Funky Sole Music: Gait Recognition and Adaptive Mapping

Kristian Nymoen, Sichao Song, Yngve Hafing, and Jim Torresen
Department of Informatics
University of Oslo
Norway
{krisny, sichaos, yngveha, jimtoer}@ifi.uio.no

ABSTRACT

We present *Funky Sole Music*, a musical interface employing a sole embedded with three force sensitive resistors in combination with a novel algorithm for continuous movement classification. A heuristics-based music engine has been implemented, allowing users to control high-level parameters of the musical output. This provides a greater degree of control to users without musical expertise compared to what they get with traditional media playes. By using the movement classification result not as a direct control action in itself, but as a way to change mapping spaces and musical sections, the control possibilities offered by the simple interface are greatly increased.

1. INTRODUCTION

In music technology, a clear distinction has traditionally been made between the performer creating the music on a musical instrument, and the perceiver receiving the music through a music playback device [18]. Increased development efforts in music technology in the past few decades have started to blur this clear distinction. Several examples exist of applications where people can play a musical instrument without needing the skill of a professional performer, for instance musical instruments like Smule's Ocarina [22] and Magic Fiddle [23], or music games, such as Guitar Hero [9]. Not only musical instruments have changed, but also music players. Where people only used to be able to have simple control options such as play, pause, skip, and volume, they may now take use of social features or recommendation services in applications such as iTunes or Spotify. As the clear distinction between instruments and playback devices is blurred out, a continuum emerges on which the two are opposite extremes (Figure 1), between these extremes are what we call *active music technologies*.

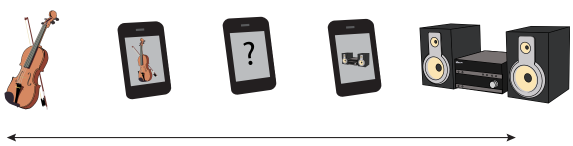


Figure 1: A continuum between musical instruments and music playback devices. In the middle we find *active music technologies*.

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

With the increased availability of motion tracking technologies, there has also been an increased effort towards the use of machine learning algorithms to recognise control actions of users. Such algorithms have been applied to control musical instruments (e.g. [7, 1, 8]), gesture based sound selection from a database [3], and study of instrumentalists' sound-producing actions [4].

Our paper presents a new active music device, using a sensor sole and a novel movement classification algorithm to continuously recognise the movement pattern of the user. Movement patterns could be different gaits, such as running or walking straight or sideways, or other activities such as foot-tapping or jumping. Being an active music device, where the user can influence the music to some extent, the *mapping* between control actions and sound output is an essential part of the decive. In our implentation, the continuous classification of movement is utilised to change the mapping space itself. With this adaptive mapping, the quite simple interface consisting of three force sensitive resistors (FSR) in a sole is given a larger range of control possibilities than it would if the mapping between control actions and musical parameters were fixed.

In the next section, we will introduce previous use of footworn interfaces in music, before moving on to describing our implementation in Section 3. The device and our approach is discussed further in Section 4 before we conclude and present our plans for future extensions of the work in Section 5.

2. BACKGROUND

Several researchers have explored foot-worn sensor systems for sound interaction. Among the first to explore this mode of interaction was Joe Paradiso and colleagues [17] who in 1997 presented a pair of dancing shoes embedded with piezoelectric pads, FSRs, accelerometers, and compasses. These sensors allowed tracking the wearer's feet in a number of dimensions, including foot pressure, orientation, acceleration, and deformation. Additionally, a laser rangefinder combined with ultrasound sensing detected the horizontal position of the shoes on a stage, and electric field sensing was applied to detect the vertical position of the shoes. The device was developed further, and an upgraded version using a jogging sneaker was presented in [15] and [16], along with a discussion of its musical application. The same system has later also been applied for medical purposes as a low-cost alternative to expensive motion capture systems in gait analysis [14].

Another approach to footworn sensors is their application in virtual reality. Choi and Ricci used force sensors combined with fuzzy logic to detect different gaits [5], and Turchet used sandals embedded with force sensors and actuators to study a range of aspects related to audio-haptic feedback of foot-step sounds [21]. Accelerometers mounted in shoes have been applied to adjust the tempo of audio files by a phase vocoder [10, 13] and for selecting tempo-tagged songs from a database [12]. Similar systems have also been developed for both hand-held and arm-worn accelerometers [6, 2].

Our system has found great inspiration in the above mentioned systems, and extends previous work by adapting the mapping space to the current movement pattern of the user.

3. IMPLEMENTATION

This section covers the implementation of our system: Controller and sensor interface, music engine, the machine learning algorithm for gait recognition, and adaptive mapping based on the gait classification.

3.1 Sensors and Interface

Our prototype consists of a sole made from rubber foam, with three Interlink 402 FSRs attached with duct tape, see Figure 2. One sensor is placed below the heel, and the two other in the front of the sole on each side, to capture sideways tilting of the foot. The FSRs are connected by cable to our General Purpose Sensor Platform (GPSP), which samples the sensors and passes the sensor data on in the Open Sound Control format via WLAN. Various filtering and thresholding can also be applied on the GPSP. The GPSP enclosure is 3D-printed and fitted with a strap made from elastic rubber and velcro, allowing attachment of the GPSP to the leg of the user. Further documentation and results from performance tests of the Arduino-based GPSP, along with links to STL files and assembly instructions for the enclosure, is available in [20].

Figure 2: The sensor sole with three force sensitive resistors attached with duct tape (top) and a sandal connected to the GPSP interface with exposed electronics (bottom).

3.2 Music Engine

The music engine of our system has been implemented using Max[1] and Reason.[2] To demonstrate the system, we have implemented one song with two parts. Part A is a twelve-measure blues chord progression, and part B is a steady tonic chord.

The sounds are generated by four instruments in Reason:

1. Drum loop (Dr. Octo Rex loop player)
2. Guitar loop (Dr. Octo Rex loop player)
3. Bass guitar (Subtractor synthesiser)
4. Wurlitzer (NN19-sampler)

The Reason loop player uses a prerecorded loop sample which has been processed by slicing it into individual samples for each onset. As such, the loop is not played back as one continuous sound file, but by triggering the individual samples at given points in time. This facilitates tempo adjustments for the two loop-based instruments. Two drum loops and guitar loops have been implemented, one for each part of the song.

The main control of the music occurs in Max. A phasor∼ object is used to keep track of tempo in Max and a corresponding BPM value is sent to the Reason loop players. Thresholding the output from the phasor∼ object enables triggering of events at various times in each measure. Simple probabilistic heuristics for parts A and B have been defined, sending MIDI-events to the reason engine and triggering tones from the bass guitar and wurlitzer at certain times in each measure. For instance, at the first beat of each measure, the bass guitar has only two tones to choose from: The root note in one out of two octaves. The remaining beats of each measure have a certain probability of being skipped, and a wider range of selectable tones. Additionally, there is a probability of triggering bass tones on the shuffled 16ths. Naturally, such heuristics hardly imitate how a real bass player would play, but the music is less static than a hard-coded bass pattern would be, and since the heuristics are designed by hand, some degree of musical coherence is ensured. Further, by adjusting the probability levels, interesting dynamic changes can occur in the music.

3.3 Movement Recognition

In order to perform continuous classification of movement patterns, we apply a movement recognition algorithm inspired by the concept of pheromones in ant colony optimisation. The ant learning algorithm (ALA) has been shown to work efficiently with only one training instance, with better recognition rates than Hidden Markov Models and similar rates to Dynamic Time Warping, outperforming both of these in terms of execution time. The details of the algorithm and various test results for its application to accelerometer data have previously been presented in [19]. We will here describe how the previous paper was adjusted to work with the sensor sole. Below, we present how the force data from the sensors is quantised into a set of basic states, and further how the sequence of such states are recognised as an ongoing movement pattern by the ALA algorithm.

3.3.1 Vector Quantisation

To be able to recognise a movement pattern, a set of four "protostates" have been defined. Figure 3 shows how the state of the foot can be either *FullRelease*, *ToePress*, *HeelPress*, or *FullPress*. The system is calibrated by a short recording of each state, to obtain a characteristic data vector for each of them. After calibration, the data is continuously mapped onto one of the four different states using a nearest neighbour algorithm, and consequently a movement pattern is represented as a sequential order of these characteristic states.

[1] `http://www.cycling74.com`
[2] `http://www.propellerheads.se/reason/`

Figure 3: The data is quantised into four states: 1-FullRelease, 2-ToePress, 3-HeelPress, 4-FullPress

Force data from a walking sequence with 22 steps is shown in Figure 4. The top plot shows the raw data from the sensors, and the lower plot shows the sequence of characteristic states. The repeating pattern in this sequence is: *FullRelease*, *HeelPress*, *FullPress* and *ToePress*.

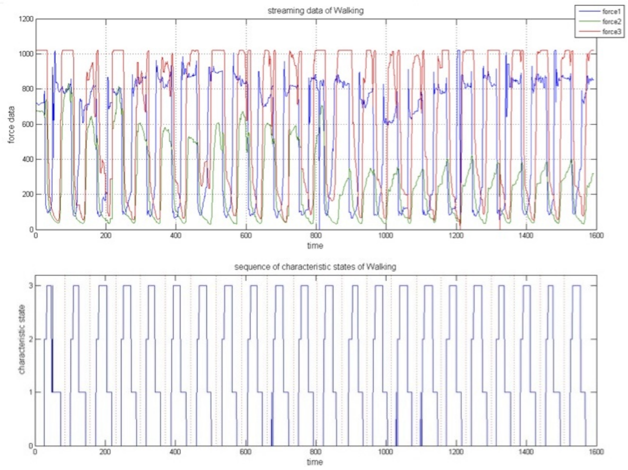
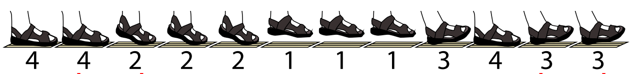


Figure 4: The top plot shows the 10 bit data from the three force sensors. The lower plot shows the quantised version with four states.

3.3.2 Ant Learning Algorithm

The foundation of classification in the ALA algorithm is a *pheromone table*, inspired by the pheromone mechanism in ant colony optimisation, where an ant leaves a pheromone trail as a record of its path. While ants in nature track a path (i.e. a sequential order of positions), our tracking is of the sequential order of characteristic states, counting the number of times each transition between states occurs. Separate pheromone tables are trained for each type of movement that is to be recognised. An example of a simplified (very short) walking pattern is shown in figure 5. For every two successive frames, a corresponding increment is found in the pheromone table below. When one state is followed by the same state, the corresponding value along the diagonal is incremented, when the state is different from the previous, one of the other cells are incremented.



previous state \ current state	1-FullRelease	2-ToePress	3-HeelPress	4-FullPress
1-FullRelease	2	0	1	0
2-ToePress	1	2	0	0
3-HeelPress	0	0	1	1
4-FullPress	0	1	1	1

Figure 5: A simplified example of a short walking sequence made from 12 successive states. Below is the corresponding *pheromone table*.

The algorithm is trained by recording a short sequence of of each movement pattern to be recognised, for instance a 30 second walking sequence. A pheromone table corresponding to the recording is generated. Additionally, the training data is split into shorter segments and one pheromone table is created for each segment. This provides a set of slightly different tables which all correspond to the same type of movement. We calculate the distances $C = c_1, c_2, \ldots$ between all pheromone tables that correspond to the same movement pattern.

$$c = \sum_{i=1,j=1}^{4} (\tau^1_{i,j} - \tau^2_{i,j})^2 \tag{1}$$

Where $\tau^1_{i,j}$ and $\tau^2_{i,j}$ represents the fields of two tables to be compared. The mean and standard deviation of C provides an indication of how much a pattern varies, and is used to deterimine a confidence interval for the realtime classification.

A fixed size moving window is used in the online movement recognition, and a pheromone table is generated for each window. The current table is compared to the set of trained pheromone tables using Equation 1, and classified accordingly if the distance to the table falls within the confidence interval of a learned movement pattern.

3.4 Adaptive Mapping

A traditional way of implementing mapping between a controller and sound engine is through a direct mapping between the available control parameters and sound parameters, often through a set of layers to create a complex many-to-many relationship [11]. In our system, we want the user to have control over many aspects of the musical output, and a direct mapping between the three force sensors and the musical engine would run the risk of being too simple, even with complex many-to-many mappings. For this reason, we use the classifications made by the ALA algorithm to change between different sections of the music, and also between different mapping spaces. In the prototype, three movement patterns are used to illustrate the concept:

Walking: While walking, the tempo of the music follows the footsteps of the user. All other parameters are predefined and the twelve measure blues is played according to the probability-based approach described in Section 3.2.

Tapping back: When tapping the heel to the floor, the tempo control is disabled. A wah-wah effect is enabled and controlled by the data from the force sensors. Also, the probability threshold of bass note onsets is controlled by the overall change in sensor data, and thus the bass activity level increased with increased foot activity.

Tapping front: Tapping with the front of the foot takes the music to a new section. In this section, a wurlizer solo is enabled when the front of the foot touches the floor.

4. DISCUSSION

The presented Funky Sole Music protoype allows a more varied control over a piece of music than what is provided by traditional media players. At the same time, the control space is far more limited than in a traditional musical instrument, e.g. without any possibility of playing out of tune or out of sync. The restrictions on the musical output ensures a certain degree of musical coherence, but can also be argued to diminish the possibilities for musical expressivity. This however, is only when compared with a traditional musical instrument — to a non-musician who would never touch a musical instrument, Funky Sole Music provides an increased possibility for musical expression compared to normal media players.

Even though the device is more of an active media device than a musical instrument, a principle is shown which could also be fruitful in development of new musical instruments. Instead of only *direct* links between control data and synthesis parameters, we can create digital musical instruments where more *indirect* control actions determine the effect of direct control actions by manipulating the mapping space. Such indirect control actions can for instance be the current state of the performer or the audience. Ranging from simple examples such as the current location of the performer on stage to the more advanced classification of the mood of a person through biosensors.

5. CONCLUSION AND FUTURE WORK

We have presented an interactive music system for high-level control of a musical piece. A novel movement recognition algorithm is applied to continuously let the user move between different mapping spaces. A video example of the system in action is available online.[3]

In future work, we aim at developing the system further towards a more generic device for active music. The current implementation of adaptive mapping is strongly connected with one particular piece of music, but future versions should enable users to select different songs, and to employ the same control actions to other musical pieces. Future developments should also include usability testing both in controlled environments and in potential use scenarios such as workout sessions and interactive installations.

Acknowledgments

This research has received funding from EU FP7 under grant agreement no. 257906, Engineering Proprioception in Computer Systems (EPiCS).

6. REFERENCES

[1] F. Bevilacqua, B. Zamborlin, A. Sypniewski, N. Schnell, F. Guédy, and N. Rasamimanana. Continuous realtime gesture following and recognition. In *Gesture in embodied communication and human-computer interaction*, pp. 73–84. Springer, 2010.

[2] J. T. Biehl, P. D. Adamczyk, and B. P. Bailey. Djogger: A mobile dynamic music device. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pp. 556–561. ACM, 2006.

[3] B. Caramiaux, F. Bevilacqua, and N. Schnell. Sound selection by gestures. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, pp. 329–330, Oslo, Norway, 2011.

[4] B. Caramiaux, M. M. Wanderley, and F. Bevilacqua. Segmenting and parsing instrumentalists' gestures. *Journal of New Music Research*, 41(1):13–29, 2012.

[5] I. Choi and C. Ricci. Foot-mounted gesture detection and its application in virtual environments. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 4248–4253 vol.5, 1997.

[6] G. T. Elliott and B. Tomlinson. Personalsoundtrack: context-aware playlists that adapt to user pace. In *CHI'06 extended abstracts on Human factors in computing systems*, pp. 736–741. ACM, 2006.

[3]`http://vimeo.com/74219398`

[7] R. Fiebrink, D. Trueman, and P. R. Cook. A meta-instrument for interactive, on-the-fly machine learning. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, Pittsburgh, PA, 2009.

[8] N. Gillian and J. A. Paradiso. Digito: A fine-grain gesturally controlled virtual musical instrument. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, Ann Arbor, MI, 2012.

[9] Harmonix. Guitar hero (software). Red Octane, 2005.

[10] J. A. Hockman, M. M. Wanderley, and I. Fujinaga. Real-time phase vocoder manipulation by runner's pace. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, Pittsburgh, PA, 2009.

[11] A. Hunt and M. M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(2):97–108, 2002.

[12] N. Masahiro, H. Takaesu, H. Demachi, M. Oono, and H. Saito. Development of an automatic music selection system based on runner's step frequency. In *Proc. of the Int. Conf. on Music Information Retrieval*, pp. 193–198, Philadelphia, PA, 2008.

[13] B. Moens, L. van Noorden, and M. Leman. D-jogger: Syncing music with walking. In *Proc. of the Sound and Music Computing Conference*, pp. 451–456, Barcelona, Spain, 2010.

[14] S. J. Morris and J. A. Paradiso. Shoe-integrated sensor system for wireless gait analysis and real-time feedback. In *Engineering in Medicine and Biology* , volume 3, pages 2468–2469. IEEE, 2002.

[15] J. Paradiso, K.-Y. Hsiao, and E. Hu. Interactive music for instrumented dancing shoes. In *Proc. of the International Computer Music Conference*, pp. 453–456, Beijing, China, 1999.

[16] J. A. Paradiso, K.-Y. Hsiao, A. Y. Benbasat, and Z. Teegarden. Design and implementation of expressive footwear. *IBM systems journal*, 39(3.4):511–529, 2000.

[17] J. A. Paradiso and E. Hu. Expressive footwear for computer-augmented dance performance. In *Int. Symposium on Wearable Computers*, pp. 165–166. IEEE, 1997.

[18] C. Small. *Musicking: The meanings of performing and listening*. Wesleyan, 1998.

[19] S. Song, A. Chandra, and J. Torresen. An ant learning algorithm for gesture recognition with one-instance training. In *IEEE Congress on Evolutionary Computation*, pages 2956–2963, 2013.

[20] J. Torresen, Y. Hafting, and K. Nymoen. A new wi-fi based platform for wireless sensor data collection. In *Int. Conf. on New Interfaces For Musical Expression*, pages 337–340, Seoul, Korea, 2013.

[21] L. Turchet. *Audio-Haptic Feedback for Natural Interactive Walking: Interfaces, Simulations & Perceptual Phenomena*. PhD thesis, Aalborg University Copenhagen, 2013.

[22] G. Wang. Designing smule's ocarina: The iphone's magic flute. In *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, pp. 303–307, Pittsburgh, PA, 2009.

[23] G. Wang, J. Oh, and T. Lieber. Designing for the ipad: Magic fiddle. In A. R. Jensenius, A. Tveit, R. I. Godøy, and D. Overholt, editors, *Proc. of the Int. Conf. on New Interfaces for Musical Expression*, pp. 197–202, Oslo, Norway, 2011.