# FOOTSTEP PATTERN MATCHING FROM PRESSURE SIGNALS USING SEGMENTAL SEMI-MARKOV MODELS

*Kalle Koho, Jaakko Suutala, Tapio Seppänen, and Juha Röning*

Intelligent Systems Group, Infotech Oulu
90014 University of Oulu, Finland
{Kalle.Koho,Jaakko.Suutala,Tapio.Seppanen,Juha.Roning}@ee.oulu.fi

## ABSTRACT

This paper describes experiments on recognizing footstep patterns from data produced by pressure-sensitive floor. A 100 square meter pressure-sensitive floor (EMFi floor), which is placed as a part of a smart living room, senses the changes in the pressure against its surface and produces voltage signal of the event. Recognition of footstep patterns is needed for data segmentation to be used in person tracking and identification. We have used a method based on Segmental Semi-Markov Models to detect footsteps from the floor data. The experiments described in this paper show this method to be a very powerful and robust tool for our particular application.

## 1. INTRODUCTION

The research on intelligent environments [1], [2], [3] aims at making smart houses, offices, tourist attractions etc., where the environment learns and reacts to the behaviour of the occupants. The methodology can be applied in homes for the elderly and disabled as an enabling technology for monitoring hazardous situations as well as in surveillance systems or in helping in child care.

One fundamental problem that has risen in connection with the development of these intelligent environments is their ability to automatically determine and keep track of the actual physical location and the identity of a person in indoor environment. Several technologies have been presented to address this problem, including camera-based systems, WLAN-positioning, combined RF- and ultrasound sensing systems, and force-sensing floor sensors based on load cells [4]. Our approach to this indoor locationing problem is an EMFi-based floor sensor, measuring pressure changes affecting on it. In order the person tracking and identification [5], [6] to work accurately, we need a technique to detect footstep patterns from EMFi-data.

For this purpose, we have applied a method described in [7] based on segmental semi-Markov models (SSMM), to detect these patterns reliably. In this method, a Markov-based piecewise linear model is constructed from a single example pattern and then a Viterbi-like algorithm is used to detect similar waveforms from EMFi-data. This paper concentrates only on the experiments on detecting the footstep waveform patterns and will not cover the issues of person tracking and identification.

In the following sections of the paper, we begin by giving general description of the EMFi-floor in Section 2. In Section 3 the pattern matching method based on SSMM is presented, and Section 4 describes the experimental results when this method is applied to EMFi-data.

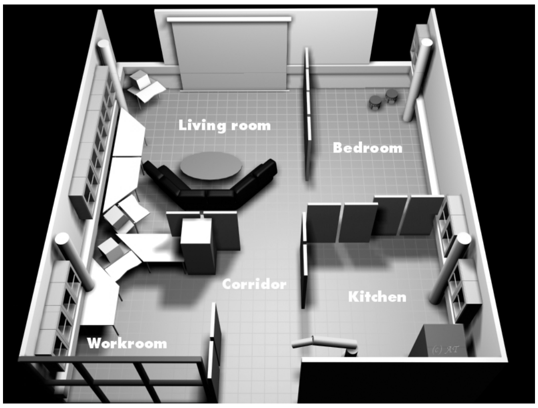


Figure 1: Computer generated view of our research laboratory. EMFi-sensor stripes are illustrated on the floor.

## 2. EMFI FLOOR

ElectroMechanical Film [8] (EMFi) is a thin, flexible, low-price electret material which consists of cellular, biaxially oriented polypropylene film coated with metal electrodes. In the EMFi manufacturing process, a special voided internal structure is created in the polypropylene layer, which makes it possible to store a large permanent charge in the film by the corona method, using electric fields that exceed the dielectric strength of EMFi. An external force affecting the EMFi surface causes a change in the film's thickness, resulting in a change in the charge between the conductive metal layers. This charge can then be detected as a voltage signal. EMFi is a Finnish innovation and a trademark of EMFiTech Ltd.

EMFi material has been installed in the Intelligent Systems Group's (ISG) research laboratory at the University of Oulu. The EMFi floor in the ISG laboratory is constructed of 30 vertical and 34 horizontal EMFi sensor stripes, 30 cm wide and 10 meters long each, which are placed under the normal flooring (see Figure 1). The stripes make up a 30x34 matrix with a cell size of 30x30 cm. Instead of simply installing squares of EMFi material under the flooring, stripes were used because this layout requires clearly less signal channels to be processed.

## 3. RECOGNITION OF FOOTSTEP PATTERNS

### 3.1 Overview

#### *3.1.1 Data*

On the left graph of Figure 2, a voltage signal from one EMFi-channel with a footstep pattern in it is shown. The waveform resulting from a footstep is clearly two-peaked, the first resulting from heel strike and the second one from toe push off. The EMFi-floor measure system is of a high-pass type with the cutoff frequency around 1.5Hz.

The footstep pattern shown in Figure 2 appears when the whole foot hits in the middle of the EMFi-stripe. Due to the fact that the EMFi-stripes are only 30 cm wide, the majority of the steps, in fact, hit in between two adjacent stripes resulting in one-peaked partial footstep patterns. This means that for example the heel strike profile can appear in one channel and the toe push off profile in the one next to it. These partial footstep patterns are poor material, especially for the user identification purposes, so we needed a more robust tool for footstep pattern recognition than just a simple amplitude based thresholding techniques. Another main reason from a data point of view for our method selection was the high-pass nature of the EMFi-floor. This causes the fact that the baseline of the EMFi-signal starts to fluctuate when several consecutive footsteps hit on the same stripe (see Fig. 3). This again makes simple signal amplitude based thresholding inadequate for footstep pattern detection.

#### 3.1.2 *Method Overview*

In order the person tracking and identification to work properly with EMFi-floor, the footstep waveform patterns must be segmented from raw data. A statistical pattern matching method based on SSMM [7], [9] was chosen for our application with some modifications made to the original algorithms. The segmental semi-Markov model is an extension to the standard Hidden Markov Model (HMM) [10].

These extensions include two major components: first, explicit state duration distributions, and second, segmental observation distributions. This means that unlike in a standard HMM, where a state generates a single observation $y_t$, a state in a SSMM generates a segment of observations $y_{t_1} \dots y_{t_2}$. The duration of this segment in time is modelled by a specific distribution (for example Gaussian) with a mean duration and some variability around that mean. In this segment observation model, the data generated by each state is in the form of some regression curve,

$$y_t = f_i(t|\theta_i) + e_t \tag{1}$$

where $f_i(t|\theta_i)$ is a state-dependent regression function with parameters $\theta_i$, and $e_t$ is additive independent noise (usually Gaussian).

With these extensions, the SSMM offers a very flexible way to model waveforms. The state duration distributions and segmental observation models bring the aspect of *shape variability* into the detection prosedure. This is very important in our application because, due to the nature of the EMFi-sensor, the footstep waveforms vary quite strongly depending on how do you hit your foot on the floor. This means that it is almost an impossible task to build a specific footstep pattern template for every kind of step.

When a footstep waveform is modeled with SSMM, meaning a specifically ordered combination of individual observation segments, each segment having its own state parameters, a Viterbi-like algorithm described in [7] can be used to detect similar waveforms in the data generated by EMFi-sensor. The footstep pattern modeling and recognition are described in more detail in the next two sections.

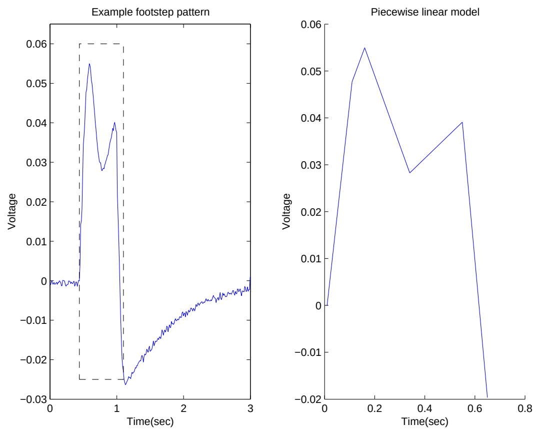


Figure 2: Example footstep pattern (squared) and its piecewise linear representation consisting of five segments.

### 3.2 Modeling of Footstep Pattern

When constructing a model for footstep pattern, the first step is to create a piecewise linear representation of the example waveform. For this purpose, an optimal piecewise linear segmentation algorithm (PLS) described in [9] was used. We chose to use the algorithm where the number of segments, $K$, is fixed. Slight modifications were done to the algorithm to make it more suitable for our application. The main alteration compared to the original PLS-algorithm was the definition of approximation error when fitting the $K$ linear segments into the example waveform. Instead of minimizing the single maximum difference between a sample in the example pattern and a corresponding point on the approximating linear segment we chose to minimize the sum of these differences for each segment. This gave better piecewise linear approximation results, especially when the number of segments $K$ was small (less than 6). On the right graph of Figure 2, a piecewise linear representation of the example footstep pattern can be seen.

From this piecewise linear representation, a $K$-state segmental semi-Markov model is constructed. Each state in the model corresponds to one segment in the piecewise linear representation of the example waveform. The state transition matrix $A$ for the model will be left-to-right, in other words, $A_{i,i+1} = 1$, $A_{i,j} = 0$ if $j \neq i+1$ and $A_{i,j}$ is the probability of going to state $j$ given that the process is in state $i$. The initial state distribution is $\pi = [1, 0, \dots, 0]$. The output probability distribution of state $i$ is now

$$\begin{aligned} p(y_{m+1} y_{m+2} \dots y_{m+d_i} | s_i) = \\ p(d_i|s_i) p(\theta_i|s_i) \prod_{t=m+1}^{m+d_i} p(y_t | f_i(\theta_i, t)), \end{aligned} \tag{2}$$

where the state-dependent regression function for this model is a linear function $f_i(\theta_i, t) = b_i t + c_i$. The state $i$'s regression parameters include now $b_i$ and $c_i$, but the intercept $c_i$ is ignored in the model and allowed to be freely fit during the detection process, allowing shifting in amplitude range. So the only regression parameter left in the model is $b_i$, which is the slope of the $i$'th segment in the piecewise linear representation. $p(d_i|s_i)$ is the state duration distribution for state $i$. It's a truncated Gaussian distribution with mean $l_i$ which is set to be the actual duration in time of $i$'th segment in the piecewise linear model. The standard deviation for $p(d_i|s_i)$ was set to be $l_i \times k\%$, where the value of $k$ was set based on a prior knowledge of the waveform to be modeled. Segmental observation distribution $p(y_t|f_i(\theta_i,t))$ is Gaussian distribution with mean $f_i(\theta_i,t)$ and additive noise variance $\sigma_y^2$. $\sigma_y^2$ is calculated for each segment separately as the mean squared error when the segments from the piecewise linear representation are compared against the original example data.

The original method described in [7] contains a *pre-pattern* and a *post-pattern* as garbage states in the model, which are used to model the data before and after the matched pattern. These extra states can be used to match the pattern in interest against the whole time series directly. In our application, only a short noise state was added to the model to appear before the actual footstep pattern. With this modification the starting point of the footstep pattern was more accurately detected. The duration of the noise state was set to be five time units and the regression parameter (the slope of the noise segment) was set to be zero. The variance for this pre-pattern state was set to be the actual noise variance of the raw EMFi-signal.

### 3.3 Footstep Pattern Matching

After the model has been created, the actual pattern matching starts with measuring the sum of cubed samples in a small sliding window (size of 5) from the incoming signal. The goal is to detect occurences of possible pressure events on the floor. Because of the zero-average nature of EMFi-signal, the summing process effectively filters the noise in the channel, and with the fixed threshold it is possible to detect the start time of an event. In addition to this thresholding, a rough estimate of signals trend is also computed for samples inside the window. This means simply calculating the slope of a line fitted to the data points in the sliding window. If the signal is at rise when the threshold limit is crossed, then the actual pattern matching is started.

First a short pre-pattern state is used to find the exact starting point of the pressure event. At each time unit, regression functions of each state are matched to find the most likely one. If the whole state sequence can be found in a right order from the time series, the matched pattern has been found. On the other hand, if the last state is not reached in the state sequence within a particular time limit, pattern matching is aborted. In both cases, the detection of the pressure changes is then started again.

Footstep pattern matching is based on finding the most likely state sequence in the segmental model $\hat{\mathbf{s}} = s_1 s_2 \ldots s_t \ldots$ for a data sequence $\mathbf{y} = y_1 y_2 \ldots y_t \ldots$. After the footstep model is constructed, as presented in previous section, the most likely state sequence can be determined using a recursive Viterbi-like algorithm.

The quantity probability $\hat{p}_i^{(t)}$ is calculated for each state $i$ in the model, at each time $t$, and recorded in a table. $\hat{p}_i^{(t)}$ is the likelihood of the most likely state sequence that ends with state $i$. The recursive function for calculating $\hat{p}_i^{(t)}$ is defined as

$$\hat{p}_i^{(t)} = \max_{d_i}\left(\max_j \hat{p}_j^{(t-d_i)} A_{ji}\right) p(d_i) p(y_{t-d_i+1} \ldots y_t|\theta_i), \quad (3)$$

where $y_t$ is the last point of segment $i$. $d_i$ is duration and $p(d_i)$ is its probability of state $i$ in the model. The last point of the previous segment will be $t - d_i$. $A_{ji}$ is the state transition matrix and $p(y_{t-d_i+1} \ldots y_t|\theta_i)$ is the probability of fitting the state $i$'s regression function to an given sequence of samples. For a given $d_i$, the inner maximization ($\max_j$) calculates over all possible previous states $j$ that transition to state $i$ at time $t - d_i$. The outer maximization ($\max_{d_i}$) is over all possible values of the duration $d_i$ of state $i$. The state $j$ and time duration $t - d_i$ for the maximum value of $\hat{p}_i^{(t)}$ are recorded in table.

Finally, the most likely state sequence for the given data sequence $y_1 y_2 \ldots y_t$ is back tracked from the table. It is the state sequence with the likelihood $\max_i \hat{p}_i^{(t)}$ and is considered as optimal in a maximum likelihood sense to describe the state-sequence against the observed data.

In the Viterbi-like algorithm which is used to calculate the most likely state sequence for the observed data sequence, we chose to calculate the actual Viterbi decoding part only when the last state ($K$th state) in the model appears to be the most likely one. In the original algorithm, the Viterbi decoding was done at every time instant regardless of the state that appeared as the most likely one. This modification made the pattern matching algorithm work faster.

## 4. EXPERIMENTAL RESULTS

### 4.1 System Implementation

Each of the 64 EMFi-stripes in the floor produces a continuous signal that is sampled at a rate of 100Hz and streamed into a PC from where the data can be analyzed in order to detect and recognize the pressure events, such as footsteps, affecting the floor. The analogous signal is processed with a National Instruments AD card, PCI-6033E, which contains an amplifier. It would be possible to increase the sampling frequency up to 1.56kHz, but 100Hz was considered adequate for this application.

The SSMM-based footstep pattern recognition system described in this paper was implemented using MATLAB programming language. The system works off-line, meaning that the EMFi-data is first streamed in a file and then processed from there with MATLAB (Product of MathWorks Inc.). We chose MATLAB programming environment for the first phase of the system implementation because it offers very powerful in-built tools for data visualization and statistical calculations and the code is fast to write and easily modified. A real-time recognition system also is being developed at the moment based on the MATLAB-implementation. This is done with C++-programming language.

### 4.2 Recognition Results

The experimental results show that the SSMM-based footstep pattern recognition application we implemented works very well on the EMFi-floor data. In Figure 3, all three footstep patterns are succesfully detected (the solid line indicates the start point and the dashed line the end point of the steps). When we look at the detected waveforms in Figure 3, the allowed shape variability is obvious. All three detected footstep patterns are different variations of the created model within the limits of the models parameters. The model used for this test run is the one presented in Figure 2. It has five states and the state durations (segment lengths in time) are allowed to have 30% variability around the mean values of the durations, meaning that the standard deviation of the state duration distributions is $l_i \times k\%$, where $l_i$ is the mean duration of state $i$.

With these two main parameters (number of states in the model and the standard deviation of the state duration distributions), our footstep pattern matching application is easily adjustable to detect different kinds of footstep profiles. For example if we only want to determine the location of a person walking on the EMFi-floor, the model we create from some example waveform can be quite general. This means that the number of states is set to be small (less than six), and the standard deviation of the models state duration distributions is set to be large ($k$ is for example 30% or even greater). We could even build several footstep models to detect both partial and whole footstep patterns, described in section 3.1.1, because in the only-tracking scheme the partial and whole footstep profiles provide equally significant information about walkers location.

On the other hand, when we use the detected footstep patterns for person identification, the waveforms should be as good as possible, meaning that whole foot has hit in the middle of one EMFi-stripe. In this case, the model created for the pattern matching must be more selective. In our experiments, we found that for person identification purposes the number of states in the model should be at least five or more, and the value of $k$ should be set to 15% or less. When we dropped value of $k$ from 30 to 10 in the model creation, only the first footstep pattern in Figure 3 was detected because its shape is closest to the created model.

It should also be mentioned that even though the number of states is the other parameter given in the model creation, the actual parameter used in the detection procedure it effects on is the noise variance in the segmental observation distributions. This is because these segmental noise variances are calculated as the mean squared error when the linear segments of the model are fitted into the example waveform, as described in section 3.2. This means that the less states in the model, the larger the values of the noise variances in the segmental observation distributions are, allowing larger variability in the slope of the linear regression functions that each state generates. The effect of the other parameter, standard deviation of the state duration distributions, is quite straightforward, because it simply defines the allowed lengths (in time) of the linear segments that are fitted in to the data with certain probability assosiated with them.

With these few examples presented, the strengths of this particular pattern mathching method are obvious when it is applied to the EMFi-data. The important aspect of shape variability was brought to the footstep detection procedure and a prior knowledge about the waveform in interest can be easily incorporated to the model creation with parameters described above. In our experiments, the system has proven very reliable and accurate, and we have been able to overcome many of the difficulties that previously appeared with some simple thresholding based detection techniques.

### Acknowledgments

This work was funded by TEKES and Academy of Finland. We would like to thank Prof. Heikki Mannila for suggesting the use of SSMM method in our application.

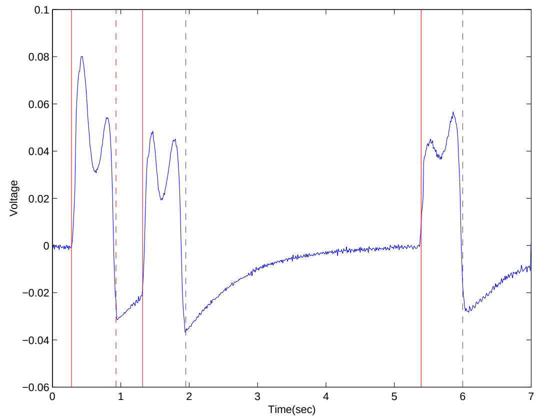


Figure 3: Three footstep patterns succesfully detected. Shape variability allowed by the SSMM-method can be clearly seen in the detected waveforms.

## REFERENCES

[1] I. A. Essa. Ubiquitous sensing for smart and aware environments: Technologies towards the building of an aware home. *IEEE Personal Communications*, October 2000. Special issue on networking the physical world.

[2] MIT's oxygen project. http://oxygen.lcs.mit.edu/. Available 07.11.2003.

[3] Aware home. http://www.cc.gatech.edu/fce/ahri/. Available 07.11.2003.

[4] J.Hightower and G.Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[5] S. Pirttikangas, J. Suutala, J. Riekki, and J. Röning. Learning vector quantization in footstep identification. In M.H. Hamza, editor, *Proc. 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA 2003)*, pages 413–417, Benalmadena, Spain, September 8-10 2003. IASTED, ACTA Press.

[6] S. Pirttikangas, J. Suutala, J. Riekki, and J. Röning. Footstep identification from pressure signals using Hidden Markov Models. In *Proc. Finnish Signal Processing Symposium (FINSIG'03)*, pages 124–128, May 2003.

[7] X. Ge amd P. Smyth. Deformable markov model templates for time-series pattern matching. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 81–90, August 2000.

[8] M. Paajanen, J. Lekkala, and K. Kirjavainen. Electromechanical film (EMFi) - a new multipurpose electret material. *Sensors and actuators A*, 84(1-2), August 2000.

[9] X. Ge. *Segmental Semi-Markov Models and Applications to Sequence Analysis*. PhD thesis, University of California, Irvine, December 2002.

[10] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, vol.77(2.):257–286, 1989.