Sensor evaluation for real-time pace and step detection for sonic interaction

Report of Short-Term Scientific Mission (STSM)

Funded by the COST IC 0601 Action on Sonic Interaction Design (SID)

July 15, 2010

1 STSM Information

1.1 Applicant

Bart Moens
Institute for Psychoacoustics and Electronic Music (IPEM)
Ghent University, Belgium
bmmoens.moens@ugent.be

1.2 Host institution

Input Devices and Music Interaction Laboratory (IDMIL)
Music Technology, McGill University, Canada
Contacts: Marcelo Wanderley (marcelo.wanderley@mcgill.ca) and Jason Hockman (jason.hockman@mail.mcgill.ca)

1.3 Purpose

The STSM took place from May 25th 2010 to June 25th 2010. The purpose was to evaluate different sensors, locations and algorithms for real-time step detection for use in sonic interaction, using both quantitative and qualitative evaluation. For the quantitative evaluation, we compare the results with a Ground Truth derived from pressure sensors in the shoes. The qualitative evaluation is done using the sonification of the heel-strikes: this allows users to hear their footsteps, as detected by the system, in real-time. A user survey polls for the accuracy of the step detection in terms of false positives, ignored steps and delay. Furthermore, the sonification had a secondary goal: do different sounds influence the gait pattern?

2 Background

Gait analysis is an active research area in the rehabilitation engineering and recreational domain. We find a typical rehabilitation example with Parkinson Patients (PP): they suffer from reduced motor control. PP can benefit from reliable gait detection systems for freeze detection. When a freeze is detected, the patient can receive rhythmical audio or tactile feedback; helping them to overcome their freezing condition **Error! Reference source not found.**.

In the recreational domain, gait analysis or step detection can be used for trajectory estimation in personal navigation devices [2]. GPS tracking is often unreliable in obstructed or indoor environments. Therefore, inertial sensors such as gyroscopes and accelerometers are used for estimating walking speed and direction for overcoming the lack of GPS tracking. Personal training devices, for example Nike+, also use gait analysis algorithms for evaluating the users' performance. Applications such as D-Jogger [3], Running Shoe [4] and SynchStep [5] use step detection for dynamic playlist generation, based on the steps per minute (SPM) of the user, reducing the need for typical training playlists and enhancing the personal experience.

While several algorithms for different sensors and sensor locations are available, it remains difficult to choose the best algorithm in different situations. Several factors have to be taken into account: individual gait pattern, walking environment, sensor type, sensor location, computational restraints, etc. While some comparative studies have been performed for specific situations [6], an overview of the subject is not yet available.

The goal of this STSM was to study and compare different algorithms with accelerometers and gyroscopes on several on the human body. For this, we use both quantitative and qualitative evaluation methods. The qualitative evaluation is done using the sonification of the heel-strikes: this allows users to hear their footsteps in real-time, providing subjective feedback about the quality of the algorithm in terms of accuracy and delay.

It is established that sound has an influence on the gait pattern of the user [7]. Research indicated a difference in speed between walking on music and walking on a metronome. This concept can be further evaluated using different sonifications during the evaluation of the step detection algorithms.

3 Work Report

The STSM consisted out of 6 phases:

- **System design**: build a sensor system, including 5 gyroscopes, 5 accelerometers and 2 FSR pressure sensors for synchronized data capture.
- **Data capture**: capture data from 4 persons during an open-air, 1 km walk.
- **Offline Analysis**: analyze the signals with different algorithms for step frequencies (steps per minute, SPM) and heel-strikes for use in sonification. This analysis is quantitative, using in-sole FSR sensors as a Ground Truth.
- **Sonification design**: create a sonification for an algorithm, making use of the gait pattern parameters
- **User evaluation**: 4 subjects walking with the sonification, providing qualitative results about the sonification and step detection algorithms.
- **Secondary research questions**: does the data contain other information coupling the sonification and gait pattern together?

3.1 System Design

3.1.1. Ground truth sensors (reference data)

A good and reliable Ground Truth was necessary for our evaluation. We used Force Sensitive Resistor (FSR) sensors embedded in a shoe-sole for measuring the pressure of the heel applied to the ground. Both sensors were connected with a voltage divider to an Arduino board to the analog pins. The Arduino board was programmed to read the analog pins at 200 Hz and send the data immediately upon readout over the serial link. We developed small driver for reading out the data send over the serial link, where a timestamp was added to each sample. Afterwards, the samples are sent to MaxMSP using the OSC protocol. Note that the samples were time stamped upon arriving on the computer, resulting in a small unaccounted delay.

3.1.2. Body sensors

We used an X-Sens System, composed of 1 X-Sens bus and 5 sensor nodes. Each sensor node features a gyroscope and accelerometer. The sensors are connected to the bus, which polls at regular intervals (from 1 Hz up to 512 Hz) for new data. While the X-sens features orientation output, we preferred the raw and unprocessed data. The bus is connected to the computer using a Serial-over-USB connection. We programmed a data retrieval driver in C++ using the SDK available for the X-Sens system. Like the FSR sensors, each sample is timestamped upon retrieval and send to MaxMSP using OSC protocol.

3.1.3. MaxMSP Patch

All incoming data was centralized in a MaxMSP patch for logging. A highly flexible logger, developed at IPEM for the D-Jogger framework, was used for synchronized logging. Manual comments in the MaxMSP patch, such as the sensor location and subject details, were also logged.

3.1.4. Overview

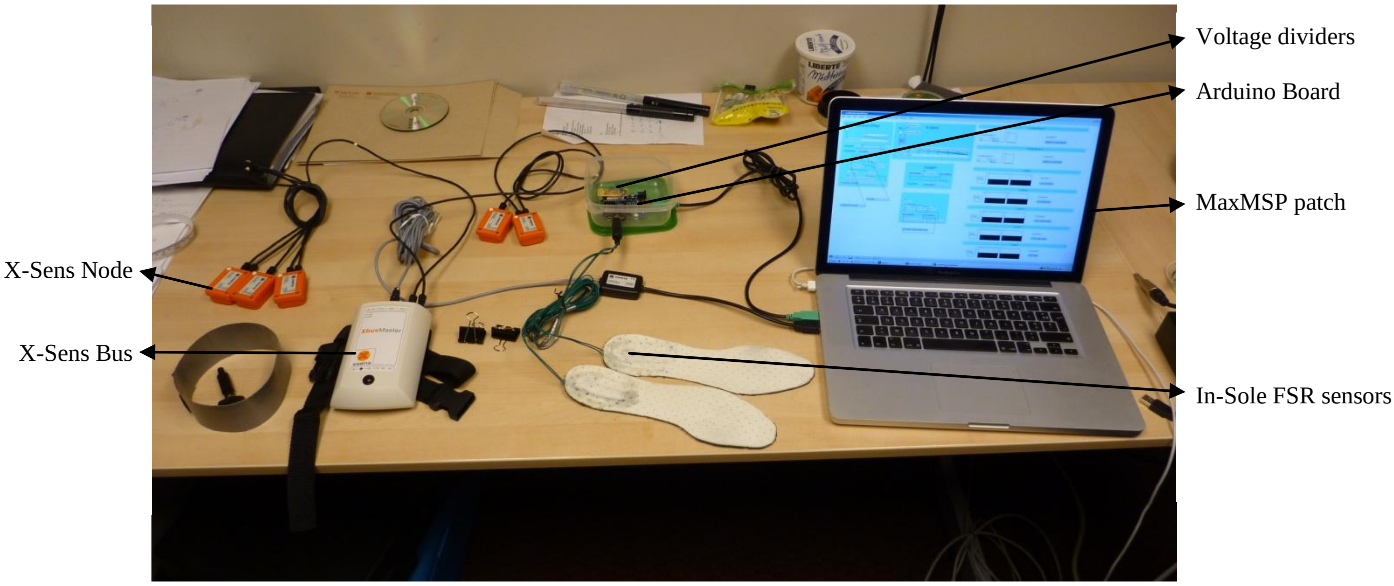

Figure 1 Overview of the data logging setup

3.2 Data Capture

Setup

Figure 2 shows the sensor attachment locations. The X-Sens bus and Arduino board were connected to the computer in the backpack of the user. The loose cables were attached to the clothing to minimize the hinder of the sensor system. The user was not restrained in his/her locomotion using this system.

Experiment

Users were instructed to walk around the building where Music Tech (McGill) is located. The first round, the user is supervised by an instructor, showing clearly the path to follow and explaining the experiment. The second and third round, the user walks autonomous, so the gait pattern is not influenced by the instructor.

The total walked distance is around 1 km, but small variations are possible because of normal pedestrian traffic on the sidewalks. The walk included a downhill and uphill (about 3° inclination) part.

The experiment was done with 4 subjects to a have a diverse dataset to analyze.

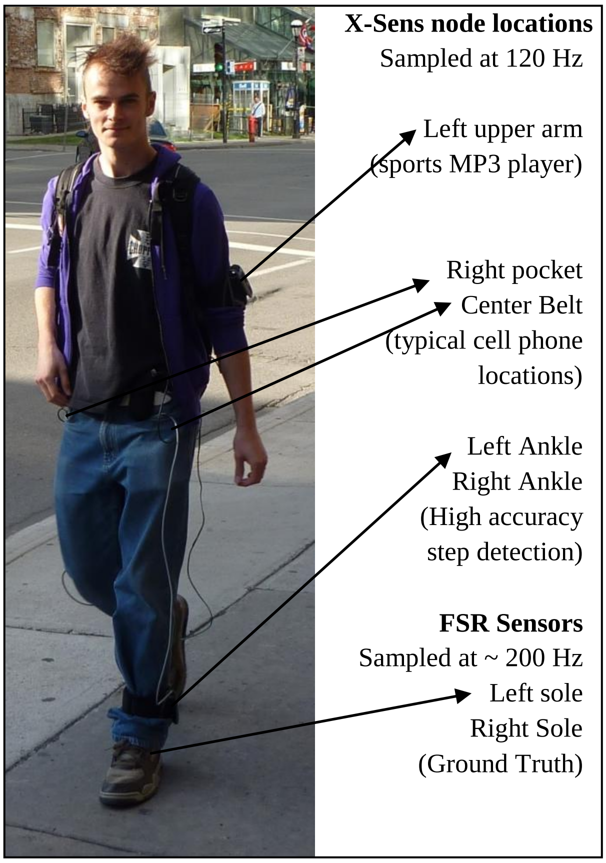

Figure 2 Data capture experiment setup

3.3 Offline Analysis

3.3.1 Ground Truth analysis (heel strike time series and SPM curve)

The FSR signal shows clearly the different phases in the gait pattern. Figure 3 shows a 2 second example of left and right FSR signals during a regular walk. [8] provides a thorough analysis of such FSR signals for Heel Strike (HS), stance, Heel Off (HO) and swing detection. Figure 3 shows these different gait events for the left FSR. A discrete timeseries is created of all HS events in both signals. We calculate the SPM value between two HS by dividing 60000 with the inter-HS-interval, resulting in our SPM Ground Truth (baseline). An example is shown in figure 4. The resulting timeseries and SPM baseline are used to compare the results of kinematic sensors and step/pace detection algorithms.

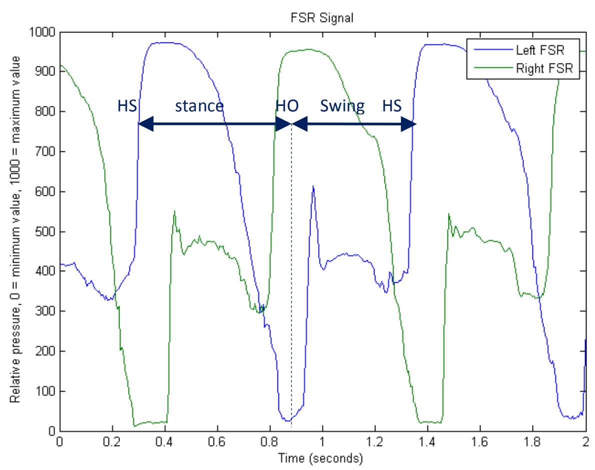

Figure 3: Typical heel FSR patterns.
Gait events are indicated for the left (blue) FSR.

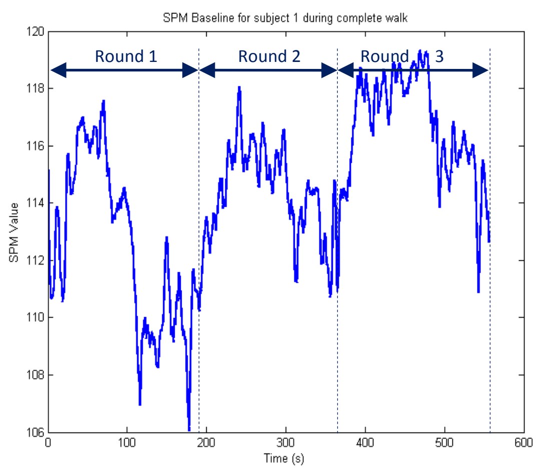

Figure 4: SPM Baseline for subject 1. The 3 different rounds are indicated. The higher SPM was reached during a downhill part of the trajectory, the lower SPM during the uphill part.

3.3.2 Step frequency (SPM curve)

In this section, we describe several methods to get the SPM curve from sensor signals. The resulting curves can be compared to our obtained FSR baseline for analysis. For determining the dominant frequency in a discrete signal, several options are available. In this work, we test three methods, but other algorithms such as an adaptive oscillator, inter-onset time calculations, template matching and DTW, PCA or reservoir computing can also be used.

We limit our search in the range [80 – 200] SPM, or [40 – 100] SPM when ankle signals are used. These values are automatically doubled for the evaluation. All algorithms are applied to the following axis or combinations thereof: $X, Y, Z, \sqrt{(X^2 + Y^2)}, \sqrt{(Y^2 + Z^2)}, \sqrt{(X^2 + Z^2)}, \sqrt{(X^2 + Y^2 + Z^2)}$

3.3.2.1. Fourier transform

We use a Short Time Fourier Transform (STFT) to convert 5 seconds of samples to the frequency domain. A Hanning window is applied to reduce noise; zero-padding is added before and after the samples for increasing the frequency resolution. The highest valid value in the frequency domain is converted to the SPM value. The STFT is applied to all valid intervals [HS – 5 seconds; HS], using the HS timestamps from the baseline. The result is a discrete timeseries with the SPM values at all heel strikes, except those before the 5 second boundary.

3.3.2.2. Autocorrelation

Another approach is to use autocorrelation. We use again 5 seconds of data for each autocorrelation, applied to all valid intervals [HS – 5 seconds; HS], using the HS timestamps from the baseline. The highest correlation lag is converted SPM value for the HS, resulting in another discrete timeseries.

3.3.2.3. Running shoe algorithm

Similar to the autocorrelation, the running shoe algorithm applies a few rules to keep the SPM value in line to avoid spurious values. The signal is low pass filtered before applying the autocorrelation, resulting lag values are weighted with a Rayleigh distribution. The algorithm uses a moving window of 5 seconds, and is applied each 50 ms.

3.3.3 Evaluation of step frequency detection algorithms

Each evaluated axis was compared to the baseline at all detected HS. If no SPM value was found in the sensor SPM result, the value was interpolated using neighboring points. We introduce the following notation:

- *n* is the number of total heel strikes
- s_i is the SPM value as calculated by the algorithm at the timestamp of the i^{th} heel strike
- b_i is the SPM value of the baseline (Ground Truth) at the timestamp of the i^{th} heel strike
- $e_1 = \frac{1}{n}\sum_{i=1\dots n}^{i}(s_i - b_i)$
- $e_2 =$ standard deviation of the differences

We define our evaluation metrics as $e_1 and\ e_2$. We can plot these results in a box plot. An example for the autocorrelation algorithm on the accelerometer can be found in figure 5, indicating clearly the superior results for the left and right ankle. Some axis on the central belt and left upper arm sensors also produce usable results.

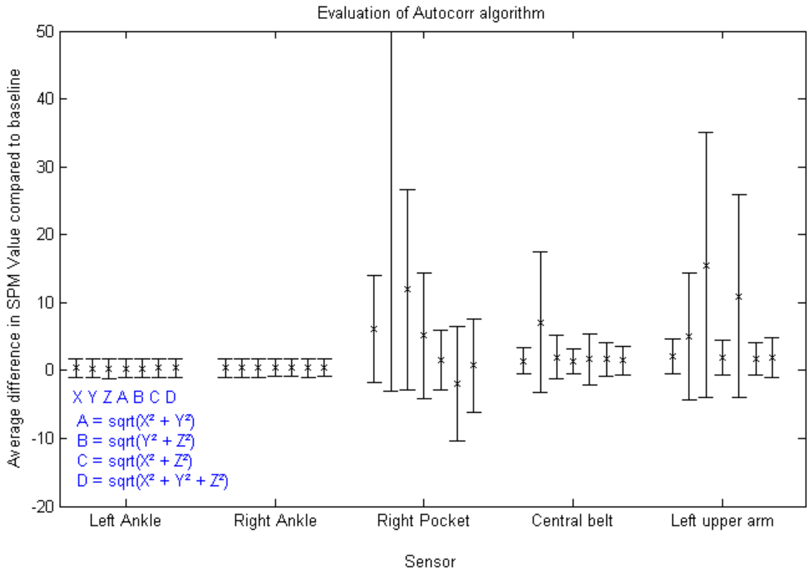

Figure 5: Evaluation of the autocorrelation algorithm, for all accelerometer sensors with subject 1.

Table 1 summarizes the results for all algorithms, averaged over all test subjects. This table gives an overview of the evaluation results of the algorithms, for all sensors and locations. Good results, defined as $\boldsymbol{e_1} < 2$ and $\boldsymbol{e_2} < 10$, are indicated in bold. Optimal location/sensor combinations are marked in light green.

Future work will refine these results with a more in-depth analysis of the data. We will check the distribution of the difference between the reference and the sensor result, and if it is Gaussian or normal we can remove some outliers that now obscure some of the results. We can also use a standardized error measure metric, namely the Normalized Root Mean Square Error (NRMSE), resulting in one number for each sensor/axis/location/algorithm combination. This will enhance the reliability and readability of the results.

Autocorrelation		Right Ankle		Left Ankle		Right Pocket		Central Belt		Left Arm	
		e_1	e_2	e_1	e_2	e_1	e_2	e_1	e_2	e_1	e_2
Accel (fig. 5)	X	0,31	1,93	0,32	1,90	3,66	5,25	2,08	2,78	2,06	2,77
	Y	0,26	1,97	0,24	1,95	3,44	9,46	6,16	20,47	7,09	10,48
	Z	0,38	1,98	0,20	2,06	11,90	15,01	1,20	2,69	16,52	21,86
	XY	0,27	2,02	0,28	1,92	5,14	9,45	1,63	2,27	2,22	4,11
	YZ	0,22	1,96	0,26	1,98	2,39	4,46	0,08	6,59	7,08	12,40
	ZX	0,35	1,95	0,31	1,97	2,38	8,34	1,53	2,42	1,94	3,36
	XYZ	0,29	1,89	0,25	1,91	2,90	6,91	1,45	2,35	2,06	4,23
Gyro scope	X	0,35	1,93	0,34	1,93	70,29	71,47	63,60	68,04	25,56	43,70
	Y	0,41	1,85	0,38	1,91	7,85	18,72	20,79	30,18	8,43	15,91
	Z	0,45	1,92	0,57	1,94	88,86	89,89	60,05	62,71	61,88	70,14
	XY	0,41	1,91	0,38	1,98	91,18	91,80	52,64	55,78	20,42	36,61
	YZ	0,43	1,86	0,52	2,02	40,81	45,07	5,92	8,89	54,94	65,43
	ZX	0,40	1,93	0,52	1,97	10,28	20,64	76,22	78,74	16,23	28,28
	XYZ	0,41	1,90	0,52	2,01	15,44	22,45	54,51	59,57	11,63	26,95
STFT		Right Ankle		Left Ankle		Right Pocket		Central Belt		Left Arm	
		e_1	e_2	e_1	e_2	e_1	e_2	e_1	e_2	e_1	e_2
Accel	X	21,46	22,36	26,29	27,14	0,17	1,62	0,13	1,87	0,03	1,89
	Y	0,40	2,15	0,24	2,13	3,85	6,31	5,91	7,37	1,17	2,44
	Z	0,05	2,35	0,24	2,56	3,48	5,90	-0,02	1,83	16,06	17,57
	XY	0,69	2,26	0,10	2,05	0,19	1,62	0,06	1,84	0,04	2,37
	YZ	0,08	2,09	0,14	1,86	0,29	1,66	0,03	1,85	4,19	5,90
	ZX	0,05	2,21	1,02	2,79	0,16	1,62	0,05	1,85	0,03	1,91
	XYZ	0,69	2,20	0,35	2,34	0,18	1,62	0,04	1,84	0,04	2,41
Gyro-scope	X	0,16	2,04	0,21	2,32	15,15	15,92	21,21	22,30	6,62	11,44
	Y	0,12	2,32	0,36	2,32	0,37	1,62	17,86	19,05	0,80	4,83
	Z	0,11	1,87	0,11	1,83	25,29	26,42	57,01	57,01	0,17	8,66
	XY	0,15	2,07	0,18	1,83	11,99	13,93	39,51	40,36	5,20	9,45
	YZ	0,06	1,86	0,11	2,12	3,24	4,17	6,88	7,95	-0,99	5,04
	ZX	0,31	1,99	1,25	2,86	13,21	13,95	38,51	38,88	12,57	20,64
	XYZ	0,29	1,96	0,91	2,30	13,00	14,89	32,72	33,71	6,41	13,97
Running Shoe		Right Ankle		Left Ankle		Right Pocket		Central Belt		Left Arm	
		e_1	e_2	e_1	e_2	e_1	e_2	e_1	e_2	e_1	e_2
Accel	X	0,43	1,89	0,51	1,91	-0,18	1,91	-0,21	1,93	-0,54	2,08
	Y	0,44	1,93	0,42	1,93	-2,56	4,52	-16,62	17,09	2,88	12,24
	Z	0,60	1,97	0,46	2,27	-11,09	16,36	-0,15	2,01	-4,28	16,82
	XY	0,46	1,93	0,40	1,92	-6,96	8,54	-0,20	1,91	-0,42	2,79
	YZ	0,44	2,01	0,45	1,94	-1,05	3,27	-0,20	1,84	-2,70	13,78
	ZX	0,45	1,88	0,46	1,96	-12,32	13,26	-0,26	1,90	-0,46	2,77
	XYZ	0,45	1,91	0,43	1,94	-12,84	16,94	-0,27	1,89	-4,94	13,21
Gyro scope	X	0,68	1,97	0,79	2,15	40,14	51,70	-29,50	30,43		
	Y	0,52	1,93	0,74	2,06			53,21	55,70	39,80	53,82
	Z	0,75	1,98	0,60	1,90			-29,29	32,05	94,19	94,19
	XY	0,57	1,94	0,74	2,11	12,41	19,95	-29,71	30,27	-3,26	10,27
	YZ	0,72	1,96	0,57	2,00	61,64	61,64	0,15	27,35	2,39	39,79
	ZX	0,72	1,97	0,55	1,93	-1,89	13,41	76,39	76,39	6,81	34,69
	XYZ	0,69	1,95	0,53	2,07	44,78	48,36	81,56	81,56	16,24	33,52

Table 1: results of the step frequency detection. e_1 is the average difference; e_2 is its standard deviation.

Interpretation of the results

- Autocorrelation seems the best method for analyzing the shoe signals for both gyroscopes and accelerometers. They yield close to 0 average difference with our reference, and have a low std.
- For pocket accelerometers and belt accelerometers, the STFT performs the best. It is interesting to see that the combinations of the axis perform better then each axis independently.
- The running shoe algorithm performs very well at the ankles. While the filtering and hacks in the algorithm should improve compared with normal autocorrelation, we noticed they did not.
- The running shoe algorithm did not produce results for some sensors and locations. The blanks in table 1 represent these combinations. They occur only at the most difficult sensor locations.
- STFT yields results closer to the reference then autocorrelation. The reason for this could be that the resolution of the autocorrelation is proportional to the sample rate; while with the STFT the resolution is increased by adding zero padding. Future work will include interpolation of the signal to enhance this resolution.

3.3.4 Heel strike detection (heel strike time series)

In this section, we describe several methods to get the heel strike time series from sensor signals. The resulting curves can later be compared to our obtained FSR HS time series for analysis. We test the algorithms proposed in [x], in addition to adaptive threshold and a gyroscope-specific analysis. All algorithms are applied to the following axis or combinations thereof: $X, Y, Z, \sqrt{(X^2 + Y^2)}, \sqrt{(Y^2 + Z^2)}, \sqrt{(X^2 + Z^2)}, \sqrt{(X^2 + Y^2 + Z^2)}$

3.3.3.1. Peak detection using adaptive threshold

The signal is first centered on 0 by subtracting the mean of the last 5 seconds of samples from the new sample. We define the adaptive threshold:

- When the amplitude of the sample is higher than the threshold, the threshold equals the new amplitude.
- When the amplitude of the sample is lower than the threshold, the threshold value is lowered by a small percentage. This percentage increases gradually each sample, until the threshold rises again (dynamic drop-off).
- The first drop after rising is a potential peak/heel strike.
- When no new potential peak is detected in the following 3 samples, we mark the potential peak as a heel strike. This 3 sample window increases accuracy and reduces false positives, but also increases the real-time reaction time by 3 samples.

Figure 6 shows an implementation of the algorithm in java as a MaxMSP external. The algorithm works real-time, but with a constant delay depending on the peak window. Filter settings such as Moving Average (MA), Low Pass (LPF) and High Pass (HPF) are changeable on the fly, as well as the drop-off factor and the peak window. This implementation is part of the D-Jogger framework.

Figure 6: MaxMSP external for adaptive threshold peak detection

3.3.3.2. Pan-Tompkins method, Template Matching, Peak detection using gait phase windowing, Zero-crossing gyroscope signal: Due to time restrictions, only the adaptive threshold algorithm could be implemented. In future work, the other methods will be evaluated as well.

3.3.5 Evaluation of heel strike detection algorithms

The adaptive threshold algorithm works very well with accelerometer signals from the ankle sensors. The axis for optimal result is the axis perpendicular to the earth, because this signal contains the impact force on the ground of the feet. A thorough evaluation and comparison with the baseline will be done in the future for all algorithms.

3.4 Sonification Design

The goal is to sonificate the heel touchdown, as close to real-time as possible. For this, we decided to use a sample-based approach, mapping several gait parameters and events to the sample parameters. From the evaluation and data analysis; we decided to use the following parameters for the sonification:

- Heel touchdown detected: trigger sample. Signal & algorithm used: Adaptive threshold detection on the ankle sensor accelerations perpendicular to the earth.
- SPM value: playback speed of the sample, using a phase vocoder to avoid pitch changes. Signal & algorithm used: autocorrelation left ankle & right ankle sensors
- Touchdown force: playback volume

For our purposes, we used four different sounds:

- Alternating between a midrange KD and a KD plus a snare drum (SD)
- Low kick drum (KD)
- Recorded high-heel sound. This sound is ambiguous because it has two sharp transients, one being the heel touchdown and the other being the toe touchdown
- No sonification (baseline)

Setup

We used MaxMSP to create a patch, coupling all elements together. We used the autocorrelation algorithm proposed in 3.3.2.2 for SPM determination. The heel-strike detection was done using the adaptive threshold algorithm described in 3.3.3.1 for 2 accelerometers placed at the left and right ankle. Figure 8 shows the MaxMSP patch. A remote control for the control and supervision of the experiment was created on the iPod. The remote control also featured the survey used in the experiment. Computer and iPod were connected using an adhoc wifi network, allowing some space between supervisor and subject.

Experiment

For the second experiment, users had to walk 4 times 100m. Each walk they were offered a different sonification. When walking in the 100m area, the system logged all sample data. After each walk, a survey asked the subject about the quality of the sonification and footstep detection. Three aspects were questioned: experienced delay between footstep and sound; missed detections and false positives. In total, 5 subjects participated in this experiment.

3.5 User evaluation

3.5.1 Accuracy of the step detection

The survey had 2 questions about the accuracy of the step detection. The user had to rate the amount of falsely detected steps, meaning the amount of sonifications that occurred random (‘false positives’). In the second question, users had to rate the amount of steps that were not correctly sonificated, meaning that a step was undetected by the system. In the rating system used, 1 means the best case; no incorrect events detected. 0 means that nothing was correctly detected. The question was repeated for each sonification, but the step detection algorithm did not change. Figure 7 shows the results.

In 10 out of 12 tries, no false steps were detected. 7 out of 12 tries the algorithm detected all steps correctly. A small user inquiry showed that when the detection was not optimal, the sonification sounded distorted. This means a buffer underrun in the MaxMSP sound system, happening when the CPU is occupied with other high-priority tasks (OS-related). In such a case, high jitter and/or sample loss occurs in retrieving sensor samples resulting in delayed or ignored sonifications. We can conclude that users rated the accuracy of step detection very good, with the exception of the buffer underrun issue which was system-related and not algorithm-related.

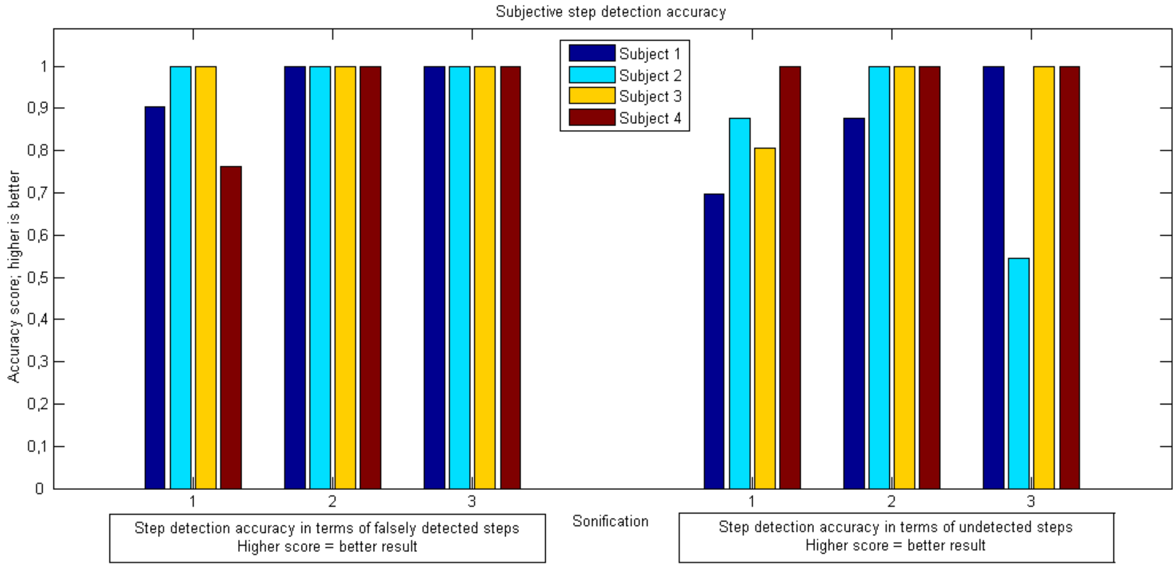

Figure 7: (left) rating of falsely detected steps, (right) rating of undetected steps. 1 means no incorrectly detected events (better), 0 means no correct detected events. The graphs show very good subjective ratings in a real-time test.

3.5.2 Experienced delay of the step detection

Due to the system design, a certain delay is unavoidable. First, there is a small negligible delay during sensor sample retrieval. This delay increases when the CPU is busy with uninterruptable tasks, something that sporadically happens on the used operating system Windows XP. A second delay is due to the step detection algorithm. To make sure that the highest peak is detected, a window of 3 samples is used before marking a point as a peak. The samplerate during the experiment was 100 Hz, resulting in a delay of 30 ms. The final delay is the sound driver buffer. We used ASIO (low latency) drivers. In several test scenarios, a buffer size of 2048 samples at a samplerate of 44100 Hz produced resulted in undistorted sonification. The buffer size resulted in an additional delay of 45 ms. We note that, during the real experiments, the computer ran on battery power, had WiFi enabled and did real-time sample retrieval, resulting in occasional buffer underruns.

A minimum delay of 75ms is thus experienced between the highest acceleration peak during the heel strike and the subject actually hearing the sonification.

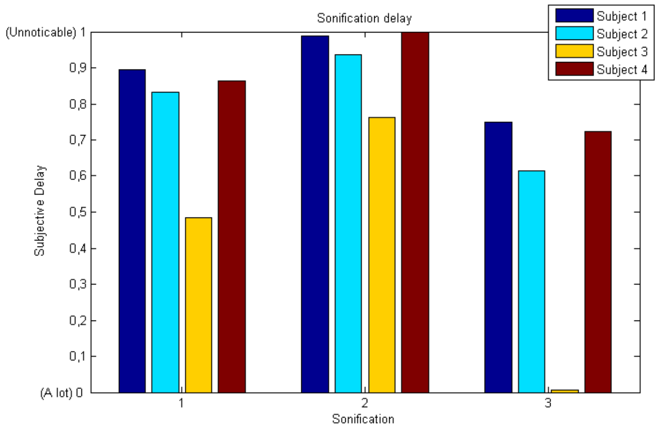

Figure 8: Experienced delay for each subject and sonification

We asked the subjects if they noticed any delay for each sonification. While the system was the same, and only a different audio sample was processed for the sonification, the results varied between each sonification. Figure 8 shows the results.

Global interpretation

There is a minimum delay between heel strike and sonification of 75 ms. However, most subjects rated the delay from 'acceptable' to 'unnoticeable'. This could mean that a step is not experienced as a fixed point in time, but rather as a time interval. This time interval can be between the heel strike and the toe strike. If the sonification is heard in this time interval, users rate it (close to) unnoticeable. We note that subject 3 mentioned he experienced a small, but consistent delay for each step and for all tests.

Sonification specific interpretation

There are significant differences when taking the sonification into account. With sonification 2, the low kick drum, users experienced the least delay. The delay was most noticeable when using the third sonification, the high heel sound.

When we recheck figure 7; we can see a correlation between the transient shape and the experienced delay. When there is a low attack for the transient, the delay is not is audible or perceivable as with a high attack sound. Also, the high-heel sound consisted out of 2 sequential attacks: the first transient was played in the step interval, the second would occur close to or after the toe touchdown. This makes it more noticeable that there is a consistent delay in the system.

3.6 Secondary research questions

3.6.1 Sonification influence on the gait stability

The use of different sonifications allows us to analyze the influence of the sonification on the gait pattern of the user. In this case, we look specifically for the gait stability. First, we plot the data in a box plot, visualizing the standard deviation, average and outliers of the SPM timeseries. Figure 12 shows the results. Second, we define gait stability as the variance of the SPM timeseries. The stability for each person and each sonification is displayed in figure 13.

Both figures show no significant correlations between the stability and the sonification. This is contradictory to what we expected: the clearer the transient, the more stable the gait pattern. We assumed this because we are used to rhythmical auditory stimuli, which could have resulted in some kind of desire to obtain a stable rhythmical auditory pattern.

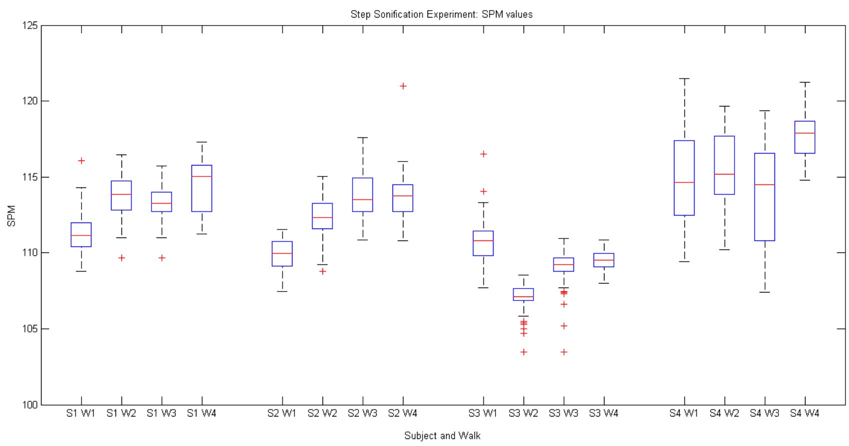

Figure 12: Box plot of the SPM timeseries for each test subject (S) and walk/sonification (W).

The data however was not conclusive to this end. The experimental setting was not optimized for this task: users were not informed of this goal so they tend (especially subject 4) to experiment with the system. In future experiments, the participants should be informed that they should try to walk as stable as they can. In this case, we could see if the sonification helps them by also using the auditory senses.

3.6.2 Sonification influence on the walking speed

We also checked a possible correlation between the walking speeds, expressed in km/h, with the sonification. Figure 14 shows the results. Unfortunately, they are again inconclusive. However, every subject has its own preferred speed and does not seem to deviate much from that.

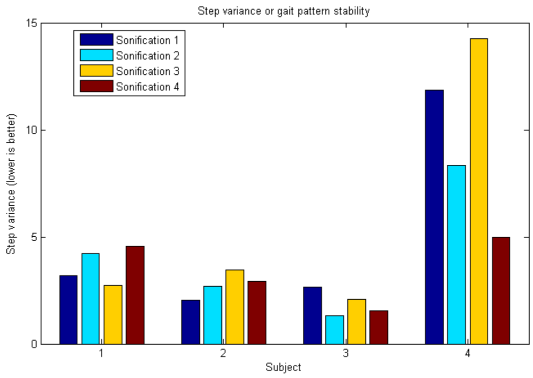

Figure 13: Gait stability or step variance for each sonification and subject

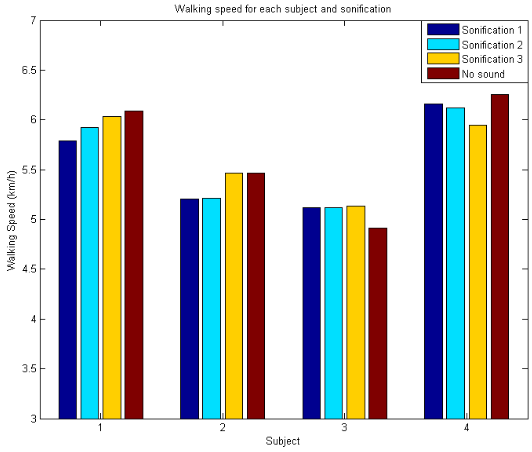

Figure 14: Walking speed for each sonification and subject.

4 Conclusion

In this work, we evaluated several step frequency and heel strike detection algorithms for accelerometers and gyroscopes. We analyzed 5 locations on the human body: ankles, hip, pocket and upper arm. Using a FSR sensor as a ground truth, we came to the following conclusions:

- On the ankles, autocorrelation resulted in stable step frequency detection on all axes.
- On the hip, Fourier transform performed best for step frequency detection with an accelerometer.
- Other locations have several usable axes, for both autocorrelation and Fourier Transform.
- The upper arm and pocket locations prove to be the most difficult to analyze. The upper arm is not moved continuously during the test, resulting in a lot of outliers. When taking these into account, reliable results can be obtained.
- Real time Heel strike detection can be performed at the ankles, using accelerometers and an adaptive peak detection algorithm. The peak detection resulted in a delay of 30 ms, however this. Results could be further improved by also using the gyroscope for limiting the peak search window.

In the second part of the STSM, a qualitative evaluation is performed using sonification of the heel-strikes. Users reported a high accuracy of the adaptive threshold algorithm, as well as an acceptable delay. Users also seem to experience their heel strike as a time interval between heel touchdown and toe touchdown.

Additionally, we checked possible correlations between several sonifications and gait parameters such as step variability and walking speed. No correlations were found however, possibly due to an incomplete experimental setting for the secondary goals.

5 Future work, possible collaborations and publications

The comparative study, when finished with more algorithms for both step frequency and heel strike detection, is very well suited for a journal publication. The study will continue at the IPEM institute.

The developed framework can also be used for a game-like application. In future collaboration with Jason Hockman (McGill, DDML), we will develop an application that rewards stable gait patterns with layers of additional sounds. The main goal is to demonstrate the research and sonification options, but a conference paper with the application is also possible.

Instead of using a sample-based sonification, we can also use real footstep synthesized sonification. A project at Medialogy (Aalborg University Copenhagen) by Rolf Nordahl, Stefania Serafien and Luca Turchet, currently uses microphones to capture the ground pressure force, resulting in an envelope used by the synthesized sound for realistic sonification of footsteps [9]. A possible collaboration, linking the D-Jogger framework and these results with their work is currently being discussed.

References

[1] M. Suteerawattananon, G. S. Morris, B. R. Etnyre, J. Jankovic, E. J. Protas, “Effects of visual and auditory cues on gait in individuals with Parkinson's disease”, *in Journal of the Neurological Sciences, Volume 219, Issues 1-2, 15 April 2004, Pages 63-69*

[2] H.-J Jang, J.W. Kim and D.-H. Hwang, “Robust step detection method for pedestrian navigation systems”, *Electronics Letters, vol. 43, no. 14, 2007.*

[3] B. Moens, L. Van Noorden and M. Leman, "D-Jogger: Syncing music with walking", *In Proceedings of the 2010 SMC Conference, 2010*

[4] J. Hockman, M.M. Wanderley and I. Fujinaga, “Phase vocoder manipulation by runner’s pace”, *In Proceedings of the 2009 Conference on New Instruments for Musical Expression, pp. 90-93, 2009.*

[5] G.T. Elliott, B. Tomlinson, “PersonalSoundtrack: context-aware playlists that adapt to user pace”, *CHI '06 extended abstracts on Human factors in computing systems, 2006.*

[6] M. Marschollek, M. Goevercin, K.H. Wolf, B. Song, M. Gietzelt, R. Haux and E. Steinhagen-Thiessen, “A performance comparison of accelerometry-based step detection algorithms on a large, non-laboratory sample of healthy and mobility-impaired persons”. *Conference Proceedings IEEE Eng Med Biol Soc 2008, pp. 1319-1322, 2008.*

[7] F. Styns, L.V. Noorden, D. Moelants and M. Leman, “Walking on music”, *in Human Movement Science, Vol. 26, Issue 5, pp. 769-785, 2007.*

[8] P. I. Pappas, M. R. Popovic, T. Keller, V. Dietz, and M. Morari, "A Reliable Gait Phase Detection System", *in IEEE Transactions on neural systems and rehabilitation engineering, Vol. 9, No. 2, June 2001*

[9] R. Nordahl, S. Serafin and L. Turchet, ”Sound Synthesis and Evaluation of Interactive Footsteps for Virtual Reality Applications”, *in Proceedings of IEEE Virtual Reality 2010. Vol. 1 1. udg. IEEE conference proceedings, 2010. s. 148-153 (2010)*