Handwaving: Gesture Recognition for Participatory Mobile Music

Gerard Roma
Georgia Institute of Technology
Atlanta, GA
gerard.roma@gatech.edu

Anna Xambó
Georgia Institute of Technology
Atlanta, GA
anna.xambo@gatech.edu

Jason Freeman
Georgia Institute of Technology
Atlanta, GA
jason.freeman@gatech.edu

ABSTRACT

This paper describes *handwaving*, a system for participatory mobile music based on accelerometer gesture recognition. The core of the system is a library that can be used to recognize and map arbitrary gestures to sound synthesizers. Such gestures can be quickly learnt by mobile phone users in order to produce sounds in a musical context. The system is implemented using web standards, so it can be used with most current smartphones without the need of installing specific software.

KEYWORDS

mobile music, gesture recognition, audience participation

ACM Reference format:
Gerard Roma, Anna Xambó, and Jason Freeman. 2017. Handwaving: Gesture Recognition for Participatory Mobile Music. In *Proceedings of AM '17, London, United Kingdom, August 23–26, 2017,* 6 pages.
DOI: 10.1145/3123514.3123538

1 INTRODUCTION

During the last few decades, the introduction of computers in music performance has had an important effect on the expectations of the audience. As noted by Keislar [10], the history of computer music has been one first of gradually abstracting the production of sound away from the body and, more recently, attempting to reconnect the body and the physical gesture to sound once again. For music where computers are involved, the audience no longer expects to see every detail of the music creation process, and it is typically assumed that performers could be checking their email. While the aesthetics of acousmatic music have reached popular culture and some musicians prefer to perform in total darkness, some sort of interaction between performers and audience usually hints that music is being performed live.

With the ubiquity of smartphones and mobile data, the situation has become even more complex: now the audience can check their email too. There is, in this sense, an opportunity for technologies that enhance interaction between audience and performers, rather than distracting each party. The standardization of mobile technologies and the recent improvements of web standards have greatly increased the potential for audience participation in music performances. Research on audience participation, initially an aesthetic pursuit, is now of interest to a wider community of computer-mediated music practitioners. While some research has focused on the use of smartphones for capturing and documenting music performances [12], their use for participation and interaction with the music creation process has a unique potential for engaging audiences. In the extreme case, the distinction between audience and performer can be eliminated, and a music performance can be entirely designed as an audience-driven process.

In this paper we describe *handwaving*, a system for enhancing audience participation in music performance through mobile phone technologies. We exploit the fact that many people carry a device capable of sound synthesis and equipped with an accelerometer sensor. Our system allows the definition of a vocabulary of gestures, which should be easy to learn by music performance audiences. Given some examples, generated in advance, a machine learing model is trained to recognize the corresponding gestures, which are mapped to different sound synthesizers. The system is implemented using web standards, which make it simple and quick to deploy software on audience devices in live performance settings. The training interface is also implemented as a web application, which allows a group of people to provide training examples, as opposed to having a single individual train the system. We expect training data from multiple users to produce more robust models for audience participation.

2 MOBILE AND PARTICIPATORY MUSIC

The idea of *mobile music* started taking shape before the introduction of smartphones, on the basis of increasing sound capabilities of mobile phones and PDAs [19]. Initial research

AM '17, London, United Kingdom

DOI: 10.1145/3123514.3123538

focused on exploring the space of interaction design enabled by the different available sensors [4], as well as by social interaction enabled by ubiquity [19]. While initial mobile orchestra performances also preceded smartphones [21], the standardization and ease of programming associated with them fostered the popularization of mobile orchestras following the tradition of laptop orchestras [17].

During the last few years, much research on mobile music has focused on audience participation. For example in *echobo* [14], the audience of a music performance was able to interact with a "master musician", playing along with an acoustic musician. In *massMobile* [7], the sound of the performance was centralized and users could interact with the system through a web client/server architecture. A similar centralized system with web control was implemented in *Swarmed* [8]. The recent development of web standards, which are rapidly implemented in mobile browsers, has greatly simplified the problem of audience participation. In addition to Web Audio, many standardized capabilities and sensors are now available to web applications, such as acceleration, vibration, or location. Recent research has focused on web-based participation [23]. As an example, the participatory concert of the second Web Audio Conference showcased a number of approaches for audience participation using smartphones and web standards [2, 9, 13, 16, 20].

3 GESTURE RECOGNITION

Body language, and particularly hand gestures, are an important part of human and animal communication. Since the popularization of three-axis accelerometers, first in game controllers such as the Wii remote, and then in smartphones, gesture recognition has become relevant to many applications, such as mobile user interfaces. Gesture recognition has also been widely used for artistic performance [3]. Most traditional approaches use either Dynamic Time Warping (DTW) [15] or Hidden Markov Models (HMM) [11]. It is also common to experiment with other common classifiers, such as K-Nearest Neighbors (KNN) or Support Vector Machines (SVM) [22]. These procedures usually require careful segmentation and annotation of gestures, and their evaluation is often confined to laboratory experiments.

Neural networks and deep learning methodologies have quickly become the mainstream method for machine learning. By stacking several types of neural network layers, deep learning systems are able to learn intermediate representations from data, thus reducing the amount of expert knowledge required. Like in other domains, deep learning techniques have been applied to smartphone sensors [18]. In this paper, we describe an application of neural networks for mobile accelerometer gesture recognition. The data used for training the network is not manually segmented, thus reducing the need of manual annotation.

Some works have provided tools for exploiting machine learning algorithms in music performances, using personal computers[1, 6]. Contrastingly, our system focuses on recognizing gestures and mapping them to musical sounds directly on a mobile phone, without the need of a PC or laptop. We propose an implementation based on web standards, which makes it very easy to quickly engage casual participants in music performances as well as other settings such as installations or museums.

4 SYSTEM DESCRIPTION

Repetitive Gestures

Our system is based on recognition of simple gestures, which are commonly associated to discrete events. Simple gestures are slowly making their way to mobile interaction, for example, shaking (for undoing something) is the only non-touch gesture in Apple's IOS human interface guidelines.[1] A "double twist" gesture for activating the camera was introduced in version 7.0 of Android. For music contexts, discrete events signaled by gestures can be useful, but associating specific gestures with the resulting musical events may take some time to users, especially in the context of audience participation. In addition, gestures for specific tasks must assume a "silent" (i.e. no gesture) background, while in music very diverse regimes of action vs inaction may be used. For these reasons, we opted for a general continuous recognition model, which includes a silence gesture.

This also means gestures are recognized in relatively short temporal windows (e.g. 2 seconds), and have no definite phases (no onset or offset). The system thus focuses on oscillatory movements, which in our experiments have been mostly limited to simple repetitive movements along each accelerometer axis (up-down, left-right) and other basic movements (twisting and circular movement). Figure 1 shows some examples.

Recognition Framework

In terms of machine learning, the system is relatively straightforward. Accelerometer data consists of three coordinates, x, y, z, that represent acceleration of the phone in each dimension. All three signals are analyzed using the Short-Time Fourier Transform (STFT) and stacked to form a feature vector. The data is fed into a neural network with one hidden layer, using sigmoid activations. The hidden layer has the same number of units as the input. A final softmax layer is used to predict the gesture class. For training, we use half-window overlaps, while in the test stage, data is analyzed

[1] https://developer.apple.com/ios/human-interface-guidelines/interaction/gestures/

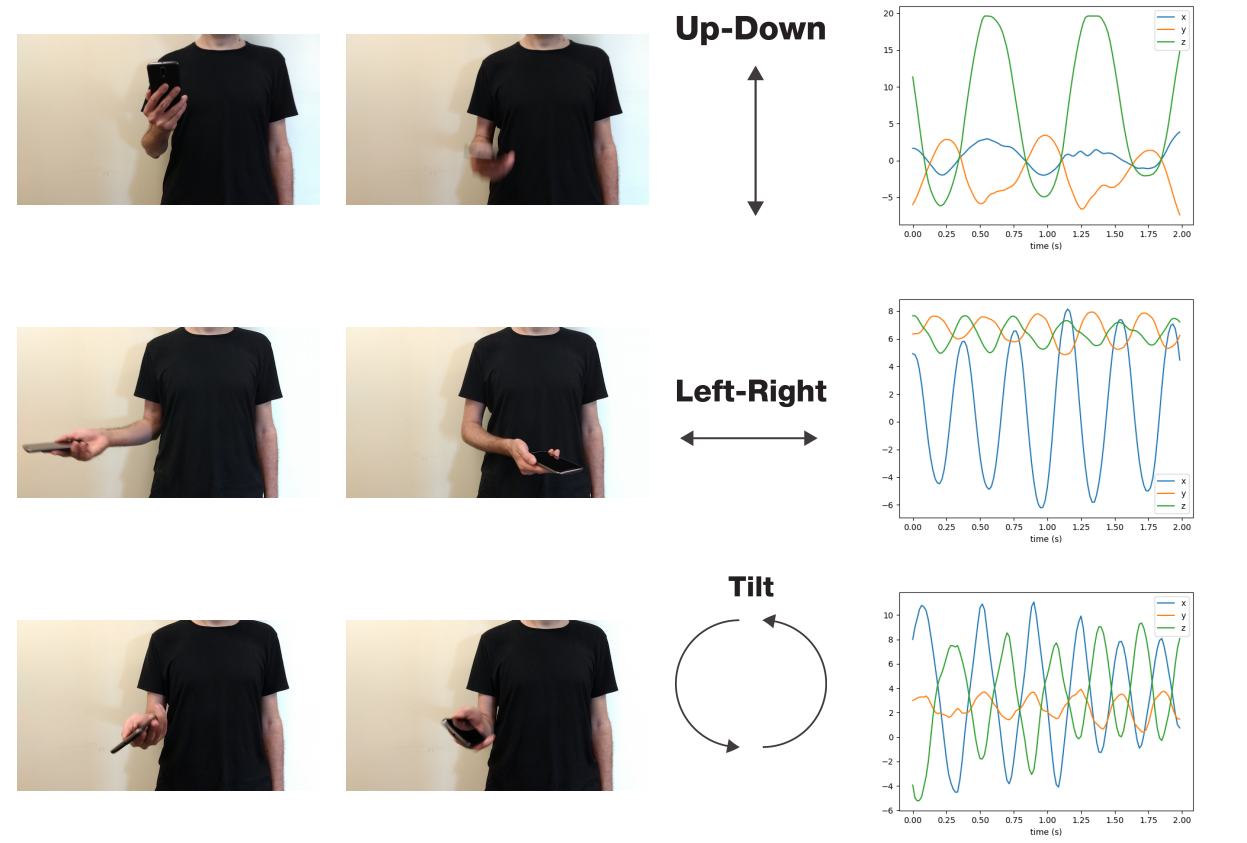

Figure 1: Examples and accelerometer data plots for three gestures.

and input to the network for each new sample. A longer hop size can be used for reducing CPU usage.

Web application

The system is implemented using web technologies and Javascript libraries. This makes it possible to quickly prototype applications that can be executed in most recent smartphones without installing additional software. Accelerometer data is available through the DeviceMotionEvent API.[2] This API currently offers a gravity-corrected version of accelerometer data, which may be supported depending on the hardware. However the most widely supported version is "accelerationIncludingGravity", which is the raw accelerometer data. This is the call currently used in *handwaving*. Accelerometer data is thus captured in real time in a browser window.

A simple web application is used for collecting training data. This step is typically done in advance by a group of users during the preparation of a specific performance or application. Training examples generated with their smartphones is sent to a web server and saved as a JSON file. The data collection application allows creating and deleting gestures, and recording examples of each class. This results in recordings of variable length which are labelled according to the gesture class. The recognition model is trained with this data using Convnetjs.[3] This step is currently implemented as an offline task, although it could be also executed in a browser. With the amount of data used in our experiments (see Section 5), a model can be trained in les than a minute with a current laptop.

The web application also allows managing sounds and mapping gestures to sounds. Sound synthesis is currently done using flockingjs,[4] a library that mimics the syntax for creating synth definitions in SuperCollider, as well as a subset of its unit generators. This allows easily coding a wide variety of sounds with a low entry fee for composers and musicians familiar with SuperCollider. Synth definitions are

[2]https://www.w3.org/TR/orientation-event/

[3]http://cs.stanford.edu/people/karpathy/convnetjs/

[4]http://flockingjs.org/

written in the web interface in JSON format, and assigned to each gesture along with a *mapping expression*. The mapping expression allows the synthesizer assigned to each gesture to make use of the raw accelerometer values as parameters for increasing the expressivity and variability of the sound. A mapping expression typically consists of simple arithmetic operations defined in Javascript. For example a mapping can associate the parameter $osc1.freq$ of the synth definition with the expression $440 * (1 + 0.01 * x)$, where x is the value for the accelerometer x-axis. The code for mapping expressions is loaded by the performance interface and evaluated in order to update the synth in real time.

Performance scripts

Given a model trained for a set of gestures, and the corresponding synths and mappings, a basic example web page is provided that allows using a phone as a musical instrument. This basic setup can already be used for mobile performance or audience participation by hosting the code in a web server. The audience can then just visit a web page that will download the neural network model along with the synth definitions and mappings to their mobile browser, detect gestures, and produce the corresponding sounds. More complex compositions can be coded as a sequence of web pages representing different parts of a composition, or in more involved web applications. In our initial performances we have explored different mappings, additional client features such as haptic feedback, as well as server functionality for compositional decisions depending on group behavior (see Section 6).

5 DATASET AND EVALUATION

In order to test our system, we collected a dataset of gestures using the web application. The data was generated by 5 different users including the authors. The dataset and a pre-trained model are available in the software repository, and can be readily used for music creation with the basic set of gestures. In this section, we describe an evaluation experiment for the recognition system using this dataset, and motivate our parameter choices.

The dataset consists of 10 recordings of each of 7 gesture classes: left / right (lr), up / down (ud), tilt ($tilt$), circles ($circ$), forward / backwards (fb), concave ($conc$) and convex ($conv$) (in our experiments, silence was better detected simply using a threshold on acceleration). The recordings were preprocessed to remove initial silence (accelerometer values between tapping the record button and starting repetitive gestures). While the specification does not include a value for the sampling rate of accelerometer data, we tested several devices and found a consistent value of 60 Hz. The recordings were cut down to a common minimum of 800 points (13 seconds), so the same amount of data was available for each class. The time series was segmented using a fixed length moving window and half-window overlap. The resulting vectors were either fed directly to the neural network or analyzed by an FFT module to extract the magnitude spectrum (i.e., an STFT with rectangular window) The network was set to the same number of units as the input, so FFT features used half the number of units both in the visible and hidden layer. A final softmax layer was used to predict the gesture class. We compared different window sizes and the use of FFT analysis in a cross-validation setting. For each fold, the model was trained with 9 recordings and evaluated on the remaining one, and computed accuracy as the fraction of correctly classified windows across the test recordings of all classes. Figure 2 shows the result for different window sizes (8, 16, 32, 64 and 128 samples). The best accuracy is achieved when using FFT features, which are able to make better use of longer windows. For very short windows (e.g. in the order of half a second), raw features could be used. While in real-time usage we implemented the system to perform recognition with a hop size of one sample, it may be desirable to modify the window or hop size in order to reduce the computational cost. We found that continuous recognition performed well with most current smartphones. Older smartphones (e.g. an iPhone 4) may have both compatibility and performance issues. We found that as a rule of thumb a phone that is able to run a current version of the Chrome browser (version 57 at the time of this writing) will be able to run our system. A capability check page is provided as part of the web application framework.

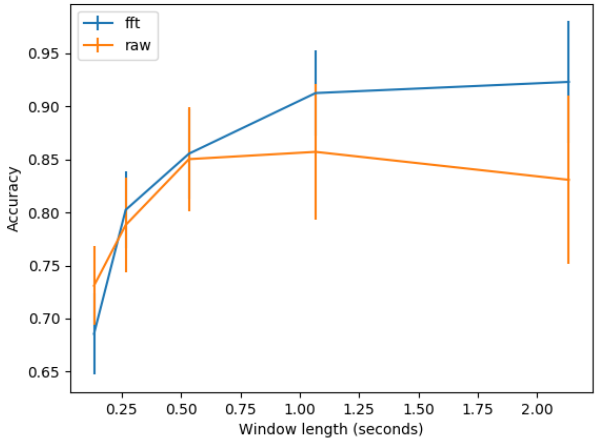

Figure 2: Mean classification accuracy using either raw or FFT features, as a function of the window size. Error bars indicate standard deviation.

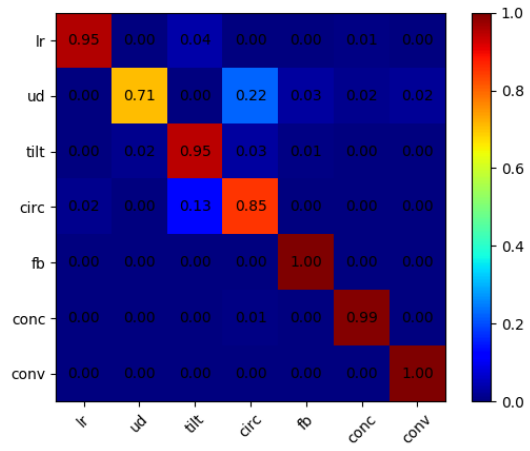

Figure 3: Confusion matrix using 128ms window and FFT features.

Figure 3 shows a confusion matrix for the different classes with the best set of parameters. Most confusions happen between "up-and-down" (which tends to involve more unintended movement along horizontal axes) and "circles" (which includes movement across the vertical axis). While we are experimenting with more complex gestures (such as alphabet letters), it is obvious that smaller number of classes will result in better recognition. At the same time, gestures which make use of different accelerometer axes will be easiest to tell apart. However, the proposed framework does not intend to explicitly model these gestures, and can in principle be trained to recognize arbitrary shapes.

6 INITIAL PERFORMANCES

The idea of audience participation in music performance affords the possibility of a shift with respect to the traditional views of authorship, and the historical roles assigned to composer, performer and spectator of music. In the extreme case of audience participation, technology can be used to create musical experiences that are focused on the audience itself, instead of on a performer. In this sense, the development of audience participation enables a performance genre that is significantly different from both traditional setting, where the role of the audience is generally limited to signs of appreciation towards the performer/s, and the acousmatic setting, where no performer is in sight, but the audience remains passive.

We used *handwaving* to try the idea of a purely audience-led music performance named "Do the Buzzer Shake". The piece was inspired by online cultural transmission through memes, while exploiting the role of imitation typically associated with gestures and gesture-based communication, music and dance. The sounds we used were based on square-wave oscillators in order to maximize the loudness of sounds produced by mobile phones.

The piece was rehearsed several times in classroom and lab environments with groups of between 5 and 15 students, and once with a group of 100 students. It was later performed in public during the second International Conference of Live Interfaces (ICLI2016), and in the first annual Concert of Women in Music Tech held at Georgia Tech in Atlanta.

During the development of the piece, a structure of three parts was devised. In the first part, participants explored the use of the accelerometer and synchronization with others by trying to achieve consonance (identical phone orientations) or dissonance (different orientations). In the second part, participants explored the different gestures and their musical mappings and learnt them from each other. In the final part, synchronization was "mandatory": the server would count the number of participants performing each gesture, and participants performing minority gestures were "punished" with a quick vibration and a short period of silence. The duration of the silence increased progressively in order to induce a sparse ending unless a total synchronization was achieved.

During the rehearsals and public performances it became clear that the audience was engaged and enjoyed the experience, and that they could easily learn the gestures and play the associated sounds. While participants were always instructed to be quiet, the absence of a central figure and the playfulness of the situation made it very unlikely that they would remain silent. Although the music was made out of drones with varying degrees of frequency stability, creating both harmonic and chaotic patterns, the atmosphere of participation had some parallels with group behavior in electronic dance music clubs, where the DJ is not necessarily the center of attention. In this sense, our research connects with previous investigations on music control by large groups [5]

We are working on new music works using the system, with the aim to improve the immersion through better sound amplification and visuals, while preserving the level of engagement and the focus on collective experience.

7 CONCLUSIONS

With so many people carrying pocket computers with multiple sensors and sound capabilities, audience participation is likely to become a more important aspect of music performance. In this paper, we have proposed a framework for participatory mobile music based on mapping arbitrary accelerometer gestures to sound synthesizers on mobile phones.

Figure 4: A moment in the ICLI 2016 performance (photo: ICLI2016 organization committee).

We have provided an initial dataset and shown that the system is able to learn new gestures with a few examples. Finally, we have described initial experiences using this system in audience-driven participatory performances. As future work, we hope to simplify the process of training and keep collecting data for new gestures, eventually contributing to the definition of collective vocabularies for participatory mobile music. The code and dataset can be obtained from https://github.com/g-roma/handwaving.

8 ACKNOWLEDGEMENTS

We would like to thank all the students who have helped testing the system and providing training data at University of Surrey and Georgia Institute of Technology.

REFERENCES

[1] Jamie Bullock and Ali Momeni. 2015. ml. lib: Robust, Cross-platform, Open-source Machine Learning for Max and Pure Data. In *Proceedings of the 15th international Conference on New Interfaces for Musical Expression (NIME2015)*. Baton Rouge, LA, USA, 265–270.

[2] Andrey Bundin. 2016. Concert for Smartphones. In *Proceedings of the 2nd Web Audio Conference (WAC-2016)*. Georgia Institute of Technology, Atlanta, GA, USA.

[3] Baptiste Caramiaux and Atau Tanaka. 2013. Machine Learning of Musical Gestures.. In *Proceedings of the 13th International Conference on New Interfaces for Musical Expression (NIME2013)*. Seoul, South Korea, 513–518.

[4] Georg Essl and Michael Rohs. 2007. The Design Space of Sensing-Based Interaction for Mobile Music Performance. In *Proceedings of the 3rd International Workshop on Pervasive Mobile Interaction*. Toronto, Ontario, Canada.

[5] Mark Feldmeier and Joseph A Paradiso. 2007. An interactive music environment for large groups with giveaway wireless motion sensors. *Computer Music Journal* 31, 1 (2007), 50–67.

[6] Rebecca Fiebrink, Dan Trueman, and Perry R Cook. 2009. A Meta-Instrument for Interactive, On-the-Fly Machine Learning.. In *Proceedings of the 9th International Conference on New Interfaces for Musical Expression (NIME2009)*. Pittsburg, PA, USA, 280–285.

[7] Jason Freeman, Shaoduo Xie, Takahiko Tsuchiya, Weibin Shen, Yan-Ling Chen, and Nathan Weitzner. 2015. Using massMobile, a flexible, scalable, rapid prototyping audience participation framework, in large-scale live musical performances. *Digital Creativity* 26, 3-4 (2015), 228–244.

[8] Abram Hindle. 2013. Swarmed: Captive portals, mobile devices, and audience participation in multi-user music performance. In *Proceedings of the 13th International Conference on New Interfaces for Musical Expression (NIME2013)*. Seoul, South Korea, 174–179.

[9] Ben Houge. 2016. Ornithological Blogpoem. In *Proceedings of the 2nd Web Audio Conference (WAC-2016)*. Georgia Institute of Technology, Atlanta, GA, USA.

[10] Douglas Keislar. 2009. A Historical View of Computer Music Technology. In *The Oxford Handbook of Computer Music*. Oxford University Press, Oxford, UK, 11–43.

[11] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Sergio Di Marca. 2006. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing* 10, 5 (2006), 285–299.

[12] Lyndon Kennedy and Mor Naaman. 2009. Less Talk, More Rock: Automated Organization of Community-contributed Collections of Concert Videos. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, New York, NY, USA, 311–320.

[13] Sang Won Lee, Antonio Deusany de Carvalho Jr, and Georg Essl. 2016. Crowd in c [loud]: Audience participation music with online dating metaphor using cloud service. In *Proceedings of the 2nd Web Audio Conference (WAC-2016)*. Georgia Institute of Technology, Atlanta, GA, USA.

[14] Sang Won Lee and Jason Freeman. 2013. echobo: A mobile music instrument designed for audience to play. *Proceedings of the 13th International Conference on New Interfaces for Musical Expression (NIME2013)* 1001 (2013), 42121–48109.

[15] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* 5, 6 (2009), 657–675.

[16] Nihar Madhavan and Jeff Snyder. 2016. Constellation: A Musical Exploration of Phone-Based Audience Interaction Roles. In *Proceedings of the 2nd Web AUDio Conference (WAC-2016)*. Georgia Institute of Technology, Atlanta, GA, USA.

[17] Jieun Oh, Jorge Herrera, Nicholas J Bryan, Luke Dahl, and Ge Wang. 2010. Evolving The Mobile Phone Orchestra. *Proceedings of the 10th International Conference on New Interfaces for Musical Expression (NIME2010)* (2010), 82–87.

[18] Charissa Ann Ronao and Sung-Bae Cho. 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications* 59 (2016), 235–244.

[19] Atau Tanaka. 2004. Mobile music making. *Proceedings of the 4th International Conference on New interfaces for Musical Expression (NIME2004)* (2004), 154–156.

[20] William Walker and Brian Belet. 2016. Musique Concrète Choir: An Interactive Performance Environment for Any Number of People. In *Proceedings of the 2nd Web Audio Conference (WAC-2016)*. Georgia Institute of Technology, Atlanta, GA, USA.

[21] Ge Wang, Georg Essl, and Henri Penttinen. 2008. Do mobile phones dream of electric orchestras. *Proceedings of the International Computer Music Conference (ICMC 2008)* 16, 10 (2008), 1252–61.

[22] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. 2009. Gesture recognition with a 3-d accelerometer. In *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing (UIC '09)*. Springer, Brisbane, Australia, 25–38.

[23] Leshao Zhang, Yongmeng Wu, and Mathieu Barthet. 2016. A Web Application for Audience Participation in Live Music Performance: The Open Symphony Use Case. *Proceedings of the 16th International Conference on New Interfaces for Musical Expression (NIME2016)* 16 (2016), 170–175.