Foot Fingerprints

Rudolfo Santos José Monteiro

INESC-ID / Instituto Superior Técnico
Lisbon, Portugal
rars@mega.ist.utl.pt, jcm@inesc-id.pt

Abstract. A footstep-based identification system is proposed in this article. The usage of average acceleration sensors on existing nodes for wireless sensor networks is all that is required, allowing for an easy implementation, without resorting to hardware development. Furthermore, wireless sensor networks permit that this application can take advantage of several techniques such as redundancy, scalability, matching, etc. to increase its robustness. The approach we follow in this paper is based on the strength of the steps and the time between steps for detecting people by their footsteps. The system goes through a learning phase, where patterns for different users can be recorded. With this information, during regular operation the system infers if the subject is one of the known users or an intruder.

1 Introduction

It is common for people or animals to detect someone by his/her footsteps. The footsteps are recognized first through their sound. However, probably sound does not have enough information to uniquely determine a given subject. Different shoes and floors can change the produced sound, without affecting the recognition capability. One of the most accurate measures is the time between footsteps. Normally people have a specific periodic footstep that enables us to identify who is it. Although this is not the best method to differentiate people, because there are different persons with similar periodic footsteps, the same speed and may even have a close physical structure. It is still a good mechanism for reducing the number of matching hypothesis for users, to inform of intrusion detection, or just for controlling domotic applications.

As an application example, let's consider a huge company with hundreds of employees. Employees must walk before reaching a determined checkpoint that needs to correctly identify that person. However, existing systems require some action from the person before starting a database search for that user. The proposed system would alleviate this action for identification, making it more efficient for the user.

For detecting footsteps the sensors must be attached to the ground in order to pick up all interesting vibrations. This means that the sensors need to be robust, small and cheap. Robust to be able to work in different conditions (strong footers, running people, etc.), small for a good ambient ubiquitous application and cheaper to allow replacement and system development more attractive.

Although the introductory part may be simple, some problems with this kind of equipment exists:

1. Different sensors have different impulse responses to the same input;
2. Distance between sensors must be known for calculating the correct footstep period;
3. Synchronization in the sensor network is crucial, to accurately determine the time between occurrences;
4. Network should be robust enough to work even in the case of failure of one or more nodes.

In this paper, we describe an approach that addresses all these points and uses wireless sensor nodes for its basic infra-structure.

2 Related Work

Wireless sensor networks (WSN) are growing in popularity, with new projects and applications appearing each day. WSNs have been used essentially in scientific research and academic projects [9, 10]. However the industry is starting to look at WSNs as a very interesting future profitable market. The main projects have been seismic and building monitoring, nature observation and medical treatments, examples are [6–8]. Nevertheless, many different applications are still to be found.

Some groups have developed work on projects similar to ours. In [1] a smart floor was proposed to detect objects on the floor and footstep characterization. To identify the users they used Hidden Markov Models, with some interesting results. Also with the same approach, [2] used a Electromechanical film (EMFi) floor and the signal processing used was also Hidden Markov Models. They achieve a footstep correction rate of 78% for 3 users. In [3] they use load cells, steel plate and data acquisition hardware. The identification processing was in charge of a nearest neighbor algorithm, which according to the article achieved a rate of 93% correct matches.

All of them however use a similar approach, where a special ground is needed. This has several disadvantages as: it may require expensive ground work; difficult replacement of malfunctioning devices; requires more robust (ie, more expensive) sensors to support heavy weight. Our approach is simpler, less expensive, and quickest to install due to the usage of off-the-shelf material.

3 Equipment

In this section we describe the equipment used, and we make an introduction to the sensor dynamics.

3.1 Sensor Nodes

Due to the objectives of the work, simulation could not be used to validate the results. Mainly because simulations are good for evaluating gains and tests, but they lack the measures and problems of the real world.

For this project we used the Micaz motes attached to a MTS310CA sensor board from CrossBow [4], as the one depicted in Figure 1. This sensor board is equipped with several sensors, e.g. Dual-Axis Magnetometer, Dual-Axis Accelerometer, Temperature, Light, etc.

The Micaz Motes use the recent wireless technology Zigbee, developed for low data rate application. This communication protocol is well adequate to the purpose of this project. The Zigbee protocol as several advantages:

1. 256kbps data rate, sufficient to support low to median flow of data;
2. Adequate for dozens to hundreds of tiny sensors in a mesh kind sensor network;
3. Quick (and auto) formation of the network.

The main disadvantage of Zigbee is the radio receive power consumption and the still young life time of the protocol.

Fig. 1. Crossbow© Micaz mote.

3.2 Accelerometers Dynamics

The accelerometer used in the MTS310 boards is the ADXL202 from Analog Devices [5]. This accelerometer is a dual-axis sensor, with bandwidth adjustment through the use of a Capacitor. It measures static and dynamic acceleration and can consume less than 0.6 mA in operation. The main applications are computer peripherals, inertial navigation, seismic monitoring and others.

The measurement range is from $\pm 2g$. The bandwidth is adjustable through the setting of two capacitors (C_x and C_y), and can vary between 0.01 Hz - 5 kHz. The operating voltage can be 3 V enabling the use for low power operation.

Based on the specifications, and expecting no rough handling of the sensors, we believe that the equipment is sufficient for the job. According to the expected, practical experiments show that different sensors have very different responses even when exposed to the same phenomenon. As we will show, this difficulty can be circumvented, allowing these devices to be used effectively.

4 Implementation

In this section, we first describe how we arrived at the parameters used for the filter, and then discuss the code implemented both at each node in the network and at the base station.

4.1 Footstep Dynamics

A typical person walks at a median speed of $v = 5$ km/h. According to the physical structure and character of each individual, his/her footstep may be longer or shorter, quicker or slower than average. However, we will take into consideration a median foot step equal to 50 cm.

With these two values we obtain the period of each footstep.

$$v = 5\text{km/h} = 1.39\text{m/s}.$$

This results in a frequency of

$$f = \tfrac{1.39}{0.5}\text{s}^{-1} = 2.78\text{Hz}.$$

These value indicates that on average in each second we take two steps. Obviously these values may slightly differ from person to person, without almost no impacts to the present work.

So to detect these events we need to choose a filter cutoff frequency at least two times larger than that. That means a Nyquist frequency of 6 Hz. We chose a value of $f_N = 15$ Hz to detect events. This value is more than five times the base frequency and is sufficient for a good sampling of the footsteps, although at the same time allowing enough time for the sensor to be turned off when not used.

4.2 Node Programming

Each node in the WSN performs environment measures and consists in a Micaz mote equipped with a MTS310 sensor board. It is responsible for executing the sampling, calculate the mean value and deciding if there was a step. If a step is detected the node will send the data to the base station.

The event of a step is detected by a FIR filter, with 16 coefficients, obtained recurring to the Matlab program [11]. The function used was the `firls`, which implements a linear-phase FIR filter design using least-squares error minimization.

As the node is a piece of hardware supposed to be left alone running for extensive periods, great importance must be given to extend its energy supply. There are several software tricks that can be performed to reduce power consumption. A list of some of them is:

1. Sleep modes: Micaz hardware has plenty of sleep modes that can be used to reduce power consumption. The Tiny OS switches the system to Idle when not performing useful computation, but further effort can be done. Turning off the radio, and go into deeper sleep modes are effective ways to reduce power consumption.
2. Transmitting data: data transmission is vital in any network. However is also responsible for the largest consumption of energy in WSNs, and must be used with great care. One of the main purpose of telecommunications is to transfer the largest amount of information possible with the least cost, hence the primary action to minimize power consumption is to send only valuable information. One technique is, instead of sending the absolute value for all data, just send the difference to a mean of the value. This way it is possible to reduce some bits. While each message may have little power savings, compounding within several messages this value can have a large impact.
3. Mathematical manipulation: some mathematical operations may require a large number of clock cycles from the CPU, specially if the operations use floating point. Hence, one optimization that leads to large power savings is to use fixed point operations instead of floating point. While this is not always possible, in many cases programmers needlessly use floating point. One good example of a mathematical optimization is the calculation of the mean value. The mean is given by $m = \frac{1}{N}\sum_{k=1}^{N} x[n]$. Each time this calculation is performed consumes N additions plus one division. Supposing that we have a circular buffer to keep the values, the newest value will overwrite the oldest. So a simple way to obtain the mean value is: $m_{new} = m_{old} + \frac{x_{new}}{N} - \frac{x_{old}}{N}$. Performing this way we just need 2 division and 2 additions. Since the denominator is the same, we can further remove one of the divisions. We have just reduced from $(N + 1)$ operations to just 3 operations. However, one of them is still a division, which consumes several clock cycles, hence energy. If N is a power of 2, $N = 2^k$, and fixed point representation is being used, it is possible to replace the division with a simple shift of k bits to the right. The final formula would be: $m_{new} = m_{old} + (x_{new} - x_{old}) \ll \log_2 N)$. So, the optimal implementation of the computation of the mean value is reduced to two additions and one shift operation.

4.3 Base Station Programming

The base station consists of a Micaz mote mounted on a communication board with a USB connection to a PC station running Linux.

The Micaz mote does not perform any computation, it just receives messages from/to the UART and sends/receives it to/from the Radio. This Micaz mote is responsible for the synchronization of all the nodes in the network.

The PC processes the packets, combines with other packets and the recognition process is executed. This information is used to detect if there was a walk or it was just a spurious event. In case a walk was detected, it tries to match the steps to the correct owner, otherwise it informs we are in the presence of an intruder.

The PC has two main different applications: the first is a learning process that adds new users to the system. The learning process expects the user to walk for a while. It then keeps its median footstep acceleration (AX and AY), the median sound produced, time between footsteps and also the last time the user was detected or made login to the system. This information is kept in a database, for latter use.

The second main application at the base station is the classifier. A close neighbor algorithm was used to detect the user. The identification of a user was based on events received from the several sensors. Each time a sensor reports an event, the classifier searchs into his database for the last events detected by each other sensor. It retains the closest pair of events, and compares them with the information stored for each user. The closest user to the chain of events is marked, and if below a pre-defined threshold it is said that the user as stepped in.

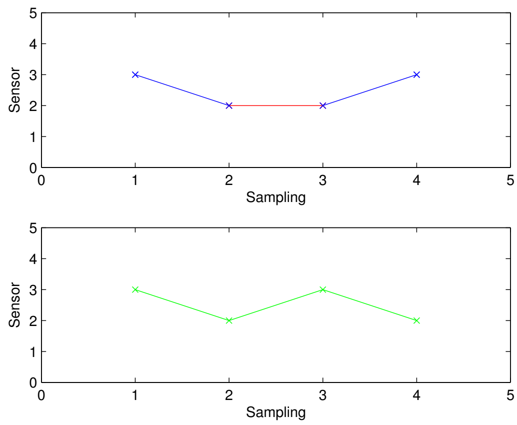

Fig. 2. Flows of the footsteps.

However, due to some footsteps not being captured by the nodes, not all events trigger a database search. For example, in Figure 2 upper graphic, samples 2 and 3 Both came from node 2. This means that between this two events, one third event is missing as it is unsual for a person to take two steps on the same location. When processing the arrived events, this problem is taken into account. So, the time between two following events from the same sensor is simply stored. A good footstep detection is the one that occurs in the lower graphic of Figure 2. There was a step in sensor 2, then an event on sensor 3, and again in sensor 2.

5 Results

5.1 Node Sensitivity

As refereed, we have used as sensors the accelerometers and the microphone at each node to detect the movement. The sensors need to be correctly deployed in order to obtain the best event description. To get a feeling of the best possible installation of the nodes, three sensors were used.

The positioning of the sensors is also relevant. In a first experiment, the nodes were placed such that the accelerometers were in direct contact with the ground, and in the second experiment the sensors were facing up. In Figures 3 and 4 it is possible to observe the response of one of the sensors to these two situations. The results indicate that the sensors should be attached to the ground, for better event detection.

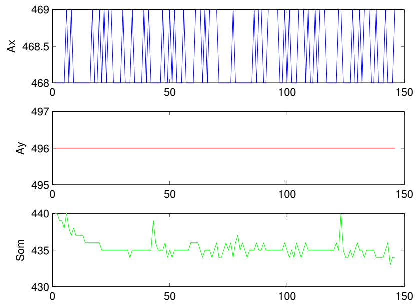

Fig. 3. Sensor 2 sensing the air.

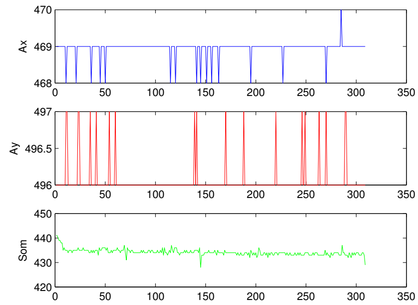

Fig. 4. Sensor 2 sensing the ground.

In each of the tests, the sensors were deployed in the same place, and with the same orientation. A person walked right next to the sensors maintaining the same walking pattern and step positioning. In these tests also an important parameter is the shoe and the floor. The type of shoe influences the type of sound and/or force transmitted to the ground. The floor is responsible for the acceleration to be correctly delivered to the sensor, or not.

Figure 5 depicts the measures obtained by each of the three sensors. The graphs on the left present the raw data, and the graphs on the right the measurements after being processed by the FIR filter. We can observe that there is a significant difference in behavior of each sensor. Sensor 1 is more *hard of hearing* to events than other nodes. Sensor 2 has a more random behavior. Sensor 3 has a behavior close to 2. This shows that similar sensors can have very different behaviors. So, a correct detection of a footstep should not be made by just one node, but by a group of sensors. This adds redundancy of information to the event, and reduces the probability of false events or the missing of a real event.

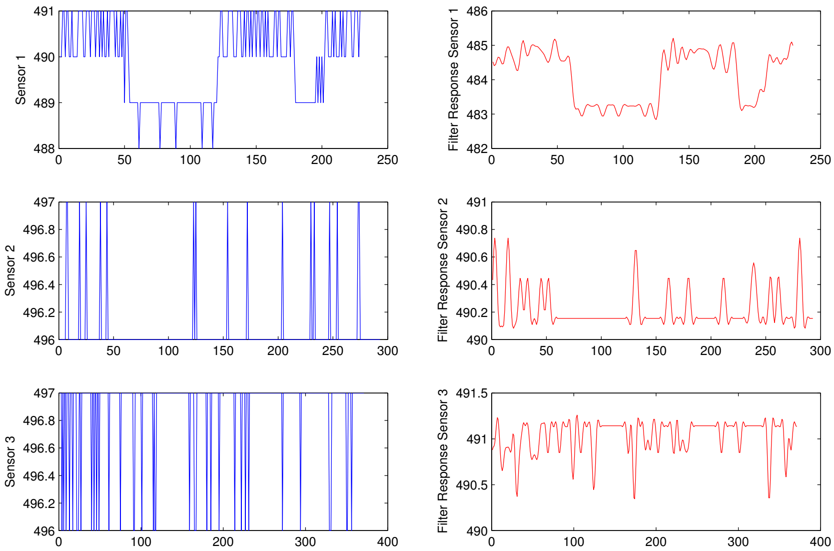

Fig. 5. Several sensor measures and filter responses.

In the right part of Figure 5 it can be observed the response of the FIR filter to each raw data. Placing a horizontal line around value 490.5 in the 2_{nd} and 3_{rd} group of images, it can be seen that there are 4 peaks that correctly identify the 4 given steps. The steps can be easily detected by comparing the difference between the current result of the filter and its median value, and comparing with a threshold value.

5.2 Footstep Detection

After the implementation, some experiments were made to test the reliability of the system. The test consisted of counting the number of correct/false footstep detections in a row of 6 events.

In Table 1, it is tested the reliability of the sensors to a step. The first column indicate the number of the sensor, the *Detected steps* indicate the detected steps. The *Not detected* are the not detected steps and also include some spurious step, and finally the *Series* indicate which series gave that result.

The average shows a 67% to 100% of correct results. This results are optimistic about the capabilities of WSNs sensors.

5.3 Classification

A important component of the system is the classifier. This component receives and process the events (footsteps). The classifier should identify a walk and try

Table 1. Node Event Sampling

Node	Detected Steps	Undetected Steps	Total Events	%	Series
2	5	1	6	83.33	1
3	6	0	6	100	1
2	5	1	6	83.33	2
3	5	1	6	83.33	2
2	4	2	6	66.67	3
3	4	2	6	66.67	3
2	4	2	6	66.67	4
3	5	1	6	83.33	4

to match the walk to a user. In case there is no user that matches the walk, then a alarm is issued. The pattern classification is a research topic in vast progress. There are several well studied algorithms that could be used in this project. As the focus of the project are on sensor networks, we use a simple classification scheme already used in [3]: the nearest neighbor algorithm. The algorithm has obtained good results, and its simplicity makes a good candidate for WSNs. The algorithm could even be used in any sensor node, due to is simplicity compared with more powerful techniques as Expectation-Maximization algorithm, Hidden Markov Models, etc. In Table 2, it is presented the results of the classification algorithm with one user in the system, with a 3 series event.

Table 2. Classifier test with one user

Series	User	Correct	False	Steps	%
1	A	6	6	12	50
2	A	5	3	8	62.5
3	A	7	3	10	70

In Table 2 the average gives a 61% of correct detected steps from that user. This result is above median, but it should be improved.

6 Conclusions and Future Work

In this project a Foot Fingerprint application was proposed. The application was divided into two parts: the nodes and the base station. The nodes run an application that periodically samples the accelerator sensors, performs a filtering scheme and informs the base station of the occurrence of an event (a step). The base station receives data messages, sends configuration messages, and performs synchronization between all nodes. In the base station a classification algorithm is running to detect who performed the footstep. The acceleration measures are not enough to correctly identify the user. The time between events in adjacent nodes is used to estimate the user, by comparing the measured time to previous learned user steps.

Several objectives within this work were established. The main one was to prove that it is possible to produce a cheap, interesting and useful application using on-the-shelf material. This was the great advantage within this work, compared to previous approaches that needed big efforts.

The core of the work is done, however the performed tests suggest that some work is still needed to better identify the users. Most of it consists on tests and upgrades. Two issues that may addressed are the implementation of a more robust classifier and using topology information. The topology control will permit to get the actual node positions, and to calculate the real distances between nodes, increasing the recognition capability. A multihop low power algorithm could be used to allow the topology and synchronization algorithms to run without depleting the nodes. The synchronization algorithm should maintain all nodes synchronized. Finally, the classification algorithm should be more accurate between different users, and more resilient to nodes failures. There is much space for further development of classifiers in network sensors. For example, distributed classifiers.

References

[1] M. Addlesee, A. Jones, F. Livesey, and F. Samaria: *The ORL Active Floor*, The Olivetti and Oracle Research Laboratory.

[2] S. Pirttikangas, J. Suutala, J. Riekki and J. Roning: *Footstep identification from pressure signals using Hidden Markov Models*, Intelligent Systems Group, Infotech Oulu, FIN-90014 University of Oulu.

[3] R. Orr and G. Abowd: *The Smart Floor: A Mechanism for Natural User Identification and Tracking*, College of Computing and GVU Center, Georgia Institute of Technology.

[4] *http://www.crossbow.com*

[5] *http://www.analog-devices.com*

[6] N. Xu, S. Ranwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, D. Estrin: *A Wireless Sensor Network For Structural Monitoring*, Conference on Embedded Networked Sensor Systems, Baltimore, MD, November 2004.

[7] R. Szewczyk, E. Osterweil, J. Polastre, et al: *Habitat Monitoring with Sensor Networks*, Communications of the ACM, vol. 47, no. 6, pp. 34–40, June 2004.

[8] A. Mainwaring, J. Polastre, R. Szewczyk, and D Culler: *Wireless Sensor Networks for Habitat Monitoring*, WSNA'02, Atlanta, Georgia, September 2002.

[9] Intel Research Center, Vibration Monitoring: *http://www.intel.com/research/exploratory/wireless_sensors.htm*

[10] Center for Embedded Networked Sensing: *http://research.cens.ucla.edu/projects/2006/Seismic/Mexico/default.htm* UCLA, UCR, Caltech, USC, CSU, JPL, UC Merced.

[11] *Matlab*: The MathWorks, Inc. *http://www.mathworks.com/*