The Design of a Pressure Sensing Floor for Movement-Based Human Computer Interaction

Sankar Rangarajan, Assegid Kidane, Gang Qian, Stjepan Rajko, and David Birchfield

Arizona State University, USA
{Sankaranaraya.Rangarajan, Assegid.Kidane, Gang.Qian, Srajko, Dbirchfield}@asu.edu

Abstract. This paper addresses the design of a large area, high resolution, networked pressure sensing floor with primary application in movement-based human-computer interaction (M-HCI). To meet the sensing needs of an M-HCI system, several design challenges need to be overcome. Firstly, high frame rate and low latency are required to ensure real-time human computer interaction, even in the presence of large sensing area (for unconstrained movement in the capture space) and high resolution (to support detailed analysis of pressure patterns). The optimization of floor system frame rate and latency is a challenge. Secondly, in many cases of M-HCI there are only a small number of subjects on the floor and a large portion of the floor is not active. Proper data compression for efficient data transmission is also a challenge. Thirdly, locations of disjoint active floor regions are useful features in many M-HCI applications. Reliable clustering and tracking of active disjoint floor regions poses as a challenge. Finally, to allow M-HCI using multiple communication channels, such as gesture, pose and pressure distributions, the pressure sensing floor needs to be integrable with other sensing modalities to create a smart multimodal environment. Fast and accurate alignment of floor sensing data in space and time with other sensing modalities is another challenge. In our research, we fully addressed the above challenges. The pressure sensing floor we developed has a sensing area of about 180 square feet, with a sensor resolution of 6.25 sensels/in^2. The system frame rate is up to 43 Hz with average latency of 25 ms. A simple but efficient data compression scheme is in place. We have also developed a robust clustering and tracking procedure for disjoint active floor regions using the mean-shift algorithm. The pressure sensing floor can be seamlessly integrated with a marker based motion capture system with accurate temporal and spatial alignment. Furthermore, the modular and scalable structure of the sensor floor allows for easy installation to real rooms of irregular shape. The pressure sensing floor system described in this paper forms an important stepping stone towards the creation of a smart environment with context aware data processing algorithms which finds extensive applications beyond M-HCI, e.g. diagnosing gait pathologies and evaluation of treatment.

1 Introduction

Movement-based human-computer interaction (M-HCI) systems are receiving increasing attention recently due to their immediate applications in a number of areas

G. Kortuem et al. (Eds.): EuroSSC 2007, LNCS 4793, pp. 46–61, 2007.

with significant impact in our daily lives, e.g., biomedical (real-time monitoring of patient rehabilitation and providing guidance), culture and arts (interactive dance performances), education (encouraging collaborative and embodied learning through real-time visual and audio feedback based on the movement of the users). M-HCI systems read and respond to the movement of the user. In order to enable M-HCI system to understand the user's movement robustly and accurately, it is important to augment the user's environment with novel sensors, paving way for enhanced human interaction with computers on the basis of position, static poses, dynamics gestures and movement qualities. Pressure sensing plays a vital role in M-HCI systems. Every movement has certain motivation driven physical effort attached. Pressure sensing systems aid to understand and comprehend the nuances of such a physical effort thereby exploring the inherent nature of the human body as a powerful communication medium. Pressure sensing systems aiming at such M-HCI applications require a large sensing area (for unconstrained human movement), high sensor densities (for detailed and accurate representation of interacting objects), high frame rate and small sensing latency (for real time application), modular, scalable and portable design (for easy reconfiguration to suit external environments) and lastly integrability with other sensing modalities (for multimodal HCI).

In related prior work, various pressure sensing systems had been developed to capture and view pressure information associated with human movement across a floor. A detailed performance comparison study of those existing pressure sensing systems in terms of the desired features are listed in Table 1.

Table 1. Performance comparison table of existing pressure sensing floor systems

Sensor System	University	Year	Sensing method	Sensing area (Sq.feet)	Frame rate (Hertz)	Sensor density (sensor/sq.inch)	Data resolution (Number of bits)	Integrability	Modular	Portable
MIT Magic Carpet [1]	MIT Media Labs	1997	Piezoelectric wires	60	60	0.06	8	Yes	No	Yes
LiteFoot [2]	University of Limerick Ireland	1997	Optical proximity sensors	42.25	100	0.3	NA	No	No	No
ORL Active floor [3]	Oracle Research Lab	1997	Load cells	10.76	500	0.01	16	No	No	No
High resolution pressure sensor [4] distributed floor	University of Tokyo, Japan	2002	Binary switch	43	15	10.57	1	No	Yes	No
Z-Tiles [5]	University of Limerick Ireland & MIT Media Lab	2004	FSR	NA	100	0.5	12	No	Yes	Yes
Floor Sensor system [6]	University of Southampton,UK	2005	Binary switch	15.68	22	1.3	1	No	No	No
AME Floor I [7]	Arizona State University 2004-05	2004-05	FSR	9	10	0.44	8	Yes	No	No
AME Floor II [8] [9]	Arizona State University 2005-06	2005-06	FSR	60	33	6.25	8	Yes	Yes	No
PROPOSED PRESSURE FLOOR			FSR	180	43	6.25	8	Yes	Yes	No

Color coding: 1, 2, 3

NA - NOT AVAILABLE

The ranking in each dimension (column) is color-coded such that the best system is in dark green, the second best in lighter green, and the third in very light green. MIT

Magic Carpet [1] and LiteFoot [2] had fairly large sensing area and frame rate but were limited by poor sensor densities. ORL active floor [3] used load cells which lack the capability of detailed pressure measurement and cannot be used for applications requiring high sensor densities. High resolution pressure sensor distributed floor [4] has the best sensor density so far but was a binary floor (poor data resolution) that just detects presence or absence of pressure and does not give any measurement of pressure values on an analog scale. Z-tiles floor space [5] utilized a modular design, had high frame rate and data resolution but again suffers from low sensor density. Floor sensor system [6] is a low cost design but again a binary floor with poor data resolution. Also most of the sensing systems except [1] were stand alone systems and lacked the capability to be integrated in a multimodal environment which is vital requirement for our application. In-shoe sensors [10] have also been considered for force and pressure measurements but they have a limited scope of foot pressure measurement only. Also the in-shoe systems tend to alter the subject's actual pressure application due to their tendency to alter foot orientations by close contact.

It is quite obvious that all the sensing systems listed above have at least one serious limitation rendering it unsuitable to meet our application goals. It is worth mentioning that two generations of pressure sensing floor systems were developed with very similar goals as ours at the Arts, Media and Engineering (AME) Program at Arizona State University, namely, AME Floor I [7] and AME Floor II [8, 9] listed at the bottom of the table. It is apparent from the comparison table that the second generation did see pronounced feature improvements over the first generation. AME floor I [7] was a smaller prototype floor with 256 force sensing resistors arranged in less dense sensor matrix. During tests [7] it was found that there were large zones of no pressure detection during several activities. Also the scan rate was low deeming it unsuitable for real time M-HCI applications. These shortcomings were addressed by AME floor II [8, 9] with high sensor densities and high frame rate. Although AME Floor II [8, 9] showed significant advances and extended capabilities over AME floor I [7], it covered only a fraction of the sensing area required for our application, showed high sensing latency and lacked user friendliness. Also it showed preliminary multimodal integrable capabilities in temporal domain only and not spatial domain.

To fully address these issues, we have developed our own ingenious and in-house pressure sensing floor system PROPOSED in this paper and listed in the last row of table 1. Our proposed floor system is characterized by large sensing area, higher frame rate, smaller latency, enhanced user friendliness, spatial and temporal integrability with other sensing modalities in a multimodal environment, modular/scalable design etc. and thereby matching our ideal pressure sensing demands for real time M-HCI application. Comparison with other systems reveals that our proposed system in this paper ranks among the top 3 in most of the dimensions of the performance criteria. Although there are four systems with frame rates higher than that of our system, the sensing area and sensor resolutions of these systems are much lower than our system.

In this paper, we present system level description of our proposed pressure sensing floor followed by a discussion on hardware and software developments. Then we discuss the design methodologies for integration of the floor system with the marker based motion capture system as a first step towards the creation of a smart multimodal

environment**.** The paper finally concludes with two interesting applications that we are currently exploring upon using such a powerful multimodal sensing set-up.

2 Pressure Sensing Floor Overview

The pressure sensing floor system consists of 96 networked pressure sensing units arranged in a rectangular matrix of 12 rows x 8 columns (shown in Fig. 1) spanning a total sensing area of 180 square feet. Each unit consists of a pressure sensing mat and associated supporting floor hardware. Each mat is in the size of 19"x17" embedded with a sensel array of 48x42 = 2016 sensels, resulting in a sensel resolution of 6.25 sensels/in^2. Force Sensing Resistors (FSR's) are used as sensel elements on the mat having an active area of 6mm x 6mm and made using pressure sensitive polymer between conductive traces on sheets of Mylar. The resistance of FSR is of the order of Mega ohms in the absence of pressure and drops to few kilo ohms when pressure is applied. Such a modular, scalable and networked architecture makes the floor readily reconfigurable and suitable for installations such as hallways, walk paths, and even spaces of irregular shape. We have estimated the cost of our 'one mat' system to be $600 and we can easily build smaller floor with fewer mats (as required by the application) at very low costs.

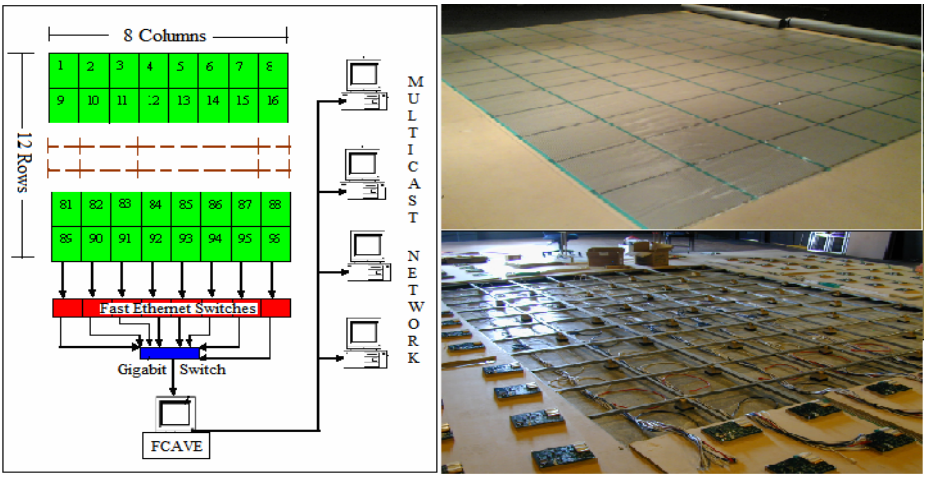

Fig. 1. Floor System overview and related network architecture (Left). Top view of the floor (Top right). Skeletal view of the floor (bottom right).

The floor hardware [9] comprises of microcontroller, multiplexers, A/D converter and Ethernet enabled rabbit controller which are all wired together on a processing and control board. The block diagram of the floor hardware is shown in Fig. 5. The microcontroller scan routine generates the timing and control signals for all hardware components to coordinate and sequence their operation. By optimizing microcontroller firmware, our pressure sensing system is made to run up to 43 Hz with average latency of 25 milliseconds. The pressure sensors of single mat are multiplexed to read the

analog pressure values and then fed to A/D converter to produce 8 bit digital pressure data. The rabbit controller collects the digital pressure data of all sensors on one mat to produce a single mat packet and then transmits the packet over the network. All the pressure sensing units are assigned static IP addresses and they form a local private network. The data output of the rabbit controllers travels through two layers of network switches to the host computer. The power and sync clock are daisy chained and distributed to all the mats on floor. The pressure sensing floor is synchronized with the motion capture system by an external sync clock.

Floor Control And Visualization Engine (FCAVE) software has been developed at the host computer for floor control and visualization. FCAVE has an interactive graphical user interface (GUI) with various control buttons and indicators (shown in Fig. 2) and it is programmed to respond dynamically to user input. This software receives the raw pressure data packet for each mat separately, assembles the data of all 96 mats, assigns an incremental frame number and creates floor data frame which is then ready for further processing. FCAVE software has two operating modes namely 'live mode' and 'playback mode'. As the name implies, real time data collection and processing is done in the 'live mode' whereas offline data processing from a recorded pressure data file is usually done in the 'playback mode'. Furthermore playback can be done in synchronous and asynchronous ways. Synchronous playback streams the recorded pressure data synchronous with the motion capture playback stream. Asynchronous playback streams the recorded pressure data at the desired frame rate without any synchronization with motion capture system. FCAVE also offers various other controls like multicast pressure data to users on network, grayscale display of pressure information, set noise filter value, perform mean shift tracking of pressure clusters ,frame counter reset, record to file etc. FCAVE Software development paved way for enhanced user-friendliness (with a lot of GUI features shown in Fig. 2), efficient data compression and mean shift tracking of active, disjoint pressure clusters in real time.

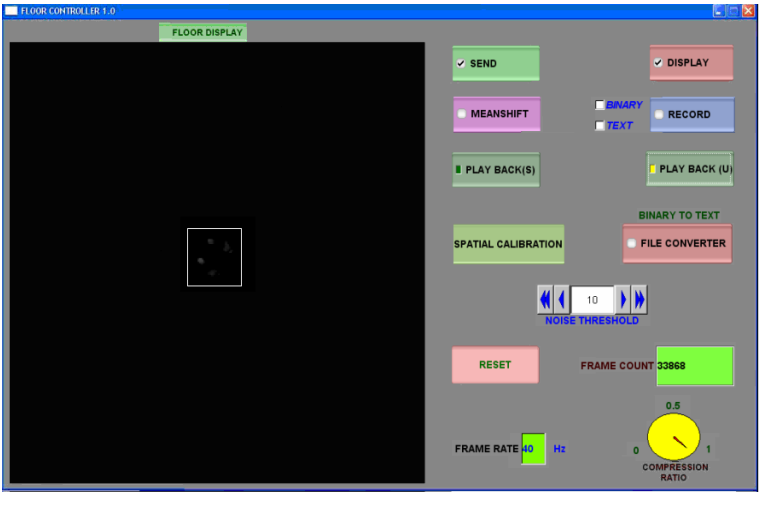

Fig. 2. Interactive Graphical User Interface (GUI) of the FCAVE software. The square box on the left highlights the pressure footprints.

3 Optimization of System Latency and Frame Rate

Small latency is critical for real time sensing systems used in M-HCI applications. Latency is defined as the time lag between the time instant of the true event and the time instant the pressure data pertaining to the true event arrives at end users on a multicast network. Labview software application and National Instruments (NI-DAQ 6020E) data acquisition hardware are used to measure and quantify the overall latency and the latency at each and every point along the data path.

The overall system latency is the sum of two components namely intrinsic latency and extrinsic latency. Intrinsic latency is defined as the latency induced by the sensor scanning process. Each sensing unit has a pressure mat with 2016 sensors and an associated hardware control board for pressure data collection and signal conditioning. All sensors are scanned sequentially from sensor 1 to sensor 2016 to read the pressure values. There is an inherent delay for the scanning process to complete and pressure packet to be produced. This delay is called as the intrinsic latency which is present due to lag in various hardware components on the hardware control board. The microcontroller generates the sensor scan signals and the scan routine incorporates all the hardware component delays. Thus total execution time of the microcontroller scan routine T_{scan} determines the frame rate F ($F = 1/ T_{scan}$) of the system. After a complete mat scan of 2016 sensors, the pressure data packet for that mat is produced. Extrinsic latency is defined as the time taken for such a pressure data packet to reach the end users on the multicast network and it accounts for the network transmission delay and FCAVE software delay. Our main focus was intrinsic latency reduction since it constitutes a major portion of the system latency and directly impacts the frame rate of system. Due to sequential scanning process, the intrinsic latency is direct function of the active sensor location given by a sensor address (An active sensor would be one that has pressure applied on it and sensors are addressed sequentially from 1 to 2016). A mathematical relationship is first established which gives an expected range of the intrinsic latency values based on the system scan rate and active sensor location. From this theoretical model, we know what latency distribution to expect when pressure is applied on a particular sensor location and later we did latency experiments to verify the same. The following section presents the mathematical relationship between intrinsic latency, frame rate and active sensor location.

3.1 Intrinsic Latency, Frame Rate, and Active Sensor Location

Let's assume that the system is running at a frame rate F and the time taken for one complete scan cycle of N sensors ($N = 2016$ in our case) is T_{scan}. We define pressure sensors applied with active load as *active sensors*. Let L be the address of such an active sensor. We are interested in finding the intrinsic latency related to this sensor at L. Let U be the address of the sensor currently being scanned at the time instant when the pressure application occurs on sensor L. Let X_L and X_U be time elapsed since the start of the scan until the sensor L and sensor U are reached respectively by the sequential scan routine, i.e.

$$X_L = \frac{1}{N} L \times T_{scan}\ ,\ X_U = \frac{1}{N} U \times T_{scan} \tag{1}$$

According to the relationship between X_U and X_L, there are two different cases to be considered which are pictorially represented in Fig. 3.

- Case 1: $X_U \leq X_L$, pressure applied on sensel L is registered in the current scanning cycle.
- Case 2: $X_U > X_L$, pressure applied on sensel L is registered in the next scanning cycle.

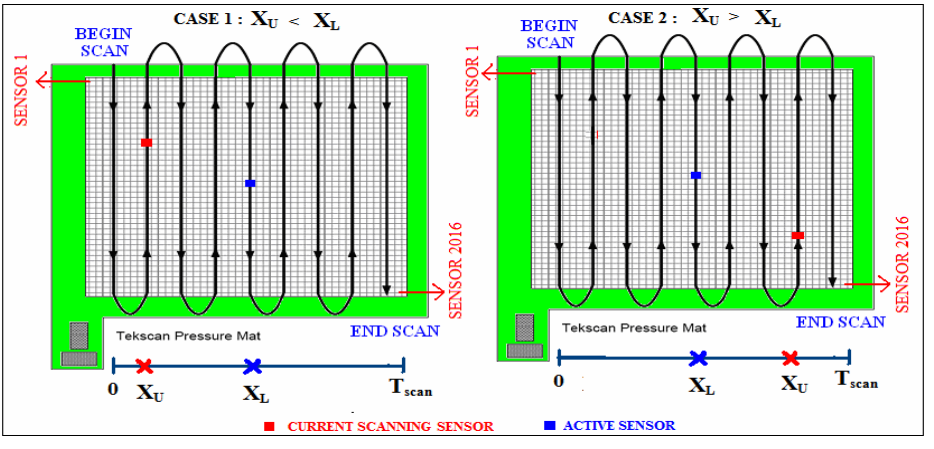

Fig. 3. Sequential mat scan process and depiction of Case 1 and Case 2

Hence, given L, the intrinsic latency τ caused by system scan is a function of X_U,

$$\tau(X_U) = \begin{cases} T_{scan} - X_U\ , \text{when } 0 \leq X_U \leq X_L \\ 2T_{scan} - X_U\ , \text{when } X_L < X_U < T_{scan} \end{cases} \tag{2}$$

Since X_U assumes a uniform distribution in [0, T_{scan}] it can be easily shown that τ is uniformed distributed in the range given below:

$$T_{scan} - X_L < \tau \leq 2T_{scan} - X_L \tag{3}$$

Therefore, the mean intrinsic latency for the sensel at L is given by

$$\tau_m = 1.5T_{scan} - X_L \tag{4}$$

Thus the mean intrinsic latency is a direct function of T_{scan} and active sensor location X_L. Furthermore, since L can also be treated as a uniform random variable between 1 and N, the mean average intrinsic latency of all sensels on a mat is given by

$$E\{\tau_m\} = 1.5\ T_{scan} - E\ \{X_L\} = T_{scan} \tag{5}$$

Latency experiments have been conducted to verify the theoretical model derived above. Pressure is applied on a set of fixed sensor locations on the mat and the mean system latency is computed for 100 trials. Fig. 4 indicates the correlation between the theoretical and practical data sets when the system is running at 40Hz. The systematic offset between the theoretical and practical data sets is due to limited accuracy of experimental measurement of latency in milliseconds.

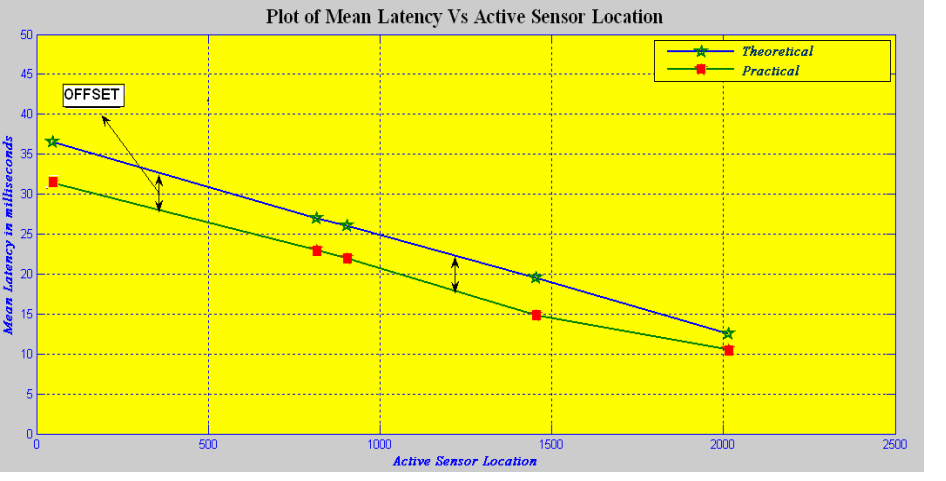

Fig. 4. Plot of mean latency vs. active sensor location

It is apparent from equation (5) that we can minimize intrinsic latency by minimizing T_{scan}, or equivalently maximizing the frame rate ($F = 1/T_{scan}$). Hence efforts were invested to increase the frame rate and reduce intrinsic latency which is described in the following section.

3.2 Maximization of Frame Rate

Frame rate of floor system is determined by the speed of hardware components on the hardware control board. Every hardware component has certain delay or lag associated with it. The microcontroller scan routine incorporates all the hardware component delays and accordingly generates the control signals. The sum of all hardware component delays gives minimum T_{scan} required whose reciprocal gives the maximum achievable frame rate. Fig. 5 shows the block diagram of floor hardware annotated with delay values for each hardware component explaining how we had achieved a maximum frame rate of 43 Hz in our proposed floor system from an old value of 33 Hz in AME Floor II (our precursor work). It is important to note that suffix (II) on Fig. 5 refers to AME Floor II whereas suffix (P) refers to floor system proposed in the paper. The block diagram quantifies the time savings obtained on each hardware component in the proposed system relative to AME Floor II. These time savings and hence increase in frame rate are obtained by choosing high performance hardware components and doing a more refined timing analysis on each component to determine their operational delay.

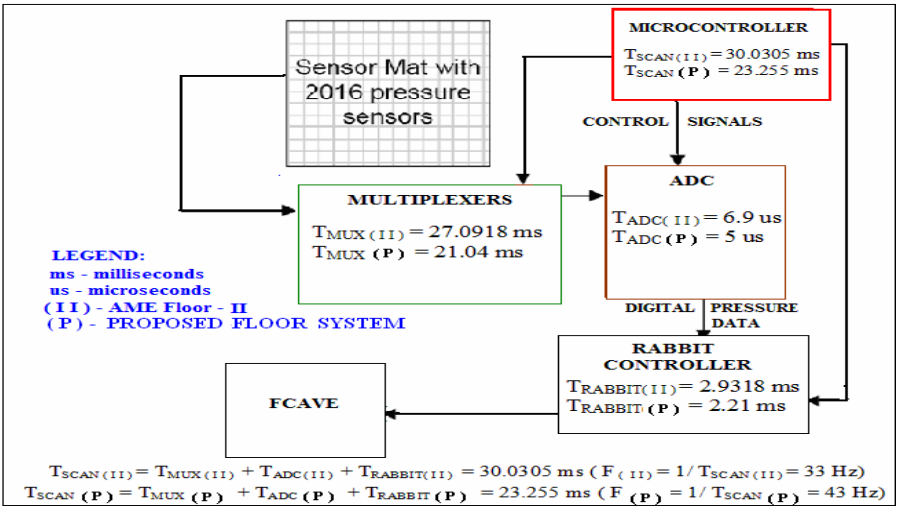

Fig. 5. Block diagram of Floor hardware annotated with hardware component delays

4 Software (FCAVE) Capabilities

Automatic processing and interpretation of the pressure information is done by FCAVE software. FCAVE not only collects and displays pressure information but also possess the capability of doing data compression and person location tracking by mean shift algorithm.

4.1 Data Compression

Each pressure mat has 2016 sensors and each sensor in turn sends one byte of pressure data at 43 Hz. Thus each mat data packet size adds to 2017 bytes which includes 2016 bytes of pressure data and one byte of frame number. The data volume from the entire floor comprising of 96 mats is a whopping 8.4 MB/sec. Usually, except a small area where the subject is in contact with the pressure sensing floor, most of the sensors do not have any load acting on them. Consequently a large proportion of the sensor data are null values of pressure or noise serving no interest to applications. Also there has been slight random noise observed in few sensors because of the nature of the sensing material which reports small values of pressure. Hence a simple but elegant compression algorithm is implemented by FCAVE to filter out all pressure values below the chosen noise threshold and pack only “active” sensor values and their addresses (location on floor system matrix) to be sent out to the end users on the network. Compression ratio as high as 0.9 is observed under normal case with five subjects which proves significant data volume reduction on the network.

It is known that compressed data packet comprise of only active sensor values and their address whereas the uncompressed data packet comprise of all sensor values (arranged in a sequence) and no address information since its address is implied by its location in the data packet. Thus the compression algorithm adds an additional

overhead of sensor address which works well for low user activity with less active sensors. However as the user activity on the floor increase or when large numbers of sensors are active, the packet size also grows and a point is reached when compressed data volume exceeds uncompressed data volume. It is determined that this breakeven point is generally high and beyond bounds for normal usage. However, we are currently working on a dynamic compression scheme whereby the system is context aware and detects the extent of user activity and makes a decision whether to do compression or not.

4.2 Mean Shift Tracking of Pressure Clusters

Context awareness is the vital part of any smart environment. Perceiving context means sensing the state of the environment and users and it can be done with regard to a person or an activity. This may involve a variety of tasks such as person recognition, person location tracking, activity detection, activity recognition, activity learning etc. The primary step to accomplish the above tasks is to develop an efficient tracking procedure that shall ascertain the person location on the floor and also shift in the pressure gradient. The latter may lead to the study of various pressure patterns tied to each and every user activity. A mean shift algorithm is used to achieve the above mentioned goal. Mean shift is a simple iterative procedure that shifts each pressure data point to the average of the pressure data points in the neighborhood.

4.2.1 Mean Shift: An Introduction

Mean shift is the process of repetitively shifting the center t to the sample mean. The sample mean of samples S under a kernel $K(x)$ centered at t, with sample weights $w(s)$, can be found using this equation:

$$m(t) = \frac{\sum_{s \in S} K(s-t)w(s)s}{\sum_{s \in S} K(s-t)w(s)} \tag{6}$$

where m(t) is the new sample mean [11]. In [12] it's proven that if the kernel $K(x)$ has a convex and monotonically decreasing profile $k(||x||^2)$, then the center t will converge onto a single point. The kernel used in our tracking algorithm is the truncated Gaussian kernel which is the combination of the flat kernel and Gaussian kernel. The truncated Gaussian kernel is given by

$$(F_\lambda G_\beta)(x) = \begin{cases} e^{-\beta\|x\|^2} & if\ \|x\| \le \lambda \\ 0 & if\ \|x\| > \lambda \end{cases} \tag{7}$$

where λ is the radius of the Gaussian kernel and β is the Gaussian kernel coefficient.

4.2.2 The Clustering/Tracking Algorithm

The algorithm is iterated for every frame of pressure data. Each and every frame of pressure data contains information about the location of pressure and value of

pressure at that location. The pressure values constitute the weights and pressure location constitutes the data points that need to be iterated using the mean shift algorithm. The full algorithm for finding and tracking the pressure clusters is given below.

1) For the first frame of pressure data or new cluster formation, cluster centers and the data points are one and the same i.e. the center set T is the same as the data set S, and both evolve with each iteration using the mean shift formula in equation (6) and truncated Gaussian in equation (7). Data points are clustered through the blurring process [11] using the observed pressure data as the weight used in (6). Once the process has converged, the data set will be tightly packed into clusters, with all of the data points located closely to the center of that cluster. (The process is said to the converged either after the maximum number of iterations defined by the algorithm or earlier when the mean shift of centers becomes less than the convergence threshold) After convergence, each cluster has a 'center' and 'label' associated with it. All data points not associated with any cluster center are classified as or orphan pressure points.
2) For every subsequent pressure data frame, centers from the previous frame are updated through the mean shift algorithm (6) using current observed pressure values as weights and checked for convergence. In practice, entirely new data points resulting in new cluster centers (new labels) can occur which is computed in step (3).
3) Calculate the number of orphan pressure points. If the number of orphan pressure points exceeds a chosen threshold then repeat step (1) to find new cluster centers. Orphan pressure points fewer than the chosen threshold are discarded.
4) Perform mean shift using the new set of cluster centers (repeat steps 2 & 3).

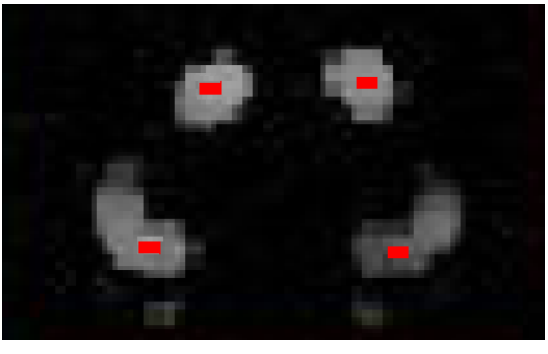

Fig. 6. Snapshot showing clustering and tracking by mean shift on left foot and right foot. Two pressure clusters are formed for each foot (one for heel and one for toe) and cluster centers are depicted by red dots.

5 System Integration for Multimodal Sensing

Multimodal systems have always proved to be robust and effective than independent uni-modal systems because it provides wide varieties of information for better realization and assimilation of the subject movement in capture space and also allows the users to interact multimodally with the system. They provide high redundancy of content information which leads to high reliability. After the completion of the pressure sensing floor, efforts have been put in to integrate the floor with the motion capture system to create a smart environment.

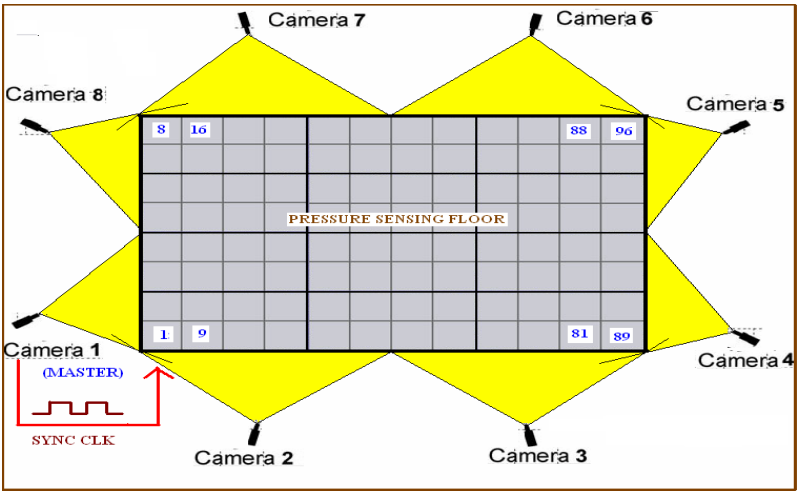

Fig. 7. Multimodal sensing set up of floor and motion capture system

A common capture volume (12' by 15') is first created within the sensing capabilities of the floor and motion capture system. The motion capture cameras are arranged around a capture volume and the floor forms a part of the capture volume as shown in Fig. 7. The location of the floor with respect to the coverage area of the cameras is important when pressure data about some movement needs to be interpreted with the marker. The pressure floor and motion capture system are integrated with respect to time and spatial domains. A subject moving in the capture space is sensed by both systems and they give information about the location and activity of the subject. Motion capture data contains the 3D location coordinates of the markers in physical space whereas the pressure data contains the pressure values and 2D location. Both sensing systems have independent coordinate set and hence spatial alignment by means of coordinate transformation becomes essential to ascertain the location of the subject in common capture space. Also any activity done by the subject is being detected by both systems simultaneously and hence both sensing modalities must operate synchronously. Thus time synchronization and spatial alignment are critical for two data sets to be highly correlated to ensure holistic inference.

5.1 Time Synchronization

Time synchronization of the floor and motion capture system is achieved by means of a common sync clock. This sync clock is generated by the master camera of the motion capture system and is used to trigger the scan of the floor. This sync clock is also used by the motion capture system to control the camera shutters. In this way, the scan of the floor and the camera image capture can be synchronized in time domain. Also the motion capture system is always set to run at multiples of the floor frequency. The common sync clock runs at the frequency of the motion capture system and that clock is down sampled by a factor to generate the scan frequency (or frame rate) of the floor. Currently we clock the motion capture system at 120 Hz and the floor at 40 Hz (which should be less than maximum achievable floor frame rate of 43 Hz). This arrangement generates 3 motion capture data frames for every single pressure date frame. So the motion capture data frames are down-sampled (redundant frames are dropped) to create an equal number of floor and motion capture data frames for comparison purposes. All data frames are referenced by means of frame numbers to track the same event detected by both systems.

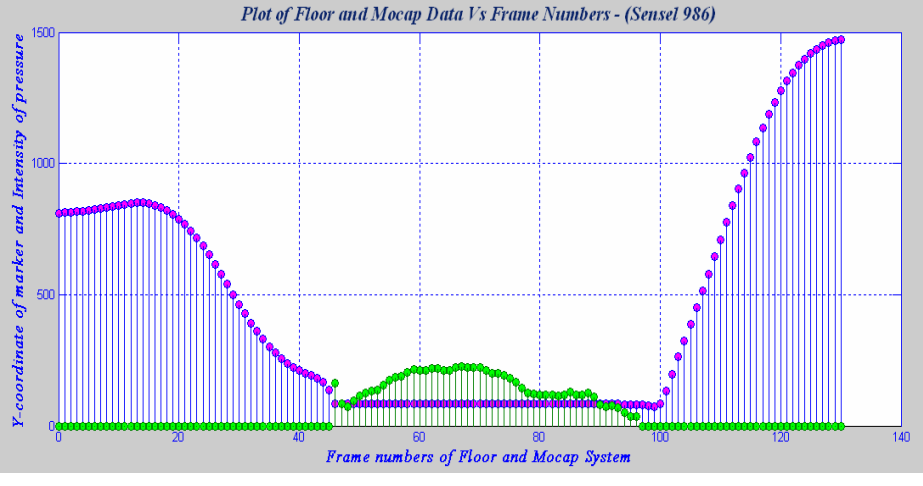

Fig. 8. Plot of Pressure and marker data v.s. frame numbers

A simple experiment is conducted to test the time synchronization of the floor and motion capture system. A mallet with a single marker on its head is banged on a single pressure sensor of the floor from a fixed height. The vertical coordinate (Y- coordinate) of the marker and pressure value on that sensor are monitored over time. Ideally the pressure sensor value should peak when the marker coordinate is at the lowest position (ground level). Fig. 8 gives the time-sampled plot of the sensor pressure value (green dots) and marker height (pink dots). The results obtained agree with our expectation thereby demonstrating a perfect frame alignment between the floor and motion capture system.

5.2 Spatial Alignment

The floor coordinate system is a two dimensional system in sensor units whereas the motion capture coordinate system is a three dimensional system in mm units. Hence it is essential to implement coordinate transformation between the floor and motion capture system so that we can view the events in one coordinate space for ease of inference and visualization. A spatial calibration procedure is in order to align the floor and motion capture system in physical space. Firstly the motion capture system is calibrated and stabilized. Three reflective markers are placed on the edge of the floor in order to get the boundary co-ordinates of the floor in motion capture coordinate space. Using this information, we compute three co-ordinate transfor-mation parameters namely rotation, translation and scaling. These parameters constitute the coordinate transformation matrix which is then applied to each and every floor coordinate to get the respective coordinate in the motion capture system. The converse also can be computed to view the data in the floor coordinate space alone. Spatial alignment computations are done by FCAVE software in real time.

6 Applications in Multimodal Movement Sensing and Analysis

6.1 Balance Analysis

Falling is one of the major health concerns for elderly people and incidence of falls is high for persons aged over 75. Hence an efficient fall detection system is necessary to detect potential situations of fall and signal the user of an impending fall or alert for assistance after the person is immobilized by fall. The state of body balance is the feature of interest in fall detection systems. We have collected data of different on-balance and off-balance body postures and currently evaluating on a fall detection algorithm. The state of body balance is characterized by center of gravity (COG) and center of pressure (COP). COG is computed from the motion capture data by assigning weight to each marker and computing the weighted mean. If the weight of each marker represents the weight of the body mass around that marker, the weighted mean is a good approximation of the center of gravity. Similarly the COP is the weighted mean of all the pressure data points. The subject's overall state of balance is determined by the relative positions of the COG and COP. If the COG is directly above the COP, the subject is in a state of balance. In other words, the subject is in a state of perfect balance when the projection of the COP and COG on a horizontal plane superimposes each other. As COP and COG moves away from each other, the subject slowly transitions into a state of off-balance. Thus it is obvious that time synchronization and spatial alignment of both sensing systems are critical for such an exercise. Since feelings of balance are visceral in human beings, such a quantitative approach paves way to tie the behavior of the system to a sensation/feeling that is very internal and apparent to the user and thereby complementing human computer interaction.

6.2 Gesture Recognition

This multimodal sensing system has also been used to drive a gesture recognition system that uses both kinematics and floor pressure distribution to recognize gestures.

Such a gesture recognition system can distinguish gestures that have similar body shapes but have different body weight distributions as shown in Fig. 9. These two gestures are recognized as one and the same by marker based motion capture system due to similar body shape. Hence pressure sensing becomes vital to distinguish between such gestures. The ability of the gesture recognition system to read and analyze both body kinematics and pressure distributions encourages users to communicate with computers in expressive ways.

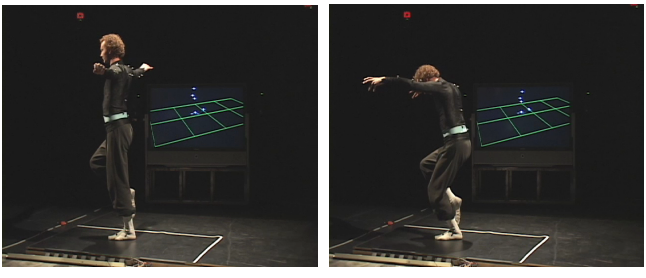

Fig. 9. Snapshots of two gestures with similar body shape but different weight distribution

7 Conclusions and Future Work

We have successfully designed, developed and deployed a pressure sensing floor system with a higher frame rate, less latency, high sensor resolution, large sensing area that can provide us with real time data about the location and amount of pressure exerted on the floor. The floor has been integrated and synchronized with the marker based motion capture system to create a smart environment for M-HCI application. Our present direction is towards extending the context aware capabilities of the floor system. We are currently working on improving the fall detection algorithm by collecting data for various on-balance and off-balance body postures and analyzing them. We are also working on an algorithm to distinguish between the left foot/right foot and heel/toe on the basis of shape. Shape descriptors such Fourier, Hu moments come in handy for such an analysis. Such intelligence to the floor to recognize and distinguish the left /right foot and heel/toe paves way for recognizing gestures with varying foot contact. Also most of the gait pathologies are reflected by abnormal pressure patterns localized to either the toe or heel. Hence results of the above work could find extensive usage in gesture recognition, video gaming, rehabilitation work etc.

References

[1] Paradiso, J., Abler, C., Hsiao, K., Reynolds, M.: The Magic Carpet: Physical Sensing for Immersive Environments, Ext. Abstracts CHI 1997, pp. 277–278. ACM Press, New York (1997)

[2] Griffith, N., Fernström, M.: LiteFoot: A floor space for recording dance and controlling media. In: Proceedings of the 1998 International Computer Music Conference, International Computer Music Association, San Francisco, U.S.A., pp. 475–481 (1998)

[3] Addlesee, M., Jones, A., Livesey, F., Samaria, F.: The ORL Active Floor. IEEE Personal Communications 4(5), 35–41 (1997)
[4] Morishita, H., Fukui, R., Sata, T.: High Resolution Pressure Sensor Distributed Floor for Future Human-Robot Symbiosis Environments. In: Intl. Conference on Intelligent Robots and Systems, Switzerland, IEEE, Los Alamitos (2002) (Ref: 0-7803-7398-7/02 @ 2002)
[5] Richardson, B., Leydon, K., Fernström, M., Paradiso, J.: Z-Tiles: building blocks for modular, pressure-sensing floorspaces. In: Extended Abstracts of the 2004 conference on Human factors and computing systems, Vienna, Austria, pp. 1529–1532 (2004)
[6] Middleton, L., Bus, A.A., Bazin, A.I., Nixon, M.S.: Floor Sensor system for gait recognition, University of Southampton. In: UK Fourth IEEE workshop on Automatic Identification Advanced Technologies (AutoID 2005), pp. 171–176 (2005)
[7] Kidané, A., Rodriguez, A., Cifdaloz, O., Harikrishnan, V.: ISAfloor: A high resolution floor sensor with 3D visualization and multimedia interface capability. AME Program, AME-TR-2003-11p (2003)
[8] Srinivasan, P., Birchfield, D., Qian, G., Kidané, A.: A Pressure Sensing Floor for Interactive Media Applications. In: Proc. of ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE), Valencia, Spain, pp. 278–281 (June 2005)
[9] Srinivasan, P.: Design of a Large area pressure sensing floor. A thesis presented for the requirements Master of Science Degree, Arizona State University (May 2006)
[10] Paradiso, J., Hsiao, K., Benbasat, A., Teegarden, Z.: Design and Implementation of Expressive Footwear. IBM Systems Journal 39(3 & 4), 511–552 (2000)
[11] Cheng, Y.: Mean Shift, Mode Seeking and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)
[12] Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2000), Hilton Head Island, South Carolina, vol. 2, pp. 142–149 (2000)