Hand Movement Controlled Grain Delay Effect

Maria Polak
Aalborg Univeristy
Copenhagen, Denmark
mpolak23@student.aau.dk

1 INTRODUCTION

The field of music Human Computer Interaction (HCI) has bloomed during the last decades. The technology, which allows for new ways of human movement capture, has provided means for development of various digital musical instruments (DMI) [9]. Recent advances in computer vision and machine learning have only increased the number of possibilities, creating ways to omit usage of uncomfortable and sometimes expensive devices, such as motion-capture suits. This paper presents hand-movement controlled grain delay effect powered by MediaPipe - a framework for computer vision ML solutions. With the use of a live camera, user can control the way the audio effect changes the sound with only the movement of their hands. The hand motion is captured and interpreted using geometric descriptors to create meaningful and expressive interaction.

2 RELATED WORK

This project aims to explore computer vision usage in the design of gesture controlled digital audio effects.
Looking at the musical part of the system, a similar topic was discussed in [9], were the authors explored gestural controllers in DMIs and sound synthesis. After conducting a critical review of various topics in the domain, authors discussed how important the interaction, the mappings and the balanced analysis are in context designing a new controller.
On the other hand, computer vision, and in particular mediapipe, have been used in many different projects and research studies related to HCI, but not so much in music-HCI. Some of the pure computer interaction examples include using hand gestures as a controller for a Hill Climb Game [4], a tracked robot [8] and a room lightning system [3].
Limited music-related papers often use Mediapipe as a data-sensor, but not exactly as a controller. For instance, in [1] authors used a hand-detection system for improving guitarists' performance.

3 METHOD

In this section the main building blocks of the system are described. The project consists of three main elements, namely: computer vision movement capture, movement descriptors and mapping to granular delay effect parameters.

3.1 Movement Capture

To accommodate the needs of simple and accessible controller, camera video capture was chosen. For that Google's MediaPipe Hand Gesture recognition model [2] was used. The model is trained to detect hand's landmarks (Figure 1) in real time and to classify popular gestures such as closed fist, open palm, thumbs down or thumbs up. The output of the model contains information about landmark's x,y,z coordinates, handedness (left/right) of the detected hand(s), and the label of the classified gesture.

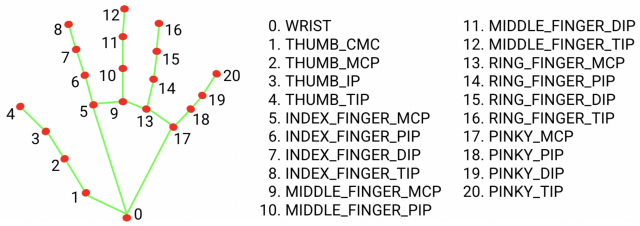

Figure 1: MediaPipe hand landmarks.

3.2 Movement Descriptor

In order to extract meaning from raw data and to address expressiveness of the gestures, geometric movement descriptions were used. Geometrical movement descriptors are used to extract spatial features of the body in relation to environment or itself [5]. In the project the two main types used are displacement and rotation. The first describes the distance of a joint k relative to a position l, which is usually ground, root of a limb or a center of mass. This can be used, for instance, to compute distance between hands or the distance from a starting position. The second usually describes the angular displacement between an orientation of joint k and an orientation joint l. In the project, rotation was used to calculate displacement between orientation of a hand and its default position - pointing upwards.

3.3 Granular Delay Effect

Granular delay effect is used for the sound manipulation in the project. This effect is inspired by granular synthesis, which comes from the idea of quantum sound. The grain delay effect repeats a short portion of a source sound (grain) adding a buzzy character to the sound and creating complex sound events. Grain is a brief acoustical event (10-100ms), smoothed by applying amplitude envelope to it. The effect's main parameters are the grain size (length of the grain in ms) and the grain spacing (number of grains per unit of time and their synchronization). Grain effects often incorporate pitch shifting, which allows for creation of more complex sounds.

Descriptor	Range	Parameter	Mapped Range	Description
Left Hand Y-Displacement	0 to 0.7	Dry Wet	0 to 70 (%)	Further from starting position - more wet
Right Hand Y-Displacement	0 to 0.7	Feedback	0 to 75 (%)	Further from starting position - more feedback
Left-Right Distance	0 to 0.7	Grain Size	257 to 10 (ms)	Further from starting position - smaller the grain
Left Rotation (abs)	0 to 1.5	High Pass Freq	20 to 5.31 (kHz)	Grater the rotation - more filtering applied
Right Rotation (abs)	0 to 2	Pitch	+1 to +12 (semitones)	Grater the rotation - grater the pitch shift

Table 1: Mapping Details

Gesture	Data source	Condition	Sound
mushroom	LiDAR distance	≤ 1.5	Trigger gong
	LiDAR distance		Gong pitch
flat on ground	LiDAR stop	= 1	stop music
lifted from ground	LiDAR stop	= 0	start music
	LiDAR distance		music volume max 1
rotating left	marker rotation	$= -1$	music speed = 0.75
rotating right	marker rotation	= 1	music speed = 1.5
not rotating	marker rotation	= 0	music speed = 1
tilt left	LiDAR tilt		HPF: increase Q and increase cutoff
tilt right	LiDAR tilt		LPF: decrease Q and decrease cutoff
ball free fall	IMU	$magAcc \leq 0.3$ & $magGyro \geq 0.8$	unmute chimes
ball rotation	IMU		tape-warble chimes

4 IMPLEMENTATION

The project is as a joint implementation of Python and Max MSP. Python was a natural choice for camera capture and machine learning part of the system. Unfortunately, real-time audio processing is not a common python task, therefore that part of the system was moved to Max MSP. The source code of the project can be found in the GitHub repository [7].

4.1 Python

The python code covers following sections of the system: camera capture, MediaPipe-based detection, and raw data pre-processing. Camera capture is done with a use of OpenCV framework. The camera is firstly initialized and then the video is read in a frame-by-frame manner. Each frame is then feed to the ML model.
MediaPipe allows for simple parameter control. The model was configured to run in a live-stream mode and to detect two hands. After the detection is complete the results are transferred to data processing functions.
The first step, done by data processing algorithm, is so-called starting position detection. To start the processing, the user is expected to close their hands simultaneously. This gesture saves the starting position, which is the base for the movement descriptors computation. As the next step, algorithm processes the output of the model to extract handedness and landmarks used for mapping. Then it computes the following movement descriptors: left and right hand displacement from the starting position, distance between both hands, rotation displacement of the left and right hand from their default positions. The resulting measurements are filtered by a simple low-pass to assure smooth transition of the data. As the last step, results are send via OSC protocol to Max MSP.

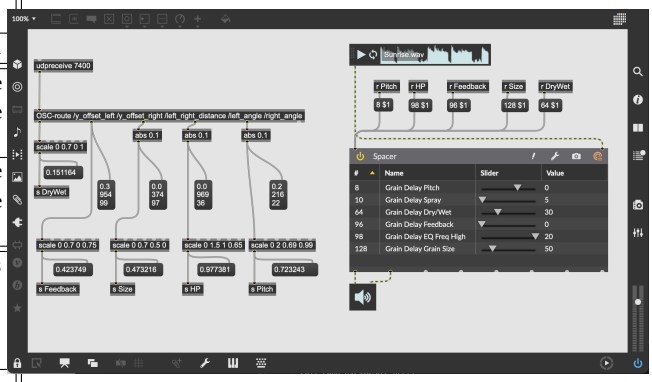

Figure 2: Max MSP Patch

4.2 Max MSP

For audio processing purpose a simple Max Patch was created (Figure 2). The patch consist of two main elements, an OSC receiver and a vst plugin. OSC receiver decodes values sent from python script and scales them to values appropriate for the vst's input. To apply the grain delay effect a free vst - Spacer [6] - was used.
Table 1. contains detailed information about mappings of the descriptors to the plugin parameters and their ranges. It is worth noting that the scaling was fine-tuned after initial experimentation. This is why most of the mapping ranges are smaller than the theoretical maximum value received from python script. For instance, the experimentation showed that it is highly unlikely for 'Left-Right Distance' to reach the maximum value of 1. That is why the dry/wet parameter will be set to the highest every time a value > 0.7 is received.

5 RESULTS

The system introduces enjoyable and expressive interaction. After starting the script and the Max patch, video and audio feedback

are provided to the user (Figure 3). The system works well in a real-time setting, without any noticeable latency being introduced. The mappings are simple enough to control the sound intentionally, especially after a short practice session. At the same time, the nature of the effect allows for interesting sound manipulation even when the user decides to "just go with the flow" and move their body with the music.

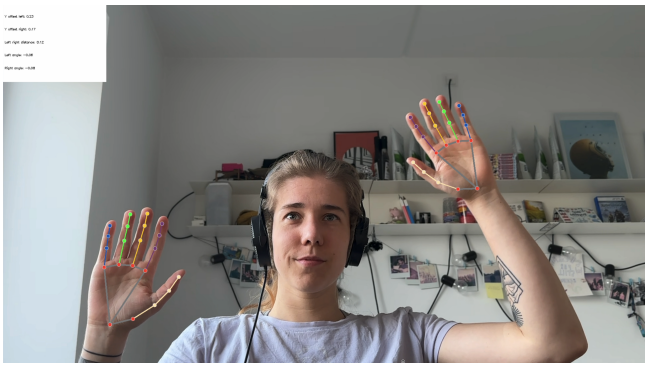

Figure 3: Visual feedback during performance

REFERENCES

[1] Walaa H. Elashmawi, John Emad, Ahmed Serag, Karim Khaled, Ahmed Yehia, Karim Mohamed, Hager Sobeah, and Ahmed Ali. 2023. A Novel Approach for Improving Guitarists' Performance Using Motion Capture and Note Frequency Recognition. *Applied Sciences* 13, 10 (2023). https://doi.org/10.3390/app13106302

[2] Google. 2023. *MediaPipe Gestrue recognition task.* Retrieved May 4, 2024 from https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer

[3] Abdulvahit Karail, Veysel Gokhan Bocekci, and Ismail Kiyak. 2023. Hand Recognition Solution as Conference Room Intelligent Lighting Control Technique. In *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE, 1–5.

[4] Chandan Kumar. 2022. Hill Climb Game Play with Webcam Using OpenCV. *International Journal for Research in Applied Science and Engineering Technology* 10, 12 (2022), 441–453.

[5] Caroline Larboulette and Sylvie Gibet. 2015. A review of computable expressive descriptors of human motion. In *ACM International Conference Proceeding Series*, Vol. 14-15-. ACM, 21–28.

[6] Spectral Plugins. 2024. *Spacer.* Retrieved May 4, 2024 from https://spectral-plugins.com/

[7] SC 2024. *GitHub Repository.* Retrieved May 4, 2024 from https://github.com/mp-smc23/GranularHands

[8] Marthed Wameed, Ahmed M. ALKAMACHI, and Ergun Erçelebi. 2023. Tracked Robot Control with Hand Gesture Based on MediaPipe. *Ai-Khawarizmi engineering journal* 19, 3 (2023), 56–71.

[9] M.M. Wanderley and P. Depalle. 2004. Gestural control of sound synthesis. *Proc. IEEE* 92, 4 (2004), 632–644. https://doi.org/10.1109/JPROC.2004.825882