Pairwise: An Interface for Dance Performance and Social Engagement

Peter Benson
Northeastern University
360 Huntington Avenue
Boston, Massachusetts, 02115
benson.p@northeastern.edu

Rebecca Kleinberger
Northeastern University
360 Huntington Avenue
Boston, Massachucsetts, 02115
r.kleinberger@northeastern.edu

ABSTRACT

Pairwise is a musical interface that maps interpersonal synchrony onto audio features during musical playback. Simple, intuitive mapping was a key consideration for this project. Previous approaches to this problem have either given the participant too much control over the output, or too little, resulting in an imbalance of participant agency. This paper will cover the background of interactive musical devices created to promote synchrony and prosocial behavior. It will also describe the design process and technical implementation of *Pairwise*, including mappings, protocols, and recommendations for effective performance. Finally, this paper will evaluate *Pairwise* in technical and performance contexts, and propose ways to improve future implementations of this technology.

Author Keywords

Social, Dance, Performance, Wireless

CCS Concepts

•**Applied computing** → **Sound and music computing;** *Performing arts;*

1. INTRODUCTION

In this paper, I will propose a new interface for social dance, *Pairwise.* It is well-known that music plays a large role in human society in social bonding [16]. Furthermore, empirical results have shown the importance of dance and synchrony for social bonding and prosocial sentiment. This interface measures synchronous movement during a dance performance, and uses it to alter musical playback.

Previous approaches have incorporated some aspects of this idea, measuring pair synchrony [7], in group settings [9], and for specific dance styles [6]. *Pairwise* is designed to incorporate successful elements from these approaches, while streamlining the performance aspects. This interface relies on simple, intuitive mapping to modify musical playback. Using microcontrollers, *Pairwise* will be able to measure and and incorporate interpersonal synchrony into musical playback.

NIME'23, 25 April, 2023, Boston, Massachusetts

2. BACKGROUND

One prominent theory for the origin of music in human society is as a means for social bonding [16]. Dance plays a major role in this theory, specifically, the process of synchrony, where two participants move together simultaneously to music. Synchrony is shown to increase increase pain tolerance [20]. One study [19] showed participants two animated figures, one moving in sync with music, and the other moving out of sync with music. The participants felt more helpful and more socially bonded to the figure moving in sync with the music. Another study [17] investigated characteristics of EDM that facilitate interpersonal synchrony and social bonding during dance, such as temporal characteristics and "sound intensity" characteristics. Together, these studies show clear evidence that music and dancing, promote high levels of interpersonal synchrony. The function of *Pairwise* is to reverse this relationship, by determining musical outcomes from interpersonal synchrony. In doing so, *Pairwise* will consist of wearable technology designed to measure synchrony of performance between participants in dance.

Previous approaches in New Interfaces for Musical Expression (NIME) have demonstrated applications of social interfaces. One project [18] mapped playground implements such as swings, merry-go-rounds, and a see-saw to musical features. Another project [5] mapped the movements of a crowd, the wind, and uses it to alter musical playback. Futhermore, software-based socialization has been studied in the form of *ChuckPad* [15], an interface for sharing code and musical fragments with other users. Finally, *resonate* [13] presents the user with a simple, intuitive interface for interaction. This project is composed of many floor-to-ceiling strings, which are plucked by the users to generate a variety of unique sounds. This is a fantastic approach to social melodic interfaces, but fails to capture the nuances of social dance and body movement.

Other technologies have implemented glove-based wearables to control musical elements. One implementation uses Wiimotes [21] as accelerometers and transmitters. Another project [11] created a glove-based subtractive synthesizer, generating sounds similar to a human voice.

Several approaches have also demonstrated the effectiveness of wearable technology and social aspects of music performance. One of the earliest examples of this is the *Synchrotron* [7], which measured participant's synchronization to a predefined beat. However, Synchrotron did not directly control the musical output – it served as a measure of synchrony and displayed a simple "score" for the participants and audience. Another early example of a social, wearable interface is Talbot3 [9]. This system gives glow sticks with radio transmitters in them to a large group of participants, giving the crowd the ability to control musical factors such as tempo, timbre, and style of music.

An ambitious study [6] combines social interaction, wearable technology, and generative music. Using wearable sensors, the specific gestures and dance moves in a tango performance can cause changes to the accompaniment and melody of a song generated during performance. This project has similar goals to *Pairwise*, but is confined to one style of dance, and uses a different mapping, where the dancers have significantly more control over the melodic aspects of the musical playback. Furthermore, it is far more computationally intensive, including melodic and harmonic elements in the control surface.

A major factor in the development of *Pairwise* is simple, intuitive mapping. The goal is that anybody can perform with this device with no explanation necessary. To do so, the mapping must be as simple as possible. One framework for mapping [8] proposes that System Response Time is key to encouraging a "flow state" in the user. This means that a highly-responsive, intuitive system depends highly on fast processing, which means a highly streamlined, simple back end.

While previous approaches have contributed a great deal to the study and implementation of social, wearable interfaces, the ultimate sticking point is mapping. Many of the designs presented above utilize many-to-many, or one-to-many mappings. [6] [11] Furthermore, several social interfaces with simpler or more intuitive limit individual control over the resulting music. [9] [5] Pairwise represents an advancement in mapping social dynamics to music by streamlining the mapping process. This allows for smoother, more intuitive, and more expressive performance.

3. TECHNICAL DESIGN

Pairwise consists of two hand-worn bracelets, each one equipped with a Bluetooth transmitter and and accelerometer. These measure the movements of the participants, which are transmitted wirelessly to the host computer, where sound is being emitted. The movements of the participants are then used to modify the sound coming from the host computer. This provides a simple, intuitive, yet powerful way to both measure interpersonal synchrony and use it in a musically expressive way.

For the devices used in *Pairwise*, I used the Adafruit Bluefruit [1] microcontroller. This comes with a built-in accelerometer, along with built-in Bluetooth capabilities. Furthermore, it serves as a capable interface for CircuitPy, a Python framework designed to work closely with hardware [3]. Since I have an extensive knowledge of Python, this was a natural fit for this application.

A major challenge was setting up the Bluetooth sending capabilities of the Bluefruit, along with an accompanying receiver on my laptop. I found a helpful tutorial [2] detailing Bluetooth data transmission using CircuitPy. This relies on a script running on the Bluefruit, along with a receiver script running on my laptop. Using this framework, the Bluefruit sent messages over a Serial connection to my laptop, consisting of its x, y, and z readings. To connect two boards, I had to determine the Bluetooth address of each board. I scanned nearby Bluetooth devices, filtering to only display Bluefruit devices, which gave me the specific ID's of the boards. Now, when connecting to the boards, I was able to reliably setup the connection for each board. More importantly, I was able to keep the board numbers the same, so I didn't have to worry about the boards being swapped accidentally during testing.

Next, I connected the receiver script to an Open Sound Control port on my computer. When my laptop received Bluetooth transmissions from the Bluefruit, it then transmitted these messages to MaxMSP. Each device had its own incoming OSC port, which allowed me to separate the devices and calculate synchrony.

The synchrony calculation is based on the idea that a fundamental aspect of dance and its role in social bonding is synchrony between the performers, as discussed in 2. This means that synchrony can be calculated by measuring the degree to which the devices are moving in parallel, or are moving opposite to one another. A vector dot product can be useful for such an application. This can be calculated as such:

Let a be the accelerometer vector of the first device, with a_1, a_2, a_3 corresponding to x, y, z of the accelerometer, respectively. Let b be the accelerometer vector of the second device, with the same mapping from vector elements to accelerometer data. The dot product can be calculated as such:

$$a * b = (a_1 * b_1) + (a_2 * b_2) + (a_3 * b_2) \tag{1}$$

This can be rewritten as:

$$a * b = ||a|| * ||b|| * \cos(\theta) \tag{2}$$

Where $||a||$ denotes the magnitude of a, $||b||$ denotes the magnitude of b, and θ denotes the angle between them. When $\theta = 0$, $\cos(\theta) = 1$, the two vectors are pointing in the same direction, or in opposite directions. Thus, the dot product will not be 0. However, if $\theta = \frac{\pi}{2}$, $\cos(\theta) = 0$, the two vectors must be at a 90 degree angle from one another. This means that the dot product of the two vectors will be 0. The synchrony of two performers can thus be calculated by the dot product of the accelerometer vectors of both devices. This means that perpendicular motion (low synchrony) will be near 0, while parallel motion (or exactly opposite motion) will be far from 0. For parallel motion, the dot product will be positive. However, for exactly opposite motion, the dot product will be negative. Thus, it is important to apply an absolute value to the calculation, such that high values will map to high synchrony and low values will map to low synchrony.

Using the port from Open Sound Control, MaxMSP implemented the synchrony calculation. I was careful to map

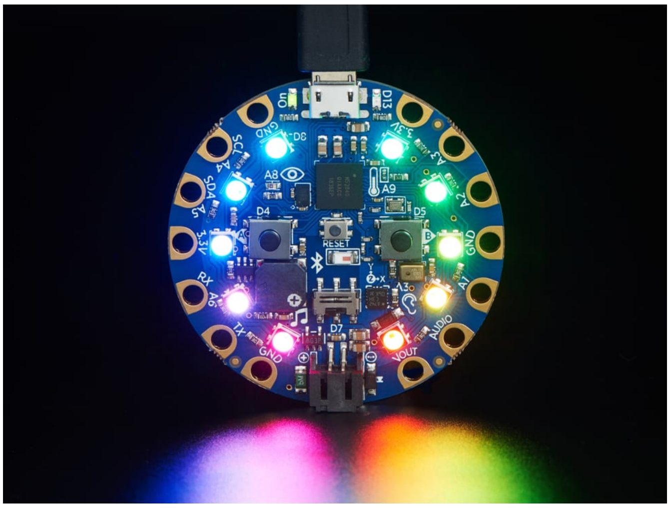

Figure 1: The Adafruit Bluefruit used in *Pairwise*

the synchrony onto a specific range, in this case, 0 to 1. An important part of the synchrony calculation was the calibration step. When the calibration button was pressed, the current synchrony value was mapped to the maximum input range of synchrony. As a standard calibration method, I laid both devices flat on the table and pressed the calibration button.

Then, I played a known piece of music and modulated it using certain audio effects. When mapping the synchrony input to musical effects, I chose low-pass-filtering and volume modulation, as well as a combination of both. On the low-pass filter setting, higher synchrony led to an increase in the cutoff frequency, resulting in more high frequencies in the output sound. On the volume setting, higher synchrony led to an increase in volume. Finally, with both engaged, higher synchrony led to both an increase in volume and an increase in the cutoff frequency of the filter.

Furthermore, I defined three input modes. The first one, "position", was the raw synchrony value without the correction of absolute value. This essentially measured the orientation of each board. The second one, "velocity", was the synchrony value with the correction of the absolute value. Finally, "single-board position" was the position value, but calculated from the dot product of one board's accelerometer vector with a known "down vector" representing a stationary board. This allowed me to perform solo, or with only one hand.

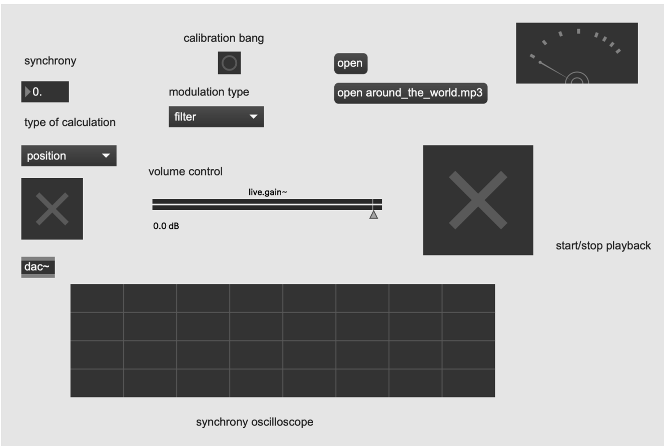

Figure 2: The prototype Max Patch, showing calculation type, mapping type, and other control surfaces

4. PERFORMANCE

To show the performance of *Pairwise*, I performed an improvised individual dance using the "single board position" mode combined with the "low-pass filter" modulation setting. I used Daft Punk's song "Around the World" as the music to be modulated. I chose this song because I am already familiar with it and it is already a highly danceable song, lending itself well to a performance such as this one. Furthermore, Daft Punk composes music in the genre of "Filter House", which relies heavily on low-pass filtering [4]. This means that the low-pass filtering modulation would sound the most natural with the song chosen.

5. EVALUATION

The audience responded well to the performance. I noticed some of them nodding their heads along to the music, which was a sign that they were also experiencing the effect of groove and synchrony. This has implications for future performances of *Pairwise* as an effective medium to associate the visual aspects of dance with the auditory aspects of the music being played.

Unfortunately, a full-scale evaluation of this project was not possible given the time frame and budget requirements. For future evalulation, we would present participants with different variations of this device (pair and solo), with different mappings (filter, volume, orientation, velocity). This study would perform a within-group analysis, where each participant would be asked to use each variation of mapping. The participants would be asked to perform with the device for a short amount of time, roughly 2 minutes. After this, they would rate how socially connected they felt to the other participants of the study, if using the pair variation. Furthermore, they would rate the interface itself – how smooth the controls were, the technical limitations, the performance viability, and how intuitive the mappings were. From this study, we would be able to derive areas in which to improve *Pairwise*, which could be implemented in a future revision of the project.

6. DISCUSSION

One major obstacle to performance was the latency. While the communications between the Bluefruit and my Laptop were reasonably fast, there was a noticeable delay between my movements and response to them. Due to this latency, I found that gradual motions worked the best and sounded most natural. Fast, jerky movements came in a few milliseconds later, leading the sound to be out of sync with my movements. One possible solution to this issue is to incorporate the gradual motions as a deliberate design decision. This would indicate a more fluid, slow dance performance. This would be especially useful

Another characteristic of this particular device is the measuring mode of the accelerometer. I expected the accelerometer to measure the relative velocity of the performers' movements. After testing, I found that it was actually measuring the orientation of the device, since "down" registered as an acceleration of $9.8m/s^2$. When the device was turned upside down, it now registered an acceleration of $-9.8m/s^2$. However, fast movements would also read as acceleration, but only lasted for a finite period of time. This also made quick movements unreliable relative to gradual motion.

During testing, it was also clear that the existing synchrony calculation failed to accurately model ground truth synchrony, and instead measured the orientation of both devices. For this iteration of *Pairwise*, this made the need for two devices redundant. The best performance results came with one device and one performer, which could be mapped to a low-pass filter or the output volume of the music, as described in 3.

One possibility for a future implementation of this technology is as a free-form DJ control, where dance is factored into the performative aspect of DJing. For this implementation, each hand could control a different aspect of the musical playback. For example, one hand could control a low-pass filter, while the other could control a high pass filter. Once again, it is important to retain the simple, intuitive mappings for this performance type. I expect that such an implementation would only have two measures of orientation.

Another possible implementation is to increase the number of possible simultaneous participants. While measuring group synchrony may be harder than pair synchrony, it would provide valuable insights into the social dynamics of group dance. One possibility for this implementation is to have a "leader" of the group dance, with synchrony measured to their movements. This is an interesting solution to the problem of measuring a "synchrony baseline" for such an implementation. However, it fails to capture the often decentralized nature of group dance. Anecdotally, during improvised group dance, there is often no set "leader". This would cause the device to fail at measuring the true synchrony of the group.

7. CONCLUSIONS

In this paper, I proposed a new interface for social dance, *Pairwise*. This interface relies on simple,intuitive mapping to modify musical playback. Further work includes refining the synchrony calculation so that it functions properly for two participants. Additionally, branching the project into an individual dance performance device and a social dance performance device could be beneficial, perhaps combining the two for a more complex performance. Finally, more research into intuitive mapping is required to finish *Pairwise* and ensure that its original goals are met.

8. REFERENCES

[1] Adafruit Circuit Playground Bluefruit.

[2] Bluetooth Sensor Capture - Adafruit Circuit Playground Bluefruit — Kinetic Fabrics.

[3] CircuitPython.

[4] J. Baines. A Bullshitter's Guide to Filter House, Aug. 2015.

[5] B. Bortz, A. Ishida, I. I. Bukvic, and R. B. Knapp. Lantern Field: Exploring Participatory Design of a Communal, Spatially Responsive Installation. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 73–78. Zenodo, June 2013.

[6] C. Brown. Machine Tango: An Interactive Tango Dance Performance. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 565–569, Tempe Arizona USA, Mar. 2019. ACM.

[7] M. Demey, M. Leman, F. Bossuyt, and J. Vanfleteren. The Musical Synchrotron : Using Wireless Motion Sensors To Study How Social Interaction Affects Synchronization With Musical Tempo. June 2008.

[8] R. A. Doherty and P. Sorenson. Keeping Users in the Flow: Mapping System Responsiveness with User Experience. *Procedia Manufacturing*, 3:4384–4391, 2015.

[9] M. Feldmeier and J. A. Paradiso. An Interactive Music Environment for Large Groups with Giveaway Wireless Motion Sensors. *Computer Music Journal*, 31(1):50–67, Mar. 2007.

[10] S. Fels, A. Gadd, and A. Mulder. Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, 7(2):109–126, Aug. 2002.

[11] Y. Han, J. Na, and K. Lee. Futuregrab: A Wearable Subtractive Synthesizer Using Hand Gesture. June 2012.

[12] Y. Han, J. Na, and K. Lee. Futuregrab: A Wearable Subtractive Synthesizer Using Hand Gesture. June 2012.

[13] B. Knichel, H. Reckter, and P. Kiefer. Resonate – A Social Musical Installation Which Integrates Tangible Multiuser Interaction. June 2015.

[14] M. Myllykoski, K. Tuuri, E. Viirret, J. Louhivuori, A. Peltomaa, and J. Kekäläinen. Prototyping Hand-Based Wearable Music Education Technology. June 2015.

[15] S. Salazar and M. Cerqueira. Chuckpad: Social Coding For Computer Music. June 2017.

[16] P. E. Savage, P. Loui, B. Tarr, A. Schachner, L. Glowacki, S. Mithen, and W. T. Fitch. Music as a coevolved system for social bonding. *Behavioral and Brain Sciences*, 44:e59, 2021.

[17] R. T. Solberg and A. R. Jensenius. Group behaviour and interpersonal synchronization to electronic dance music. *Musicae Scientiae*, 23(1):111–134, Mar. 2019.

[18] M. St. Clair and S. Leitman. Playsoundground : An Interactive Musical Playground. June 2009.

[19] J. Stupacher, G. Wood, and M. Witte. Synchrony and sympathy: Social entrainment with music compared to a metronome. *Psychomusicology: Music, Mind, and Brain*, 27(3):158–166, Sept. 2017.

[20] B. Tarr, J. Launay, E. Cohen, and R. Dunbar. Synchrony and exertion during dance independently raise pain threshold and encourage social bonding. *Biology Letters*, 11(10):20150767, Oct. 2015.

[21] K. Tsoukalas, J. Kubalak, and I. I. Bukvic. L2Orkmote: Reimagining A Low-Cost Wearable Controller For A Live Gesture-Centric Music Performance. June 2018.

[22] K. Tsoukalas, J. Kubalak, and I. I. Bukvic. L2Orkmote: Reimagining A Low-Cost Wearable Controller For A Live Gesture-Centric Music Performance. June 2018.