HANDMONIZER: AN ARTIST-ORIENTED VOCAL IMPROVIZATION TOOL

Pappas Antonios, Lionetti Davide

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano
Piazza Leonardo Da Vinci 32, 20122 Milano, Italy
[antonio1.papa,davide1.lionetti]@mail.polimi.it

ABSTRACT

The rapid evolution in technology has found its way to introducing novelty in today's live music performances. Music Interaction Design is a field dedicated to the enhancement of musical creativity by introducing new interaction methods between the performer and the system. This paper presents the development of the "Handmonizer", an interactive harmonizer for musical performance adapted to the needs of a specific singer. A key component of our work is the combination of hand motion recognition and audio signal processing which is regulated by this type of interaction. We describe the development methodology, but we also focus on our collaboration with a well-known singer to conceptualize and then refine this tool until the development of the final product. At the end of this report, we defined an evaluation strategy, collecting feedback with a questionnaire addressed to the singer. The aim of the latter is to help other engineers that would like to develop cutting-edge technologies working alongside artists.

Index Terms— Music interaction design, artist interaction, hand gesture recognition, vocal harmonizer, live performance

1. INTRODUCTION

The whole way of interacting with machines is changing, since gestures, movements, and direct graphic manipulation are co-existing with physical interfaces like keys, buttons, and knobs. In the musical context, these new interface technologies have given us countless possibilities in the creation of Digital Musical Instruments (DMI) or New Interfaces for Musical Expression (NIME), artifacts that connect inputs (interface controllers) and outputs (sound synthesis modules) according to a mapping strategy [1]. A DMI is composed of a control interface and a sound synthesizer, which work separately [2]. The communication between these two components is made possible using some industry-standard communication protocols which include MIDI and OSC. This is a considerable change compared to traditional instruments, where the musical gesture and the sound generation take place on the same instrument [3]. In the case of a DMI, the design of the Human-Computer Interaction (HCI) is important to give a sense of musical meaning not only to the artist but also to the audience. The main goal in designing such a system is to make HCI and software engineering sides work together seamlessly [4]. Previous examples of similar interactive systems have been developed in the past few years (e.g.: [5], [6]). In our specific case we are supposed to design a DMI performance tool tailored to a specific artist, the well-known Italian jazz singer, Maria Pia De Vito[1]. This scenario allows us to be creative, taking into consideration the artist's needs based on the nature of her live performance. The main goal is to build a custom harmonizer to enhance her solo performance, a DMI with which she can interact creatively and intuitively. Collaborating and discussing her necessities, we develop the Handmonizer, an unusual and special harmonizer controlled in real-time by hand gestures. Just by moving her hand she can explore the different screen areas each associated to a specific setting, creating different harmonic combinations by exploiting the smooth transition between the different voices. At the same time, she can control the amount of different effects (e.g Reverb or Delay) using fixed intuitive hand movements. This interaction strategy allows the artist to be focused on her musical intention while performing, instead of having to think about switching between physical devices that slow the flow of the performance.
In this paper we first describe our cooperation with the artist in order to set the goals, then we explain the technical aspects of our system, going through the tools and methodologies we used to develop the final product. Finally, we present the evaluation given by the artist through a questionnaire. She provided her detailed feedback on every aspect of her experience with the Handmonizer after experimenting with it during two demo sessions. The full code and demo video are available on Git Hub in our repository.[2]

2. ARTIST ORIENTED SYSTEM DESIGN

In this section, we introduce the development process based on the artist's constraints, the discussions and the decisions taken based on her feedback. This system consists of a vocal harmonizer and a hand gesture recognition interface able to map some parameters in real-time based on the movements of the singer's hand.

2.1. The Artist needs

During our first meeting with the artist, we were presented with the challenges she was facing with her previous setup and her vision of her ideal performance scenario. Her previous setup is made up of an Eventide harmonizer guitar pedal and an Echoplex looping machine. She used these two to create her performance's backing tracks along with which she used to improvise[3]. The main issues she faced with her last setup were related to the unnatural and mechanic interaction with the tools. Changing the various patches using switches and knobs is not a quick and comfortable task and reduces her creativity. At the same time, dealing with physical devices like the Eventide, leads to clicking noises caused by knobs that disrupt the musical flow.

[1] Maria Pia De Vito Wikipedia page

[2] GitHub repository containing all the code

[3] Maria Pia De Vito solo performance

The main issue we mutually decided to consider was developing an interface that would be easy for her to use so that she can focus mainly on her improvisation without having to struggle with the interaction. We decided to implement a vocal harmonizer which gives her the opportunity to explore and experiment with the different harmonic configurations in a natural way, just moving her hand in front of a camera.

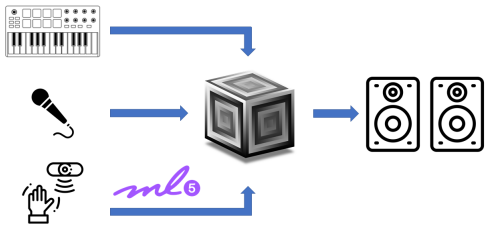

Figure 1: The Handmonizer signal flow

2.2. Interaction Design approach

Due to the separation between gesture input and sound output, the mapping strategies are the essence of digital musical instruments. Different mapping strategies for the same set of inputs and outputs affect how the performer reacts musically and psychologically to the instrument [1]. In general, while designing a DMI we need to follow three main principles [7]:

- **Control:** The way the interaction is mapped to the output, should give full control to the artist who is using the system.
- **Legibility:** also referred to as "transparency", this is related to the importance of having a mapping that is easily understood by both the artist and the audience.
- **Sound:** determining what kind of sounds the mapping is going to produce. This could mean creating new sounds, processing an input sound or even both.

Moreover, when having to design an interface tailored to a specific artist, we as designers should try and put ourselves in the position of the artist in order to develop something meaningful for her. A good starting point to decide on the gesture-sound mapping is to use a bottom-up approach. We start with simple interactions, like opening and closing the hand, that trigger a intuitive response, like turn on or off the harmonizer, and at the same time think of how we can make this interactions meaningful for the audience. Audience engagement is strongly related to the mapping problem. Since the gestures and the produced sound have not necessarily a direct connection, the audience may not feel engaged during the performance [1] [6].

3. HANDMONIZER IMPLEMENTATION

The Handmonizer structure is based on a Node server that allows the communication between the two main parts: the hand gesture recognition and the real time sound harmonization algorithm which communicate through OSC messages. Moreover, a MIDI controller is connected to the harmonizer to switch between different patches during the live performance. In this section we first describe each component separately and finally their communication and parameter mapping.

3.1. Hand gesture recognition

The harmonizer can be controlled through hand gestures captured from a webcam. For the hand motion recognition, we use a pretrained model from ml5.js [4], a JavaScript framework for creative coding built on top of TensorFlow.js that allows the use of GPU-accelerated machine learning algorithms in a web browser. The TensorFlow ecosystem provides this easy-to-use tool to convert pretrained ML models trained in Python or C++ into web targets. Some of these models were specifically designed for creative applications to facilitate the development of real-time music related Web-applications [8]. We use the model called *Handpose*[5] that performs hand-skeleton finger tracking. It takes the video stream frame by frame and returns the coordinates of 21 hand points over the palm of the hand[6] as shown in Figure 2. Based on the coordinates of these 21 points we compute three main parameters

- The hand centroid: displayed as the light-green central dot, it is the arithmetic mean position of the 21 dots.
- The palm length: defined as the distance between the tip of the middle finger and the base of the palm (displayed as the length of the white line).
- The hand orientation: computed as the slope of the white line between $[0, \pi]$.

With a custom mapping of these three parameters we define 5 hand gestures that we can use to change the harmonizer settings in real time. We retrieve the two coordinates of the centroid in the screen x_c, y_c and we map each to a specific parameter. The palm length is used to implement two gestures. We could close the hand to fade out the harmonics or we could move the hand back and forth with respect to the screen to modify the volumes of the harmonized voices. In that way we are able to also map the z-position of the hand in the space. The palm slope controls the amount of effect (Reverb or Delay) that the user wants to add to the voice, as a dry-wet knob; when the white line is perpendicular to the bottom border the voice has no effect, while it reaches its maximum value when the line is parallel to the lower border.

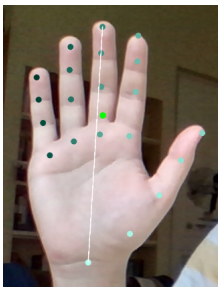

Figure 2: The hand is mapped by Handpose model into 21 points

[4] ml5.js: Friendly machine learning for the Web.https://ml5js.org
[5] https://learn.ml5js.org//reference/handpose
[6] This process is GPU intensive. To achieve the best performance a system with a dedicated GPU is advised.

3.2. Harmonizer implementation

The audio signal processing part of the Handmonizer is developed entirely on SuperCollider. The algorithm of the harmonizer is composed of separate sections, including pitch tracking and shifting, effects definition and communication protocol definitions (OSC, MIDI) for parameter mapping. Here we explain each component separately.

3.2.1. Pitch tracking and shifting

The very first part of the harmonizer aims at recognizing the signal input. Since we need to perform pitch shifting in order to create harmonic voices, the first thing we need to do is track the fundamental frequency of the input audio signal. For this purpose we use an external class called *Tartini* instead of the standard Pitch class from SuperCollider since it performs the pitch tracking more precisely and in a shorter amount of time [9].
To avoid harmonies that sound very unnatural, the pitch shifting needs to be performed using the *Pitch-Synchronous Overlap and Add* (PSOLA). One of the main advantages of this method is the preservation of the formant positions (spectral envelope) which allows us to keep the original timber [10]. Here we use a SuperCollider class called *PitchShiftPA* which is based on PSOLA. We pass the fundamental frequency value tracked by *Tartini* as well as the pitch shift ratio which we fix depending on the harmony we want to achieve. The same procedure is used to generate three harmonic voices, both higher and lower than the input signal (six voices in total). The harmonic voices are mixed together in a separate channel so they can be processed without affecting the lead voice.

3.2.2. Patches and effects

As mentioned previously, one of the main features needed by the artist is the possibility to switch interactively between different settings. We defined 4 patches with fixed intervals following the specifications from the artist: 3rd - 5th; 4th - 5th - 7th; 4th - augmented 4th - 7th; octaver.
These patches include a delay effect that can also be controlled by the user by rotating the hand. The harmonizer also includes two patches that allow the artist to control only the effect amount by rotating the hand; one for the reverb and one for the delay without any harmony. These two patches can be used for single voice improvisations over looped backing tracks.

3.2.3. Cross-fading

A special feature of the Handmonizer is the smooth transition between the voices, creating different patterns during the performance without have any sudden jumps in volume level. Here we manipulate the different amplitudes of each voice group before mixing them. A similar procedure is used for the overall volume fader of the harmonies for which we use a dB scale mapping to approximated the behaviour of the human auditory system. We use *XFade2* as a cross-fade knob to control the dry/wet ration for the reverb and delay effects.

3.2.4. In-scale harmonizer

Finally, we implemented a second version in a separate Supercollider script. Instead of fixed intervals, the Handmonizer can be used as a classic harmonizer following a specific scale, where the user can set the key and scale type (major, minor, etc). By hard-coding the first MIDI note for each key, we use an external class called *MiscFuncs* to retrieve the array of MIDI notes for the selected scale. Then to retrieve in real time the precise MIDI note sang by the singer, we use another external class called MyKFiddle. Finally, the algorithm checks if the input note is part of the scale. If this is the case, it computes the pitch ratio and feeds it to the pitch shifter. The two classes mentioned above, are developed by *Matthew Yee King*[7] and slightly modified by us to be better adapted to our purpose.

3.3. Communication protocols and architecture

To switch between patches, we use a MIDI controller where we assign each pad to a patch by changing the necessary parameters. In addition to the patches mentioned above, we use one pad as an ON/OFF toggle button and another pad as a bypass for the harmonic voices.
The user interface is hosted as a web page/application in an Express server, the connection is set up through the framework Socket.io. All the control parameters mentioned above are computed in the client and then sent to the server. From the server, the parameters are written in OSC messages and forwarded to SuperCollider.
The hand motion recognition features are sent to SuperCollider as OSC messages in real time and are used to control different parameters that define the harmonizer's performance. The palm centroid is tracked into two point : (x_c, y_c). We use the x_c to add more voices as we move from left to right. We can imagine the screen divided into three columns where on the first we only have one additional voice and every time we visit the next column we add one more voice. Similarly, we map the y_c as a switch between low octave and high octave harmonics. We can imagine the screen divided into two rows where the upper row represents the high octave harmonics and the bottom row represents the low octave harmonics. All these changes in number of voices and octaves are mixed together in a smooth way as explained previously, so the artist can explore different sounds without abrupt changes.
Another feature that we use is the palm length represented by the white line in Figure 2. We have mapped this feature to the harmony fader using a dB scale to control the volume of the shifted voices. There are two ways to exploit this feature. The first and most intuitive way is to open and close our hand to turn on or off the device and the second is to move our hand back and forth to change the volume mix of the harmonized voices with the lead one. If the artist wants to emphasise the harmonic voices she can simply move her hand closer to the camera, if she wants to fade out the harmonies she only needs to gently close her hand.
Finally, we use the hand orientation as an imaginary knob that controls the dry/wet level of the reverb or delay effects. When we keep our hand straight we have a fully dry signal (e.g.: no effect). While we rotate our hand either left or right we change the amount of the wet signal and decrease the amount of the dry signal using a cross-fade effect.

4. TESTING AND EVALUATION

For the instrument evaluation and to collect the artist prospective, we design a questionnaire following the typical DMI evaluation methodologies ([11] [12]) focusing our attention on the final

[7] Matthew Yee King Supercollider collection of classes

singer's performance and the ability of the system to fulfil the specific initial constraints. During the two workshop we notice that the artist learns in a fast way how to interact with the tool and in a few trials she is able to come up with very interesting vocal solos. She is not bothered by the interaction, instead she is stimulated to find new ideas and patterns by moving freely her hand. The questionnaire is composed of the following questions. On some of them the artist provided a grade from 0 to 10 and a short comment, while on others she was asked to elaborate more.

1. **Playability:** How much control do you feel that you have over the tool while using it?
   Grade: 8/10
   Comment: I still depend on the presence of an expert to manage it.
2. **Learnability:** How easy was learning to use it?
   Grade: 9/10
   Comment: It takes a little practicing, but the use is pretty intuitive, so very good.
3. **Expressiveness:** How much does it help you to enhance your creativity and express your musical intention?
   Grade: 9/10
   Comment: With some practice the result could be amazing!
4. **Enjoyability:** How enjoyable was your experience while using the Handmonizer?
   Grade: 9/10
   Comment: Very enjoyable. I need practice.
5. **Novelty:** How much novelty does it introduce to your performance?
   Grade: 10/10
   Comment: The use of hand movements instead of just pressing buttons and turning knobs is pretty new and enjoyable for me. I move a lot while improvising!
6. **Effectiveness:** How much does this tool manage to solve the issue with your previous setup?
   Grade: 8/10
   Comment: The problem for me is still to enhance the previous, pretty analog setup! Other than that, it opens up a different field of action, pretty liberating!
7. **Sound quality:** Rate the perceived sound quality obtained through the Handmonizer setup.
   Grade: 9/10
   Comment: The perceived sound quality is very good. I still need to bridge some gaps with the previous setting with practice.
8. How much were we able to understand your needs and fulfill them?
   Grade: 9/10
   Comment: Very good job in this sense! We have been in touch and consulting each other on every step of the way.
9. How much were we able to improve the Handmonizer following your feedback after the first workshop?
   Grade: 10/10
10. Can you describe your experience of being involved in the development of your own custom instrument (expectations, results, impressions)?
    Answer: It has been a very inspiring experience. Still a work in progress for me, having to create instantly, improvising, new soundscapes. I hope to be able to use it in live performances.
11. How confident would you feel using it in a live performance?
    Grade: 7/10
    Comment: I will need the presence of an assistant. I am not a "nerd", but an intuitive musician. It would be reassuring to have somebody able to manage more complex issues.
12. What are the main limitations that you would face if you had to use the instrument on your own?
    Answer: As I said, I am afraid of dealing with the setup and possible issues, having for instance to re-launch the system in case of problems.
13. What should we carry on doing, and what could we do better, in order to work with artists like you to develop cutting edge performance technologies that you would use on concerts?
    Answer: To describe technicalities is a bit out of my strict competence. My wish is that the system becomes pretty "stable", so to become more and more independent from the help of others. I guess this is something that any creative artist would wish for himself/herself. But please, carry on doing "hands on" work with artists!

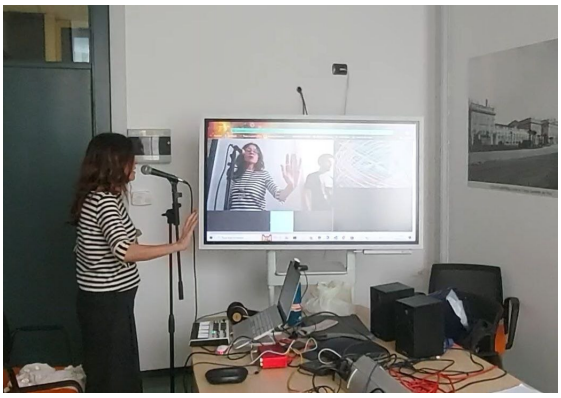

Figure 3: Testing the harmonizer during the workshop

5. CONCLUSIONS AND FUTURE WORK

We have explained in detail how we decided to approach the whole design process keeping into account the feedback from the artist in each step. We have also explained the methodology we used including what available technology we exploited and how we created the architecture of our Handmonizer. We managed to create a fully working harmonizer which interacts effectively with the hand gesture. However, there are a number of improvements to make the experience even smoother. Instead of the ml5.js framework, a leap motion controller could be used instead for a faster and smoother interaction. Another issue is the fact that it is difficult for the artist to run the application on her own. To solve this problem we suppose to have a technician to set-up and run the tool. Finally, a slight latency especially while using the delay prevents the artist from looping her voice in a rhythmical way, which is something that should be solved.

6. ACKNOWLEDGEMENTS

The authors would like to thank everyone involved in this project. First of all, the singer Maria Pia De Vito for her availability, her kindness and her positivity in introducing some novelty in her performance. A special thanks to Augusto Sarti for creating and organizing this brilliant team. Luca Comanducci for his constant feedback concerning the implementation and report writing. Special thanks to the two professors from Goldsmiths University of London. Mark D'Inverno for his insight concerning the evaluation procedure and Mathew Yee King for his availability and valuable SuperCollider support.

7. REFERENCES

[1] R. Medeiros, F. Calegario, G. Cabral, and G. Ramalho, "Challenges in designing new interfaces for musical expression," in *International Conference of Design, User Experience, and Usability*. Springer, 2014, pp. 643–652.

[2] E. R. Miranda and M. M. Wanderley, "New digital musical instruments: Control and interaction beyond the keyboard (computer music and digital audio series)," 2006.

[3] I. Bergstrom, A. Steed, and R. Lotto, "Mutable mapping: gradual re-routing of osc control data as a form of artistic performance," 01 2009, pp. 290–293.

[4] J. Borchers and M. Muhlhauser, "Design patterns for interactive musical systems," *IEEE MultiMedia*, vol. 5, no. 3, pp. 36–46, 1998.

[5] Y. Ikawa and A. Matsuura, "Playful audio-visual interaction with spheroids," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds. Birmingham, UK: Birmingham City University, July 2020, pp. 188–189. [Online]. Available: https://www.nime.org/proceedings/2020/nime2020_paper36.pdf

[6] J. Leonard and A. Giomi, "Towards an interactive model-based sonification of hand gesture for dance performance," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, R. Michon and F. Schroeder, Eds. Birmingham, UK: Birmingham City University, July 2020, pp. 369–374. [Online]. Available: https://www.nime.org/proceedings/2020/nime2020_paper72.pdf

[7] T. West, B. Caramiaux, S. Huot, and M. M. Wanderley, "Making Mappings: Design Criteria for Live Performance," in *NIME 2021*, may 3 2021, https://nime.pubpub.org/pub/f1ueovwv.

[8] A. Correya, P. Alonso-Jiménez, J. Marcos-Fernández, X. Serra, and D. Bogdanov, "Essentia tensorflow models for audio and music processing on the web," in *Web Audio Conference (WAC 2021)*, 2021.

[9] E. Kermit-Canfield, "A comparison of real-time pitch detection algorithms in supercollider," *Journal of the Audio Engineering Society*, october 2014.

[10] N. Schnell, G. Peeters, S. Lemouton, P. Manoury, and X. Rodet, "Synthesizing a choir in real-time using pitch synchronous overlap add (psola)," in *ICMC*, 2000.

[11] J. Barbosa, J. Malloch, M. M. Wanderley, and S. Huot, "What does" evaluation" mean for the nime community?" 2015.

[12] P. J. C. Reimer and M. M. Wanderley, "Embracing Less Common Evaluation Strategies for Studying User Experience in NIME," in *NIME 2021*, apr 29 2021, https://nime.pubpub.org/pub/fidgs435.