FP7-ICT-2007-C
Objective ICT-2007.8.0
FET-Open 222107

NIW
Natural Interactive Walking

Deliverable 3.2

Sensor Data Fusion and Gesture Analysis Tools

Yon Visell, Rishi Rajalingham, Severin Smith, Martin Otis, Guillaume Millet, Jeremy R. Cooperstock - McGill Univ.

Federico Fontana - UNIUD
Marco Civolani, Pietro Polotti - UNIVR

Sep. 30, 2010
v. 1.0

Classification: PU

Contents

1 Multimodal Sensing Architecture

The NIW multi-touch floor surface uses multiple input modalities to infer a representation of the user's state. An array of 36 tiles, each equipped with four force sensors in their corners (144 sensors in total), measures the forces acting at any given time on the surface. Simultaneously, one or more motion tracking systems can be available to track the user's feet or other points of interest using infrared reflective markers. For modularity and independence of programming language, this data is processed in a tiered fashion with a strong reliance on Open Sound Control (OSC) for data transfer.

The design of our current architecture is illustrated in Figure 1.1.

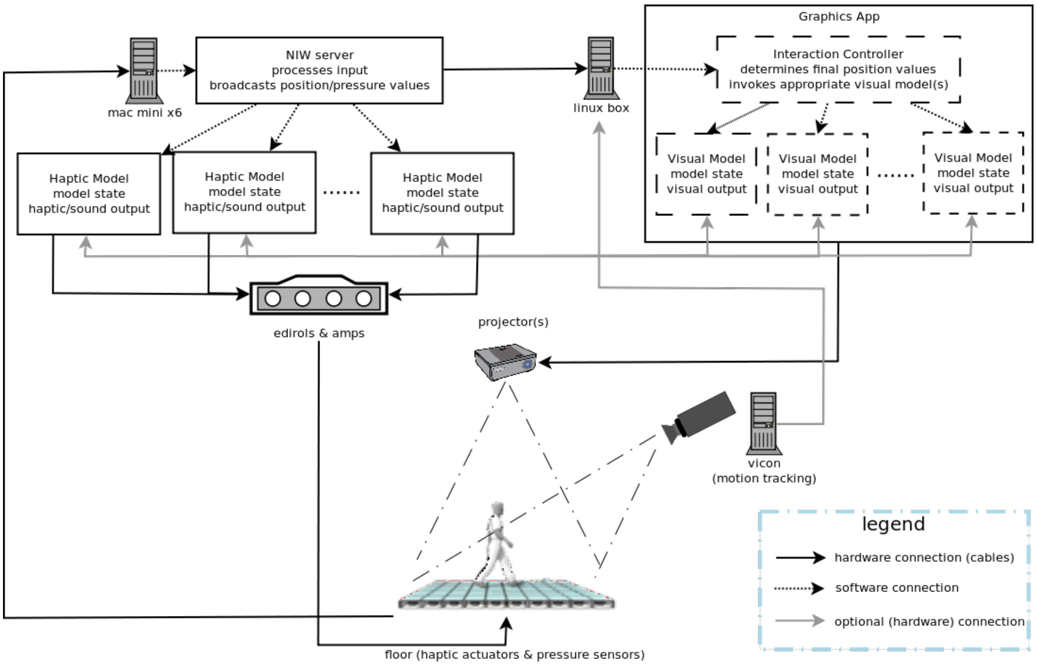


Figure 1.1: System architecture

At the lowest level, the output of the force sensors is relayed by an FPGA module using OSC protocol to the *haptic feedback rendering cluster*, a set of Mac Mini computers. This cluster hosts an OSC-based data server that performs filtering and basic feature extraction on the raw force information for downstream clients. In addition to configurable discreet filters, the server can determine user-floor contact points and label them with unique identifiers that are persistent in time. Normally, such data is either requested internally, via a loopback call, to drive the real-time haptic rendering or it is passed over the network to the data aggregator at the next tier

The downstream data aggregator, which is outside the haptic rendering cluster, is responsible for sensor fusion. It subscribes to the data stream of high level features extracted from the force sensor data in addition to the data streams of any available motion capture systems. Using this data, it tries to build a more reliable estimation of the user's state, including 3D position, rotation and contact forces. With multiple data sources, the aggregator can attempt to compensate for possible occlusion of the motion capture hardware and can perform sanity checking of the output data, in effect, carrying out a certain degree of sensor fusion processing. This capability is currently under development.

The data is then available, either at the software level via a Java API, or to clients subscribing over the network using OSC. In most cases, data would be used directly to control visual feedback on the floor and to drive the principle parameters of the haptic feedback.

In addition to analysis and sensor fusion tasks, the aggregator system acts as a centralized input mechanism for all graphical floor based applications. This standardizes input response behavior across all applications and allows all applications to benefit retroactively from improvements to the data analysis and fusion logic.

2 Gesture Spotting for Body Movements

Foot-based interaction has received growing interest as a new modality for performing spatial navigation and exploration tasks. Additionally, the availability of new, low cost, commercial sensing systems have spurred increased, if somewhat limited, prototyping activities. However, most research has been limited by two factors: the lack a large enough sensing surface to allow significant user movement and the absence of floor projection to provide context for actions and gestures. This motivates our investigation and comparison of various gesture-spotting possibilities for foot-based spatial navigation. For our initial experimentation, we begin with navigation of geosptial data sets, given the significant body of work that has already been conducted in this area with multi-touch displays, in particular using mobile devices. However, it has been suggested that tasks of a non-accurate and secondary nature, like as those involved in such navigation, may be well adapted to foot-based interaction methods [8].

Our design consists of a CAVE-like environment, in which a map is projected onto a multi-touch pressure sensitive floor surface. The pan and zoom are controlled through foot and body gestures. Simultaneously, corresponding street view images are projected on the front and side walls. The various interaction paradigms being explored are described as follows:

2.1 Overview of Paradigms

2.1.1 Classic Interface

This interface applies the traditional map interface: four arrow buttons for panning and a slider for zoom, to virtual controls on the floor.

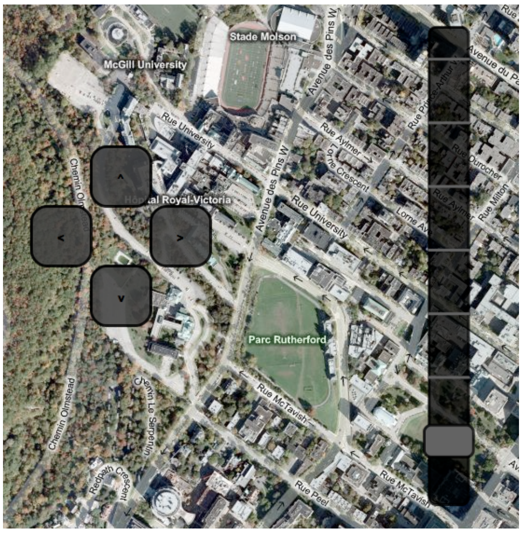


Figure 2.1: Four button arrows (left) corresponding to panning in the cardinal directions and a slider (right) for zoom.

2.1.2 Bezel Interface

The users establishes a pivot point by standing still for a short period. Placing one foot outside of the pivot area, indicated by a circle around the feet, pans in the direction specified as the vector from the pivot center to the outside foot. The participant can, at any time, exit the pivot area and establish a new one elsewhere.

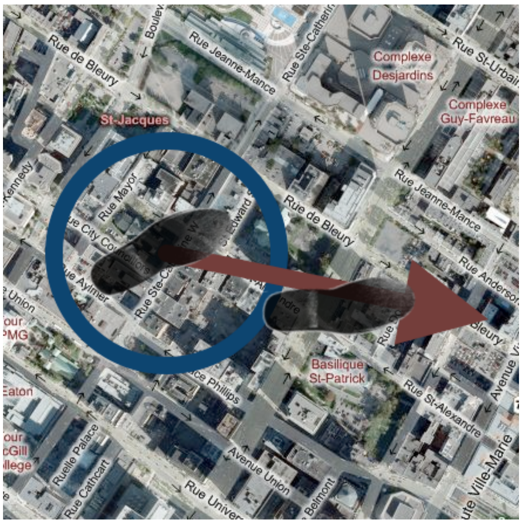


Figure 2.2: The first foot to remain still establishes the pivot point, surrounded by a blue circle. The second foot then specifies the vector, relative to the pivot, for panning.

2.1.3 Sliding Interface

As with the bezel interface, the users first establishes a pivot point by standing still for a short period. Then, by placing one foot outside the pivot area and using sliding or dragging gestures, akin to touch-screen scroll on an iPhone, the user can pan the map.

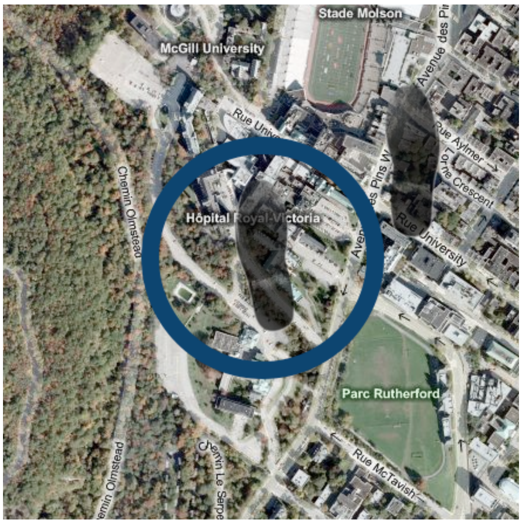


Figure 2.3: Here, the first foot establishes the pivot point, and the second foot slides or drags the display in a scrolling motion, with inertia.

2.1.4 Magic Tape Interface

Inspired by earlier work [2], this metaphor allows users to navigate freely in the center of the floor space, without altering the displayed map contents. However, when participants walk past the boundary region of the floor surface, the map pans in the direction designated by the user's position. The farther the user from the center, the greater the panning speed.

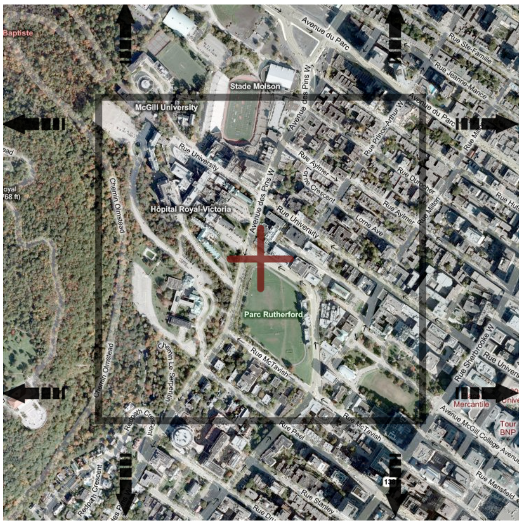


Figure 2.4: The rectangular outline indicates the magic tape boundary, beyond which, the user's footstep will result in panning in the direction formed by a vector from the center crosshair and the user's foot.

2.1.5 Crouch and Jump Zooming

In all interfaces except the classic interface, short body gestures can be used to control zooming. A curt "crouching" gesture zooms the map in, while a curt "jumping" gesture (raised onto the toes) zooms out.

Since such gestures have distinctive normal-force profiles trough time they can be matched to a reference signal. The use of a dynamic time warping algorithm for the matching in conjunction with preprocessing normalization allows for variations in gesture speed and amplitude from the user.

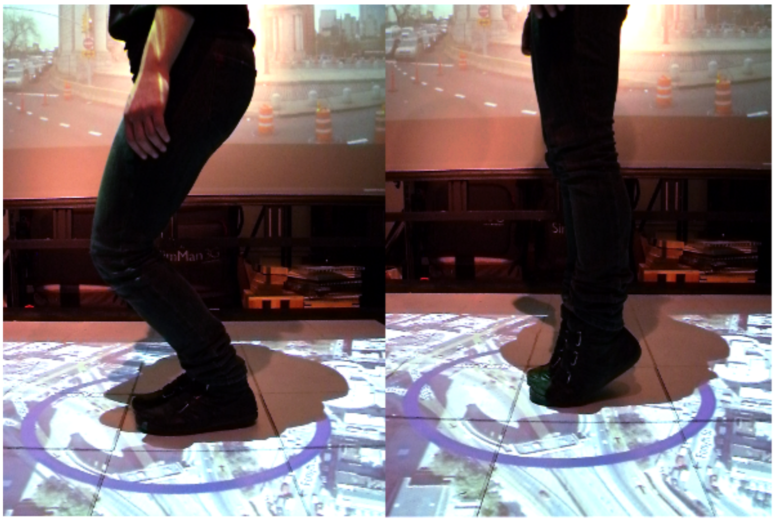


Figure 2.5: Crouching and jumping gestures for zoom control.

2.2 Gesture Recognition

The task of gesture recognition on a multi-touch foot interface can be broadly broken into two areas: the detection of spatial gestures and the detection of weight gestures.

2.2.1 Spatial Gestures

The simpler task, that of spatial gestures detection can be separated in the analysis of absolute gestures and relative gestures. In this context, any object contacting with the floor, such as a foot, is simply considered a punctual source of force at the barycenter of the object. Each punctual source can then be treated as a single cursor, or contact point, in the context of a multi-touch interface.

Absolute gestures are then fairly trivially recognized; the presence of a contact point at a particular spatial location can trigger an action or event sequence. Similarly a number of more complex graphical objects can function in an absolute spatial context. Knobs, sliders or switches can be controlled easily using absolute input.

For relative gestures, a similar basic approach is taken. As before, force input is treated as punctual and originating from the barycenter of an interacting object, i.e., a contact point. However, it must be possible to create a standard relative reference frame from which the contact points can be interpreted. This is accomplished by leveraging the physical constraints of standing, which, ignoring the possibility of jumping, require at least one foot to be in contact with the ground at all time. When the user first lifts a foot, we assume that this initiates a gesture sequence. The foot that remains in contact with the floor will then be constrained to its current position until the end of the gesture sequence. The contact point representing this foot provides a natural reference point, or *pivot*, for relative user gestures. If the user was instead walking or stepping rather than initiating a gesture, the pivot would be released almost immediately, thereby breaking the relative reference frame and any possible subsequent gesture sequence. Robust detection of the pivot rests on the simple assumption that there will always be a window, before the start of a gesture, when the pivot is the only user contact with the floor. This occurs when the gestural foot is raised in anticipation of the upcoming interaction. Once a pivot is established, the second, gestural foot can easily be used for various types of relative interaction. Objects positioned relative to the user, not the space, can be pressed and controlled as they follow the user in the space. Alternatively, the spacing of the pivot and the gestural contact point can denote a vector, for example, to indicate desired movement direction. Or, a relative dragging movement of the gestural foot can serve for analog input, such as zooming or volume, or indicate desired movement, such as the dragging and release gesture used to navigate virtual maps.

2.2.2 Weight Gestures

On the other hand, the analysis of weight and posture-centric gestures is significantly different. The idea is to recognize distinct sequences of whole body movements based on weight shifts and posture changes as unique gestures. Many such movements are characterized by distinctive transient force signal from the floor sensors, due to the reaction forces the user generates to move about. The gestural vocabulary in question is one of crouching, jumping, leaning and tapping. Such movements are accompanied by force impulses as the user lifts and moves his or her body weight. However, there are two obstacles to detection of these gestures. First, they can significantly vary in amplitude due both to the manner of execution and the strength and intensity displayed by the user. This issue is easily resolved through an initial normalization of the data, which also reduces the sensitivity of the system to noise. Secondly and more significantly, the timing of sub-steps comprising such movements may vary greatly between users, resulting in a signal that is distorted in time. For instance, one user's "curt jump" might consist of a quick rise to the toes followed immediately by a drop back down, while another user may pause briefly at the peak the rise, stand on tiptoe, and then slowly drop down. While both inputs will share a common distinctive shape, they are stretched and distorted in time relative to each other.

In order to compensate for this, a dynamic time warping algorithm (DTW) is used to match reference gestures to the input waveform. The system performs a continuous windowed DTW analysis of the force input from the sensor with respect to a number of reference force waveforms, generated beforehand from a training data set. Given an input function $i(t)$ and a reference waveform $r(t)$, the DTW algorithm builds a 2D graph such that every point $p(x,y) = |i(x) - r(y)|$. A perfectly diagonal path through the graph would travel at constant and equal time through both samples. However, curved paths can travel at variable speed through the samples, allowing for warping in the signal matching process. The sum of all the values along any monotonic and continuous path (moving forward in time only) represents the matching cost of the given path, with lower costs representing better matches.

We understand that the "start" of any gesture can be difficult to determine prior to analysis. This suggests that we should relax the typical DTW constraint of computation proceeding through each sample in its entirely, i.e., the matching path extending from $p(0,0)$ to $p(x_{max},y_{max})$, Instead, the matching algorithm might be allowed to enter anywhere along the bottom edge ($p(x_a,0)$) and exist anywhere along the top ($p(x_b,y_{max}),x_b > x_a$) to deal with the case where the input gesture was performed much faster than the reference. In this case, the algorithm remains constrained to traverse the entire reference sample, but is free to match only the subsection of the analysis window containing the input gesture. Once the path of least cost through the graph is computed, the similarity of the reference and input waveform can be evaluated with allowances for time warping. This allows users flexibility to vary the speed of different segments of the gesture, as long as the overall transient characteristics are still present.

While the implemented DTW algorithm is currently used only to detect curt rise/jump and drop/crouch gestures, it could easily be adapted to match more complex gestures, assuming sensors behave with sufficient linearity and the system is sufficiently trained. A gestural vocabulary might, for instance, include rocking gestures, rhythmic gestures such as distinctive tapping, or specialized gestures such as rolling onto the sides of the feet.

3 Probabilistic Tracking of Pedestrian Movements via In-Floor Force Sensing

3.1 Tracking Problem and Algorithm

Consider a dynamic system with states x_t and observations z_t, both indexed by time. A Bayesian filter probabilistically estimate at time t the state x_t by sequentially updating a belief distribution $\mathrm{Bel}(x_t)$ over the state space, defined by $\mathrm{Bel}(x_t) = p(x_t|z_t, z_{t-1}, z_{t-2}, ...) = p(x_t|z_{1:t})$. Assuming the states comprise a Markov process, the Belief distribution at each subsequent time step can be obtained, using Bayes' Theorem, in terms of the belief state at the prior time step $t-1$, the assumed motion model $p(x_t|x_{t-1})$, and the likelihood $p(z_t|x_t)$ of the newly acquired observation z_t given the state x_t:

$$\mathrm{Bel}(x_t) \propto p(z_t|x_t) \int p(x_t|x_{t-1}) \mathrm{Bel}(x_{t-1})\, dx_{t-1} \tag{3.1}$$

Bayesian filters can be distinguished, in part, by the form of the Belief distribution $\mathrm{Bel}(x_t)$ that is assumed. Here, we adopt a Monte-Carlo approach, in which $\mathrm{Bel}(x_t)$ is represented by a set of weighted samples, or particles, given by $S_t = \{(x_t^i, w_t^i), i = 1, 2, \dots N_s\}$. Our method uses the Sampling-Importance-Resampling (SIR) algorithm [5], described in Alg. 1, and schematically illustrated in Fig 3.1.

For each new observation, SIR re-weighs the set of particles depending on their likelihood, evolves them in time using the assumed dynamic model, and updates the particle set S_t by resampling based on these weights.

Specifically, the likelihood function $L(x_t) = p(z_t|x_t)$ is computed in two steps as follows.

1. Using the observation model $\mathbf{H}$, defined in Section 3.1.1, that maps states into the observation space, generate expected observations $z_t^* = \mathbf{H}(x_t)$.

2. Using the similarity measure $S(z, z') = p(z|z')$, defined in Section 3.1.2, compute the likelihood as $L(x_t) \equiv S(z_t^*, z_t)$.

As seen in Eq.3.2, each particle is weighted according to its likelihood while also considering a prior distribution $p(x_t)$. As described in Section 3.1.2, the prior distribution $p(x_t)$ is useful for encoding constraints on states x_t, thus leading to an efficient search of the state space.

The attributed weight is then used to resample the particle set (see Eq. 3.3). This step distinguishes SIR from SIS (Sampling Importance Sampling), and is meant to eliminate degeneracy of particles; by concentrating on high weighted particles to create a new uniformly distributed particle set, the extreme case where most particles have negligible weight is avoided. The opposite extreme, consisting of a single strong hypothesis, can be avoided by the inclusion of a roughening process. Roughening consists of the addition of random, zero-mean, normally distributed noise to all particles after resampling (Eq. 3.5), to allow for a thorough search of the solution space. Additionally, particles are propagated forward in time based on the assumed dynamic model (see Eq.3.4), which implicitly defines the motion probability $p(x_t|x_{t-1})$. Our dynamic model consists of two processes, accounting for the continuous and discrete aspects of the motion of interest. The dynamic model and roughening process are defined in detail in Section 3.1.3.

In our system for tracking via in-floor force measurements, the relevant variables consist of:

- Observations $\mathbf{z}_t$, consisting of a 12×12 array of force values, f_i.

- States, $\mathbf{x}_t$, describing kinematic lower-body poses, are 19-dimensional vectors: $\mathbf{x}_t = (\phi_{l,t}, \dot{\phi}_{l,t}, \dot{\phi}_{l,t-1}, \phi_{r,t}, \dot{\phi}_{r,t}, \dot{\phi}_{r,t-1}, \beta)$. They include planar midpoint coordinates $\mathbf{u}$ and orientations θ for each foot, where

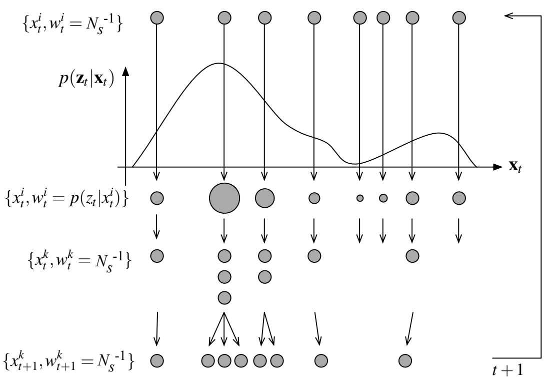

$\{x_t^i, w_t^i = N_s^{-1}\}$
$p(\mathbf{z}_t|\mathbf{x}_t)$
$\mathbf{x}_t$
$\{x_t^i, w_t^i = p(z_t|x_t^i)\}$
$\{x_t^k, w_t^k = N_s^{-1}\}$
$\{x_{t+1}^k, w_{t+1}^k = N_s^{-1}\}$
$t+1$

Figure 3.1: The SIR particle filter algorithm.

Algorithm 1 Sampling Importance Resampling

```
Initialize particles randomly: $S_0 \sim N(\mu_S, \sigma_S)$
while $t > 0$ do
    Observe $z_t$.
    for $i = 1$ to $N_s$ do
        Likelihood: $p(z_t|x_t^i) = S(z_t, H(x_t))$
        Weight: $w_t^i = p(x_t^i) \times p(z_t|x_t^i)$    (3.2)
    end for
    for $i = 1$ to $N_s$ do
        Normalize: $W = \sum_i w_t^i, w_t^i \leftarrow W^{-1} w_t^i$
        Resample: $x_t^i \sim p(x_t^{i*}|w_t^i), w_{t+1}^i = N_s^{-1}$    (3.3)
        Draw: $x_{t+1}^i \sim p(x_{t+1}|x_t^i)$    (3.4)
        Roughen: $x_{t+1}^i \leftarrow x_{t+1}^i + \eta_{t+1}, \eta \sim N(0, \sigma_x)$    (3.5)
    end for
    $S_{t+1} = \{(x_{t+1}^i, w_{t+1}^i)\}_{i=1}^{N_s}$
end while
```

$\phi_l = (\mathbf{u}_l, \theta_l)$ and likewise for ϕ_r, along with first time derivatives. The state $\mathbf{x}_t$ also includes a binary-valued vector $\beta = (\beta_l, \beta_r)$, implemented as a quaternary variable, which indicates the foot-floor contact condition ($\beta_i = 1$ if there is contact) for the left and right feet.

Figure 3.2 illustrates this state description within a skeletal model. The algorithm definition is completed by specifying the observation model $\mathbf{z}_t = \mathbf{H}(\mathbf{x}_t)$, the likelihood model $p(\mathbf{z}_t|\mathbf{x}_t)$, and the motion model, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$.

3.1.1 Observation model

We model expected observations $\mathbf{H}(\mathbf{x}_t)$ for a state $\mathbf{x}_t$ by simulating the mechanical forces associated with a pose. Ignoring shear forces, foot-tile contact results in a normal pressure distribution $p(\mathbf{u})$, where $\mathbf{u} = (u, v)$ are 2D coordinates on the floor. The pressure distribution on a tile is conveniently summarized by a contact centroid $p_c = (\mathbf{u}_c, F)$, where $F = \int d\mathbf{u}\, p(\mathbf{u})$ is the net normal force and $\mathbf{u}_c$ is the pressure centroid. A normal force with magnitude F, applied at $\mathbf{u}_c$, would give rise to the same force measurements f_i as $p(\mathbf{u})$ [12, 1] (Fig 3.3). Our observation model associates a pose $\mathbf{x}_t$ to a set of contact centroids, $p_{c,j}, j = 1, 2, \ldots, N_c$. One centroid is placed on each tile that a foot pose is determined to be in contact with ($\beta_i > 0$), and the total weight F of the user is partitioned among these centroids. The placement of a contact centroid is determined by the average

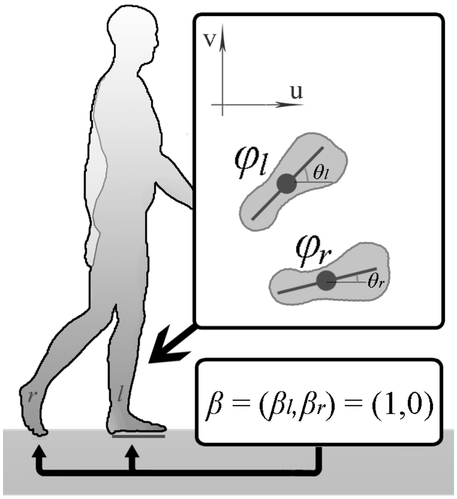


Figure 3.2: State description of lower body poses: $\mathbf{x}_t$ defines feet and foot-floor contact.

position of foot to tile contact. Expected sensor readings f_i are obtained from the static equilibrium equations for each tile. Each pressure centroid p_c yields a contribution $f_{i,c} = d_i^{-1}(\sum_{j=0}^{4} d_j^{-1})^{-1}$, where $d_j = |\mathbf{u}_c - \mathbf{u}_j|$, and $\mathbf{u}_j$ is the sensor location. In this way, we predict observed force values $\mathbf{z}_t^* = \{f_{i,t}\}$ for a state $\mathbf{x}_t$.

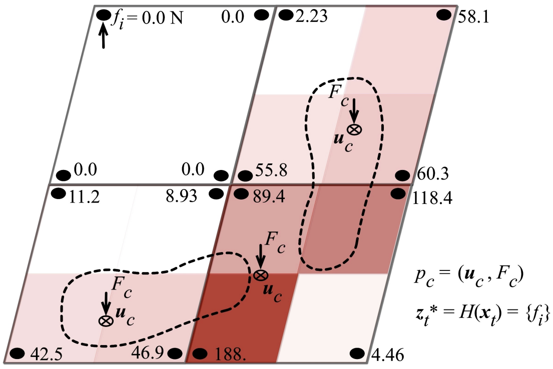


Figure 3.3: The observation model $H(\mathbf{x}_t)$ maps a pose $\mathbf{x}_t$ (illustrated here as two feet) to a set of contact centroids $(\mathbf{u}_c)$. These are then converted to force sensor readings $\mathbf{z}_t = \{f_i\}$. Expected force observations are illustrated as grids, with each quadrant intensity proportional to corresponding force sensor value (given in Newtons).

3.1.2 Likelihood model

The likelihood function $L(\mathbf{x}_t) = p(\mathbf{z}_t|\mathbf{x}_t)$ describes the probability of observing force values $\mathbf{z}_t$ given the state $\mathbf{x}_t$. As described above, we use the observation model to generate expected observations, $\mathbf{z}_t^* = \mathbf{H}(\mathbf{x}_t)$. The likelihood function is then defined in terms of a normalized similarity measure $p(\mathbf{z}_t|\mathbf{x}_t) = S(\mathbf{z}_t^*, \mathbf{z}_t)$ between true force pattern observations $\mathbf{z}_t$ and expected force observations $\mathbf{z}_t^*$.

Similarity measure S The similarity measure $S(\mathbf{z}, \mathbf{z}') = p(\mathbf{z}|\mathbf{z}')$ models the probability of observing $\mathbf{z}$ if the true observation is $\mathbf{z}'$. Conventional similarity measures make use of metrics such as Euclidean or Mahalanobis distance functions. However, since $\mathbf{H}$ is a high-dimensional discontinuous map from states $\mathbf{x}_t$ to observations $\mathbf{z}_t$, these measures cannot properly gauge similarity between observation vectors $\mathbf{z}_t$, $\mathbf{z}_t^* = \mathbf{H}(\mathbf{x}_t)$. Moreover, the force observations are comparatively sparse, with most values being zero. As an alternative, we compute pair-wise similarity between such patterns, based on a measure of their area of overlap. Specifically, we employ a similarity measure that has proved useful in tracking via binary image masks [9], computing $S(\mathbf{z}^*, \mathbf{z})$ as the

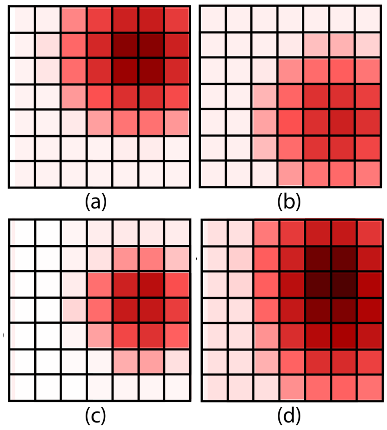


Figure 3.4: 2D pressure distributions for (a) $\mathbf{z}_t$, (b) $\mathbf{z}_t^*$, (c) $\mathbf{z}_t \cap \mathbf{z}_t^*$, (d) $\mathbf{z}_t \cup \mathbf{z}_t^*$. Pressure distributions are illustrated as grids, with each quadrant intensity proportional to corresponding force sensor value. The intersection ($\cap$) and union ($\cup$) of force observations are used in computing similarity $S(\mathbf{z}_t, \mathbf{z}_t^*)$.

relative area of overlap between the true and expected 2D pressure distributions,

$$S(\mathbf{z}_t^*, \mathbf{z}_t) = \frac{\cap(\mathbf{z}_t^*, \mathbf{z}_t)}{\cup(\mathbf{z}_t^*, \mathbf{z}_t)} = \frac{1}{N_z} \sum_{i=1}^{N_z} \frac{\min(\mathbf{z}_t(i), \mathbf{z}_t^*(i))}{\max(\mathbf{z}_t(i), \mathbf{z}_t^*(i))}$$

Fig. 3.4 illustrates conceptual examples of observations $\mathbf{z}_t$, $\mathbf{z}_t^*$ as 2D pressure distributions, as well as their intersection $\cap(\mathbf{z}_t^*, \mathbf{z}_t)$ and union $\cup(\mathbf{z}_t^*, \mathbf{z}_t)$.

We note that the average overlap of these resulting pressure distributions is an effective metric for capturing similarities between force observations as a probability:

$$\begin{array}{rcl} S(\mathbf{z}, \mathbf{z}) & = & 1 \\ \mathbf{z}_1 \neq \mathbf{z}_2 & \Rightarrow & 0 < S(\mathbf{z}_1, \mathbf{z}_2) < 1 \\ S(\mathbf{z}_1, \mathbf{z}_2) & = & S(\mathbf{z}_2, \mathbf{z}_1) \end{array} \tag{3.6}$$

Postural constraints The likelihood model is modified to encode human postural constraints, via a prior distribution $p(\mathbf{x}_t)$. The latter is defined to consist of a set of independent postural priors for human walking, in the form of univariate Gaussian distributions $N(\mu_\varsigma, \sigma_\varsigma)$, $N(\mu_\Theta, \sigma_\Theta)$ over stance width $\varsigma = \|\mathbf{u}_l - \mathbf{u}_r\|$ and relative orientation $\Theta = |\theta_l - \theta_r|$ respectively. The postural prior is introduced as the product $p(\mathbf{x}_t) = N(\varsigma; \mu_\varsigma, \sigma_\varsigma) N(\Theta; \mu_\Theta, \sigma_\Theta)$. The prior distribution $p(\mathbf{x}_t)$ is applied when computing the particle weights from the likelihood L (see Eq. 3.2).

3.1.3 Dynamics Model

We model the movements of the lower body of a walker via the motion probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The state x_t consists of continuous configuration variables ϕ_i and discrete contact variables β_i. We therefore approximate foot motion in a hybrid (stochastic) framework, incorporating continuous, linear movements of the limb coupled to discrete state transitions reflecting changes of the foot-floor contact conditions.

Continuous, linear dynamics The state representing the left or right foot is denoted respectively by a vector $\phi_i = (\mathbf{u}_i, \theta_i)$ giving the midpoint and orientation of either the left ($i = l$) or right ($i = r$) foot at time t (Fig. 3.2. The dynamics used in our algorithm can be described by the following linear, discrete time system:

$$\phi_{i,t} = \phi_{i,t-1} + \beta_i \Gamma_i (\dot{\phi}_{t-1} dt + \eta_t^0) \tag{3.7}$$

$$\dot{\phi}_{i,t} = \beta_i [\alpha \dot{\phi}_{i,t-1} + (1 - \alpha)(\dot{\phi}_{i,t-2} + \eta_{t-1}^1)] \tag{3.8}$$

The parameter β_i is the binary contact variable for foot i. Thus, ϕ_i is constant when there is contact ($\beta_i = 0$) and otherwise drifts, with position and velocity driven by additive Gaussian noise processes η_t^0 or η_t^1, where $\eta \sim N(0, \Sigma)$. For efficiency, we parametrize drift via a single noise process, defining $\eta \equiv \eta_t^0 = \eta_t^1 dt$ for all t. The noise covariance Σ is a 3×3 diagonal matrix with diagonal entries $\sigma_u, \sigma_v, \sigma_\theta$. To mimic walking, velocity drift in the direction that the foot is oriented is assumed to be larger. This non-isotropic drift is implemented through the factor Γ_i, a diagonal matrix with entries $(\cos\theta_i(\gamma + \sqrt{1-\gamma^2}), \sin\theta_i(\gamma - \sqrt{1-\gamma^2}), 1)$, where $0 < \gamma < 1$ is a dimensionless scalar defining the longitudinal bias. α is a dimensionless scalar defining the velocity noise mixing rate. It approximates the dynamics of a free foot during walking by means of a saturating linear drift velocity.

Although this continuous dynamic model violates the Markov assumption made in Eq.3.1, we note that there exists a Markov representation of this system, via the following change of variables:

$$\Phi_t = \begin{bmatrix} \dot{\phi}_t \\ \dot{\phi}_{t-1} \end{bmatrix} = \begin{bmatrix} \alpha & 1-\alpha \\ 1 & 0 \end{bmatrix} \Phi_{t-1} \tag{3.9}$$

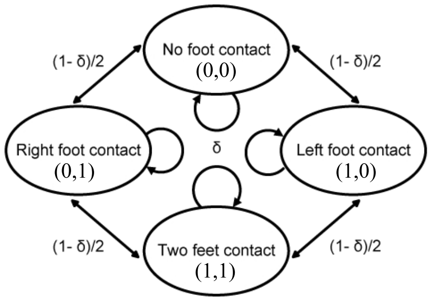


Figure 3.5: Stochastic state transition diagram approximating stepping motion.

Roughening As presented in (Alg. 1), noise $\eta \sim N(0, \sigma_x)$ is added to the continuous state components ϕ_i at each SIR step in order to avoid particle degeneracy.

Discrete state transition model The dynamic model includes discrete transitions from the foot-floor contact states $\beta = (\beta_l, \beta_r)$, where $\beta = 0$ or 1, via the stochastic process, shown in Figure 3.5. Despite its simplicity, this model is effective in approximating discrete stepping motion. δ is an empirically determined probability of no change in contact state β. All remaining transitions are symmetric, with transition probability $\frac{1-\delta}{2}$.

3.2 Experiment and Results

The system described above was evaluated by measuring the absolute positions of the feet of pedestrians using data acquired synchronously via motion capture (Vicon Motion Systems). Reflective markers were attached to the walkers' shoes, providing an accurate estimate of 3D foot positions. Five recordings of walking sequences between 5.7 and 12.4 seconds in length were acquired via the apparatus described in previous reports. Synchronous motion capture and force data were recorded. Errors were computed based on maximum a posteriori (MAP) foot position estimates obtained from the tracking algorithm.

Figure 3.6 shows the state estimates and force observations at four stages of the walking sequence: the initial particle set has high variance, but is quickly narrowed down to a few hypotheses which are evolved based on our motion model. Figure 3.7 shows the resulting motion trajectories and errors for the two foot locations, in planar coordinates. Average RMS error values are reported in Table 3.1. The experimental parameters are given in Table 3.2. Position errors during foot-floor contact are found to be slightly less on average than when a foot is not in contact with the floor. Temporal alignment mismatches were found to have a large effect, so we also performed a windowed error calculation in which an acceptable time shift of 10 samples (at 20

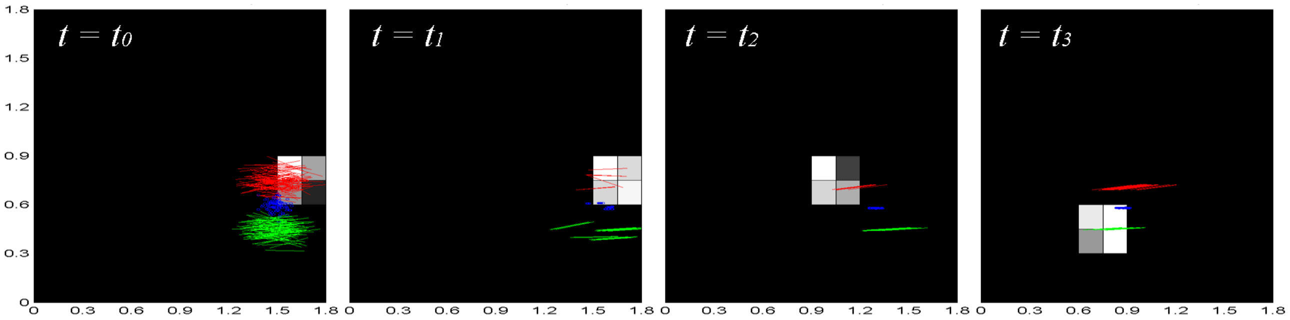


Figure 3.6: Force observations and pose estimates at 4 stages of the sample walking sequence. Observations are shown as 2D pressure distributions, with quadrant intensity proportional to force sensor readings. Poses are illustrated as green and red lines, corresponding to a top view of left and right feet estimates respectively, with blue circles corresponding to inferred center of mass.

Foot	**Error (m)**	**Windowed error (m)**
Right (no contact)	0.1901	0.1245
Left (no contact)	0.1649	0.0336
Average (no contact)	**0.1775**	**0.0791**
Right (contact)	0.2025	0.1058
Left (contact)	0.1406	0.0306
Average (contact)	**0.1716**	**0.0682**
Right (all)	0.1982	0.1122
Left (all)	0.1490	0.0316
Average (all)	**0.1736**	**0.0719**

Table 3.1: RMS Position Error

Hz) was permitted. This greatly reduced RMS position errors (see Table 3.1), suggesting that system tracking performance may be most acceptable in situations in which temporal accuracy is not important. Tracking performance in more temporally demanding settings might be greatly improved if a better alignment can be achieved. Video documentation of these results is provided in the supplementary material, and available online at `http://www.cim.mcgill.ca/~rishi/video.swf`.

In addition to position estimates, this system provides continuous labels identifying the walker's right and left feet. During tracking of the sequences used for evaluation, left and right feet were continuously and coherently identified with 100% accuracy. The capability of this system to maintain and propagate these labels may be useful for applications including floor-based touch screen user interfaces, where it may be desirable to assign each foot a different functional operation, or to render a different response to left and right foot [12].

3.3 Conclusion

This work presented Bayesian filtering techniques to track the lower body pose of a pedestrian from foot-floor interaction forces acquired via a coarse array of in-floor force sensors. The system achieves continuous and labeled tracking of the lower limbs of a walker via a coarse sensor array, by combining prior knowledge about the mechanical structure of the interface and a simple, but consistent, model of the dynamics of the feet during walking. In the experiment described above, our system never confused the left and right feet of the walker, and was able to track the locations of each with an average resolution on the order of 15 cm, and with improved resolution when feet are in contact with the surface. Potential applications of these techniques include tracking in smart environments in which motion capture is impractical (due to occlusion or other factors), and to interaction with distributed, floor-based touch surface interfaces for the feet [12, 10].

Despite the promising nature of the results presented above, further experiments are needed to evaluate the

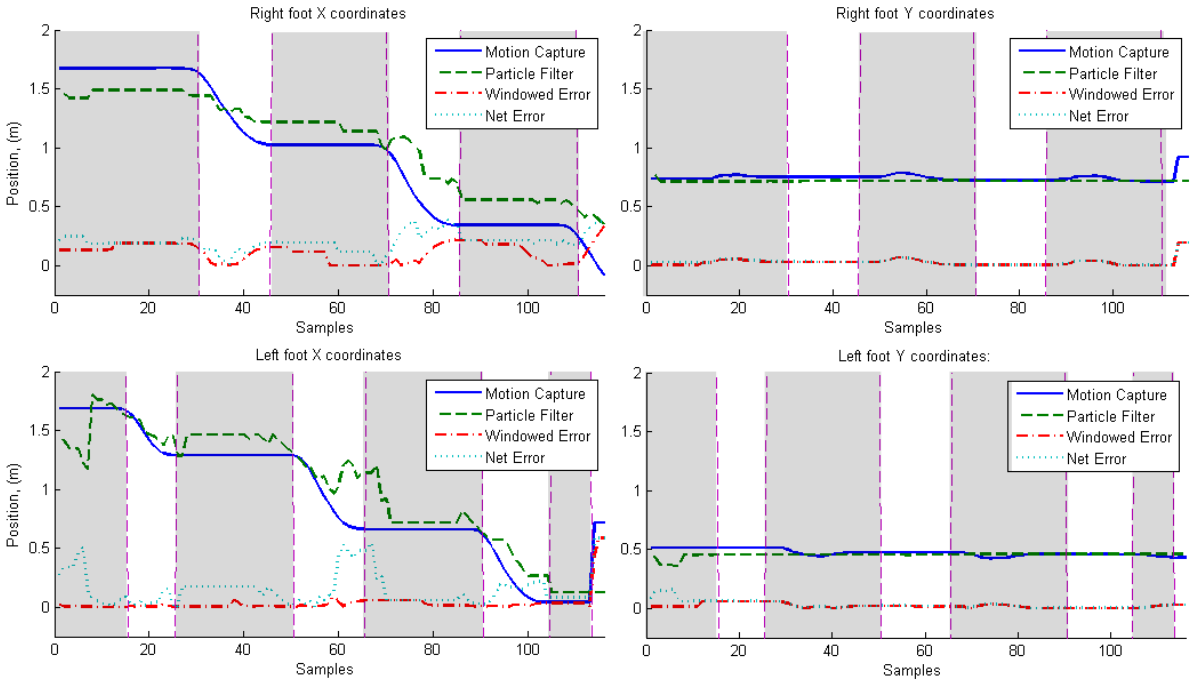


Figure 3.7: Sample results for a pedestrian crossing the apparatus, comparing measurements from motion capture data and estimates obtained from the particle filter. Results are shown for both feet ϕ_l, ϕ_r. The shaded domains illustrate contact states, with grey and white corresponding to $\beta_i = 1$, $\beta_i = 0$ respectively.

quality of the tracking. In addition, it is clear that the system itself can be improved in several respects. A higher-density sensor network would improve position estimates during contact, albeit at greater cost. A model for the non-contact portion of walking movement that is more sophisticated than the random drift model used in our system could significantly improve estimates in foot tracking when a foot is not in contact with the floor. The incorporation of additional prior knowledge about the kinematic constraints on lower limb positions during walking would also be expected to contribute improvements. In ongoing work, we are exploring the possibilities for optimally fusing information from in-floor force sensors with motion capture or other video sensors, in order to resolve data loss due to camera occlusion, or to provide contact forces and timing information that cannot be accurately estimated from video.

SIR Algorithm	
Number of particles	$N_s = 200$
Initial Noise	$\sigma_S = 4.572$ cm
Roughening Position Noise	$\sigma_x = 0.003048$ cm/rad
Likelihood Model	
Postural prior stance size	$(\mu_\varsigma, \sigma_\varsigma) = (30.48, 30.48)$ cm
Postural prior stance angle	$(\mu_\Theta, \sigma_\Theta) = (0, \pi/2)$ rad
Dynamic Model	
Additive Position Noise	$\sigma_{u,v} = 0.3048$ cm
Additive Angle Noise	$\sigma_\theta = 0.003048$ rad
Longitudinal bias	$\gamma = 0.9$
Velocity mixing rate	$\alpha = 0.55$
Probability of no β change	$\delta = 0.55$

Table 3.2: Experimental parameters

4 Modeling of the User and the Haptic Device

Transparency in haptics is defined by the capacity to cover the range of our perceptual capacity, by producing high wrenches and vibration amplitudes (tactons and icons) [7] in a wide frequency bandwidth and with high fidelity. For a vibrotactile display, the coupling effects of user impendance with the dynamics of the device can impede its transparency. In the case of a floor display, the impact of the foot on the device can confuse the identification of ground material and therefore should be reduced.

A previous study shown that the structural properties of the tile could be compensated for obtaining a flat response in the desired frequency bandwidth [11]. A 14-order FIR filter was disigned for this static compensation. By inverting this filter, and applying it to all subsequent outputs of the actuator, we now have a significantly corrected frequency response. The main issue is then to find a stable inverse filter. To improve the results obtained with this filter and avoid some known issues, it's also possible to used uniform and warped low delay filter-banks equalizer such that only the phase delay of a second order subfilter will be effectively measured between the input and the output of the vibrotactile response. However, this static compensation is not enough and this research is exploring the effect of a dynamic compensation. The dynamic compensation presented here is an approximation and thus using an adaptive equalizer technic like with a filter bank or with DFT/IDFT could be another solution like in speach enhancement application [6].

The first component of the compensation is thus a filter bank equalizer which adjust the flatness of the haptic device in a static configuration. The filter bank adptively adjust the gain of each subband filter in function of the static configuration of the tile. The second component of this frequency correction system lies in dynamically correcting the frequency response of the moving user. The main hurdle here is to be able to simplify the computations involved in order to minimize hardware costs while still maintaining a reasonable level of fidelity. In particular, we could do away entirely with the realtime correction algorithm and instead provide a preset equalizer for that specific user based on his or her precalculated parameters. Thus, by studying a variety of individuals and building up a sizable database of mappings from parameters to corresponding correction equalizers, the entire correction system could function via a simple lookup table, interpolating any parameter sets in between those from known users similar to the previous work presented in [4].

One area of interest lies in accurately determining the position of the tile from accelerometer data. One subproject therefore involves investigating the efficacy of different integration techniques for noisy signals, and heuristically estimating the tile's pose to within a certain degree of accuracy. Furthermore, force sensing resistors could be used in refining the accuracy of the tile's pose.

5 Efficient Acquisition of Force Data in Interactive Shoe Designs (UNIVR)

While developing an interaction design concept for the acquisition of force data capable of providing sufficient accuracy along with low latency and low encumbrance and obtrusion, we soon realized that the above issues, if taken altogether, had still been substantially left unanswered by existing mobile solutions which could be potentially applied to our active shoe prototype, without explosion of the costs.

For this reason, we have devoted some project resources for investigating on that direction. By moving from the prototype developed during the first year of the project (refer to Deliverable 3.1), a new four-channel sensing system has been proposed for the capture of force data from the feet during walking tasks. The system now solves general issues of latency of the response, accuracy of the data, and robustness of the transmission of digital signals to the host computer. By adopting widely used force sensing (Interlink) and analog-to-digital conversion and pre-processing (Arduino) components, the proposed system is also expected to raise interest among a broader interaction design audience.

Two main problems needed to be dealt with during the development of the system. The former consists of the electro-mechanical characterization of the force sensing resistors. The latter asked to access to some low-level resources of the Arduino board. An extensive discussion on the realization of the system is provided in [3], also present in the project website private area.

5.1 Characterization of the FSR

In our previous prototype, a reasonable balance between front and rear sensitivity had been achieved by placing an Interlink Force Sensing Resistor (FSR from now on) model 400 under the heel and a 402 under the toe. While realizing the new prototype found that the model 400 reaches mechanical saturation much faster than the 402. This time we have characterized the nonlinear behavior of the overall system, caused by the electrical coupling between the FSR and the analog-to-digital converter (ADC) in the Atmel microcontroller on-board the Arduino. The measurement setup included a hand-made wooden press mounted onto a weighing scale.

Each FSR was connected to the Arduino analog input through a voltage divider circuit. Thanks to this overall setup, several sets of curves have been obtained for each sensor mapping force into ADC values, for changing values of the resistance R1 in the voltage divider. In practice, each curve represents the relationship between the applied force and the ADC output (ranging from 0 to 1023) for a specific value of the voltage divider resistance. Fig. 5.1 displays the corresponding plots, respectively for sensors model 400 (a) and 402 (b). Their inspection shows that the system is more linear when using the model 402.

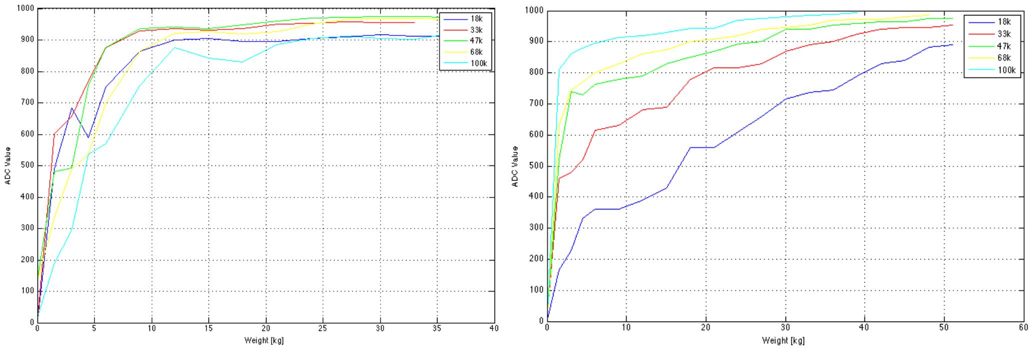


Figure 5.1: Force/ADC maps for changing values of the voltage divider resistance.

5.2 Firmware

To guarantee continuous interaction, the cheap hardware onboard the Arduino must perform stable analog-to-digital conversion together with low-latency transmission of the digital signal toward the host.

Both the polling sequence and the transmission routine of the previous prototype had been written using functions (such as `analogRead` and `Serial.print`) and procedure calls provided by the Arduino SDK. The use of this framework allows to write code very easily, thanks to the good level of abstraction provided by the embedded C++ API. Unfortunately the compiled program is in general not efficient, especially if running on applications requiring a constant low latency.

Our solution consists in a program that makes use of API calls and lower level AVR Libc instructions together. Furthermore, a custom transmission protocol has been implemented which transmits binary data instead of ASCII values, resulting in 64 instead of 160 bit messages meanwhile carrying the same information. In conclusion, a linear relationship between the transmission rate and sampling frequency could be figured out. This relationship is summarized in Table 5.1.

TXrate [baud]	9600	19200	38400	57600	115200
T_s [ms]	2.1	1.04	0.52	0.36	0.17
F_s [Hz]	476	961	1923	2778	5882

Table 5.1: Relationship between sampling frequency (F_s) and transmission rate (TXrate) using the custom firmware.

In our application, involving four channels, the sampling frequency per channel is $F_s = 5882/4 \approx 1470$ Hz. Considering that the FSR's have a response time of about 2 ms, a latency that is certainly smaller than any human response to psychophysical cue changes, the obtained sampling frequency is well above twice as much the Nyquist limit of $1/0.002 = 500$ Hz.

Fig. 5.2 shows some plots of force signals recorded using the proposed system. These plots provide evidence of a substantial absence of noise and jitter in the data. If perhaps not accurate enough for applications where extremely high resolution is mandatory, such data are certainly suitable for most interaction design applications.

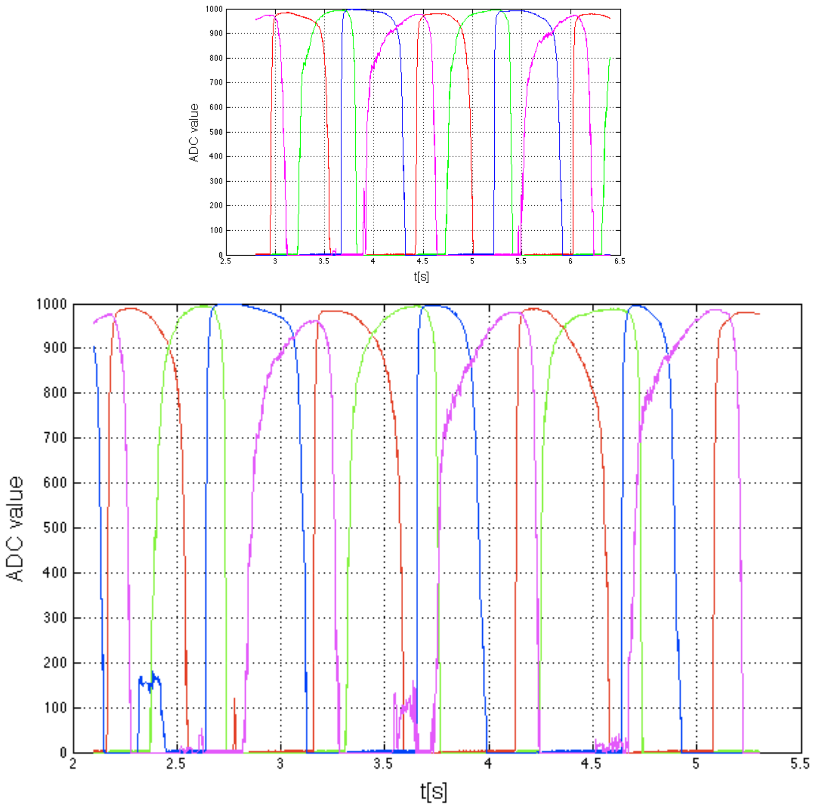


Figure 5.2: Force plots. Red = left heel, green = left toe, blue = right heel, magenta = right toe.

6 Acoustically-Sensitive Pavements (UNIVR)

In this chapter, we discuss further developments of the sensitive pavement system introduced in Deliverable 3.1. The system performs an analysis of the mechanical waves generated by the interaction of walkers' feet with the pavement and provides information about the position of the steps.

One of the appealing aspects of this approach is that it exploits the natural, pervasive and costless sensing-network embedded in any object due to its solid state and its elastic properties that allows the propagation of mechanical waves. Furthermore, the system is portable and, in principle, it can be applied to any pavement and any walker wearing sufficiently rigid ordinary shoes.

As widely discussed in Deliverable 3.1, modelization of wave propagation in-air and localization of sound sources by means of wave capturing and analysis is a well established field based on microphone array signal analysis techniques. On the other side, in-solid wave propagation is much more cumbersome due to the occurrence of complex phenomena and dispersive propagation. For this reason, we decided to adopt the Time Reversal (TR) method illustrated in detail in Deliverable 3.1.

We provide the results of the measurements obtained through a new arrangement of the sensing accelerometers and a validation of the system for step position detections outside the sampled monitored area. The latter aspect is particularly important, when it is necessary to distinguish between a monitored area and an external area (the rest of the world). In fact, in the most general case of an ordinary pavement, there is no possibility to isolate an area from the rest of the pavement or even from a contiguous space (the next room), where people could accidentally walk for whatever reason.

6.1 Methodology and application scenario

As already discussed in Deliverable 3.1, the complexity and non-linear behavior of in-solid wave propagation can be effectively exploited. In fact, the continuously changing waveforms supply an univocal characterization of an event at any other point of the medium, where the propagating wave is captured. This is the strategy adopted in the Time Reversal (TR) method illustrated in Deliverable 3.1. The method is based on the principle that the impulse response in a chaotic cavity is unique for a given source location. In other words, a wave propagating in a chaotic cavity contains the memory of its source location. In our case, the matter is to decide how many "footprints" on the pavement have to be considered in order to be able to evaluate with a sufficient accuracy the position of a footstep, i.e. of the acoustic source. Operatively speaking, a cross-correlation test is performed. As already argued in Deliverable 3.1 the TR method has many advantages crucial for our problem, such as that it is material-independent, not influenced by the homogeneity of the medium. Also, it is scalable according to the kind of excitation (type of shoes wore by the user) and pavement characteristics. In fact, in optimal conditions, i.e. with impulse-like steps and low reflection effects within the medium, the method could work with a single sensor. Furthermore it is portable, since the set of accelerometers can be applied to any pavement, even in presence of steps and different levels.

As a simple but general test-bench application, a basic scenario has been chosen: a boundary has been established subdividing the pavement into two areas, one of which is forbidden. The sensitive (monitored) area is within the side where the user is allowed to walk. Only a small part of the pavement near the boundary is monitored (in our case an area of 2.5 by 3 meters). When a user approaches the boundary, the system reacts by means of warning signals with a higher level of urgency as the walker gets closer to the boundary.

6.2 Pavement setup

After numerous testing of different configurations of accelerometer location and different sampling points, we ended up with the setup depicted in Figure 6.1. In particular, the eight accelerometers are situated in proximity of the most critical "alarm" level, just beyond the boundary. Also, they approximate an equidistant placement of the sensors with respect to the center of the sensitive area. This is the configuration that till now provided the best detection results, much better, for example, than the results obtained by means of the array configuration illustrated in Deliverable 3.1. The sampling points allowed a tiling of the sensitive area of the kind represented in Figure 6.2. The warning levels were subdivided into 3 increasing degrees, corresponding to lines 6-5, 4-3 and 2-1, respectively. A number of different paths starting outside the sensitive area, approaching it and crossing it up to the border line were tested. Figures 6.3 and 6.4 show a couple of examples of such free walk towards and across the sensitive pavement. Considering the method, the response of the system to footsteps outside the sensitive area was an incognita. Fortunately, the footsteps outside the sensitive area are associated by the localization system to the nearest part of the sensitive area (and not, for example to some inner position), meaning that there is a continuity in the deformation of the waves due to the dispersion. This is crucial as well for any detection of footsteps occurring within a tile and not exactly on one of the corners of the tiles, where the sampling of the "footprints" takes place.

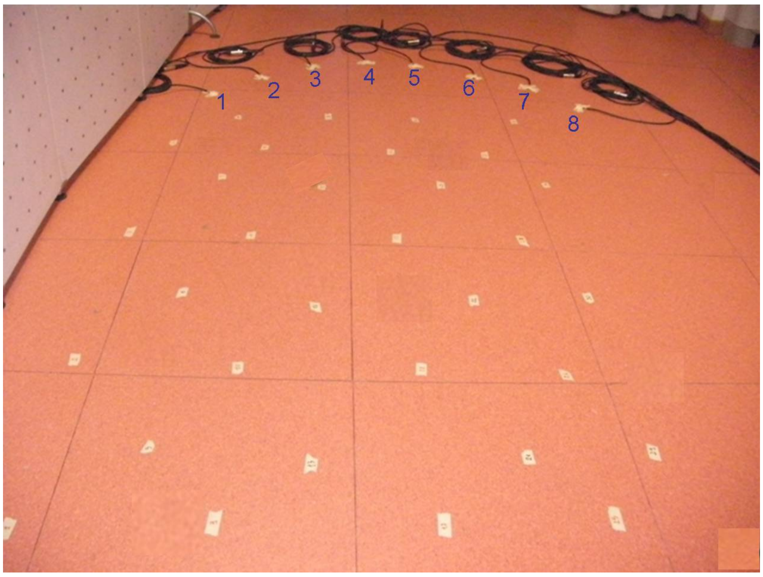


Figure 6.1: Floor setup.

6.3 Experimental results

The adopted position resolution corresponds to the size of the tiles shown in Figure 6.2. For each tile, the median and the mean over the cross-correlation values of the four "footprints" of the corners of the tile are calculated. Furthermore, the median value over all of the eight accelerometers is considered. Also, the operations on the tiles and on the accelerometers are commuted and, playing around with the non-linearity of the median operator, we compute four different estimations of the step localization. An example of estimation, based on the median over the eight accelerometers of the means over the four corners of each tile is depicted in Figure 6.5. The first four steps are outside the monitored area. The last three are inside the area and show as the walker approach the border line in the left corner of the sensitive area. In general, the localization of the line of the tile matrix is accurate up to plus/minus one line, which means, for example, that a step in the last line (line 1) was always attributed to the maximum warning level, associated to levels 1 and 2 (as already mentioned, the warning levels were finally ranked into three increasing degrees corresponding to lines 6-5, 4-3 and 2-1, respectively).

Presently, we are working on a real-time version of the algorithms, in order to achieve a system that can be

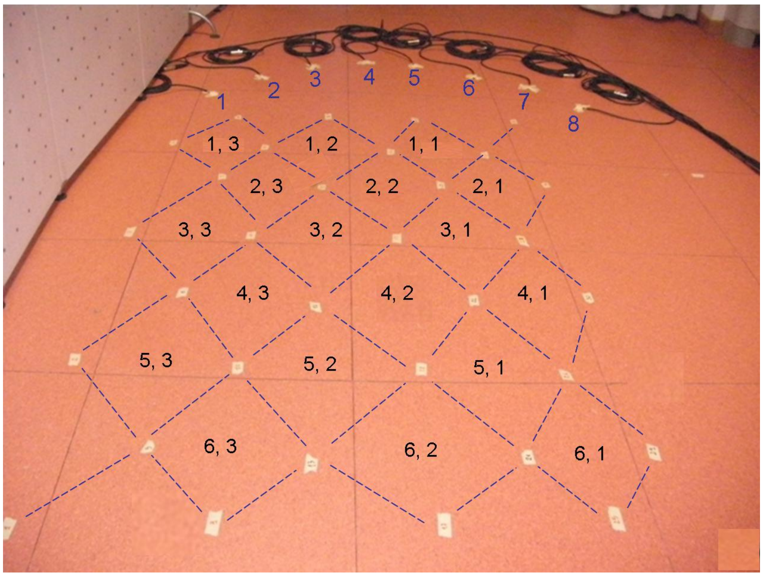


Figure 6.2: Floor tiling. The 8 accelerometers arrangement and the 32 sampling position of the“footprints”.

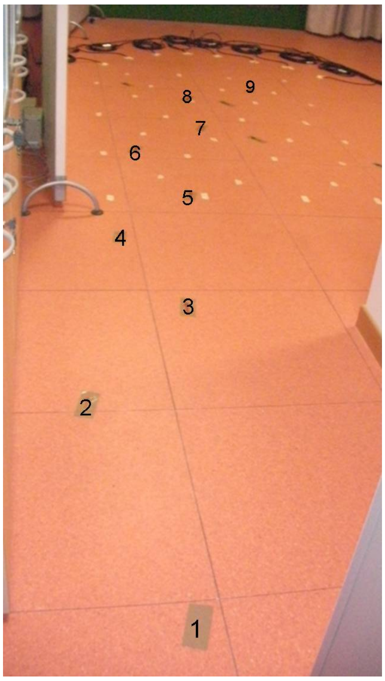


Figure 6.3: An example of free walk.

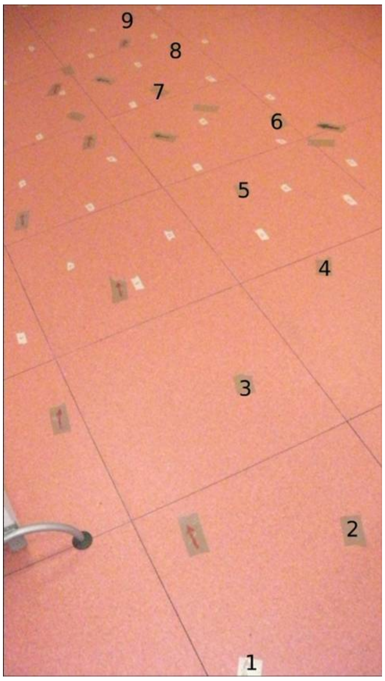
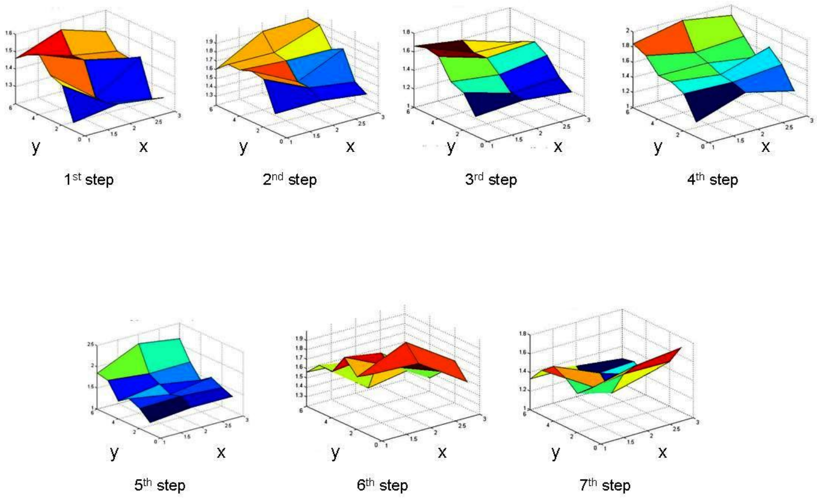


Figure 6.4: A second example of free walk.



1st step

2nd step

3rd step

4th step

5th step

6th step

7th step

Figure 6.5: Detection results for a free walk approaching the sensitive area from the bottom-right corner - tile (6,1) in Figure 6.2 - and crossing it diagonally towards the upper-left corner - tile (1,3)in Figure 6.2 - . Over ten repetitions of the path the last step near the border was always classified as line 1 or 2 and, thus, the highest degree of alarm

tested and employed in real-life scenarios. Also, the real-time implementation is necessary in order to proceed to massive testing and a more robust validation of the system. In particular, we are evaluating and optimizing the computational charge of the algorithms, in order to implement the algorithm as a Pure Data external, so that to exploit the facilities of that development environment for audio applications.

References

[1] A. Bicchi, J.K. Salisbury, and D.L. Brock. Contact sensing from force measurements. *The International Journal of Robotics Research*, 12(3):249, 1993.

[2] Gabriel Cirio, Maud Marchal, Tony Regia-Corte, and Anatole Lécuyer. The magic barrier tape: a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace. In *VRST '09: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 155–162, New York, NY, USA, 2009. ACM.

[3] M. Civolani, F. Fontana, and S. Papetti. Efficient acquisition of force data in interactive shoe designs. In *Proc. 5th Int. Haptic and Auditory Interaction Design Workshop*, pages 129–138, September 2010.

[4] Jonathan Fiene, Katherine J. Kuchenbecker, and Gunter Niemeyer. Event-based haptic tapping with grip force compensation. In *14th Symposium on Haptics Interfaces for Virtual Environment and Teleoperator Systems*, volume 2006, pages 117–123, Alexandria, VA, United states, March 2006. IEEE Computer Society.

[5] V. Fox, J. Hightower, Lin Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *Pervasive Computing, IEEE*, 2(3):24–33, July-Sept. 2003.

[6] Heinrich W. Löllmann and Peter Vary. Uniform and warped low delay filter-banks for speech enhancement. *Speech Communication*, 49(7-8):574 – 587, 2007. Speech Enhancement.

[7] K.E. MacLean. Foundations of transparency in tactile information design. *IEEE Transactions on Haptics*, 1(2):84 – 95, July 2008.

[8] Toni Pakkanen and Roope Raisamo. Appropriateness of foot interaction for non-accurate spatial tasks. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1123–1126, New York, NY, USA, 2004. ACM.

[9] Ignasi Rius, Xavier Varona, F. Xavier Roca, and Jordi Gonzàlez. Posture constraints for bayesian human motion tracking. In *Articulated Motion and Deformable Objects*, pages 414–423. Springer Verlag, 2006.

[10] Y. Visell and J. Cooperstock. Visell, Y. and Law, A. and Smith, S. and Ip, J. and Cooperstock, J. In *Proc. of IEEE Haptics Symposium*, 2010.

[11] Y. Visell and J.R. Cooperstock. Optimized design of a vibrotactile device via a rigid surface. *IEEE Haptics Symposium*, 2010.

[12] Y. Visell, A. Law, S. Smith, R. Rajalingham, and J. Cooperstock. Contact sensing and interaction techniques for a distributed multimodal floor display. In *Proc. of IEEE 3DUI*, 2010.