KINETIC SONIC PERFORMER

Move to create

Creative Programming and Computing
2022 / 2023 Project

Pisanu Nicolò (ID: 10578782)
Bertazzi Andres (ID: 10488849)
Viviani Marco(ID: 10528650)
Kubler Riccardo (ID: 10560345)

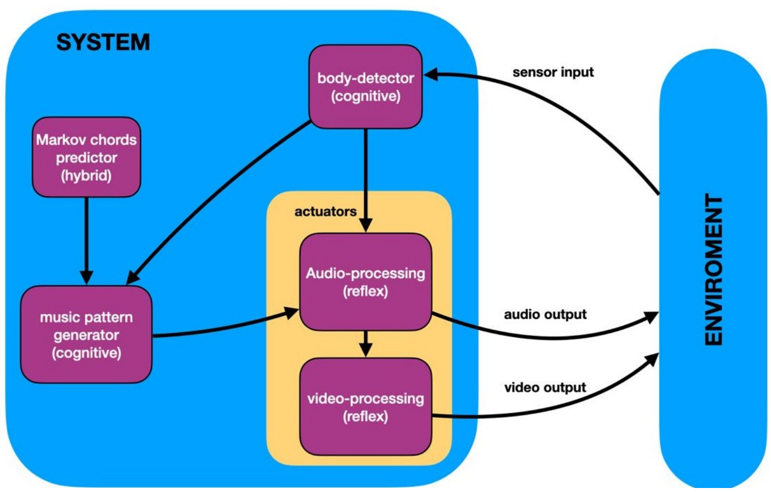
October 02, 2023

Contents

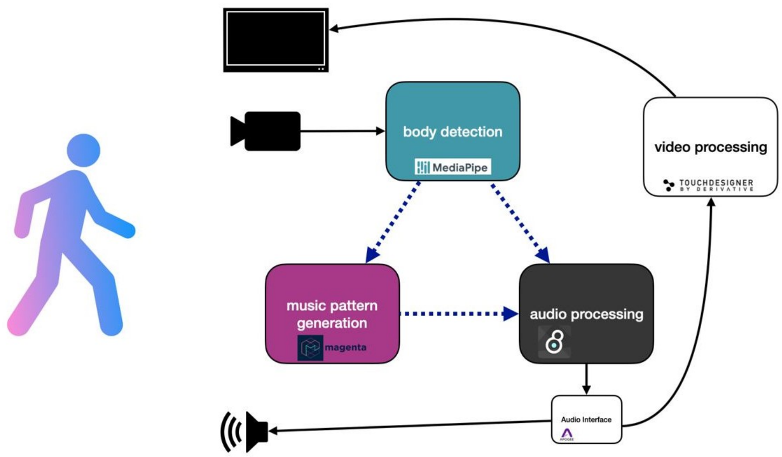
1 Introduction

In this report, we'll explore an innovative project that seamlessly combines music and visuals in a captivating live performance driven by full-body movement. This project empowers anyone to actively participate in music creation through adaptable interfaces, bridging the gap between artists' creative expressions and audience expectations.

The project is built on a full-fledged model designed to foster continuous interaction between the user and the system. It thrives on learning and adaptation to ensure an engaging experience. The fundamental idea is to enable individuals to generate music by moving their bodies within "Hybrid Contexts," where artists craft unique interactive systems.



Utilizing a camera as a sensor to capture the user's movement, the system processes it through MediaPipe.
These parameters in turn control MusicVAE (Variational Autoencoder), generating real-time music sequences played in real-time by Max8.
The audio output also contributes to the creation of a video using TouchDesigner, further enriching the overall sensory experience.



This project caters to a wide range of audiences, including artists seeking new forms of expression, an audience eager for immersive experiences, educators interested in interactive teaching tools, technology enthusiasts exploring advanced applications, and the entertainment industry seeking novel ways to engage audiences. Furthermore, it provides a platform for artistic and creative exploration, creating opportunities for the creative community to push beyond traditional boundaries in art and entertainment.

In the upcoming sections, we'll delve into the technical details and the creative process of this innovative project. It empowers both artists and audiences to explore the exciting realm of interactive live music and synchronized visuals, where creativity and technology converge to redefine the boundaries of live performance. Join us as we navigate this captivating fusion of creativity and technology within these dynamic Hybrid Contexts.

2 Technologies

In this paragraph, we will shortly discuss the technologies used and the challenges, and lessons learned in implementing this project.

2.1 Body Detection

The first module of this project is represented by the "Body Detection" code.
It deals with identifying some user movements that will control the generation of the music in many aspects like the rhythmic structure developed by MusicVAE or some filters and effects generated by Max8.
The music generated by Max8 will then animate an audio-reactive visual implemented in TouchDesigner via OSC.
To do this, we wrote a Python code that makes use of several libraries useful for our purposes.

The first libraries we use are *Mediapipe* and *Mediapipe Holistic*.
They provide Deep Learning tools to recognize the body through a camera. Each part of the body is identified by a landmark which is nothing more than a 2D spatial coordinate on which geometric and non-geometric calculations can be performed.
By exploiting these landmarks we are able to calculate the positions of the affected body parts on the screen and to improve the landmarks' performances we make use of some buffer arrays.
The presence of these buffer arrays allows the user to record a history of the latest values of the calculated quantity and therefore take into account the distance from the screen thanks to the MIN/MAX formula.
At every iteration, a new value is appended to the buffer while the last one is removed and we compute the min and the max of the vector.

By using the formula $x = (X - X_{min})/(X_{max} - X_{min})$ the values are normalized between 0 and 1 and at the same time the distance is taken into account because the values change with respect to distance from the screen.
Usually, the X value is a mean of the last values of the buffer in order to "smooth" the increase or decrease of the quantity we are calculating.
In this way the quality of the calculated quantity increases significantly: body parameters are adaptively normalized in order to consider also the distance of the user from the screen.
The second library is Python OpenCV, an optimized computer vision library suitable for our purposes.

The code also makes use of a function *calcOpticalFlowFarneback*, which allows the user to calculate the optical flow of the screen, and two other functions (*draw_flow*, *draw_hsv*) to see the flow itself and its vectors on the screen.
The optical flow is used to calculate the momentum present in the computer screen and is implemented by calculating the motion of objects in the screen

between two consecutive frames.
This motion is represented by several oriented vectors.
A weighted average of these magnitude vectors with respect to the angles is made.
In this way, the average momentum is calculated also taking into account the direction of the body movement.
Once all these quantities have been calculated, all the values (with a frequency of 10 fps) are sent to MusicVAE and Max8 to check the parameters via OSC (*setOSC_Client* function).

Below are all the calculated quantities, which are sent to MusicVAE or Max8, and which parameter they change:

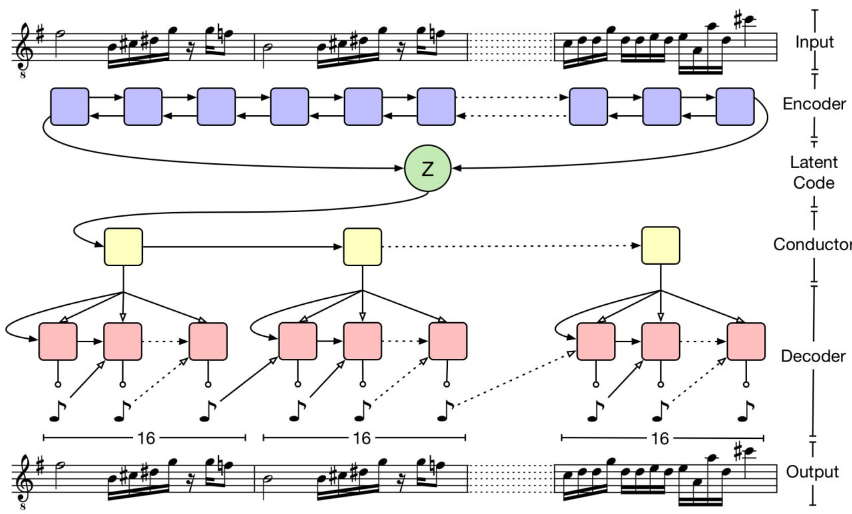
1. Momentum in the screen (via optical flow). Modifies MusicVAE's temperature;
2. Open/Closed right hand. Controls Max8's freeze effect;
3. Right-hand rotation.
4. Left-hand rotation. Controls a filter and a feedback delay;
5. Height of the hands
6. Expansion of the hands. Controls the reverb effect

2.2 Music Pattern Generation

A crucial part of the design of the project is the problem of how to generate harmonic and rhythmic structures that we can control.
So we relied on a generative model that outputs MIDI files from which we can play with personalized instruments.
Also, the generation of an audio signal is very difficult that it is coinvincing in reality that's why we looked for models that output just the pattern structure.
A Variational Auto Encoder (VAE) is perfect for our purpose since can be controlled by sampling a latent vector and it is a generative model able to create new data.
For these reasons, our choice fell on MusicVAE.

2.2.1 MusicVAE

MusicVAE is a machine learning model that lets us create a melodic and rhythmic that we can condition through some hyperparameters.
This model is the best to accomplish the goal of this project since it is able to generate multitrack MIDI files by decoding a vector sampled from a learned latent space.
MusicVAE has a two-layer bidirectional LSTM network for the encoder and a novel hierarchical RNN for the decoder.



In particular, we relied on the multitrack extension of the MusicVAE model able to handle up to 8 tracks.
A novel event-based track representation that handles polyphony, micro-timing, dynamics, and instrument selection.
It also introduces chord-conditioning to a latent space model which is important for our purpose since we can control it.
Magenta proposes some pre-trained models, for the demo we used "*hier-multiperf_vel_1bar_med_chords*" since produces convincing rhythmic patterns and has various horizontal melodic possibilities.
The model generates a pattern with a duration of 2 seconds played at 120 bpm.

2.2.2 Interpolation

A latent space can be used to interpolate between two musical sequences. Given 2 vectors we can interpolate between them, then for any number of steps using spherical linear interpolation.

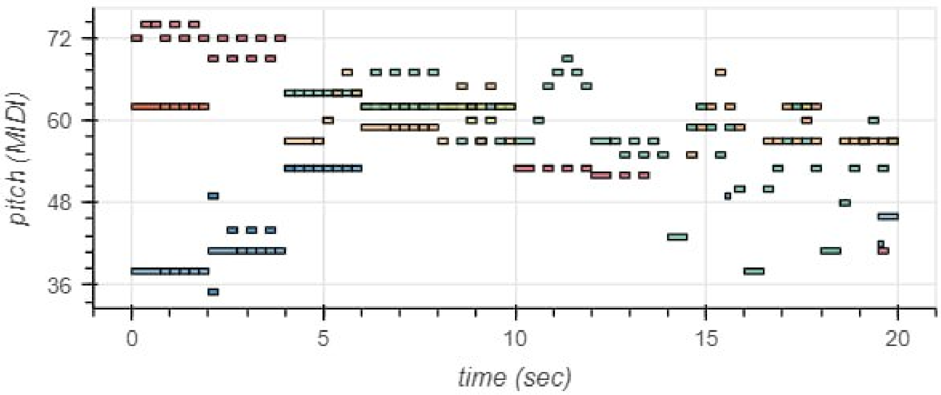
2.2.3 Controllable parameters

- Temperature of the softmax

  Increase the randomness approaching a uniform distribution while increases

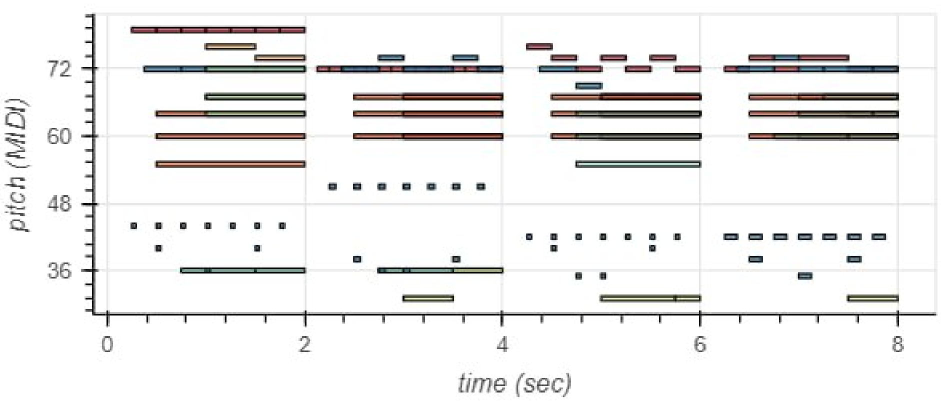
- Chord conditioning

  The same latent vectors are decoded under several different chords.
  For a given latent vector, the instrumental choice and rhythmic pattern



(a) Example of an interpolation in 10 steps between 2 latent vectors that we choose.

remain fairly consistent, while the harmony changes.
In this way, we can concatenate multiple patterns to create a harmonically coherent multi-measure sequence.



2.3 Max8

Max/MSP is a visual programming language for multimedia and music developed by Cycling '74.

It is commonly used for interactive art installations, music composition, and sound synthesis; a versatile and powerful tool for artists, musicians, and multimedia creators to explore, experiment, and create interactive and dynamic audiovisual experiences.

- Introduction
  responsible for real-time MIDI sequence interpretation, playback, synchronization with MusicVAE and integration of audio effects (controlled by MediaPipe body parameters). Moreover, it transmits the resulting audio to the TouchDesigner module that will processes this audio data to generate a synchronized real-time video output.

- OSC Communication and MIDI Sequence Interpretation
  This module interfaces with MusicVAE through OSC, obtaining MIDI sequences representing musical measures. It performs MIDI sequence interpretation to extract musical parameters such as program number, instrument, pitch, velocity, onset, and offset time. These parameters enable real-time sound synthesis. MusicVAE accommodates up to 8 instruments in its sequences, with one of them representing percussion only. To meet this specification, we have programmed 8 instruments and a percussion set. The complex timbral texture emphasizes the structured sound output produced by MusicVAE.

- Real-Time Playback with Audio Effects
  A distinctive feature of this module is its dynamic application of audio effects to the generated sound. These effects are controlled by body parameters obtained from MediaPipe via OSC. This integration enhances the expressiveness and adaptability of the musical output, contributing to a more immersive experience.

- Synchronization with MusicVAE via Markov Chain
  Additionally, the Max/MSP module manages synchronization between the MIDI-based music generation and MusicVAE. It employs a second-order Markov Chain (programmed in MAX) to generate chords, which serve as input to request new MIDI sequences from MusicVAE. This synchronization ensures coherence and harmony between the two modules, enhancing the overall musical output.

- Integration with TouchDesigner for Real-Time Video Generation
  A pivotal extension of this Max/MSP module is its ability to transmit the resulting sound to the TouchDesigner module. TouchDesigner processes

the audio data in real time and generates synchronized video output accordingly. This integration introduces a multisensory dimension to the project, aligning audio and visual elements in harmony.

2.4 TouchDesigner

The music generated by Max8 will then animate an audio-reactive visual implemented in TouchDesigner via OSC.
For the occasion, a graphic has been developed.
The idea is to transform a SOP element into two audio-reactive points' clouds.

To do this the incoming audio is processed through an audio analysis process block and from it we isolate in five different channels the presence of low frequencies, middle frequencies, high frequencies kicks and rhythm.
The first cloud, after appropriate transformations, is repeated 4 times as the "outline" of our screen.
It reacts to the presence of low frequencies and low kicks of any instrument.
Low frequencies scale the dimensions of the image, the kick triggers its movement too.

The second cloud is central and represents the melody.
The presence of high frequencies scales the dimensions of the image, the rhythm triggers the movement of it too. The color of the outline clouds and of the central image is modified by the low and high frequencies with a little component of randomicity controlled by white noise.

3 Challenges

- General
  Synchronization and communication between modules. Create a system with the lowest possible latency while respecting the temporal and computational requirements of the various modules. Choice of the right Deep Learning Model.

- Body detection
  Mediapipe is a powerful tool for body recognition, but the values for our application are not robust. With the implementation of buffers and through adaptive normalization we managed to obtain significantly greater performance and excellent robustness of the code.

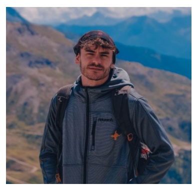
- Magenta
  Manage a 512-dimensions latent space and how to move within it, how to do interpolation and exploit the conditioning of the generation in a meaningfull waydefinitively.

- Max8
  Manage large quantities of packages from Magenta via OSC, manage them and be synchronized with the other modules to play them.

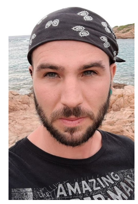
- TouchDesigner
  Create an audio analysis system with correct parameters to make the graphics highly audio-reactive.

4 Students

(a)
Andres Bertazzi
Body Detection



(b)
Riccardo Kluber
MusicVAE



(c)
Nicolò Pisanu
Max8

(d)
Marco Viviani
TouchDesigner
BodyDetection

5 Links

GitHub Repository:
`https://github.com/rickykubler/CPAC_Group_7`