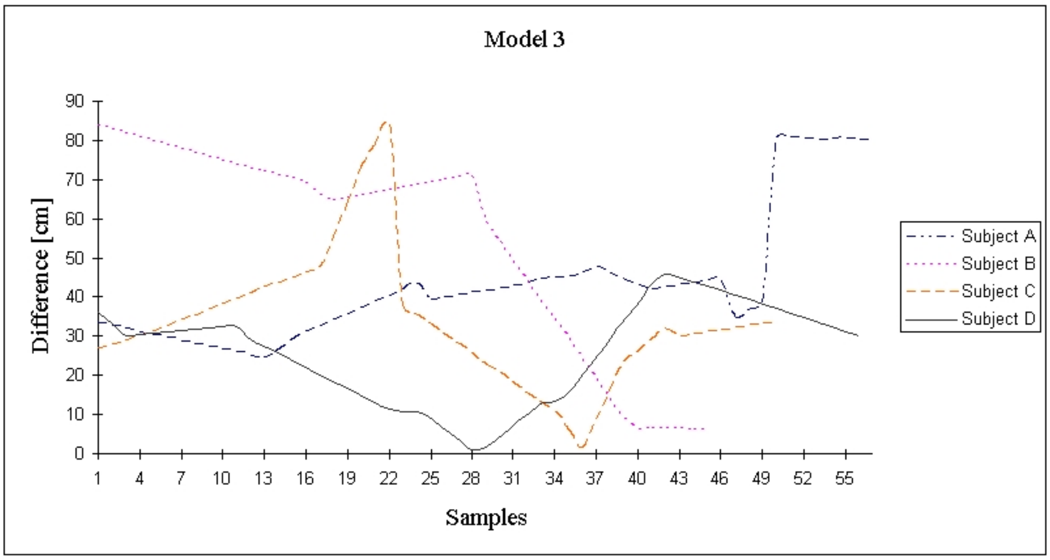


Figure 5.15.: Error difference using model 3 for pattern "C"

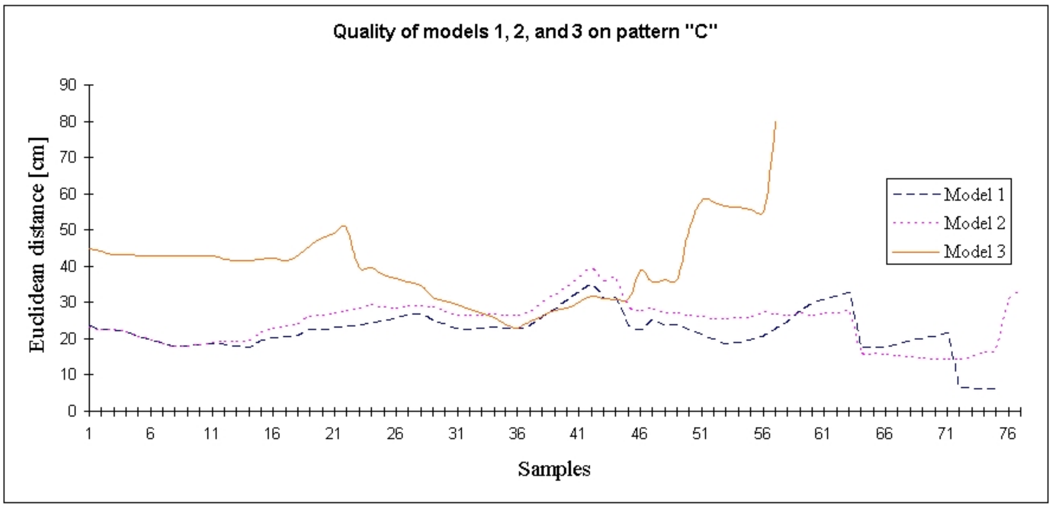


Figure 5.16.: Analysis of models 1, 2, and 3 for pattern "C"

Apart from curved structures, corridors or pavements consist of squared or rectangular blocks of building space. It is worthwhile to observe the performance of the data mining algorithms using patterns that involve sharp turns or zig-zag trails. Pattern "N" (fig 5.1 on page 94) can be considered as one such trail. Subjects were requested to trail pattern "N" on the carpet. The start and the finish points were outside the carpet. The carpet was also marked at the edges to indicate the subjects where to make the turn. The estimated trajectories are then plotted and compared to the actual trails. The results are presented from figure 5.17 on the next page to 5.20 on page 109).

The difference between the actual and the estimated trajectories are not that high compared to the results obtained for pattern "C". Since two sharp turns are involved here, there are two steep peaks observed (points B and D in fig. 5.17 on the next page and 5.18 on the following page) in model 1 and 2 as the subject makes the turns. The reasons why the number of errors does not increase when the number of twists increases compared to the slow curvey trail in pattern C can be easily explained. The subject makes a lot of steps during the curve. The subject curves slowly at each step. Although this does not introduce propagating errors, each step does shift the body center away from its middle point. On the contrary, a twist is made with a quick turn and the error is localized to perhaps one or none of the sensors. As a result, the errors introduced are local and they can be modulated by the inter cluster distances. This distance is very small at the clusters where the twists are formed because, the turn is quite fast in some instances, even less than a stance speed of the previous straight walk. This problem can be overcome by reducing the size of the sensors, because a smaller number of sensors offer more resolution and the contour of the foot can be traced in each cluster. As a result, an overlapping foot can be easily separated. Between the turns the subject exhibits straight line walking and the valleys A and C reflect these details in figure 5.17 on the next page and 5.18 on the following page.

Pattern "N" results show that the model 1 has an error distance of 20.9 cm while for model 2 it was 25.8 cm and model 3 resulted in an error distance of 25.3 cm. Observing the comparisons, estimates have large differences at the locations where the twists are made in the zig-zag pattern of walking (fig 5.20 on page 109). At the places where the subject made the twists, the error difference is quite high (point B and D in fig. 5.20 on page 109). In the rest of the areas, as the subject walks in a straight line, the error difference tends to reduce (points A, C and E in fig 5.20 on page 109). These errors are not large compared to the pattern where the subject walked a uniform curve. There were also other patterns which were analyzed and compared, for example when the subject was making a big circle (pattern "O"), the errors were even less, and the shape of the estimated trajectory had a error less than 26.7 cm for all the methods for all subjects. In all the patterns it was observed that the shape of the trajectory was always maintained and tracked in the estimated images. This proves that the data mining algorithm can detect the trajectory of the subject and follow the path the subject took while walking on the carpet. The accuracy of the estimated trajectory depends on the nature of the trail and the model used to estimate.

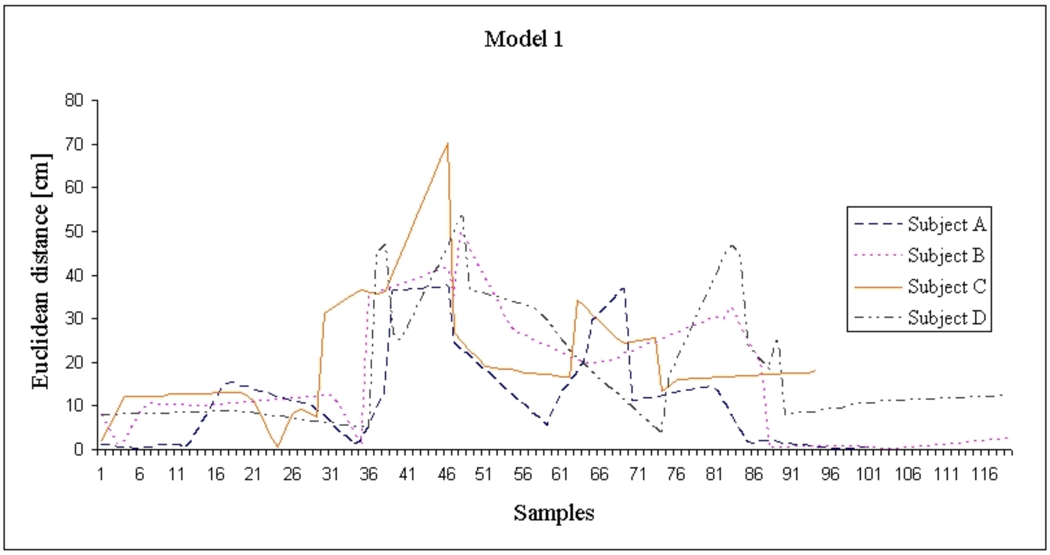


Figure 5.17.: Error difference using model 1 for pattern "N"

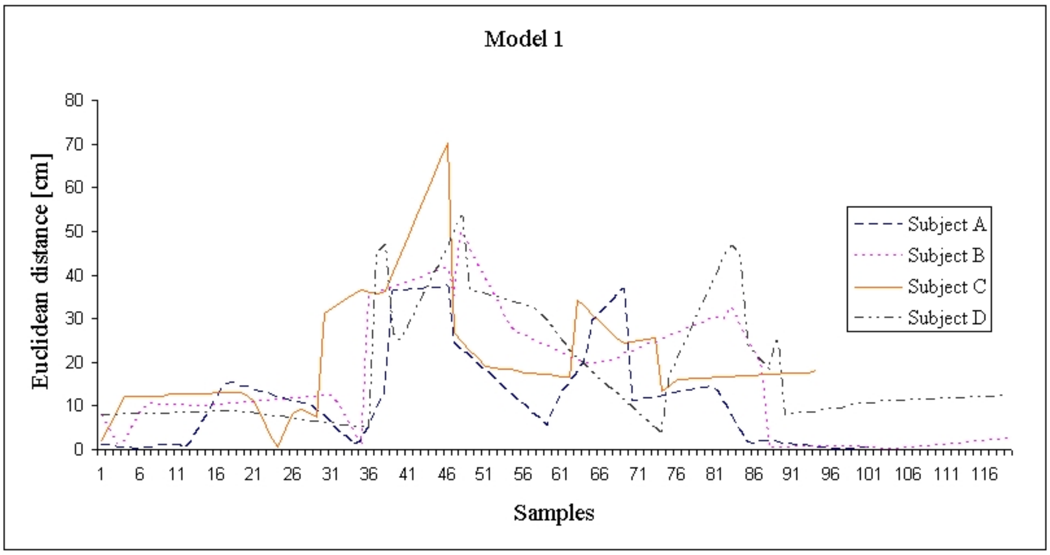


Figure 5.18.: Error difference using model 2 for pattern "N"

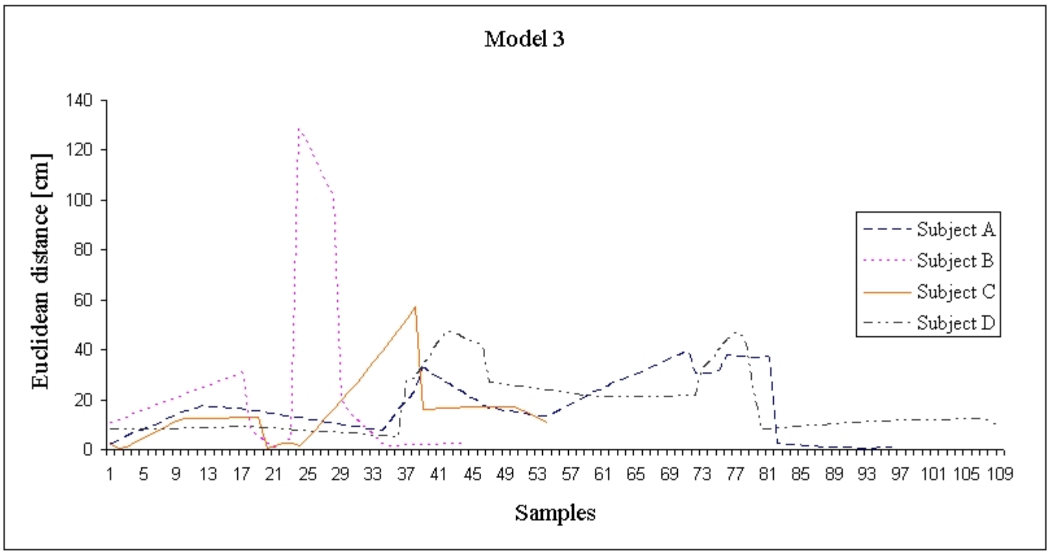


Figure 5.19.: Error difference using model 3 for pattern "N"

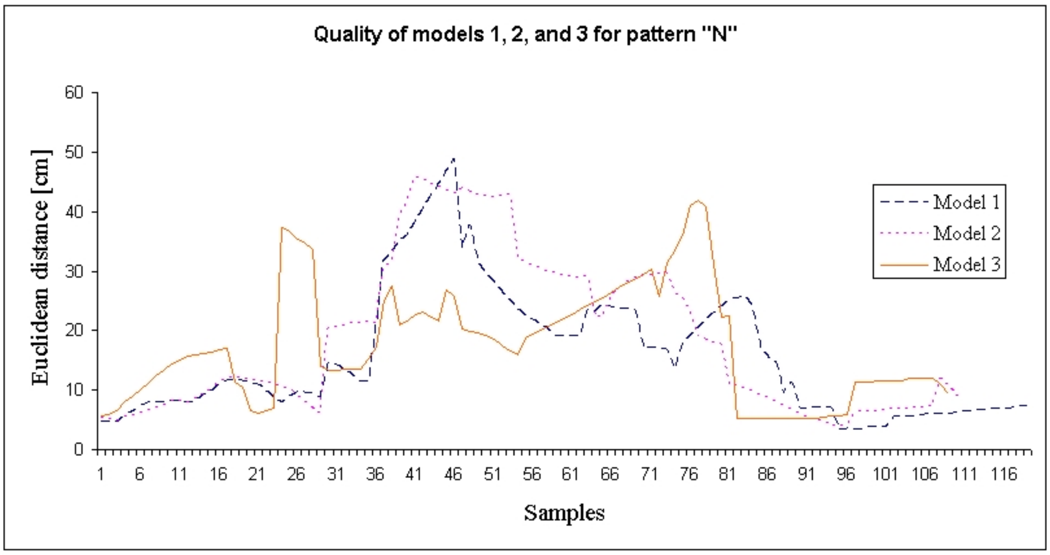


Figure 5.20.: Analysis of models 1, 2, and 3 for pattern "N"

A cumulative comparison of the average error difference is given in table 5.1 on the following page. It can be seen that, when the subject deviates from the straight line walking, errors are increased and are dependent on the nature of the trail. The trajectory of the subject causes errors due to the size of the sensors currently used. However, these errors are small when compared to the previous contributions of Human tracker and Geta Sandals [TMI04, KOC05]. The results are compared for the straight line experiment which was replicated in this set up as pattern "I". In this experiment, although the equipments and the methods used were different, the objective of identifying the trajectory of the subject when walking straight was the same and the results of the data mining algorithms used in the Smart Carpet performed better (tab. 5.2).

Table 5.1.: Mean Standard Error in cm of estimated coordinates for walking patterns

Walking pattern	Model 1	Model 2	Model 3
I	6.8	4.9	1.9
C	25.2	27.6	43.3
N	21.0	25.8	25.6
V	18.1	16.2	17.3
O	17.8	18.7	22.0
S	20.5	22.1	27.0

Table 5.2.: Comparison of state-of-the-art for straight line walking

Project	Mean error [cm]
Geta Sandal	158
Human Tracker	58
Smart Carpet	1.8 (model 3)

When the size of the sensors are reduced, more number of sensors would be triggered on a single foot step. This would bring a large difference between the inter-cluster and the intra-cluster time differences. The conditions of forming clusters can be still used in that case. Hence a more perfect center of the subject's body can be found. Since the trajectory of the subject passes thru the coordinates where the center of the subject is estimated, the accuracy of the trajectory can be further increased in this way.

The timing information is estimated by observing the time information on the foot stamp. This data is then used to estimate the location of the subject at a particular coordinate where the center of the subject's body is estimated. The timing deviation is then calculated between the estimated time stamp at a particular coordinate that lies on the trajectory and the actual time stamp observed from the video sequence, on the path that lies near to that coordinate. Comparing the three models, the results of the performance are presented here (tab 5.3 on the following page). It can be seen that, the mean error distance is nearly half the time for a complete foot transition of a single step. The time error is local due to the activation of the sensors, the error is not propagated as the subject moves from one point to another. Hence, this error is under a well acceptable level.

Table 5.3.: Timing deviation in milliseconds

Algorithm	Timing deviation in ms
Model 1	332.6
Model 2	410.4
Model 3	334.6

5.4. Summary

Before a subject walks on the carpet a specified pattern of his trail is detailed. As the subject walks on the carpet, the log files record the foot stamps of the trail. Simultaneously the video mapping technique briefed in the previous chapter is employed to track the trajectory of the trail and serve as a reference. Three data mining models presented in chapter 3 have been used to estimate the trajectory of the subject. The results of the estimated trajectory is compared with the reference. The timing deviation between the actual and the estimated time stamps are found to be under considerable error tolerance. The time stamp recorded in the log file cannot be considered ideal when the number of sensors increase. The time stamp recorded during referencing the real time situation, is considered to be ideal. The difference between the ideal and the estimated time stamp is less than 470 ms.

As the images of the actual and the estimated trajectories are available, the spatial coordinate of the track is extracted using ImageJ plug-ins. This data is exported to a spreadsheet. The Euclidean distance is then calculated between the actual and the estimated tracks. Since the data about the tracks is available in an image, the coordinates of the pixels which represent the tracks are extracted to the spreadsheet and then converted into centimeters. This data is then used to calculate the mean error distance between two corresponding points. The number of points on a trajectory or the segment points are then plotted on an chart to identify the quality of the models for different walking patterns.

It was found that the models 1 and 2 perform well in conditions when the subject is walking on a straight line. However, model 3 is able to track the subject more closely in following the trail. Results of subjects walking on curves and making sharp turns show that the models have large error differences exactly at the curves and turns, but still they are able to identify these features of the trail. Observations indicate that the implemented data mining methods, estimate the trajectories the subject took during the trail. The data mining models are able to follow the path of the subject. The accuracy of the algorithms depend on the nature of the trail. When there are curves, the mean error between the estimated and the actual trajectories increase exactly at this place. The results of the models contributed in this thesis are also compared with the state-of-the-art technologies for similar experiments. It was observed that the results of the models presented in this work are better in performance compared to the others. Although the data mining models can be further improved, the next chapter provides some more details on how the hardware can be improved. It provides some examples of how the models can be improved to identify and discriminate two subjects walking on the carpet simultaneously. The chapter gives an overview of the entire work and ends with possible developments and applications.

6. Conclusion and Future Work

The goal of the thesis was to investigate the information recorded by a self organizing sensor network embedded in a carpet and how to extract and represent the information in ways that facilitate tasks such as tracing a trajectory of a person's walk on the carpet. We studied the nature of the recorded data, its features and the possible parameters that can be extracted from it. Various methods of reading the patterns hidden in the data are explored. The trajectory of a subject's walk has been plotted with spatial and timing information with an accuracy of 99 % as the subject walked on a straight line.

6.1. Thesis Summary

Not only is the idea of integrating electronics and textiles interesting, rather, it creates new areas of research, for example, capturing the foot steps of a subject walking on a carpet. Pervasive means of computing information have generated interests in this area of observing walking. Various approaches have been seriously considered and some of the noted ones were discussed in chapter 1. The experiments carried out during recent years were aimed towards recognizing foot prints and trying to profile a subject. This resulted in costly equipment which cannot be embedded in each sensor. State-of-the-art developments led to Z-tiles where signal processing was done locally and a self organizing network improved location identity. However the experiments were not only aimed towards tracking or tracing a person's trajectory of motion on the floor. Geta Sandals and Human Tracker were the earliest systems that were focused towards this problem. The former cannot be integrated on a large scale. It loses accuracy when the subject walks on long distance. The latter is too expensive to be commercially deployed, neither does it provide timing information on any instance of the trajectory. Combining the former efforts and state-of-the-art materials to form a flexible cloth integrated with sensors and electronics, the Smart Carpet contains possible answers ubiquitously on an easily transportable commercially deployable platform, to identify individual footsteps of human walking. It consists of sensors embedded in a cloth woven with conductive fibers. A 16-bit microcontroller with additional electronics is embedded and interconnected to form a regular mesh topology in the carpet. Construction, functions, and detailed operation of this Smart Carpet is discussed in chapter 2.

The data that describes the organization of the network in the smart carpet is recorded in a log file. As a person walks on the carpet, the footsteps activate the underlying sensors. The activation and the deactivation of the sensors with their time stamps are also logged. The log file serves as the primary source of data to trace the trajectory the person took while moving on the carpet. The nature of this data that describes the walk, its pattern, and underlying information explain certain patterns of clusters that describe the foot prints of the walk. It was identified that footsteps activated a cluster of sensors and the midpoint of two clusters represent the arbitrary center of a subject. This laid the concepts of a specification that can be implemented in an algorithm that can extract and interpret the data that describes a subject's walk. Having outlined the specifications, Chapter 3 presents three algorithms that can extract the data and describe the spatial coordinates of the trajectory with timing information. All these algorithms provide a solution to form clusters from the data, calculate the centroids of the clusters , and estimate the midpoint between two clusters. The spatial coordinates of these midpoints represent the coordinates of the trajectory. Timing information is calculated with respect to these midpoints. Each algorithm follows a unique method to identify these clusters. The first method determines clusters based on the arithmetic mean of the time difference between two sensor activations. The second method determined the clusters based on maximum likelihood parameter estimation. The third algorithm followed a regression analysis technique where three consecutive sensor entities are compared simultaneously and ranked. The ranks are based on the merit of the cluster near its border. A truth table that describes the possible members and non-members is applied to the final rankings. The estimated coordinates of the trajectory are stored in database tables and are also drawn on an image scaled to the physical size of the carpet.

The result of the algorithms was to be validated. An independent source that can be considered as a unique and accurate representation of the subject's walk has to be identified. Three different techniques for evaluating the models are explored in chapter 4. In the video analysis technique, the video frames are compressed to form an image where the trajectory can be extracted by image processing techniques. In the video mapping technique three cameras at different angles are deployed to observe the subject's feet while walking. Grid like markers are drawn on the Smart Carpet which serve as reference lines. Images of the subject's feet, the carpet and the grid on carpet's surface are scaled to a uniform ratio. Each footstep of the subject is imposed manually on the carpet image by observing every frame of the video taken from the subject's walk. Timing information is logged separately. Simulation of the carpet and the subject's walk is considered in the third technique. Due to the dynamic nature and behavior of the subject's walk, simulation is quite out of scope from this thesis. Comparing all the three techniques, the video mapping technique is taken as the most accurate means of referencing a subject's walk. It is chosen as the unique reference source against which the estimation of each algorithm is compared for its accuracy. Chapter 5 describes what is measured, the relevance of the estimations, and the accuracy of the models. The subject is made to walk in different patterns that reflect the trails one would take in daily routine. The estimates of the data mining models are then observed for their performance under these patterns. The reference and the estimated trajectories are exported to a spreadsheet. The distance between the two are calculated and the Euclidean distance between the two trajectories is calculated. The error deviation of the estimated values is found to be better than the performance of Human Tracker and Geta Sandal experiments for the same test case.

6.2. Contributions of the Work

In reaching the objective of plotting the trajectory of the subjects walk, this thesis made a considerable contribution in identify footprints in sensor data, which are summarized here. The idea of tracking a subject's motion using a non vision approach pervasively is highly encouraged in this thesis. In this approach, the subject does not notice the presence of a sensor under his feet. The subject has the freedom to move at their own pace and direction which is not the case of other systems like Active Floor.

The major contributions of this thesis are the methods used to estimate the trajectory of the subjects walk on the carpet. These methods were explored and implemented using Java 2 software development kit. Eclipse was used as the integrated development environment. The mathematical part of the methods was implemented as algorithms and tested before they were functional in an application platform. A means of testing the algorithms were thoroughly specified, implemented and investigated. The results of the algorithms were compared to the performance of commercial implementations. It was found that the results were comparable and in certain cases, the difference was negligible.

Another important contribution of this is the validation process. An innovative and accurate model validation process is researched to evaluate the data mining models. The result of the models needs to be accurate and creditable depending on the application that uses the estimated trajectory. The real time observation of the walk is plotted on an image file. Utmost care is taken to reproduce the exact location of individual foot in each walk. A trajectory that represents the motion of the subject is then plotted based on the location of the foot. This trajectory is then segmented in grids. The estimated track is also embossed with a similar grid. Segments of the trajectory are extracted from both the images and exported to a spreadsheet. This process is atomized using plug-ins developed and integrated in the ImageJ Java image processing software.

The results are then compared and it was found that the data mining models are able to track the subjects walk. The models can follow the path of the subject if they walk straight, turn or twist.

6.3. Discussion and Further Improvements

The problem of data extraction and representation with respect to a light weight sensor network and a human walk has be extensively contemplated in this work, that resulted in tracking the trajectory of the subject's walk. Although it can be compared to other tracking problems, issues such as location and timing accuracy, representation with respect to the sensitivity of the observed data, noise, and signal attenuation should be dealt thoroughly. Most of the tracking systems are attributed with video or signal analysis where the available information is quite extensive. A large set of parameters can be extracted and few important ones can result in a most accurate estimation of the trajectory. Some systems have much less sensory information with low computing facilities. Mostly they are conveying the presence or absence of the target. The system presented in this thesis can cover a large detectable area where segments of the sensors can be trimmed to fit the ambiance. They are flexible enough to adapt to the changing space and attrition tear. The information conveyed by the sensors are binary in nature and extremely light weight. They confirm the presence or the absence of a foot. It is chimeric to modulate a system with algorithms that can exploit the nature of the available data and produce an accurate tracking system for pedestrians. However, the Smart Carpet along with the data mining algorithms presented in this thesis contribute a major step towards this goal.

6.3.1. Discussion

This thesis provides a glimpse of such a system that can be foreseen in the near future. The results obtained from such a system seems to be quite promising in this aspect. The accuracy of the system, from the trajectory point of view, has improved the performance when compared to other system available at the time of writing this thesis. The shape of the trail the subject took is accurately estimated. The data mining methods track and follow the subjects when they make a curve, twist or walk in a straight line at the ease of the walker, whereas systems like human tracker need a reference line that guides the subjects to a predetermined trail. The errors introduced in the system are confined to the location and size of the sensor and they are not propagating on each step like that of the Geta Sandals.

Application that would use the data that is estimated by the data mining models need to determine if the accuracy and the credibility of the available results does satisfy the requirements of its usage. Naturally, a perfect system is hard to fabricate. This thesis offers a step nearer to perfecting a system that can track a subject with a commercially implementable modular solution.

6.3.2. Data Representation

The shape of the smart carpet can be flexibly adapted to the available space. The location of the sensors are uniquely defined when the network in the carpet is booted. This feature of the self organizing network and the light weight sensor frame which conveys the presence and absence of the foot step makes the Smart Carpet a versatile platform to implement data mining algorithms to track people walking on top of it. As the subject walks on the carpet, a log file records the sensor activations along with time stamps which serve as the primary information for the trajectory estimation models. Parameter estimation, cluster formation and centroid calculation are done with respect to the nature of the available data from the network. The concepts used in these algorithms can be further extended or adapted to other domains of data extraction provided the estimation parameters are fairly comparable. Comparing the straight line experiment and the experiments which involve the subject to take curves in his walk, the algorithms do not have the same result. Although they perform far better than the best results of Human tracker [TMI04], improvements can be done by reducing the size of the sensors. It increases the resolution of the carpet. More than 3 sensors would be representing a foot print. This would lead to adding more nodes in a small area compared to the existing prototype where a sensor covers an area of 225 cm^2. Adding two sensors to fit the current area would double the number of nodes. Although 6 sensors would represent a foot press, traffic increases latency and loss of data. Further improvements can be done to remove traffic delay by still reducing the size of the sensor frame by bit stuffing and area blobbing. Improvements in the models can be done by experimenting with standard algorithms. However, considering the nature of the Smart Carpet and the available data, the improvements would be minimal and comparing the area of 2.32 m^2 a person occupies during walking, improving the estimation accuracy of more than 1.8 cm would have a negligible effect. Missing data, false sensor activations due to a sensor malfunction or amplified stretch of the subjects foot on the floor during a stride could be considered as noise. Kalman's filter can be used to eliminate noise. However, noise of this nature is random and does not depend on the function of the system. It is quite difficult to eliminate this noise. This thesis opens the ground for further research in this area.

This work solved the problem of estimating the trajectory of the subject by the known footprints. Alternatively knowing some of the footprints and its location or by giving the source and destination, a trajectory can be estimated. This data when conveyed to an external receiver can be used for a guidance system. It could find applications where people with disabilities can be easily guided in a huge shopping complex or an airport from a terminal to a terminal or in a railway station for a blind person to find a seat in the train compartment. Still the Smart Carpet as a whole can be advanced, when new materials and semiconductors are available. Introducing more than one host computer increases the speed of hop count. This would also reduce the bottleneck at the host computer.

6.3.3. Tracking Multiple Footprints

Considering further developments that can be made using the existing developed methods in this research, an immediate problem that can be foreseen, is to track multiple subjects walking at the same time on the carpet. To deal with this problem more insight has to be given on the nature of the walk two subjects would take. In the first phase, it is obvious that, the subjects either meet and cross over their trails or they are walking parallel or opposite and they never cross over their trails at all. The latter condition can be easily realized because, a subject must have a certain area around his feet or body which would directly reflect on his walk and the sensors that get triggered. Hence the radius of the activated sensors can be considered as an important criteria and the data need to be clustered based on this condition too.

In the former case where subjects would cross at a certain point on their trail, certain other aspects have to be considered to cluster the data and associate the clusters to a particular trail.

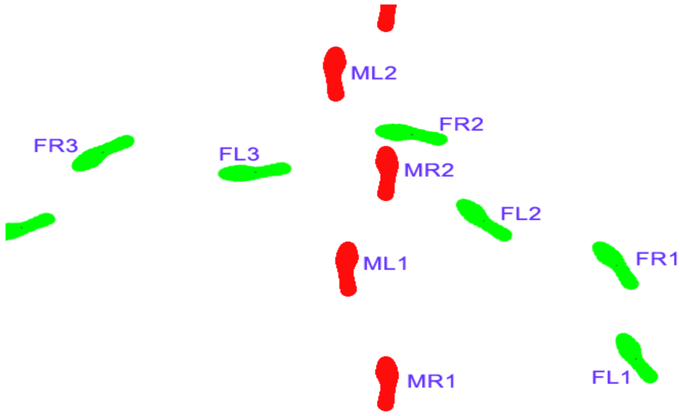


Figure 6.1.: Cross over trails of two subjects

When two subjects (F and M) are walking and their trails would seem to cross over (fig 6.1), their trails can be separated and traced by the following way. Considering the sequence of events that happen, it can be seen that subject F would keep the left foot at position FL2 and then the right foot at FR2. At this point, the status of subject M would be at position MR1, at MR2 or at ML2. If the subject M is at MR1, FR2 will be followed by the group of sensors that are being activated at location FL3 sequentially. This would be the same when the subject M is already at position ML2 when subject F is at position FR2.

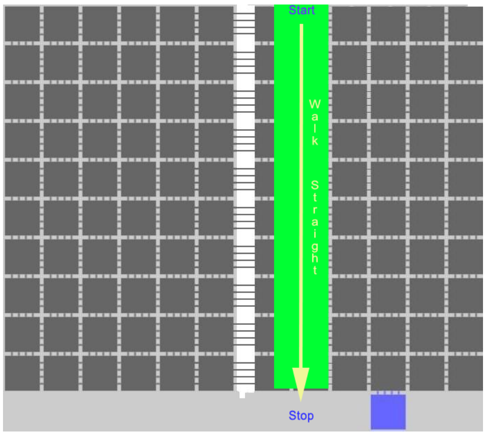
But, if the subject M is at position MR2 and subject F is at position FL2, either one of the subjects has to wait, or rather will wait in real life situations. Then, one of the subjects would cross over and the other would follow his intended trail. The time difference between two foot steps would be normal for one subject whereas for the other, they would have a waiting period. This criteria can be implemented to discover who waited and where was their next foot step. This never confuses the activation of sensors, provided the sensors are implemented as shown in the architecture of the Smart Carpet. To determine if the subject has taken the right trail, the cadence of the subject and the history of the foot steps can be related and compared.

6.4. Summary

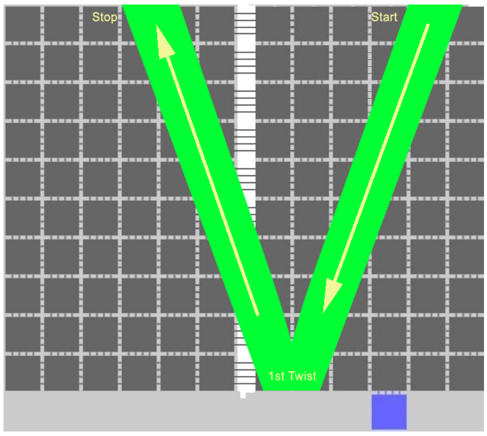
This thesis explored the possibilities of tracking a subject while walking on an area covered with the Smart Carpet. Each footstep triggers a set of sensors on the carpet and this information is logged in a log file. Data mining algorithms which describe the data in the log files, read and interpret the data and identify the trajectory of the subject. An independent method of observing the actual trail of the subject is important to validate the data mining models. This independent observation reflects the real time events of the walk and can serve as a reference. The data is extracted from the reference and the estimated trajectories and exported to spreadsheets that the data is available in an analytical form. The mean error between the estimated and the actual trajectories are found to have increased in performance. Possible applications that could use the data from the trajectory estimation methods can determine if the results of the validation presented here are under their acceptance level. In some cases, the data that is generated out of the data mining models have to be adapted or the application that uses the data needs to be modified to use the algorithms. Future problems like, tracking multiple trails, when more than one subject is walking on the carpet is also discussed.

A. Appendix I

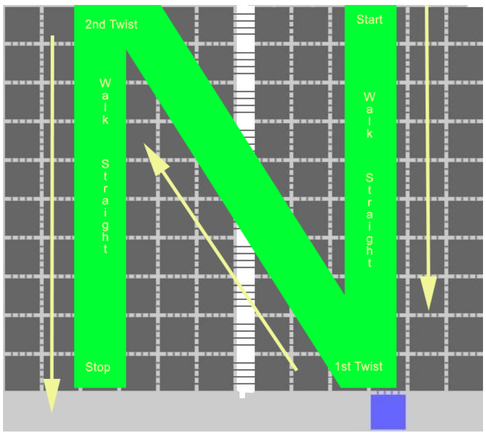
Subjects walking on the carpet follow common trials. These trials are a set of paths anyone would walk each day in their motion. To have a uniform set of trials for all subjects, the trials were described as routes. These routes were defined as a set of walking patterns. They are a combination of straight line walks with zig zags (part I), and curves (Part II).



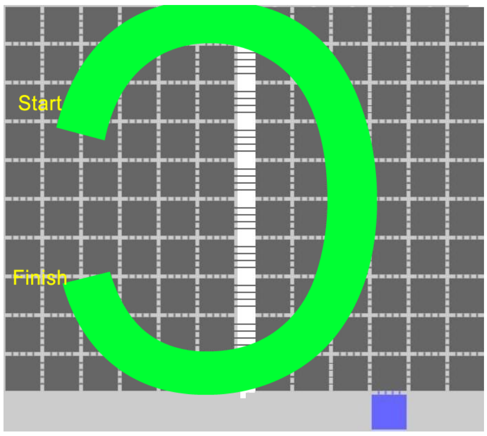
(a) Pattern I



(b) Pattern V

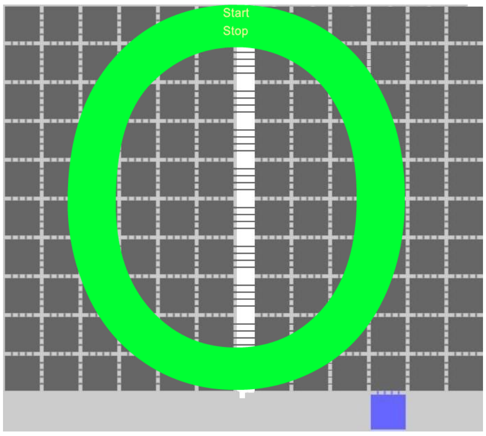


(c) Pattern N

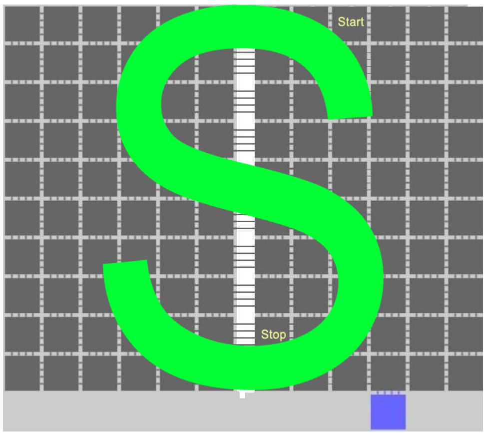


(d) Pattern C

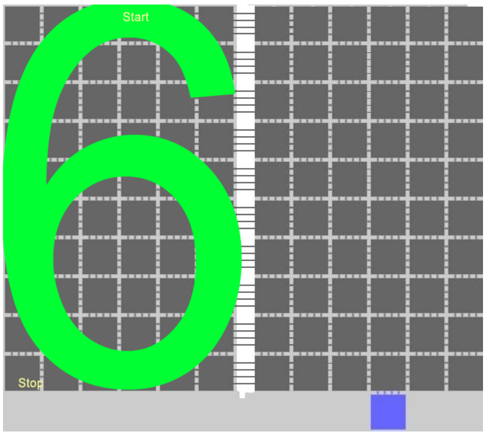
Figure A.1.: Walking patterns - Part I



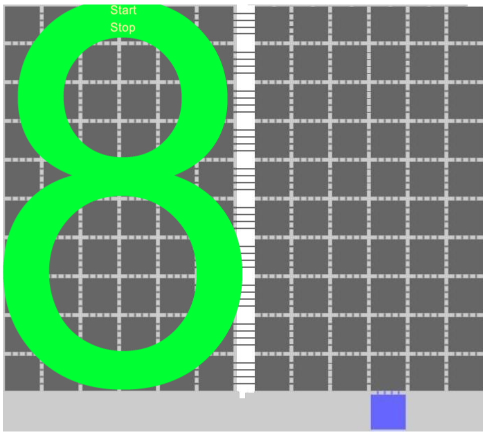
(a) Pattern O



(b) Pattern S



(c) Pattern 6



(d) Pattern 8

Figure A.2.: Walking patterns - Part II

The subjects were given other tasks while walking these patterns. In one instance they were asked to drag a trolley, in another instance they were requested to carry a weight while walking on the carpet. In another instance, the subject was requested to jog on the carpet for a pattern. differences were noticed when the subjects jogged on the carpet. The time difference was less between the sensors and the number of activated sensors were also very less because the size of the carpet itself was not big enough to capture enough data to mine it.

List of Figures

List of Tables

References

[BA02] V.J. Blue and J.L. Adler. Flow capacities from cellular automata modeling of proportional splits of pedestrians by direction. In *Pedestrian and Evacuation Dynamics*, page 115Ű122. Springer, 2002.

[BGF02] R. Brooks, C. Griffin, and D.S. Friedlander. Self-organized distributed sensor network entity tracking. Technical report, International Journal of High Performance Computing Applications, Special Issue on Sensor Networks, 2002.

[BM98] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. of CVPR*, 1998.

[Cho70] Ya Lun Chou. *Statistical Analysis*. Holt International, 1970.

[Daa04] W. Daamen. *Modelling Passenger Flows in Public Transport Facilities*. PhD thesis, University of Delft, 2004.

[DCC95] M.S. Nixon D. Cunado, J.M. Nash and J.N. Carter. Gait extraction and description by evidence gathering. In *Proc. of the International Conference on Audio and Video Based Biometric Person Authentication*, pages 43 – 48, 1995.

[DHS00] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[Dje05] C. Djerba. State of art in body tracking. In *In UMR 8022 CNRS USTL*, Jun. 2005.

[DK97] JM. Donelan and R. Kram. The effect of reduced gravity on the kinematics of human walking: a test of the dynamic similarity hypothesis for locomotion. In *Journal of Experimental Biology*, volume 200. Issue. 24, pages 3193–3201. Company of Biologists, 1997.

[Doe06] R. Doerfler. New chip encapsulation components. *Onboard Technology*, Feb. 2006.

[Ess00] I. A. Essa. Technologies towards the building of an aware home. *IEEE Personal Communications*, Oct. 2000.

[Fis20] R.A. Fisher. A mathematical examination of the methods of determining the accuracy of an observation by the mean error, and by the mean square error. Technical report, 1920. Notices of the Royal Astronomical Society 80 758 770.

[FSS02] G. Stromberg F.T. Sturm, S. Jung and A. Stöhr. A novel fault-tolerant architecture for self-organizing display and sensor arrays. Technical Report 33, SID Symposium Digest of Technical Papers, 2002.

[Fur71] J. Furin. Designing for pedestrians. http://ntl.bts.gov/DOCS/public-8.html, 1971.

[Glo05] Christian Gloor. *Distributed Intelligence in Real-Word Mobility Simulations*. Hartung-Gorre, 2005.

[Gup06] V. Gupta. *Distributed estimation and control in networked systems*. PhD thesis, CIT, California, May 2006.

[HBD02] S.P. Hoogendoorn, P.H Bovy, and W. Daamen. Microscopic pedestrian way finding and dynamics modeling. In *Pedestrian and Evacuation Dynamics*, pages 123–155. Springer, 2002.

[HHN98] P.S. Huang, C.J. Harris, and M.S. Nixon. Comparing different template features for recognizing people by their gait. In *Proceedings of the British Machine Vision Conference*, 1998.

[HHS+99] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *5th MOBICOM*, pages 59 – 68, 1999.

[HM95] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, May 1995.

[Hol03] Darryl J. Holman. *mle: A Programming Language for Building Likelihood Models ver 2.1*, volume 1. 2003.

[HSS02] M. J. Black H. Sidenbladh and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *Proc. European Conference on Computer Vision*, volume 1, pages 784 – 800, 2002.

[Ini01] Aware Home Research Initiative. Aware home. http://www.awarehome.gatech.edu/, 2001. Last accessed: Oct. 2006.

[Joh75] G. Johansson. Visual motion perception. *Scientific American*, 232:76 – 89, 1975.

[JPR97] K. Hsiao J. Paradiso, C. Abler and M. Reynolds. The magic carpet: Physical sensing for immersive environments. *Extended Abstracts of CHIŠ97*, pages 277 – 278, Mar. 1997.

[KKR04] T. Seppänen K. Koho, J. Suutala and J. Röning. Footstep pattern matching from pressure signals using segmental semi-markov models. In *12th European Signal Processing Conference*, Sept. 2004.

[Klu03] Hubert Klupfel. *A Cellular Automaton Model for Crowd Movement and Egress Simulation.* PhD thesis, University of Duisburg-Essen, 2003.

[KOC05] C. Wu K. Chang K. Okuda, S. Yeh and H. Chu. The geta sandals: A footprint location tracking system. In *Lecture Notes On Computer Science*, volume 3479, pages 120 – 131, 2005.

[KPN96] R.L. Knoblauch, M.T. Pietrucha, and M Nitzburg. Field studies of pedestrian walking speed and start-up time. Technical report, 1996. Transportation Research Record No. 1538, Pedestrian and Bicycle Research.

[KRCC95] A. Kale, A. Roy-Chowdhury, and R. Chellappa. Gait-based human identification from a monocular video sequence. In *Handbook on Pattern Recognition and Computer Vision*, pages 103–104. Scientific Publishing Company Pvt. Ltd, 1995.

[KRCK02] A. Kale, A.N. Rajagopalan, N. Cuntoor, and V. Krĺuger. Gait based recognition of humans using continuous hmms. In *Proceedings of the International Conference on Face and Gesture Recognition, Washington DC*, 2002.

[LB96] J. Little and J. Boyd. Recognizing people by their gair: the shape of motion. In *MIT Press Journal Videre*, 1996.

[LG02] L. Lee and W.E.L. Grimson. Gait analysis for recognition and classication. In *Proceedings of the IEEE Conference on Face and Gesture Recognition*, pages 155 – 161, 2002.

[LWHS02] D. Li, K. Wong, Y.H. Hu, and A. Sayeed. Detection, classification and tracking of targets in distributed sensor networks. Technical Report 19(2), IEEE Signal Processing Magazine, Mar. 2002.

[MAS97] F. Livesey M.D. Addlesee, A. Jones and F. Samaria. Orl active floor. In *IEEE Personal Communications*, volume 4.5, pages 35 – 41, 1997.

[MR05] Oded Maimon and Lior Rokach. *The Data Mining and Knowledge Discovery Handbook.* Springer, 2005.

[New06] Ascribe Newswire. Physical, psychological benefits of walking more each day. http://img.azcentral.com/health/fitness/articles/0405walkingbenefits.html, 2006. Last accessed: June 2006.

[NIS06] NIST/SEMATECH. e-handbook of statistical methods. http://www.itl.nist.gov/div898/handbook/, 2006. Last accessed: Jan. 2007.

[OA00] R.J. Orr and G.D. Abowd. The smart floor: A mechanism for natural user identification and tracking. In *Proc. 2000 Conf. Human Factors in Computing Systems*, 2000.

[OB80] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. In *IEEE Transactions on PatternAnalysis and MachineIntelligence*, volume 2(6), pages 522 – 536, 1980.

[PCB00] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location- support system. In *6th MOBICOM*, 2000.

[Ren03] Renesas Corp. *Hitachi Single-Chip Microcomputer, H8S Series Hardware Manual*, 3.0 edition, Mar. 2003.

[Ric04] B.S. Richardson. Implementation of efficient data extraction from a scalable sensor system). Master's thesis, Univ. of Limerick, Ireland, Sept. 2004.

[RPM95] J. Kerkhoff R. Pinkston and M. McQuiclken. Touch sensitive dance floor/midi controller. In *Proc. of the 1995 International Computer Music Conference*, pages 224 – 225, Sept. 1995.

[RR97] H. A. Rowley and J. M. Rehg. Analyzing articulated motion using expectation maximization. In *Proc. CVPR*, 1997.

[RRF04] K. Leydon R. Richardson, J. Paradiso and M. Fernstrom. Z-tiles: Building blocks for modular, pressure-sensing. In *CHI2004*, 2004.

[RS03] R.Jeanson and S.Blanco. A model of animal movements in a bounded space. In *Journal of Theor. Biology*, volume 225, Issue. 4), pages 443 – 451, Dec 2003.

[RS04] G. Kovacsc D.I. Perrett N. Kanwisher R. Saxe, D.K. Xiao. A region of right posterior superior temporal sulcus respondsto observed intentional actions. *Neuropsychologia*, 42:1435 Ű 1446, 2004.

[Sav04] D. Savio. Communication on rotating priorities. Master's thesis, Univ. of Applied Science. Heidelberg, Germany, Apr. 2004.

[Sch79] et al. Schlesinger. Terminology for model credibility. In *Simulation.* ACM Press, 1979.

[SD05] James Scott and Boris Dragovic. Audio location: Accurate low-cost location sensing. In *Pervasive*, pages 1–18, 2005.

[SIT00] Takushi Sogo, Hiroshi Ishiguro, and Mohan M. Trivedi. Real-time target localization and tracking by n-ocular stereo. In *OMNIVIS '00: Proceedings of the IEEE Workshop on Omnidirectional Vision*, page 153, Washington, DC, USA, 2000. IEEE Computer Society.

[SPR03a] J. Riekki S. Pirttikangas, J. Suutala and J. Röning. Learning vector quantization in footstep identification. In *Proc. 3rd IASTED International Conference on Artificial Intelligence and Applications*, page 413 Ű 417, 2003.

[SPR03b] J. Riekki S. Pirttikangas, J. Suutala and J. Röning. Footstep identification from pressure signals using hidden markov models. In *Proc. Finnish Signal Processing Symposium*, page 124 Ű 128, May 2003.

[SS73] P.H.A. Sneath and R. R. Sokal. Taxonomic structure. Technical report, W. H. Freeman Co., San Francisco, Calif, 1973.

[SS00] A. Stöhr and F.T. Sturm. Displays auf der basis von netzwerken: Algorithmen zurfehlertoleranten selbstorganisation. Technical Report intern, Siemens AG, ZT PP 2E, Nov. 2000.

[SSvL+02] Albrecht Schmidt, Martin Strohbach, Kristof van Laerhoven, Adrian Friday, and Hans-Werner Gellersen. Context acquisition based on load sensing. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pages Springer–Verlag, London, UK, 2002.

[Stu00] F.T. Sturm. Algorithmic display network organization simulator (adnos). Technical Report intern, Siemens AG, ZT PP 2E, Dec. 2000.

[TM94] P.A Thompson and E.W. Marchant. Simplex. http://www.ies4d.com, 1994. Last accessed: Dec. 2006.

[TMI04] T. Ikeda T. Murakita and H. Ishiguro. Multisensor human tracker based on the markov chain monte carlo method. In *Joint 2nd International Conference on Soft Computing and Intelligent Systems*, 2004.

[Uni01] John Hopkins University. Improved foot sensor. http://apps.nciia.org/WebObjects/NciiaResources.woa/, 2001. Last accessed: Oct. 2006.

[WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, second edition, 2005.

[WHFG92] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. In *ACM Transaction on Information Systems*, volume 10:1, pages 91 – 102, 1992.

[Wil06] U. Wilensky. Netlogo. http://ccl.northwestern.edu/netlogo, 2006. last accessed: May 2006.

[WJH97] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. In *IEEE Personnel Communications*, volume 4:5, pages 42–47, 1997.

[WP98] C. R. Wren and A. P. Pentland. Dynamic models of human motion. In *Proc. of FG 98, Nara, Japan*, Apr. 1998.

[YSP00] X. Feng Y. Song and P. Perona. Towards detection of human motion. In *Proc. CVPR*, pages 810 – 817, 2000.